Technical Report 1065

Boundaries and Topological Algorithms

Margaret M. Fleck

MIT Artificial Intelligence Laboratory

Boundaries and Topological Algorithms

by

Margaret Morrison Fleck

B.A., Linguistics, Yale College, 1982

M.S., Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 1985

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

September, 1988

Signature of Author Margaret M. Fleck
Department of Electrical Engineering and Computer Science
August, 1988

Certified by J Michael Brady
J. Michael Brady
Professor of Information Engineering
Department of Engineering Science, Oxford University
Thesis Supervisor

Certified by Hal Abelson
Harold Abelson
Associate Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ______________________
Arthur C. Smith, Chairman
Committee on Graduate Students

Boundaries and Topological Algorithms

by

Margaret Morrison Fleck

Submitted to the
Department of Electrical Engineering and Computer Science
on August 30, 1988 in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Abstract

This thesis develops a model for the topological structure of situations. In this model, the topological structure of space is altered by the presence or absence of boundaries, such as those at the edges of objects. This allows the intuitive meaning of topological concepts such as region connectivity, function continuity, and preservation of topological structure to be modelled using the standard mathematical definitions. The thesis shows that these concepts are important in a wide range of artificial intelligence problems, including low-level vision, high-level vision, natural language semantics, and high-level reasoning.

A formal framework for manipulating space and boundaries is developed, called cellular topology. Combinatorial methods of representing the topological structure of digitized space are developed and used to develop formal models of the changes in space induced by boundaries. The cellular structure imposed on space restricts the form of representations in ways that are useful for artificial intelligence applications. The cell structure, together with descriptions of the support and error neighborhoods of functions, provides a convenient model for the scale or resolution of representations used in applications.

Two algorithms were implemented for this thesis: an edge finder and a stereo matcher. The edge finder takes advantage of the topological structure of images to distinguish real features from camera noise. The stereo matcher constrains possible matches by requiring that they preserve the topological structure of the image. In informal tests, both algorithms show improvements over previous proposals. The matching algorithm was also used to develop quantitative tests of edge finder performance. Using these tests, the new edge finder was compared to one of the better recent algorithms and performed better than it.

Thesis Supervisor: J. Michael Brady
Title: Professor of Information Engineering
Department of Engineering Science, Oxford University

Thesis Supervisor: Harold Abelson
Title: Associate Professor, Electrical Engineering and Computer Science

Acknowledgements

Although the main work of this thesis has been concentrated in the last few years, the ancestors of the ideas in it date back substantially further. So many people have aided me in developing them that there is no possibility of acknowledging all of you separately. Even if your name doesn't appear explicitly, I probably do remember.

I would like to thank my advisor Mike Brady for his advice, inspiration, and help of all kinds through the past several years. The rest of my committee–Hal Abelson, Ellen Hildreth, and Rod Brooks–not only survived having a thesis dumped on them half-completed, but bounced back to give me large amounts of helpful advice, particularly with the presentation of a thesis that has threatened at every turn to degenerate into a chaotic mass of unconnected ideas.

The artificial intelligence and computer vision ends of this thesis have been greatly enriched by conversations with other members of the three labs I have worked in during the past several years: the MIT Artifical Intelligence Laboratory, the Robotics Research Group at Oxford, and the robotics group at AT&T Bell Laboratories. People who stand out particularly include: Mike Brady, Mike Brown, David Chapman, Dave Clements, Jon Connell, Bruce Donald, Ken Forbus, David Forsyth, Eric Grimson, Jim Little, Alison Noble, and Rich Zippel.

This thesis has also benefited from my experiences dealing with people from outside my own field. The students and faculty of the Yale linguistics department and the linguistics group at AT&T Bell Laboratories gave me a basic grounding in linguistics that even six years in another field hasn't entirely erased. The community of the Science Center at Smith College, particularly my father, George Fleck, helped me understand how mathematicians and physical scientists see the world. David Anick showed me how real topologists build mathematical models. There is an elegance to that work that it may take us some time to emulate in finite resolution mathematics. Finally, repeated questions from Tomás Lozano-Pérez forced me to think longer and harder about methods of testing computer vision algorithms.

Aside from the intellectual contributions, I have also received large amounts of the psychological aid, comfort, and support that is a prerequisite to getting "serious" work accomplished. Over the years, both Mike Brady (my official advisor) and Mitch Marcus (my official mentor) have fished me out of all kinds of trouble. More recently, Jon Connell, Lenore Cowen, Davi Geiger, and Carol Mariens have helped keep me sane through the final struggles of finishing this thesis. Finally, my parents, my sister, and David have been there so much that no specific comment could do any of them justice.

This report describes research that was done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology and at the Department of Engineering Science of the University of Oxford. Support for the Artificial Intelligence Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014–85–K–0124. The author was also supported by the Fannie and John Hertz Foundation and by the AT&T Bell Laboratories Graduate Research Program for Women.

The photographs used in this thesis come from the on-line collections at MIT and Oxford. The stereo image shown in Chapter 10, Figure 8 was obtained

from the University of British Columbia. The other stereo images were taken by David Braunegg and Walter Gillett at MIT. Also, the image shown in Chapter 3, Figure 1 and the room corner image in Chapter 4, Figure 17 were extracted from David Braunegg's stereo pairs. The grey-scale image of parts shown in Chapter 4, Figure 26 was taken by Ellen Hildreth. An extract from this image appears in Chapter 9, Figure 33. The cleaning cloth image shown in Chapter 4, Figure 22 was taken by John Canny. The image of cloth textures used to create Chapter 5, Figure 9 was taken by Harry Voorhees. The picture of a vision researcher shown in Chapter 4, Figure 3 was the work of Harry Voorhees and Walter Gillett. An extract from this image appears in Chapter 9, Figure 35. The rest of the images were taken by me, in some cases working with David Forsyth or Jonathan Connell.

Table of Contents

Chapter 1: Introduction

1. Introduction

Informal discussion of problems in reasoning, perception, or language understanding often makes use of topological concepts. These concepts include connectivity of regions and paths, continuity of functions, and whether two representations have the same topology. These same discussions also refer to entities called "boundaries" and concepts related to them, such as the "edges" of a region. These topological concepts are crucial to certain types of reasoning. For example, connectivity of wires and pipes must be represented in order to solve problems in qualitative physics. In many other areas of artificial intelligence, these concepts have shown some promise as descriptive tools but this promise has not been systematically exploited.

Boundaries are central to any discussion of topological properties, because the intuitive meaning of topological notions changes as the (intuitive) boundaries change. For example, as Figure 1 illustrates, a region of space may be intuitively connected when it is empty, but not connected when it is filled by two objects. The presence of the object boundaries has changed the topological structure of space. How boundary locations are chosen depends on the application at hand. For example, textured patterns on floor tiles are significant for determining the location of the floor from stereo image data, but not for planning motions of objects. Similarly, two wires can be electrically connected without

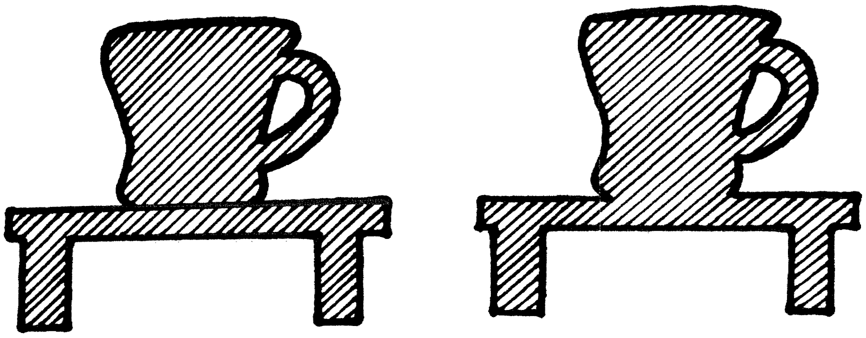


Figure 1. The cup and the table shown on the left are not, intuitively, connected. However, the set of points they occupy, shown on the right, is connected when considered as a subset of empty space.

being physically connected, and vice versa. However, within an application, all of these topological terms are used consistently.

The under-use of topological concepts derives largely from a history of repeated problems formalizing them. There exist standard and well-developed mathematical definitions for "connectedness," "continuity," and "having the same topology." However, the standard definitions for the other topological concepts cannot be applied without a model of boundaries and there are no standard mathematical models for them. Previous attempts to provide formal models for boundaries have not been successful, because the connections between boundaries and topological concepts have not been clearly understood.

In the thesis, I develop two formal models for boundaries. In both models, the presence or absence of boundaries changes the topological structure of space. Given either of these models for boundaries, the informal uses of the other topological concepts can be successfully modelled using the corresponding definitions from standard mathematics. Armed with these formal definitions, I show how topological concepts can be used to provide simpler descriptions of data and improved reasoning algorithms in a variety of domains, such as computer vision, natural language, and naive physics.

Figure 2 shows a sketch map of some major domains within the field of artificial intelligence.[1] The stated goal of the field is to link research in these domains together to form a reasoning system that can interpret sensory data, use this information in manipulating objects, and discuss what it is doing in natural language. Since hard data on human behavior is only available for certain domains, and sometimes only about the form of the input or the output but not both, theories of individual domains are difficult to test unless the domains are linked together. In practice, however, research in different domains has tended to proceed independently, with only weak connections between domains.

In this thesis, I illustrate how descriptions using topological concepts can provide three types of benefits. First, they can provide simpler descriptions of observed data and algorithm behavior in each individual domain. Secondly, the increased clarity can lead to better algorithms. Finally, apparently different phenomena from different domains can be described in a common language. This makes commonalities among the domains clearer, reduces the amount of special-purpose machinery required, and will eventually make it easier to build interfaces between domains.

The thesis involves work of several types. First, the formal mathematical models of space and boundaries are developed. Secondly, two computer vision algorithms that make use of topological properties are implemented. The first algorithm, an edge finder, detects boundaries in digitized (grey-scale) images. The second algorithm performs stereo matching based on the output of the edge

[1] Different researchers might draw slightly different maps.

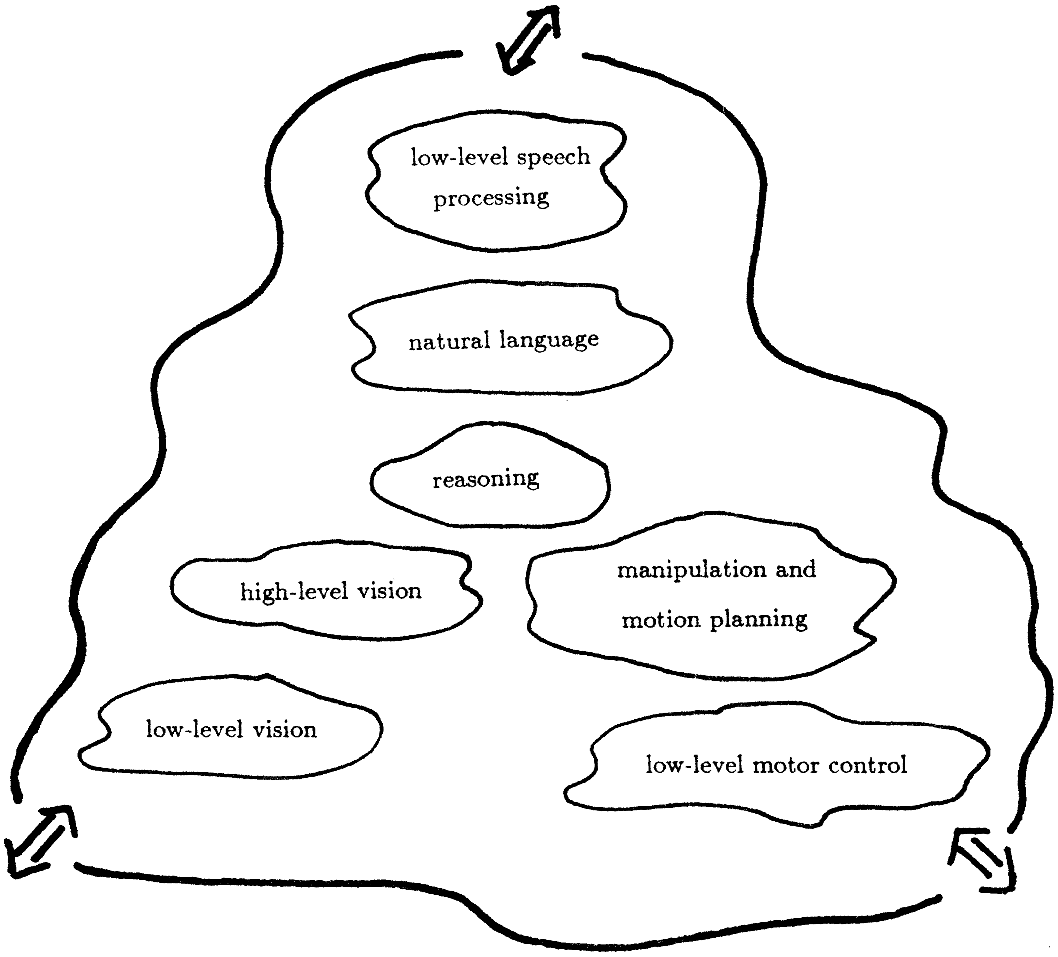


Figure 2. A sketch map of artificial intelligence.

finder. The performance of these algorithms is evaluated and compared to that of previous algorithms. We will see that, by using topological properties, these new algorithms can perform more robustly than previous algorithms.

Two other examples are discussed in detail. The first example consists of data on types of actions and the meaning of tense and aspect in English. The second example consists of work in high-level vision and reasoning that is concerned with representing events in time and objects in space. These examples illustrate how topological concepts are useful in domains other than vision, as well as how the new models of space and boundaries can be used to solve technical problems encountered by previous researchers. Finally, other examples illustrating similar phenomena, such as algorithms for representing region shape, are presented in less detail.

2. Models of topology

Models of topology used by previous researchers fall into two categories: region-based models and boundary-region models. In region-based models, space is segmented into a number of regions, each representing the area or volume covered by an object, event, or other significant part of the scene. The problem of representing the topology is divided into two parts. Topological features of each region, such as the number of holes in it, are determined using the standard mathematical definition of the topology of a subset of a larger space (the subspace topology). Topological relationships among regions, however, are represented using symbolic primitives. The clearest description of this model is by Davis (1984a,b). Similar approaches are also used by Allen (1983, 1984), Allen and Hayes (1985), and Pavlidis (1977).

The region-based approach has several weaknesses. First, difficulties arise in deciding which of two adjacent regions contains the points along their common boundary. Secondly, regions that touch themselves cannot be represented. Thirdly, the symbolic region relations are not related to standard mathematical definitions. Thus, two independent versions of each topological concept are created, one for within a region and one for operations that span more than one region. Finally, the region relations are poorly developed, particularly for 2D and 3D situations. For example, it may be possible to represent whether two regions are connected, but not whether they are connected along one face or along two distinct faces.

In boundary-region models, boundaries are treated as infinitely thin regions. As in region-based models, topological properties of non-boundary regions are computed using the topology that the non-boundary regions inherit as a subset of space. This implies that non-boundary regions are open, which may or not be correct. The primary weakness of this model is that boundaries have a number of special properties, different from other regions, that are accounted for in an *ad hoc* manner. First of these is that they are removed during topological computations. Secondly, it is difficult to assign property values to boundary points in a systematic way if the regions touching at that boundary have different values for the property. The simplest resolution of this is not to assign property values to boundary points. This type of model is proposed by Hayes (1985a) and seems to be the idea underlying many computer vision discussions, such as that in Marr (1982).

The first of the proposed new models of boundaries is similar to the boundary-region model, except that the boundary points are deleted from space rather than being endowed with special properties. Deleting the boundary points accounts for why they are not there during topological computations. Furthermore, since they are not part of space, they are not in the domain of property functions and thus cannot receive values. This model, called the *open-edge* model, is shown in Figure 3 (left). The second new model is similar to the open-edge model, except that new points are added to "close off" the edges of space. You can think of this as splitting boundary points into multiple copies, although the actual mathematical construction works differently. I call this the *closed-edge* model of boundaries.

Although both the open-edge and closed-edge models of boundaries are drawn with space between opposing edges, it is important to realize that this is just a

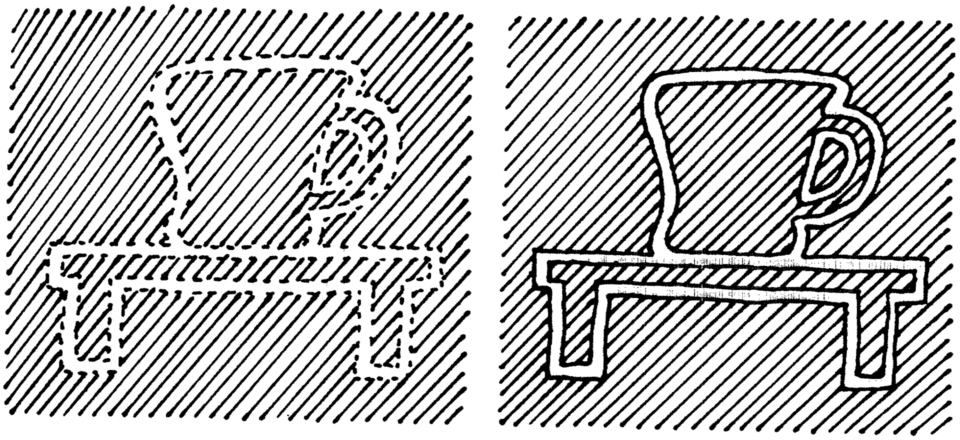


Figure 3. The two proposed models of boundaries: the open-edge model (left) and the closed-edge model (right).

graphic device. Distances in either new space (with boundaries) should be the same as they were in the original space (without boundaries).[2] The right way to visualize these spaces is to think of cutting cloth or paper with a sharp knife. The cut edges are right next to one another and touching one another, but they are no longer connected.

Formal models that look like these pictures are difficult to construct directly. Furthermore, it can be difficult to relate these models directly to some of the representations used in artificial intelligence. Thus, this thesis develops a new set of representations in which space is represented using space-filling cells, illustrated in Figure 4. These representations are based on *regular cell complexes,* a type of structure frequently used in algebraic topology (Munkres 1984, Massey 1980). Using these representations, the topology of a bounded region of space can be completely specified using a finite (and typically concise) description.

[2] Points on the edges of adjacent regions in the closed-edge model are zero distance apart.

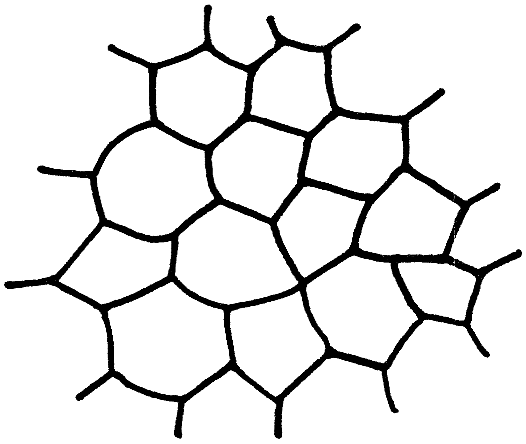


Figure 4. A section of space can be represented using a set of space-filling cells.

Furthermore, boundary locations are easy to specify and manipulate.

The thesis develops two combinatorial representations for these cell complexes, called *incidence structures* and *adjacency structures*. One of these representations, incidence structures, completely represents the topological structure of an arbitrary regular cell complex. The second representation, adjacency structures, is closer in form to representations commonly used in computer vision. However, it is only a complete representation of the topology for a restricted class of cell complexes. The thesis gives the details of these restrictions and a proof that they are sufficient for representing topological structure. (A related, but slightly different discussion for the 2D case is given by Grünbaum and Shephard (1987).) Representations proposed previously typically specify only pairwise connectivity relations between cells (Poston 1971, Pavlidis 1977, Lee and Rosenfeld 1986). This does not, in general, uniquely specify the topological structure of the cell complex.

Using cellular models, most of the work involved in modelling boundaries and defining topological concepts becomes straightforward. The interesting mathematical questions center around how to determine whether two cell complexes are homeomorphic, using only their combinatoric descriptions. Three basic techniques are developed for doing this:

- showing that the complexes have isomorphic adjacency or incidence structures,
- showing that one complex is a subdivision of another, and
- showing that the two complexes are the same, except that boundaries in one have been "thickened."

Combinatorial conditions for subdivision and boundary thickening are fully developed, in this thesis, only for the 2D case. Sequences of applications of these three techniques are sufficient to handle many of the cases required for practical reasoning. In particular, they are essential in working out the details of the stereo matching algorithm.

3. Using cellular representations

Cellular representations form a useful intermediate language for relating existing representations, as shown in Figure 5. For example, they are a convenient framework for describing computer vision algorithms since they avoid both the complexities of point-set topology and the complexities of data structures required by efficient implementations. Previous attempts to relate different representations, such as interval-logics and $\mathbb{R}^n$-like models, have had difficulties because they tried to bridge too large a gap at once. Intermediate representations make it possible to break a difficult transition down into more manageable pieces.

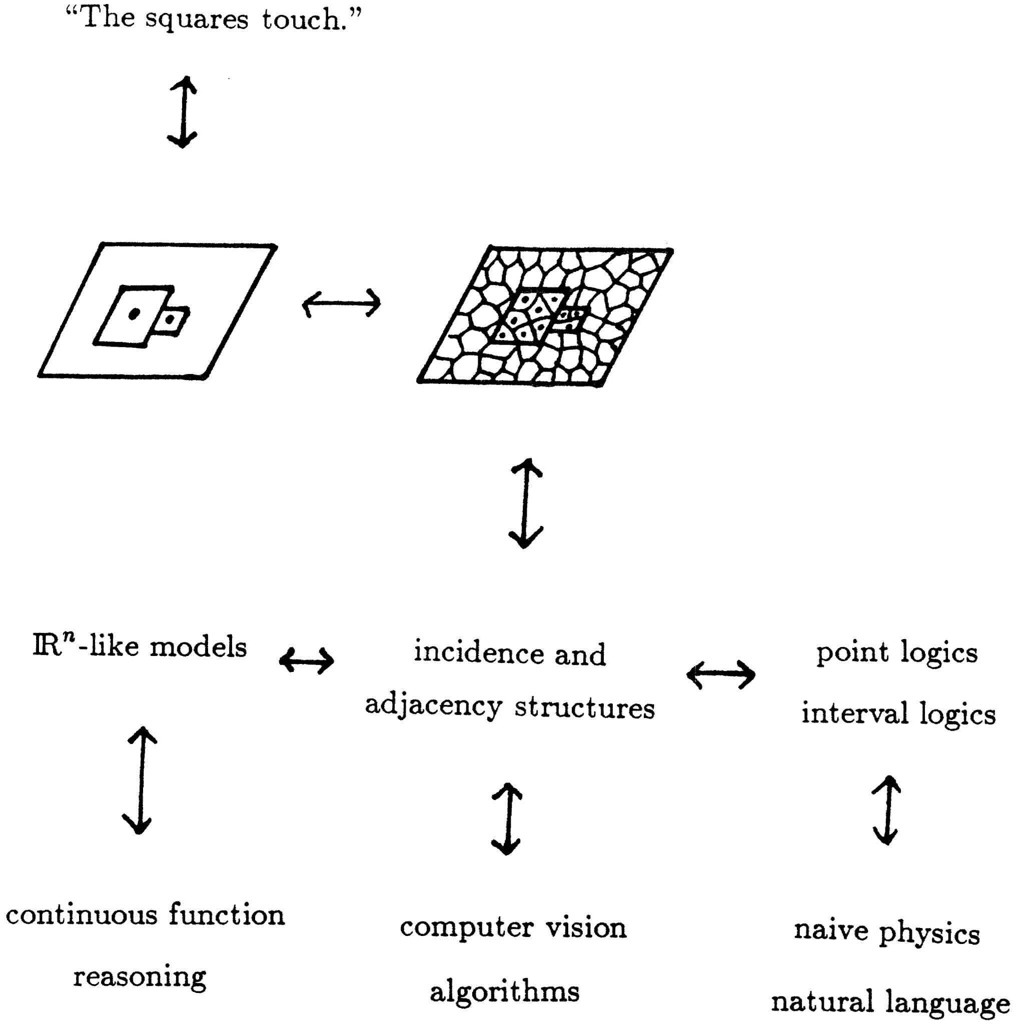

"The squares touch."
$\mathbb{R}^n$-like models
incidence and adjacency structures
point logics
interval logics
continuous function reasoning
computer vision algorithms
naive physics
natural language

Figure 5. Cellular representations are useful in relating representations used in different sub-areas of artificial intelligence.

Cellular representations impose some restrictions on the form of space and boundaries. For example, spaces must be locally like $\mathbb{R}^n$ in order to have cellular representations. This forbids some of the unpleasant spaces that can be constructed in topology, such as the Cantor set or the long line (see Munkres 1975). However, it does not force space to be globally like $\mathbb{R}^n$. Branching models of time have been proposed by some previous researchers (McDermott 1982, Dowty 1979). The new model allows space or time to branch, but the branching cannot be infinitely dense. Boundaries in space also cannot be infinitely dense. The thesis compares these restrictions to those imposed by previous researchers and argues that the proposed restrictions are neither too tight nor too loose.

In order to describe data from any artificial intelligence domain, I also need a model of the "scale" or "resolution" of a representation. The model I use has two components. First, the cellular representations provide a flexible model for digitizations of space. Functions between cellular spaces can also be digitized. That is, the values of the function at all points in a cell are summarized into one value and this value is approximated to the nearest cell. This is illustrated in Figure 6.

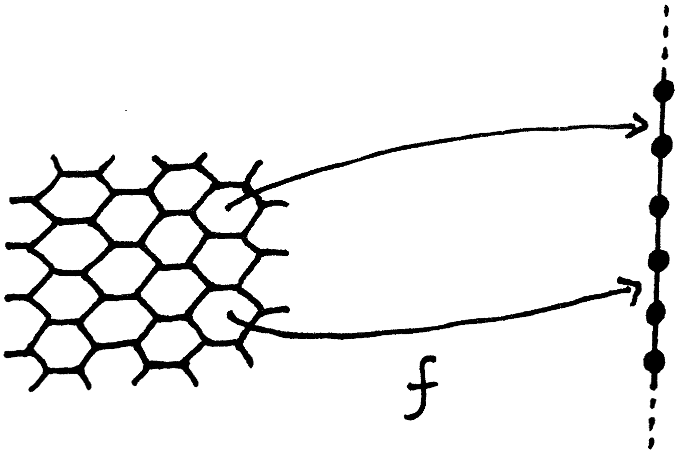


Figure 6. A digitized function maps each cell of the domain to a cell of the range.

The second component required for representing resolution is a description of the support and error neighborhoods of functions, whether they are digitized or continuous. The support neighborhood of a function f at a point x is the set of points whose values are used to derive the value of f at x. For example, the texture periodicity at a point in an image cannot be determined by considering only the intensity at that point. Rather, a texture analysis algorithm must consider intensity values from a neighborhood of the point that is large enough to contain at least two repetitions of the pattern. The error neighborhood of y consists of all the values $f(x)$ that might be reported as y, given the prevailing noise or other sources of errors. In particular, in a digitized function, the error neighborhoods are always at least a cell in size. The support and error neighborhoods of a function are illustrated in Figure 7.

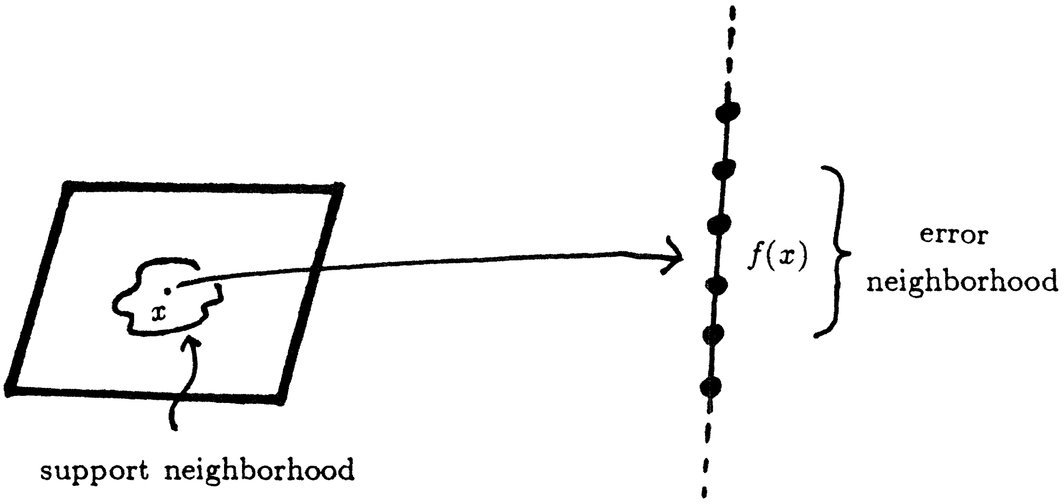


Figure 7. The support neighborhood of a function f at a point x contains all points used to derive the value $f(x)$. The error neighborhood of f at $f(x)$ contains all values that might be reported as $f(x)$.

The thesis illustrates how this model of scale can be used to describe data from different domains in artificial intelligence. Taking notice of digitization is particularly important in computer vision algorithms, where the input data is digitized and some features to be detected are small compared to the cell size. Support regions are also interesting because their shape may be affected by any boundaries present. For example, if support regions in stereo analysis cross depth discontinuities, the stereo results for points near discontinuities are corrupted. The stereo matching algorithm described in this thesis trims support regions for such points so that they do not cross these boundaries. This is a pattern that appears across several domains.

4. Using boundaries and topology

This new model of boundaries and topology predicts a number of patterns that might appear in data and a number of techniques that might be useful in building algorithms. These include:

- explicit references to boundaries,
- requirements that a region be connected,
- restriction of support regions by boundaries,
- abrupt changes in function values at boundaries,
- clustering of abrupt changes in different functions,
- ocurrence of abrupt changes in function values at the same locations as lack of material connectivity, and
- use of homeomorphism as a constraint in matching.

If the model of boundaries and topology is correct, these patterns should appear and these techniques should prove useful in a wide range of problems from different domains.

Explicit reference to boundaries occurs in a number of different domains. Most high-level vision programs, for example, compute descriptions of the shape and arrangement of regions based on the locations of boundaries in the image.

Supporting this type of description, a variety of words in natural language refer to boundaries or to the regions on either side of a spatial or temporal boundary. These include such words as "boundary," "edges," "become," "until," and "touch." These terms show up repeatedly in discussions of naive physics as well as natural language analysis.

In many tasks, a region or interval is required to be connected. Examples of this occur in naive physics, where liquids or electricity can only flow via connected paths (Forbus 1984, Hayes 1985a,b, de Kleer and Brown 1984, Williams 1984). Objects moving through space must also follow connected paths (Lozano-Pérez 1981, 1985). In reasoning about manipulation of objects, it is important to know whether two objects are physically connected or not, because that helps determine whether one object will move if forces are exerted on the other. Causal connections are limited to histories that are connected in both space and time (Hayes 1985b). Reasoning about events in time is often restricted to connected intervals. Representing the meaning of certain constructions in natural language, such as perfect aspect, seems to require that certain intervals be connected. Finally, most high-level vision algorithms require some type of connectivity, either of regions (Brady and Asada 1984, Connell 1985, Fleck 1985, 1986) or at least of extended edges.

Connectivity requirements may also occur in less obvious forms, such as changes in the shape of function support regions near boundaries. For example, when depth discontinuities occur in stereo matching or motion analysis, support regions that cross boundaries generate inaccurate output values. If support regions are required to be connected, more accurate answers can be obtained. This generalization has been noticed by previous researchers (Grimson and Pavlidis 1985, Ponce and Brady 1985), but has proved difficult to implement. The algorithms implemented for this thesis illustrate how support region trimming can be implemented in both edge finding and stereo matching. Similar discussion applies to determining texture properties. Textures include not only patterns of change across space, as in computer vision (Julesz and Bergen 1983, Matsuyama, Miura, and Nagao 1983, Bovik, Clark, and Geisler 1987, Vilnrotter, Navatia, and Price 1986, and Laws 1979) but also patterns of change across time, as in natural language (Taylor 1977, Dowty 1979) and naive physics (Weld 1986).

One immediate consequence of the new definition of boundaries is that continuous functions can have abrupt changes in value across boundaries, because the regions to either side of the boundary are no longer connected to one another (at least locally). Thus, a typical reason for hypothesizing a boundary is to account for abrupt changes observed in some property. For example, boundaries are introduced in computer vision to account for sharp changes in intensity, color, or texture in a camera image. Natural language and naive physics provide examples of abrupt changes in function values over time. For example, the sentence "Michael passed his oral exam" describes a history in which Michael suddenly changes from not yet having passed his exam to having passed it.

The new model of boundaries not only allows sharp changes in function values to occur, but also predicts that they will tend to cluster at a limited number of locations. Suppose we introduce a boundary in space to account for sharp changes in one property, such as region color. The change in topology caused by the boundary allows the color function to jump abruptly across that boundary. In addition, it allows (but does not require) other functions, such as texture, to jump abruptly across the same boundary. Furthermore, the region to the two sides of the boundary is no longer connected. Thus, by hypothesizing a boundary to account for the behavior of one function, the model licenses changes in the behavior of other functions relevant to the same reasoning task, as well as related types of connectivity. This clustering of effects should be observable.

Clustering of abrupt changes in different functions, along with lack of connectivity, occurs in a number of domains. For example, computer vision researchers (Gamble and Poggio 1987, Poggio et al. 1988) have been interested in integrating different types of boundaries, such as color and texture, into one set of boundaries. This only makes sense if the abrupt changes in different properties occur at a common set of locations. Similarly, programs for manipulating objects may deduce boundaries in material connectivity from boundaries in visual input. Finally, naive physics programs (Forbus 1984, Hayes 1985a,b, de Kleer and Brown 1984, Williams 1984; compare also Erdmann and Lozano-Pérez 1987) that predict a course of events from its initial state typically stop whenever any property[3] changes suddenly and re-evaluate whether other properties are still valid. Again, there is a pattern of relatively sparse points of change ("limit points"), with multiple properties changing abruptly at each one.

The final use for topology and boundaries is requiring correspondences in matching to be homeomorphisms. In many practical reasoning tasks, two situations that are to be matched do not have exactly the same size and shape but share a common topological structure. For example, an action must have a particular topological shape and temporal ordering to be described using the present perfect tense, but the length of the interval over which the event occurs is not restricted. In stereo matching, corresponding regions in the two images must be similar in shape and must share the same topological structure. However, they may differ slightly in shape due to the changes in viewpoint and digitization. For similar reasons, a periodic texture tends to match translated versions of itself in

[3] Including whether some process is active or not.

topology, but not in exact shape. All of these cases suggest that matches should be constrained to be homeomorphisms, i.e. to preserve topological structure.

Using the full power of a homeomorphism constraint on matching requires clearer understanding of topology and boundaries than has previously been available. For example, some type of figural continuity requirement has repeatedly been proposed in stereo and motion matching (Mayhew and Frisby 1981, Baker 1982, Grimson 1985, Mutch and Thompson 1985, Koenderink and van Doorn 1976, Callahan and Weiss 1985). Chen (1985) even proposes using full topological structure, based on psychophysical evidence. However, these ideas have only been implemented in weak forms, such as checking connectivity along individual boundaries or via bounds on changes in displacements over an image. Topological features, such as Euler numbers, are sometimes extracted for object identification (Ballard and Brown 1982, Ullman 1984) but purely topological features are weak and poorly behaved under projection and noise. The stereo algorithm and the edge finder implemented for this thesis illustrate how topological constraints can be combined with other constraints to yield effective algorithms. The stereo matching also illustrates how the full power of a homeomorphism constraint can be used in matching.

5. Overview of the applications explored

In this thesis, I explore applications of topology to problems in three domains: low-level vision, linguistic semantics, and high-level vision and reasoning. I have implemented two low-level vision algorithms: an edge finder and a stereo matcher. The performance of these algorithms shows that topological structure can be useful for performing practical tasks in noisy, real-world conditions. I also discuss examples from the other two domains in detail, re-examining previous research in light of the new models of space and boundaries. This section describes these applications briefly.

The new edge finder, called the Phantom edge finder, is based on directional second differences. Its most interesting new feature is its method of noise suppression, which takes advantage of the topological structure of the second difference responses. Previous edge finders eliminate noise by smoothing the image (for example, Canny 1983, 1986, Marr and Hildreth 1980, 1983) or by fitting a rigid model of a boundary to each pixel in the image (for example, Haralick 1980, 1984, Nalwa and Binford 1986, Sher 1987). The new edge finder uses the observation that each second difference edge response covers a connected region in the 2D image. Thus, evaluation of whether the response at one pixel is due to noise can be confined to the connected region of same-sign responses containing that pixel. This idea is originally due to Watt and Morgan (1985; compare also Huttenlocher 1988 and Huertas and Medioni 1986). I have extended their idea to 2D, using the concept of a set of points being *star-convex,* and developed the details so as to make it work on real images. Star-convexity combines metric and topological constraints in a way that preserves the advantages of both approaches.

I have tested the performance of the noise suppression algorithm in some detail, comparing the performance of the new edge finder to that of Canny's (1983, 1986) edge finder. Two features of performance must be evaluated: noise resistance and acuity. Noise resistance is measured by comparing the results of the edge finder on two images of the same (real world) scene. Such a pair of images differ only in having different patterns of random camera noise. By comparing edge finder outputs for the two images, it is possible to assess the stability of the output topology and the amount of fluctuation in output boundary locations. Using the matcher developed for the stereo implementation, these comparisons can be performed automatically.

In the low noise conditions typical of recent camera setups, the Phantom edge finder exhibits consistently better resistence to noise than Canny's edge finder. This holds not only for images differing in noise, but also for images that have been translated and thus exhibit differences in digitization. Under high noise conditions, small amounts of smoothing make a substantial difference in Phantom's performance. Without smoothing, Phantom's stability is very close to that of Canny's edge finder with mask size 8. Using smoothing, its performance is comparable to that of Canny's edge finder with mask size 12. In all cases, the amount of fluctuation in boundary locations is small and shows no substantial difference between the two algorithms.

Even when the edge finders have output of comparable stability, however, a noticeable difference in output resolution is apparent. A second series of tests attempts to characterize these differences in resolution precisely, using simple synthetic images. The two edge finders show similar ability to resolve closely-spaced boundaries, comparable to human performance. There is, however, a substantial difference in performance on boundaries with high curvature and on boundary intersections. The Phantom edge finder resolves boundaries well in these cases, though at the cost of generating spurious boundaries on staircase-like patterns of intensities. Canny's edge finder performs poorly on these images, breaking intersections and sharp corners and introducing spurious boundaries. It also generates spurious responses on ramp-shaped intensity profiles, such as those produced by smooth shading. These patterns of behavior are confirmed with finely-textured extracts from natural images.

Topological structure is also used to build a new algorithm for matching two images. This algorithm is given two images and an approximate alignment between them. The matcher first adjusts the alignment so as to create a correspondence between the images that is a homeomorphism, i.e. preserves topological structure, and that preserves the edge finder's dark and light labels. This adjustment is done using a small set of operations that move boundaries in an image without changing its topological structure. Proving these operations correct is an interesting demonstration of the power of the mathematical framework. After adjustment, the matcher reports which areas of the images could be made to match successfully, it evaluates how good the match is about each cell, and it describes the amount and direction of boundary motion used to achieve the match.

The image matcher can be used for a variety of tasks in low-level vision. It is used in the edge finder evaluations to distinguish stable features from noise and performs this task very reliably. It is also used in the second major implementation for this thesis: a stereo matching algorithm. Finally, I have experimented with using this matcher for analysis of texture periodicity, motion sequences, and combination of edge finder results from different scales. In all cases, the results are very promising.

The implemented stereo algorithm uses a relatively standard control structure to test the new matching algorithm. The algorithm works from coarse scales to fine scales, using coarse-scale results to guide fine-scale analysis. At each scale, it generates a series of candidate translations of one image against the other. The two images are matched at each translation, using the new matching algorithm, and the best matches are chosen over all candidate translations. The computed disparity field is used to adjust the relative positions of the two images prior to computation at the next scale.

The new stereo algorithm shows two improvements in performance over previous algorithms. First, its measure of matching strength is more robust than those used by previous stereo algorithms (such as Grimson 1981, 1985, Mayhew and Frisby 1981, Pollard, Mayhew, and Frisby 1985, Baker 1982, Nishihara 1984, Medioni and Nevatia 1985, Ayache and Faverjon 1987). This allows the new algorithm to tolerate larger search neighborhoods without becoming confused about the correct match. In particular, the new algorithm can handle substantial vertical disparities, because it can tolerate the multiplicative increase in the size of the search space caused by considering the possibility of vertical misalignments. This ability is extremely important, as exact vertical alignment of stereo images is difficult to achieve in practice and humans are relatively tolerant of vertical displacements.

The second change is that, in the new stereo algorithm, the computation of matching strength and disparities is confined to the region for which a correspondence is established. Because of this, the new algorithm can return a dense depth field with less smearing of depths across depth discontinuities than in previous algorithms. The stereo algorithm has been run on a variety of synthetic and natural images to test its performance and demonstrate these improvements.

Finally, the thesis contains detailed discussion of examples from natural language and naive physics. This discussion largely focuses on re-working examples from previous research so as to show how technical problems can be eliminated, using cellular representations and the new models of boundaries, and to highlight features of interest to the main points of this thesis. The natural language examples center around how to represent different classes of actions and how these representations interact with representations of tense and aspect distinctions in English. My description of this data is based on work by Dowty (1979), Woisetschlaeger (1976), and Johnson (1981), which is in turn based on a substantial body of previous research. I show how the new model of boundaries solves several technical problems encountered in describing this data, including how to represent sharp changes in properties over time and how to distinguish states from actions. I also show how topological connectivity may be useful in describing the meaning of certain aspect forms and certain temporal connectives.

The final body of data comes from work in high-level vision and reasoning. Researchers in this area (e.g. Forbus 1984, Hayes 1985a,b, Allen 1983, 1984, McDermott 1982, Brady and Asada 1984, Lozano-Pérez 1981, 1985) have encountered technical problems similar to those in natural language semantics. However, these phenomena appear not only in 1D temporal situations, but also in 2D and 3D spatial situations. Again, I show that the new models can avoid these technical problems. I also discuss how topological properties, such as connectivity, are important in designing reasoning algorithms and I show how cellular models impose constraints on representations that make them a better match to data available from real measurements.

6. Roadmap

The rest of this thesis breaks down into four groups of chapters. Chapters 2 and 3 provide a more detailed introduction to the formalism of cellular topology (Chapter 2) and the domains to which it is applied (Chapter 3). These two chapters are crucial to understanding the rest of the thesis and should be accessible to all readers.

The next five chapters discuss the applications in detail. Chapter 4 presents the edge finder. Chapter 5 presents the image matching algorithm and Chapter 6 discusses how to use it in stereo matching. Chapter 7 discusses the natural language data and Chapter 8 discusses high-level vision and reasoning examples.

While there are some inter-dependencies between these five chapters, they are designed to be read independently.

Chapters 9 and 10 present the results of testing the edge finder and stereo implementations. The edge finder testing procedure depends on the matcher described in Chapter 5. However, both these testing chapters can be read independently. Some readers may find it useful to skim through Chapter 9 while reading Chapters 4 and 5, and Chapter 10 while reading Chapter 6. These two chapters consist primarily of pictures and graphs illustrating algorithm behavior.

Chapter 11 presents the details of the mathematical development and compares my formalism for representing digitized spaces to previous proposals. This chapter assumes familiarity with point-set topology, as well as some knowledge of algebraic topology. However, the rest of the thesis is comprehensible without it. Finally, Chapter 12 gives a summary of the main results, draws conclusions, and suggests plans for future research.

Chapter 2: Cellular Topology

1. Introduction

In this chapter, I describe the mathematical formalism used in this thesis, called *cellular topology.* This formalism is used to define region connectivity and function continuity, which are needed to implement the edge finding and stereo matching examples described in Chapter 1. This presentation is informal, stressing how cellular topology can be used in designing algorithms and comparing it to formalisms used in previous research. The definitions and lemmas used in this chapter are presented formally in Chapter 11.

As we saw briefly in Chapter 1, topological properties such as connectedness are affected by the presence of boundaries. In this chapter, we see how boundaries change the topology of space and how these changes affect reasoning algorithms. In addition to changes in connectivity, we see changes in the behavior of continuous functions, changes in what types of continuous correspondences between situations are possible, and changes in the shape of support regions used in computing function values.

2. Cell complexes and boundaries

In cellular topology, space is represented using regular cell complexes. These mathematical structures are sets of space-filling cells, such as the ones shown in Figure 1. Imposing this cell structure on space makes it easy to specify the

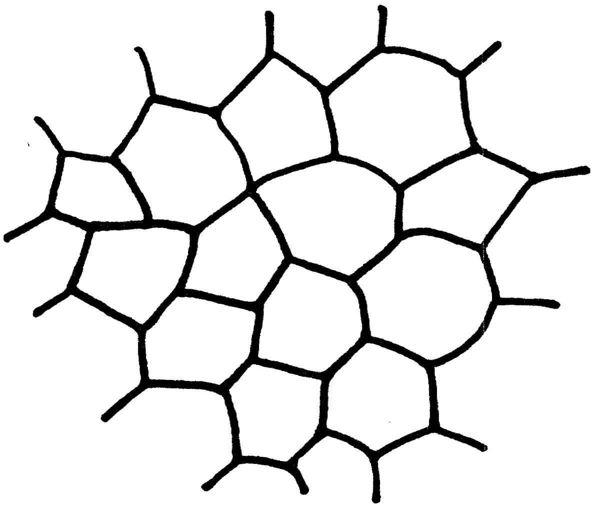


Figure 1. A set of space-filling cells.

locations of boundaries in space. The cell structure and boundaries determine the topological structure of the situation being represented.

A cellular representation of a situation consists of two parts: a description of the cell structure and a specification of boundary locations. The structure of a cell complex is specified as a list of the N-dimensional cells in the complex, together with a list of all sets of cells (*adjacency sets*) that share a common face. I refer to this description as the *adjacency structure* of the set of cells. Figure 2 shows the adjacency structure for a small cell complex.

In cellular topology, boundaries are simply a designated collection of adjacency sets. For example, Figure 3 shows how boundaries might be added to a cellular representation of space to delimit the edges of a cup and the table on which it is sitting. Boundaries can be placed either between cells or on cells, depending on which adjacency sets are chosen. On-cell boundaries are created by marking single-cell adjacency sets as boundaries. Inter-cell boundaries are created by using only multi-cell adjacency sets. Cells belonging to boundary

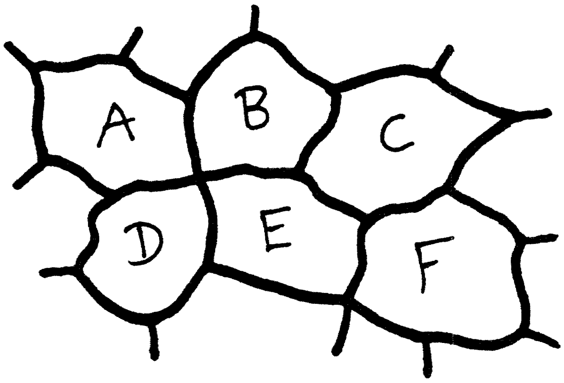


Figure 2. In this cell complex, there are 17 adjacency sets involving only the cells A, B, C, D, E, and F. The adjacency sets $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{E\}$, and $\{F\}$ are 2-dimensional. The 1-dimensional adjacency sets are $\{A, B\}$, $\{A, D\}$, $\{B, E\}$, $\{D, E\}$, $\{B, C\}$, $\{C, F\}$, $\{C, E\}$, and $\{E, F\}$. The 0-dimensional adjacency sets are $\{A, B, D, E\}$, $\{B, C, E\}$, and $\{C, E, F\}$.

adjacency sets are called *edge cells.*

For the cell complexes used in this thesis, each adjacency set uniquely designates either a cell or a common face or vertex shared by several cells. Section 8 describes the conditions necessary to make this true. Under these conditions, the adjacency structure of a set of cells fully specifies the topological structure of the space they fill (see Chapter 11 for details). This is not very exciting, because most sets of cells used in this thesis are topologically equivalent to rectangular sub-sections of the plane and thus do not have interesting topological structure. The interesting point is that *this equivalence allows us to specify precisely how the topological structure of space is changed when boundaries are added to it.* This means that the topological structure of situations such as the one shown in Figure 3 is completely specified by the combination of the adjacency structure and the boundary markings.

Chapter 11 develops two models for how adding boundaries changes the un-

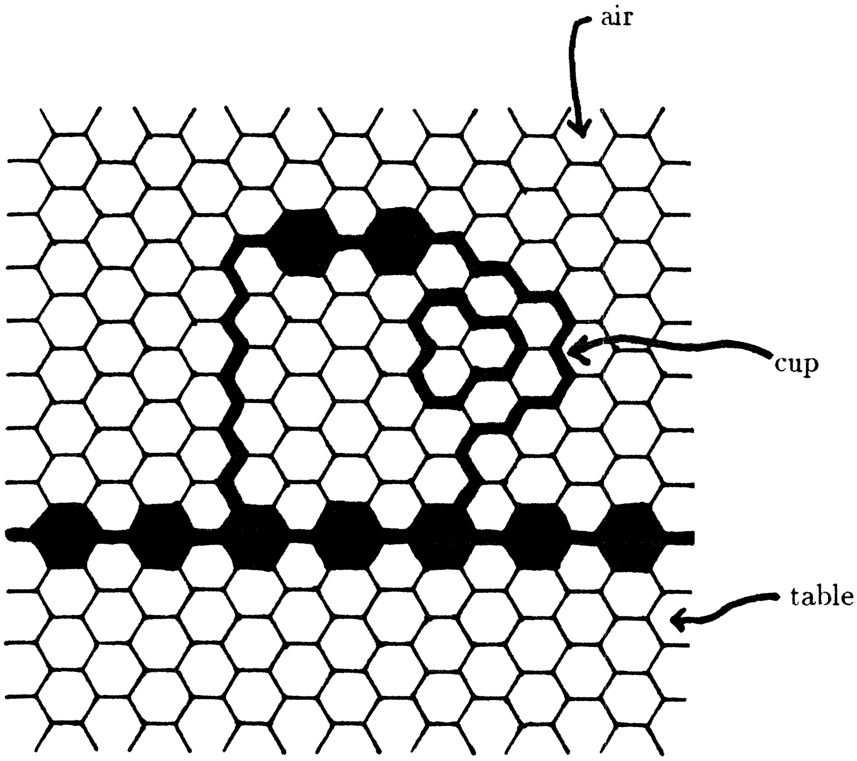


Figure 3. Adding boundaries to space at the edges of a cup and a table.

derlying space. The two models are illustrated in Figure 4. In the *open-edge* model of boundaries, points corresponding to boundary adjacency sets are simply deleted from space. The regions to either side of each boundary remain right next to one another, but they are no longer connected to one another. Figure 4 only shows space between the two sides of the boundaries because that is necessary in order to show the topological structure intelligibly.

The second, *closed-edge*, model of boundaries is similar, but points are added to "close"[1] the edges of the new space. The new points on either side of the boundary are right next to one another, but distinct. The formal details of

[1] I.e. make them look locally like closed subsets of $\mathbb{R}^n$. Local neighborhoods near the boundaries are topologically both open and closed in both models.

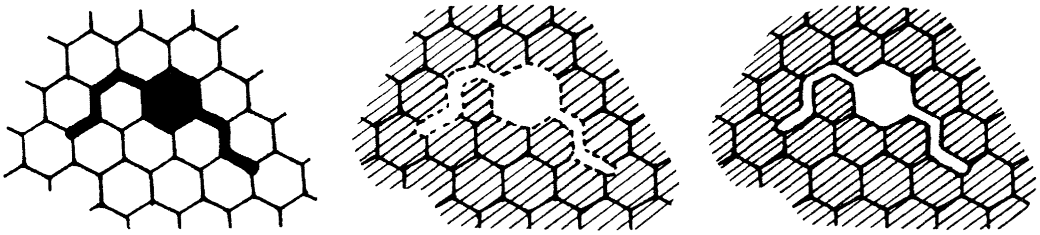


Figure 4. Left: a cell complex with boundaries indicated by thick lines. Middle: the open-edge model of these boundaries. Right: the closed-edge model of these boundaries.

this construction are given in Chapter 11, Section 6 and are somewhat involved. This second model can only be constructed if the cell complex meets additional conditions.

For most practical purposes, these two models of boundaries behave in the same way. Since few[2] applications make use of the special features of either model, there is little reason for choosing between them. The important point to note is that both models of boundaries modify the topological structure of space. For example, when boundaries are added, regions that used to be connected to one another are no longer connected. In later sections, we see that these topological changes have far-reaching consequences.

In this section, I have defined cell structures and how to add boundaries to them. In this thesis, I am primarily interested in the effects of the topological changes caused by adding boundaries. As Figure 5 shows, the topological structure of a situation is independent of the cell structure used to represent it. The cell structure serves two purposes. First, it makes models of space and boundaries easier to specify and manipulate. In particular, as we see in Section 9,

[2] Later chapters discuss the cases that I know of. None of them provide conclusive evidence in favor of either model.

the cellular framework constrains the form of space so as to avoid unwanted pathological cases. Secondly, cell structures provide a formalism for describing digitized functions, as described in Section 6.

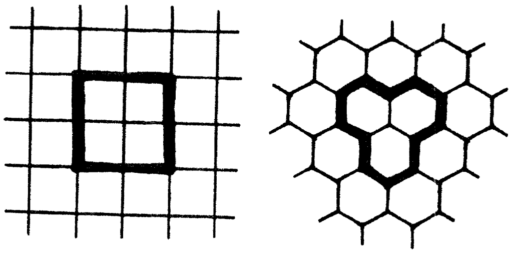


Figure 5. The same situation can be represented using different cellular structures.

3. Paths and connectedness

The most familiar topological properties are those involving path and region connectedness. In this section, we see how the standard mathematical definitions for these concepts can be re-phrased in cellular terms. We see how adding boundaries affects these properties. Finally, we see how connectedness can be combined with rough metric information to yield the notion of a *star-convex* neighborhood, which is used repeatedly in the practial applications described in later chapters.

The definition of *connectedness* in cellular topology is based on the notion of a connected path. A *(connected) path* is a finite ordered list of cells such that adjacent elements in the list share a common non-boundary adjacency set. If A is the first cell in the list and B is the last, the path is said to *connect* A and B. Figure 6 shows examples of paths and non-paths.

In standard mathematics, a path between two points a and b in a space X is a continuous map f from a connected interval $[p, q]$ of the real line into X, such

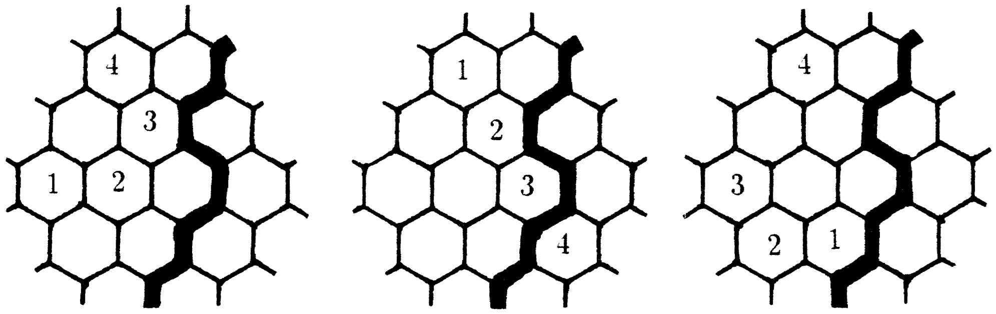


Figure 6. Left to right: a connected path, a set of cells that is not a path because it is broken by a boundary, a set of cells that is not a path because adjacent elements do not share a common adjacency set. Boundaries between cells are indicated by thick lines.

that $f(p) = a$ and $f(q) = b$. Such a continuous map cannot cross boundaries, in either the closed-edge or open-edge model, because there is no way to make the points correspond. The cellular definition of a path cannot refer to individual points, but only to cells. However, the two definitions are otherwise equivalent. Specifically, there is a cellular path between two cells A and B if and only if there is a point-wise path connecting some point (equivalently, any) in A to some (equivalently, any) point in B.

Using the definition of connected paths, we can now define what it means for a region to be connected. A region in a cellular representation is any set of cells. A region X is connected if there is a path connecting every pair of cells in X that uses only cells in X. The corresponding standard definition requires that there be a path between any two points in X that uses only points in X.[3] Thus, the cellular definition of region connectedness is equivalent to the standard definition. Figure 7 shows examples of connected and non-connected regions.

There are a few types of reasoning that can be done using connectivity infor-

[3] Path-connectedness and connectedness are equivalent for these spaces, because they are locally path-connected.

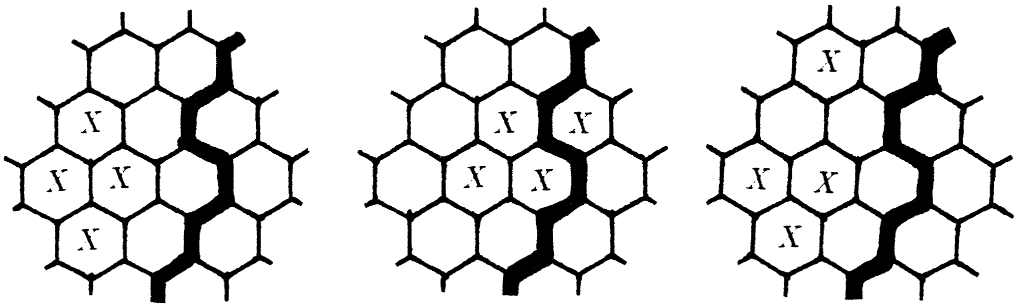


Figure 7. Left to right: a connected region, a region that is not connected because it is cut by boundaries, a region that is not connected because it consists of two separated pieces.

mation alone. For example, suppose that I pour water into a coffee maker, shown schematically in Figure 8. If the machine is functioning properly, the input and output are connected by a tube and thus the water must eventually come out the bottom of the machine. If the water does not come out, something must be blocking the tube so that they are no longer connected.

Figure 8. If water is poured into the top of a coffee maker, it will eventually flow out the bottom, because there is a tube connecting the water input and the water output.

Other types of reasoning, however, require combinations of connectivity and metric information. In later chapters, we see that the concept of ***star-convexity*** is useful in a number of applications. In standard mathematics, a region is *star-convex* about a point x if every point y in the region is connected to x by a straight path that is entirely contained in the region. Remember that these paths cannot cross boundaries, so the presence or absence of boundaries changes which regions are star-convex. Figure 9 shows examples of star-convex and non-star-convex regions.

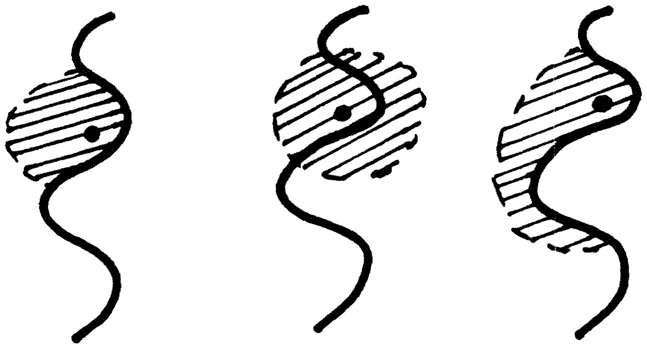


Figure 9. Left to right: star-convex region, region that is not star-convex because it crosses boundaries, region that is connected but not star-convex about the marked point.

In cellular representations, paths are rarely exactly straight. Thus, a cellular region is considered star-convex about a cell A if any cell in the region can be connected to A using an *approximately* straight path entirely contained in the region. Which paths are considered approximately straight depends both on the shape and arrangement of the cells and also on the application at hand. The algorithms implemented for this thesis use rectangular cell arrangements. They considers a path between cells A and B to be straight if it uses the minimal number of cells of any path connecting A and B and, among the paths containing the minimum number of cells, it uses a minimal number of diagonal moves. Notice that there may be more than one approximately straight path connecting a given pair of cells. The definition of star-convexity requires that one path of the appropriate type exist and does not depend on whether it is unique.

The applications described in this thesis use star-convex neighborhoods that are *maximal,* relative to some bound r on the radius of the region. What this means is that each cell in the neighborhood about a cell A must be connected to A via a path of length at most r. The largest star-convex neighborhood meeting this condition is then used. Figure 10 illustrates the maximal star-convex neighborhood about several cells. Notice how the shape and size of these neighborhoods depends on the presence of nearby boundaries. Thus, if a computation uses maximal star-convex neighborhoods, its result changes as the boundary locations change.

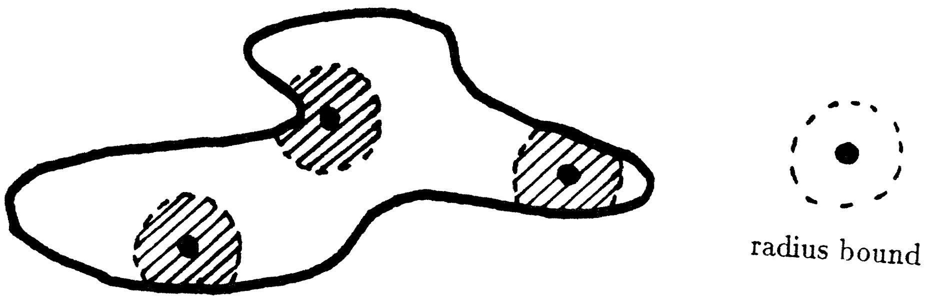


Figure 10. The maximum star-convex neighborhoods of several cells.

4. Continuous functions

Changes in the topology of space affect not only region and path connectivity, but also the behavior of continuous functions. Continuous functions appear in two contexts in practical reasoning: assigning property values to locations in space and matching two situations in space. The two cases behave slightly differently, because the matching problem requires functions not only to be continuous but also to have continuous inverses. In this section, I discuss how boundaries affect the behavior of continuous property functions. The matching problem is discussed in Section 5.

Boundaries in space or time are often hypothesized to account for abrupt changes in property values. Consider a cup sitting on a table, shown in Figure 11. Light intensity, color, texture, and material properties vary smoothly within the cup and within the table, but change abruptly at the transition between the two objects. We can account for this behavior by modelling all of these properties as continuous functions, but putting a boundary in space between the cup and the table.[4]

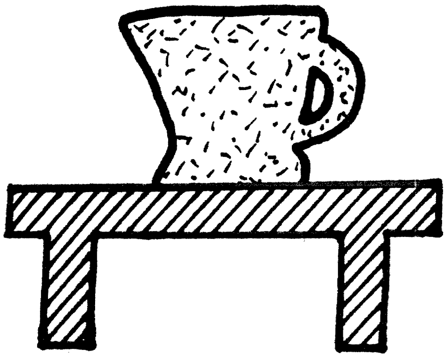


Figure 11. A cup sitting on a table is not connected to the table. Furthermore, many material and visual properties change at the transition between the table and the cup. These effects can be explained by postulating a boundary separating the two objects.

[4] One would also want boundaries separating each object from the air around it.

Placing a boundary in space, such as between a cup and the table it rests on, allows continuous functions to have arbitrary changes in value across the boundary. Recall that a function f is continuous if the inverse image of any open set is open. Figure 12 shows a plot of an abrupt change in light intensity across the cup/table boundary and an open interval A in the space of intensity values. In the model with no boundary, the inverse image of A is not open and thus the intensity function is not continuous. For both models of boundaries, however, the inverse image of A is open and the intensity function is continuous. Although the inverse image of A in the closed-edge model looks like a half-closed region of $\mathbb{R}^n$, it is topologically open.[5]

In the same way, continuous functions can change between discrete values across topological boundaries. Consider a student passing an oral exam. This event can be represented by a function from time to a space with two discrete values, as shown in Figure 13. The event divides time into two intervals separated by a boundary. In the first interval, the exam is not yet passed, and in the second interval it has been passed. Since there is no such thing as having "partway passed" an exam, there is an abrupt jump in value between the two intervals.

Thus, there are two ways to model an abrupt change in the values of a property across space or time. Either the function is discontinuous or else there is a boundary in space or time. In this thesis, I assume that all functions are continuous and thus that all abrupt changes indicate the presence of a boundary. This method of modelling abrupt changes has two consequences: (1) clustering of abrupt changes in different functions is easy to model and (2) lack of region connectivity must occur at the locations of abrupt changes in function values.

[5] In $\mathbb{R}^n$, the shape of a region is closely related to whether it is open or closed. However, under more general circumstances, a region can be specified as topologically open or closed, no matter what its shape. See, for example, Munkres (1975).

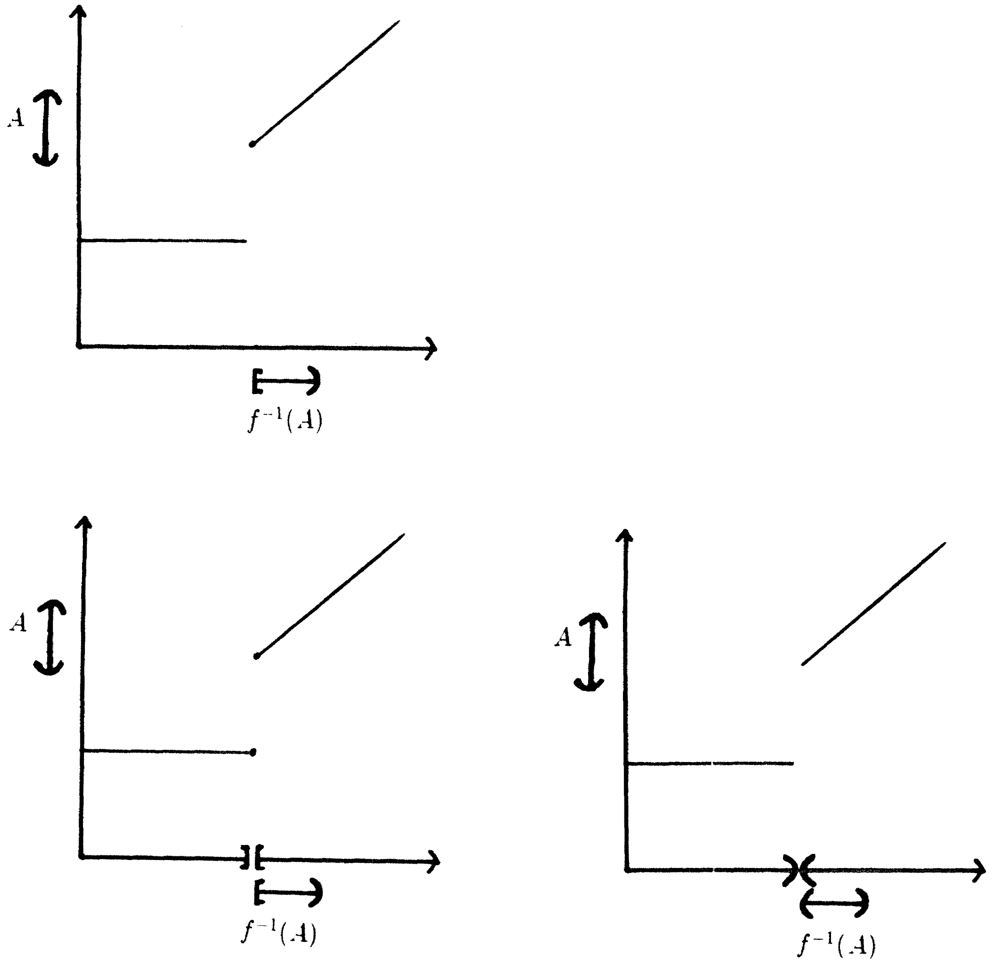


Figure 12. Light intensity for a path across the cup/table boundary. Top: a model with no boundary. Bottom left: a model with the closed-edge model of boundaries. Bottom right: a model with the open-edge model of boundaries.

Adding a boundary to space not only changes the potential behavior of the function that caused it to be hypothesized, but also the behavior of other functions. The change in topology that allows the values of one function to change abruptly also licenses abrupt changes in other functions. Thus, a cluster of apparent discontinuities in many functions can be explained by postulating only one boundary. For example, in Figure 11, many types of properties change abruptly across the cup/table boundary, including color, texture, and material structure.

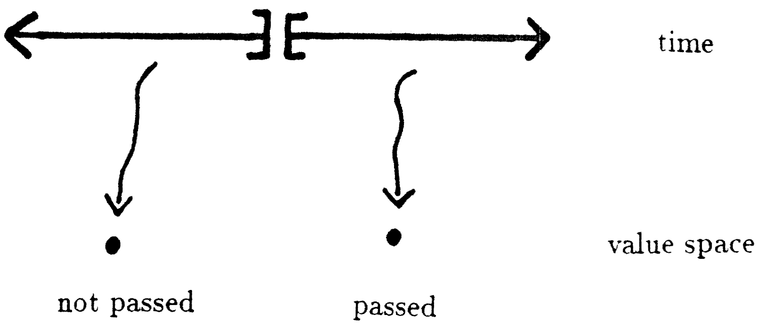


Figure 13. The process of passing an oral exam can be modelled using a function from time to a space containing two discrete values.

If discontinuous behavior were a property of individual functions, a separate explanation would be needed for each function involved in such a cluster.

The second consequence of the topological boundary model is that lack of region connectivity must occur where there are abrupt changes in property values. Suppose, for example, that we hypothesize a boundary between the cup and the table in Figure 11 to explain the change in material between the two objects. According to the definition of connectivity developed in Section 3, the cup is not materially connected to the table. That is, if you lift the cup, the table should not move with it. This prediction is limited to functions and types of connectivity that are relevant to the same task, such as material properties and material connectivity or visual boundaries and visual region connectivity. In this thesis, we will see that both clustering and coincidence of connectivity and changes in function behavior occur in a variety of domains.

5. Same topology

Function continuity appears in a second form in practical reasoning: constructing matches between two situations in space. Matching examples differ from the property functions discussed in the previous section in that matching correspondences must not only be continuous but must also have continuous inverses. Thus, the behavior of these functions is more tightly constained.

In later chapters, we will see a number of applications in which two situations must be matched in a way that preserves topological structure. For example, topological structure can be used to distinguish the two whole chain links in Figure 14 from the damaged chain link. As we will see in Chapters 5 and 6, two views of the same patch of surface from different perspective typically have the same topological structure. As shown in Figure 15, this can be used to constrain the process of matching images from two viewpoints,

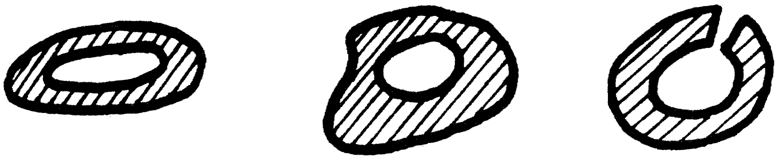


Figure 14. Two whole chain links and a damaged chain link.

Intuitively, two representations have the same topological structure if one can be deformed smoothly into the other. So, for example, the two situations shown in Figure 16 have the same topological structure. This is defined formally in terms of continuous functions. That is, two spaces (with boundaries) have the same topological structure[6] if there is a bijective function from X onto Y that is continuous and has a continuous inverse. Figure 16 shows a continuous correspondence between a cup and a ring.

[6] That is, are homeomorphic.

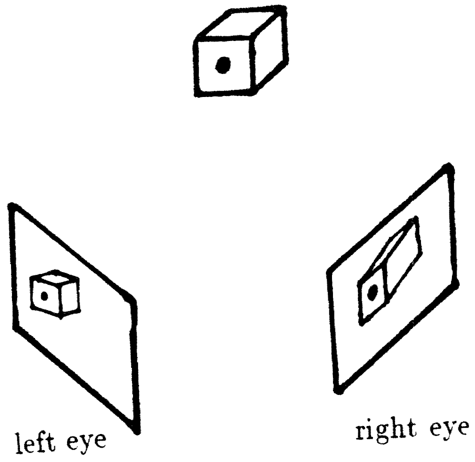


Figure 15. A 3D situation, as seen from the left eye and from the right eye.

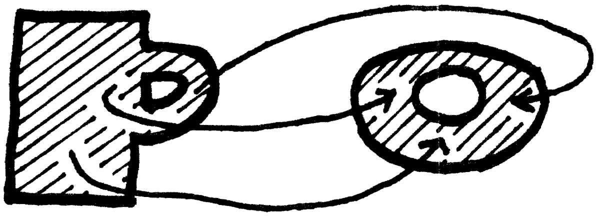


Figure 16. A cup and a ring, seen in 2D projection, have the same topological structure, because they can be matched using a correspondence that is continuous in both directions.

Locations of boundaries play a crucial role in determining what correspondences are continuous. A continuous function cannot map a connected set onto a set that is not connected. Thus, a correspondence that is continuous in both directions can only associate a patch of space that does not contain a boundary with another patch of space that also contains no boundaries. For example, the two situations shown in Figure 17 do not have the same topological structure. In the model presented here, boundaries are not actually part of space, but,

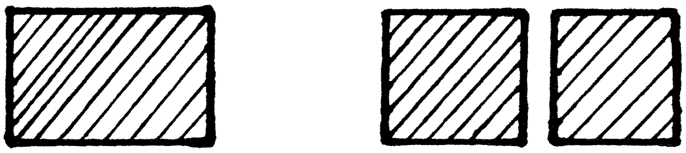


Figure 17. Two situations that do not have the same topological structure.

intuitively, continuous correspondences must match boundaries with boundaries.

This model of matching in terms of continuous functions is a standard mathematical approach, but one that may seem unfamiliar to researchers in other fields. In low-level vision, for example, the matching problem has typically been stated as a problem of matching discrete features, such as short sections of boundary. In high-level reasoning, topological structure is typically approached via topological properties such as the presence or absence of holes. Because the continuous function approach is more general, it can lead to more powerful constraints on algorithm behavior, as we will see in later chapters. It also extends well to cases in which we may only be able to construct a continuous correspondence between subsets of the two situations and in which additional considerations may limit the choice of correspondences.

6. Digitized functions

The adjacency structure and boundary markings represent the topological structure of a situation exactly, even though this topological structure may represent only a limited resolution view of the situation. Representing functions, whether properties or correspondences, is typically not exact. In manipulating functions used in practical reasoning applications, we must consider effects of both digitization and measurement error.

Consider the process of representing a camera image for computer vision analysis. Because a computer can only store finite amounts of information, we cannot store the exact intensity value at each point in the image. Rather, only a finite number of intensity values are stored, each one representing an average over a small patch of the image. Each intensity value is represented with only finitely many bits of precision. We can model this as a mapping between two cellular representations, as shown in Figure 18. The real intensity function maps points in the image onto exact intensity values. The approximation maps cells in the image onto intensity cells. Such approximations are not peculiar to computer vision, but occur in any application that involves interpreting measurements of real situations.

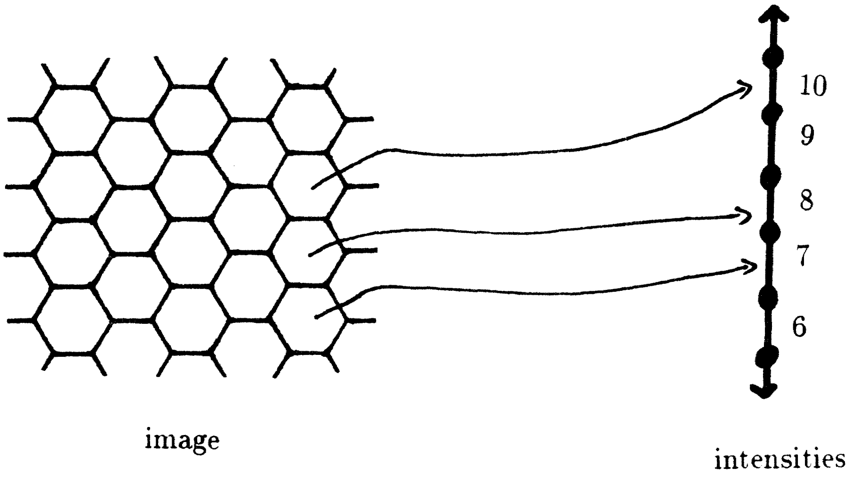


Figure 18. A digitized image.

It is important to realize that digitized functions are not maps between spaces of discrete values, but rather approximations to continuous functions. Suppose that we labelled the image with two discrete values, dark and light, as shown in Figure 19. Whenever a dark cell is adjacent to a light cell in the image, there must be a boundary in the image, because a continuous function on a connected region cannot jump between two discrete values. Adjacent cells in the image can, however, bear different intensity values without there being a boundary in the image, because intensities form a connected space.

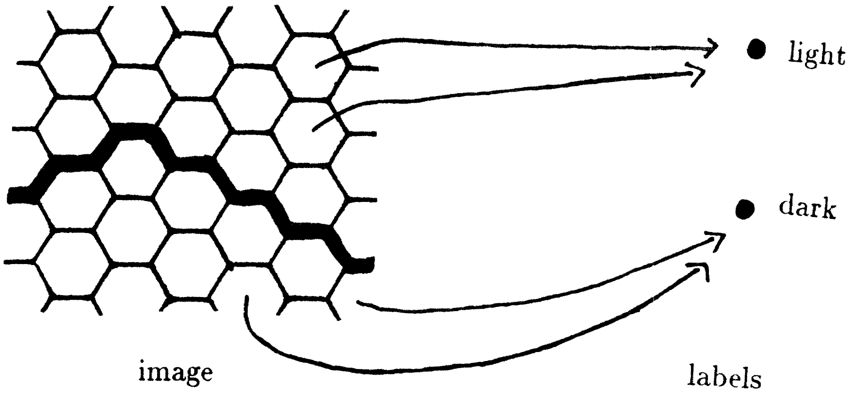


Figure 19. Labelling an image with discrete values.

Except in rare cases, such as functions with discrete values, topological analysis of the raw digitized values does not provide sufficient information for practical applications. Consider first the relationship between the digitized function F and the continuous function f that it approximates. Each digitized value has two associated neighborhoods: a support neighborhood and an error neighborhood. The support neighborhood at a point x contains all the points whose values (from the function f) were used to derive the digitized value $F(x)$. The support neighborhood for each cell must include at least all the points in the cell and often points from other cells. Types of support regions are discussed in Section 7.

The error neighborhood at a point x consists of the points in the range that might be represented by the digitized value $F(x)$. Since the value $F(x)$ is reported only to the nearest cell, the error neighborhood must clearly include all points in the cell $F(x)$, including the boundaries it shares with adjacent cells. Error neighborhoods are typically somewhat larger than this, due to various sources of noise present in real measurements.

In designing algorithms that operate on digitized functions, it is important to be aware of the error neighborhoods associated with the values of these functions. This is particularly important when comparing the values at two cells. Following Poston (1971), I refer to two values as *indistinguishable* if their error neighborhoods overlap. Indistinguishable values could represent measurements of the same underlying value.

Using error neighborhoods, it is possible to deduce the presence of boundaries even when function values form one connected region. Two cells that are adjacent, but not separated by a boundary, overlap along their common face, edge, or vertex. The underlying values for each common point must belong to the error neighborhoods of the digitized values for both cells. Thus, the values at the two cells must be indistinguishable. If a digitized function assigns distinguishable values to two adjacent cells, they must be separated by a boundary.

Algorithms using digitized functions may also be able to take advantage of constraints on the class of continuous functions under consideration. For example, it may be possible to assume that the underlying function satisfies certain bounds on slopes, second differences, or derivatives of various orders. Depending on the application, these constraints may be formulated so as to respect the topological structure. For example, bounds on slopes might apply only to differences taken along *connected* paths. If so, a topological boundary would license apparent violations of these constraints, just as it licenses apparent violations of continuity.

Although constraints on slopes or differences may be formulated as constraints on the underlying function, they often imply similar constraints on digitized approximations to that function.[7] For example, the smoothing and sampling procedures commonly used in computer vision do not increase the magnitude of finite differences. Thus, if a difference of the sampled function exceeds a given bound, the underlying infinite-resolution function must also contain a difference that exceeds the bound. Thus, the presence of boundaries can be inferred from apparent violations of the constraints, even when only digitized approximations to function values are available.

7. Support regions

In the previous section, we saw that cellular approximations to continuous functions may combine information from many points to yield a digitized value for each cell. In most domains, combining information from wide support regions is essential to producing well-behaved approximations. In this section, we see how pathological situations can be created by poor choices of support functions. We also see how wide support can be used for other interesting purposes, such as describing textured patterns, and how support regions can be modified by the presence of boundaries.

It is well-known in computer vision that undesirable behavior can happen if a function is digitized without adequate amounts of smoothing. Because these problems may not be familiar to researchers from other domains, I review them briefly in this section. Consider the striped pattern shown in Figure 20. The top two sampling options in this figure show ways of sampling this pattern with sufficient smoothing. If the sample points are sufficiently dense, the stripes can be resolved, otherwise the pattern looks uniformly grey.

[7] Depending on the support function used in creating the digitization.

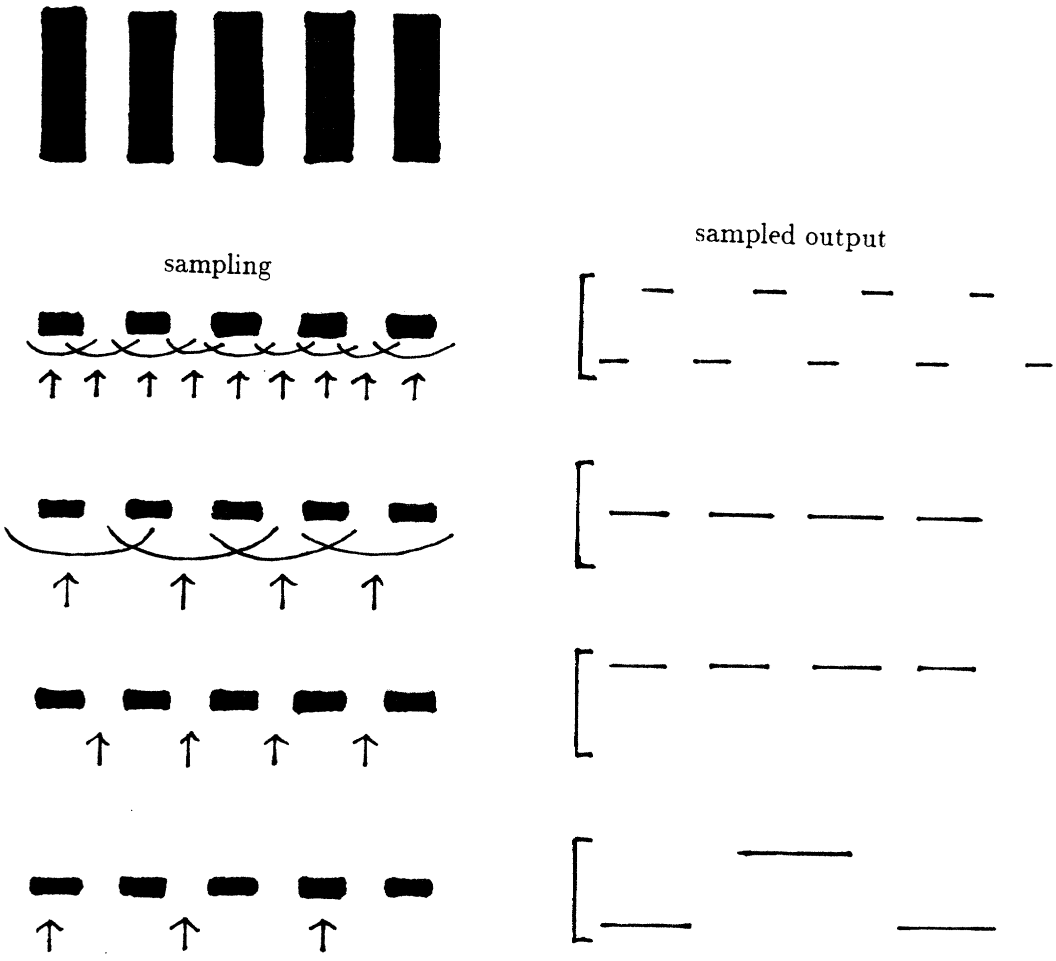


Figure 20. A striped pattern and several ways of sampling it.

The second two options in Figure 20 show sampling with non-existent or inadequate smoothing. Two pathological effects can occur. First, the samples can miss the dark stripes entirely, resulting in a representation of the pattern as entirely white. Secondly, the samples can pick out some of the stripes, but not all of them, resulting in a representation of the pattern with stripes, but at the wrong density. I refer to the first effect as *drop-out* and the second effect as *aliasing.* In both cases, the representation can change completely if the sample locations are translated relative to the pattern. This instability is a problem for most applications.

Thus, wide support regions are needed for producing well-behaved digitized representations. Wide support regions can also be used to capture texture properties that are only defined for extended regions of space. Consider the striped pattern from Figure 20. In order to decide that a given cell is in a region of striped texture, it is necessary to examine a neighborhood of that cell that is big enough to contain several stripes. In later chapters, we see other examples of properties that can be defined at every cell, but require wide support regions.

In many applications, the shape of support regions can be changed by the presence of boundaries. Consider the textured situation shown in Figure 21. In order to describe the texture about each cell reliably, the support region about each cell should be adjusted so that it does not cross sharp changes in texture. If the texture boundaries can be identified, these adjusted support regions can be computed as the maximal star-convex neighborhood about each cell, as described in Section 3.

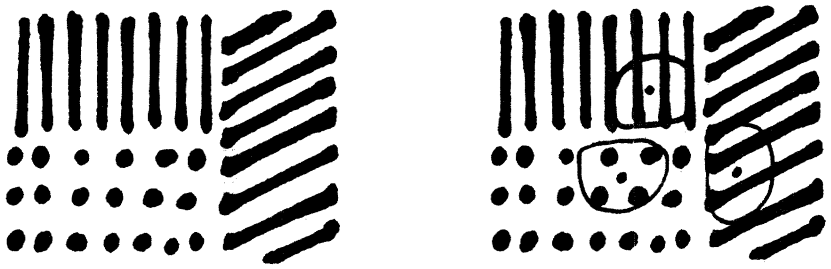


Figure 21. Left: A situation containing several types of texture. Right: Support regions about several points, restricted so as not to cross texture boundaries.

This section has described several important points about support regions for functions. I illustrated these effects with 2D patterns, because they are easy to draw. However, the same effects occur in spaces of other dimensions and in a wide variety of application domains. Wide support neighborhoods are required for describing patterns of events over time and textures in camera images, analyzing 3D patterns of material structure, route planning, and matching images.

8. When are adjacency structures sufficient?

The adjacency structures and boundary markings used in previous sections can only represent a limited class of regular cell complexes. Although a more general representation is available, it represents cell complexes in a less useful form. Furthermore, this restricted class of complexes seems to include all those required by practical reasoning algorithms. Chapter 11 gives the details of these restrictions and the proof that they are sufficient. In this section, I summarize these results.

There are two ways to view space-filling cells. In previous sections, I have described them as composed of cells, all of the same dimension, touching in various patterns. This description is close in form to those used in most computer algorithms and in mathematical work on tilings. Alternatively, the common faces, edges, and vertices can also be seen as cells, but of lower dimension. This is the picture typically presented in topological descriptions of regular cell complexes. Figure 22 shows some cells of different dimensions in this second description of cell complexes.

The topology of a regular cell complex can be completely specified by a list of cells in it and a *face* relation among the cells. The face relation specifies when a lower-dimensional cell A is a face of a higher-dimensional cell B, i.e. when A forms part of the boundary of B. For example, in Figure 22, B is a face of A and C is a face of both A and B. By convention, every cell is also considered a face of itself. I refer to this representation as the *incidence structure* of the cell

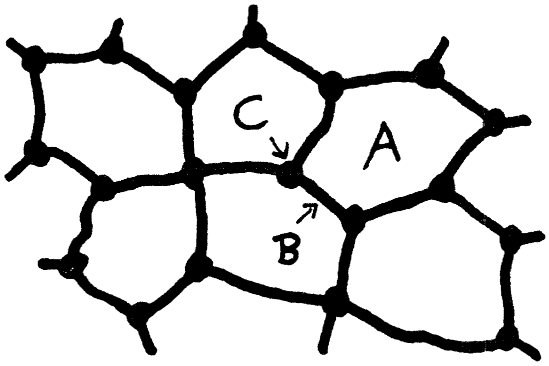


Figure 22. Faces, edges, and vertices can be viewed as cells of lower dimension. *A* is a 2-dimensional cell, *B* is a 1-dimensional cell, and *C* is a 0-dimensional cell.

complex. The proof that this representation fully specifies the topology of the cell complex is given in Chapter 11, Section 2.

The incidence structure representation is somewhat more general than the adjacency structure representation used in the previous sections. The two representations are interchangeable when each adjacency set corresponds to exactly one cell in the incidence structure representation. As detailed in Chapter 11, Section 3, this is true if the cell complexes meet three conditions, all of which seem reasonable for practical reasoning applications.

The first condition required for adjacency set representations is that there must be some fixed dimension N, such that each cell in the complex is a face of some N-cell. That is, each cell must either be an N-cell itself or it must be a lower-dimensional face of an N-cell. This forces space to have a consistent dimension, without any sections of different dimensionality. It also prevents space from having an infinite range of cell dimensions. Neither one of these situations would be desirable in practical reasoning.

The second condition on the form of cell complexes is that every (N-1)-cell must be a face of at least two N-cells. Intuitively, this means that space has no edges. The representation can still be used for finite cell complexes, which may have edges, so long as they are part of a larger complex without edges. So, for example, Figure 23 shows a cellular representation for a bounded 2D region. The cells being represented are shaded. The unshaded cells are *border cells*, which share edges and vertices with the shaded cells. Since the adjacency sets corresponding to these edges and vertices contain border cells, these cells must be mentioned in an adjacency structure description of the region, although they themselves are not part of the region being described.

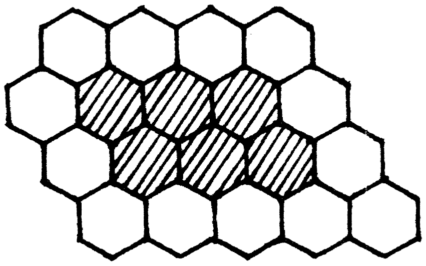


Figure 23. In order to specify the topology of a cellular region using adjacency sets, border cells must be added around the boundaries of the region.

The final condition that a cell complex must satisfy in order for adjacency structures and incidence structures to be equivalent is that the intersection of any set of cells must be exactly one cell or empty. This prohibits two cells from touching along two disconnected faces or two faces of different dimension. It also prevents gratuitous sub-division of the common face of several N-cells. The forbidden possibilities are illustrated in Figure 24. Note that the first condition is not a restriction on the form of regions, but only on the form of the digitization used to represent them. Regions that touch along multiple faces can be represented by breaking them up into several cells, as shown in Figure 25.

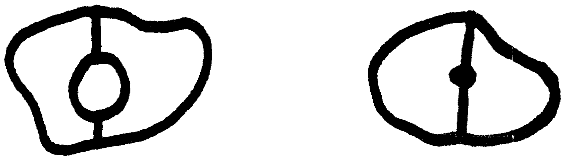


Figure 24. In an N-space structure, the intersection of any set of cells must be exactly one cell or empty. Thus, two cells cannot touch along two disconnected faces, as shown on the left. Nor can the common face of two cells be split into several cells, as shown on the right.

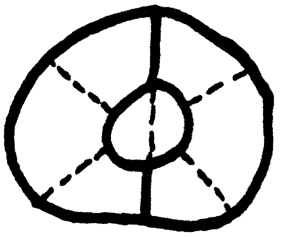


Figure 25. Two regions that touch along multiple faces can be modelled by using several cells to represent each region.

Adjacency structures are more convenient for practical reasoning than incidence structures. The analyses used in previous sections make a sharp distinction between cells of maximal dimension and cells of lower dimension. Digitized functions, for example, are maps between cells of maximal dimension. Lower-dimensional cells are only used as locations in which to place boundaries. Adjacency structures make this distinction explicit, whereas incidence structures treat all cells alike. Because of this, adjacency structures more closely match the data structures used in practical reasoning algorithms.

Adjacency structures also restrict the form of representations in ways that eliminate pathological cases that would case difficulties in practical reasoning. For example, spaces represented using adjacency structures must have a consistent dimension and cannot include stray pieces of lower dimension. Adjacency structures can only describe situations that end abruptly as if they were pieces of some larger situation with no edges. This fits nicely with the intuitive belief that the universe does not have edges.

9. Restrictions on the form of representations

Whether represented by incidence structures or adjacency structures, regular cell complexes impose restrictions on the form of space, boundaries in space, and the cells used in digitizing space. Boundaries induced by label contrasts, such as those used in this thesis, also have restrictions on their form. However, these restrictions primarily eliminate pathological cases that are not desirable in practical reasoning. Furthermore, we see that cellular topology allows more flexibility than previous representations.

The most basic restriction imposed by the cellular representations is that space must look locally like $\mathbb{R}^n$. This is because each cell used in building regular cell complexes is an n-ball of some dimension and the conditions for using adjacency structures require that the maximum dimension of space be consistent. This prevents a number of unpleasant pathologies found in topology textbooks, such as the long line. It also forbids space or time from looking like the rational numbers or the hyperreals. Although these two possibilities have been proposed for practical reasoning (see van Benthem 1983, Weld 1988), their topological structure has many undesirable properties. For example, intervals in either of these spaces are not connected.

The cells used to represent space are not restricted in shape, arrangement, or dimensionality. Previous formalisms have been confined to regular cell arrangements (e.g. Pavlidis 1977) or low dimensions. Many representations handle only rectangular arrays. It is not materially easier to define the topological structure of these restricted classes of cell complexes and non-regular cell arrangements are occasionally useful. For example, biological systems, such as the human retina, do not have perfectly regular cell arrangements. Non-regular tessellations are useful in creating compact variable-resolution representations for situations (Brooks Lozano-Pérez 1985, Rom and Peleg 1988, Funt 1980). Also, we see in Chapter 4 that it is convenient to be able to use non-regular cell shapes for proving algorithms correct, even when these algorithms only manipulate regular cell arrangements.

A cellular representation also cannot use more than finitely many cells to represent a bounded region of N-space. It is possible to create cellular representations in which infinitely many cells touch at a point or along a face, but these representations cannot have the topology of $\mathbb{R}^N$ or an N-manifold. Because boundaries are placed on or between cells, this restriction also makes it impossible to represent infinitely dense sets of boundaries directly. Cellular representations can branch, as shown in Figure 26. In later chapters we will see a few applications in which researchers have proposed such models for time. However, the branches must occur at cell boundaries and thus infinitely dense branches cannot be directly represented.

A second, and closely-related, limitation of cellular representations is that digitized functions cannot distinguish functions that approach a limiting value asymptotically, without ever reaching it, from functions that actually reach the

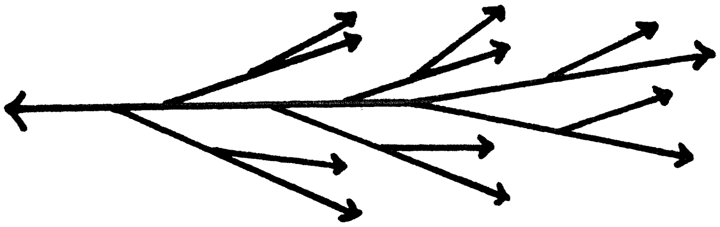


Figure 26. A branching time line.

limiting value. When the differences from the limit value become small enough, the values must be represented using the same cell in the value space as the limit value. Since it cannot distinguish the two cases, an algorithm using digitized data must treat the asymptotic function as though it actually reached the limit point.

We can cast this observation into a second form which is more directly relevant to practical applications:

> If a property is changing in value with a slope of constant sign and it is moving towards a limiting value, the property either becomes indistinguishable from the limiting value after some finite amount of time or else the slope becomes indistinguishable from zero after some finite amount of time.

Suppose, for example, that you are shovelling snow out of a driveway.[8] After some finite period of time, it must either be the case that you have removed all but negligibly much of the snow, or else your rate of shovelling has become negligible. This generalization will prove useful in explaining data from both linguistic semantics and high-level reasoning.

In the applications discussed in this thesis, boundaries are always induced

[8] Of finite extent!

by contrasts in cell labelling. Because of this, they always satisfy the *subset condition.* This condition states that an adjacency set must be in the boundaries if any subset of it is in the boundaries. For example, if the edge between two cells belongs to the boundaries, its endpoints are also part of the boundaries. Similarly, if an entire cell belongs to the boundaries, so do all of the edges and vertices that it touches.

Aside from the subset condition, boundaries can be any collection of adjacency sets. Boundaries can intersect one another and a boundary can end abruptly in the middle of a region. Figure 27 shows examples of real situations in which boundaries end abruptly. In the applications presented in this thesis, it is typically best to place boundaries between cells. However, it is occasionally helpful to place boundaries on cells and even to create boundaries more than one cell wide. The formalism allows all of these options.

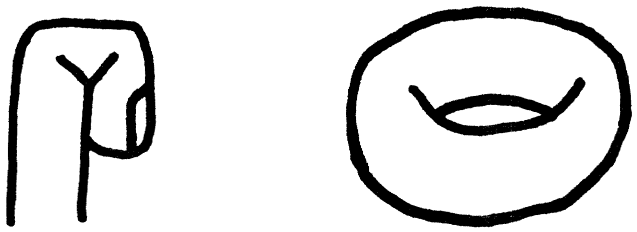


Figure 27. Boundaries can end abruptly in the middle of regions. Left: a bent finger. Right: a torus seen in 2D projection.

Thus cellular topology imposes a number of restrictions on the form of representations for situations. However, these restrictions seem to eliminate only pathological situations that are of little use in practical reasoning. Later chapters discuss how some of these restrictions apply to various application domains and confirm that they are not prohibiting useful types of representations.

10. Chapter summary

In this chapter, I have presented the basic machinery needed to represent space for visual analysis and other practical reasoning tasks. This representation of space, called cellular topology, is based on regular cell complexes. The topological structure of these cell complexes can be fully specified using simple, combinatorial representations. We have also seen how boundaries can be added to these representations and how this changes the topological structure of space. Although these representations impose some conditions on the form of space and how situations can be represented, the forbidden possibilities involve pathological cases that are not useful in practical applications.

In practical applications, functions cannot be represented in full detail, but must be approximated using only finitely many values with only finite precision. The cellular representation provides a good framework for analyzing these digitized functions. We have also seen how the relationship between digitized functions and underlying continuous functions can be important in producing robust reasoning algorithms.

Topological changes due to boundaries affect practical reasoning algorithms in a number of ways. The regions to either side of a boundary are not connected. Continuous functions can have abrupt changes in value across boundaries. Continuous matches between two situations must match boundary locations. Finally, the presence of boundaries can affect the shape of support regions used in computing function values.

As I said at the beginning, the presentation in this chapter is informal. Formal details and proofs missing from this discussion are to be found in Chapter 11. Some readers may wish to look at Chapter 11 before continuing. Chapter 11 also compares my representation of cell structures to previous proposed methods of specifying the topology of a digitized space.

Chapter 3: Domain Examples

1. Introduction

In Chapter 2, we saw a number of ways in which topological structure, induced by the presence of boundaries, could affect reasoning algorithms. In this chapter, I introduce the application domains considered in this thesis and briefly describe how topological phenomena appear in each domain. In Chapters 4-8, I consider each of these domains in more detail.

In this thesis, I consider examples from three domains: low-level vision, natural language semantics, and high-level vision and practical reasoning. I have grouped high-level vision and practical reasoning together because they are closely related and consider similar examples. Because the implementation for this thesis is in low-level vision, the discussion of this area is more extensive. Three algorithms have been implemented: an edge finder, an image matching algorithm, and a stereo analysis program using the image matcher.

These domains illustrate a number of ways in which topological structure can affect reasoning algorithms. We see that algorithms may require connectivity of regions, including function support regions, and may require correspondences used in matching two representations to be continuous. These topological constraints are often combined with other types of constraints, yielding mixed topological and metric properties such as star-convexity. We see evidence that lack of material connectivity and sharp changes in functions tend to cluster at a restricted set of locations, indicating the presence of boundaries.

In addition to the topological phenomena, we also see a number of other examples important to the thesis. We also see how digitized representations are used in several domains and how the digitization occasionally affects algorithms and representations. We see examples of functions that require wide support regions. Finally, we see a number of places where previous researchers have run into technical problems modelling boundaries.

2. The edge finder

In Chapter 2, we saw that abrupt changes in function values indicate the presence of boundaries in space. The goal of edge finding is to detect locations of sharp change in real input data, typically arrays of light intensities delivered by a video camera and digitizer. The difficult problem in designing edge finding algorithms is to make them detect the wide variety of boundaries present in natural images without being sensitive to camera noise. The algorithm implemented for this thesis takes advantage of the connectivity of edge finder response regions to separate real features from noise.

The Phantom edge finder finds boundaries in an image by locating regions of the image in which directional second differences are significantly different from zero. Regions of significant response are then labelled as darker or lighter than neighboring regions, depending on the sign of the response.[1] Boundaries are placed where dark and light regions meet. Figure 1 shows these response signs and boundary locations. The boundaries and dark/light labelling form the input to later visual processing, such as stereo analysis, motion analysis, texture description, and shape description.

[1] These response signs are produced by combining responses from directional differences in several directions. See Chapter 4 for details.

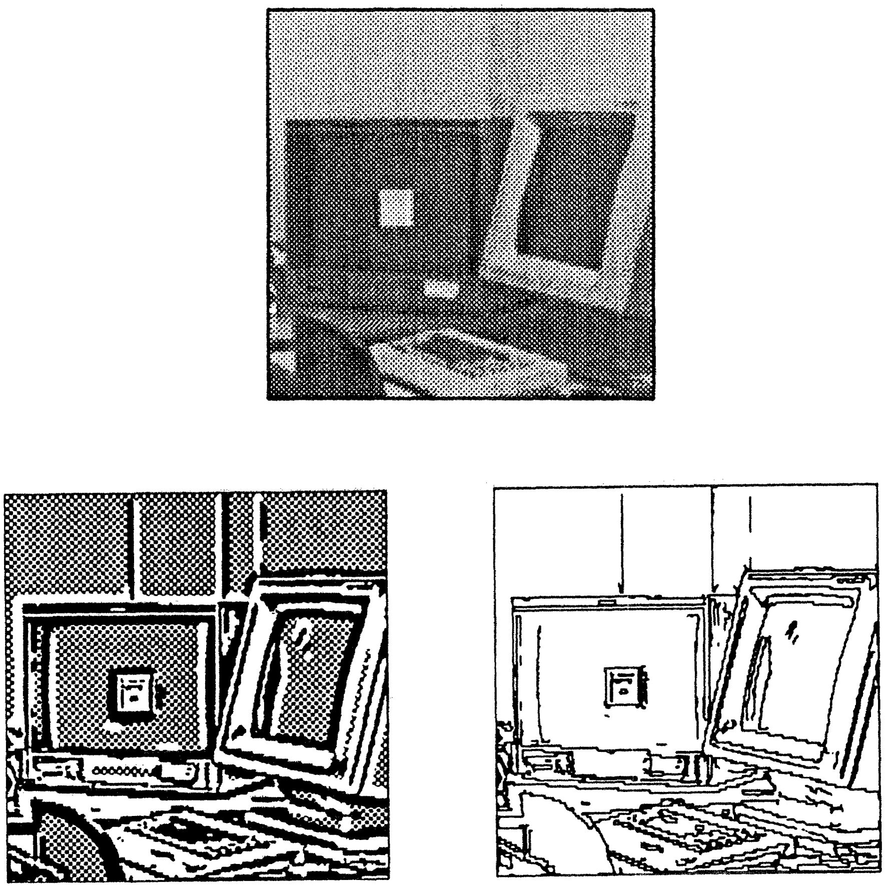


Figure 1. Top: a digitized image (300 by 300 cells). Bottom left: second difference response sign. Positive and negative responses are shown in black and white. Responses not significantly different from zero are displayed using a checkerboard pattern. Bottom right: boundaries induced by transitions between dark and light regions. Cells on or to the dark side of boundaries are shown in black.

The main challenge in doing this type of edge finding is suppressing effects of camera noise. These images are taken with a video camera attached to a digitizer that converts the camera output into arrays of integers. This system blurs the image slightly and introduces low-amplitude random noise. Figure 2 shows the edge finder's dark/light labels for the same image with no noise suppression. As you can see, the camera noise generates spurious dark and light markings, particularly in regions of uniform intensity. The noise suppression algorithm

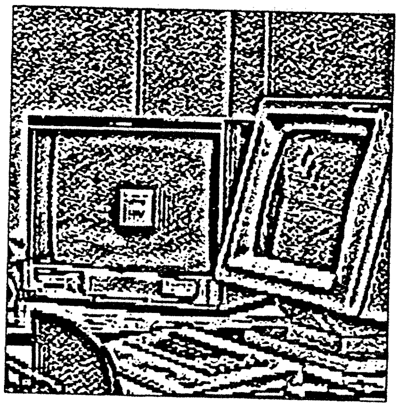


Figure 2. Second difference response sign for the image in Figure 1, but with no noise suppression.

removes these spurious responses, producing the clean output shown in Figure 1.

In the past, responses due to noise have often been identified on the basis of their amplitude. This is not a robust method for distinguishing noise responses, because some responses to real features have low amplitudes. Notice, however, that one can roughly identify the noise responses using only the response sign information shown in Figure 2. Noise generates responses with only small connected regions of the same sign, whereas real responses typically generate wide response regions. Thus, both response amplitude and response region shape provide useful information about which responses are due to camera noise.

The Phantom edge finder combines both shape and amplitude information into one operation that sums response amplitudes over a neighborhood of each cell. The neighborhood about each cell x is the maximal star-convex neighborhood, defined in Chapter 2, within which the second difference response does not change sign. If this sum is too low, the response at x is classified as due to noise. Because this operation does not cross the boundaries defined by sign changes, the evaluation of each response region is not corrupted by the presence of nearby response regions. This test is able to robustly distinguish real responses from those due to camera noise.

The edge finder is interesting for several reasons. The main noise suppression algorithm shows a simple, but important, use for connectivity in low-level visual processing. In Chapter 4, we see that connectivity can also be used in distinguishing step edge and roof edge responses. The edge finder provides an example of a digitized function whose range has an unusual structure, as well as many examples of strange boundary shapes. Finally, the edge finder shows how we can extract a clean topological structure for an image out of real intensity data, despite camera noise and scene irregularities.

3. Image matching

The second major algorithm implemented for this thesis matches two images in ways that preserve their topological structure. This matcher can be used in a number of different application domains. I present the matcher first in the context of testing edge finder output for stability under noise (in Chapter 5), because this application uses the matcher in a straightforward way. I then show how this matcher can be used in stereo analysis (in Chapter 6). This application is more interesting, but it requires a non-trivial control structure in addition to the basic matcher. In this section, I give an overview of the edge finder testing domain and the matcher algorithm. Stereo analysis is summarized in the next section.

Chapter 8 presents a number of tests of the performance of the new edge finder. Among these is a test for stability under noise introduced by the camera and digitizer system. The basic idea behind this test is simple. Two pictures of the same scene are taken with the same camera position, but a few minutes apart. Thus, the two pictures represent the same image, but corrupted with different samplings of random noise. The edge finder is run on both images and the results compared. Any differences between the two results reflect instability under noise. Most previous experiments have compared output on one image to some "correct" output (see Chapter 9 for further discussion), but this does not change the character of the comparison problem.

The difficulty in doing such a test is how to compare the two edge finder outputs in a meaningful way. Noise causes two types of changes to the edge finder output: changes in boundary topology and changes in boundary location. In later chapters, we see that many high-level programs, from stereo to object recognition, make use of image topology. The two types of changes in edge finder output affect these programs differently, and thus they should be reported separately. Previous studies of edge finder performance (Haralick 1982, Nalwa and Binford 1986, Sher 1987a, Pratt 1978, Fram and Deutsch 1975) attempted to separate these two effects, but their heuristic methods seem only applicable to images with sparse boundaries and/or small amounts of boundary motion. Using the new model of image topology developed in previous chapters, we can produce a more general and principled algorithm for matching two edge finder outputs.

The image matcher separates the matching problem into three phases: adjustment, computation of match strength, and analysis of boundary motion. In the first phase, the algorithm adjusts one image so as to make it as similar as possible to the other, without changing its topology. A successful match between the two images requires not only that the boundary locations match, but also that the edge finder's dark/light labels match. Chapter 5 describes the set of operations used to adjust boundaries and labels and proves them correct, using techniques developed in Chapter 11.

Requiring that dark/light labels match simplifies the adjustment process. Consider the situation shown in Figure 3. Without label information, the adjustment process would have to explore two candidate matches for each boundary, one to either side of it. If labels are required to match in the two images, however, the boundary must be adjusted so as to reduce the region of label conflict. In fact, the adjustment process can be thought of as a method of getting as many cells as possible to have matching labels. The raw match map is produced by comparing labels in the original and adjusted images. Figure 4 shows a match between two images before and after adjustment.

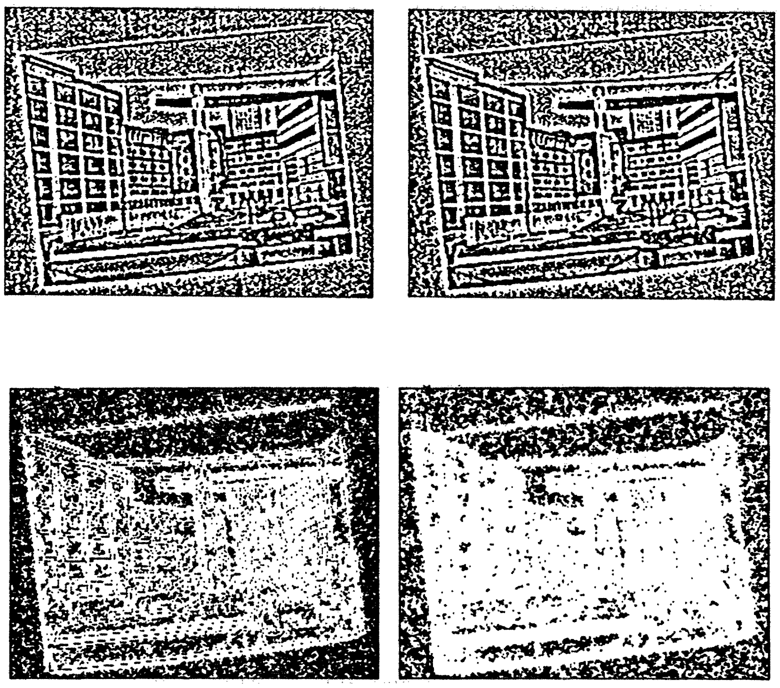


Figure 4. Top: two noisy edge finder responses. Bottom: the match between them, before (left) and after (right) adjustment. Matching cells are shown in white and non-matching cells shown in black.

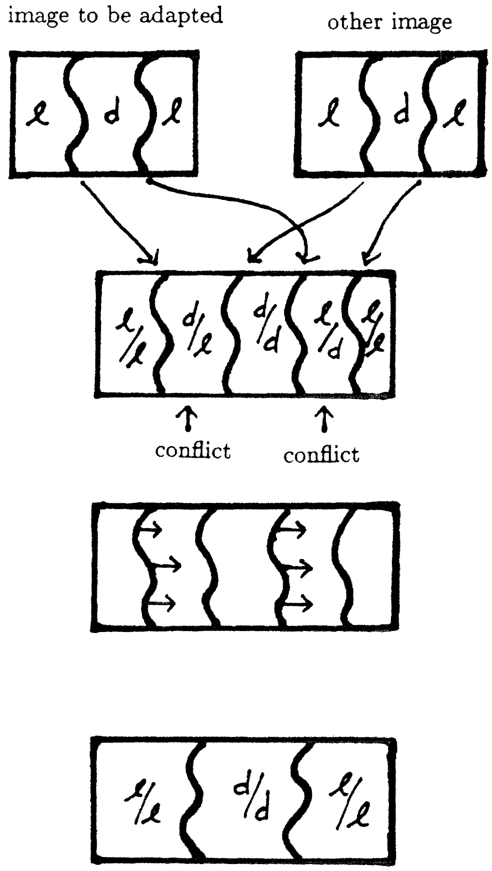


Figure 3. Boundaries are adjusted so as to reduce the regions of conflicting labels. Top to bottom: the two images, identifying label conflicts, moving boundaries, and final (identical) boundaries and labels.

Adjustment eliminates effects of boundary motion and thus the match map after adjustment reflects only topological mismatches between the two images. Notice, however, that even non-matching regions contain many matching cells at this point. Good and bad matches are distinguished by how much the non-matching cells break up the image. In a region of good match, extended connected regions are marked as matching. In a region of poor match, only very small connected match regions occur. Therefore, the area of a connected (star-convex) neighborhood about each cell is used as a measure of the goodness of the match about that cell. A clean match map can then be produced by re-classifying cells with low strengths as non-matching. Figure 5 shows the clean match map for the image match in Figure 4. As you can see from the example, regions where the edge finder response reflects camera noise are now classified as non-matching, whereas regions where the response reflects primarily the scene are classified as matching.

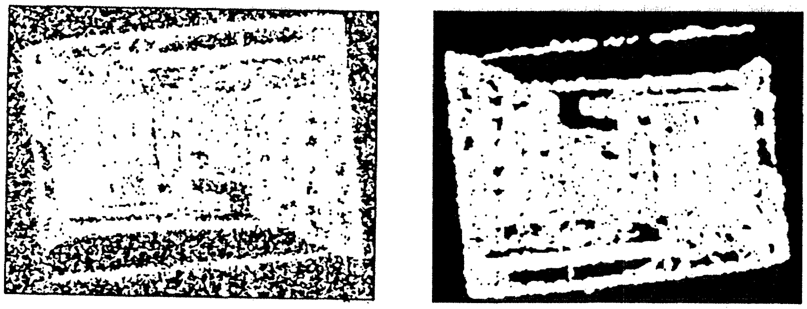


Figure 5. The match map from Figure 4 before (left) and after (right) low-strength responses have been removed.

After adjustment, the adjusted and non-adjusted versions of the image are compared, to identify which cells have had their labels changed. Because of way boundary adjustment is done, there is a characteristic pattern to the locations of these adjusted cells. As shown in Figure 6, a connected region of adjusted cells lies directly to one side of each boundary that was moved during adjustment. The width of this band of cells reflects the amount that the boundary has been moved. The final stages of matching analyze these adjustment regions to extract information about boundary motion.

For edge finder testing, we do not expect any net movement of boundaries in any one direction, over an extended section of the image. Rather, boundary

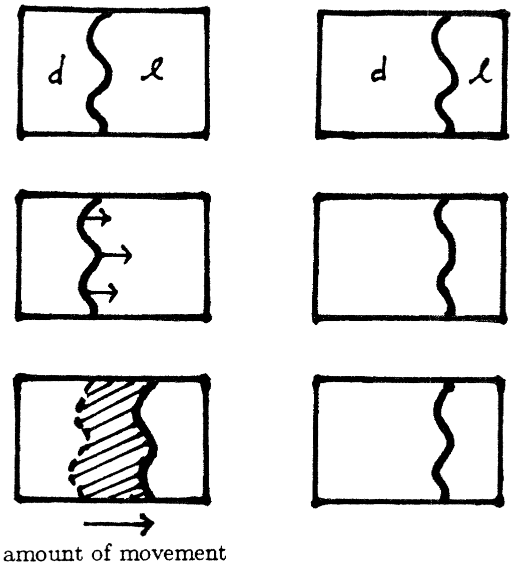


Figure 6. A boundary that was moved during adjustment has a connected region of adjusted cells to one side of it. Top to bottom: the two images, moving boundary in one image, the region of adjusted cells.

locations typically move back and forth, as noise varies. The total amount of fluctuation in boundary locations can be measured by comparing the number of adjusted cells to the total number of edge cells (cells next to boundaries). For other applications, such as stereo analysis, this fluctuation should be suppressed and any net motion in some direction extracted. This involves extracting horizontal and vertical components of the boundary motion at each edge cell, and then smoothing these measurements to suppress fluctuations due to noise.

From the standpoint of this thesis, there are two interesting aspects to the image matching algorithm. First, boundary adjustment is required to preserve image topology. This is a direct test of the hypothesis that topological structure is useful in building practical algorithms. Furthermore, the process of proving adjustment operations correct illustrates the usefulness of the mathematical machinery developed in Chapters 2 and 11.

The second interesting feature is that both the matching strength and the boundary motion computations are restricted to connected regions. Strengths are measured using the area of a star-convex neighborhood about each cell. The horizontal and vertical components of boundary motion are calculated by measuring the length of connected straight paths through the adjustment region. Finally, these two components are smoothed by averaging values over the star-convex neighborhood of each cell. This restriction to connected regions allows these computations to use wide support regions while not crossing boundaries between matching and non-matching regions.

4. Stereo matching

The image matching algorithm presented in the previous section handles only one alignment of the two images. In order to do tasks such as stereo matching, a control structure must be built that can search a variety of alignments for possible matches. This section sketches the implemented stereo control structure, described fully in Chapter 6. Because stereo matching involves a change in viewpoint, in addition to the effects of camera noise, it provides a stiffer test of the matcher's capabilities than edge finder testing.

The input to a stereo matching algorithm consists of two images of the same scene, taken at the same time from slightly different viewpoints, as shown in Figure 7.[2] In human vision, the images would come from the two eyes. In computer vision, they come from two cameras positioned in a manner roughly similar to human eyes. In both cases, the viewpoints are sufficiently similar that most 3D points that are visible to one eye are also visible to the other. A stereo matching algorithm must reconstruct a correspondence relating points in the two

[2] Appendix A explains how to view such a pair of images so as to see apparent depth.

images that are projections of the same 3D point. From this correspondence and the relative positions of the cameras, the 3D locations of surfaces in the scene can be computed.

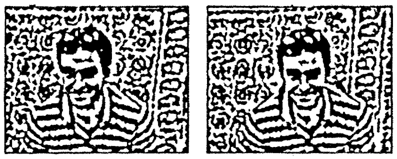


Figure 7. A stereo pair contains two images of the same scene, taken from slightly different viewpoints. This figure shows the edge finder output for such a pair of images.

Stereo correspondences are typically presented in the form of *disparity* values for each pixel in the images. This representation assumes that some reference alignment of the images has been selected (e.g. matching cells with the same coordinates in two images). The disparity at a pixel is then a vector representing the difference between the corresponding location in the other image as given by the alignment and the true corresponding location as provided by the stereo matcher. This is illustrated in Figure 8.

Fully accurate models of stereo geometry and optical distortions for a camera system are quite complex. Camera modelling is tangential to the main point of this thesis. Therefore, I use a simplified model of the viewing geometry. For the images I use, the errors caused by deviations from this model are small enough not to cause problems in matching.[3]

Figure 9 shows the positions of two cameras in a standard stereo arrangement. The cameras lie in approximately the same horizontal plane, so we can consider

[3] This algorithm is more tolerant of errors than previous stereo matchers.

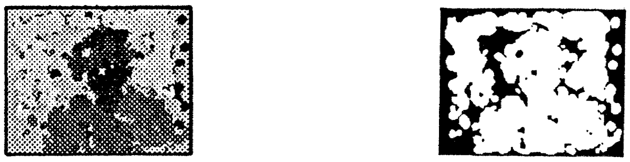


Figure 8. Left: stereo disparities for the images in Figure 7. Darker regions in this figure have larger disparities and correspond to 3D surfaces that are closer to the cameras. Right: match map showing (in white) which regions of the stereo images were successfully matched.

them in 2D projection, from above. The cameras are pointed at a common 3D location, probably representing some object of interest in the scene. The *vergence*, i.e. the difference between the directions in which the two cameras point,[4] changes as a human or a (hypothetical) computer system looks around the world, so as to keep both cameras pointed at whatever is currently of interest.

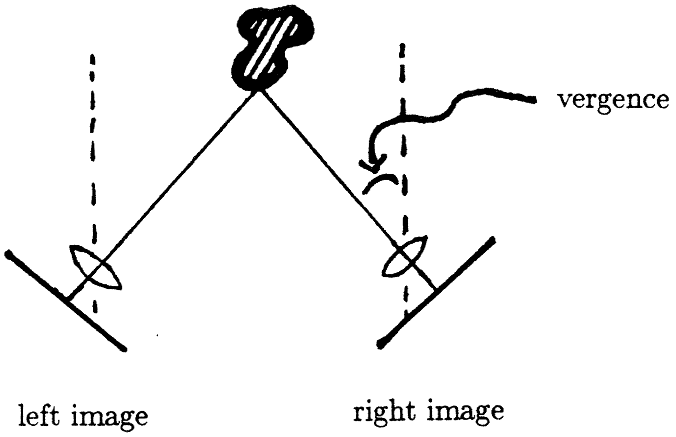


Figure 9. A stereo arrangement.

[4] Details of how this is defined are not relevant to the following discussion.

In an actual camera system, the vergence is small relative to the distance between the cameras and the object. Thus, I suppose the image planes of the two cameras lie in a common plane and treat the deviations from this as an unmodelled source of error. Disparities are then 2D vectors in this plane. The *vertical disparity*, i.e. the component of disparity perpendicular to the line joining the two image centers, is ideally zero for all points in the images. The *horizontal disparity*, i.e. the component parallel to that line, is (roughly) inversely proportional to the *depth*, i.e. the distance between the 3D object and a line passing through the two cameras. For convenience, researchers often arrange for the scan lines of the images to be parallel to the line connecting the image centers, so that horizontal disparities are then parallel to the scanning direction.

There are three sources of error in this model: mis-alignment of the two cameras, simplifications made in the model, and distortions introduced by the camera system (particularly when wide-angle lenses are used). The model simplification seems to create only small errors. Camera distortions change only slowly over time[5] and can be handled by normalizing the images prior to stereo analysis, using a pre-computed calibration. Camera mis-alignment causes larger inaccuracies, which change as cameras are moved to new vergences or so as to point in new viewing directions. The stereo system must be able to supply the information required to adjust camera positions and update camera calibrations.

Because vertical disparities should be zero in an ideal camera system, the stereo algorithm uses vertical disparities to estimate the required adjustments to camera position. From computed vertical disparities, the algorithm estimates two camera adjustment parameters: vertical translation and rotation about the

[5] In a camera system, they might not change measurably. In a human, they would.

center of the image.[6] These parameters are used to adjust the positions of the images so as to achieve a better match. Determining errors in camera calibration would require observation of disparities over extended periods of time, in order to detect any trends that persist systematically across diverse scenes, and has not been implemented.

The stereo matcher implemented for this thesis uses a coarse-to-fine control strategy. The edge finder supplies boundary locations and dark/light labels at a variety of scales. Matching results for coarser scales are computed first and they are used to adjust camera positions (simulated in software) and plan the set of alignments to be searched at the next finer scale. This implementation differs from previous implementations in having a wide search area at each scale, relative to coarse-scale positions, and in considering the possibility of vertical displacements in addition to horizontal ones. Although the larger search areas require more computation time, they are required in order to match human capabilities.

Stereo matching at each scale involves a search over a range of alignments of the two images. At each alignment, the image matcher described in the previous section is used to determine which parts of the image match and how well. It also supplies estimates of the disparity of individual patches of the image, relative to the alignment. When matching has been done for all alignments in the search area, the best candidate match is chosen for each image location. The decision among alternative matches is based on their matching strengths, as well as how close they are to coarser-scale results. The coarser scale context is required in order to handle regions with translational symmetries at the finer scale, such as striped regions and regions of uniform color. Finally, a modified version of the

[6] More sophisticated models of camera misalignment could be used. Again, this is tangential to this thesis.

edge finder's noise suppression algorithm is used to remove outliers and fill small holes in the output disparities.

A slight modification of the stereo control structure could be used for the analysis of motion sequences. Motion sequences involve a wider space of possibilities than stereo analysis, because vertical disparities are not as tightly constrained. A full discussion of control strategies for motion analysis would involve issues of what objects the reasoner was interested in, because it may only be possible to track the motion of certain parts of the visual field at fine scales. Nevertheless, the same techniques developed for stereo analysis should be applicable and Chapter 10 presents a brief example showing how they might be used.

There are two ways in which the topological matcher is important in building the stereo algorithm. First, the larger search areas at each scale place more demands on the robustness of the matching algorithm. Previous stereo algorithms have used constraints similar to the requirement that dark/light labels match. They have also used "disparity gradient" or "local constancy" conditions, similar to those imposed by the search through alignments in my algorithm. However, the new stereo algorithm also requires that the correspondence preserve topological structure. This type of constraint has previously been used only rarely (e.g. Grimson 1985, Mayhew and Frisby 1980, 1981, Chen 1985) and implemented in weaker forms. Without the additional constraint provided by the continuity requirement, previous algorithms find it difficult to disambiguate large numbers of candidate matches.

Secondly, computation of strength and disparities at each alignment is confined to connected regions of matching cells. This prevents most support regions from crossing sharp changes in depth or overlapping occluded regions, without restricting the size or shape of support regions. Because of this, support regions can be made as large as is necessary to achieve good accuracy in computed disparities and to consider enough context for good assessments of match strength. Furthermore, cells near the edge of a region can gather support from large support neighborhoods, despite the fact that these neighborhoods cannot be centered about them. Previous algorithms have been forced to trade the benefits of wide support neighborhoods off against the problems of smearing and contamination across sharp changes in depth.

Thus, the stereo matching algorithm shows how the topological ideas developed in this thesis can be used in solving practical problems. Furthermore, the images being matched can be quite complicated, with large amounts of fine texture. Most discussions of topological properties, both in mathematics and computer vision, consider only examples with simple structure. One is tempted to think of topology in terms of Euler-number classifications of surfaces or to reduce it to connectedness for more complicated problems. Stereo matching and edge finder testing illustrate how one can use the full topological structure of even very complicated images.

5. Linguistic Semantics

The next group of data that I consider in this thesis comes from linguistic semantics. The goal in this field is to formulate rules for describing the meaning of sentences. Since a full description of sentence meaning would require solutions to much of Artificial Intelligence, researchers in linguistic semantics are particularly concerned with classifying those aspects of meaning that are important in determining whether a string of English words is an acceptable English sentence. Not only does this data suggest interesting uses for topological properties in semantics, but cellular models of time avoid technical problems encountered by previous researchers.

The data that I describe in most detail involves models for the temporal structure of different types of situations,[7] verb tense and aspect, and temporal adverbs and connectives. I represent time using a cellular model, shown in Figure 10. Situations in time will be modelled by associating descriptions of properties with cells in time and descriptions of processes with (connected) intervals of time.

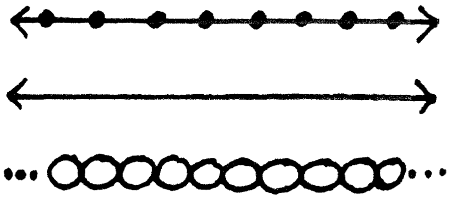


Figure 10. A cellular model of time consists of an ordered set of 1-cells, each one joined to the next at a common endpoint, as shown in the upper drawing. The underlying space is just like the real number line (center). The lower drawing shows an alternative graphic representation for this set of cells.

The situations described by natural language verb phrases seem to fall into a limited number of classes: *states*, *activities*, *state changes*, and *accomplishments*. For example, Sentence 1 describes a state, Sentence 2 describes an activity, Sentence 3 describes a state change, and Sentence 4 describes an accomplishment.

(1) Sussman was in the machine room.

(2) The aide shredded incriminating documents.

(3) Bonnie passed her area exam.

(4) Eric made a fresh pot of coffee.

[7] I use the term "situation" to cover both actions, such as "running," and states, such as "being green." I am using this term in an informal sense and do not intend it to imply any particular theory of how actions and states are represented.

States are descriptions of the world at a moment in time, activities describe ongoing patterns of change, state changes describe abrupt changes in the world, and accomplishments describe an activity brought to an edge by an abrupt change. I refer to activities, state changes, and accomplishments as *actions.* Figure 11 shows the models for the temporal structure of these four classes.

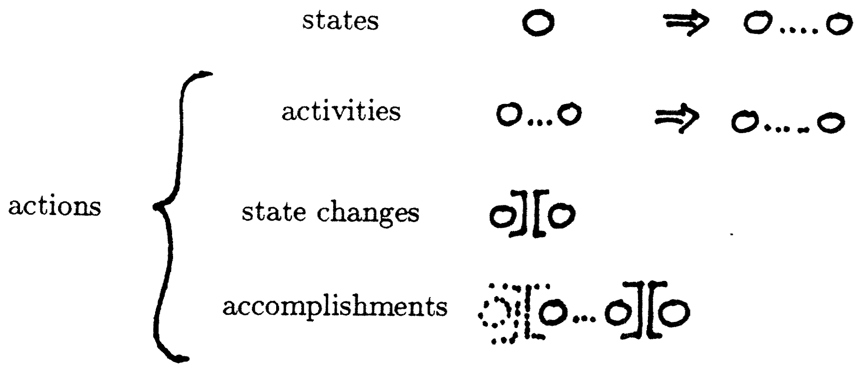


Figure 11. The topological patterns of different types of situations.

The four classes of situations can be distinguished linguistically, because certain constructions place restrictions on the class of the verb phrase (or other constituent) used in them. For example, state descriptions can be verified from a description of the world at only one moment of time,[8] whereas verifying that an action has occured requires examining the world at two or more moments of time. Since the present tense in English refers to only a moment of time, this means that only states can appear in the present tense. Thus, Sentence 5 is acceptable, whereas Sentence 6 is not acceptable unless re-interpreted as a (state-like) description of the aide's habits.

(5) Sussman is in the machine room.

[8] Represented as a cell in these models.

(6) #The aide shreds incriminating documents.[9]

Conversely, only actions can appear in the *progressive* aspect, as illustrated by Sentences 7-8:

(7) #Sussman is being in the machine room.

(8) The aide is shredding incriminating documents.

This can be explained by observing that progressives of actions behave as if they were states. In the model for progressives described in Chapter 7, the progressive of a state would mean the same thing as the original state and it would thus be redundant.

The three types of actions can be distinguished by similar types of tests. For example, only activities can occur with prepositional phrases using "for" to measure an amount of time, as illustrated by Sentences 9-11:

(9) The aide shredded incriminating documents for several minutes.

(10) #Bonnie passed her area exam for several minutes.

(11) #Eric made a fresh pot of coffee for several minutes.[10]

State changes are distinguished from the other two classes because they cannot occur in constructions of the form "stop X-ing," as illustrated by Sentences 12-14:

(12) The aide stopped shredding incriminating documents.

(13) #Bonnie stopped passing her area exam.

[9] I use the hash mark (#) to indicate that a sentence is unacceptable and a question mark (?) to mark sentences of dubious quality. When sentences are so bad as to be ungrammatical, an asterisk would be more traditional. However, in the data I present, clear cases of ungrammaticality are rare. It is more typical that a sentence could be acceptable, but would have to describe a bizarre situation or be embedded in a bizarre context. I am using the hash mark to indicate this looser type of unacceptability.

[10] Dowty (1979) finds this type of sentence acceptable. I do not.

(14) Eric stopped making a fresh pot of coffee.

Chapter 7 discusses other tests which distinguish different classes of situations and how they can be explained in terms of the cellular models.

This pattern of topological classes and various types of internal structure for actions is roughly paralleled by English noun classes. Consider Sentences 15-18:

(15) I picked up a pencil.

(16) #I picked up some pencil.[11]

(17) #I picked up a sand.

(18) I picked up some sand.

Nouns can be divided into two classes: count nouns and mass nouns. Count nouns, such as "pencil," describe objects. Mass nouns, such as "sand," describe types of stuff. These two types of nouns can be given representations analogous to those for accomplishments and activities, respectively.

In Chapter 7, we see several ways in which the new model of space and boundaries can help in analyzing this linguistic data and in which the linguistic data provides evidence for the new model. There are two important points. First, cellular topology predicts a relationship between boundary locations and region connectivity. The linguistic data provides suggestive examples supporting this prediction. Secondly, cellular models avoid technical problems encountered by previous analyses (e.g. Allen 1984, Dowty 1979), due to a combination of the new model of boundaries and the use of digitized functions.

Connectivity, in the sense of cellular topology, seems to be useful in explaining the meaning of the *perfect* aspect in English. Consider Sentences 19-20:

(19) John has been in the kitchen for two hours.

[11] This can be acceptable, but only if "some" is stressed.

(20) Hal has fed the panther.

These sentences, in the perfect aspect, assert that a state was true or that an action occured over some interval in the past. In addition, they also assert that some state has persisted from the end of the state or action through to the present. The perfect leaves the details of the persisting state vague. So, depending on the context, Sentence 19 can either imply that the panther is no longer hungry or that Hal has experience feeding panthers. We can model this in cellular topology by requiring that no boundary relevant to the current context intervene between the state or action and the present moment, i.e. that the present moment is connected to the end of the state or action.

There is also suggestive evidence from certain English language constructions that boundaries due to different actions tend to coincide. For example, the connective "until" indicates that one state or actitivity occured over an interval ending at some specified boundary, as in Sentence 21:

(21) The panther stared hungrily at me until Hal fed him.

Forms with "until" do not actually assert that the first situation stops when the state change occurs, but they strongly imply it.

Chapter 7 presents these two examples in more detail, along with other examples involving the progressive aspect and the connective "when." These examples provide evidence that the behavior of different types of situations is consistent with the topological details of the models I have given them, particularly the boundaries used in representing accomplishments and state changes. While this evidence is fragile, it is a useful addition to evidence from other sources.

The new model of space and boundaries also avoids several technical problems encountered by previous researchers. First, cellular topology allows the distinction between states and actions to be expressed in terms consistent with real measurements, by distinguishing intervals containing only one cell from longer intervals. In previous models, this distinction was expressed as a distinction between points and intervals. This is difficult to connect with real measurements, because data is not available at individual points in time.

Secondly, cellular topology provides a problem-free model for state changes, as in Sentence 22:

(22) Bonnie passed her area exam.

This sentence expresses a change over time between two discrete property values. Previous researchers have encountered two problems modelling such sentences as functions from $\mathbb{R}$ to a property space. First, it is unclear which of the two values to assign to the point exactly at the transition. Secondly, state changes occur over a minimal-sized interval surrounding the change. In models based on $\mathbb{R}$, there may exist no such minimal interval. Cellular topology provides solutions to both these problems.

Finally, digitized functions can provide an explanation for why certain verb phrases become temporally bounded when they contain a spatial bounded direct object. Consider Sentence 23:

(23) John drank a glass of water.

This sentence describes an accomplishment, although the verb "drink" describes an activity. The direct object "a glass of water" is a count noun describing a bounded amount of water. Because it is bounded in space, the action of progressively consuming it must be bounded in time. This line of reasoning, proposed by Tenny (1987), works in cellular topology,[12] but it does not work if

[12] Chapter 2, Section 9 presented this briefly.

standard real functions are used.

The linguistic data provides an interesting extension of the domains in which cellular topology can be applied. Analyses used by authors in this domain can not only be re-written within cellular topology, but technical problems can be eliminated. There is suggestive evidence that connectivity and topological boundaries are useful in modelling this data. Furthermore, this data is closely related to the data considered in high-level reasoning, which I describe in the next section.

6. Reasoning and high-level vision

The final group of examples come from reasoning and high-level vision. In these areas, researchers try to emulate the human ability to identify and describe situations, predict what will happen to objects in them, and plan actions for changing a situation. This research is somewhat removed from any source of concrete data, either visual or linguistic. However, the phenomena that this work attempts to explain are more varied and more intuitively appealing. Ideally, it should provide the link between low-level vision, motor control, and low-level language processing. This section provides a summary of the relevant parts of this work and it is discussed in more detail in Chapter 8.

Reasoning examples of interest to this thesis can be divided into four types of problems: modelling physical objects, modelling changes over time, route planning, and recognizing objects. Suppose, for example, that we are training a robot to make coffee. We might first describe the shape of the coffee maker, the sink, the water container, and the coffee pot. The robot must be able to recognize all of these items visually in order to orient itself and start work. Route planning would be used to determine how the robot must move its arms in order to put the pot in the machine, fill the container with water, and pour the water into the machine. Models of changes over time would be used to describe how the water is heated and to predict that coffee will flow into the pot for a time and then eventually stop. There are many other things involved in making coffee, but I consider primarily these four aspects of the problem.

Models of physical objects are essential to reasoning about practical problems. and topological properties of these models are considered important by many researchers. For example, in the coffee making example, water can flow through the coffee maker precisely because the water input for the machine is connected to the coffee output by an open tube. If you pull on one end of an electrical wire, the other end will move because the entire wire is physically connected. Current can flow between the two ends of the wire because it is also electrically connected.

Other types of reasoning, such as route planning, require metric information in addition to topological information. Consider the bowl shown in Figure 12. The bowl is connected, so water and objects cannot pass through it. The interior of the bowl is connected to the outside of the bowl, so water and objects can move into and out of the bowl. However, since the paths out of an upright bowl all involve motion against the force of gravity, water will not move out of the bowl spontaneously. The first two deductions depend only on topological properties of the bowl. The third deduction requires metric information. However, because the metric information is augmenting a topological description of the bowl, rather than standing alone, it need not be very precise. The reasoning depends on the presence of a concavity, but not on the details of its shape.

Topological properties are also important in recognizing objects and situations. Most algorithms for analyzing the shape of objects use connectivity, in the form of routines that parse image boundaries into extended connected segments.

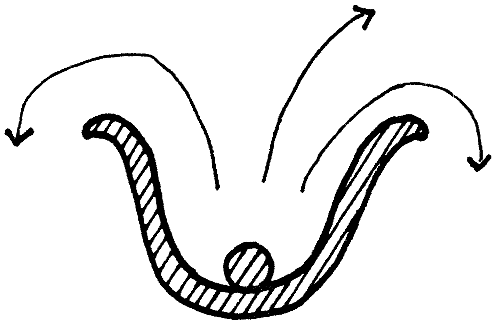


Figure 12. Things can move out of a bowl, because it is topologically open, but they do not do so spontaneously, because of gravity.

Objects are often described as assemblies of sub-regions, each of which must be connected and sometimes convex. Topological properties are important in recognition because 3D objects must be identified from their 2D projections. Distances change as an object is viewed from even slightly different directions. Topological and convexity features are stable over larger ranges of viewing positions.

The clustering of sharp changes in different functions, together with lack of connectivity, is a well-known phenomenon in high-level reasoning. Consider a cup sitting on a table, as shown in Figure 13. The cup is not connected to the table. Furthermore, all manner of properties, from color to temperature to material composition, change abruptly at this boundary. Because of this clustering, it is possible to make intelligent guesses about material discontinuities relevant to manipulation on the basis of intensity or texture discontinuities discovered during visual processing. When abrupt changes in two properties, such as intensity and color, are observed in similar locations, they can be coelesced into one common boundary. Postulating a common boundary not only reduces the complexity of the representation.

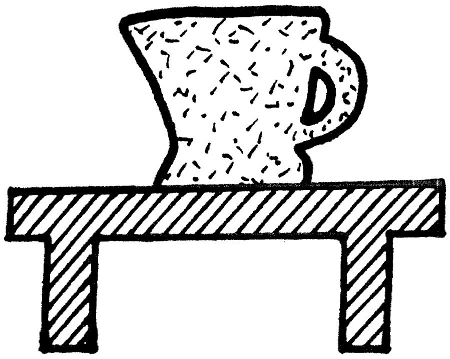


Figure 13. A cup sitting on a table.

When reasoning about changes in properties, we must consider the structure of time as well as the structure of space. Since we have few intuitions about temporal connectivity, evidence about boundaries in time comes almost entirely from the behavior of properties across time. Consider the process of freezing water in an ice-cube tray. As long as the temperature of the water remains above the freezing point, the temperature changes steadily, at a rate determined by the temperature of the freezer. When the water reaches the freezing point, however, its temperature stops falling (more or less), but more and more of the water changes to ice. When all of the water is ice, its temperature starts to fall again. Thus, as shown in Figure 14, we have three periods of time during which the rate of change of temperature varies smoothly and the water is in a constant set of phases. These periods are separated by boundaries at which a new phase appears or disappears and the rate of temperature change is abruptly altered.

An number of implemented reasoning algorithms (Forbus 1984, de Kleer and Brown 1984, Williams 1984, Kuipers 1984, 1986) are concerned with describing and predicting these patterns of change over time. During periods of smooth change, these *qualitative physics* algorithms use only rough models about rates

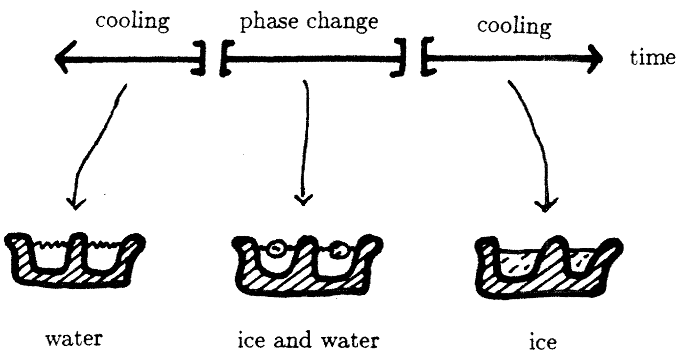


Figure 14. Freezing water involves three regions of smooth change over time, separated by boundaries at which something changes abruptly.

of change to predict property values. These predictions are used primarily to hypothesize what types of abrupt changes can take place and in what order. Estimates of metric information, such as how long the water will take to freeze, may also be provided. However, precise metric information is not essential. This type of algorithm has been used for reasoning about changes in temperature and phase, fluid flows, behavior of circuits, and motion of objects.

In addition to these new uses for boundaries and topological structure, high-level reasoning provides more examples of phenomena already seen in low-level vision or natural language. In Chapter 8, we see that researchers in high-level reasoning have had formal problems modelling sharp changes in properties across time, similar to those found by researchers in linguistic semantics. In reasoning, however, these problems occur in representations of 2D and 3D space, as well as in representations of time. Cellular models constrain space so as to avoid infinitely dense boundaries and phenomena such as Zeno's paradox. However, cellular models can represent the full variety of boundaries and regions needed for reasoning, including regions that touch themselves and boundaries that end abruptly.

In reviewing research on reasoning, we also see more examples of properties with wide support. Wide support regions are needed in producing coarse-scale models of situations, analyzing spatial texture, and analyzing textured patterns of events over time. Researchers in this area and in natural language semantics typically take it for granted that such support regions are trimmed so as not to cross relevant boundaries. However, there is more tendency to propose point-sampling models that can cause the aliasing and drop-out problems described in Chapter 2.

In Chapter 8, I also compare cellular models of phenomena to point-based models. Some previous researchers have advocated models in which one can refer to certain individual points, such as the exact boiling point of water, the top of the arc through which a thrown object moves, or the surface of an object. Although these methods can be accomodated within cellular topology, using the closed-edge model of boundaries, I argue that these "points" are not exact in real situations, even for such seemingly precise examples as phase equilibria. Thus, it may make more sense to allow for measurement error and refer to the cells adjacent to boundaries, rather than the points right at the boundary itself.

7. Conclusions

In this chapter, we have seen a wide range of application domains in which topological structure is useful. We have seen connectivity requirements appear in many places, including noise suppression in edge finding, building support regions for evaluating image matches, parsing object shapes into parts for identification, analyzing flows of fluid, planning motion of objects, determining the effects of forces on objects, and in describing sequences of actions across time. We have seen how homeomorphism can be used as a powerful constraint on matching two images, a task required by a number of low-level vision applications.

We have also seen some evidence from all domains that multiple functions tend to have sharp changes in value at the same locations. We have also seen that there is often a lack of connectivity at these same locations. This is evidence for the proposed model, in which all of these effects would be caused by a boundary in space (or time), and against a model that treated them as discontinuities in individual functions and isolated quirks in the definition of connectedness. The implemented edge finder also shows how these locations can be detected in real sensory input.

Digitized functions are commonly used in computer vision and occasionally in high-level reasoning. In Section 5, we saw how they may also be useful in explaining phenomena in natural language semantics. In all domains, we have seen examples of functions requiring wide support. These functions include those used in noise suppression, evaluating stereo matches, and descriptions of textured patterns in space and time. I have also briefly indicated a number of places in which previous researchers have had technical problems modelling boundaries. In Chapters 4-8, we return to all of these examples in more detail.

Chapter 4: The Edge Finder

1. Introduction

The first step in analyzing visual input is to detect locations of sharp changes in light intensity that might indicate the presence of boundaries in the scene. The edge finder implemented for this thesis uses a relatively standard approach, based on analyzing second directional differences of the image intensities. The main new feature of this algorithm is that it uses the topological structure of the responses in determining which responses represent real features and which are due to camera noise. The edge finder is named "Phantom" after Watt and Morgan's (1984) MIRAGE algorithm, to which it is closely related.[1]

I divide the problem of detecting boundaries into two sub-problems. First, the algorithm detects regions of the image in which directional second differences are significantly different from zero. The pattern of second differences is then analyzed to determine where boundaries should be hypothesized to account for the observed second difference responses. This decomposition of the problem dates back to Marr and Hildreth (1980). It allows one to separate the problem of suppressing effects of camera noise from the problem of classifying the wide variety of boundary shapes that occur in images of natural scenes.

Both steps in edge finding incorporate algorithms to suppress effects of camera noise. The current implementation contains only one noise suppression algorithm, which is used at three points during the edge finding process. As we saw in

[1] An earlier version of this edge finder is described in Fleck 1988.

Chapter 3, responses to real features are distinguished from those due to camera noise based on the sum of second difference responses over a maximal star-convex neighborhood of each cell. This constitutes the main use of topological structure in the edge finder.

Sections 2-3 present the algorithm for describing second difference responses. Since the noise suppression algorithm does not interact with the rest of the edge finding process, it is presented separately, in Section 4. Sections 5-7 discuss how the clean responses are classified and boundary locations hypothesized and Section 8 discusses problems of combining information from different scales. In Section 9, I present results of the edge finder on a range of images and Section 10 compares the Phantom edge finder to previous edge finding algorithms.

2. Taking differences

The Phantom edge finder finds boundaries in an image by locating regions of the image in which directional second differences are significantly different from zero. This is done by taking differences in several directions independently and then combining results over all directions. This produces a four-way classification of cells in the image, which is used in determining where to place boundaries. In this section, we see the details of this process, ignoring issues of image noise.

As we saw in Chapter 2, boundaries in space license abrupt changes in the behavior of continuous functions. These changes in behavior may involve changes in value that could not be achieved by any continuous function or, more often, changes in value that would require some other constraint to be violated, such as bounds on function differences or derivatives. In detecting boundaries from image intensities, we assume a bound on second differences of intensity.[2] Therefore,

[2] Strictly speaking, these differences must be normalized by the distance between the points used to determine the difference, before any bound is applied. In

any second differences larger than this bound must indicate the presence of a boundary.

Boundaries in images can be classified on the basis of the shape of the intensities in a straight 1D path across the boundary. Figure 1 shows several common intensity shapes and their second differences. In all of these patterns, intensity varies continuously. Because images are represented only to finite resolution and the space of intensities is connected, we can never observe a pattern of intensities in a digitized image that could not represent a continuous function. However, in the patterns representing boundaries, the second difference is significantly different from zero.

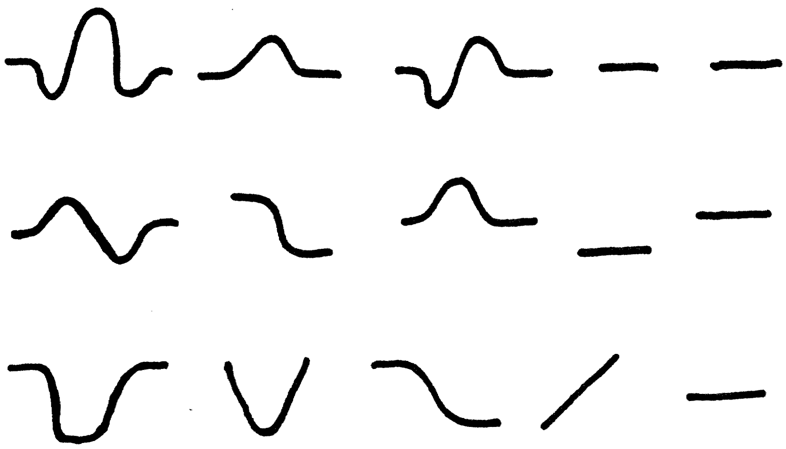


Figure 1. Common patterns of intensity values along straight 1D paths in an image. Left to right: no change, smooth variation in lighting, step edge, roof edge, thin bar. The top row shows the image intensities, the middle row shows their first differences, and the bottom row shows their second differences. The righthand three cases indicate the presence of boundaries, whereas the lefthand two cases do not.

We are more accustomed to thinking of abrupt changes in properties in terms

the applications discussed below, all differences are taken using a consistent spacing, so this point can be finessed.

of high first differences. For example, if I place a white cup in front of a dark background, first differences taken across this object boundary will be very high. However, in analyzing camera images, high first differences are not reliable evidence of a boundary. Smooth variations in light intensity and smooth shading can create high first differences even within a region whose physical properties (material, surface color, and so forth) are homogeneous. Imposing a bound on first differences would cause spurious boundaries to be reported in regions with variations in shading. Such markings would be intuitively unreasonable and unstable under changes in viewpoint and lighting.

Analysis of image intensities is not unusual in having usable bounds on second differences but not on first differences. This pattern occurs also in reasoning about changes in physical properties over time, because processes of change, such as boiling water or moving objects, often create high first differences (see Chapter 8 for discussion). When a textured surface is seen at an angle, perspective distortion causes the size of regions composing the texture to change rapidly across the visual field. In all of these cases, high second differences or changes in first difference sign reliably indicate the presence of a boundary and high first differences do not.

In analyzing camera images, or other real input, we do not have access to the underlying function values, but only to digitized versions of these values. Intensity values are smoothed before sampling, to avoid the aliasing effects discussed in Chapter 2.[3] The second difference values depend on the amount of smoothing and the density at which the image has been sampled. However, the Gaussian-like smoothing used in most computer vision systems consistently de-

[3] More or less. I have occasionally seen aliasing in video camera images, so apparently the smoothing is not exactly the right shape to accomplish this goal or perhaps some of it is applied after, rather than before, smoothing.

creases differences taken between any two points in the image. Thus, any high difference detected in an image must reflect a high difference in the underlying continuous intensities. The converse, naturally, does not hold.

The second differences used in my edge finder are taken along a straight five-cell path. Intensities at the cells along the path are added together, weighted by the values $[-1,0,2,0,-1]$.[4] All other things being equal, differences should be computed using cells as close together as possible, to provide the most detailed representation. However, the narrowest second difference, using three-cell paths with weights $[-1,2,-1]$, detects artifacts due to the interlacing used in most video cameras. The differences are taken along straight paths, because the processes responsible for high first differences in images produce differences that are constant along straight paths, at least locally.

Readers familiar with recent research in computer vision may notice that I have been very cautious in making assertions about the real world. It is currently the fashion for theoretical analyses of computer vision algorithms to build very precise models of reality. Unfortunately, these more specific models are typically unverifiable or, in some cases, incorrect. For example, it is often stated that physical properties change discontinuously across boundaries. First of all, not even the physicists have any solid evidence about the differential structure of space and an algorithm whose input is digitized can hardly depend on structure finer than its digitization. Secondly, at a macroscopic level, most physical processes change in a way that seems continuous, if viewed at a high enough resolution.

[4] The usual definition of the second difference is the negative of this mask. I have inverted the mask so that lighter regions of the image produce positive values, because that seems intuitively more natural.

3. Combining results from different directions

In the previous section, we have considered only how individual directional differences indicate the presence of boundaries. Directional differences, however, can be taken in number of directions about each cell in a 2D image, although digitization limits this to a finite set of distinct directions. In this section, we see how to summarize the pattern of differences about each cell into a single label for that cell.

The basic idea behind Phantom's method of summarizing second differences is that differences between cells can be described well by grouping them into four classes. Consider the four cells shown in Figure 2. The first type of cell, which is labelled *zero* has no significant second difference response. If there were no noise, "significant" would be determined by the bound on second differences. However, in practice, the bound is concealed by the stricter requirement that it must be possible to distinguish the second difference response from the effects of camera noise.

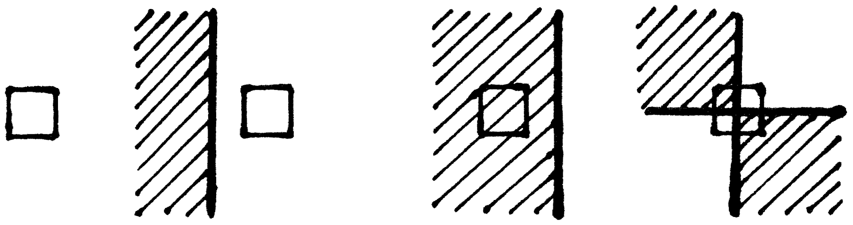


Figure 2. The four types of cells. Left to right: no significant second differences, positive second differences, negative second differences, a mixture of positive and negative second differences.

The other three types of cells have significant second difference responses. In the first two cases, directional differences crossing the boundary have significant amplitude. All of them have the same sign and the largest amplitude occurs when the difference is taken perpendicular to the boundary. Differences that do not cross the boundary may have either sign, but they have much lower amplitude. In these cases, the pattern of second differences can be reasonably summarized by giving the sign and amplitude of the strongest response. Cells of this type are labelled *dark* or *light*, as appropriate.

In the final example, the cell has both significant positive and significant negative second differences. These cells are *saddles*. Specifically, in order to distinguish this case from *dark* and *light* cells, the edge finder considers the amplitudes of the strongest positive and negative responses, across all directions. If the weaker response is at least $\frac{6}{10}$ of the stronger, the cell is considered a saddle. Cells labelled as saddles are considered to lie in the middle of the boundaries, when boundaries are finally generated. Identifying such cells is crucial to insuring connected boundaries when multiple regions touch at a common vertex.

Thus, all four types of situations can be distinguished by finding the maximum amplitude positive and negative second difference responses, over all directions. Figures 3 and 4 shows an image, directional differences in one direction, and the directional differences combined over all directions.[5] As you can see, the directional difference only responds well to boundaries that are perpendicular to the direction in which the difference is taken. Boundaries parallel to the direction of the difference are not detected at all and the locations of boundaries at other angles are distorted. The combined result, however, detects boundaries of all orientations correctly, because the highest amplitude responses come from differences perpendicular to the boundaries.

In the current implementation, differences are taken in four directions: hor-

[5] These outputs have also received the noise suppression described in Section 3.

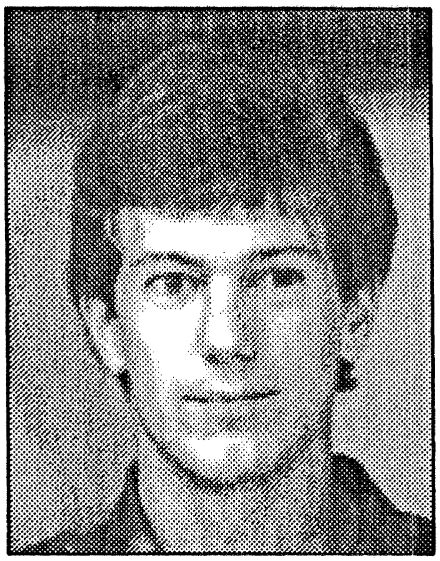


Figure 3. A digitized image (330 by 420 cells).

izontal, vertical, and two diagonal directions. The algorithm has been tested with other sets of directions and it makes little difference to the output. Performance is improved slightly as more directions are used. Output also seems to be changed only slightly when the differences are taken using triples of cells that deviate slightly from straight lines.

This method of classifying cells performs well on two types of situations that cause problems for most edge finders: sharp corners and vertices at which several regions meet. Good examples of this problem are shown in Chapter 9, Section 6, where Phantom's performance is compared in detail against that of Canny's (1983, 1986) edge finder. Similar problems occur for many other edge finders. These problems occur because these edge finders make stronger assumptions about the pattern of directional difference responses over different directions. When the responses do not fit this pattern, the edge finder typically fails.

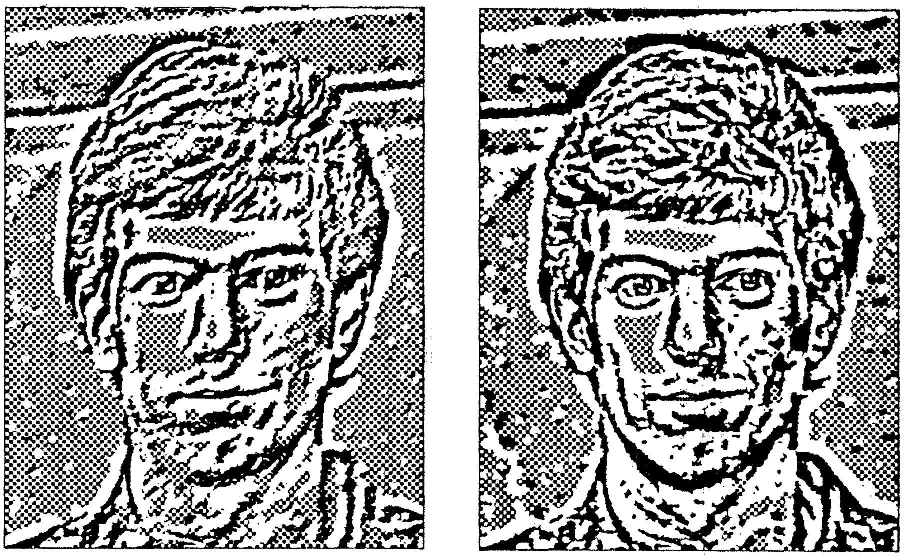


Figure 4. Left: Sign of directional difference in one direction (diagonally down and to the right) for the image in Figure 3. Right: sign of directional difference combined over all directions. Positive responses are shown in white, negative responses in black, and zero responses in a checkerboard pattern.

Consider first the Marr-Hildreth edge finder (Marr and Hildreth 1980, Hildreth 1983). This edge finder uses the sign of either the Difference of Gaussians or the Laplacian of a Gaussian to classify cells as dark or light. In either case, ignoring issues of smoothing and noise suppression, the effect is similar to taking second differences in a number of directions, evenly sampling the space of directions, and adding them together. This works properly on straight boundaries, because positive and negative responses are approximately balanced at the boundary. Near sharp corners, however, the sum is skewed because cells inside the corner have too many responses of the correct sign and cells outside the corner have too few, as shown in Figure 5. As Berzins (1984) shows, the boundary shape is deformed near the corner. Furthermore, since the outside response is weak, the boundary shape tends to be corrupted by camera noise. Ulupinar and Medioni (1988) and Chen and Medioni (1987) discuss other types of bias in the locations produced by this type of edge finder. However, their method of reducing them has not been extensively tested.

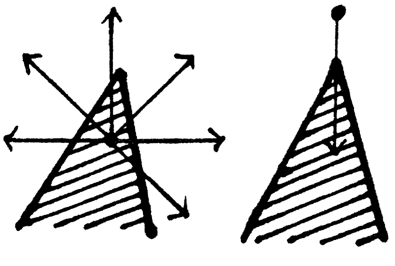


Figure 5. Cells inside sharp corners have high amplitude directional difference responses from too many cells and cells outside the corners have responses from too few directions.

Canny's edge finder has problems on corners for a different set of reasons. His edge finder detects local maxima of the first difference. He assumes that the first differences about each cell approximate a linear transformation. Therefore, he takes directional differences in only two directions and uses these to compute gradient direction and magnitude. Unfortunately, this approximation fails near sharp corners, such as the one shown in Figure 5, and region intersections, as in the lefthand picture in Figure 4.

The problem here is that differences only behave like derivatives in the limit. It is plausible to assume that the intensities underlying all of these situations are continuous, since the image has been smoothed by the camera system. Thus, we are guaranteed that first differences about each point in the image approximate[6] a linear transformation within some neighborhood of the point. This is the Taylor series approximation from standard Calculus. However, there is no guarantee

[6] For any desired goodness of fit.

that this neighborhood is even one cell wide. Thus, finite differences taken one or two cells apart may fail to approximate a linear transformation.

Because its assumptions are not satisfied, Canny's edge finder displays a number of undesirable behaviors near sharp corners and region intersections. The exact behavior depends on details of the image, including the angle between the boundaries and the two directions in which differences are taken. It may deform the boundary shape, break the boundaries, and/or create spurious boundaries. Phantom avoids these problems by making weaker assumptions about the pattern of second difference responses near boundaries. Detailed examples of the behavior of both edge finders are presented in Chapter 9.

4. Noise suppression

The algorithm described in the previous two sections does not consider effects of image noise. If this algorithm were run by itself, with no noise suppression, the results on the example image would look as shown in Figure 6. Even images that do not look noisy have considerable high-frequency fluctuation in intensity values. This section describes a new algorithm, based on star-convex neighborhoods, that removes these effects of noise and produces clean outputs, like those shown in Figure 4.

The traditional method of eliminating camera noise consists of two parts: smoothing and thresholding. First, the image is smoothed prior to edge finding. Since camera noise is largely concentrated in the high frequencies, this tends to reduce the amount of noise relative to the amount of response to real features. An edge detection process is then run and its responses are thresholded to eliminate responses due to the remaining noise. However, available methods of measuring response strength have not been very sensitive and thus excessive amounts of

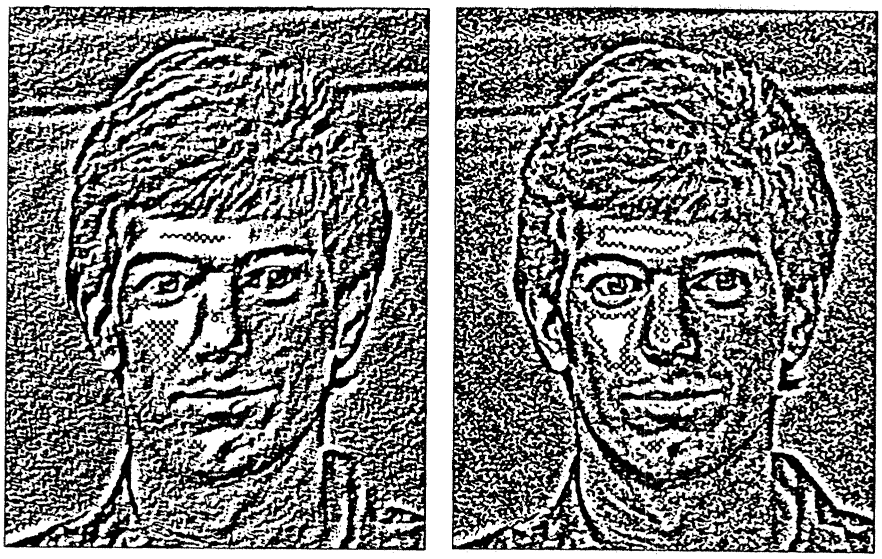


Figure 6. A directional difference (left) and the combination of differences from all directions (right), with no noise suppression.

smoothing are required in order to eliminate noise. Therefore, previous edge finder have had difficulty detecting fine texture and fine details of boundary shape.

The Phantom edge finder uses a new method of distinguishing real responses from noise that takes advantage of both the response amplitude at each cell and the shape of the response region. For low-noise images, such as those produced by modern camera systems, this method can distinguish real features from noise without any image smoothing. Under higher noise conditions, smoothing becomes desirable, but less smoothing is required to achieve stable output than in previous algorithms. By reducing the amount of smoothing, the Phantom edge finder can detect more fine detail than has previously been possible.

Second difference responses due to random camera noise have two properties that are useful in distinguishing them from responses representing real features of the scene. First, responses due to noise have low amplitude at all cells. Secondly, noise responses vary in sign, forming only small regions of the same sign. Real scene features typically generate responses that have higher amplitude than effects of noise. Furthermore, camera systems blur the image before introducing noise, so that real boundaries are blurred but noise is not. Thus, even when real responses have amplitudes similar to that of noise, they typically generate responses that are both longer and broader than those due to noise. This is illustrated in Figure 7.

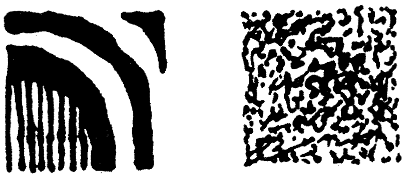


Figure 7. Responses to real features are generally longer and wider than responses due to camera noise.

Amplitude and shape information can be combined by summing amplitude over the response region. This technique was proposed by Watt and Morgan (1984) and has also been used by Huertas and Medioni (1986) and, in curvature analysis, by Huttenlocher (1988). In the 1D cases considered by these authors, the strength of the response region containing each cell can be assessed by summing responses over the largest connected region about that cell in which responses have a consistent non-zero sign. This is illustrated in Figure 8.

There are three problems involved in extending this approach to 2D images. First, some bound must be placed on the radius of the region used in summing, because connected response regions can extend for substantial distances across the image. Secondly, even within a restricted radius, connectivity is too weak

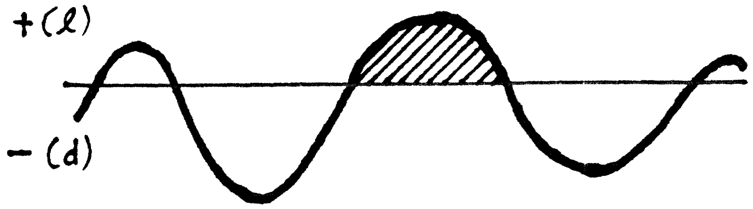


Figure 8. To measure the strength of a second difference response region in 1D, responses are summed over each connected region of same-sign responses.

a requirement on the region shape. Full connected regions are expensive to compute. Furthermore, they provide poor discrimination between noise and real responses, because noise can generate quite large connected regions. Finally, noise not only creates spurious response regions where the response should be zero, but also breaks up real response regions.

Phantom defines sensible regions for summing responses using the maximal star-convex neighborhoods defined in Chapter 2. These neighborhoods are restricted in radius (currently at most 3 cells from the starting cell) and are not allowed to contain cells whose sign does not match that of the starting cell.[7] As Figure 9 illustrates, since these neighborhoods cannot cross regions of opposite sign, they are confined to one response region. The star-convexity requirement prevents large neighborhoods from being generated in twisty response patterns typical of noise. At each cell in the image, the sum of responses over the star-convex neighborhood of that cell gives a robust evaluation of whether the response at that cell is due to noise or to a real response.

The examples shown in Figure 9 only contain *dark* and *light* cells. As we saw in Section 3, cells can also be labelled *saddle* or *zero*. Two steps are taken

[7] Cells labelled *saddle* can belong to neighborhoods of either sign.

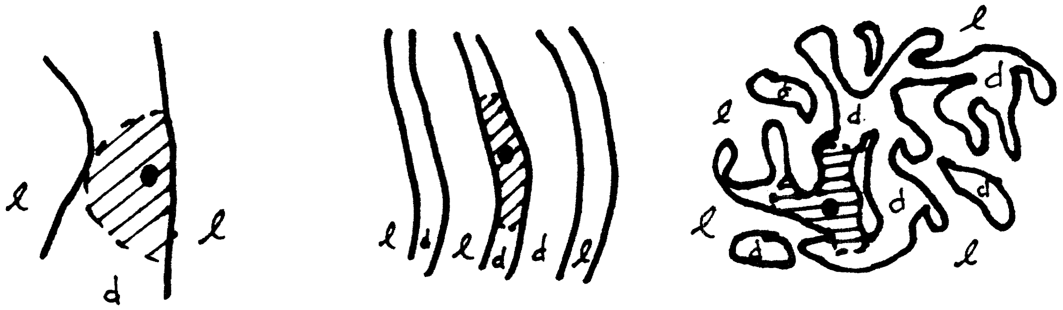


Figure 9. The star-convex neighborhoods about cells are restricted in radius (left), cannot cross into neighboring response regions (middle) and do not follow twisted response regions due to noise (right).

to handle these cells. First, two sums are computed for each of these cells, one treating them as if they were *light* and one treating them as if they were *dark.*[8] Secondly, the star-convex neighborhood about each cell[9] is allowed to contain cells labelled *zero* or *saddle*, as illustrated in Figure 10.

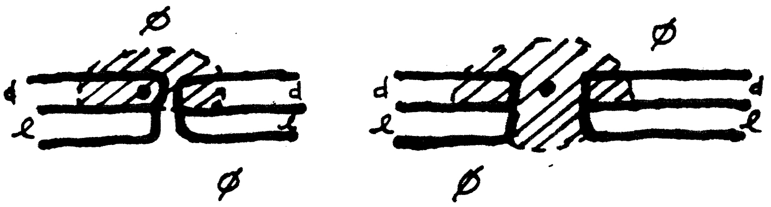


Figure 10. The star-convex about a cell can cross cells labelled *zero*. Left: the star-convex neighborhood of a *dark* cell. Right: the star-convex neighborhood of a *zero* cell, if it is treated as if it were labelled *dark.*

The evaluation at each cell is used to re-label cells as *zero* if they reflect only the effects of camera noise. If the cell has the label *dark* or *light* but an evaluation

[8] One could consider that two computations are also done for the *light* and *dark* cells, but one of the computations is guaranteed to return zero.

[9] No matter which of the four labels it bears.

below the noise threshold, it is relabelled as *zero*. Cells labelled as *saddles* are treated as being labelled both *light* and *dark*. Either or both of these labels can be removed if the corresponding sum is below the noise threshold. The noise threshold must be adjusted for the camera setup in use. For the images presented in this thesis, the noise threshold is set at 60, based on the results of evaluations presented in Chapter 9.

The cell evaluations are also used to fill small gaps in response regions. If a cell is labelled *zero* but one of its sums is above the noise threshold, the cell is re-labelled *dark*, *light*, or *saddle*, as appropriate. This allows small gaps in response regions to be filled with an appropriate label. When the second difference response happens to be zero in the middle of a zero crossing,[10] this process labels it as a *saddle*. As described in Section 7, these saddles allow boundaries with the correct topology to be generated in these cases.

Noise suppression is done at three points in the Phantom edge finder algorithms. It is used first to clean up directional differences taken in each individual direction. Weak responses to real features are easier to detect in the individual directional responses than in the combined response. The same algorithm is used a second time to clean up the result combined from all directions. In each case, noise suppression is done twice. The main reason for the second pass is to fill in holes in response regions created where the first pass suppressed responses with the wrong sign. As we will see in Section 6, noise suppression is also used in identifying response regions not due to step edges.

As you can see by comparing Figures 4 and 6, this method of suppressing

[10] This is not as unusual as it may seem. Responses are only represented to 8 bits of precision. Blurred edges often have low amplitude near zero crossings and can easily generate zero responses, particularly after one round of noise suppression.

noise is quite effective, even in the absence of smoothing. Smoothing is only used for a few images presented in this thesis, taken with particularly noisy camera setups.[11] Even in these cases, less smoothing is required to achieve stable output than with previous edge finders. This technique was designed to work on noise roughly resembling Gaussian noise. Other techniques would need to be employed for camera systems with very different noise characteristics. For example, Horn and Woodham (1978) present techniques for de-striping images.

A final point to note is that the noise in many camera systems is primarily high frequency. As described in Section 8, Phantom is run not only on the original image, but also on coarser-scale versions of the image. Although the current implementation uses the same noise threshold for all scales, it should probably be adjusted for each scale independently. Since the sub-sampled images have much less noise than the finest scale, it is important not to test algorithms on sub-sampled images.

5. Inducing boundaries

The algorithms described in Sections 2-4 produce clean maps of significant second difference responses in an intensity image. The final step in edge finding is to hypothesize boundary locations that might explain these observed response patterns. In this section, I discuss how boundaries are hypothesized for responses due to step edges. In the next section, I show how to identify responses that cannot be accounted for in this way.

Most boundaries in camera images can be roughly approximated as *step edges*. The simplest type of step edge is shown in Figure 11. This type of boundary generates a characteristic pattern of second difference responses, in which a *dark*

[11] These cases are all explicitly marked.

response region touches a *light* response region, occasionally with some *saddle* cells on the boundary. The boundary is located where *light* and *dark* cells touch and where there are *saddle* cells. This is illustrated in Figure 12. For consistency with traditional terminology, I refer to these locations as *zero-crossings.*

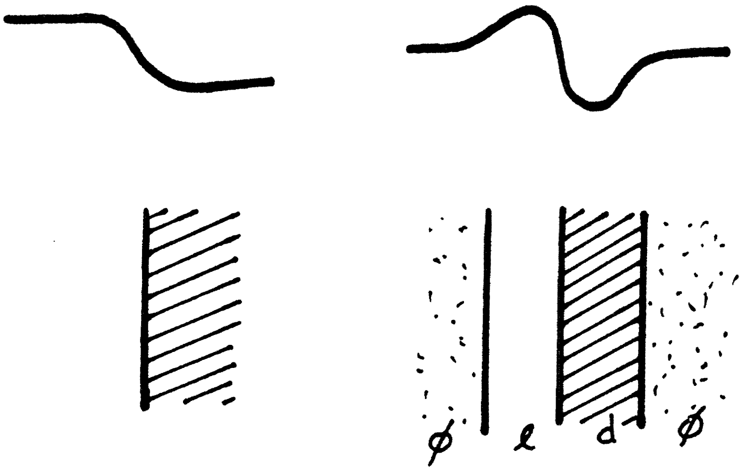
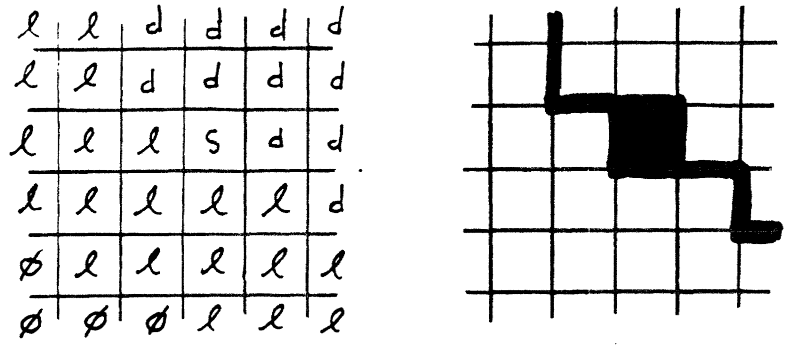


Figure 11. A step edge and its second difference. Top: intensity profile. Bottom: 2D response regions.

In natural images, step edges may have slightly different intensity profiles and/or variations in 2D shape, as illustrated in Figure 13. For these variant intensity profiles, there is no generally accepted definition of where the boundary should be placed. Cellular topology tells us that there should be some boundary in such a response pattern, but does not provide any direct guidance as to where it is. The simplest option seems to be to treat these responses just like the step edges and hypothesize boundaries at zero-crossings. Then we can deduce boundaries from the second difference responses using the following rule:

- Place an adjacency set in the boundaries whenever it contains either a cell labelled *saddle* or both a cell labelled *dark* and a cell labelled *light.*



l	l	d	d	d	d
l	l	d	d	d	d
l	l	l	s	d	d
l	l	l	l	l	d
ø	l	l	l	l	l
ø	ø	ø	l	l	l

Figure 12. The labelling of the cells shown on the left induces the boundaries shown on right. Boundaries are placed on adjacency sets containing both a *dark* (*d*) cell and a *light* (*l*) cell. These adjacency sets correspond to the boundaries of cells. Adjacency sets containing a cell labelled *saddle* (*s*) are also placed in the boundaries. In particular, each single-cell adjacency set containing a *saddle* cell is placed in the boundaries. These adjacency sets correspond to whole cells.

Figure 14 shows an example of boundaries found by this method. Since most boundaries fall between cells, this figure shows both boundary cells and cells to the dark sides of boundaries, to insure connected boundaries.[12]

One way to represent these boundary assignment rules is to model the set of labels as the cellular space shown in Figure 15. If cell labels (after noise suppression) are assumed to have little or no measurement error, the rules follow directly from this representation. Since the *saddle* label is a boundary cell, any cell mapping onto it must also be a boundary cell. Since there is a boundary

[12] Exact display of boundary output requires enlarging the image by a factor of two in each dimension, so that locations between cells can be represented. Chapter 9, Section 6 shows such enlargements for small details of images. However, they become unwieldy for larger images. The dark cell representation is inspired by the discussion given by Pearson and Robinson (1985). They point out that if boundaries are drawn darker than the background, boundaries in the line drawing are perceived as being at edges of the dark lines. Thus, if boundaries are represented by dark lines on a light background, they should be drawn slightly to the dark side of the boundary.

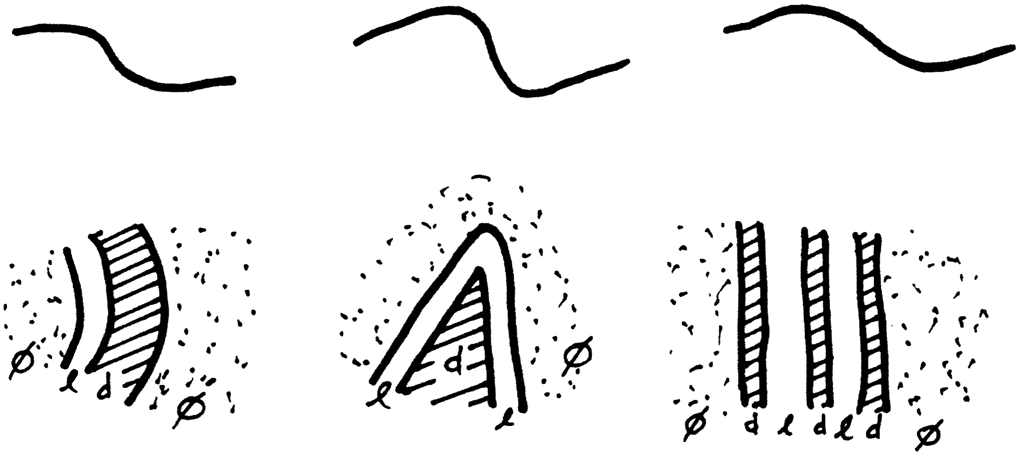


Figure 13. Variations in step edge shape. Top: variations in intensity profile. Bottom: variations in 2D shape of boundary.

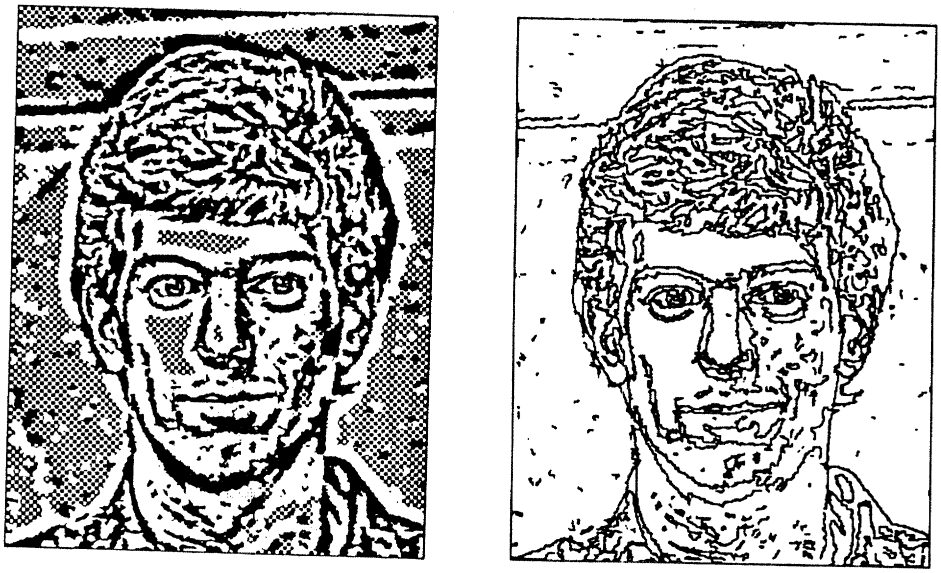


Figure 14. Significant second difference responses, boundaries placed at zero crossings.

between the *dark* and the *light* label, a continuous map changing from one label to the other would have to pass through the label *zero*. If measurement error is assumed to be small enough, we can assume that such a transition would always generate at least one cell labelled *zero* in the path. Thus, a direction transition between *light* and *dark* also indicates the presence of a boundary.

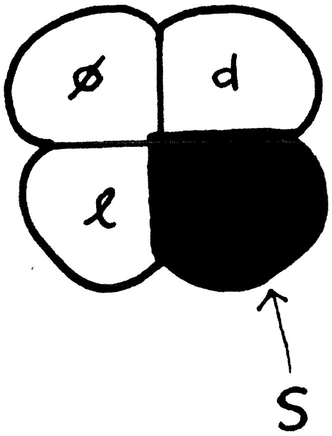


Figure 15. A cellular representation of the value space consisting of the four cell labels.

For most situations found in natural images, these rules provide boundary locations that are stable, that are intuitively acceptable, and that can be used successfully in most computer vision applications. There are two sources of criteria for evaluating theories of boundary placement. First, to the extent that we want to emulate human performance, we can make use of human intuitions about boundary placement. This is helpful for general guidance, but it is difficult to obtain precise psychophysical data in this area. Secondly, we can consider the requirements of applications using the boundaries. The stability evaluations provided in Chapter 9 are a first step towards developing such criteria. It is popular in computer vision to develop theories of boundary placement by considering what types of 3D objects might have created the image. However, this approach does not solve the problem, but simply pushes it back one step, since there is no generally definition of where the edge of a 3D object is.

6. Identifying other types of responses

Marking boundaries at zero-crossings accounts for many of the situations found in natural images. However, there are two configurations in which this method performs poorly. On staircase-like intensity patterns, the algorithm may generate spurious "phantom" zero-crossings. In pictures of scenes with large amounts of smooth shading, it is also possible to get regions of significant second difference responses that are not well explained in terms of zero crossings. This section discusses these two cases and how they might be handled.

Under one set of conditions, the zero-crossing rules given in Section 5 cause the Phantom edge finder to hypothesize intuitively unacceptable boundaries. In staircase patterns, the dark response region from one boundary may touch the light response region from another boundary. This is illustrated in Figure 16 and real edge finder examples are shown in Chapter 9, Section 6. The rules for deducing the presence of boundaries mark this label transition as a boundary. These spurious responses only happen when the regions in the staircase are relatively narrow, less than about 10-12 cells in width.

I do not know of any robust way to identify and remove extraneous boundaries in staircase patterns using only one scale of analysis. In theory the sign of the intensity change should not agree with the dark/light labelling at such a phantom boundary (Clark 1986, Ulupinar and Medioni 1988, Chen and Medioni 1987). However, I have not been able to convert this observation into a robust algorithm. The problems lie in distinguishing these spurious boundaries from

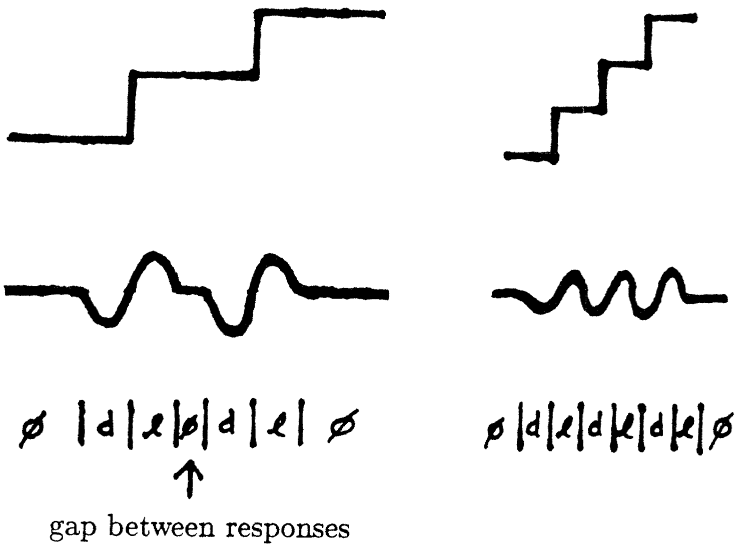


Figure 16. Zero-crossings of second differences need not correspond to step-edge boundaries. Top to bottom: two staircase intensity profiles, their second differences, and their dark/light labelling. For the narrower staircase, zero-crossings are not only created between the steps, but also in the middle of them.

real boundaries that have low contrast. Watt and Morgan (1983) suggest that humans may also have problems correctly interpreting fine staircase patterns.

Since the Phantom edge finder produces multi-scale output, as described in Section 8, it may be possible to eliminate many phantom boundaries by comparing edge finder output at different scales. In order for this to succeed, the phantom boundaries must occur at a scale that is not the finest representation of the image and the staircase pattern must be correctly represented at some finer scale. Chapter 5, Section 5 discusses briefly how edge finder output from different scales can be compared, to determine where the representation has changed between the two scales. Suppose this matching process can be modified so that fine and coarse scale representations match even when a phantom boundary appears only at the coarse scale. For example, we might fill in *zero* regions in both representations, so that both scales have phantom boundaries. Staircase phantom boundaries could then be identified as boundaries present only at the coarse scale, but in regions where the two scales match exactly.

A second type of problem with the zero-crossing method is that there are occasional second difference responses that do not fit the step edge pattern. Figure 17 shows two images containing such responses.[13] In some cases, the response region is simply not connected to a zero crossing. In other cases, the region is connected to a zero-crossing, but it is too wide or has the wrong amplitude profile to be entirely due to a step edge at that zero-crossing. Previous proposals for parsing algorithms, such as Watt and Morgan's (1984) MIRAGE algorithm, have considered only the first type of example. Both types of examples seem to be relatively rare in natural images. I have implemented an algorithm to identify such regions, but it is unclear where to hypothesize boundaries to explain them.

Phantom identifies responses not due to zero-crossings by estimating how much of the second difference response might be due to the observed zero crossings. This is done using an algorithm that examines straight paths through response regions. The path is required to start at a zero crossing and it is terminated when a zero-crossing boundary is reached, as shown in Figure 18. The first three elements of the path are assumed to belong to the zero-crossing response and are used to estimate the height of the response pattern. If the average of these first three values is h, the zero-crossing response for the next four cells is assumed to have the pattern h, $\frac{h}{2}$, $\frac{h}{4}$, $\frac{h}{8}$.[14] All response up to these levels is marked as belonging to the zero-crossing response.

[13] The image containing the hand was smoothed using a Gaussian with $\sigma = 1$ cell before edge finding, due to high noise conditions.

[14] This model was produced by informal experimentation, based on the second difference of an ideal step edge with Gaussian smoothing.

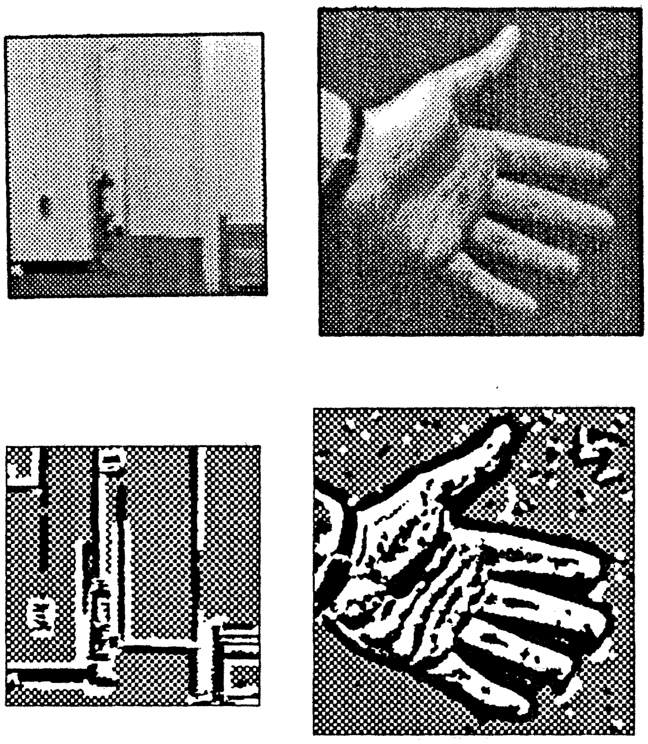


Figure 17. Top: two images containing examples of second difference responses not due to step edges. Bottom: significant second difference responses for the two images. The image of the room corner has very low contrast and has been displayed with enhanced contrast. The image of the hand has been smoothed prior to edge finding, because it was taken with a noisy camera system.

The marking algorithm is repeated for horizontal, vertical, and two types of diagonal orientations (both opposite directions are considered for each orientation). At each cell, the algorithm accumulates the maximum response amplitude that could be due to a zero-crossing, over all path directions. These responses are subtracted from the original response amplitudes, to yield a map of response amplitudes not due to zero-crossings. The noise suppression algorithm (two passes) is used to clean up these response regions, yielding the clean map shown in Figure 19. As you can see, it does a relatively good job of identifying the problem

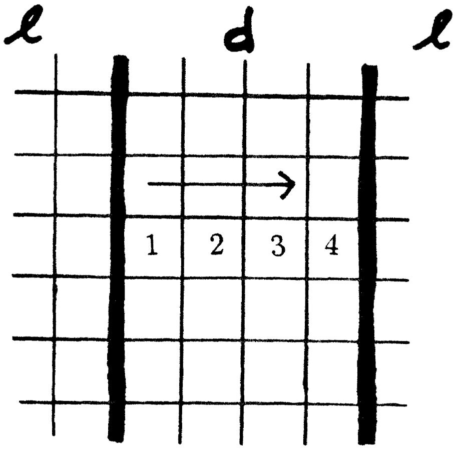


Figure 18. Responses due to zero-crossing boundaries are identified by examining straight paths through the response regions. Each path must start with an edge cell, i.e. a cell next to a zero crossing. It continues until another zero-crossing boundary is reached.

regions.

Having identified response regions not due to zero-crossings, the algorithm should hypothesize boundaries to account for them. Unfortunately, it is unclear where these boundaries should be placed. The traditional suggestion is that boundaries should be placed at the point of maximum response amplitude. However, notice that these response regions often continue the line of one side of a zero-crossing response. In many cases, the intuitively best location for the boundary would be along one side of the response region. Such a placement would insure that boundaries remain connected when they shift between zero-crossing and non-zero-crossing response patterns, but it requires a method for determining which side of the response region to place the boundary on. A final option would be to treat all cells in non-zero-crossing response regions as boundary cells. Designing a robust method of hypothesizing boundaries for these

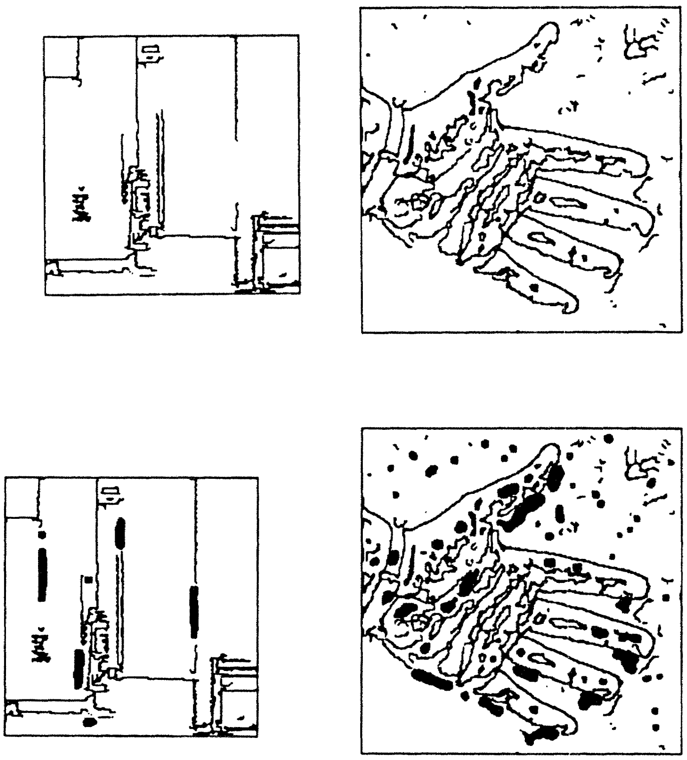


Figure 19. Top: zero-crossing boundaries for the images in Figure 17. Bottom: zero-crossing boundaries, together with response regions not due to zero-crossings (shown as black regions).

response regions requires examination of more examples than I have been able to gather.

7. The form of boundaries

Cellular topology allows a wide variety of boundary shapes, because it imposes few restrictions on boundary shape and because it allows both inter-cell and on-cell boundaries (including thick boundaries). The Phantom edge finder takes advantage of this flexibility to produce stable representations for the full variety of natural boundary shapes. Previous representations have imposed more restrictions on the form of boundaries. For example, they may allow only inter-cell or only on-cell boundaries or prohibit boundaries from ending abruptly. In this section, we see that these restrictions cause problems in handling real input.

The most common restriction on the form of boundaries is a requirement that they occur either between cells or on cells, but not both. Most current edge finders (e.g. Canny 1983, 1986, Sher 1987, Heurtas and Medioni 1986) seem to use on-cell boundaries. A few algorithms, including Geman and Geman (1984) and Blake (1983), use inter-cell boundaries. Both of these choices create problems.

The main problem with on-cell boundaries is that they use up cells that could otherwise be used to represent regions. This can be a problem in fine texture, where regions occasionally narrow to only one cell in width. A second problem is that many edge finder algorithms, particularly those based on first or second differences, most naturally locate boundaries between cells. Placing boundaries on cells requires introducing a small bias into the edge locations (as the MIT implementation of Canny seems to do) or using complicated tests to insure that the best on-cell approximation is chosen (see Huertas and Medioni 1986).

Inter-cell boundaries, on the other hand, misrepresent the boundary topology when a boundary location falls in the middle of a cell. If the boundary is close to the middle of the cell, the edge finder may not be able to make a stable decision as to which side of the boundary to place the boundary on. This happens particularly often when the boundary is low contrast or blurred and thus has low response amplitude near the boundary. In such a situation, choosing either site, or even both sites, leads to incorrect boundary topology, as shown in Figure 20. In such situations, the Phantom edge finder treats the disputed cells as belonging entirely to the boundary. Although this makes the boundary thicker, it insures

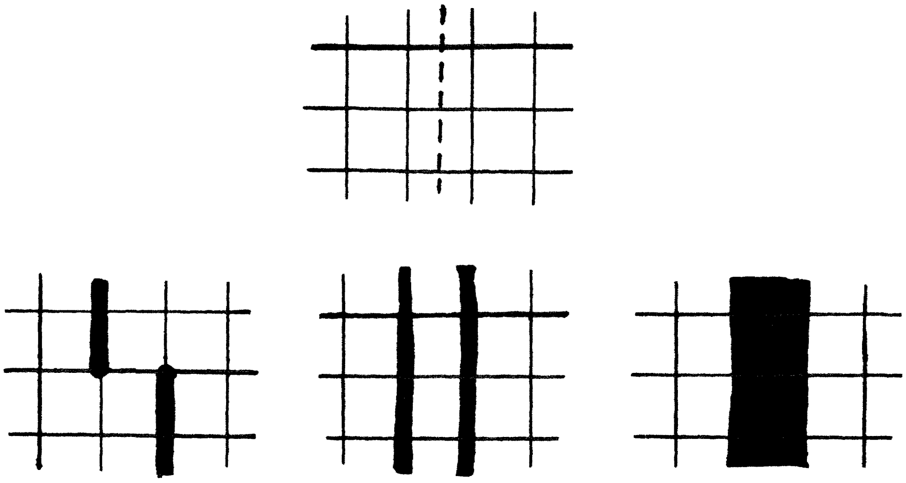


Figure 20. When a boundary falls midway between two sites, a connected boundary cannot be insured using only inter-cell boundaries. If only one site is chosen (left), but the choice is unstable, the boundary is broken. If both sites are chosen (middle), an extraneous region is formed. With on-cell boundaries (right), the correct topology can be insured.

the correct topological structure.

Occasional use of boundary cells is helpful in other situations. For example, they can be used to represent blurred boundaries. They are essential in formalizing the boundary motion operations used in Chapter 5. Finally, they are useful in representing situations in which many regions touch at a point. Such situations may be difficult to represent using only inter-cell boundaries, if the regions are a poor match to the digitization. For example, in a hexagonal cell arrangement, only three cells touch at each vertex. Thus, a checkerboard in which four cells touch at one point cannot be represented directly using only inter-cell boundaries, as shown in Figure 21. Stable representations for such situations can be achieved by using small numbers of on-cell boundaries near the intersection point. Chapter 9, Section 6 shows many examples of boundary cells in Phantom's output.

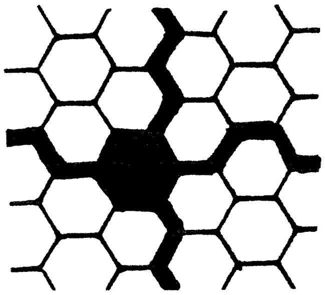


Figure 21. In a hexagonal cell arrangement, only three cells touch at each vertex. Therefore, a situation in which four cells touch can only be represented using boundary cells.

A second restriction sometimes imposed on the form of boundaries is that boundaries cannot end abruptly in the middle of a region. The algorithms proposed by Geman and Geman (1984) and Blake (1983) strongly discourage such boundaries and they are forbidden by region-based segmentation algorithms. However, we saw in Chapter 2 that such boundaries can occur in 2D views of scenes. In some cases, they represent slits in the 3D object that terminate abruptly in the middle of a 3D region. However, as Koenderink and van Doorn (1982) show, such boundaries can be produced in an image even when the 3D object represented by the image is smooth and has no internal boundaries. Thus, such a restriction would make it impossible to correctly represent the boundaries in many natural scenes.

8. Multi-scale output and reconstruction

The output magnitudes and labels after direction combination provide representations of the image at multiple scales of resolution. This multi-scale representation is used in later applications, such as stereo matching. By reconstructing the image from the edge finder output, we can see that very little important information has been lost during this processing. When only the sign bits of the edge finder output are used, the image is still recognizable, but smooth shading information is lost. Removing sign information entirely makes the image difficult to interpret.

The Phantom edge finder is run not only on the original image, but also on smoothed and sampled versions of the image. Thus, it analyzes each image at a range of resolutions. Each sampled version of the image is only one quarter the area of the next finer version. Thus the entire multi-scale computation takes only $\frac{4}{3}$ times as long as the computation for the single finest scale.[15]

Multi-scale results for two images are shown in Figures 22-23. The first image preserves the same structure at coarser scales, except for loss of detail. The second image contains blurry boundaries that appear only at coarser scales and thus it exhibits qualitative changes in representation between scales. In Chapter 5, we will see how these two cases might be distinguished by matching results of adjacent scales. In this section, I will discuss ways of displaying multi-scale output.

The information present in such a multi-scale representation can best be appreciated by reconstructing the original image from it. There are quite a number of ways in which reconstructed images can be produced, suitable for different types of applications. These reconstructions are useful for display purposes and also for assessing what types of information would be lost if certain parts of the representation were not used. Figures 24-25 show four ways of displaying the information in the edge finder output.

Figure 24 (top) shows a representation in which coarse-scale labels are used to fill in areas with no significant fine-scale response. This filling process starts

[15] Because the image sizes form a geometric series.

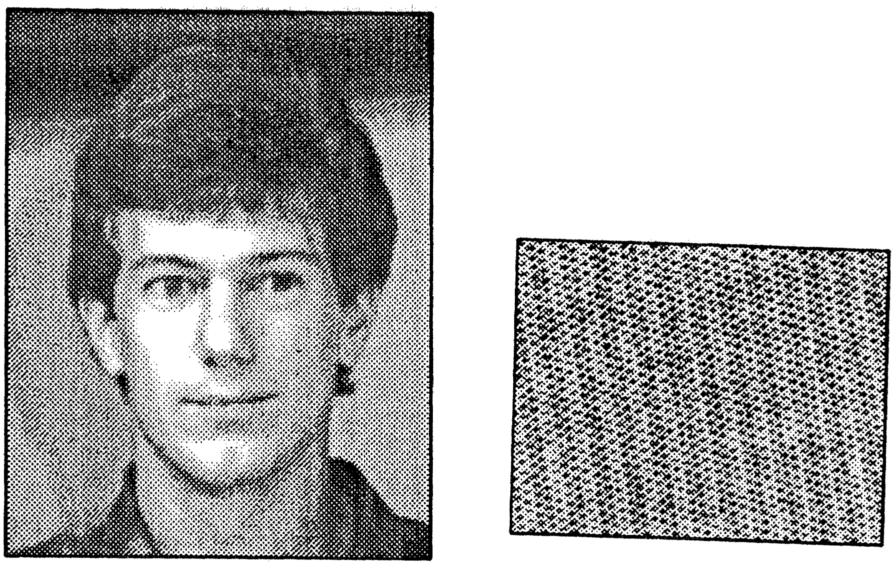


Figure 22. Two images (330 by 420 cells and 288 by 227 cells).

at the coarsest scale and proceeds to finer scales. At each step, the coarser-scale labels are expanded and smoothed, and then used to fill in regions labelled *zero* at the next finer scale. This technique results in a vivid binary cartoon of the image. My experience has been that individual people are easily recognized from this type of representation. For comparison, it often requires some thought even to identify human faces in dark edge displays, such as those shown in Figure 24 (bottom).

Figure 25 shows two grey-scale images reconstructed from the edge finder output. The top version uses only the dark/light labelling at all scales. The bottom version also takes account of the magnitude of edge finder responses at each cell. In both cases, reconstruction proceeds from coarse to fine scales. At each step, the reconstruction based on coarser scales is interpolated (by expanding and then smoothing) and combined with the edge finder results from the next finer scale. This yields a finer-scale reconstruction of the image. This process is

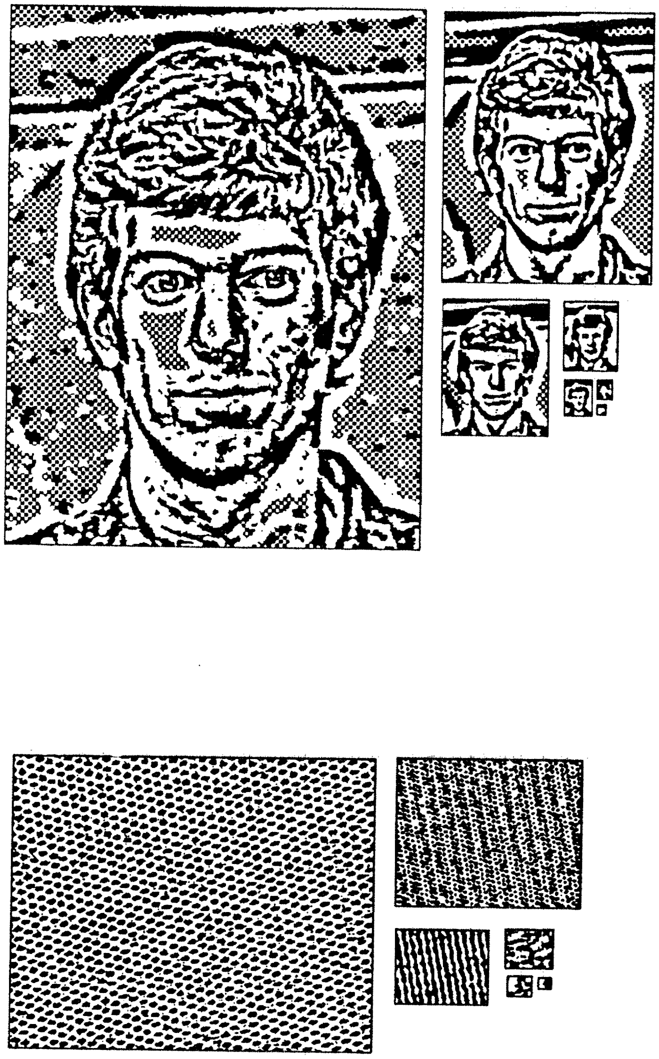


Figure 23. Multi-scale edge finder results for the two images in Figure 22.

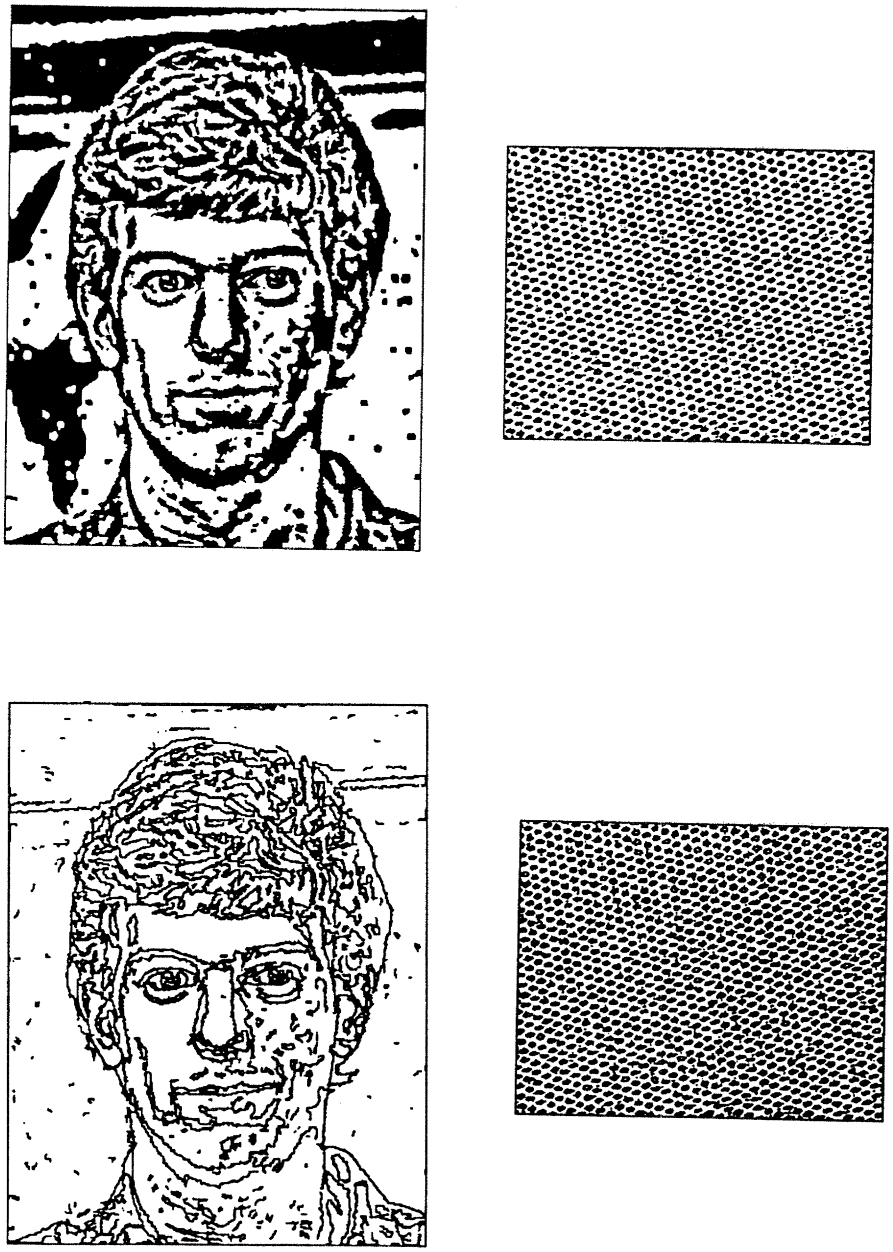


Figure 24. Two ways of displaying multi-scale edge finder output. Top: fine scale results with fill-in from coarser scales. Bottom: fine-scale boundaries.

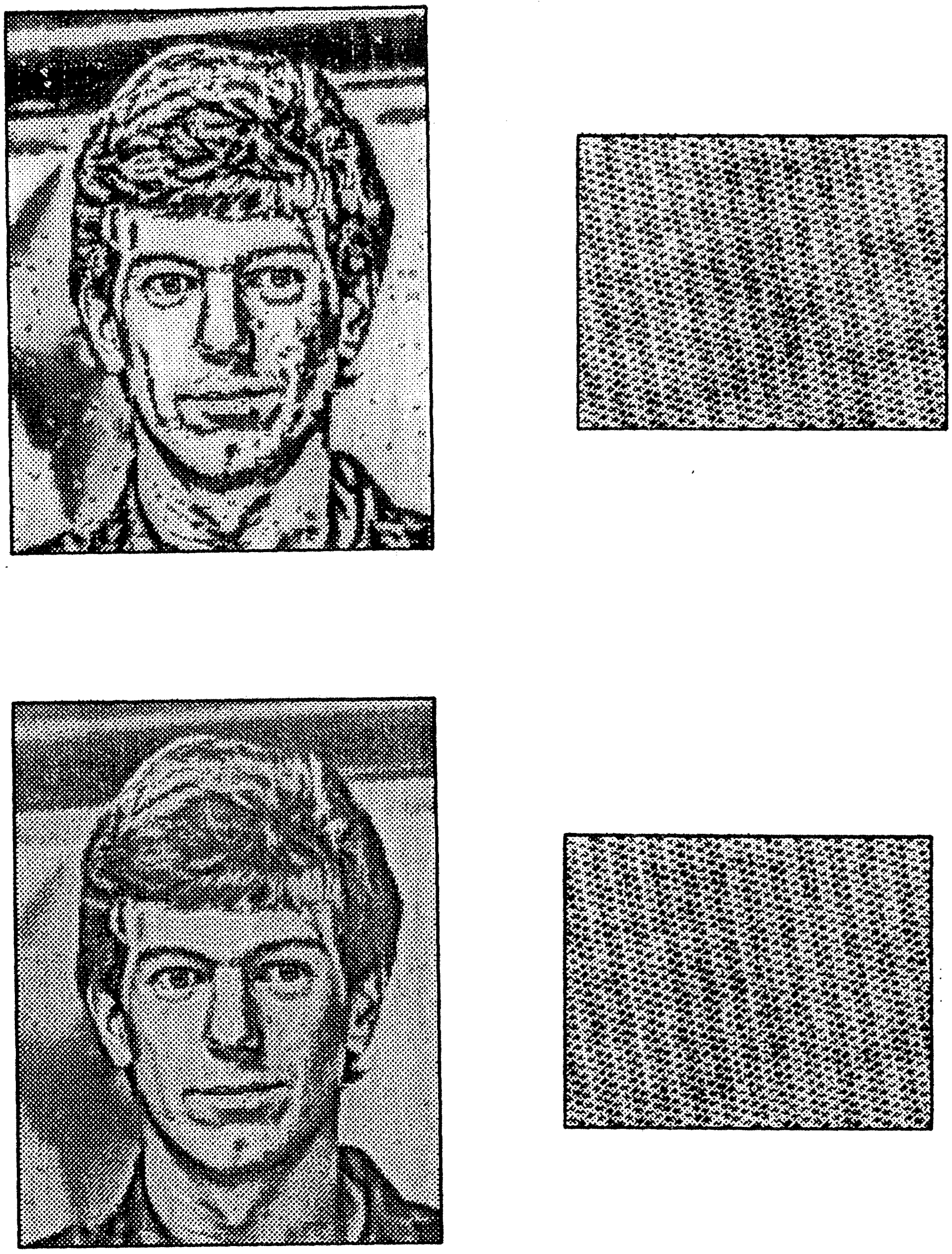


Figure 25. Two ways of displaying multi-scale edge finder output. Top: reconstruction from sign bits. Bottom: reconstruction using magnitude information.

repeated until the finest scale is reached.

The algorithm for combining results at each step must accomplish two things: fill in areas of zero response at the finer scale and average the two results. Whenever the finer scale had no significant edge finder response, the coarse-scale response is used to fill in the fine-scale image. The two images are then averaged to produce the final reconstruction. In this averaging, the coarser-scale image is weighted by the number of coarser scales it represents and the fine-scale image is weighted twice as heavily as each coarser scale. This sharpens boundaries slightly and makes and fine-scale texture more visible.

As you can see from Figure 25, the reconstruction using magnitude information preserves almost all useful information in the image. It differs from the original in two ways. First, information about the overall intensity of the image is lost. Thus, if the image had been lighter or darker overall, the reconstruction would have been the same. Secondly, some slopes in intensity may be lost. This does not affect all of the shading on curved objects, because shading often generates some second difference response, particularly at coarser scales. However, an even gradient across the image, such as might be caused by changes in lighting, might disappear entirely.

The reconstruction using only sign labels clearly loses more information. Relative contrast of regions is no longer visible except in extreme cases and smooth shading is lost. However, when the image has significant changes in structure across scales, this representation conveys much more information about the image than a fine-scale cartoon does. The matching applications described later in this thesis all use multi-scale sign information, without taking magnitude information into account. Thus, this reconstruction conveys a good sense of the information available to these algorithms.

9. More examples of algorithm performance

Figures 26-31 show more examples of Phantom's output on scenes containing both natural and manmade objects. Further examples are presented in Chapters 9 and 10. These images, and those presented earlier in the chapter, were chosen to represent a range of scenes with approximately constant camera noise characteristics, so that the edge finder could be run with a constant noise threshold. For other camera systems, it may be necessary to adjust the noise threshold, smooth the image slightly before running the edge finder, and/or add de-striping algorithms. However, the edge finder has been tested on a large number of images over the past year and a half and the examples presented are typical of its performance.

The examples presented in this thesis were generated by a LISP implementation running on a Symbolics LISP machine. The main liability of this current implementation is that it runs very slowly, 4-7 minutes per 100 by 100 block of image, depending on the image contents. The primary problem is the star-convex sum operation and its speed could be improved in several ways. First, for historical reasons, the current implementation uses a large set of paths in growing star-convex regions. A previous implementation at Oxford used fewer paths without any substantial difference in performance. Secondly, the current implementation was designed for easy experimentation, often sacrificing speed to modularity. Finally, this algorithm is ideally suited to parallel implementation and would speed up greatly on appropriate parallel hardware.

There has been some recent interest (Hildreth 1983, Huertas and Medioni 1986, Young 1986, Nalwa and Binford 1986) in sub-pixel localization of boundaries. I have implemented a simple interpolation algorithm for Phantom's boundaries. This algorithm uses smoothing to interpolate response values and the

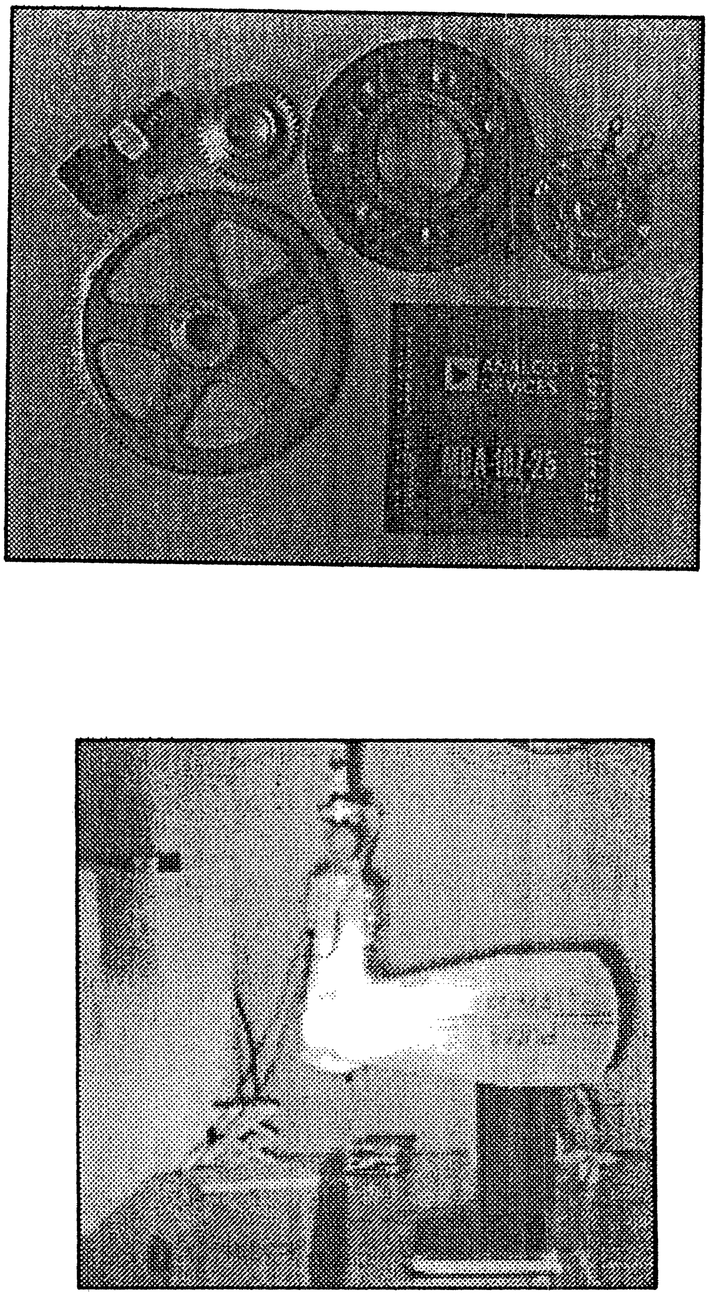


Figure 26. An image of some parts (540 by 425 cells) and an image of a Puma robot (450 by 420 cells).

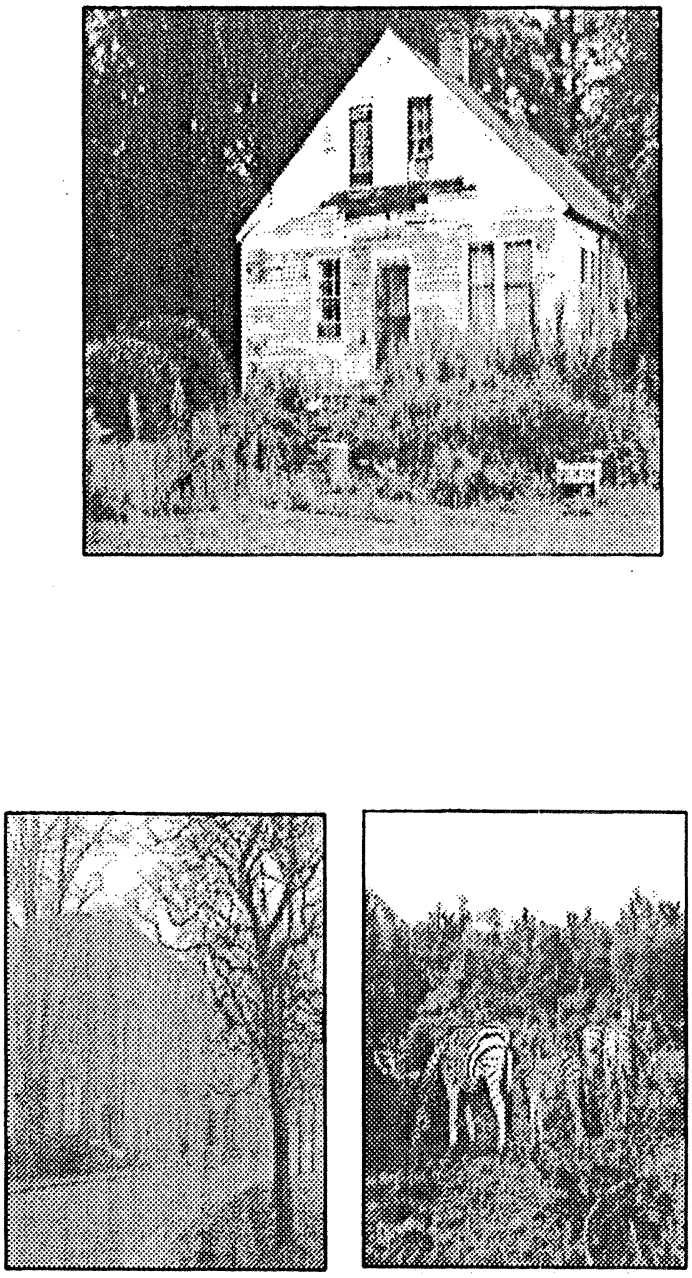


Figure 27. Images of a house (450 by 420 cells), a building (250 by 350 cells), and some zebras (250 by 350).

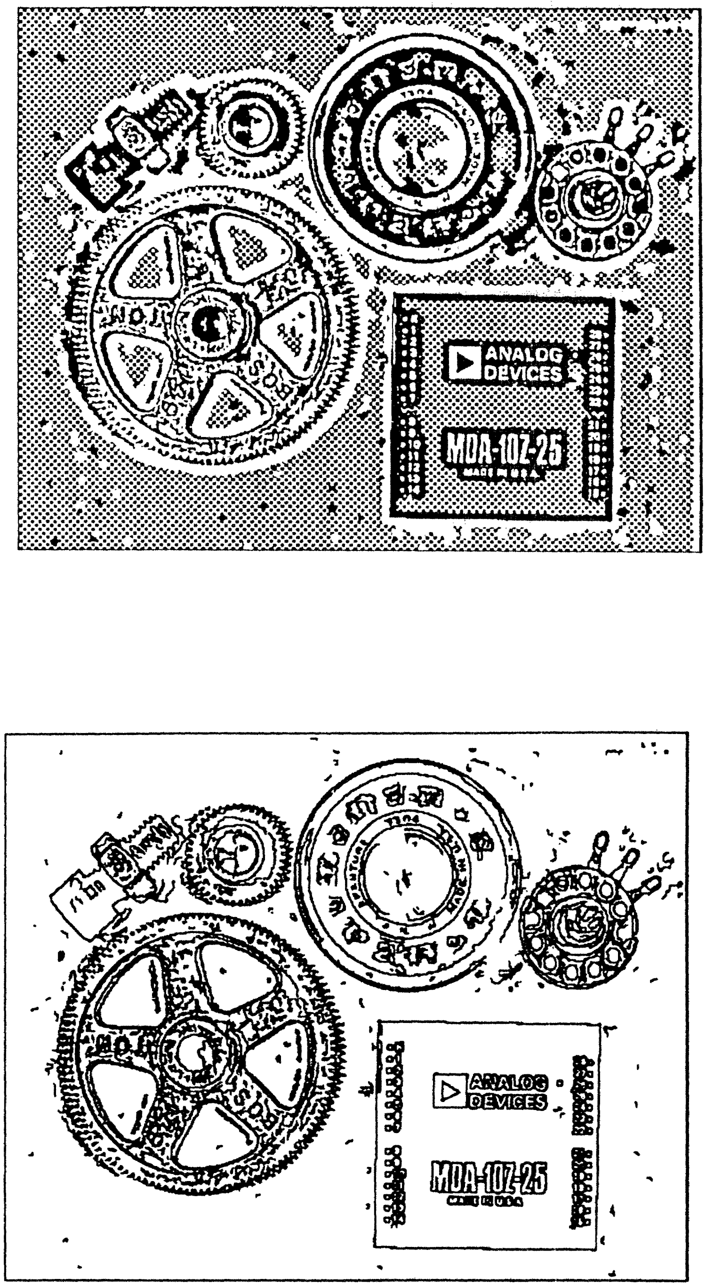


Figure 28. Phantom output on the parts image.

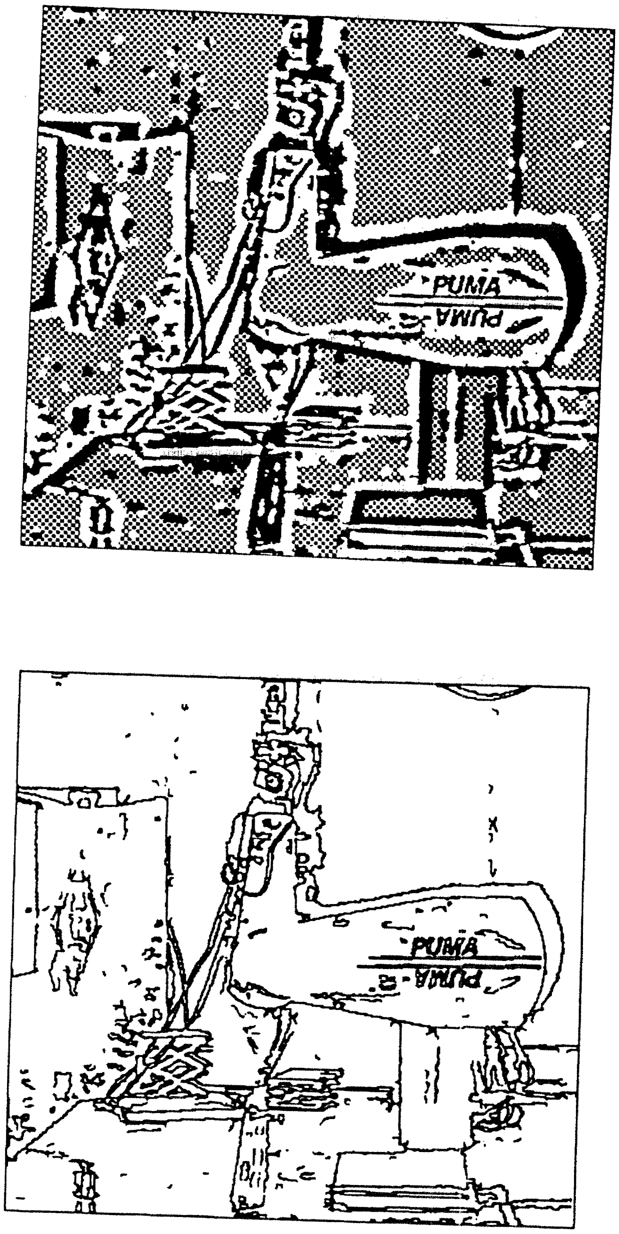


Figure 29. Phantom output on the robot image.

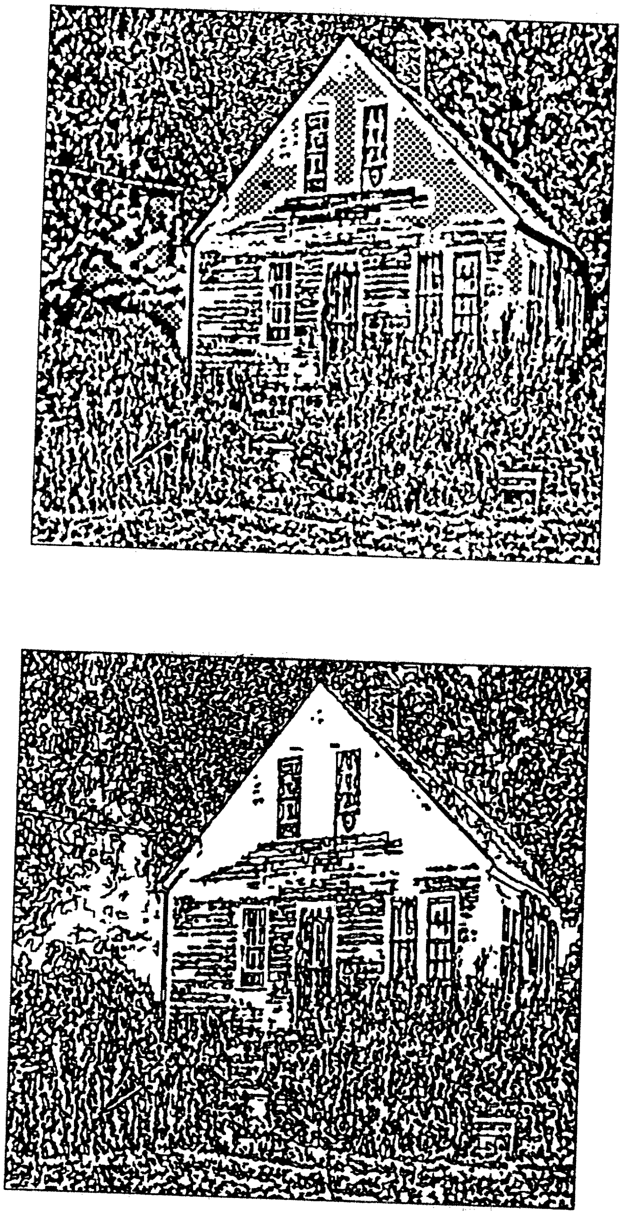


Figure 30. Phantom output on the house image.

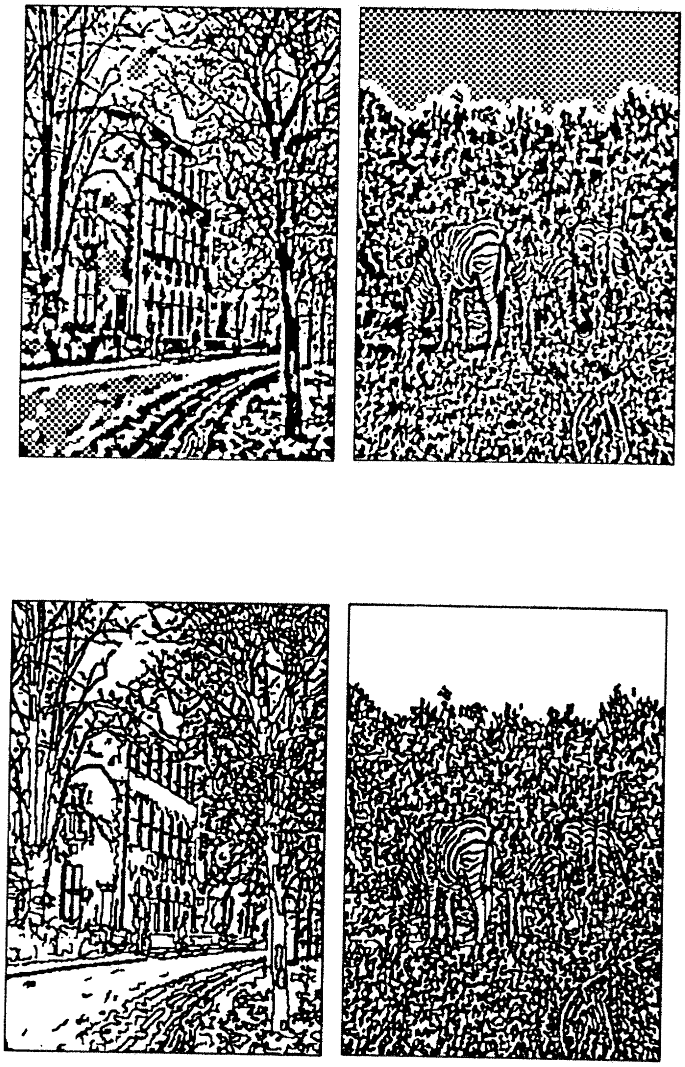


Figure 31. Phantom output on the building and zebra images.

boundary adjustment algorithm described in Chapter 5 to move boundary locations to reflect those interpolated responses. Figure 32 shows an example of its output. Clearly this process can be repeated to enlarge the image to arbitrary size. I have not, however, been able to study how much of this reconstructed precision is stable.[16] Clearly this depends on the contrast of the intensities across the boundary. High-contrast boundaries can probably be localized to high precision, but low-contrast boundaries may be moved 2-3 cells by varying camera noise.

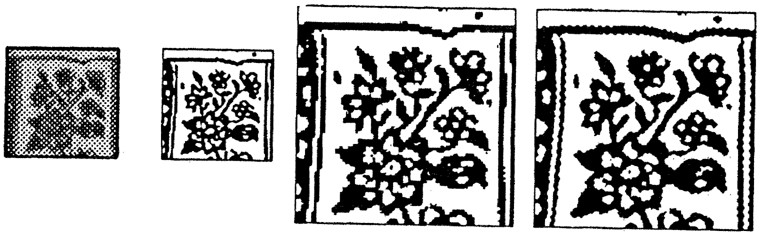


Figure 32. Left to right: an image, edge finder output, expanded version of output (made by repeating each value over a 2 by 2 block), interpolated output.

The boundary adjustment algorithm described in Chapter 5 can also be used for two other operations on edge finder output. First, the Phantom edge finder sometimes produces boundaries that are thicker than topologically necessary, reflecting uncertainty in the boundary locations. The boundary thinning algorithm described in that Chapter can be used to reduce these boundaries to minimal size, if this is desirable for some application. Secondly, the matching algorithm described in Chapter 5 can be used to compare edge finder output from different scales, determining which coarse-scale boundaries represent new features

[16] Stability is the only well-defined criterion for success. Except for the rare special case of perfect step edges, psychophysical judgements of the "correct" location of boundaries only provide this location to within perhaps one or two cells. Although more precise definitions exist, they are *ad hoc.*

and which are simply blurred variants of the fine-scale representation (cf. Canny 1983, 1986). Examples of this are presented in Chapter 5, Section 5.

Finally, boundaries could be detected in properties other than light intensity. Generally accepted theories of visual processing suggest that sharp changes must also be detected in color, depth (as from stereo), motion fields, and surface texture properties (such as periodicity). It seems likely that techniques developed for intensity edge finders could be adapted to these other applications. For example, Figure 33 shows boundaries detected by the Phantom edge finder (noise threshold 240) in a map of stereo disparities.

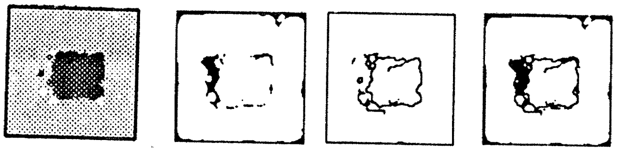


Figure 33. Detecting boundaries in a stereo depth map. Left to right: stereo disparities, match map, boundaries, boundaries and non-matching regions. The match map shows which cells have been assigned a stereo match (in white) and which have not been assigned a stereo match (in black). Cells not assigned a match may represent either errors in matching or surfaces visible to only one eye.

These other types of properties are, however, somewhat more difficult to handle than grey-scale intensities. The space of values for some properties is more complicated. For example, texture orientation may require a circular space of values and color a spherical one. Stereo depth data is only a partial function, because no depths are available for occluded regions of the image. Finally, edge finding algorithms in these other domains must operate on the results of analysis algorithms that are, themselves, still experimental. Because of these factors, there has been no systematic study of how to extend edge finder algorithms to these other types of properties.

10. Comparison to previous algorithms

The Phantom edge finder differs from previous edge finders in two ways. First, it uses a more flexible model of boundaries than previous algorithms. As we will see in Chapter 9, this enables it to perform reliably on sharp corners, region intersections, and dense texture. These types of features cause problems for previous edge finders. Secondly, it uses a more reliable method of distinguishing real responses from those due to camera noise. In this section, I survey previous algorithms for edge finding and discuss how they differ from the method used by Phantom.

There have been three recent approaches to edge finding: boundary modelling, surface modelling, and edge operator.[17] In the boundary modelling approach, used by Sher (1987), Hoff and Ahuja (1987) (stereo depth data), Hueckel (1971, 1973), and Nalwa and Binford (1986), models are developed for all desired boundary shapes. These models are then fit to patches of the image. Statistical considerations are used to determine how good a fit is required in order to hypothesize a boundary, given an estimate of the camera noise. The problem with this approach is developing a sufficiently flexible set of models for boundaries. Models have typically limited to isolated, straight boundaries and with a small variety of intensity profiles across the boundary. These algorithms perform poorly at region intersections, at sharp corners, and in dense texture, where none of the set of models is a good fit to the image. The proposal of Leclerc (1985) is

[17] For earlier approaches to the problem, see the surveys in Davis (1975), Pratt (1978), Ballard and Brown (1982), adding also the algorithms describe in Binford (1981) and Persoon (1976).

more general, but has not been tested on real images.

In the surface modelling approach, represented by Haralick (1980, 1984) Haralick, Watson, and Laffey (1983), and Parvin and Medioni (1987), the image intensities in each patch of the image are modelled. The model for each surface patch is then analyzed to detect the presence of boundaries, e.g. by looking for zero-crossings of the second-differences of the model. The weakness in this approach is, again, the set of models. Surface models in current use can only provide good approximations for patches of image in which the intensities vary smoothly or in which there are only restricted types of boundaries (typically, again, isolated straight step edges). Thus, these approaches also fail on intersections, sharp corners, and dense texture. Brooks (1978) discusses how some earlier edge operators can be viewed in terms of surface modelling. The segmentation algorithm of Besl and Jain (1988), the regularization proposal of Torre and Poggio (1986), and the corner detector of Noble (1987) represent similar approaches to image description.

In the edge operator approach, some operation (such as taking second differences) is applied to the image to yield a map of "edge responses." Some test is then applied to distinguish significant responses from those due to camera noise and boundaries are hypothesized to account for significant boundaries. The Phantom edge finder falls into this class of algorithms. Other recent examples include Marr and Hildreth (1980), Hildreth (1983) Canny (1983, 1986), Pearson and Robinson (1985), Grimson and Pavlidis (1985) (stereo depth data), Watt and Morgan (1985), Huertas and Medioni (1986), Young (1986), Gennert (1986), Boie, Cox and Rehak (1986), Deriche (1987), Spacek (1985), Argyle (1971), Macleod (1972), Nevatia and Babu (1980), Huttenlocher (1988) (curvature data), and Lee, Pavlidis, and Huang (1988). These algorithms can be described in terms of two independent problems: what operator to use and how to distinguish real responses from noise.

Quite a variety of operator shapes have been proposed, most of them close variants of one another. Consider the 1D case first. There are two basic shapes of edge operators: first difference and second difference.[18] Boundaries are hypothesized at maxima of first difference responses and at zero-crossings (and occasionally isolated maxima) of second difference responses. On a perfect step edge, the two types of operators behave similarly. First difference operators have the problem of producing spurious responses on ramps, formed by blurred boundaries and smooth shading. They are also unable to detect isolated maxima of the second difference, known as creases or roof edges. Second difference operators, on the other hand, produce spurious boundaries in staircase patterns. These problem behaviors are shown in Chapter 9, Section 6.

Many of the edge finders listed above use second difference operators, as the Phantom edge finder does. Those using first difference, or similar, operators include Canny (1983, 1986), Argyle (1971), Macleod (1972), Spacek (1985), Deriche (1987), Nevatia and Babu (1980), and Gennert (1986). Gabor filters (cf. Young 1986) and Difference of Gaussian operators (Marr and Hildreth 1980) are similar in shape to the second difference. Residual operators (Grimson and Pavlidis 1985, Lee, Pavlidis, and Huang 1988, Huang, Lee, and Pavlidis 1987) also seem similar in shape to second differences. The details, however, depend on the type of approximation used and have not been explored in detail. Boie, Cox, and Rehak (1986) use a combination of first and second difference type operators.

There are three methods of extending these operators to 2D: directional,

[18] Either type may, of course, be combined with smoothing. See below.

oriented, and isotropic, shown in Figure 34. In the directional method, the 1D operator is applied along straight paths through the 2D image. This is the method used by the Phantom edge finder. Oriented operators are formed by taking directional responses from a set of parallel paths and averaging them. This favors extended straight boundaries. Isotropic operators are created by averaging responses from directional differences taken about a common point, but in different directions. Isotropic and oriented operators both distort the shape of boundaries that are not straight.

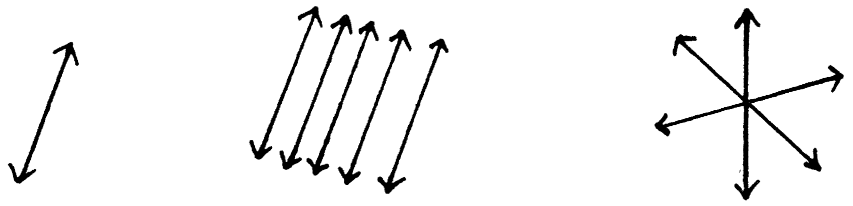


Figure 34. Left to right: directional, oriented, and isotropic methods of taking differences.

When directional or oriented edge operators are used, the results from different directions must be combined. The Phantom edge finder is unique in having a robust method for combining directional responses. Nevatia and Babu (1980) use a similar method for combining oriented first difference operator responses, but it is unclear that their later thinning and linking algorithms are robust. Canny (1983, 1986) assumes that the directional first differences approximate a linear transformation and summarizes them into a gradient direction and magnitude on this basis. As we saw in Section 3, this assumption is not valid near sharp corners and intersections, at which Canny's edge finder performs poorly. Gennert (1986) accepts a directional response at a cell only if it is an extremum over all directions and larger than the response in the perpendicular direction. This has not been thoroughly tested, but it also seems liable to make errors at corners and intersections.

The other main variation in edge operator algorithms is in how they eliminate the effects of camera noise.[19] The most popular method of eliminating noise is to smooth the image before applying the operator and then remove responses with low amplitude. The problem with this technique is that smoothing reduces the resolution of the edge finder output. Because of this, recent work has attempted to reduce the amount of smoothing required by better methods of distinguishing real responses from noise in the output of the edge operator. Methods using edge linking (Nevatia and Babu 1980, Persoon 1976) have been proposed, but it is unclear how well they work.

Matching representations from different scales is occasionally suggested as a method of identifying spurious edge finder responses (Marr and Hildreth 1980, Hildreth 1983, Schunck 1987, Bergholm 1987). Other researchers have suggested evaluating responses based on a sum or product of responses from different scales (Watt and Morgan 1985, Rosenfeld 1970, Schunck 1987). While preservation over multiple scales or occurrence at a sufficiently coarse scale may be useful as a measure of the importance of a boundary, neither criterion seems helpful in identifying spurious boundaries. First, many legitimate features in images appear only at the finest scale, because they are simply too small to be detected at any other scale. Secondly, in images with qualitatively different representations at different scales, such as the cleaning cloth image discussed in Section 8 and

[19] Pearson and Robinson (1985) seem to achieve good results with only minor amounts of noise suppression. However, since my re-implementation of their algorithm is sensitive to camera noise, their low-resolution images may have been produced by some type of sub-sampling. Since camera noise is primarily high-frequency, sub-sampled images contain far less noise.

Chapter 5, Section 5, many legitimate features last only one or two scales.

Two methods for distinguishing real response from camera noise have recently been proposed, both using image topology in addition to response amplitudes. Blake (1983) and Geman and Geman (1984) use iterative procedures to assemble responses into extended boundaries. Although interesting, these techniques have not yet been developed into robust algorithms. Furthermore, they make excessively strong assumptions about the form of boundaries (see Section 7). As discussed in Section 4, algorithms similar to Phantom's have also been proposed by Watt and Morgan (1985), Huertas and Medioni (1986), and Huttenlocher (1988) (curvature data). However, these researchers discuss only the 1D case and, thus far, Phantom's algorithm is the only robust 2D version of this idea. Lee, Pavlidis, and Huang (1988; also Huang, Lee and Pavlidis 1987) propose another 2D version, but the details are unclear and it has not been extensively tested.

11. Conclusions

In this chapter, we have seen how boundaries can be detected robustly in digitized camera images. More detailed evaluation of its output and a detailed comparison to Canny's (1983, 1986) edge finder is provided in Chapter 9. The new algorithm produces boundaries at higher resolution than previous algorithms without sensitivity to camera noise. It also performs more reliably on sharp corners, regions intersections, and dense texture. If later algorithms use topological properties based on these boundaries, as this thesis claims, this ability to detect stable boundary locations from real sensory input is extremely important.

Use of topological structure is also important in the edge finder algorithm itself. First, connectedness, in the form of star-convexity, is used in assessing response strength. This constraint prevents evaluation of one response region from being corrupted by nearby responses. The resulting evaluations are able to distinguish real responses from camera noise more robustly than previous proposed methods. Connectedness was also used in the algorithm for deciding which response regions were associated with zero-crossings and which were not.

Finally, intensities are an important example of a digitized function. We have seen two examples of how the digitization interacts with the process of finding boundaries. First, we saw that edge finder results can be produced at a range of scales, by changing the digitization. Secondly, we saw that the set of directional differences about a cell may not approximate a linear transformation, unlike the directional derivatives about a point. Although this means that techniques from calculus cannot be used directly on digitized functions, we saw that patterns of finite differences can still be analyzed, by looking at maximum amplitude responses.

Chapter 5: Image matching

1. Introduction

As we saw in Chapter 3, both stereo analysis and edge finder evaluation require an algorithm for matching two edge finder outputs. For each image, the edge finder specifies both a labelling of cells in the image as dark, light, zero, or boundary and a set of boundaries induced by this labelling. The matching algorithm should preserve both the topological structure of the images and the dark/light labels. In this chapter, we see how this matching is done for a fixed alignment of the two images. In Chapter 6, I show how a stereo analysis algorithm can be built using this matcher and, in Chapter 9, I show how the matcher can be used in edge finder evaluation. Examples illustrating potential uses in other domains, such as texture analysis, are also discussed briefly in these chapters.

As we saw in Chapter 3, matching images is divided into three phases: adjustment, computation of match strength, and analysis of boundary motion. This decomposition of the matching problem allows two difficult problems to be tackled separately. Consider the situation shown in Figure 1. If we decide to adjust boundary A to match boundary B, boundary A must be moved through the shaded region and cells in this region must have their labels altered. However, there are many ways that individual points in A could be paired with individual points in B. The adjustment phase of matching builds matches between extended sections of boundaries, making arbitrary decisions about the point-wise pairing. The analysis phase then solves the *aperture problem*, i.e. it determines which point-wise pairing is appropriate. This can be done by analyzing the shape of the adjustment region, without considering the details of how the adjustment was done.

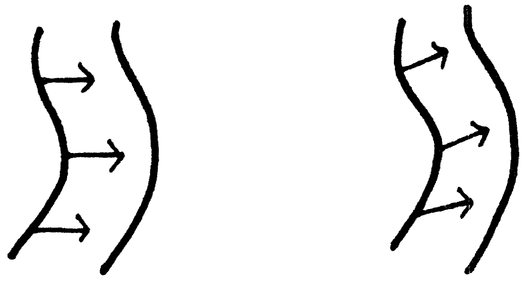


Figure 1. Two ways of adjusting the same boundary.

The two halves of the matching algorithm have very different requirements. Adjustment must consider the detailed topological structure, in order to decide how boundaries can be moved. This is made tractable by the fact that each adjustment operation considers only a small section of the image. On the other hand, solving the aperture problem, which the analysis phase must do, requires examining a large enough area of the image to extract a reliable direction of motion. Such a support region would not be tractable if topological detail had to be considered at the same time.

The matcher illustrates two important uses of image topology. First, sections of image can only be matched if they have the same topological structure. Using such a requirement for practical applications such as stereo analysis or edge finder testing is a direct test of the main claim of this thesis, that topological structure is useful. Enforcing this requirement during boundary adjustment also provides a good example of how to use the mathematical machinery developed in Chapter 11. Finally, algorithms in the analysis phase of matching use the same star-convex sum operation that the edge finder used, but for different purposes. Thus, they illustrate some of the variety of applications for which connectivity requirements are useful.

Section 2 presents the basic operations used to adjust boundary locations. Section 3 discusses how they are used to build an adjustment algorithm. Section 4 explains the computation of matching strength and Section 5 presents details of how boundary motion is estimated. Sections 6 and 7 review previous proposed matching algorithms.

2. How to adjust boundaries

The key to understanding the boundary adjustment algorithm is that the details of the correspondence between the two images are going to be thrown away before the analysis phase. Boundary adjustment operations must guarantee that:

- regions through which boundaries are moved consist of exactly those cells whose labels are altered during adjustment, and
- there exists a correspondence between the original and the adjusted image that preserves topological structure.

However, so long as both of these conditions can be guaranteed, *the adjustment algorithm need not reconstruct the correspondence explicitly.* This is very useful, because cell labellings are easy to handle explicitly in a computer program and correspondences are not.

Since we only care about the existence of a correspondence, not the correspondence itself, development of adjustment algorithms involves discussion of when two images are *homeomorphic*, i.e. have the same topological structure. Chapter 11 develops three techniques for showing that the spaces represented by two cell structures are homeomorphic.[1] Recall the adjacency and incidence structures discussed in Chapter 2. The first technique for proving spaces homeomorphic says that if the cells in two cell structures can be paired so that the adjacency/incidence structure and the boundary markings are preserved, then the spaces represented by these cell structures are homeomorphic. Thus, for judging homeomorphism, we only need to pay attention to the incidence or adjacency structure and the boundary markings. I call this technique *redrawing*, because it implies that we are free to redraw a cell complex with cells of different shapes and positions, without altering its topological structure. This is very convenient, because it means that proofs can be written using pictures of cell complexes, rather than detailed analytic descriptions of the underlying spaces.

The other two techniques are not so trivial. The second technique, called *subdivision* says that a cell can be split into two cells sharing a common (non-boundary) edge, without changing the topological structure of the underlying space. This is illustrated in Figure 2. This technique alone can be used to relate two images if they contain no boundaries. Suppose that the initial alignment between the two images was not bijective, because a cell in image X was associated with more than one cell in image Y. We can split the cell into X as many times as it takes to create exactly one cell corresponding to each of the cells in Y. Similarly, if the initial alignment is bijective, but does not preserve adjacency/incidence structure, it can be made bijective by subdividing cells in both images. In the applications presented in this thesis, alignments are always integer translations of rectangular arrays, so they always preserve topological structure. However, in more general applications, it may be necessary to do this

[1] For technical details of these operations, see Chapter 11, Sections 5-6. The following discussion is consistent with these technical details, but does not presuppose that the reader is familiar with them.

type of subdivision in order to create a correspondence that preserves the cell structure.

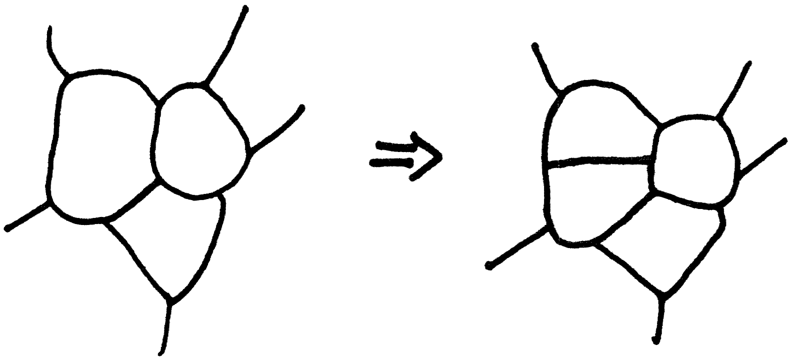


Figure 2. Subdivision of a cell in a cell complex.

In most applications, a small number of alignments can be pre-compiled and used repeatedly. For example, humans are only able to fuse a limited range of stereo disparities without eye movement. The alignments in this range, known as *Panum's area* might be pre-compiled. Thus, we can assume that the two images X and Y have been subdivided in advance and that the initial alignment preserves cell structure. What boundary adjustment must do is make the two images have not only the same cell structure but also the same boundary labelling. Where this can be achieved, redrawing implies that the two images must represent homeomorphic spaces. Thus, we have converted a problem of proving two images homeomorphic into one of moving boundaries in one image without changing its topological structure.

In order to develop operations for moving boundaries, we need a third technique for proving homeomorphism, called *boundary thickening.* This technique allows a vertex or an edge that is marked as a boundary to be replaced by a whole boundary cell. Remember that in either the closed-edge or open-edge model of boundaries, points in boundary cells are deleted from space. Thus, the cell complexes before and after thickening have underlying spaces that look exactly the same, as shown in Figure 3. More precisely, they might have different points or different shapes, but they must have the same topological structure. The formal details of this operation are slightly difficult and are given in Chapter 11. However, a pictorial understanding of boundary thickening is sufficient for reading the rest of this chapter.

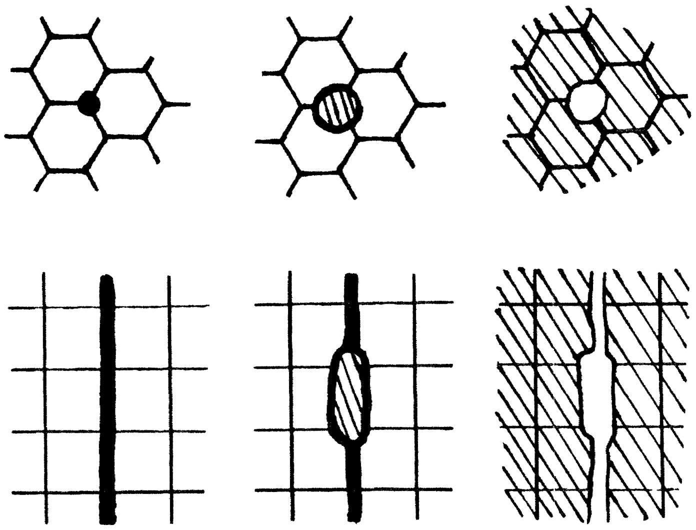


Figure 3. Thickening a boundary in a cell complex. The top pictures show a vertex being thickened. The bottom pictures show an edge being thickened. From left to right: before thickening, after thickening, and closed-edge model of underlying space.

Figure 4 shows the final boundary adjustment operations used in the matcher. These operations relate a cell structure in which some cell x is a boundary cell to a similar cell structure in which x is not a boundary cell. The patterns described by these operations can be applied in rotated or reflected form. To avoid explicitly testing these possibilities, they are compacted into one *boundary* *test* in the current implementation. This test is described in Appendix B.

Each of the four adjustment operations specifies a topological equivalence between two cell structures. Thus, each operation can be applied in either direction. In one direction, the operation thickens a boundary and, in the other direction, it thins a boundary. In either case, each operation changes the boundary marking of only one cell.[2] Because each operation makes such a small change to the cell structure, it is not difficult to prove that it preserves the topological structure of the underlying space. However, larger adjustments can be produced by repeated application of the operations.

Using the three techniques given above—redrawing, subdivision, and boundary thickening—we can develop simple proofs that the boundary adjustment operations preserve the topological structure. Each proof is a sequence of local cell structures, starting with the input to the operation and ending with its output, in which consecutive structures can be related via one of the three basic operations. These proofs are given in Figures 5-8. Because each of the basic operations preserves the topological structure, so must their composition.

This set of adjustment operations cannot relate an arbitrary pair of representations with the same topological structure. There are three limitations that seem to hold, though I do not have a formal characterization of them, still less any proof that they are a full description of the limitations. First, the operations cannot relate an infinite cell complex to a finite one. For example, a region consisting of the real number line and a single-cell region that is a subset of the real line are homeomorphic in the open-edge model of boundaries. However, these

[2] It is tempting to confuse the effect of these operations with that of boundary thickening. Boundary thickening adds a new cell in the middle of a boundary, whereas the adjustment operations re-label an existing non-boundary cell as a boundary cell.

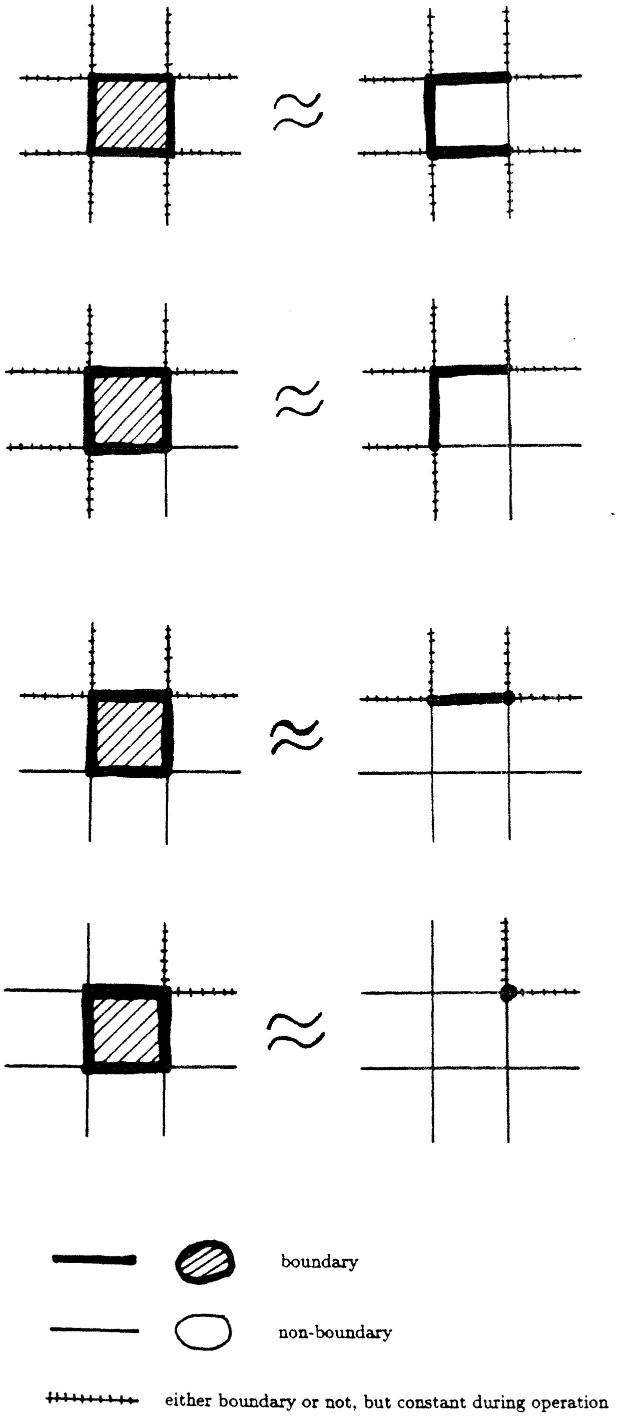


Figure 4. The four boundary adjustment operations.

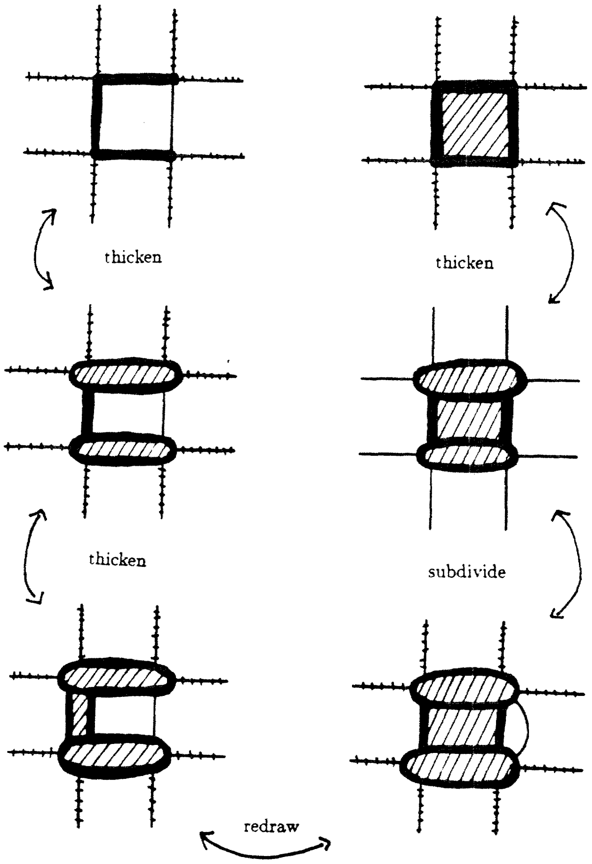


Figure 5. A proof that one of the adjustment operations preserves the topological structure of the image.

two regions cannot be related via any finite sequence of operations. Among other reasons, the same operations work for the closed-edge model, in which these two sets are not homeomorphic.

Secondly, I do not believe that the adjustment operations can relate arbitrary mirror-reversed representations, even finite ones. Consider two scenes containing handed objects, such as granny knots. If the matcher is given an alignment of the two scenes in which one knot is lefthanded and the other knot is righthanded, I do not believe it can successfully match the two knots. Such a match would

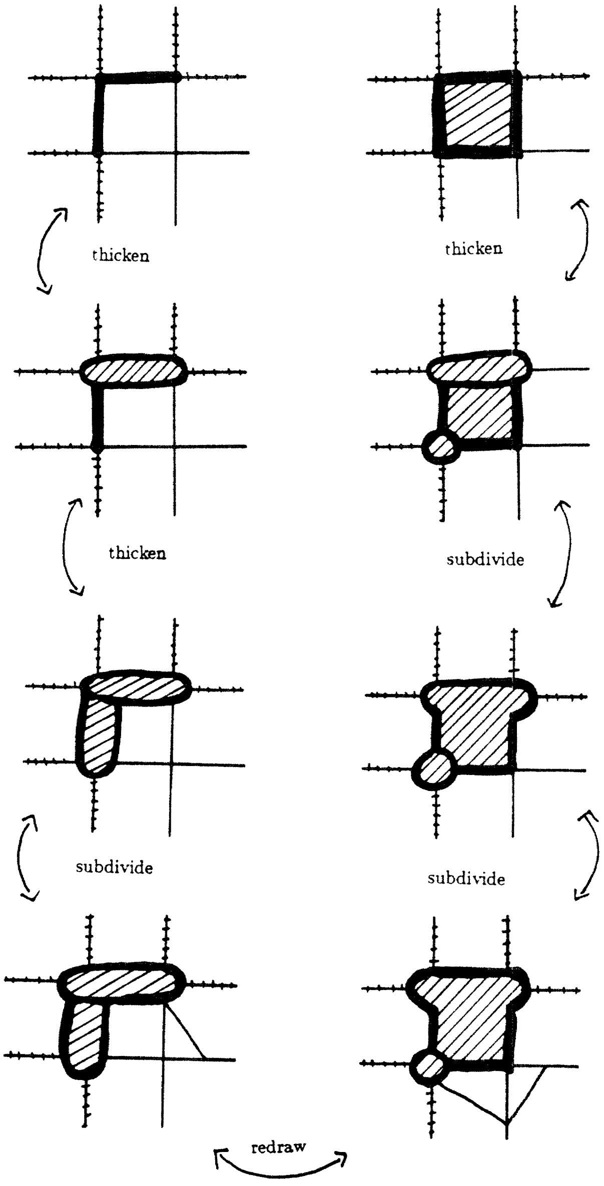


Figure 6. The second operation.

require changing the orientations of cells in the region representing one knot, relative to that of cells in the other knot. I do not believe that the current set of adjustment operations can do this.

Finally, the adjustment operations cannot change inclusion relationships. That is, they cannot remove one region from inside another region, as illustrated

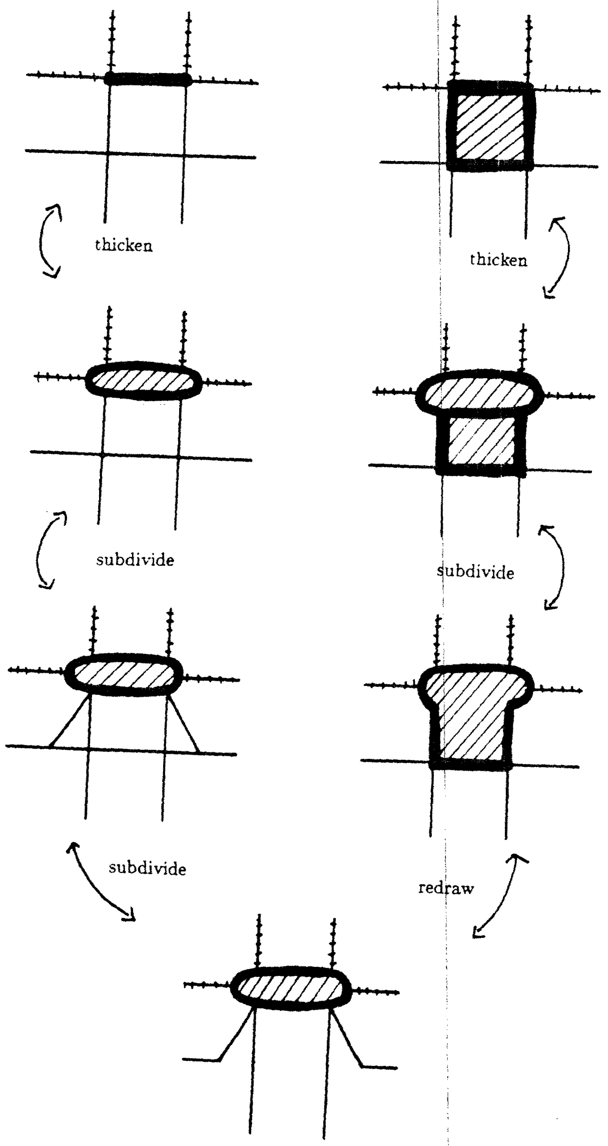


Figure 7. The third operation.

in Figure 9. This limitation seems to match intuitive judgements about what changes are structurally important. Despite these three types of limitations, the boundary adjustment operations are sufficient for matching images robustly, as we will see in Section 3.

As a conclusion to this section, I should emphasize one point about these

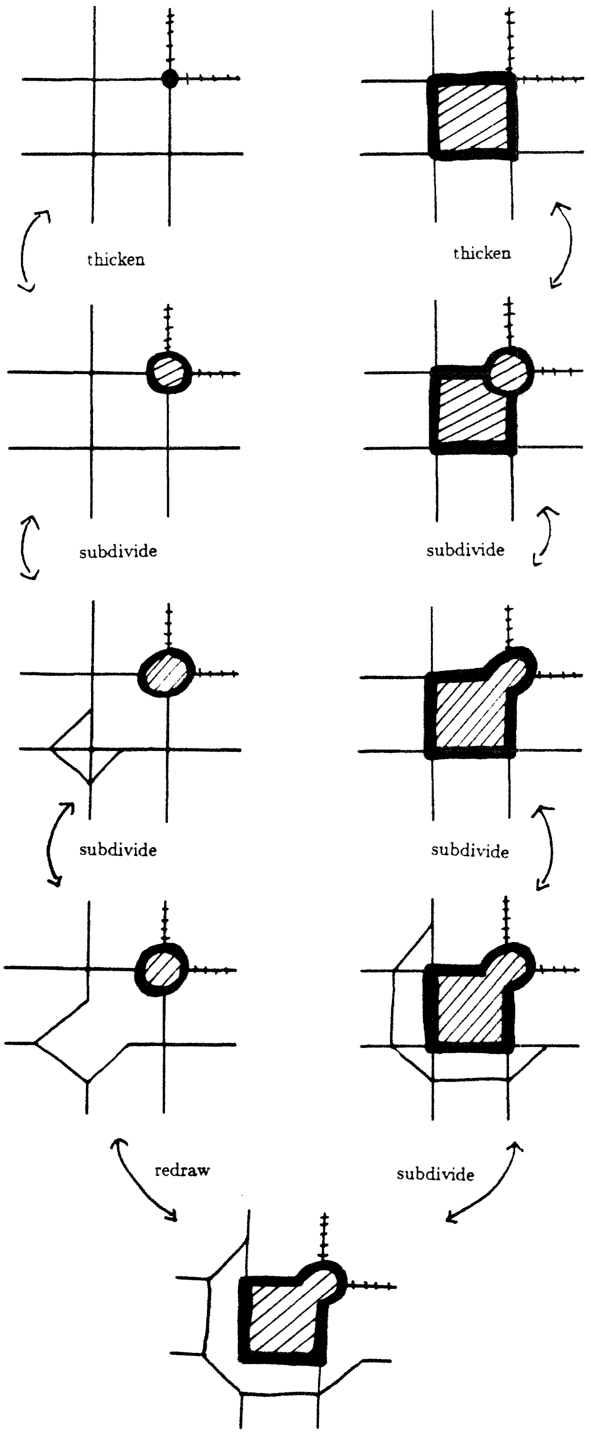


Figure 8. The fourth operation.

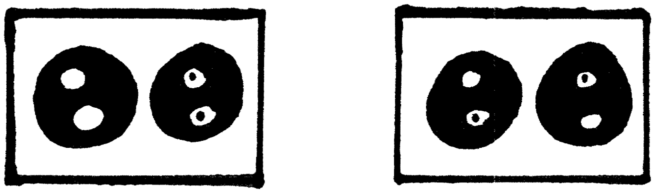


Figure 9. The adjustment operations cannot change inclusion relationships. Thus, they cannot successfully match the image on the left to the image on the right, although they are homeomorphic. Intuitively, these images have different structure.

adjustment operations. The operations are specified in terms of changes to the combinatorial cell configurations. The correspondence whose existence is guaranteed, however, relates the underlying, infinite-resolution spaces represented by these complexes. Thus, when I say that the matcher preserves topological structure, I mean that in the usual mathematical sense, not in some sense peculiar to digitized spaces. It is typical in computer vision algorithms to use approximations to mathematical concepts, e.g. smoothness or differentiability. Although there may be noise in the boundaries that are input to the matcher, the transformations performed by the adjustment phase of the matcher are mathematically exact.

3. Using adjustment operations

This section explains how boundary adjustment operations are used by the topological image matcher. In image matching, cell labels must be adjusted as boundary locations are changed. Furthermore, unrestricted application of the adjustment operations could scramble the contents of an image in undesirable ways. The actual matching algorithm restricts the application of these operations so as to allow only minor adjustments to region shapes.

Requiring two images to have the same topological structure, using the model of boundaries developed in Chapters 2 and 11, is a very weak condition on the images. It does not, for example, constrain the order of regions to be the same, as shown in Figure 10. The four image adjustment operations cannot be used to relate any pair of images that have the same topological structure, as we saw in Section 2. However, they can scamble patterns of 2D regions in ways that are not desirable in image matching. The image matcher applies the operations only in limited ways, so as to make only small adjustments to the images.

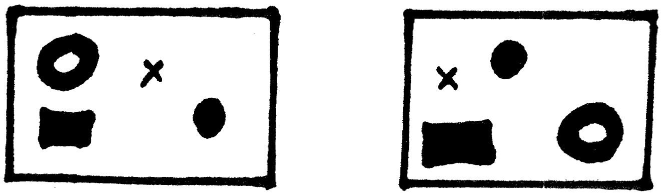


Figure 10. These two images have the same topological structure.

Boundary adjustment is applied to an image in two phases. The input to adjustment is a pair of images, one of which is to be modified so as to match the other (*target image*) as well as possible. The first phase, *thickening*, identifies all cells whose labels are not the same in the two images and moves as many of these cells as possible into the boundaries. The second phase, *thinning*, then moves as many cells as possible out of the boundaries. A cell is moved out of the boundaries only if it can be re-assigned the label of the corresponding cell in the target image. As Figure 11 illustrates, this process of thickening boundaries and then thinning them has the effect of moving boundary locations. The details of this process are described in Appendix B.

This pattern of applying adjustment operations restricts the ways in which

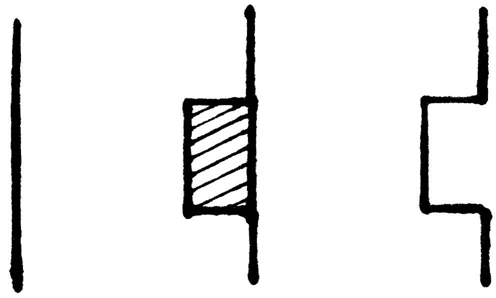


Figure 11. A boundary location can be moved by thickening the boundary with cells from one side and then moving these cells out the other side.

boundaries can be moved. Boundaries are only moved through regions in which labels conflict in the original images. Cells whose labels agree in the original images are not altered. This means that two regions can only be matched if they overlap in the original alignment. Furthermore, a boundary can only be matched to one of the boundaries nearest to it in the original alignment and it cannot "hop over" any intervening boundaries.

Both the thinning and the thickening phase involve multiple passes through the image. Since the adjustment operations are local, they can be done at many image locations in parallel. However, each pass can only thicken or thin each boundary by one cell. Since most applications involve larger boundary motions, multiple passes are needed. In the current implementation, three passes are used in each phase, so each boundary can be moved approximately three cells in any direction.[3] This amount of motion seems sufficient for all of the applications I have considered, though it could be increased without great consequence.

Limiting the number of adjustment passes restricts changes in region shape

[3] Due to details of the algorithms, described in Appendix B, slightly more movement may be possible in some cases. The actual bound varies between 3 and 6 cells, depending on the details of image geometry.

to those that are plausible for the current application. More generally, the minimum number of operations required to transform one image into another can be used as a measure of how different two topologically equivalent representations are.[4] This distance function measures, roughly, the amount of work required to determine that the two representations are equivalent. The algorithms described in this thesis can only prove two representations topologically equivalent when this requires very little work, that is when the representations are also very similar in metric and cell structure. As Figure 12 illustrates, it is difficult for people to determine whether two situations are topologically equivalent if their metric structure is very different. I doubt that the general problem of proving topological equivalence for cellular representations is computationally tractable.

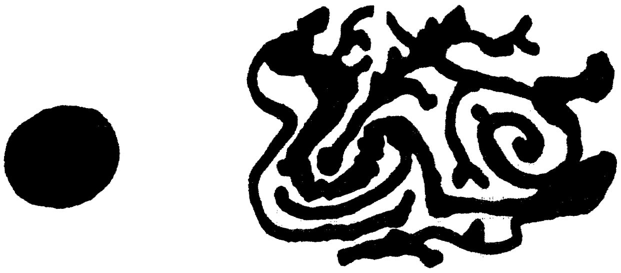


Figure 12. If the metric structure of two situations is very different, it is difficult to determine whether they have the same topological structure.

After both phases of adjustment are finished, the adjusted image is compared to the target image. A cell is marked as matching if it has the same label in the adjusted and target images and as non-matching otherwise. Because boundaries in edge finder output are induced by label transitions, all boundary mis-matches must involve label conflicts. Thus, it is not necessary to flag boundary mis-

[4] The details of this distance function depend, of course, on the details of the operations provided.

matches explicitly. Figure 13 shows match results for two images used in edge finder testing (see Chapter 9). These images represent the same scene, but have different samplings of random noise. The match results correctly identify which regions of the images have been corrupted by the noise.

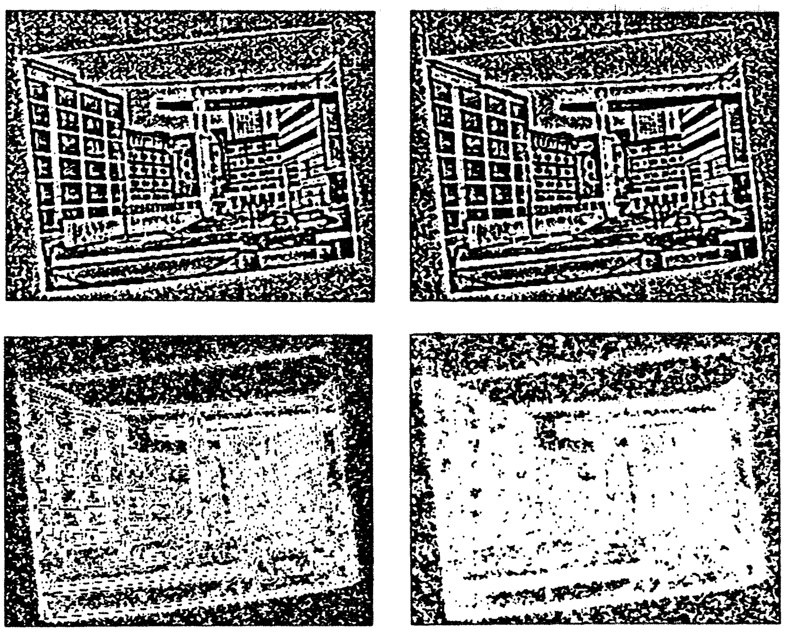


Figure 13. Top: Noisy edge finder output for two images used in edge finder testing. These images reflect the same scene, but with different samplings of random noise. Bottom: the match between the two images before (left) and after (right)adjustment . Matching cells are shown in white and non-matching cells in black.

Cells that match after adjustment are further classified into those whose label was changed during adjustment and those whose label was not altered. This is done by comparing the adjusted image to the original image from which it was derived. This information is used in the analysis phase to determine the amount of boundary motion. Thus, the output of adjustment is a three-way classification of cells into matching, adjusted, and non-matching. I refer to this as the *raw* *match map.*

The adjustment process described above does not treat the two images symmetrically. When the images contain matching boundaries, the two outputs from the two directions differ primarily in that the final boundaries lie to opposite sides of the adjustment regions. However, if a boundary in one image does not correspond to any boundary in the other image, the two outputs differ more substantially. Consider two images, one blank and the other containing a dot, as in Figure 14. When the image containing the dot is adjusted, the mismatch can be reduced to a single point. When the other image is adjusted, however, the mismatch covers the full area of the dot, because no adjustment is possible. In order to handle such cases properly, the matcher does adjustment in both directions, in parallel. The two raw match maps are then reconciled by re-classifying a cell as non-matching in one image if it is non-matching in the other. In cases such as the missing dot, this combined match map contains a non-matching region covering the entire area of the dot.

4. Computing match strength

As we saw in Section 3, the adjustment phase produces the *raw match map*, indicating which cells match after adjustment and which cells had their labels changed during adjustment. The second phase of matching assigns strengths to the match at each cell. We see in this section that these strengths can be used to remove those matches that cannot be distinguished from random noise, yielding a more meaningful *clean match map.* Section 5 then discusses how the analysis algorithms extract information about boundary motion from the clean match map.

Consider the image match shown in Figure 13. As we saw in Chapter 3, the

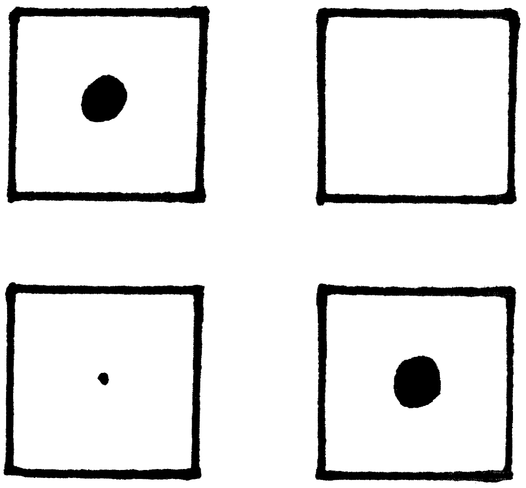


Figure 14. If there is a dot in one image and nothing in the other (top), then a mismatch the size of the dot is generated in one direction of match and only a single point mismatch is generated in the other direction (bottom).

size of a connected match area is a good indication of how good the match is. In particular, regions where two different random noise patterns are matched have only very small connected match regions. This difference in connectivity is used to calculate matching strength. For edge finder testing, matching strength is used only for pruning responses due to noise. For other applications, such as the stereo analysis algorithm described in Chapter 6, these strengths are also used to choose among competing matches.

Matching strength is computed using the star-convex sum operation described in Chapter 2. Recall that this operation builds the largest neighborhood of a cell x, up to some maximum radius r, in which every cell can be joined to x by a connected, straight path consisting entirely of cells in the neighborhood. Since the paths must be connected, star-convex neighborhoods cannot cross boundaries. In the case of the matching strength computation, all non-matching cells are interpreted as boundaries. Thus, the star-convex neighborhoods are required to contain only cells marked as matching.

The star-convex sum operation implemented for the edge finder uses a maximum radius of 3 cells. For computing matching strength, a larger support neighborhood is desirable. For this implementation, I have cascaded two iterations of star-convex sum using radius 3 cells. Another option would have been to build another version of star-convex sum using a wider radius.[5] In the input to the first layer of star-convex sum, all matching cells are given the value 1. The final output values are divided by 10, yielding strengths in the range $[0, 240]$.

Star-convexity was used, rather than connectedness, for two reasons. First, it can be computed more efficiently, because it requires searching only straight paths, rather than all paths, out from the cell of interest. Secondly, it reduces the amount of "leaking" through small gaps in the boundaries. Finally, because the shape of the neighborhoods adapts to the boundaries present, cells near the edges of match regions and in thin match regions can gather as much support as possible without contamination from the nearby non-matching regions.

Once matching strengths have been computed, the algorithm removes responses indistinguishable from noise. This is done using the same noise suppression algorithm built for the edge finder, except that no gap filling is done.[6] Specifically, a (third) iteration of star-convex sum is done. If the result of this sum falls below a set threshold (currently 3000), the cell is considered to be noise and is re-classified as non-matching. This is repeated twice, as in the edge finder. Figure 15 shows the over-threshold matching strengths and the clean match map computed for the image match from Figure 13. In Figure 13, many cells in the

[5] This is easy in theory, but difficult in practice, because the star-convex sum operation is hand-coded, for efficiency.

[6] In retrospect, I think that it was probably a bad decision not to use gap filling after noise suppression, to eliminate tiny topological flaws.

noisy regions were classified as matching. As you can see, in the clean match map, these regions are entirely classified as non-matching.

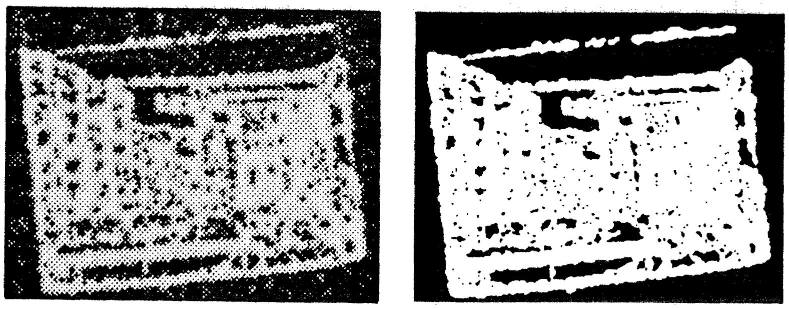


Figure 15. Left: The matching strength computed for the match in Figure 13. Right: The clean match map obtained by pruning matches with low strength.

The same matching procedure can also be used for other matching tasks. For example, Figure 16 shows a clean match map for two alignments of a stereo pair. In stereo analysis, the two images must be matched at a range of alignments and the best matches chosen over all alignments. Chapter 6 describes in detail the control structure needed to handle this. As we see in Chapter 6, the same control structure used for stereo matching may also be useful in motion analysis, because the two problems are very similar.

Figure 17 shows a match of a textured pattern against itself. At the alignments at or near the period of the texture, many cells match in the clean match map. At other alignments, few cells are identified as matching. As in stereo analysis, additional machinery would be required to extract an estimate of the period for each cell from such a sequence of matches. This is a topic for future research.

Finally, Figure 18 shows the results of matching outputs from different scales of the edge finder. At each scale, the program has identified those cells that rep-

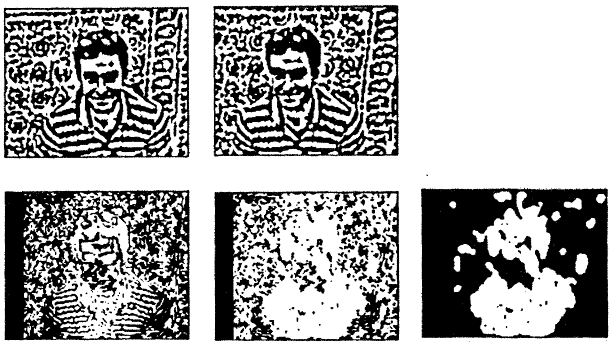


Figure 16. Top: Edge finder output for two images in a stereo pair. Bottom: matching them at an alignment appropriate for the man's shirt and nearly appropriate for the rest of the man. From left to right: the match before adjustment, the raw match map (after adjustment), and the clean match map. In all cases, matching cells are shown in white.

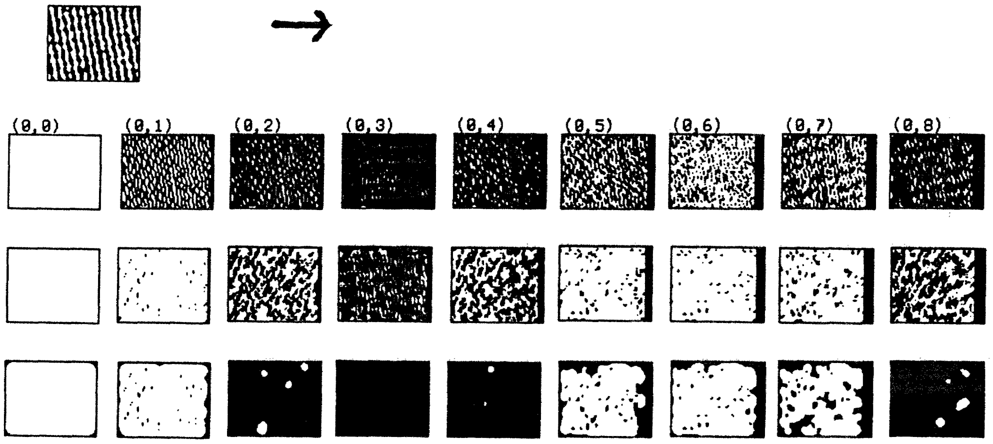


Figure 17. A match of a textured pattern against itself at a range of displacements, moving the image horizontally against itself. Top to bottom: image, match before adjustment, raw match map (after adjustment), and clean match map. In all cases, matching cells are shown in white.

resent edge information that is topologically different from that at the next finer scale. As you can see, the second finest scale shows much the same regions as the finest scale, but in less accurate form, but the third scale shows a totally distinct set of edges. The matcher correctly identifies the third scale as representing primarily new information and the second scale as largely redundant.

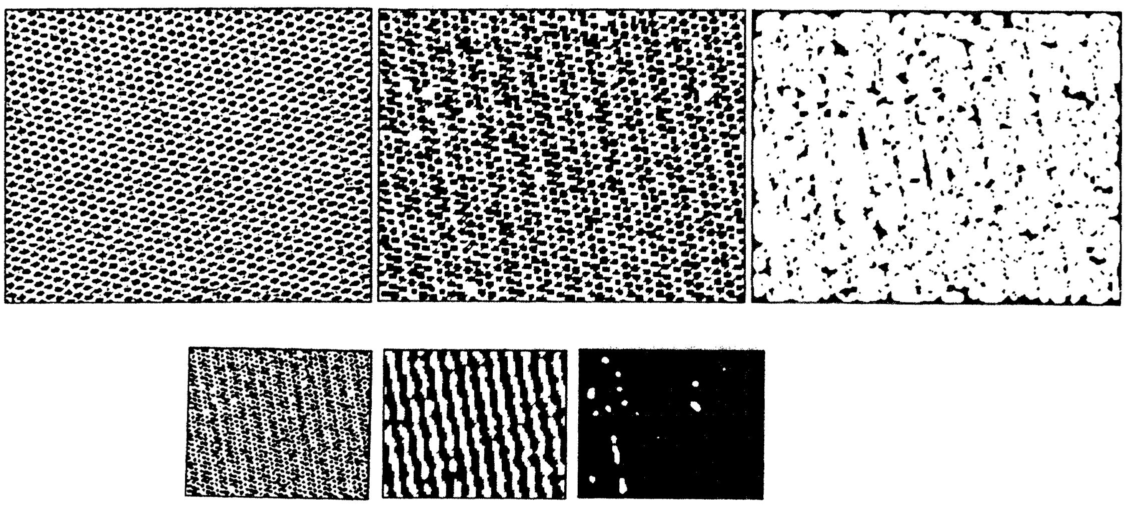


Figure 18. A match between edge finder outputs at different scales. Each row shows the match between consecutive scales of representation. Left to right: fine-scale output, coarse-scale output (expanded to the same size as the fine-scale output, and clean match map.

5. Measuring boundary motion

The final stage of analysis in the matcher computes descriptions of how boundaries were moved during adjustment. These descriptions include both estimates of the overall motion of patches of the image and also estimates of the local fluctuation in boundary locations. In this section, I describe how both types of measurements are computed, using the clean match map.

How boundary motions should be described depends on the application. In the edge finder evaluations presented in Chapter 9, there is no overall motion of the images relative to one another. Thus, boundary adjustment only corrects for fluctuations in boundary locations caused by camera noise (and, in one test, changes in digitization). In stereo matching, on the other hand, one of the images may be shifted relative to the other. As we will see in Chapter 6, the amount and direction of this motion must be assessed. However, local fluctuations in boundary locations are not interesting to this application and should be suppressed.

The amount of fluctuation in boundary locations, required by edge finder testing, can be assessed very easily. It is measured by counting the number of cells marked in the clean match map as matching and as having had their labels altered during adjustment. This figure depends both on the amount of motion of each boundary and the total amount of boundaries in the image. Therefore, the numbers reported in Chapter 9 are normalized by the number of edge cells in the image (divided by two).[7]

The more difficult task is to determine overall motion of a patch of image from the clean match map. One difference between overall motion and local fluctuation is that overall motion is a signed (vector) quantity and total fluctuation is an unsigned (vector magnitude) quantity. Thus, calculating overall motion requires determining the direction of motion at each cell, in addition to its magnitude.

As we saw in Chapter 3, adjustment regions in the clean match map have a special form. Each boundary that has been moved has a connected adjustment region to one side of it, as shown in Figure 19. This is a consequence of the method of applying adjustment operations that was described in Section 3. When motion is perpendicular to the boundary, the width of the adjustment region indicates the amount of motion the boundary has undergone. The direction of the motion

[7] See Chapter 9 for more detailed discussion of this measure.

is indicated by the side of the boundary to which the adjustment region lies. If the boundary may have moved in other directions, the parsing problem is more complicated.

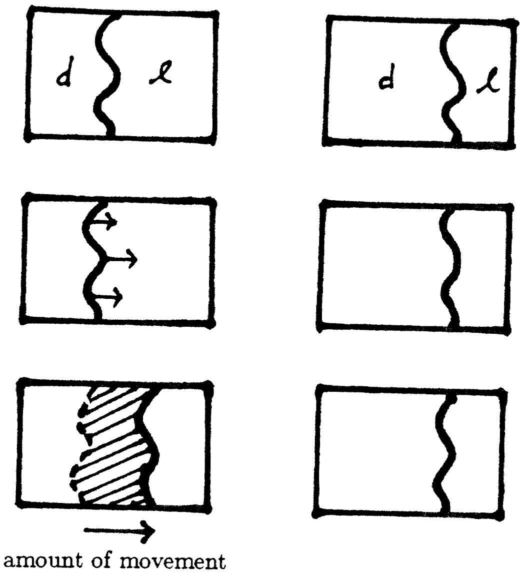


Figure 19. Adjustment regions lie to one side of boundaries and indicate how much the boundary has been moved.

The current implementation makes the assumption that the horizontal and vertical components of motion can be measured separately. That is, the width of the adjustment region is computed for a horizontal and a vertical search path, starting from each edge cell, yielding measurements of the two components of motion at that cell. This is a dubious heuristic for computing the motion, justified primarily by the observation that the errors introduced in this method tend to cancel out in later smoothing. Since this computation was not central to this thesis, more sophisticated methods such as those described by Hildreth (1984) were not explored.

Figure 20 shows a picture of the computation for horizontal motion, starting at some edge cell x. The computation is done in two halves, one of which moves left and one of which moves right. The figure shows the computation for the leftward pass. The algorithm first verifies that x is an edge cell and that there is a boundary between x and the cell to its right. It then proceeds leftward, counting cells until it reaches either a boundary or a cell that is not marked as having been adjusted. The cell x is included in this count. The count reflects the amount of leftward motion of the boundary to the right of x. The computed horizontal motion at x is the sum of the leftward and rightward computations, though it is rare for more than one of them to return a non-zero result.

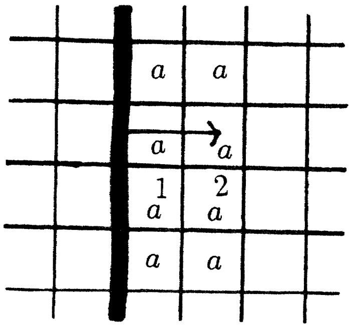


Figure 20. The amount of motion in a direction is computed by measuring the length of a straight path through the adjustment region (cells marked a). Each path starts at a edge cell and stops when a non-adjusted cell is encountered.

Once the two components of motion have been estimated for each edge cell, these measurements are interpolated to non-edge cells and smoothed to cancel out effects of local fluctuations in boundary locations (e.g. due to camera noise). Both interpolation and smoothing are simple applications of star-convex sum. To interpolate motion values, all edge cells are marked with their computed motion and other cells are given the value 0. Star-convex sum is then run on the result. For each cell, this sum must be normalized by the total number of edge cells in the star-convex neighborhood, which can also be computed using star-convex sum. If there are no edge cells in the neighborhood, some default value must be provided. For the current stereo system, this value is 0.[8] Each boundary is flanked by two sets of edge cells, but only the edge cells in the adjustment region return non-zero motion measurements. Thus, the average motion over all edge cells in some region must be multiplied by 2. Smoothing is done in a similar manner, using two applications of star-convex sum to average motion measurements.

It is important to note that the spreading and smoothing steps in this motion calculation are confined to cells marked as matching in the clean match map. That is, the non-matching cells are considered boundaries that the star-convex sum operation cannot cross. This prevents values in matched regions from being corrupted by values from non-matched regions. In stereo and motion analysis, this reduces smoothing across depth discontinuities, because typically only one of the surfaces meeting at a discontinuity matches at any given alignment.

In some contexts, such as stereo, humans can judge relative motion of two images to extremely high precision. There are several ways in which a matcher of this sort could achieve sub-pixel accuracy in boundary motion measurements. First, the measurement of motion at edge cells could use sub-pixel edge locations, if they are available from the edge finder. All that is required is to indicate, for each cell, how much of the cell is dark and how much is light. The matching algorithm would use the whole cell locations and proceed as described above. However, the motion measurement would count these cells using the appropriate

[8] By itself, this would cause stereo values to drift to zero in regions of uniform color. However, as described in Chapter 6, the stereo matcher biases matching strengths so as to prefer matches similar to those obtained at coarser scales. This causes disparities computed in regions of uniform color to tend towards the coarse scale disparities and the default value is only important at very coarse scales.

fractions. This idea has not be incorporated into the current implementation.

Secondly, sub-pixel disparities could be computed due to the smoothing of motion measurements, because errors due to quantization and noise in boundary locations tend to cancel out. There is, of course, a tradeoff between the precision of the calculated disparities and the precision with which changes in disparities can be localized in space. A final possibility is that, if the same scene can be viewed for an extended period of time, small eye movements (always happening in human vision) would cause boundaries to move relative to the digitization. If it is possible to match stereo outputs obtained from these different views, the temporal averaging would increase the precision of discriminations. The current implementation reports boundary motion to the nearest tenth of a cell,[9] but I do not have detailed data on errors in these measured disparities.

6. Other approaches to matching

The image matcher implemented for this thesis compares images on the basis of edge finder output. Previous image matchers can be classified by the types of features they match. There are four types of features commonly used: raw intensity values, easily identified points, extended boundary segments, and edge finder output. In this section I survey these four approaches to matching, concentrating on the first three types. In Section 7, I provide a more detailed discussion of recent algorithms based on edge finder output. Barnard and Fischler (1982) also provide a survey of some of the earlier techniques used in stereo matching.

The features used in matching have a large influence on the types of matching strategies employed. For example, algorithms using features such as boundary locations return disparity values at a large number of points. However, the large

[9] All calculations are done using integer arithmetic and this is implemented using an integer multiplier.

number of features to be considered forces them to use relatively simple matching strategies. Algorithms using easily identified points or extended boundary segments can use more elaborate strategies for disambiguating candidate matches, because they have fewer features to consider per image. However, they also return disparity values at only a sparse set of points.

A number of previous algorithms (e.g. Quam 1984, Mori, Kidode, and Asada 1973, Levine, O'Handley, and Yagi 1973, and Barnard 1986) have matched images directly, without going through an edge finder. The basic idea behind these intensity-based stereo systems is to match cells with similar intensities. Simple correlation of intensities over neighborhoods has been used, e.g. by Gillett (1988). Baker (1982; Baker and Binford 1981) uses similarity of intensity values in interpolating disparities between matched boundaries. A typical problem with intensity correlation is that intensity values may differ in the two views, due to the change in viewing angle or varying adjustments of the cameras. In order to cope with this problem, Gennert (1986) adds a smoothly varying multiplier term, which is reconstructed along with the match. Scott's (1986) intensity-based motion algorithm calculates the reliability of each component of the motion estimate at each point and uses these reliabilities to influence reconstruction of the motion field. Another variant on this theme is due to Kass (1983a,b), who suggests using smoothed derivatives of the image intensities, at a range of scales, as matching features.

Intensity-based matching has slightly more information at its disposal than boundary-based matching, because edge finders discard information about smooth slopes in intensity. Also, although contrast magnitude information is available from most edge finders, it is often ignored in boundary-based matching. A good way to get a feeling for what information is being lost is to look at the examples of reconstruction from sign bits presented in Chapter 4, Section 8. Because edge finder output is available at multiple scales, some types of intensity differences and some types of intensity slopes are preserved even in sign-bit representations. It is unclear whether the additional information offered by full grey-scale information is a help or a hindrance in stereo matching. Random-dot stereograms can be fused even when there are large differences in contrast, however Bülthoff and Mallot (1987) present psychophysical data suggesting that intensity values can play a role in matching.

There are a few examples of intensity-based algorithms that use frequency-space techniques rather than direct spatial analysis. Several researchers (Bajcsy 1972, 1973, Matsuyama, Miura, Nagao 1983) have used Fourier transform techniques to analyze texture periodicity. Yeshurun and Schwartz (1987) propose an analytic algorithm for stereo matching of grey-scale images. This technique juxtaposes two patches, one from each stereo image, so as to create one image. The algorithm then looks for stereo disparity using a technique, known as cepstral filtering, originally developed for detecting echos in auditory signals. Both techniques transform spatial periodicity into features in the frequency domain and then transform the results back into the spatial domain. It is unclear whether this is an improvement over direct spatial matching.

There are also a few techniques that re-cast the matching problem as one of matching image sequences or textured images against templates describing idealized features. We have seen this approach used in edge finder design. It has not been used in stereo analysis, but it has been used in texture and motion analysis. For example, Bolles, Baker, and Marimont (1987) analyze motion by detecting the 3D surfaces traced out by image boundaries across time. Heeger (1987) uses spatio-temporal Gabor filters that are tuned to an ideal edge moving through time. Bovik, Clark, and Geisler (1987) use spatial Gabor filters in a similar way to detect a subclass of periodic textures.[10] Zucker (1985) and Kass and Witkin (1985, 1987) use similar techniques to detect texture orientation.

The second basic type of matching algorithm looks for features in the image that can be easily identified in the other image. These features might include simple configurations such as corners, spots, or more complex patterns of local texture. The features can be identified either in the grey-scale image directly or in the output of an edge finder. Researchers in stereo and motion analysis who have used this type of approach include Barnard and Thompson (1980), Lawton (1983), Moravec (1977, 1981), Nevatia (1976), Hannah (1980) and Gennery (1977).[11] There are two difficulties with this approach. First, it has proved difficult to define features that can be reliably detected. Secondly, under the best of conditions, relatively few locations in the images are matched. This results in a very sparse disparity field that must be filled in by unspecified means.

The third group of stereo algorithms uses edge finder output, but the boundaries are parsed into extended linear segments and these segments are then matched. This approach is used by Medioni and Nevatia (1985) and Ayache and Faverjon (1987). The linear segments matched by these systems are relatively sparse, though not as sparse as easily identified features. However, the sparseness allows more sophisticated matching strategies to be used than is feasible for matchers using raw edge finder output. Furthermore, this technique imposes a limited type of figural continuity. However, because boundaries must be described using sets of line segments, curved boundaries are poorly represented. Boyer and Kak (1988) carry this approach one step further and match

[10] They can only detect textures that are not only periodic, but where the texture matches itself on the half-period, but with opposite phase.

[11] These researchers all treat motion and stereo processing as instances of the same problem.

extremely sparse high-level descriptions of regions in the two images.

Finally, there are quite a variety of algorithms that match images on the basis of raw edge finder output. These algorithms make use of the location and contrast sign of all points on boundaries, and sometimes also orientation and contrast magnitude information. They produce relatively dense disparity measurements, except in extended regions of uniform intensity. My matcher is most closely related to this class of algorithms and a detailed comparison of this class of algorithms is done in the Section 7.

7. Other matchers using edge finder output

Recent algorithms from a number of domains match edge finder outputs. Applications for this type of matching include stereo matching, motion analysis, analysis of texture periodicity and orientation, evaluating edge finders, and matching edge finder outputs from different scales. The techniques used in different domains are very similar. By definition, boundary locations are used in all such matchers. Shape information, such as boundary orientation, is occasionally used, but connectivity or topological information is rarely exploited. The contrast sign across boundaries, i.e. which side of the boundary has darker intensity values, is widely considered a reliable feature that must be matched. Although contrast amplitude is occasionally used, it is unclear that it is reliable. In this section, I review previous proposals for matching edge finder output.

The most heavily studied edge finder matching problem is stereo matching. Boundary-based stereo matchers have been proposed by Mayhew and Frisby (1981), Pollard, Mayhew, and Frisby (1985), Grimson (1981a,b, 1985), Marr and Poggio (1976, 1979), Hoff and Ahuja (1987), Prazdny (1985), Ohta and Kanade (1983, 1985). Drumheller and Poggio (1986), Baker (1982) and Baker and Binford (1981). Nishihara's (1984) algorithm matches dark/light labels, without explicit boundary information, but is still similar. Medioni and Nevatia (1985) and Ayache and Faverjon (1987) parse boundaries into extended linear segments and match these segments between two stereo images.

Boundary-based matching approaches to motion analysis seem to be less common. The only one that seems parallel to the stereo matching examples is described by Little, Bülthoff, and Poggio (1987). Spacek (1985) also matches contours, but constrains the matching process by identifying and matching high curvature points along boundaries. Short-range motion algorithms, such as the ones described by Hildreth (1984) and Buxton and Buxton (1984), are interesting from the point of view of estimating the direction of boundary motion. However, they need not solve the matching problem, because they deal with only small boundary motions.

Boundary-based matching in other domains has been explored more sporadically. Although the idea of comparing edge finder output across scales has been around since at least Marr and Hildreth (1980), no researcher has properly addressed the question of how it should be done. Since Witkin's (1983) scale-space proposal, it has become very popular to track features across scales (e.g. Bergholm 1987, Ponce and Brady 1986, Asada and Brady 1984, Canny 1983, 1986). However, as Witkin and others (particularly Canny) have noted, features can change drastically between scales. Thus, it is necessary to distinguish which coarse-scale features are blurred versions of finer-scale features and which coarse-scale features represent new information. Witkin's original proposal for matching features assumes certain constraints on the transition between representations at different scales. For example, he assumes that new features cannot appear out of nowhere at coarser scales. While this is true for the 1D features he considers, it is not true for real image features.[12] So far, Canny's feature synthesis proposal still seems to be the only algorithm that matches real image features.

Edge finder evaluation algorithms are all but non-existent. Recent researchers who have attempted quantitative evaluations include Sher (1987a,b), Pratt (1978), Nalwa and Binford (1986), and Haralick (1982). All of these researchers state that they want to separate boundary motion from real missing or extraneous boundaries. All of the evaluations, however, are done for simple, synthetic images and the matching techniques described seem inadequate for handling complex natural images. For example, Nalwa and Binford, as well as Sher, assume that boundaries move less than a cell from the correct location. This is not adequate for handling natural images.

Boundary-based analysis of texture periodicity and orientation is roughly in the same state. Vilnrotter (1981; also Vilnrotter, Navatia, and Price 1986) describes the only boundary-based periodicity algorithm that I know of. Although it is not expressed this way, her algorithm is equivalent to matching the image against itself, as I did in the texture example in Section 4. As far as I know, matching techniques have never been used to analyze texture orientation, although from a mathematical point of view it is similar to periodicity. For example, if the example from Section 4 is matched against itself at displacements along its dominant orientation, the match pattern is as shown in Figure 21. Whereas a periodic pattern matches against itself only at discrete locations, an oriented pattern matches against itself for an extended connected set of locations, along some straight path.

The basic information available in boundary-based matching is the set of boundary locations. Edge finders may be able to provide these locations to sub-

[12] In particular, adding noise suppression to a feature detection algorithm, as is commonly done in edge finders, allows new features to emerge at coarse scales.

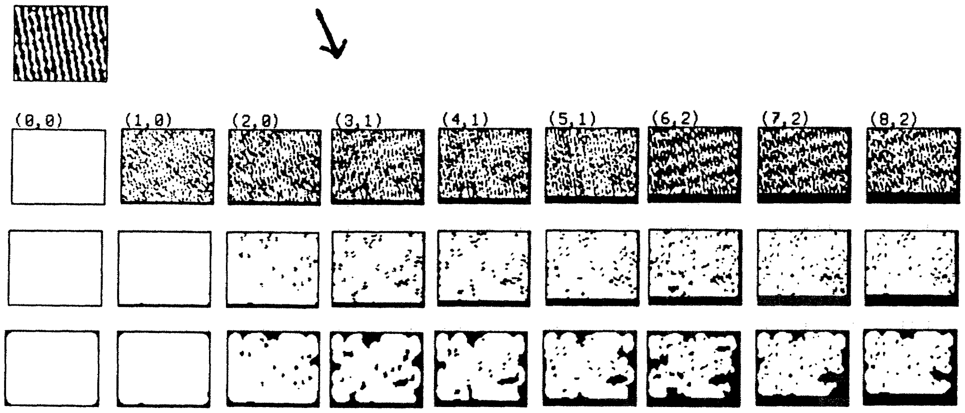


Figure 21. Matching an oriented pattern against itself at a range of displacements. Top to bottom: image, match before adjustment, raw match map (after adjustment), and clean match map. In this series, the image was moved down and to the right, in a direction approximating the orientation of the stripes. Notice that most of the image matches at all displacements.

pixel resolution and this information could be used to compute disparities to higher precision. However, sub-pixel edge finders are still at the experimental stage and are rarely incorporated into matching algorithms. In stereo analysis, if vertical disparities are assumed not to exist, the matching problem can be made one-dimensional. In this case, representing sub-pixel boundary locations is not technically difficult. If vertical displacements are possible, however, direct representation of sub-pixel information requires expanding the size of the image. Depending on the details of the matching algorithm, it may be possible to use partial representations of the information, such as the cell fraction descriptions suggested in Section 5. Note also that for inter-scale matching, and perhaps for texture analysis, sub-pixel boundary locations are not useful.

Many boundary-based matchers use boundary orientation information in addition to boundary locations. Orientations provide information about the shape of the boundary near the cell of interest or, in some cases (e.g. the edge finder described in Canny 1983, 1986) limited sub-pixel information. In my system, this information is made largely redundant by the use of figural continuity information. It is unclear to me whether orientation information is reliable for most matching tasks. Typically, a wide allowance (±30 degrees in Grimson's algorithm) is made for errors in orientation. When a boundary has fine-scale shape, such as serrations, boundary orientations may be extremely sensitive to changes in scale (in stereo, motion, or inter-scale matching) or scene irregularities (in texture analysis).

My matcher makes extensive use of the image topology in deciding whether two sections of image can be matched. This information has only rarely been exploited by previous matching algorithms and then only in the weaker form of boundary connectedness. The only proposal using full homeomorphism of regions is Chen (1985). He proposes using topological structure to explain the results of some psychophysical experiments on motion perception. However, his experiments are confined to simple, isolated shapes. It is unclear how to translate his proposal into an implementable algorithm.

Use of boundary connectivity information in matching has been proposed by Mayhew and Frisby (1980, 1981), Baker (1982), Baker and Binford (1981), Ohta and Kanade (1983,1985), Mohan, Medioni, and Nevatia (1987). The first three proposals are confined to requirements that adjacent boundary cells on different horizontal lines be matched to adjacent boundary cells in the other image. The proposal of Mohan, Medioni, and Nevatia (1987) seems confined to individual straight boundary segments. In addition, stereo matching algorithms that match extended boundary segments (Ayache and Faverjon 1987, Medioni and Nevatia 1985) use boundary connectedness implicitly, but they match only relatively sparse segments and handle non-straight boundaries poorly. The proposal closest to the one described in this thesis is due to Grimson (1985). He extends Mayhew and Frisby's idea to a requirement that every match belong to a connected boundary of sufficient length (perhaps with one or two gaps) all of whose points match at a similar disparity.

A major weakness in all of these formulations is the restriction to a single connected boundary. Consider an image whose texture consists of many small dots. If the dots are too small, no single boundary will meet minimum length requirements. If the dots are larger, all dot-to-dot matches will be accepted. In an areal formulation, such as mine, a potential boundary match can collect support from other nearby boundaries, even if they are not connected to it (as in dot-like textures). Furthermore, my matcher can split boundaries[13] when they cross depth boundaries. When objects are covered with fine texture, as in some of the stereo pairs presented in Chapter 10, intensity boundaries often run across depth boundaries.

Boundary-based matchers typically also use information about the sign of the contrast across each boundary, i.e. an indication of which side of each boundary has higher intensity values. This information can be expressed in many forms. In Grimson's (1981a,b, 1985) algorithms, it is expressed as a sign in the boundary description, a technique that is only stable because boundaries near horizontal are not used. Alternatively, boundary orientation and contrast sign may be combined into a signed orientation with a 360 degree range. In my matcher and in Nishihara's (1984), contrast sign is encoded using cell labelling. Contrast sign seems to be reliably preserved between images in most matching applications. Occasional exceptions occur, e.g. at occlusion boundaries in stereo analysis, but they seem to be rare in practice. In one form or another, almost all image

[13] Even single dots in random-dot stereograms! See Chapter 10.

matching algorithms require that contrast sign be preserved.

Contrast sign information seems to be very important in human visual perception. For example, random-dot stereograms with reversed contrast cannot be fused. In higher-level processing, Cavanagh (1987) shows that shadows such as those on faces are only parsed correctly if they have lower intensity than the surrounding regions. Pearson and Robinson's (1985) work on low bit-rate image coding of sign language also suggests that contrast sign is essential to producing output acceptable to naive observers. The effect of sign information can be appreciated by comparing edge and cartoon output from my edge finder, shown in Chapters 4 and Chapter 9. While untextured objects with simple shapes can be recognized from unsigned boundary locations, it is difficult to parse complex scenes, textured regions, or objects with complex shape without sign information. Human faces, in particular, look extremely poor when represented with unsigned boundary maps.

Contrast magnitude, on the other hand, is typically ignored in stereo matching other visual analysis tasks. Although most edge finders can measure the magnitude of the intensity change across boundaries and humans can clearly estimate this magnitude, only a few matching algorithms use this information to evaluate boundary matches. Researchers using this information include Canny (1983, 1986) (inter-scale matching) and Pollard, Mayhew, and Frisby (1985) (stereo matching). What evidence is available suggests that this information is less important to human perception than contrast sign information. For example, random-dot stereograms with different contrast magnitudes, but the same sign, can be fused without problems. Furthermore, objects and scenes can easily be recognized from black and white versions of images (such as those produced by my edge finder).

Algorithms matching images on the basis of edge finder output invariably place boundaries where there are step-edge-like responses, e.g. at peak responses of a first difference operator or zero-crossing of second difference operator. However, other types of edge information can be detected and there is some evidence that they should be used in matching. For example, Mayhew and Frisby (1981) present psychophysical data suggesting that humans must be using information in addition to step-edge boundaries when matching stereo images. They suggest that this additional information may consist of locations of peaks and troughs in the second differences. Watt and Morgan (1983) make a similar suggestion, based on psychophysical experiments on human perception of edge blur.

The Phantom edge finder detects both zero-crossing and roof edge responses, but my matcher uses only zero-crossing boundaries are used in my matcher implementation. There are at least two ways that roof edge information could be incorporated. The stereo matcher could be extended to use roof edge information directly. Alternatively, the matching program could use locations of all label transitions, not just zero-crossings, as boundaries. This would allow responses of both types to be used together in matching. Classification of responses into roof edges vs. zero-crossings would then be postponed until after stereo fusion. This solution might be able to account for Mayhew and Frisby's (1981) data, though additional experimentation would be required to test this.

8. Conclusions

This chapter has shown how to build a matcher that preserves topological structure. The matcher is interesting for several reasons. It illustrates several ways in which topological structure can be useful in solving an important practical problem. It also exercises the mathematical machinery developed in

Chapter 11 more thoroughly than the other applications presented in the thesis. Finally, it is interesting as a possible solution to problems that are both central to visual analysis and difficult for existing computer algorithms to handle. The acid test of its performance comes in Chapters 6, 9, and 10, when the matcher is applied to analysis of stereo images and to edge finder evaluation. I summarize the other points in this section.

The matching algorithm developed in this chapter directly tests one central hypothesis of this thesis, that topological structure is important in solving practical reasoning problems. Equivalence of topological structure is the main constraint on the matching process. If the only requirements were that labels be preserved and the correspondence not deviate much from the original alignment, considerable scrambling of images would be possible. Using this constraint, the algorithm makes a sharp and intuitively reasonable distinction between matches and non-matches. This is illustrated by the results presented in this chapter and later chapters. In particular, the results of edge finder testing presented in Chapter 9 show convincingly that the algorithm consistently rejects matches between two random noise patterns, but not between two copies of the same signal, even when slightly corrupted by noise.

The analysis phase of the stereo computation also contains several algorithms that use connectivity. Two of these algorithms measure the size of a connected neighborhood. The matching strength computation measures the area of star-convex match neighborhoods, whereas measurement of boundary motion measures the length of a connected path through an adjustment region. Furthermore, motion measurements are interpolated and smoothed by algorithms that are constrained not to cross boundaries. Thus, in addition to the use of the full topological structure in the adjustment phase, the matcher also offers more examples of uses of connectivity similar to those in the edge finder described in Chapter 4.

The development of boundary adjustment operations, unlike most other applications presented in this thesis, fully exercises the mathematical machinery developed in Chapter 11. Although the idea of using boundary topology has been proposed before (particularly in stereo matching), previous researchers have not been able to provide a sufficiently clear or powerful formulation to make full use of the idea. To attack matching in the way that I did requires a large investment in mathematical machinery and development of techniques for building algorithms. This investment would never have made sense without the additional context of problems from other domains requiring similar machinery.

Chapter 6: Stereo analysis

1. Introduction

As we saw in Chapter 3, the task in stereo matching is to establish a correspondence between two images of the same scene taken from slightly different viewpoints. In this chapter, I present a new stereo matching algorithm based on the image matcher discussed in Chapter 5. We have seen how this matcher can compare two images at one fixed alignment. This chapter describes the control structure needed to search a series of alignments to locate good matches.

Stereo matching is a good domain for testing the image matcher, because it is a well-studied problem and the correct answer to each matching task is relatively clear. Some evaluation problems still arise. For example, what people see in a synthetic stereogram rarely corresponds exactly to the input depth specifications. However, since stereograms produce vivid subjective perceptions, the desired output is much clearer to human observers than in tasks such as inter-scale matching. Furthermore, substantial psychophysical data about human stereo perception is available. This data is useful in making design decisions for computer algorithms.

This chapter begins with an overview of the control structure used in the stereo algorithm. This control structure consists of two parts. First, camera positions are adjusted and the algorithm chooses the set of alignments at which to search for matches. This is described in Section 3 and compared to previous algorithms in Section 4. After matching is done at each alignment, the results from different alignments must be combined. This process is described in Section 5. Sections 6 and 7 discuss types of matching constraints used in previous stereo algorithms and analyze how they are related to the constraints used in my implementation.

As I mentioned in Chapter 3, the new stereo matcher offers two advantages over previous algorithms. First, the topological continuity constraint makes its match evaluations more robust. This allows it to disambiguate larger numbers of candidate matches without becoming confused. Secondly, the matcher requires support neighborhoods for strength and disparity to be connected sets of cells at a similar disparity. This prevents results for cells near depth boundaries from being contaminated by values on the other side of the boundary. Chapter 10 presents detailed results of the stereo algorithm's performance on both natural and synthetic images. It also shows an example of how an adaptation of the algorithm might be used for motion analysis.

2. Overall control structure

This section provides an outline of the stereo algorithm as a whole and a brief description of its main components. I describe both the control structure of the implemented off-line stereo matcher and also a control structure that would be more plausible for real-time or biological processing. I also sketch the form of the input and output to each stage of the stereo algorithm. Later chapters discuss each step in more detail.

The input to the stereo analysis is the result of Phantom edge finder applied to both images in the stereo pair. Two points about this edge finder output are relevant to stereo matching. First, in these outputs, the effects of camera noise have been suppressed. In a few previous stereo algorithms, described by Grimson

(1981a,b, 1985) and Gillett (1988), noise is not adequately suppressed in regions of uniform intensity. This makes it difficult to distinguish regions with rivalrous fine texture from regions with matching uniform intensities.

Secondly, many previous algorithms eliminate boundaries that are close to horizontal before doing matching. The reason for this decision is that such boundaries cannot contribute useful information in assessing disparity. This is true for horizontal disparity, but not for vertical disparity. In fact, these boundaries are the *most* useful type for constraining vertical disparity. Thus, when both types of disparity may be present, it is essential to use boundaries of all orientations, as the implementation described in this thesis does.

The stereo matcher uses a coarse-to-fine control structure. As we saw in Chapter 4, the edge finder produces edge maps at a range of scales Stereo analysis at each scale is given the output of the edge finder at that scale, together with the disparity and match maps computed at the next coarser scale. In order to avoid dependence on choice of the coarsest scale, all scales available from the edge finder were used. Since the coarsest scale is smaller than 10 cells in one or both dimensions, it typically provides no successful matches, but little time is wasted in analyzing it.

There are three steps of processing at each scale:

- adjusting the relative positions of the images and choosing a set of alignments,
- matching images at each alignment, and
- choosing the best disparities over all alignments.

The details of the second step were discussed in Chapter 5. This chapter concentrates on the first and third steps. A diagram of this control structure is shown in Figure 1.

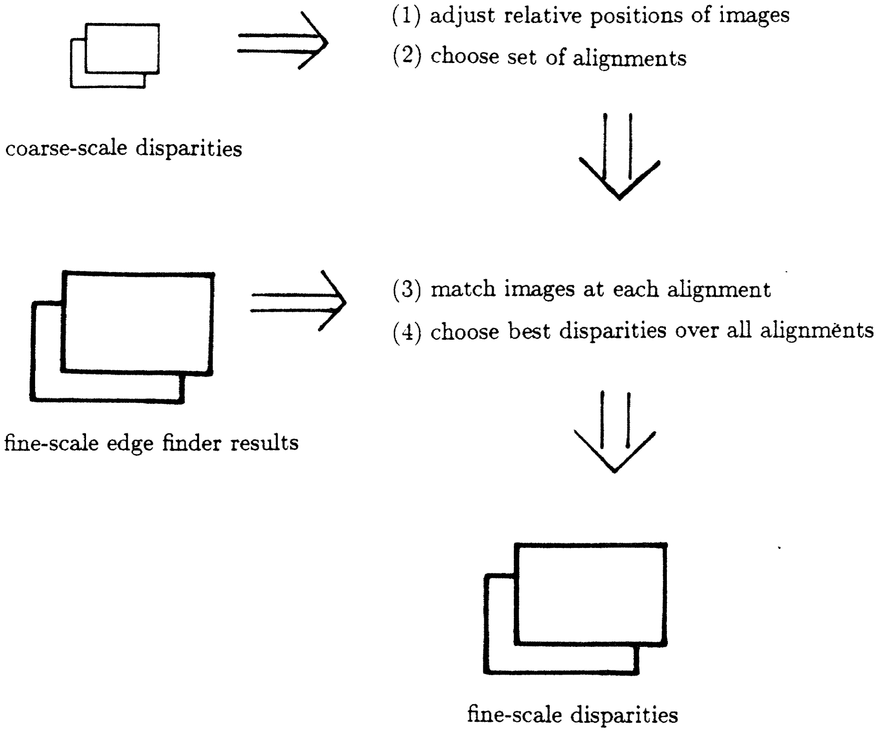

(1) adjust relative positions of images
(2) choose set of alignments
coarse-scale disparities
(3) match images at each alignment
(4) choose best disparities over all alignments
fine-scale edge finder results
fine-scale disparities

Figure 1. The control structure for the implemented stereo algorithm.

The implemented control structure was designed to operate efficiently for off-line processing of stereo images. In real-time stereo matching, the control structure shown in Figure 2 would be more appropriate and a better match to what is known about human stereo processing. These two control structures are able to fuse slightly different types of stereo pairs. Although the differences may be significant in detailed comparisons to human performance, they are small enough not to be of interest to my main goal, testing the matcher.

The on-line control structure has the disadvantage that it re-matches each alignment many times if eye position is varied slowly and Panum's area[1] is large. Since the current implementation runs relatively slowly, this would be a serious

[1] The range of disparities that can be fused without eye movement.

problem. Furthermore, implementing the on-line control structure elegantly requires a good model of deciding how to explore the image via eye movements (both foveation and vergence). These issues are beyond the scope of this thesis.

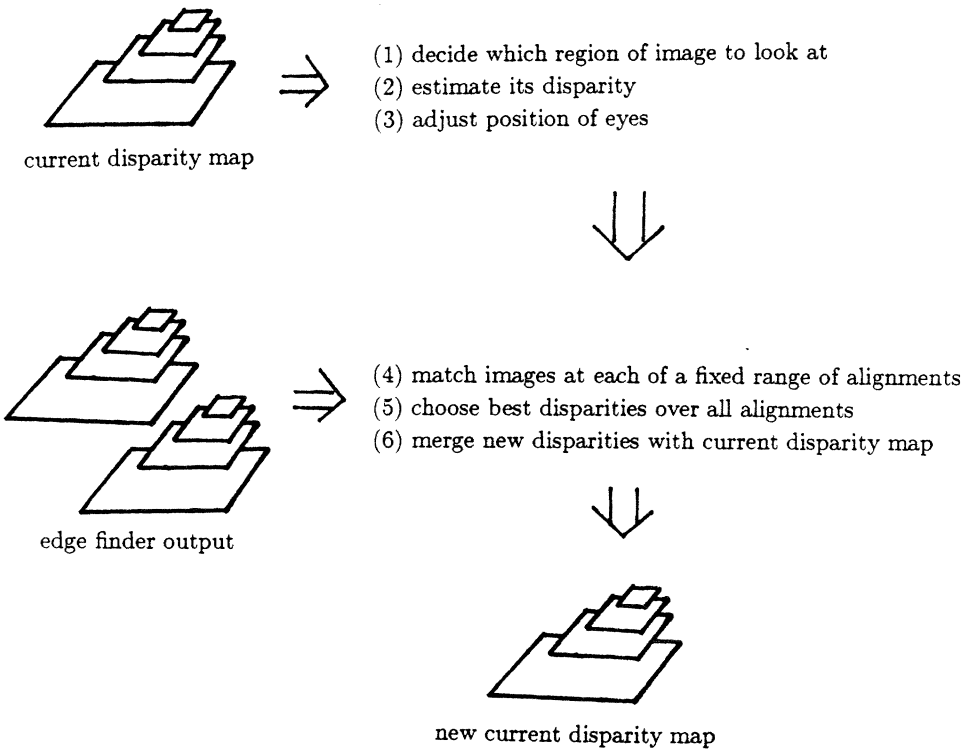

(1) decide which region of image to look at
(2) estimate its disparity
(3) adjust position of eyes
current disparity map
(4) match images at each of a fixed range of alignments
(5) choose best disparities over all alignments
(6) merge new disparities with current disparity map
edge finder output
new current disparity map

Figure 2. A control structure for on-line stereo processing.

The first step in analyzing a given scale is to adjust the relative positions of the images. The goal of position adjustment is to bring the two images as close as possible to exact vertical alignment. Three adjustment parameters are computed from the coarse-scale disparities: a vertical translation, a horizontal translation, and a rotation about the image centers. After these calculations are done, the edge finder output for the two images is shifted (both horizontally and vertically) and rotated. Although this is done in software, the intent is to simulate the effects of corrective eye movements.

All alignments explored by this stereo matcher are translations.[2] Choice of alignments is based on the assumption that the correct fine-scale disparity for each patch of surface is similar to the disparity computed at the coarser scale. Thus, the algorithm searches only a limited range of disparities about each coarse-scale value. The search area was chosen to reflect roughly human capabilities, as discussed in Sections 3 and 4, and is somewhat larger than that considered by most stereo implementations. In particular, the new matcher can hypothesize substantial vertical disparities, which most previous algorithms cannot do.

Once the set of alignments has been chosen, the stereo algorithm then matches the two edge finder outputs at each alignment. As we saw in Chapter 5, the matching algorithm produces three outputs. First, it specifies which cells of the image match at this alignment. For each matching cell, it also supplies a number representing the strength of the match about that cell. Finally, for each matching cell, it estimates the amount of boundary motion, both vertically and horizontally, at this alignment.

The final stage of matching combines the results from different alignments into one match map and one disparity map. At each cell, the disparity value with the highest strength is chosen. This decision is biased in favor of disparities similar to those computed at the next coarser scale. This allows the algorithm to take advantage of the wider context available at coarser scales in deciding among multiple possibilities. This is particularly important in regions of uniform color where many alignments may all match perfectly. The resulting map is then processed to remove outliers and fill small gaps using a modification of the noise

[2] These alignments are all relative to the adjusted image positions. The combination of an alignment and the effects of image adjustment can also contain a rotational component.

suppression algorithm used in the edge finder.

3. Adjusting image position and choosing the set of alignments

The first step in stereo analysis at each scale is to adjust the position of the two images so as to bring them into a vertical correspondence and then choose the set of alignments to be explored at the current scale. The adjustment in position is made on the basis of the vertical disparities computed at the next coarser scale. The set of alignments chosen also depends on the coarse-scale results, but uses both the horizontal and vertical components of disparity. This section describes the details of both algorithms. In general outline, these algorithms are similar to those used in previous stereo algorithms, particularly those of Grimson (1981a,b, 1985).

The software adjustment of image positions used in my implementation is intended to minic the effects of adjusting camera positions in a real-time system. For the images available at MIT, I have been able to use an extremely simple model of distortions due to errors in camera position. This model assumes that alignment errors can be expressed as a translation of one image relative to the other, plus a rotation of one image (equivalently: both images) about the center of the image. Since modelling camera geometry was not my main interest in building this algorithm, I have not explored more sophisticated models of these distortions.

I have also assumed that optical distortion is small enough to ignore. Since my algorithm is tolerant of small errors in image alignment, this assumption is satisfied for the images I have been using. Noticable optical distortion seems to be significant primarily for systems using very wide-angle lenses. If my algorithm were used in such a system, it would be necessary to estimate the distortion beforehand and correct the images so as to eliminate its effects.

At each scale, three adjustment parameters are estimated: mean horizontal disparity, mean vertical disparity, and average rotation about the image center. This estimation is based on the disparity values at all cells that were successfully matched at the next coarser scale. In order to eliminate estimates based on excessively small numbers of cells, the current implementation requires that at least 25% of the image be matched in order to compute a non-zero correction to the image alignment. The coarse-scale disparities, on which estimation is based, are stored internally to the nearest tenth of a cell, although I have not been able to assess their precision in detail.

The two mean disparity parameters are simply the averages of each component of disparity at all cells that matched successfully at the next coarser scale. Rotation is estimated using only the vertical component of disparity, because the vertical component of disparity depends only on the the relative positions of the cameras,[3] whereas the horizontal component also depends on surface depth. Rotation is estimated as the average, over all cells in the image, of

$$\frac{(V - MV)}{Y}$$

where V is the vertical component of disparity at the cell, MV is the mean vertical disparity, and Y is the (signed) location of this cell relative to the image center. This is a relatively unsophisticated method of estimation, but seems adequate for the purposes of the current implementation.

Once mean disparities and rotation have been calculated, each image is rotated and translated by half this amount, so that the two images are aligned vertically. The effects of this translation and rotation are then subtracted from

[3] To a first approximation. Although vertical parallax is possible, its effects will be quite small for most standard stereo viewing conditions.

the vertical and horizontal disparities inherited from the coarser scale, yielding a set of disparities relative to the new image positions. It is these ***net disparities*** that are used in the rest of the stereo computation. Although only vertical disparities are used in estimating rotation, horizontal disparities are also corrected for any effects of rotation.

Before calculating the set of alignments to be searched, the program imposes bounds on the net vertical and horizontal disparities. Net vertical disparities are due to four factors: vertical parallax, inaccuracies in the translation plus rotation model, inaccuracies in the estimates of translation and rotation, and inaccuracies in the coarse-scale disparities. All of these factors should produce only small net disparities. Thus, the current implementation limits vertical disparities to ±2 cells. That is, cells with net disparities beyond this limit are assigned a net disparity of 2 or -2 cells, as appropriate. Since the vertical search radius (see below) is ±2 cells, the program can explore alignments that move the image at most ±4 cells vertically from the adjusted image position.

Net disparities in the horizontal direction reflect differences in surface depth and can be quite large. These disparities are bounded primarily in order to limit the running time of the program. The bound depends on the scale of calculation: net disparities of ±60 cells are allowed at the finest scale and bounds for coarser-scales are adjusted proportionately. For example, the disparity bound at the third finest scale would be ±15 cells. Since the horizontal search radius (see below) is ±10 cells, the largest alignment that could be considered at the finest scale is ±70 cells. Remember that this is the maximum displacement from the mean horizontal disparity. The maximum calculated disparity (mean plus net disparity) could be much higher.

The bound on horizontal disparities was imposed as a placeholder, rather than as a final solution. When an image is complicated and contains wide ranges of disparities, it would seem reasonable to employ more sophisticated search strategies than searching every candidate presented by a coarser-scale match. These considerations would be particularly important for eventual real-time systems that must control not only stereo vergence but also which part of the scene is covered by a high-resolution fovea. One possibility would be to stop search in a given region when a good enough match has been found. This could be most easily done within the real-time control structure sketched in Section 2, rather than using the implemented control structure. A final answer to how stereo exploration is controlled may have to incorporate information about the reasoner's interests, which is beyond the scope of this thesis.

At each scale of analysis, certain cells are not assigned a net disparity, because they did not match at the next coarser scale. At the coarsest scale of analysis, this is true for all cells in the image. These cells are assigned a net disparity of zero. This default value determines the set of alignments considered for these points, as well as the bias used in the final selection of the best disparities.

The adjusted images and the net disparities form the input to the later stages of stereo analysis at this scale. This later processing also requires a set of alignments to explore. The alignments used in the current implementation translate the image by an integral number of pixels. The range of alignments considered is computed by taking the range of disparities suggested by the next coarser scale and extending this by a search radius of ± 10 cells horizontally and ± 2 cells vertically.

As an optimization in the current implementation, not all cells in the image are considered for matching at each disparity. A cell is only considered for matching at a given alignment if the net disparity computed from the next coarser scale differs from that alignment by at most ±10 cells horizontally and ±2 cells vertically. Because there may be errors in the location of depth discontinuities, the program considers not only cells meeting this criterion, but also any cells within ±8 pixels of them. As we see in Section 5, each scale computes two separate disparity maps, one from the perspective of each eye. A cell is considered for matching at a given alignment if either of these coarse-scale estimates satisfies the above conditions. In a parallel implementation, there may be no advantage to this type of optimization, because it may take just as long to process part of an image as to process the whole image.[4]

4. Comparative discussion of search space limitations

Imposing sensible restrictions on the search area at each scale involves a tradeoff between speed of computation and robustness. If the search area is small, then stereograms containing high-frequency patterns cannot be fused at large disparities. Furthermore, the program is sensitive to errors in coarse-scale edge finder output and disparities. Stereograms with extremely large disparities can only be fused to the extent that they contain clear coarse-scale cues as to the correct disparity. On the other hand, the larger the search area, the slower the stereo algorithm runs. The current implementation was run with relatively large search areas, both in order to match estimates of human performance and also to test the robustness of the matching evaluations.

There has been extensive discussion of how large a range of disparities humans can fuse, but the psychophysical data is not definitive. There are two difficulties with determining search areas. First, if the search areas are proportional to the scale of analysis, experiments must be designed so that it is clear which range of

[4] Even in the current serial implementation, there are fixed costs associated with processing an alignment. These limit the effects of this type of optimization.

scales is responding. Secondly, the total range of disparities that can be fused is very large. The search area explored for each cell at each scale corresponds not to this total range of disparities, but to the range of disparities that can be fused without eye movement (Panum's area).

The psychophysical data are summarized by Poggio and Poggio (1984). Measured values for Panum's area seem to be approximately ±10 minutes of arc in both the horizontal and vertical dimensions. Since the measurements in question are for foveal vision, where the center-to-center distance between adjacent cells is about 0.5 minutes of arc (Yellott, Wandell, and Cornsweet 1984, p. 273), this translates into about ±20 cells. It is unclear, however, what scale of analysis this reflects.

Two experiments seem to shed more light on the problem. First, Nielsen and Poggio (1983) report two figures for vertical disparities. They report that an entire image can be fused if it is shifted by no more than 6.5 minutes of arc (13 cells). Secondly, a portion of the image can be shifted by no more than 3.5 minutes of arc (7 cells) relative to the rest of the image. The first case is improved if viewing time is long enough to permit eye movements, whereas the second case remains difficult even with eye movements. These numbers were obtained from judgements of relative depth. Nielsen and Poggio also attempted a form discrimination task, but found that form discrimination was extremely poor.

Nielsen and Poggio's results suggest two things. First, the difficulties in form discrimination suggest that their vertical disparity measurements do not reflect fusion at the finest scale, but at some coarser scale. Secondly, the differences between whole image disparity and disparity of part of the image, suggest that vertical eye movements are used to correct the relative positions of the two images as a whole. This is also supported by some observations in Duwaer and van den Brink (1981). Horizontal eye movements, by contrast, are used to search a wide area of displacements, bringing successive parts of the image into fusion individually. It was this observation that motivated the bounds on vertical net disparities used in my algorithm.

The second interesting experiment was reported by Mowforth, Mayhew, and Frisby (1981). They presented subjects with random-dot stereograms that had been high-pass filtered, at a range of disparities and tracked the subjects' eye movements. They found that stereograms filtered at 3.75 cycles/degree could initiate smooth eye movements resulting in fusion for (horizontal) disparities as high as ±56 minutes of arc, and that stereograms filtered at 7.0 cycles/degree could initiate fusion for disparities as high as ±28 minutes. Higher frequencies do not initiate movements resulting in successful fusion. Since small features become reliably visible to my edge detector when they are about 2 cells wide, this would translate into a search radius of about ±13 cells horizontally, at each scale. Notice that ±56 minutes of arc translates into over ±100 pixels of disparity. However, the entire stimulus was at this disparity, so this experiment cannot be used to test whether the bound on net horizontal disparities imposed by my program is reasonable.

Previous stereo algorithms have used a large range of constraints on search areas. The matching algorithm proposed by Marr and Poggio (1979) and implemented by Grimson (1981a,b) uses relatively small search areas. For a Marr-Hildreth operator with $w = 4$ cells,[5] this stereo algorithm searches a horizontal range of ± 4 cells at each scale. It appears to have had a limited ability to deal with vertical disparities in a multi-scale fashion, but the published reports claim

[5] Approximately the resolution of my edge finder.

that it handled vertical disparities at most ±3 at the finest scale. The same numbers seem to hold also for Grimson's more recent algorithm (Grimson 1985). Small search neighborhoods are crucial for these algorithms, because they have only limited ability to disambiguate rival matches at each scale.

Pollard, Mayhew, and Frisby (1985) assume the images are perfectly registered vertically and search ±30 cells horizontally, in a single-scale algorithm, using a Marr-Hildreth operator with $w = 4$ cells. The early algorithms described by Mayhew and Frisby (1980, 1981) were not developed in enough detail for search issues to be explored. The algorithm described by Drumheller and Poggio (1986), also used by Gillett (1988), searches a range of ±20 cells at only one scale, with Canny edges using $\sigma = 1.5$ cells.[6] This is approximately equivalent to a range of ±13.3 cells for my algorithm, because of the difference in edge finder scales.

Nishihara's (1984) correlation-based stereo algorithm uses a multi-scale algorithm to limit search. The correlation operation can find disparities within a ±2 cell range, both vertically and horizontally. An extremely limited amount of search is done at the coarsest scale and search at subsequent levels is only used near discontinuities. This system only produced disparities to relatively coarse resolution. Baker (1982; also Baker and Binford 1981) describes a multi-scale algorithm for matching edges. This algorithm assumes no vertical disparities. It appears to use coarse-to-fine matching to restrict search areas at each scale, but the details are unclear.

Search area limitations are less critical for stereo algorithms using sparse features. Medioni and Nevatia (1985), and Ayache and Faverjon (1987), have implemented stereo algorithms that match extended linear edge segments. Al-

[6] The constant for Canny's edge finder was supplied by Walter Gillett, personal communication.

though the details of their search areas are not specified precisely, they leave the impression that they are large. Barnard and Thompson (1980), Hannah (1980), Gennery (1977), and Moravec (1977, 1981) detect sparse local features that are easy to identify in the other image. Gennery and Moravec use a multi-scale matching strategy to identify points with both vertical and horizontal disparities, apparently using small search windows at each scale. Barnard and Thompson use a single scale algorithm and, as far as I can determine, search areas of ±15 cells in both the horizontal and vertical dimensions.

Compared to these previous systems, my stereo implementation uses relatively large search areas. Allowing for differences in edge finder resolution, the horizontal search area of ±10 cells is moderately large for any type of algorithm. Among algorithms that use multi-scale analysis, where small coarse-scale suggestions can translate into large fine-scale displacements, it is even larger. More importantly, my algorithm searches for vertical displacements as well as horizontal ones. The only previous algorithms that have done this have either used sparse features or coarse-resolution images. These vertical displacements cause a multiplicative increase in the search space and place correspondingly larger amounts of pressure on the evaluation of candidate matches.

The implemented matching algorithm can successfully handle large vertical displacements. Chapter 10 presents successfully fused images that have vertical disparities up to 16 cells and rotations up to 5 degrees. These images can also be fused by human observers, although some of them take noticably more effort than simpler stereograms. The exact amount of deviation that can be tolerated in an image depends on the scale at which reliable features appear, which depends on the size and contrast of regions in the scene.

5. Building the final disparity map

The last stage in stereo matching combines results from different alignments, by choosing the disparity at each cell which has the highest strength. A variant of the edge finder's noise suppression code is run over the resulting disparity map, to prune outliers and fill small gaps. This process is very similar to the directional combination step in the edge finder. Most of this computation is straightforward. Thus, this section deals almost entirely with niceties and special cases.

At each alignment, there are actually two sets of match results, because the matching process described in Chapter 5 is asymmetrical. One set of match results describes disparities from the perspective of the left image and one describes them from the perspective of the right image. The implemented stereo algorithm does two parallel computations, one starting from each of the two images. Information is passed between them at two points: once when the raw match maps are reconciled and once when suggestions from both images are used to determine the search area about each cell. Otherwise, they proceed independently. Because of the communication and the inherent similarity of the two tasks, the two computations typically return similar answers, but they are not guaranteed to be identical.

Reconciling the two final disparity maps may be desirable, but it is not clear how to accomplish it. Certain stereograms, such as the one illustrated in Figure 3, require a correspondence that is not bijective. People can fuse such examples and they can be handled by the current algorithm,[7] because it reconstructs two half-correspondences. Each half-correspondence must be a function for the algorithm to run properly, but it need not be bijective. When two patches in one image

[7] Chapter 10 shows this example and a similar example from a natural stereogram in more detail.

must be matched to one patch in the other image, one side of the computation can succeed in finding all of the matches, even though the other side of the computation cannot. Any algorithm for reconciling two correspondences must be able to handle such examples.

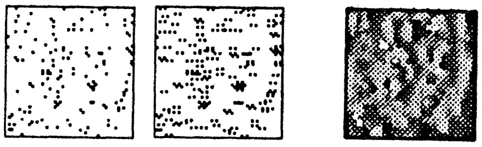


Figure 3. Panum's limiting case: stereo pair and computed disparities.

It is often proposed that two disparity maps from two halves of a stereo computation be fused into one common disparity map. It is unclear how to do this when there may be occlusion, so that certain surface patches are visible in one image, but not in the other. Such regions are subjectively visible in the field of view and are often perceived as having depth, perhaps extrapolated from the surface that they continue. Thus, these regions must be preserved in any fused stereo map. However, if it is to preserve the occluded regions from both images, a common coordinate system cannot be a flat piece of 2D space. Rather, it must be distorted and the distortions depend on the scene being viewed. Perhaps this could be handle using multiple disparity values at each 2D point, but no robust method for doing this has yet been proposed.

Another important point in combining disparity values is that the disparities computed by the matcher are relative to the alignment from which the matcher started. The alignment itself introduces disparities, which must be added to the computed disparities, yielding disparities relative to a global reference alignment between the two images. Since all alignments used by my implementation are translations, this computation is straightforward. Because images can be matched even from slightly incorrect alignments, the computed disparity range may be slightly larger than the search area for alignments.

Strength measurements are altered before combination, in order to bias the matcher in favor of disparities similar to those computed at the next coarser scale. Specifically, for each candidate match, the algorithm computes the distance d between the coarser-scale disparity and the disparity computed for the fine-scale match.[8] The matching strength is reduced by $2d$ units of strength. The strength bias has little effect where there is one match that is clearly better than the rest. It exists primarily to make the stereo algorithm behave reasonably in regions of uniform color or in other situations (e.g. stripes) where several equally good matches are possible. The bias allows the program to use the wider context of coarse-scale matching to influence a choice among fine scale alternatives of similar quality.

Previous algorithms have used constraints similar to the strength bias. For example, Grimson (1981a,b, 1985) requires matches at fine scales to differ from the coarse-scale value by no more than the matcher's search radius. In his algorithm, this search radius is ±4 cells horizontally. This form of the constraint becomes weaker as search neighborhoods are made larger. This is particularly a problem near depth discontinuities, where cells may generate candidate matches using suggestions from either surface. The bias formulation seems more helpful, because it can help disambiguate good matches, rather than just eliminating implausible ones.

Notice that my algorithm handles interpolating of disparity values in a different way from some previous algorithms, such as Grimson's (1981a). The matcher

[8] These distances are computed using a lookup table to avoid the square root calculation.

interpolates and smooths values separately for each alignment. The best disparity value at each cell is chosen using these dense maps. In Grimson's algorithm, the order of operations is reversed: the final disparity for each edge cell is chosen before disparities are interpolated and smoothed. Grimson's approach requires less computation, because interpolation and smoothing are done only once for the whole disparity map, rather than once for each alignment. However, it makes it more difficult to prevent interpolation and smoothing from crossing occluded regions or sharp changes in depth.

The final step in combination is to suppress noise in the final disparity maps. This noise suppression algorithm is adapted from that used by the edge finder. About each cell x, the maximal star-convex neighborhood (up to 3 cells in radius) is built using only cells whose disparity is within 1.5 cells of x's disparity. If the sum of the matching strengths in this neighborhood is below 3000, the cell is re-classified as not matching. At each disparity, sums are also computed for non-matching cells and they are assigned that disparity if their sum is above 3000. As in the edge finder, this process is repeated twice. This noise suppression cannot fix extended patches of incorrect match, but can only prune outliers and fill small gaps.

6. Disparity gradient constraints

A number of proposed stereo matching algorithms use *disparity gradient* constraints.[9] These constraints place bounds on the rate of change of disparity across the field of view, without taking image structure[10] into account. We see

[9] This term has become traditional. However, "slope" would be more appropriate than "gradient." The term "gradient" is typically used for a differential quantity and the disparity differences are taken between points that are substantial distances apart.

[10] I.e. locations of boundaries and image topology.

that disparity gradient constraints can also be expressed as requirements that disparities be nearly constant over a neighborhood of every cell, at every scale of analysis. The new stereo algorithm implemented for this thesis uses both this local constancy constraint and also the constraint that the stereo correspondence preserve topological structure, i.e. be continuous in both directions. These two types of constraints are independent and most previous stereo algorithms use only the first type of constraint.

The disparity gradient can be defined as follows. Suppose that C is a putative correspondence between two images, mapping the points a and b in one image onto $C(a)$ and $C(b)$ in the other. Suppose further that we align the two image planes so that a and $C(a)$ are at the same place and directions (e.g. up, right) are preserved. Then, the disparity gradient between the two pairs of points is the distance between b and $C(b)$, divided by the distance between a and the point halfway between b and $C(b)$, as shown in Figure 4.

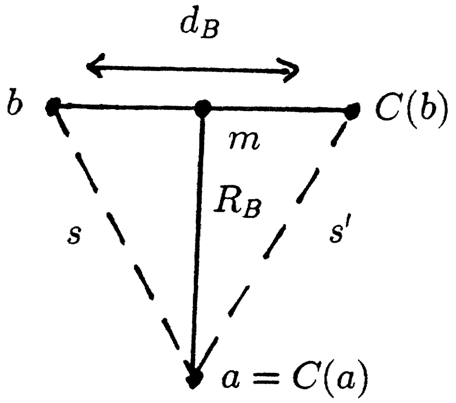


Figure 4. To compute the disparity gradient between two pairs of points $(a, C(a))$ and $(b, C(b))$, first line up the images so that a and $C(a)$ are at the same point and directions are preserved. Then, let m be the point halfway between b and $C(b)$. The disparity gradient is $d(b, C(b)) = d_b$ divided by $d(a, m) = R_b$.

The form in which I have defined the disparity gradient constraint differs slightly from that given by Burt and Julesz (1980a,b). First, I have extended the statement to cover possible vertical disparities. I have also specified one particular way of aligning the two images in order to compute the disparity gradient. Burt and Julesz do not pin down exactly how this should be done. Finally, translating the images so that a and $C(a)$ are at the same position makes the construction easier to understand without changing the computed gradients.

Burt and Julesz (1980a,b) introduced the disparity gradient in order to account for certain psychophysical data. Their stimuli were stereograms consisting of pairs of dots at varying spacings. When a match between two pairs of dots would exceed a disparity gradient of 1, human observers were not able to fuse both dots simultaneously. Thus, Burt and Julesz claim that human stereo fusion algorithms impose a bound of 1 on disparity gradients.

A difficulty with direct application of this disparity gradient constraint is that it predicts difficulties with stereograms containing sharp changes in depth between two surfaces. Sharp changes in depth do not, in general, cause problems for human observers and they are subjectively reconstructed as sharp. However, strict interpretation of the disparity gradient constraint would imply that either such stereograms should be impossible to fuse, or else some type of blurring across the boundary should take place. It is not clear to me that any existing proposals successfully account for both sharp depth boundaries and the data that Burt and Julesz present. More extensive psychophysical experiments would be needed in order to clarify what happens in human processing.

An effect close to that of the disparity gradient bound can be achieved by local constancy constraints used in my stereo algorith, as well as many others (such as Grimson 1981a,b, 1985, Marr and Poggio 1979, Drumheller and Poggio 1986).

These constraints specify that cells in a specified neighborhood of each cell x must have disparities within a bound d of the disparity at x. Section 7 discusses varying ways of defining neighborhoods. Suppose for the moment that the neighborhood specified for each cell x is a circle of radius r about x. Suppose further that this constraint holds about every cell in the image. Then, the disparity gradient between any two cells in the image is at most $\frac{d}{r} \pm d$. This could be construed as $\frac{d}{r}$ with some allowance for measurement errors. Thus, local constancy constraints and disparity gradient constraints can have much the same effect, for appropriate choices of r and d.

Most stereo algorithms impose some type of disparity gradient bound or local constancy condition. The main effect of these constraints is to prevent surfaces with steep slants (relative to the viewer) from being reconstructed. In my algorithm, this type of constraint is imposed by the requirement that each cell have a matching strength larger than the noise threshold. In order to accumulate enough support to exceed this threshold, a cell must belong to a large enough patch of image that matches at a single alignment. Since each alignment is a translation, this means that disparities must be close to constant in the neighborhood.

Disparity gradient bounds and local constancy requirements can be used in a number of ways. In my algorithm, the local constancy requirement implicit in the search through alignments is used to limit the region from which a cell can collect matching strength. This matching strength is used both to prune unacceptable matches and to rank acceptable ones. A number of other algorithms (Pollard, Mayhew, and Frisby 1985, Ayache and Faverjon 1987 Prazdny 1985, Hoff and Ahuja 1987) use these constraints in a similar way. In the algorithms described by Marr and Poggio (1979) and Grimson (1981a,b, 1985), strengths are computed in a similar manner, but they are only for pruning unacceptable matches, not for ranking acceptable ones. The final noise suppression step in my algorithm also uses strengths in this way.

There are two other ways in which disparity gradient bounds can be used. In the algorithms proposed by Medioni and Nevatia (1985) and Marr and Poggio (1976), an iterative optimization scheme is used to minimize the disparity gradients. The original Burt and Julesz formulation, followed by Drumheller and Poggio (1986), requires that all pairs of disparities in the image satisfy the disparity gradient bound. Thus, when a conflicting pair exists, one of the two matches must be removed. In Drumheller and Poggio's algorithm, this decision is made on the basis of matching strength. These strengths represent the number of matches at a similar disparity in a neighborhood of the cell.

In my model of topology, satisfaction of disparity gradient constraints is independent of whether the stereo correspondence is continuous. This illustrates an important point about the new model of boundaries proposed in this thesis. Since the presence or absence of boundaries changes the topological structure without changing distances, constraining the metric behavior of a function is not sufficient to guarantee that it is continuous. If space had no boundaries, a bound on the disparity gradient of less than 2 would imply that the stereo correspondence was continuous. But this implication does not hold when boundaries are present. This separation of metric from topological structure is unproblematic within standard mathematics. However, since it is an option that is rarely used in practice, I will discuss it in detail.

Suppose that the two images involved in a stereo correspondence C have no boundaries and suppose that the disparity gradient bound is $b < 2$. Consider the situation shown in Figure 4. The disparity gradient constraint requires that $d_B \leq b(R_B)$. But m is the midpoint of $[b, C(b)]$, so $R_B \leq \frac{1}{2}d_B + s$ by the triangle inequality. Thus $b(R_B) \leq \frac{b}{2}d_B + bs$, so $d_B \leq \frac{b}{2}d_B + bs$, which implies that $d_B \leq \frac{2bs}{2-b}$. But $s' \leq d_B + s$, again by the triangle inequality. Thus $s' \leq \frac{2bs}{2-b} + s$. If $b = 1$, this reduces to $s' \leq 3s$, which Pollard, Mayhew, and Frisby (1985) mention for the special case of points on the same epipolar line. Furthermore, for any $b < 2$, this relationship implies that $s' = d(C(a), C(b))$ must go to zero as $s = d(a, b)$ does, which shows that the correspondence C is continuous. This fact was originally proved by Trivedi and Lloyd (1985), but their proof is more complicated.

This construction, however, does not take the boundaries into account. If boundaries are added, according to either the open-edge or closed-edge models, a continuous correspondence must match boundary locations. As Figure 5 shows, a correspondence satisfying a disparity gradient bound of 1 can still fail to match boundary locations. Thus, using the new model of image boundaries, disparity gradient bounds and continuity conditions are totally independent.

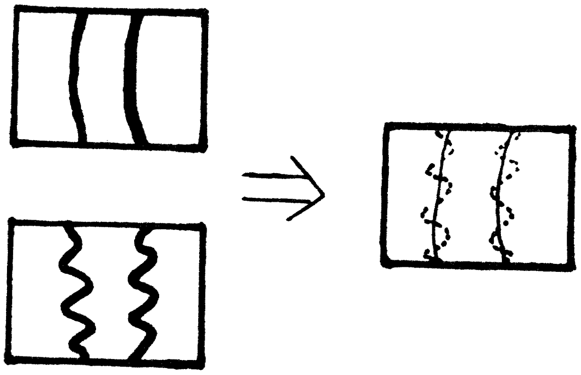


Figure 5. A translation of one image onto another has a disparity gradient that is uniformly zero. Thus, it satisfies any bound on disparity gradients. However, it does not preserve the topological structure of these two images, because it does not match boundaries with boundaries.

7. Support neighborhood shape

One crucial issue in implementing disparity gradient and local translation constraints is what shape support neighborhoods they are computed over. Differences in support neighborhood shape determine whether sharp changes in depth can be reconstructed and what forms they can take. In this section, I survey neighborhood shapes used by previous researchers and compare them to the star-convex neighborhoods used in my implementation.

Three types of support neighborhoods have been proposed:

- fixed-shape neighborhoods,
- ragged neighborhoods, and
- adaptive neighborhoods.

By "fixed-shape" neighborhoods, I mean that the shape of the support neighborhood about a point x is the same for every point x. By "ragged" neighborhoods, I mean computations using neighborhoods of fixed shape, but ignoring some of the points. Such a neighborhood might, for example, contain all nearby points at a similar depth. By "adaptive" neighborhoods, I mean ones in which the size of the neighborhood or its location relative to the point of interest can change, depending on local context, but with more constraint than the set of selected points used in ragged neighborhoods. These three options are illustrated in Figure 6.

Fixed-shape neighborhoods are the simplest formulation, used by Marr and Poggio (1979) and Grimson (1981a,b, 1985). In these algorithms, information is integrated over a fixed-shape neighborhood of each cell, typically a circular or square one centered on the cell. Alternatively, depending on how the constraints are used, each neighborhood of this form may be required to satisfy a disparity gradient or local constancy requirement. The problem with fixed-shape formu-

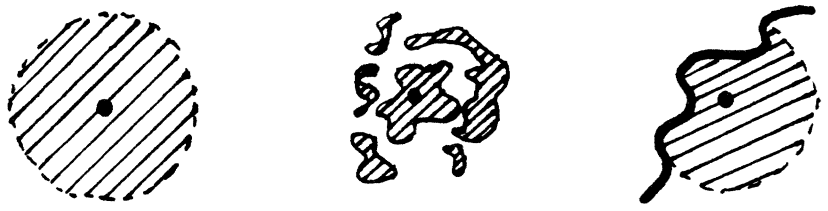


Figure 6. Left to right: fixed-shape neighborhoods, ragged neighborhoods, adaptive neighborhoods.

lations is that they cannot handle sharp changes in disparity, such as those at object edges. Depending on the details of the algorithm, cells near sharp changes are either assigned no disparity or else assigned a disparity that is an average of the disparities assigned to the two surfaces. Neither output is appropriate. Burt and Julesz's (1980a,b) global formulation of the disparity gradient is approximately equivalent to imposing the constraint over fixed-shape neighborhoods about every cell in the image and/or at multiple scales.

Some researchers (Grimson 1981a,b, 1985, Ponce and Brady 1985, Hildreth 1983) using fixed-shape algorithms have suggested a multi-stage method of coping with smearing of disparities across depth boundaries. They propose first computing the smeared disparities, then detecting sharp changes in the smeared output, and then recomputing the disparities using (effectively) adaptive neighborhoods. This seems to be shutting the barn door after the horse is gone. Recent research (Grimson and Pavlidis 1985, Marroquin 1984) has attempted to find better solutions, by earlier detection of sharp changes in disparities. However, there is still no robust implementation which avoids smearing.

Ragged neighborhoods, used by Pollard, Mayhew, and Frisby (1985), and Drumheller and Poggio (1986), are one solution to this problem with smearing. In these algorithms, fixed-shape neighborhoods are used, but information from only certain points is integrated to produce the final evaluation. In Drumheller and Poggio's algorithm, the selected points are those within some error neighborhood of the same disparity. The algorithm proposed by Pollard, Mayhew, and Frisby chooses those points whose disparities satisfy a disparity gradient bound of 1. In either case, the total amount of positive support is used to evaluate the match and negative evidence, such as nearby points at different disparities, is not considered. This allows points near depth discontinuities to receive acceptable evaluations, because they are no longer penalized by the presence of points near them but on the other surface.

The problem with ragged neighborhoods is that noise that happens to lie near a surface of similar value is given a high rating. This is illustrated in Figure 7. Such noise values might reflect poor matches found for cells that should correctly be marked as belonging to occluded regions, i.e. regions that are visible in only one of the images. Furthermore, using ragged neighborhoods undermines the power of disparity gradient or locally constant disparity constraints. Ragged neighborhoods of the form used by Pollard, Mayhew, and Frisby allow arbitrarily jagged surfaces, so long as enough nearby jags end up at similar disparities.

Adaptive neighborhoods are similar to ragged neighborhoods, but the set of points used for support is restricted so as to be connected or, in the case of my algorithm, star-convex. Several stereo-matching algorithms use adaptive neighborhoods, including my algorithm, Grimson's (1985), and the algorithm proposed by Ayache and Faverjon (1987) that matches linear segments. Like ragged neighborhoods, adaptive neighborhoods do not cross sharp changes in disparity. However, adaptive neighborhoods force reconstructed surfaces to be more cohesive and they avoid the problem of picking up nearby noise.

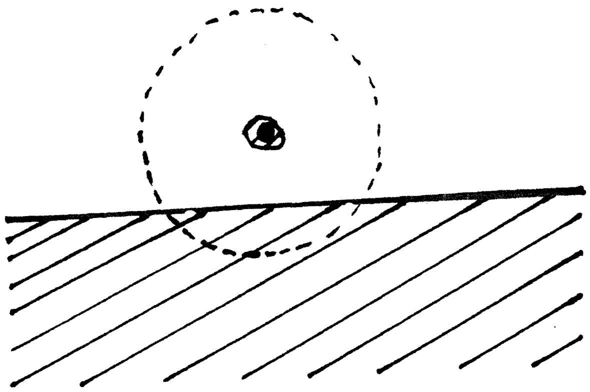
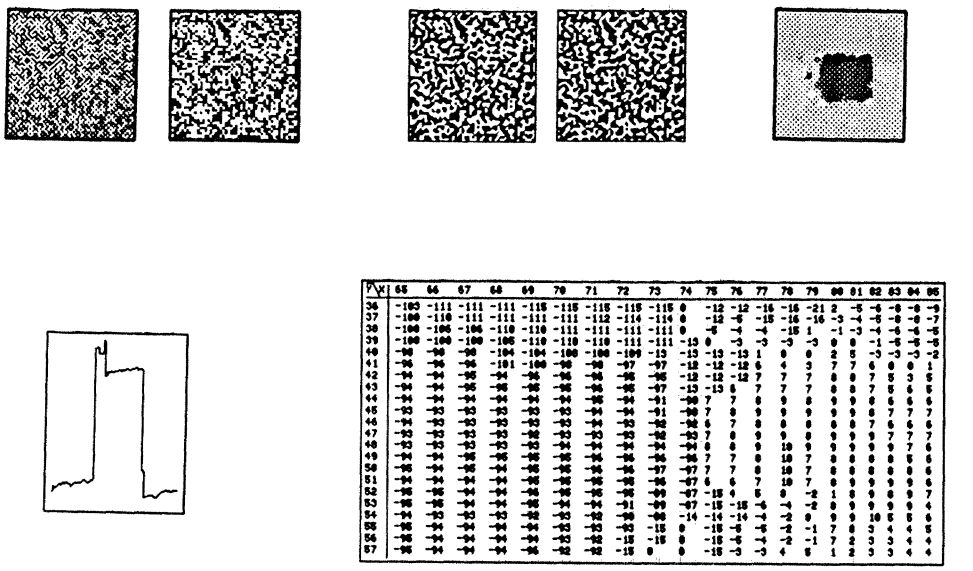


Figure 7. Ragged neighborhoods can cause noise to be given a high rating.

Because my stereo algorithm uses adaptive neighborhoods, it can reconstruct sharp changes in disparity values where appropriate. Figure 8 shows disparity values from one of the examples described in Chapter 10. Although there may be some slight curving of values near the discontinuity, there is a sharp change in disparity between the two surfaces, without intermediate values.

In implementing adaptive neighborhood algorithms, it is important to make sufficient allowance for errors in measuring disparities. If this is not done, these errors can cause large pieces of a support neighborhood to become disconnected from the cell of interest and the support from them lost. Allowance for noise is not so critical for ragged or fixed-shape neighborhoods, because they are not dependent on neighborhood connectivity. There seems to be some tendency for adaptive neighborhood algorithms to take explicit notice of the potential for errors and for other types of algorithms (e.g. Pollard, Mayhew, and Frisby 1985 and Drumheller and Poggio 1986) to require closer agreement with disparity gradient constraints than the precision of boundary locations warrants.

Substantially the same discussion presented here for stereo applies also to



y\x	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85
36	-103	-111	-111	-111	-115	-115	-115	-115	-115	0	-12	-12	-16	-16	-21	2	-5	-6	-8	-8	-9
37	-100	-110	-111	-111	-111	-111	-112	-114	-114	0	-12	-6	-15	-16	-16	-3	-4	-5	-8	-8	-7
38	-100	-106	-106	-110	-110	-111	-111	-111	-111	0	-5	-4	-4	-15	1	-1	-3	-4	-6	-6	-5
39	-100	-100	-100	-106	-110	-110	-110	-111	-111	-13	0	-3	-3	-3	-3	0	0	-1	-5	-5	-5
40	-98	-98	-98	-104	-104	-100	-100	-109	-13	-13	-13	-13	1	0	0	2	5	-3	-3	-3	-2
41	-96	-96	-96	-101	-100	-98	-98	-97	-97	-12	-12	-12	6	4	3	7	7	6	0	0	1
42	-94	-94	-95	-94	-96	-96	-96	-96	-95	-12	-12	-12	7	7	7	0	0	7	5	3	5
43	-94	-94	-95	-96	-96	-96	-96	-95	-97	-13	-13	6	7	7	7	0	8	7	5	6	5
44	-94	-94	-94	-94	-94	-94	-95	-94	-91	-90	7	7	8	9	8	9	9	8	6	6	6
45	-93	-93	-93	-93	-93	-93	-94	-94	-91	-90	7	8	9	9	9	9	9	8	7	7	7
46	-94	-93	-93	-93	-93	-93	-94	-93	-92	-92	6	7	8	8	8	8	8	7	6	6	6
47	-93	-93	-93	-93	-92	-93	-93	-93	-92	-93	7	8	9	9	8	9	9	9	7	7	7
48	-93	-93	-93	-93	-93	-94	-94	-94	-94	-94	8	8	9	10	9	9	9	9	9	7	6
49	-94	-94	-95	-95	-95	-95	-96	-96	-96	-96	7	7	8	10	7	8	8	8	8	5	6
50	-95	-94	-95	-94	-95	-95	-96	-96	-97	-97	7	7	8	10	7	8	8	8	8	8	6
51	-94	-94	-94	-94	-95	-96	-95	-95	-96	-97	6	6	7	10	7	8	9	9	9	9	6
52	-96	-96	-94	-94	-96	-96	-95	-95	-99	-97	-15	4	5	0	-2	1	8	9	8	9	7
53	-96	-96	-94	-94	-96	-94	-94	-91	-99	-97	-15	-15	-6	-4	-2	8	9	9	9	9	4
54	-94	-93	-93	-93	-92	-93	-92	-90	-98	-14	-14	-14	-4	-2	0	9	9	10	5	5	6
55	-96	-94	-94	-94	-94	-93	-93	-93	-15	0	-15	-5	-5	-2	-1	7	8	3	4	4	5
56	-96	-94	-94	-94	-94	-93	-92	-15	-15	0	-15	-5	-4	-2	-1	7	2	3	3	4	4
57	-96	-94	-94	-94	-96	-92	-92	-15	0	0	-15	-3	-3	4	5	1	2	3	3	4	4

Figure 8. Top: a stereo pair, edge finder output, and reconstructed depth map. Bottom: a slice across the raised square and numerical depth measurements for a portion of the image crossing the edge of the square.

other domains. Wide support neighborhoods are essential in motion analysis and texture analysis, including not only the periodicity and orientation examples presented earlier, but also other types of texture descriptions. In these domains, all currently available algorithms (e.g. Hildreth 1984, Horn and Schunck 1981, Heeger 1987) use fixed-shape neighborhoods and thus smear or contaminate results across sharp property changes. Scott (1986) proposes a method for detecting discontinuities in the blurred motion field resulting from his (fixed-shape neighborhood) algorithm, but it has not yet been extensively tested.

I believe that adaptive neighborhood algorithms, such as the ones used here for stereo analysis, might also be useful in these other domains. For example, in Chapter 5, Sections 5 and 7 we saw a brief example of how texture periodicity might be detected using a matching scheme. Figure 9 illustrates how a sharp change in periodicity would generate sharp changes in matching results. Hopefully, this sharp change would be preserved in periodicity estimates computed from the match results.

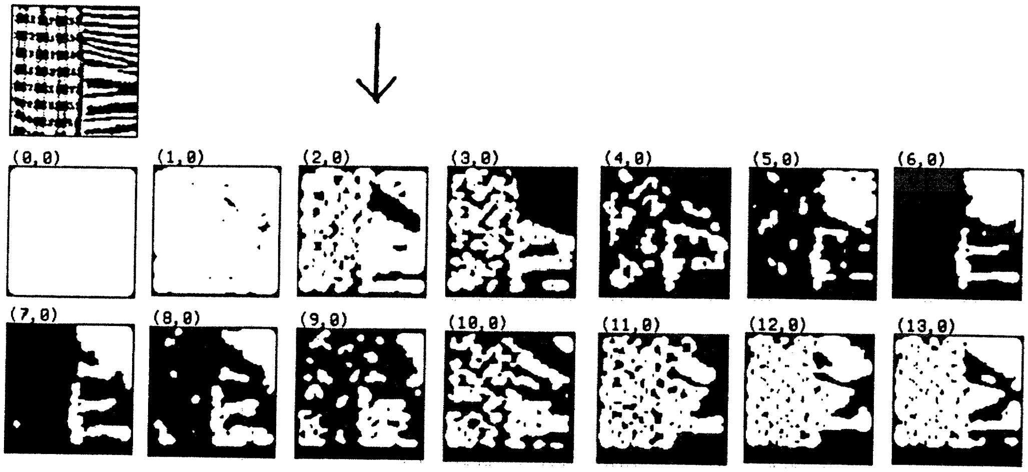


Figure 9. An image with a sharp change in periodicity generates sharp changes in the matching results. The image shown represents edge finder output for three cloth patterns and it was translated against itself vertically.

8. Ordering and uniqueness constraints

Stereo algorithms often impose an *ordering* constraint on stereo correspondences. This constraint states that a stereo correspondence must preserve the left-to-right ordering of points in the two images. In this section, we see some of the evidence for such a constraint and how it might be applied. It has often been observed that the disparity gradient constraint implies the ordering constraint.[11] However, we see that this is only strictly true when fixed-shape support neighborhoods are used and when no allowance is made for measurement error.

[11] For the traditional statement of the ordering constraint, involving only pairs of points on the same epipolar line, the proof is trivial.

Human observers seem not to be able to fuse stereo pairs if this would involve violations of the ordering constraint. If an order-preserving correspondence is possible, this is chosen, whether it reflects physical reality or not. For example, Krol and van de Grind (1980) report a series of experiments in which pairs of thin objects, such as nails, were systematically mis-fused in situations where this was the only order-preserving match, as illustrated in Figure 10. Two nails are seen, but they both lie at an apparent depth halfway between the two physically correct positions. If the only available correspondence would violate the constraint (e.g. because the objects are too distinct to support a mis-match interpretation), one or another of the regions involved is seen as doubled.

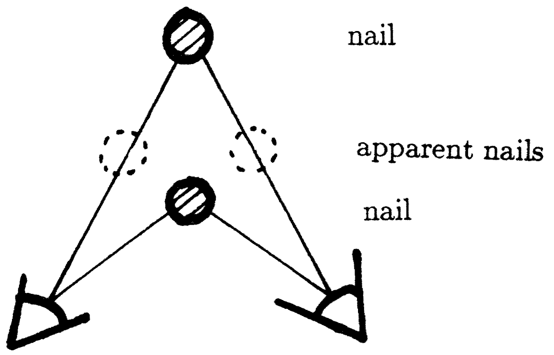


Figure 10. The physical layout for Krol and van de Grind's double nail illusion. Human observers do not see nails at the physically correct locations, but rather at an intermediate depth. This suggests that human stereo matching tends to preserve the relative order of regions in the images.

Thus, there are two phenomena to consider: the bias towards order-preserving correspondences and the inability to fuse violations of the ordering constraint. A bias towards order-preserving correspondences, such as would be needed to explain the Krol and van de Grind data, is easy to account for using most models of stereo matching. For example, consider fusing two thin nails using my algorithm. Thin regions persist to quite coarse scales if they have sufficient contrast. At a coarse enough scale, the two nails are closer together than the width of the support neighborhoods used in the matching strength computation. When this is true, the alignment generating the physically incorrect fusion generates matches with higher strength than either of the alignments representing the physically correct fusions. Thus, the physically incorrect fusion would be chosen at this coarser scale and this would bias finer scales towards a similar interpretation.

Additional machinery, however, is probably required to account for the inability of humans to fuse violations of the ordering constraint when these represent the only possible matches. An important point to note about this constraint is that it is a constraint on what can be fused *at one fixation.* For example, suppose I attempt to duplicate Krol and van de Grind's conditions with two objects that are not very similar to one another, such as two bent paperclips. My impression from informal experimentation is that one of the objects is fused and the other doubled and that I have voluntary control of which object is fused. This seems to agree with some of the observations in Krol and van de Grind's paper as well.

Recent stereo algorithms proposed by Drumheller and Poggio (1986) and Baker (1982; also Baker and Binford 1981) have imposed ordering constraints directly during stereo matching. When two candidate matches would violate the ordering constraint, the weaker match is eliminated. Since their systems use few other methods of constraining candidate matches, the ordering constraint apparently helps prune many unacceptable matches. Many of the incorrect matches generated by my system seem to represent violations of the ordering constraint, performance might be improved by explicitly enforcing it.

However, the formulations used by Baker and by Drumheller and Poggio will deadlock or become unstable when confronted with two matches of similar strength. Such examples occur when textures contain repeats of similar elements. For example, Chapter 10, Section 4 shows an example of a natural image that contains two very similar patches, one of which is occluded in the other image. This example generates two matches that violate the ordering constraint because they violate uniqueness. However, the two matches have similar evaluations and both appear good to informal inspection.

Matches of similar strength could be handled in two ways. Since the constraint on human processing seems to be imposed at a single fixation, one might bias strengths in favor of the match closest to the fixation point. Alternatively, the algorithm might keep both matches when they are of comparable strength, despite the ordering violation. This would also prevent the algorithm from suppressing correct matches at coarser scales, where they are not always easy to distinguish from incorrect matches.

I did not use an ordering constraint in the final implementation of my stereo matcher. Since my primary interest was in testing the power of the topological matching constraints, an additional matching constraint would have made it harder to interpret the experimental results. Furthermore, my control structure folds together the processes of searching Panum's area and moving eyes over a range of fixations, as an optimization. Since individual fixations are not distinguished, implementing a constraint tied to single fixations would be difficult.

The disparity gradient constraint, in the form originally proposed by Burt and Julesz, implies the ordering constraint. This result is widely cited and easy to prove. Unfortunately, it only holds if fixed-shape neighborhoods are used in enforcing the constraint. Adaptive neighborhood or ragged neighborhood formulations of disparity gradient constraints do not enforce the ordering constraint, though they may tend to reduce the number of places at which it is violated locally. In particular, neither my implementation nor that of Pollard, Mayhew, and Frisby (1985) enforces the ordering constraint.

In imposing disparity gradient or ordering constraints, it is important to make appropriate allowances for errors in the computed disparities. If disparities are required to meet such requirements exactly, as in many stereo algorithms, points may be unnecessarily deprived of legitimate support due to image noise. This is particularly true when fixed-shape or adaptive support neighborhoods are used. For example, the current MIT implementation of the Drumheller and Poggio (1986) algorithm makes no allowance for measurement errors in imposing its ordering constraint. To judge from the results reported by Gillett (1988), this algorithm frequently assigns no match to thin strips of cells in regions of smoothly changing disparity, presumably because they are suppressed by matches from nearby points. To avoid this, matches should be suppressed only if they are in violation of the ordering constraint by more than the prevailing measurement errors.

An often-cited corollary of an ordering constraint is that it forces the stereo correspondence to be bijective. This result will not hold for cells near one another if allowance is made for measurement errors. Furthermore, there is some question as to whether the ordering constraint is imposed on each half-correspondence separately or on some type of unified correspondence? For human perception, the correct answer depends on exactly how humans perceive examples such as Panum's limiting case.[12] Previous authors seem to disagree as to whether such examples involve one fixation or more than one fixation and I have seen no definitive psychophysical evidence on this question.

[12] Shown in Figure 3.

9. Conclusions

In this chapter, we have seen how to build a stereo matching algorithm using the image matcher described in Chapter 5. The new stereo algorithm has been implemented and results from this implementation are presented and discussed in Chapter 10. A control structure similar to that used in stereo matching could also be used for motion analysis. Chapter 10 presents a brief example illustrating how this might be done.

The results presented in Chapter 10 show that the new stereo algorithm can match both natural and synthetic images robustly. There are two important improvements over previous proposals. First, where there are abrupt changes in disparities, the algorithm reconstructs disparities near the boundaries without blurring. This is because the adaptive support neighborhoods, implemented via a star-convexity requirement, do not cross abrupt changes in depth.

Secondly, the requirement that topological structure be preserved in matching allows a more robust evaluation of whether two patches of image match and, if so, how well. This allows the algorithm to search larger numbers of alignments for matches, without becoming confused. In particular, the new algorithm can handle vertical displacements between stereo images, which previous algorithms of similar type have not been able to do. It has successfully fused images with vertical disparities of up to 16 cells and rotation of up to 5 degrees.

The current stereo implementation is quite slow, taking multiple days to run test images such as those shown in Chapter 10. This is due to a number of factors. First, it is a parallel algorithm running on a serial machine. Secondly, the implementation was made modular, so that even such items as the method of inducing boundaries from edge finder labels could be altered during experimentation. Finally, the star-convex sum operation is quite slow. I am still experimenting with different ways to code this operation, including both varying the paths used and varying the way in which the computation is done. Since star-convex sum accounts for a major portion of the work in both the edge finder and the stereo computations, optimizing its speed is an important issue for the line of research proposed in this thesis.

Chapter 7: Natural Language Semantics

1. Introduction

In the previous three chapters, we have seen how boundaries and topological properties can be useful in low-level visual processing. Similar phenomena occur also in other domains. For example, in visual analysis, we added boundaries to 2D space to model sharp changes in intensity. In a similar way, we can add boundaries to 3D space in order to model sharp changes in material and we can add boundaries to time in order to model sharp changes in what events are taking place. In this chapter and the next one, we see how topological properties reappear in domains other than vision. This chapter discusses data from natural language semantics. Data from high-level reasoning domains will be discussed in Chapter 8.

In Chapter 3, we saw that sentences in English can be divided into those that describe states and those that describe actions. Actions can be further subdivided into activities, accomplishments, and state changes. Which class of situation a sentence describes is important in determining what types of verb forms (tense and aspect) it can appear in, what types of temporal adverbs can occur in it, and how it can be related to other sentences using temporal connectives such as "when" and "until." The differences between different classes of situations can be characterized in terms of differences in topological structure. We also saw that nouns seem to fall into similar classes.

There are several reasons for presenting this linguistic data. The linguistic phenomena are closely related to phenomena we will see later in the context of high-level reasoning. Thus, the linguistic data provides additional context and support for the analyses used in high-level reasoning. Describing the linguistic data using cellular topology allows linguistic semantics to share a common formalism with other areas of artificial intelligence and it widens the coverage of cellular topology. As I mentioned in Chapter 3, topological connectedness may be useful in explaining the meaning of several constructions. Furthermore, the new analysis avoids several technical problems encountered by previous researchers.

Sections 2-5 discuss the classification of situations (states and actions), models for situations of different classes, syntactic and semantic tests for establishing class, and how the class distinctions interact with tense and aspect distinctions in English. We see that topological connectedness may be useful in explaining the meaning of the perfect and progressive aspects. Section 6 discusses a similar classification for nouns and noun phrases.

In Sections 7-11, I discuss certain representational issues in detail. These include constraints on the form of time and intervals over which actions occur (Section 7), use of textures, scale, and support neighborhoods (Section 8), methods of modelling the distinction between states and actions (Section 9), methods of modelling abrupt state changes (Section 10), and the relationship between spatial boundedness of direct objects and temporal boundedness of verb phrases (Section 11). Finally, Section 12 discusses the meanings of two temporal connectives and how they interact with the topological models of different classes of situations.

2. Classes of situations

A four-way classification of situations has been used for some time in linguistic semantics. This classification, due originally to Vendler (1967, chapter 4), distinguishes four types of situations:

Activities: run, swim, drive a car

Accomplishments: build a house, do a problem set

States: be green, like mathematics, own a car

State changes:[1] reach the finish line, find a free terminal

Similar classifications have been used by Ryle (1949), Taylor (1977), Dowty (1979, chapter 2), Mourelatos (1981), and Allen (1984).[2] Apparently the general idea can be traced back to Aristotle. I use the term "actions" as a cover term for activities, accomplishments, and state changes.

As Dowty (1979) discusses, this traditional classification subsumes several types of distinctions. These include differences in temporal structure, agentiveness or lack of it, and a distinction between predicates describing static situations and those describing situations in which something changes over time. Conflating these distinctions makes it difficult to establish a consistent set of defining properties for the four classes. Because of this, I use the four traditional classes of situations only to refer to differences in temporal structure. Appendix C discusses two other types of distinctions, which I consider orthogonal to the issues discussed here.

[1] I am avoiding the traditional term "achievement" because it is too easily confused with "accomplishment."

[2] Allen's work seems to fall somewhere between linguistic semantics and the types of practical reasoning discussed in Chapter 8.

I model time using a cell complex whose underlying space is the real number line $\mathbb{R}$.[3] This is illustrated in Figure 1. Time is distinguished from the other spaces we have seen in this thesis: there is an *order* defined on it. This order is essential to distinguishing the past from the future, both for linguistic semantics and for the practical reasoning that we will see in Chapter 8.

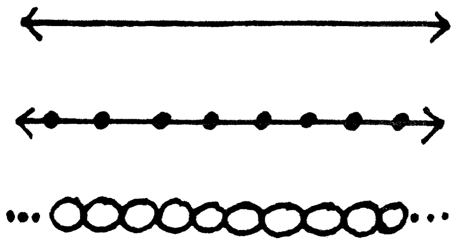


Figure 1. Time is represented as a cell complex whose underlying space is $\mathbb{R}$ (top). Two graphic representations of this cell complex are shown (middle and bottom).

Using this cellular model of time, I model the temporal structure of the four types of situations as shown in Figure 2. A state describes the properties of the world at a single cell in time. State changes describe an abrupt change in properties, with associated boundary in time, between two cells. Activities describe the contents of a connected interval of time containing at least two cells. Accomplishments describe the contents of a connected interval of time containing at least two cells (like an activity), together with a state change at the end of this interval and perhaps at the beginning of the interval as well.

A state or activity description mentions only a small interval of time, but it can also be used to describe longer intervals if each sub-interval of appropriate size fits the description. So, for example, a description of "falling" might specify

[3] This assumption can be relaxed to allow limited types of branching time. See Section 7.

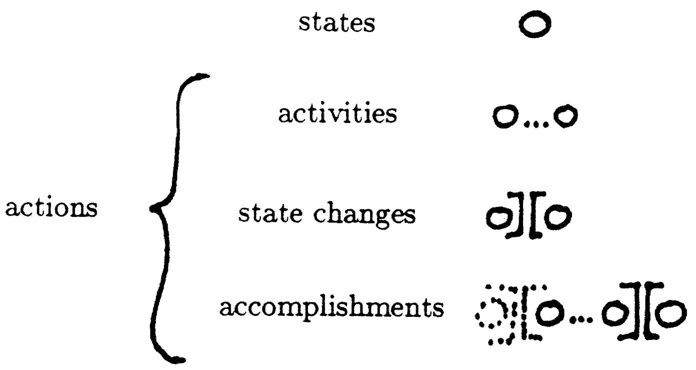


Figure 2. Cellular topology models for the four types of situations. States describe the contents of a single cell and actions describe the contents of an interval. These models specify the connected intervals over which situations occur, the locations of boundaries between these intervals, and whether each interval consists of only one cell, one or more cells, or at least two cells.

a change in height between two cells. This could be applied not only to a two-cell interval, but also to a longer interval in which every adjacent pair of cells displays a change in height. This is illustrated in Figure 3.

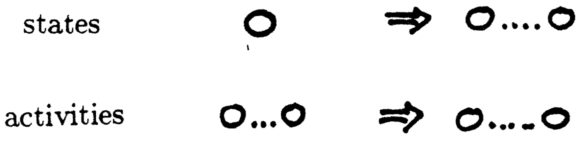


Figure 3. States and activities describe short intervals of time. They can also describe longer intervals in which each sub-interval of appropriate size meets the description.

It is also possible to provide pointwise versions of these situation models, using the closed-edge model of boundaries. These are shown in Figure 4. They differ from the cellwise models in representing state changes as consisting of two points, rather than two cells. Because a reasoner must understand the world via finite-precision measurements, I use the cellwise models. However, the discussion in this chapter can easily be modified to use the pointwise models.

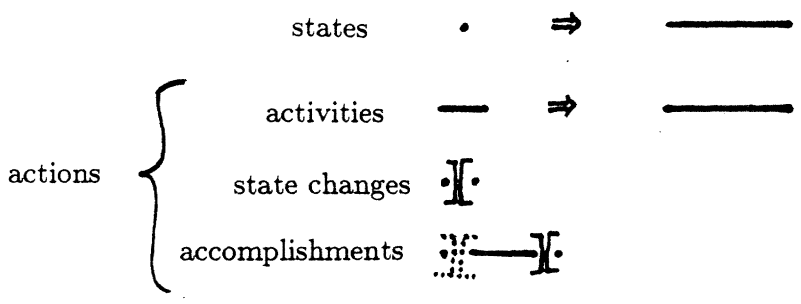


Figure 4. Point-wise alternative models of situations. These are similar to the models shown in Figure 3, but use single points in place of single cells. These models require the closed-edge model of boundaries.

These models refer both to topological properties and to the cell structure of space. However, the reference to cell structure is confined to distinguishing single cells from intervals containing more than one cell. The models of activities, for example, does not specify how many cells must belong to its interval. I do not make any claims about how these models should be represented in a language understanding system. One might, for example, use an abstract interval representation such as that described by Allen (1983, 1984). In this thesis, I am concerned only with the temporal models underlying whatever representations are used.

The models of states and activities are vague as to whether their interval ends in a boundary. In building up a model of events from an English language description, other considerations often force such a boundary to be constructed. For example, in Sentence 1, the direct object forces[4] the "eating" activity to end after a finite period of time. By itself, "eating" does not specify such a boundary.

[4] For details, see Section 11.

(1) The leopard ate a young zebra.

Thus, the four types of situations can be used to describe not only bare verbs, such as "eating," but also verb phrases, such as "eat a young zebra," and perhaps other types of constituents. For describing the meaning of more complex linguistic constituents, a four-way taxonomy of situation types is unlikely to be sufficient. I assume that they can be extended to a more flexible description language as the data becomes better understood. The first step in understanding the data well enough to design such a language is to build concrete models for the meaning of particular examples.

There are a number of standard tests for which of the four classes a verb or verb phrase belongs to. The most compact summary is given by Dowty (1979, pp. 55-60 and p. 184). Most authors use similar sets of tests. A revised list of tests is summarized in the following table:

Test	State	Activity	Achievement	State Change
For an hour	ok	ok	bad	bad
In an hour	bad	bad	ok	ok
Take an hour	bad	bad	ok	ok
At 3:00	ok	bad	sometimes	ok
Stop	ok	ok	ok	bad
Finish	bad	bad	ok	bad
Almost	one	one	two	one
Present	ok	bad	bad	bad
Progressive	bad	ok	ok	sometimes
Perfect/Past	no end	end?	end	end
X-ing ⇒ has X-ed	n/a	yes	no	no or n/a

These tests distinguish the four types of situations from one another. I have left out some of Dowty's tests that seemed difficult to apply or that gave unclear results.

In the next several sections, we see these tests in detail and I show how one might explain the differences in behavior in terms of the cellular topology models of situations. Section 3 discusses those differences in behavior that do not involve tense and aspect distinctions. The meaning of different tenses, and differences in behavior related to them, are discussed in Section 4. Finally, Section 5 discusses the meaning of different aspects.

Sorting out examples of different situation classes is complicated because a phrase that normally describes a situation of one class can sometimes be interpreted as describing some other type of situation. I refer to these reinterpretations as *coercions*, on analogy with the term from programming language design. Coercions frequently have some change in meaning associated with them, such as interpreting an action as being repeated multiple times, but require no overt morphological change. Appendix D describes some common types of coercions in detail.

3. Miscellaneous tests for verb class

In this section, I describe a number of tests for which type of situation a verb phrase describes. These tests include restrictions on the types of adverbs that can modify verb phrases of different types, as well as restrictions on what types of verb phrases can be combined with "stop" and "finish." The behavior of each type of situation in these tests can be accounted for using the cellular models of situation types. In Sections 4-5, we see other differences in behavior, involving tense and aspect distinctions.

Various temporal adverbs place constraints on the types of situations they can modify. For example, prepositional phrases headed by "for" expect the situation to fill up the specified interval of time. Thus, situations that impose their own natural endpoints are unacceptable, as are instantaneous state changes:

(2) Your cat was in the kitchen for twenty minutes.

(3) The aide shredded incriminating documents for hours.

(4) #Eric made a fresh pot of coffee for ten minutes.[5]

(5) #Bonnie passed her area exam for a few hours.

Conversely, prepositional phrases headed by "in" require that the action impose its own endpoint and thus states and activities are unacceptable:

(6) #Your cat was in the kitchen in twenty minutes.

(7) #The aide shredded incriminating documents in hours.

(8) Eric made a fresh pot of coffee in ten minutes.

(9) Bonnie passed her area exam in a few hours.

Sentences 6-7 can be made acceptable if the verb phrase can be construed as referring to the start of the state or activity (so-called "inceptive" readings).[6] The construction "take an hour to X" behaves similarly.

Prepositional phrases indicating very short amounts of time are unacceptable with activities or accomplishments, if the action could not plausibly unfold in that short an amount of time. The details vary with the action being discussed. However, states and state changes seem to be acceptable, no matter how short an amount of time the phrase picks out:

[5] As mentioned in Chapter 3, I mark sentences with a hash mark (#) to indicate that they are unacceptable, without making any claims as to whether the problems are best regarded as syntactic, semantic, or pragmatic.

[6] See Appendix D for further discussion of this type of re-interpretation.

(10) Your cat was in the kitchen at 3:00.

(11) #The aide shredded incriminating documents at 3:00.

(12) ?Eric made a fresh pot of coffee at 3:00.

(13) #The Simpsons built a house at 3:00.

(14) Bonnie passed her area exam at 3:00.

Again, activities may be acceptable if they can be coerced into an inceptive reading. Since the length of time depends so much on the context, this test is difficult to apply. I have added it to the standard list because it is useful in illustrating how these types of situations differ.

Words like "start" and "stop" are used to refer to the endpoints of an action or state that happens over a non-trivial period of time. State changes cannot occur as the complement of any of these words, as illustrated in Sentences 15-18:

(15) Your cat stopped being in the kitchen.

(16) The aide stopped shredding incriminating documents.

(17) Eric stopped making a fresh pot of coffee.

(18) #Bonnie stopped passing her area exam.

The unacceptability of state changes in such constructions may be due to the fact that they last for only trivial amounts of time, the fact that they occur only with difficulty in the progressive (see Section 5), and/or the fact that this construction is used to create references to state changes and would thus be redundant applied to a word that already refers to a state change.

When the word "stop" is used with an accomplishment, as in Sentence 17, it indicates that the action was interrupted prior to reaching its natural endpoint. The word "finish" is used when the action halts due to reaching its natural endpoint. This restricts its complement to being an accomplishment:

(19) #Your cat finished being in the kitchen.

(20) #The aide finished shredding incriminating documents.

(21) Eric finished making a fresh pot of coffee.

(22) #Bonnie finished passing her area exam.

Sentences 19-20 can become acceptable if some natural endpoint to the situation can be imagined, e.g. if the aide stopped because his shift ended. Sentences like 22 and 18 can become acceptable if the state change is viewed as lasting for some non-trivial amount of time or as being iterated many times.

Dowty notes another interesting test for achievements: they lead to two readings in sentences containing the adverb "almost." Consider Sentences 23-26:

(23) Your cat was almost in the kitchen.

(24) The aide almost shredded incriminating documents.

(25) Eric almost made a fresh pot of coffee.

(26) Bonnie almost passed her area exam.

Unlike the other sentences, Sentence 25 has two readings. Eric might not have started the action at all or he might have started the action but never finished it.

4. Tense

In Section 3, we saw that the type of a situation described by a verb phrase can affect what types of modifiers can be added to it. Situations of different classes also interact differently with tense and aspect distinctions. In this section, I describe how to model the meaning of tense distinctions in English. We see that the present tense is confined to descriptions of states.

A sentence in English can appear in three different tense forms: past, present, and future. These are illustrated by Sentences 27-29:

(27) Bruce loved mathematics.

(28) Bruce loves mathematics.

(29) Bruce will love mathematics.

The meaning of different tenses in English can be described in terms of the temporal relationship between a *reference interval* during which the action or state takes place and the moment of speech. A sentence in the past tense describes a situation in which the reference interval lies entirely before the moment of speech, it is entirely after the moment of speech in a future tense sentence, and it lies entirely in the moment of speech in a present tense sentence. These possibilities are illustrated in Figure 5.

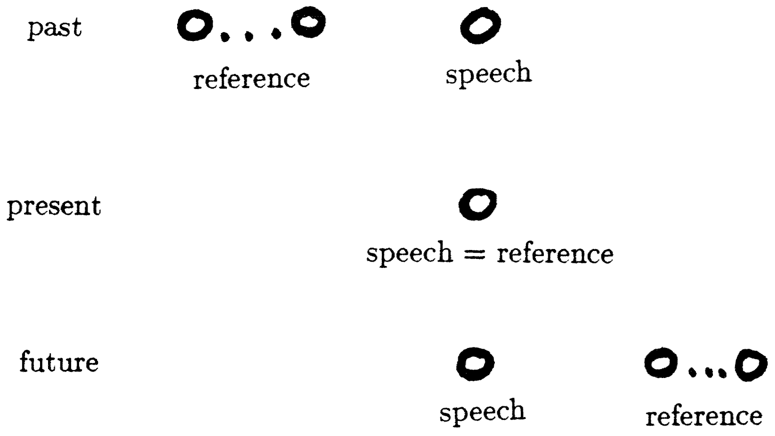


Figure 5. The relationship between the reference interval and the moment of speech in a past tense sentence (top), a present tense sentence (middle), and a future tense sentence (bottom).

The interesting feature of this model of the English tense system, proposed by Woisetschlaeger (1976) is that the moment of speech is conceived of as very small. He models it as a single point. Within cellular topology, it is more appropriate to use a single cell. Thus, the model of time is as shown in Figure 6. This places a severe constraint on what can occur in the present tense. Because the reference interval must lie entirely within the moment of speech, it contains at most a single cell. Since the situation must occur during the reference interval only states can occur in the present tense, because only states occur over a single point.

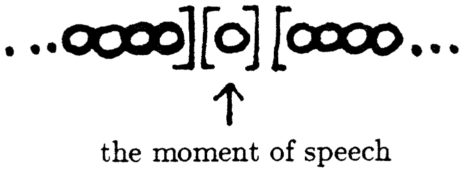


Figure 6. The model of time used in the English tense system.

The prediction that only states occur in the present tense seems to be born out by the data. Consider Sentences 30-37:

(30) Your cat was in the kitchen.

(31) Your cat will be in the kitchen.

(32) The aide shredded incriminating documents.

(33) The aide will shred incriminating documents.

(34) Eric made a fresh pot of coffee.

(35) Eric will make a fresh pot of coffee.

(36) Bonnie passed her area exam.

(37) Bonnie will pass her area exam.

Sentences 30-31 describe a state, Sentences 32-33 describe an activity, Sentences 34-35 describe an accomplishment, and Sentences 36-37 describe a state change. All four types of situations occur in both past and future tense forms.

In the present tense, however, sentences describing actions are unacceptable, as illustrated by Sentences 38-41:

(38) Your cat is in the kitchen.

(39) #The aide shreds incriminating documents.

(40) #Eric makes a fresh pot of coffee.

(41) #Bonnie passes her area exam.

More exactly, these sentences are unacceptable as descriptions of on-going events. They are acceptable only if they are given a *habitual* reading. Such a reading can be forced by adding appropriate adverbs, as in Sentence 42.

(42) Eric makes a fresh pot of coffee every day.

As I describe briefly in Appendix D, Woisetschlaeger (1976) argues that habitual readings describe the *structure* of the world at a particular time, rather than what is happening in the world during that time. These structual descriptions are states.

States also differ from actions in the implications of a past tense form. A past tense sentence describing a state, such as Sentence 43, is neutral as to whether the state continues into the present.

(43) Shimon was in Cambridge last month.

In this case, the reference interval is still located entirely before the present moment. However, saying that the contents of the reference interval match the description of some state does not imply that it is the *maximal* interval that matches it.

The models for accomplishments and state changes, however, specifies a natural end to the action. If the contents of an interval match such a model, then the action must have come to an end during that interval. Thus, past tense sentences describing accomplishments and state changes imply that the entire action occured before the present moment. This is illustrated by Sentences 44-45:

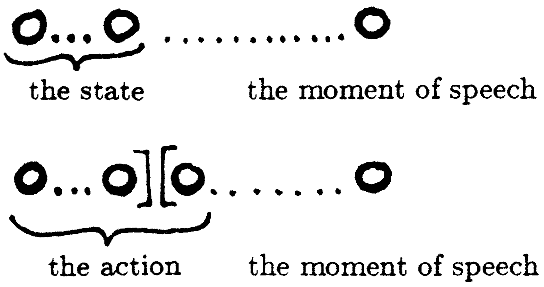


Figure 7. The past tense form of a state description (top) does not imply that the state has ended before the present moment, because it does not specify a boundary at which the state must end. The past tense form of an accomplishment (bottom) ends in a boundary. Thus, the accomplishment cannot continue into the present.

(44) Mike made a pot of tea.

(45) George turned the light on.

Sentence 44 implies that the whole action of making the tea precedes the current moment, including at least some period in the past when the finished pot of tea existed. Similarly, Sentence 45 implies that there was a period of time before the moment of speech when the light was on.

Activities are more problematic. If the sentence describes an activity, the reference interval must contain a recognizable sample of the activity. The model of activities suggests that past tense sentences describing activities, like those describing states, should be able to continue into the present. There does seem to be a contrast between activities and accomplishments. So, for example, Sentence 46 seems acceptable, whereas Sentence 47 is clearly bad.

(46) Eric lectured all morning and, for all I know, he may still be lecturing.

(47) #Mike made a pot of tea and, for all I know, he is still making it.

However, there seems to be some implication that an activity must have ended if it is described using a past tense form. Sentence 48, for example, seems to imply that Anita has stopped running.

(48) Anita ran along the river.

In this section, we have seen how to model the meaning of tense distinctions in English, using cellular topology. We have also seen how the model for different types of situations can be used to predict which tense forms are possible and what they mean. In the next section, we see how this system can be extended to perfect and progressive aspect forms. This model for English tenses is put together from the discussion in Dowty (1979), Woisetschlaeger (1976), and Johnson (1981). It seems to be consistent with their analyses, but I have had to do some extrapolation to produce a unified description. Comrie (1985) gives a good pre-theoretical description of tense phenomena across different languages.

5. Aspect

Sentences in English vary not only in tense, but also in *aspect.* English has four different aspect forms for each tense. For example, Sentence 49 has unmarked aspect, Sentence 50 is progressive, Sentence 51 is perfect, and Sentence 52 is both perfect and progressive.[7]

(49) Dan wrote his thesis.

(50) Dan was writing his thesis.

(51) Dan had written his thesis.

(52) Dan had been writing his thesis.

[7] More precisely, the two aspects are composed to form Sentence 52. The progressive aspect is applied first, i.e. more tightly bound to the verb. The other possible order of composition is forbidden as a side-effect of a general constraint that progressives cannot be formed from states, described later in this section.

In this section, we see how to describe the meaning of perfect and progressive forms. We also see that these constructions provide further tests for distinguishing different classes of situations. We also see a potential use for topological connectivity in describing the meaning of perfect forms.

The progressive aspect creates descriptions of states from descriptions of actions. This state is true of cells belonging to an interval over which the action occurs. Compare Sentences 53 and 54:

(53) David read Koenderink's new book yesterday afternoon.

(54) David was reading Koenderink's new book yesterday afternoon.

The verb phrase "read Koenderink's new book" in Sentence 53 refers to the entire action of reading the book. Thus, Sentence 53 implies that David finished the whole book during the period specified by the phrase "yesterday afternoon." The verb phrase "was reading Koenderink's new book," however, refers to only some cell or cells during the period over which the action "read Koenderink's new book" occured. Thus, Sentence 54, states that David read some of the book, but perhaps not the whole thing, during "yesterday afternoon." This description of the progressive has been proposed, with slight variations, by Taylor (1977), Dowty (1979), Woisetschlaeger (1976), and Bennett and Partee (1978).

Because progressive forms describe states, they can occur in the present tense and can take adverbials that refer to very short intervals of time. Thus, Sentences 55-56 are acceptable, even though Sentences 57-58 are not.

(55) David is reading Koenderink's new book.

(56) David was reading Koenderink's new book at 3:00.

(57) #David reads Koenderink's new book.

(58) #David read Koenderink's new book at 3:00.

Conversely, progressives cannot be made from states, because such a form would mean exactly the same thing as the original state. So, for example, Sentences 59 and 60 are unacceptable.

(59) #Bruce is loving mathematics.

(60) #Ian is being by the coffee machine.

Progressives can not, however, refer to any cell in the middle of a situation. The progressive can only be used to describe a cell that belongs to a *connected interval of at least two cells* during the situation. Figure 8 shows which parts of a situation the progressive form can refer to, for each of the four types of situations. This restriction forces progressives of accomplishments to refer only to cells within the period before the final state change. For example, Sentence 61 is not acceptable if Mitch has finished writing the book, even recently.

(61) Mitch is writing a book.

Such a restriction is needed in my model, as in Johnson's (1981) model of accomplishments, because the model of an accomplishment includes some points after the state change. Other authors (e.g. Allen 1984) have proposed models of achievements that only include points in the main activity part of the achievement. These systems, however, have difficulty modelling state changes.

This restriction on progressive forms prohibits progressives of sharp changes. So, for example, Sentence 62 is not acceptable.

(62) #Bonnie is passing her area exam.

(63) King Hamelbar was dying.

In some cases, as in Sentence 63, the state change can be interpreted as taking an extended amount of time, as illustrated in Figure 9. In these cases, a progressive

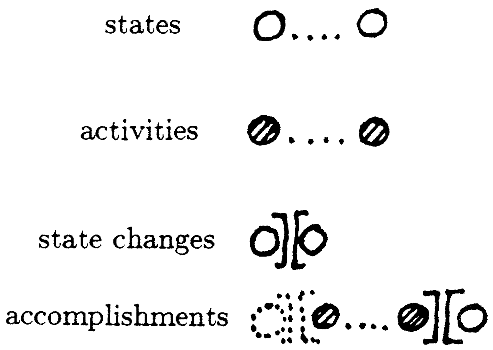


Figure 8. Progressive forms can only refer to cells in a situation that belong to a connected interval of non-trivial length. The shaded cells indicate the parts of each type of situation that the progressive can refer to.

form is possible. The situation described by Sentence 62 is one for which such a re-interpretation does not seem plausible, on pragmatic grounds. Appendix D describes other examples of re-interpretations, including conditions under which progressives can be made from verb phrases that normally describe states.

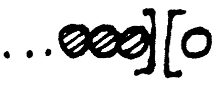


Figure 9. A representation for the action in Sentence 63. The shaded cells can be referred to using a progressive form.

The progressive form can be used even if the action has not yet occured, as in Sentence 64 or even if it may not ever occur, as in Sentence 65.

(64) Dan is making a pot of decaf coffee.

(65) Pierre Curie was crossing the street when he was killed.

To use the progressive form, it is sufficient that the speaker have some reason for viewing the on-going activity as a partial instance of this action. As Dowty (1979) and Woisetschlaeger (1976) point out, this is similar to descriptions of partial objects. For example, I can refer to several pages of typescript as "part of my thesis," even if the thesis does not yet exist (and might never exist, for all I know).

Sentences 66-69 illustrate perfect aspect forms of verb phrases, for the four types of situations.

(66) Tomás has been in his office.

(67) Anita has run.

(68) Eric has made a fresh pot of coffee.

(69) Bonnie has passed her area exam.

Like the progressive aspect, the perfect aspect makes descriptions of states from descriptions of various types of situations. However, the perfect picks out cells in an interval immediately after the action, rather than cells during the action, as shown in Figure 10. Thus, the perfect relates the situation to the moment of speech only indirectly.

The perfect also occurs in future and past tense forms, as in Sentences 70-72.

(70) James has turned the light on.

(71) James had turned the light on.

(72) James will have turned the light on.

Figure 11 shows the relationship between the state or action, the reference interval, and the moment of speech for these three forms. Perfect forms can also be made from progressives, as illustrated by Sentence 73. (Progressives of perfects are forbidden by the general restriction against creating progressives from states.)

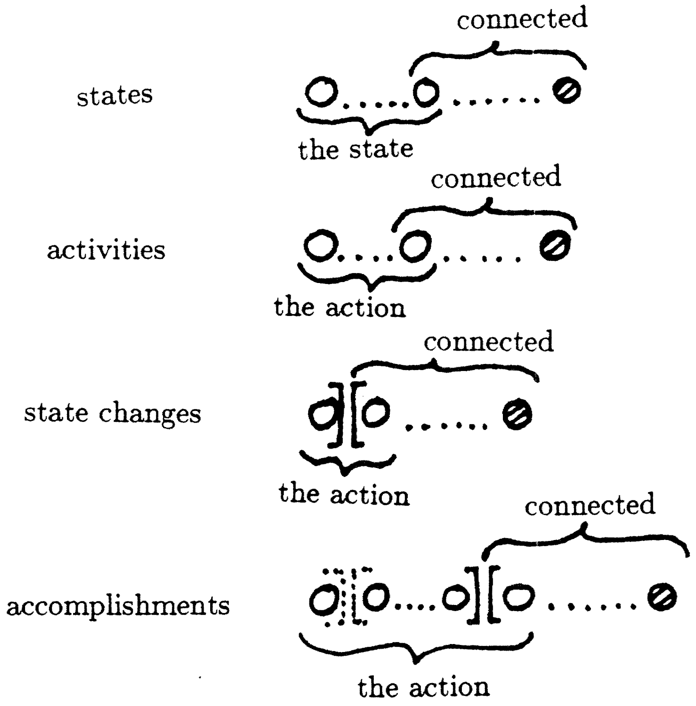


Figure 10. The cells that can be referred to by the perfect aspect are shaded. These cells must occur after the situation described by the verb. Furthermore, no boundary relevant to the current context can intervene between the end of the situation and the cell that the perfect refers to.

(73) Dan has been playing Go for four hours.

The meaning of the perfect form, however, cannot be accounted for solely in terms of the temporal relationship between the action or state, the reference interval, and the moment of speech. There is an additional requirement, traditionally expressed (Comrie 1976) by saying that the action must be "relevant" to whatever is happening at the moment of speech. Following Johnson (1981) and Woisetschlaeger (1976), I reformulate the constraint as a requirement that some result of the action or the state persist until the moment of speech. This persistence seems to involve both a requirement that the two are causally connected

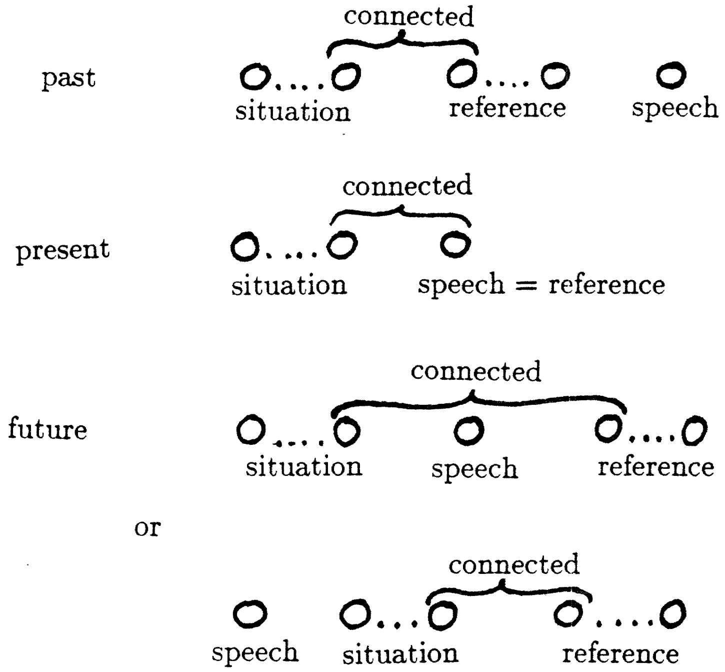


Figure 11. The perfect form specifies a relationship between the situation and the reference interval. Since the tense of the sentence constraints the temporal ordering of the reference interval and the moment of speech, the perfect indirectly imposes constraints on the relationship between the situation and the moment of speech.

and also a requirement that the intervals of time be connected.

Consider Sentences 74-78:

(74) Your cat has been in the kitchen.

(75) Eric has been making a fresh pot of coffee.

(76) Anita has run for two hours.

(77) William has lost his term paper.

(78) James has turned the light on.

The cell described by the perfect form is required to be connected to the end of the situation described by the verb.[8] According to cellular topology, this means that *no boundary relevant to the properties under consideration can intervene between the situation and the cell.* What properties are under consideration depends on the context in which the sentences is used, and thus the exact interpretation of perfect forms is very sensitive to the context.

State changes and accomplishments specify a distinctive result state that holds for the last cell in the action. This state may persist for some time and the perfect form is usually interpreted to refer to a cell during this period of persistence. Thus, Sentence 78 would typically imply that the light is still on. Other interpretations, however, are possible. If, for example, James is a young child, Sentence 78 might imply that James has learned to turn on lights by himself and make no committment as to the current state of the light.

For states and activities, one possible interpretation of perfect forms is that the state or activity continues through to the cell to which the perfect refers. So, for example, Sentence 74 could imply that the cat is still in the kitchen. However, it could also be used if the cat is no longer in the kitchen but its former presence in the kitchen had created some effect that has persisted up to the present. For example, it may have left tongue prints in the butter.

Persistence of consequences is very important to practical reasoning algorithms, such as those described in Chapter 8 (see, in particular, McDermott 1982). Suppose that I am thinking about getting coffee and someone tells me Sentence 79.

(79) Eric made a fresh pot of coffee.

(80) Eric has made a fresh pot of coffee.

[8] In the case of perfect progressive forms, this is the situation described by the progressive form.

This does not guaranteed that any coffee is left, because it might have been consumed already. However, if the speaker knows my intentions and tells me Sentence 80, he indicates not only that the coffee was made, but also that some of it is still around.

For accomplishments, the perfect and the progressive forms pick out disjoint sets of cells. Cells described by the progressive must precede the state change and cells described by the progressive must follow it. Thus Sentence 81 and Sentence 82 cannot both be acceptable statements, if uttered at the same time, unless they refer to different actions.

(81) Eric is making a fresh pot of coffee.

(82) Eric has made a fresh pot of coffee.

Sentences 83 and 84, however, can refer to the same action, though different subsections of it.

(83) The aide is shredding incriminating documents.

(84) The aide has shredded incriminating documents.

This is another test for distinguishing accomplishments from activities.[9]

The line of analysis that I present is common to a number of recent authors. Most of the phenomena described in this section are not specific to English, but occur also in other languages (see, for example, Johnson 1981, Anderson 1982, Li, Thompson, and Thompson 1982). The idea of using a reference time in explaining perfect forms dates back to Reichenbach (1947, pp. 287-298), though the details of his analysis have been modified by later researchers. In addition to the authors specifically cited in this section, Comrie (1976, 1985) provides a

[9] This test is traditionally stated, e.g. by Dowty (1979), in the following form: Sentence 83 entails Sentence 84, but Sentence 81 does not entail Sentence 82. However, the entailment for activities is only true if the activity has gone on long enough that a recognizable section of it has already occured.

clear overview of tense and aspect phenomena in a variety of languages and Bruce (1972) uses Reichenbach's representation to model tense and aspect distinctions for a reasoning program.

6. Classes of nouns

Nouns and noun phrases in English exhibit patterns of behavior similar to those of verbs and verb phrases. They appear in two difference classes, distinguished by their syntactic behavior. The differences seem parallel to the distinction between states and activities on the one hand and accomplishments and state changes on the other.

English nouns and noun phrases come in two basic types:

Objects: pear, mouse, hammock, computer
Stuffs: sand, rice, metal, wine

Nouns that typically refer to objects are known as *count nouns* and those that typically refer to types of stuff are known as *mass nouns.*

Count nouns can be distinguished from mass nouns by a number of syntactic tests. First, mass nouns can occur with the definite article, but not with the indefinite article:

the	the pencil	the wine
a	a pencil	#a wine
plural	pencils	#wines

The noun phrase "a wine" can be made acceptable, but only if it is construed as referring to a type of a wine. Furthermore, count nouns can occur in plural forms and mass nouns cannot. Again, if a mass noun is re-interpreted as the name of a kind, plural forms like "wines" become acceptable.

Mass nouns, but not count nouns, can appear with the determiner "some" (the unstressed version) or "more." They can also appear with measure phrases, such as "a cup of X." This is shown in the following table:

some	#some pencil	some wine	some pencils
more	#more pencil	more wine	more pencils
measure	#a cup of pencil	a cup of wine	a cup of pencils

In these constructions, plurals of count nouns behave like mass nouns. Thus, one could consider that the plural makes stuffs out of objects. This analysis seems to be consistent with Carlson's (1977a,b) analysis of both plurals and mass terms as names of kinds. Measures, such as "a cup of X," have the opposite effect: they make count noun phrases out of mass nouns.

I model nouns as describing the contents of connected sets of cells in 2D or 3D space, time, or abstract spaces.[10] Count nouns describe sets of cells partially[11] or totally surrounded by topological boundaries.[12] Mass nouns describe sets of cells without making any claims about boundaries. Thus, mass nouns are similar to states and activities, whereas count nouns are similar to accomplishments and state changes. This point has been noticed by a number of previous researchers, including Langacker (1987), Tenny (1987), Mourelatos (1981), and Bach (1986). The use of topological boundaries proposed here is a formalization of Langacker's "bounding" and Tenny's "delimiting."

In English, nouns are only classified into two groups. Other languages make finer grammatical distinctions among nouns. Depending on the class of the noun,

[10] For detailed discussion of abstract spaces, see Jackendoff (1983).

[11] I do not rule out objects such as infinitely long wires.

[12] Certain count nouns, such as "edges," are confined to representing an area of minimal size next to a boundary, like state changes across time. This parallel, while interesting, seems not to be linguistically relevant.

different *classifiers* may have to be attached to numerals, determiners, or verbs occuring with the noun. Many types of features can be used to establish noun classes, including approximate shape, animacy, arrangement, and material qualities (such as rigidity). (See Allan (1977) for a survey of noun classification systems.) Some of these classifications of objects may be analogous to the classification of actions presented in this section. Classification of actions, however, is limited by the fact that 1D shape differences are not very interesting.

7. Modelling time and intervals

In previous sections, I have used $\mathbb{R}^n$ as a model for time and I have represented actions using bounded, connected intervals of time. Furthermore, cellular topology constrains how time can be divided up into intervals. In this section, I re-examine these representational choices, show how branching models of time can be accomodated, and discuss why the constraints on the form of intervals make sense.

As we saw in Section 2, I model time using a cellular representation of $\mathbb{R}^n$. Some researchers in linguistic semantics, e.g. Dowty (1979), have advocated models in which time branches towards the future, as shown in Figure 12. Similar proposals also appear in some models of high-level planning (McDermott 1982). These models allow alternative possible sequences of events to be modelled in one representation of time. One might also want to use models of time that split and merge, to represent worlds that differ in some sequence of events, but then come to be the same again (at least as far as the reasoner is concerned). In this thesis, I am not going to take a stand on whether this is a useful idea or not. At the moment, there seem to be no compelling arguments for or against it.

Branching time models can be created in cellular topology, so long as the

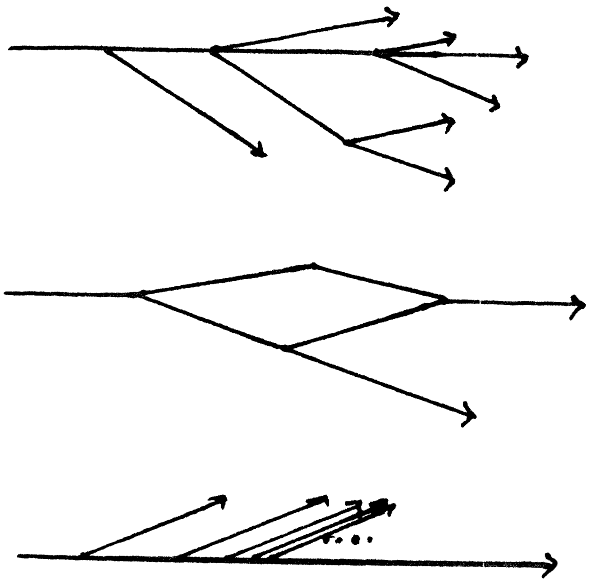


Figure 12. Time might branch (top) or split and merge (middle). In either case, infinitely many branches could join at a single point in a cellular model, but the branch points cannot be infinitely dense in time (bottom).

branch points are not infinitely dense in time, as shown in Figure 12. One way to produce infinitely dense branch points would be to model uncertainty in the length of an event by many branches in the time line. For such cases, however, it may be more effective to construct only one model for each qualitatively different sequence of events. The length of an interval could be modelled as a property associated with it, and lack of precision in length modelled just like lack of precision in a numerical property value. In cellular models, it is possible for infinitely many branches to join at a branch point, though it is unclear whether a reasoner should ever create such a model.

Many researchers in natural language semantics and high-level reasoning seem to agree that actions must be represented as descriptions of contents of intervals, rather than descriptions of properties of points. The exact status of "intervals" in the formal theory varies from researcher to researcher. Some researchers model them as primitives in so-called "interval logics" (e.g. Allen 1984) and some researchers define them as subsets of the real line (e.g. Taylor 1977, Bennett and Partee 1978). Certain researchers in reasoning (e.g. McDermott 1982, Shoham 1987a,b) define them as pairs of points in the real line.

I define an *interval* in a cellular model of time is a set of cells with no gaps. That is, it must be connected as a subset of empty time.[13] Depending on the context, I also use the word to refer to the underlying space that is the union of these (closed) cells. Thus, when no topological boundaries are present, intervals are closed intervals of the real line whose endpoints happen to fall at cell boundaries. Thus, adjacent intervals overlap at their common boundary point. When topological boundaries are present, adjacent intervals do not overlap, using either of the two models of boundaries described in Chapter 2.

Interval-based models make two claims about the representation of situations. First, verbs and verb phrases describe the contents of an interval as a whole, not the state of the world at each point in it. For example, Sentence 85 describes a situation in which the world changes over time according to some specific pattern.

(85) John repaired his car's brakes.

Thus, in order to decide whether Sentence 85 is an acceptable description of a course of events, one must examine the state of the world at many moments in time, not just one.

The second claim made by interval-based models is that situations occur without interruptions. As many authors have observed, there are many apparent counter-examples to this. For example, the activities implied by Sentence 85 can occur in two disconnected stretches of time. John might have first taken

[13] It may not be connected in the actual model of time, because state changes may impose boundaries that interrupt its connectivity.

the brakes apart in subinterval A, then eaten lunch in subinterval B, then put the brakes back together in subinterval C. However, even when the action is fragmented, the speaker must still assume responsibility[14] for an interval time without gaps, because he is making a claim not only about the pieces of action, but also about the non-relevance of what happens during any gaps in the action. Since eating lunch is not relevant to the action, it forms an acceptable gap.

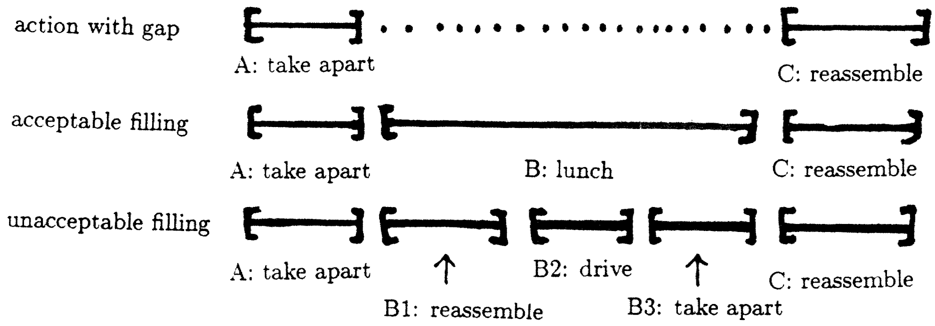


Figure 13. Acceptable and unacceptable gaps in actions.

Suppose, however, that instead of eating lunch during B, John put the brakes back together during B1, drove the car around during B2, and took them back apart during B3, as illustrated in Figure 14. Sentence 85 could then be applied to the union of A and B1, or to the union of B3 and C, but it would seem odd applied to the union of A, B, and C, because John has actually fixed his brakes twice during that time. The rationale behind this may be that the speaker is making a claim about the causal structure of the sequence of events, in addition to the claim about their temporal structure. Since causation is typically viewed as flowing along connected paths in both space and time, verifying a causally connected sequence of events requires examining all moments in a connected

[14] The phrase is borrowed from Woisetschaeger's analysis of the perfect aspect.

stretch of time.

As we saw in Chapter 2, cellular topology places constraints on the form of intervals of time used in modelling situations. Cellular models whose underlying space is homeomorphic to the real number line can only divide any bounded region of time into finitely many cells. Thus, if intervals are defined as sets of cells, an interval cannot contain isolated points. Furthermore, since boundaries can only be placed between cells, only a finite number of sharp changes in properties are permitted in any bounded region of time. This makes it impossible to describe situations such as a property true only on the rational numbers or only on the Cantor set. Such situations are not required in modelling the meaning of natural language sentences.[15]

In the models of situations presented in this chapter, I have assumed that each description specifies only a bounded interval during which the action or state occurs. I think that this is a reasonable model of the meaning of verb phrases. However, models of situations described by verb phrases must also include intervals that are unbounded, as in Sentence 86:

(86) The universe has always existed.

These sentences could be represented using a description of a bounded interval of time, together with a quantifier implying that every bounded interval in some range (in this case, prior to the present moment) fits this description.

8. Texture, scale, and support neighborhoods

The models presented in previous sections abstract away from the internal detail in actions. We have seen that this abstraction is useful in explaining

[15] At least, not sentences without overt quantifiers. It may be possible to imply these types of situations using explicit quantification, but I doubt that concrete models are appropriate in these cases.

how the linguistic data works. However, previous researchers in linguistics and philosophy (e.g. Dowty 1979, Taylor 1977, Kamp 1979) have been disturbed about how this abstraction could be done. Examples that cause problems occur in several forms: gaps, multiple resolutions, and textured activities. However, these examples are exactly like phenomena that occur also in computer vision. Although we do not fully understand how to solve these problems in computer vision, the insights from this field may be helpful in understanding how sequences of events could be parsed into the correct form for linguistic analysis.

First, as we saw in Section 7, gaps can occur during the course of an action. If the gaps do not affect the flow of the action and are small enough, they are often ignored by speakers. Gap filling, of one sort or another, must be done at all levels of visual analysis. Edge finders must attempt to reconstruct connected stretches of boundary. Shape analysis programs must reconstruct regions whose boundaries have been broken up by attachments or cut-outs. Object recognition programs must be able to identify and ignore regions due to specular reflections or surface markings.

Secondly, a given sequence of events can be described in multiple ways, particularly at different levels of resolution. For example, the single action described by Sentence 87 could also be described by the several sentences given in 88.

(87) Phil made breakfast.

(88) Phil took two eggs out of the fridge. Then he found the frying pan, ...

This, again, is familiar from vision. Only under artificial conditions (such as industrial environments) is there a single preferred scale of representation for any situation. For example, the output of the edge finder and the stereo algorithm described in Chapters 4-6 comes at a variety of different resolutions, depending on how finely the image is sampled.

Finally, activities described by natural language sentences often involve internal texture. For example, the action of "walking" involves a periodic pattern of movements of the legs. As Taylor (1977) points out, similar examples also occur in representing the meaning of mass nouns describing the material composition of different types of stuff. For example, "sand" consists of a texture of small roundish bits of mineral and "fruitcake" contains small pieces of dried fruit embedded in a background of cake. The crucial observation made by Taylor is that a point (or cell in my formalism) belongs to a region of "walking" or "fruitcake" if it is part of an interval that displays the required texture. This interval to which the point must belong is exactly like the support regions used in the stereo and texture algorithms.

Interestingly, researchers analyzing the linguistic data seem to have had less trouble with one point than researchers in computer vision. Because of the quantifier-logic approach used in linguistic semantics, researchers such as Taylor allow properties such as "walking" to be true at a point if it belongs to *any* interval that displays the required pattern. With rare exceptions (such as Tichý 1985), they do not assume that this interval must be centered about the point of interest. Thus, the formulations made by linguistic researchers are a close match to the way I formulated the requirements for stereo and texture support regions in Chapter 5. The formulations typically used in computer vision, involving centered support regions, may be an artifact of the practical problems involved in designing algorithms to compute non-centered ones.

9. Modelling the state/action distinction

The models of situations provided by cellular topology avoid certain technical problems that previous researchers have encountered. The next three sections discuss the details of these problems. This section discusses how to model the distinction between states and actions. Section 10 discusses how to model sharp state changes and Section 11 discusses how to account for the fact that spatial boundedness of direct objects can imply temporal boundedness of the verb phrases containing them.

The distinction between states and actions is linguistically important. As we saw in Sections 4-5, the constraints on different tense and aspect forms affect states and actions differently. State changes and achievements are easy to distinguish from states, because they contain a distinctive boundary. Activities, however, closely resemble states semantically. The method I have used in previous sections for distinguishing states from actions can be stated as follows:

Cellwise proposal:

> A state is a description that can be verified for individual cells.
>
> An action is a description that can only be verified for intervals containing at least two distinct cells.

Two other types of proposal have been put forth recently, the *pointwise proposal* and the *interval axiom proposal.* The pointwise proposal is similar to my cellwise proposal and is also capable of accounting for the relevant linguistic data. The interval axiom proposal, on the other hand, seems to be inadequate.

The second option, suggested by Taylor (1977), Dowty (1979), and Tenny (1987), can be stated as follows:

Pointwise proposal:

> A state is a description that is true of individual points.
>
> An action is a description that is only true of intervals.

Dowty and Taylor find it necessary to stipulate that the interval over which an action holds must be larger than a single point, because they allow isolated points to be intervals. Since intervals in cellular topology are sets of cells, they cannot consist of just a single point. Thus, I need not state this condition explicitly.

Both the pointwise proposal and the cellwise proposal express a common idea. Verifying that an action has occured requires at least two distinct measurements. For example, to verify that a rock is falling, we must observe some change in the height of the rock over time. In either case, data from two distinct moments of time must be considered. Many actions, such as "fixing the car's brakes," require more extended sets of observations. I use the cellwise proposal, because the pointwise proposal is unusable for practical purposes. Real measurements, machine or biological, can only pin down the state of the world over a period of time that has finite width. They cannot sample its state at infinitely small points.

In the interval axiom proposal, states are distinguished from actions by axioms describing relationships between truth over intervals of different sizes. For example, Allen (1984) proposes:

Interval axiom proposal for distinguishing states from actions:

> Both states and actions are descriptions of intervals.
>
> If a state is true of an interval, it is true of all of its subintervals.
>
> If an action is true of an interval, it may not be true of all of its subintervals.

Shoham (1987b) assumes a similar approach to classifying situations and counterexamples to this axiom as an argument for a finer classification. This may work for reasoning, but not for explaining the linguistic data.

The difficulty with the interval axiom proposal is that it mis-classifies verbs such as "falling" and "standing" as states, because they meet the sub-interval conditions. The standard syntactic tests for situation type, however, classify these verbs as activities. For example, as Sentences 89-90 illustrate, they can occur in the progressive. Similarly, simple present tense forms of these verbs have habitual meaning, as in Sentences 91-92.

(89) The shuttle was falling towards the earth.

(90) Marvin was standing in the playroom.

(91) #The shuttle falls towards the earth.

(92) #Marvin stands in the playroom.

The cellwise and pointwise proposals correctly classify these verbs as describing activities. It may seem strange that static situations such as "standing" should be described as activities. However, consider what it takes to verify that Sentence 90 is true. It is not true if Marvin merely passes through a standing position in the course of some gymnastic maneuver, without stopping. In order to be standing, he must remain in a standing position for some non-negligible (though perhaps short) length of time (Dowty 1979, pp. 176-177). Similarly, as someone once pointed out at an AI lab lunch, the traffic police will not consider that you have stopped at a stop sign just because the velocity of the car passes through zero. Coming up to the sign, reversing abruptly, then switching into forward abruptly and driving past the sign is not legal. A legal stop requires a *noticable* stretch of zero velocity.[16]

Interval axiom proposals have also been used to distinguish states and activities, on the one hand, from accomplishments and state changes, on the other. This can be stated as follows:

Interval axiom proposal for distinguishing states and activities from

[16] No matter *what* they usually do in Boston.

accomplishments and state changes

> If a state or activity is true of two intervals, it is true of their union.
>
> If an accomplishment or state change is true of two intervals, it is not necessarily true of their union.

Tenny (1987) attributes this statement of the distinction to Hinrichs. It also appears in discussions of the count/mass distinction for nouns (see Bach 1986 and op. cit.). This categorization clearly holds for states, because they refer to only single cells at a time. For textured activities, such as "waltzing," it need not hold, because there might be a mis-match in the pattern if the two intervals touch without overlapping.

The second clause of this second interval axiom proposal is often formulated as requiring that the union of two accomplishments or state changes cannot be an accomplishment or state change. This is too strong, both for the spatial and temporal domains. As someone once pointed out, there exist tables that link together into larger tables. Similarly, certain accomplishments can be concatenated into larger accomplishments of the same type, as illustrated by Sentence 92.[17]

(93) The robot travelled an even number of miles.

10. Representing sharp changes

A second problem in modelling actions is how to model abrupt changes in properties, such as those that occur in accomplishments and state changes. For example, in Sentence 94, we have a period of time in which Bonnie has not yet passed her exam, followed immediately by a period of time in which Bonnie has passed her exam.

[17] This example is due to Shoham (1987a,b), but he uses it in a different context.

(94) Bonnie passed her area exam.

In cellular topology, we can model this situation by putting a boundary in time. Depending on which of the two models of boundaries is used, we get one of the two situations shown in Figure 14.

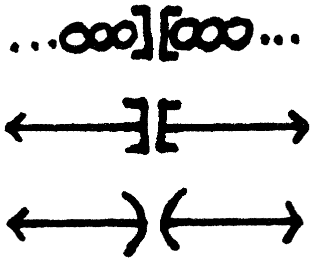


Figure 14. An abrupt change is modelled in cellular topology by adding a boundary to time, between two cells (top). This corresponds to two possible infinite-resolution models (middle and bottom), depending on which model of boundaries is used.

If time is modelled using the real numbers, there are four ways in which this situation could be modelled, shown in Figure 15. The point at the common boundary of the two periods could be assigned to the first period, or to the second period, or to both, or to neither. There are difficulties with all four methods of modelling this situation. The option in which the two periods overlap claims that there is a moment at which Bonnie has both passed and not passed her exam, which is a contradiction. If the point belongs to neither period, then there is a moment at which Bonnie has neither passed her exam nor not passed her exam.

The two asymmetrical options avoid problems with property values at individual points. However, there is no principled reason for choosing between them. It is possible to invent *ad hoc* rules for doing this assignment for intervals of time, e.g. always assign the point to the preceding interval. However, we will see in

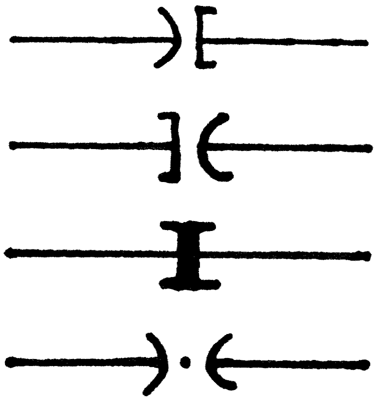


Figure 15. There are four ways of modelling an abrupt change in $\mathbb{R}$. The common boundary point can either be assigned to one of the two intervals, both, or neither.

Chapter 8 that the asymmetrical solutions extend poorly to 2D and 3D spatial examples.

A second problem with these models is that, to account for the linguistic data, state changes should occur over the minimal interval of time that contains at least one point from each of the two periods (Dowty 1979). If any interval containing a stretch of one state and then a stretch of the other state were considered an instance of the state change, then Sentence 95 should be acceptable.

(95) #Bonnie passed her area exam from 3:00 to 5:00.

Also, the constraint that state changes occupy intervals of minimal size was used in Section 5 to explain why progressives forms of many state changes are not acceptable. Both of these arguments hold not only for state changes, but also for accomplishments, which end in state changes.

There are two ways to construct minimal intervals for state changes in cellular topology. First, the minimal interval could be defined to be an interval consisting of exactly two cells, one on each side of the state change, as in the models used throughout this chapter. Secondly, if the closed-edge model of boundaries is used, the minimal interval could be taken to contain the adjacent endpoints of these two cells. (This cannot be done with the open-edge model of boundaries.) These options are illustrated in Figure 16. Full point-wise models of different types of situations were shown in Figure 3.

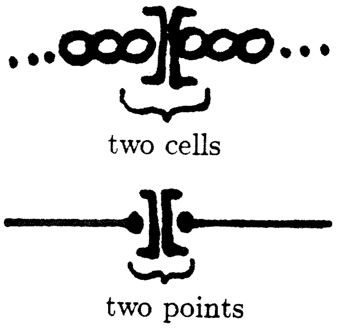


Figure 16. Two ways of finding a "minimal" interval for a state change in cellular topology.

Neither method of modelling state changes is available if time is modelled as IR, without cell structure. Under any of the four models, one of the two intervals is open-ended at the state change. Thus, no matter what interval about the state change is chosen, there is always a smaller interval that still includes points from both periods. Without the cell structure, there is no natural definition of a finite "minimal" size for an interval.

A final weakness of models based on segmenting the real line is that they do not explain why otherwise continuous functions should have sharp changes in value at these locations. As we saw in Chapter 2, because boundaries in cellular topology change the topological structure of space, continuous functions are allowed to have abrupt changes in value across boundaries. If events are modelled by segmenting $\mathbb{R}$ without changing its topology, it is unclear how to explain the abrupt changes in property values. Function continuity has not received much attention in the linguistic literature, but it is vital to the algorithms for reasoning about events over time that I discuss in Chapter 8.

I think that many researchers in this area are aware of these problems in modelling state changes, though they are not often clearly expressed. Particularly clear discussions of the problem with property values have been provided by Kamp (1979)[18] and Hamblin (1972). Perhaps to avoid the minimal interval problem, Allen (1984) represents state changes as occuring over an interval abutting the boundary from one side. However, he specifies no principled way for deciding which side of the boundary to choose. Nor is it clear how to account for the distinction between activities and accomplishments or other aspects of the meaning and syntactic behavior of verbs using such a model.

11. Combining verbs and objects

As we have seen in previous sections, the class of a verb phrase is determined not only by the verb but also by any direct object or other arguments[19] associated with it. So, for example, Sentence 96 describes an activity, whereas Sentence 97 describes an accomplishment.

(96) Eve ate fruit.

(97) Eve ate an apple.

According to Tenny's (1987) analysis, this change occurs because the verb "eat" implies a progressive change in the object described by its direct object. Since the mass noun in Sentence 96 places no limits on the amount of stuff it represents,

[18] Intellectually clear, but difficult to read.

[19] Perhaps only of restricted syntactic types, see Tenny (1987).

the action described by that sentence could continue indefinitely. However, since the object in Sentence 97 represents only a bounded amount of stuff, it must all be consumed in a bounded amount of time.

Tenny's analysis only works if a process changing the value of a property must reach a finite limit point in finitely much time. This is not true for standard models of time and processes. For example, Eve might have eaten smaller and smaller bits of the apple as time went on, so that the apple was never completely consumed.[20] However, as we saw in Chapter 2, this type of undesirable behavior cannot occur with digitized models. After the amount of apple gets small enough, a reasoner working from real measurements will be unable to distinguish it from zero. When that happens, the reasoner must treat the apple as completely gone. Alternatively, if Eve's rate of eating slows down enough, the reasoner will not be able to distinguish it from not eating. In that case, the reasoner must treat her as having stopped. One of these two things must happen after a finite period of time.

Thus, using cellular models, we can provide an accurate explanation for how boundedness of direct objects can imply boundedness of the action carried out on them. This explanation does not depend on explicitly using full digitized representations of actions or objects in analyzing natural language. It is sufficient to limit the reasoning system to situations that can, in principle, be given such a digitized model. The consequence of this limitation, that bounded objects can imply bounded actions, could be manipulated as an axiom about behavior of actions over intervals. Examples similar to this occur in high-level reasoning, as we will see in Chapter 8.

This type of pattern only appears when the verb and the direct object have

[20] Tenny seems to be unaware of this potential problem.

appropriate types of meaning. The direct object must be not only a count noun, but one representing a bounded object. For example, Sentence 98 seems plausible if one imagines a cable of infinite length:

(98) We reeled in the cable for two hours.

The verb must describe some pattern of progressive change to the direct object. Verbs that describe a bounded temporal pattern, regardless of the direct object, do not exhibit these contrasts. For example, Sentences 99 and 100 both describe accomplishments:

(99) The miner struck oil.

(100) The miner struck a rock.

Nouns describing patterns of events over time can also cause changes in verb class. For example, Sentence 101 describes an activity, whereas Sentence 102 describes an accomplishment:

(101) We sang for three hours.

(102) We sang Handel's *Messiah* in ten minutes.

Tenny (1987) interprets these examples as instances of progressive change in the direct object, but I do not find her discussion convincing.

In combination with verbs, plurals behave similarly to mass nouns. So, for example, Sentence 103 describes an accomplishment whereas Sentence 104 describes an activity:

(103) Ian compiled a program.

(104) Ian compiled programs.

Plurals can also be used to create activities out of verbs that do not normally describe activities (and often cannot take mass noun arguments), as in Sentences 105-106:

(105) Phil killed roaches for twenty minutes.

(106) Markus did problem sets all term.

Such a pattern of iterated copies of an accomplishment or a state change has no natural endpoint and thus behaves as an activity.

12. Action connectivity and models of temporal connectives

In previous sections, I have only discussed single, isolated descriptions of actions and states. However, it is also possible to express temporal relationships between pairs of actions, using connectives such as "when":

(107) The aide shredded documents when I was in the room.

In extended discourse, the order of descriptions also determines a default interpretation for their temporal relationship. In this section, I sketch a few examples of these ways of specifying temporal relationships between situations. My account of the meaning of temporal connectives and discourse sequence is based largely on Heinämäki (1978), Dowty (1986), and Hinrichs (1986).

The model of actions presented in previous sections described actions as occuring over a connected interval, flanked by boundaries. So far, we have discussed connectivity only for one isolated action. If this is really a correct representation of the topology of an action, we would expect that when temporal relationships are specified between two actions, the composite description should preserve the topology of each action. That is, either the two actions should not overlap or they should place boundaries in the same locations within the overlap region.

Such a restriction on relationships between actions should be most visible for accomplishments and state changes. As mentioned in Section 4, tensed sentences describing activities do imply that the activity ends at boundaries. However, since the activity itself does not specify what these boundaries are, it can use boundaries suggested by the context, even when these boundaries do not reflect the starting and stopping points of the activity. In fact, sentences describing activities do not usually occur in the simple aspect. If they are not transformed into an accomplishment by a measure phrase or other means, they typically occur in the progressive.

Unlike actions, states do not impose any boundaries. Thus they cannot interrupt the connectivity of any actions they might overlap. Furthermore, since states can be verified for arbitrarily small intervals (or points), they can be freely interrupted by boundaries. Therefore, there should be a difference in behavior between the overlaps of two actions, which are heavily constrained, and overlaps between states and actions, or states and states, which are relatively unconstrained.

These two patterns of behavior can be illustrated by the behavior of the temporal connective "when". This word seems to have two readings:

when X, Y

causal: X directly causes Y.

overlapping: X and Y both occur over some common interval.

These two readings are illustrated by Sentences 108-109:

(108) When his new car blew up, Mitch complained to the dealer.

(109) David was in the kitchen when I was making dinner.

The restrictions on these two types of readings are somewhat different, so I consider them separately.

The causal reading of "when" requires that the situation described by X directly cause the situation described by Y. This type of reading can occur no matter what type of situations are described by X and Y. Sentences 110-115 illustrate the variety of cases that can occur:

(110) When Susie was in school, her mother went back to work.

(111) When Phil finished his thesis, we were all very happy.

(112) When John waltzed, Mary waltzed too.

(113) Martin broke his arm when he crashed his bicycle.

(114) When Curtis pushed the button, the lights went off.

(115) The light was only on when the motor was running.

The required causal connection in these sentences restricts the types of temporal relationships that are possible. The situation described by X must occur over an interval that starts before, or simultanteously with, an interval over which Y occurs. Thus, Sentence 115 allows the light to go on either simultaneously with the start of the motor's running or, more likely, slightly afterwards. If there is a delay, no boundary can intervene, i.e. the beginning of one situation must be connected to the beginning of the other, as in the perfect aspect. Furthermore, when both of the situations in the causal reading are accomplishments or state changes and they overlap in time, the boundaries they specify seem to agree. Thus, Sentence 114 has two readings: the lights might have gone off at exactly the same time as the button pushing, or else this might have occured after a slight delay.

The overlapping reading of "when" specifies a temporal relationship between the two situations. There are three cases, depending on whether the situations are states or actions, as shown in Figure 17. If X and Y are both states, occurring over a common interval only forces some overlap between the maximal intervals over which they occur. There is a strong implication that this is a maximal interval over which X is true. As Sentence 116 illustrates, it may be possible to cancel this implication.

(116) When I was making dinner, John was in the kitchen for a few minutes, but then left.

However, I find such examples difficult to construct, contrary to the assertions of Heinämäki (1978) and Woisetschlaeger (1976). Hinrichs (1986) seems to share my impression that the overlap must contain the entire interval in which X is true.

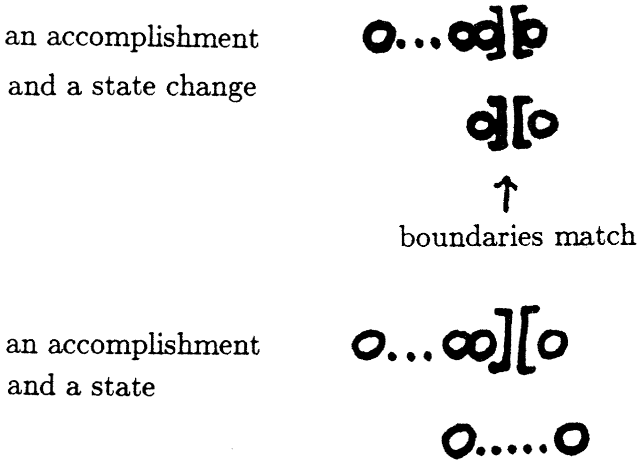


Figure 17. In the overlapping reading of "when," the boundaries imposed by actions must agree in the overlap region. When one or both of the two situations is a state, the constraints are much weaker.

If either X or Y is an action and the other situation is a state, then the state must last for an entire period over which the action takes place. This is illustrated by Sentences 117-120

(117) John broke his leg when he was at camp.

(118) When John was at camp, he broke his leg.

(119) Bobbie built the model airplane when he was living at home.

(120) When Bobbie built the model airplane, he was living at home.

If the action is an activity, the overlap must include some period over which the activity happened, but not necessarily a maximal one. For example, Sentence 121 could be used to describe a situation in which I was in the room before the shredding started, but left before it ended.

(121) The aide shredded documents when I was in the room.

Overlapping readings are difficult to get when both X and Y describe actions. In fact, Heinämäki claims they do not exist. The problem may be due to the difficulties involved in making the boundaries imposed by X and Y agree exactly. The boundaries of two actions are unlikely to agree exactly unless the two actions have a tight causal relationship. However, sentences such as Sentence 122 can be taken to describe actions that only coincide accidently.

(122) When Curtis pushed the button, the lights went off.

In such cases, the two actions must occur at exactly the same time.

Previous authors, such as Heinämäki (1978), Woisetschlaeger (1976), and Hinrichs (1986), give a slightly different analysis "when." Rather than dividing readings into causal and non-causal readings, they divide them into overlapping and sequential readings. They then propose the generalization that sequential readings occur exactly when both situations are actions. There are three problems with this approach. First, it misses the generalization that sequential readings are only possible when causality is involved. Secondly, it has trouble explaining sentences that have two actions and still overlap in time. Hinrichs, who notices such cases, addresses them by weakening "when" to allow actions to have any temporal relationship whatsoever, as long as they are close in time. Finally, such analyses have trouble accounting for examples in which one of the two situations is a state, but the reading is still sequential, as in Sentence 111.

In extended discourse, each sentence is interpreted by default as describing a situation that occurs after the situation described by previous sentences. However, in these examples, we also have a difference between states and actions. Compare Examples 123 and 124:

(123) John walked into Patrick's office. He jumped onto Patrick's desk and stared down at him.

(124) John walked into Patrick's office. The fire extinguisher was sitting next to the table.

As Example 124 shows, the action described by one sentence is typically taken to occur after the actions described by previous sentences. However, as Example 123 shows, states are typically taken to overlap previous actions, unless there is some explicit indication to the contrary.

These facts, and others, are described by Dowty (1986) and Hinrichs (1986). They claim that the basic principle in interpreting sequences of sentences is that the reference interval for each sentence is taken to follow the reference interval for the previous sentence, without overlapping. As illustrated in Figure 18, the consequences of this restriction are different, depending on the types of situations involved.

Because their models contain natural endpoints, sentences describing accomplishments and state changes are taken to describe maximal intervals over which the action occurs. Thus, two of these actions cannot overlap unless their reference intervals do. As we saw in Section 4, however, sentences describing states do not imply that their reference interval is the maximal interval over which the state

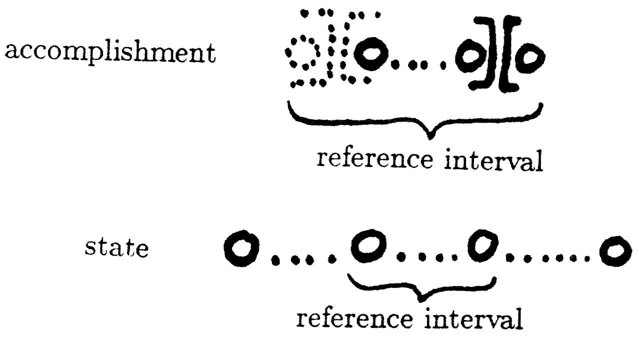


Figure 18. The restrictions on the interpretation of sequential sentences depend on the types of situations involved. States (and perhaps activities) are free to extend beyond their reference interval, whereas actions do not.

holds. Thus, the state described by a sentence is free to extend beyond the reference interval specified for that sentence. For activities, the facts are less clear. Because activities occur so often in the progressive, I have trouble generating natural-sounding examples containing activities in the unmarked aspect.

As we saw in Chapter 2, cellular topology makes it easy to represent many abrupt changes at a common location, because one boundary can license them all. A good source of coinciding boundaries is the temporal connective "until," whose meaning is described by Heinämäki (1978). Consider Sentences 125-128:

(125) Gerry kept playing with the switch until he broke it.

(126) The lecturer droned on until everyone was asleep.

(127) Until they shredded the incriminating documents, they were afraid of being caught.

(128) John kept baiting Gerry until he jumped up and down with annoyance.

The construction "until X, Y" specifies that Y occurs over an interval that ends at a state change specified by X. The details of how to define the boundary for "until" depend on the type of situation described by clause X. If X is an accomplishment, this bound is the final state change that ends the accomplishment. If X is a state or activity, the state change is taken to be the start of the state or activity.

In either case, the "until" construction seems to indicate a causal connection between the situations described by X and Y. This favors a default assumption that the situation described by Y ends at the boundary. This is consistent with the topological models, in which the boundary would license changes in any properties of similar types, not just the one that caused the boundary to be hypothesized. However, there is no requirement that the properties must change. For example, Sentence 126 might be taken to imply that the droning continued indefinitely.

Finally, words describing an absence of change, such as "keep" or "stay" can be used specifically to indicate that a state or activity persists despite the presence of a boundary at which it might naturally end. This is illustrated by Sentences 129-130:

(129) When the fire alarm rang, the lecturer just kept talking.

(130) When I pushed the button, the light just stayed green.

(131) David stayed in bed all day.

The adverb "still" can be used to indicate a similar persistence:

(132) After the bomb exploded, two pillars were still standing.

Such descriptions are traditionally described as involving some type of "resistance to change." This phrase implies some type of deliberate activity that is not necessarily present. A better description might be that such words are used to indicate a lack of change when change might be expected.

In this section, we have seen examples of how temporal relationships between sentences can be specified. As in describing tenses and aspects, the topological boundaries and connectivity of the situation models seem to be useful. In particular, when descriptions of two situations are combined, boundaries are not inserted into the middle of the connected interval over which an action occurs. Rather, two actions either do not overlap in time or else their boundaries coincide. Certain constructions imply a property change at a specified boundary and other constructions specify explicitly that the property does not change even at a boundary that should be relevant.

13. Conclusions

In this chapter, we have seen that linguistic data on verb and noun classes, tense, and aspect can be described using cellular topology. In general, the description follows the lines of those used by previous researchers. However, cellular topology makes it possible to avoid technical problems faced by previous analyses. The data on tense, aspect, and temporal connectives also provides some suggestive examples of how connectivity and the boundary co-incidence prediction of cellular topology might be used in this domain.

The topological models of situations can be tested by considering cases where intervals are closely related in time. We saw that modelling both the perfect and the progressive aspects required connectivity. Connectivity is particularly apparent for the perfect, which expresses persistence of the end-state of a situation. Consideration of the meaning of sentences in discourse and the temporal connectives "when" and "until" provides suggestive evidence that combination of two actions preserves their topological structure, i.e. locations of boundaries and interval connectivity. While this evidence is fragile, it is a useful addition to evidence from other sources.

We have also seen that the new model of space and boundaries solves several technical problems encountered by previous researchers. Using cellular topology, the distinction between states and actions can be expressed in terms consistent with real measurements. Digitized functions can provide an explanation for why certain verb phrases become temporally bounded when they contain a spatial bounded direct object. Finally, we have seen that cellular topology can provide models of sharp state changes without questions as to the values at boundary points and without difficulties in defining the minimal interval surrounding a state change.

Chapter 8: High-level Vision and Reasoning

1. Introduction

In this chapter, I survey applications of topology to reasoning about physical objects. As we saw in Chapter 3, practical reasoning involves a number of different problems, including modelling physical objects, modelling changes over time, route planning, and recognizing objects. This research is traditionally split between high-level vision, abstract planning, and robot motion planning. However, the representational problems I discuss are not specific to any of these approaches.

Reasoning about the behavior of physical objects provides examples of the same points we have seen in previous chapters, but from a slightly different perspective. We have sharper intuitions about problems in this area, particularly about connectivity, than we do about natural language semantics or computer vision. Research in natural language and computer vision concentrates on how to generate representations, whereas research in high-level reasoning concentrates on how to use these representations to plan actions. This allows reasoning research to consider a wider range of examples, particularly those involving 3D objects, but at the cost of missing some of the problems and complexities of real input data.

Sections 2 and 3 discuss how topological structure affects high-level vision and reasoning algorithms. Section 2 discusses how connectivity and the topological structure of regions is used in these algorithms. Section 3 discusses how the presence of boundaries affects the behavior of continuous functions. For both properties of objects in space and properties of events over time, we see the co-occurence of boundaries with lack of region connectivity predicted by cellular topology. We also see abrupt changes in many functions at a common location, which cellular topology can represent more easily than models postulating discontinuities in individual functions.

In Section 4, I discuss previous models of boundaries used in high-level reasoning. The proposed models are similar to those used by researchers in linguistic semantics, discussed in Chapter 7. Again, we see that the previous models of boundaries all have technical problems, which the new models of boundaries avoid. The arguments are somewhat different, however, because practical reasoning considers 2D and 3D examples, in addition to the 1D examples available in linguistic analyses of tense and aspect.

Finally, Sections 5 and 6 discuss ways in which cellular models limit the resolution of representations. As we saw in Chapter 2, cellular topology limits the form of space and boundaries, particularly when functions are digitized. Section 5 explores how these limitations affect practical reasoning algorithms. Section 6 discusses how properties with wide support neighborhoods might be used in practical reasoning and how support regions interact with digitization.

2. Topology in physical systems

Reasoning about physical objects offers the most intuitively compelling examples of how topological properties are useful. The situations presented in high-level reasoning typically have simpler structure than those used in computer vision. Furthermore, in this domain, it is sometimes possible to make useful deductions based only on topological information. Although most applications also require metric information, these examples help illustrate what topological structure can do in isolation.

Naive physics (Forbus 1984, de Kleer and Brown 1984, Williams 1984, Kuipers 1984, 1986, Hayes 1985a,b) makes extensive use of connectivity for analyzing flows. Flows can be used to describe the movement of fluids, the movement of electrical current, and the transmission of forces through objects. Connectivity information is essential to analyzing any type of flow. Suppose, for example, that we open the faucet on a sink, as shown in Figure 1. Because the pipe is now open from the water source to the faucet, water flows out of the faucet, into the sink. Whether water flows out the drain depends on whether the drain is open, i.e. whether the inside of the drain pipe is connected to the inside of the sink.

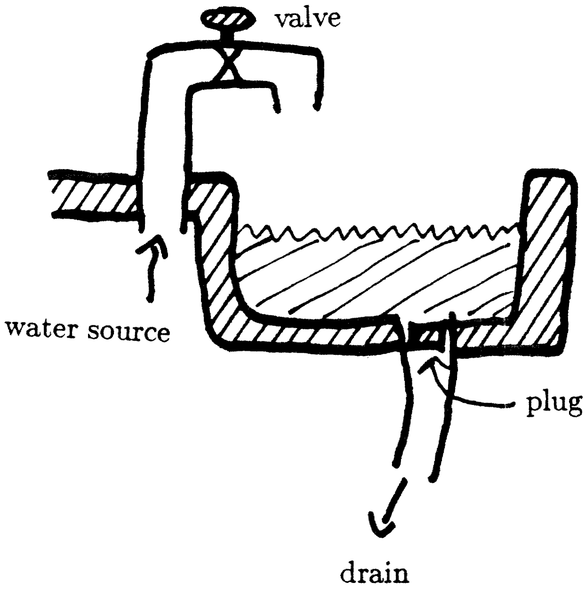


Figure 1. A sink.

The sink example illustrates both the usefulness of connectivity information and its limitations. Water can, in principle, flow through any connected path.

However, its movement is also subject to the force of gravity. So, the shape of the cross-section of the drain pipe is irrelevant to deciding whether water can flow through it. If, however, the drain pipe goes upward from the sink, water will only flow through it until it reaches the level of the water in the sink. Similarly, if the drain pipe goes downwards, we can deduce that water will only flow out the top of the sink if the drain pipe is blocked. Thus, this task requires both topological information and rough metric information.

Flows can also be used to analyze force transmission and electrical circuits. Electrical current, like water, only flows through connected paths. Since gravity does not affect current, it can flow along a connected path in any direction, given appropriate voltage differences. In analyzing object motion, metric information plays a larger role, because it is necessary to specify the direction of applied forces. For example, if one pulls on the end of a desk, the rest of the desk will also move because it is one connected object. In addition a wastebasket under the desk may also move with the desk. This second effect, however, involves pushing, which is transmitted by mere contact, rather than pulling, which requires connectivity.

One of the crucial ideas in practical reasoning is that connectedness can be used to limit causality. This idea was introduced to Artificial Intelligence research by Pat Hayes (1985b), though it is also essential to work in other fields, such as Thermodynamics (Levine 1983). The essential idea is to isolate what types of flows[1] could influence the reasoning task at hand and surround the system with barriers through which this type of flow cannot pass. In Thermodynamics, for example, the system being analyzed is surrounded by barriers that are unable to move and/or unable to transmit heat. For force transmission, it is often sufficient to surround the objects with sufficient quantities of empty space. Fluid

[1] Or similar things, such as moving objects.

flows can be contained by impermeable barriers and, with care, by barriers open only vertically. This approach can be extended so as to allow influences to pass through the barrier, but in a controlled manner.

Connectivity is also used in reasoning about object motions and in recognizing objects. Analyzing object motions has been explored both by researchers in naive physics (e.g. Forbus 1984) and in robot motion planning (e.g. Lozano-Pérez 1985, 1981). Like flows, objects can only move through connected regions of empty space. The shape of objects, however, can adapt only in limited ways during the motion.[2] Thus, these problems cannot always be solved using connectivity alone.

Figure 2 shows two examples of object motion planning problems that require only topological structure. The lefthand example can be analyzed using only connectivity relationships. Because the inside and the outside of the closed box are not connected, it is impossible for the bug to move from the inside to the outside. It requires more sophisticated topological techniques, however, to determine that the two rings in the righthand example cannot be unlinked.[3] However, neither inference depends on the shape of the objects, but only on their topological structure.

The two examples in Figure 3 require metric structure in addition to topological structure. We know from topological considerations that the bug can only leave the box through the top opening. In order to decide whether it will fit through, however, we must also compare the relative sizes of the bug and the opening. Planning motions around obstacles can also be done by transforming the problem into *configuration space* representations (Lozano-Pérez 1985, 1981,

[2] So far, research has been primarily concerned with perfectly rigid objects. Eventually, however, it will be necessary to consider objects that can deform, as most real-world objects can, and flexible objects, such as ropes and branches.

[3] I suspect people may learn to understand such examples by experimentation, rather than by deductive reasoning.

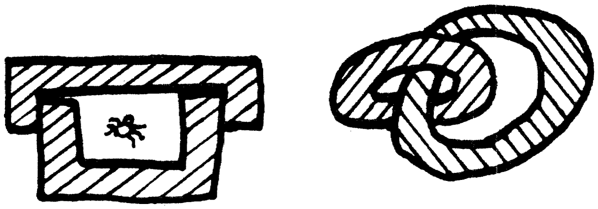


Figure 2. Some object motion planning examples can be solved using only topological information. In the lefthand example, it is sufficient to note that the inside of the box is not connected to the outside in order to infer that the bug is trapped. The righthand example requires more sophisticated topological analysis.

Lozano-Pérez, Mason, Taylor 1984, Donald 1984, 1987a,b, Erdmann 1984, 1986, Mason 1984). In these representations, each position or arrangement of the object is represented by a point. Information about the object's shape is used to compute which points in the new space would involve collisions between the object and the obstacles. Figure 4 shows a configuration space for this simple problem.[4] Using these transformed obstacles, path planning can be reduced back to connectivity checking.

Representations proposed for describing and recognizing object shape also use both metric and topological information. Particularly explicit examples of this occur in *local symmetry representations* (Brady and Asada 1984, Fleck 1985, 1986, Connell 1985, compare also Blum 1973, Blum and Nagel 1978). In these representations, regions are required to have connected boundaries[5] and the boundaries must satisfy approximate metric conditions. For example, patches of boundary opposite one another in an elongated region must be approximately reflections of one another, as shown in Figure 5. Patches of boundary in a round

[4] For ease of presentation, I have approximated the bug as a circle, to avoid building a third dimension for rotation. Configuration space for real problems typically have higher dimensionality.

[5] Sometimes by dint of filling gaps with virtual boundaries.

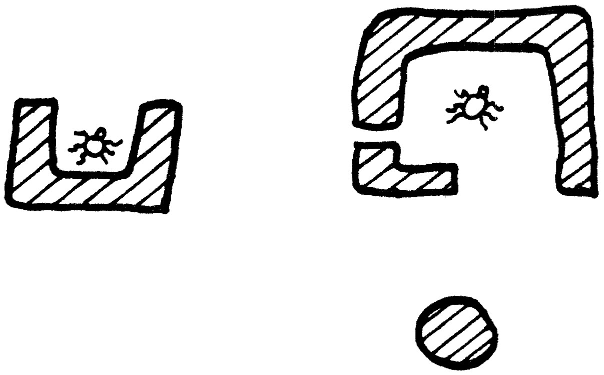


Figure 3. These two motion planning examples require metric information in addition to topological properties.

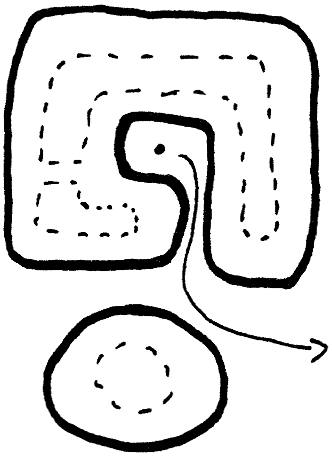


Figure 4. The configuration space for one of the motion planning problems in Figure 3 (with the bug approximated as a circle).

region must be approximately rotationally symmetric about the region's center, i.e. tangent to a circle about the center. In both cases, the connectivity requirement on the boundaries allows substantial errors in the metric conditions to be tolerated.

All shape representations in current use take advantage of both metric and

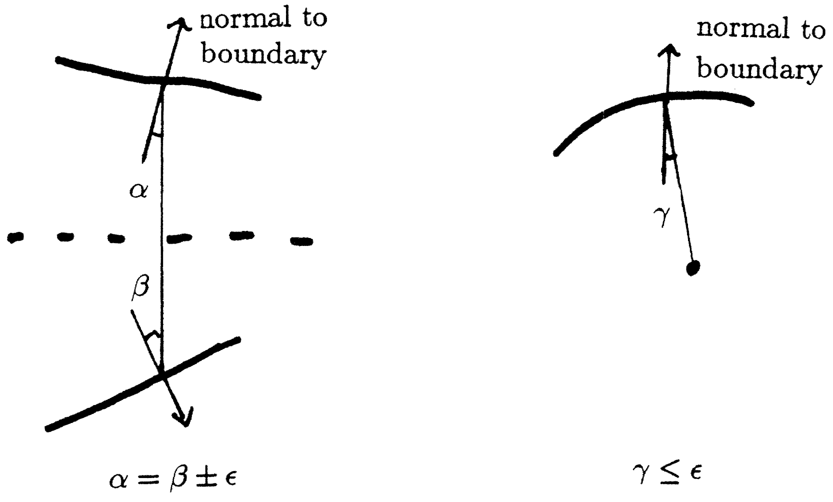


Figure 5. Representing regions using local symmetries: reflectional (left) and rotational (right).

topological constraints. In a few cases, such as the Hough transform (Ballard 1981, Davis 1982), topological information is reduced to the simple question of which points lie on boundaries. In many other algorithms (e.g. Brooks 1981), however, boundary connectivity is used to build intuitively plausible regions. This topological condition is often buried in routines that parse boundaries into extended segments, rather than being stated explicitly. It is also used implicitly in representations of boundary or surface shape (Asada and Brady 1986, Huttenlocher 1988, Brady et. al. 1985, Ponce and Brady 1987, Richards and Hoffman 1985). Descriptions of certain texture properties, such as region size, width, or orientation, involve rudimentary shape processing. This processing is often confined to connected regions (Voorhees and Poggio 1987, Kjell and Dyer 1985).

Purely topological shape descriptions are occasionally proposed (Ballard and Brown 1982, Ullman 1984). These might include descriptions of region or curve connectivity, curve intersections, and identification of topological features such as holes, homology groups, or homotopy groups.[6] As we have seen, however, only limited types of practical reasoning can be done with purely topological information. Topological features are expensive to compute,[7] difficult to combine with metric information, and poorly behaved under projection.[8] For these reasons, it seems best to use representations that combine metric and topological constraints.

Koenderink and van Doorn (1976) (see also Callahan and Weiss 1985) propose another interesting use for topological structure. A 3D object might potentially be viewed from any position around it. These positions form a sphere. Koenderink and van Doorn propose dividing these sphere at *singularities* of the views. These singularities are, informally, viewing positions at which contours appear or disappear or change shape abruptly. The singularities divide the sphere of views into regions within which the projection of the object has a constant topological structure. This proposal formalizes the idea, first expressed by Minsky (1975), that the possible views of a 3D object could be represented compactly by collapsing ones that differ only by deformations and not structural change. Huttenlocher and Ullman (Ullman 1986, Huttenlocher 1988, Huttenlocher and Ullman 1987, 1988) seem to suggest a similar idea. Although their algorithms make no explicit use of topological structure, it is preserved in all of their examples.

3. Properties and boundaries

In Sections 2, we have seen examples of how to use the topological structure

[6] For the later two types of descriptions, see Munkres (1984).

[7] Consider mazes.

[8] For example, a 3D object with no topological holes can project onto a 2D shape with holes. A convex object, however, can only project onto a convex region, under either orthographic or perspective projection.

of physical space. As we saw in Chapter 2, this same topological structure determines how continuous functions can behave. High-level reasoning algorithms must consider both the structure of situations in space and also the structure of processes across time. In both cases, continuous functions display the pattern of changes in behavior at boundaries that we have seen in previous chapters. High-level reasoning, however, offers a greater range of properties than other domains.

High level reasoning must handle two types of models: models of situations in space and models of events over time. For example, suppose that we want to describe the process of freezing water in an icecube tray. This process is really a four-dimensional object, since it involves a 3D situation changing over time (cf. Hayes 1985a,b). In human and machine reasoning, this 4D situation is described via a 1D temporal model, together with one or more 3D spatial models (shown here in 2D projection). Figure 6 shows a simple model of the course of events over time and a picture of each qualitatively different spatial situation that occurs.

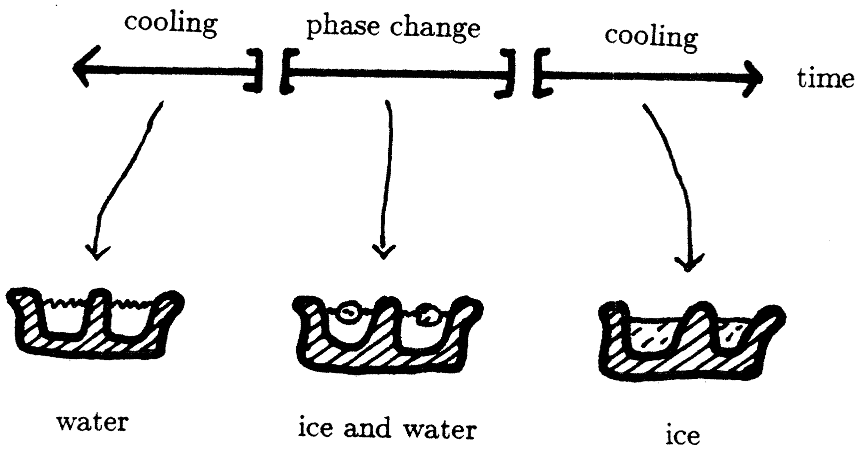


Figure 6. Freezing water.

Both spatial and temporal models involve properties and changes in properties. We saw in previous chapters that the topological model of boundaries makes it easy to represent changes in many properties at a common set of locations. Furthermore, it predicts that these changes should occur at locations where adjacent regions are not connected. The boundaries that are relevant depend on the task at hand. So, for example, two pieces of wire can be electrically connected, but not physically connected, or vice versa. Thus, the empirical predictions are restricted to similar properties and related types of connectivity.

For spatial properties, the co-occurence predicted by the topological boundary model is very important, because the properties important to high-level reasoning cannot be observed directly. For example, to understand the freezing process shown in Figure 6, we need to understand the material connectivity of the situation, together with material properties of the objects. For example, to infer that water will not flow through the tray, we need to know that the tray is materially connected and made of a plastic impermeable to water. We need to know that the plastic is solid throughout the range of temperatures involved in the freezing process (perhaps 80F to 0F) in order to predict that the pot will have a stable shape while the ice is freezing. If the tray were enclosed on the top, we would have to consider both its brittleness and its elasticity in order to decide whether it might break due to the expanding ice.

Material properties and material connectivity, however, can only be measured by trying actions and seeing whether they fail. When this is not feasible, they must be predicted from properties that can be passively observed and boundaries in these properties. Direct visual observation may yield measurements of color, light intensity, and shinyness, which help in predicting material composition. Boundaries in these properties not only yield predictions about material connectivity boundaries, but also predictions about derived visual properties such as shape, depth (from stereo or motion), and texture. These derived properties are also useful in predicting material properties. For example, we can assume that objects in a kitchen that look like icecube trays are made of a material that can withstand the changes in temperature required in normal use.

In spatial situations, both material and observable properties tend to have abrupt changes at a common set of locations. Consider, for example, the boundary between the ice and the icecube tray in the last frame of the freezing sequence. At this boundary, we probably[9] have changes in visual color, light intensity, and visual texture (e.g. shinyness). At the same place, we have a change in material, subsuming changes in melting point, brittleness, heat capacity, molecular structure, density, and opacity. Finally, the ice is not materially connected to the icecube tray. Identifying and classifying these common boundaries has been a subject of recent interest in computer vision (e.g. Poggio et al. 1988). The greatest difficulties come from the fact that different observable properties (e.g. texture vs. color) may display different boundaries[10] and common boundaries may be located at slightly different locations, due to measurement errors.

One interesting use of boundary fusion is to increase the accuracy with which certain types of changes can be localized. In the human visual system, sharp changes in intensity can be located with higher accuracy than changes in color, particularly changes in the blue-yellow color channel. If boundaries obtained from intensities can be fused with those obtained from color perception, one can obtain better localization of the color changes.[11] Such a gain in resolution is most important to high-level reasoning when the property with lower resolution

[9] Depending on the ice cube tray.

[10] Remember that properties are allowed to change abruptly across boundaries, but are not required to do so.

[11] My understanding of this point owes much to conversations with David Forsyth.

is important for infering a material property of interest.

In analysis of processes over time, we see similar patterns of boundaries at which many properties change abruptly. This is the basis of recent theories of qualitative reasoning (Forbus 1984, de Kleer and Brown 1984, Williams 1984, Kuipers 1984, 1986, compare also Erdmann and Lozano-Pérez 1987). In these theories, a course of events over time is represented by dividing time into intervals over which the world has a constant qualitative state, separated by points at which this state changes abruptly. Qualitative states are relatively simple descriptions that abstract away from numerical details not required by the reasoning task. For example, in order to predict that water will eventually freeze as it is cooled, we only need to know that the temperature is dropping steadily.[12] It is not necessary to specify the rate of temperature change, unless we want to predict how long it will take.

In qualitative reasoning, the process of freezing water might be divided into three intervals. In the first interval, the temperature of the water drops, as it is cooled. In the second interval, the temperature remains constant as the water changes phase. In the third interval, the temperature of the ice drops. At the boundaries between intervals, the type of process changes (between heating and phase transitions), the phase composition of the water changes, and the slope of temperature changes. The abstraction of "phases" of water actually conceals several co-occurring changes, including molecular arrangement, hardness, constancy of shape, density, and opacity.

Thus, we have seen that changes in multiple properties occur at common locations, both in arrangements of objects in space and in patterns of events over time. In spatial representations, lack of material connectivity often occurs at

[12] In cellular topology. For a potential source of problems in standard models based on $\mathbb{R}$, see the discussion about asymptotic function values in Section 5.

these same boundaries. (For temporal models, we have no direct evidence about interval connectivity.) This is consistent with the predictions of the topological boundary model and would be difficult to account for if discontinuities were features of individual property functions. Identifying these common boundaries is useful, because it simplifies the representation. This, in turn, simplifies the task of reasoning about the situation.

4. Modelling boundaries

The model of boundaries presented in Chapter 2 is not standard. Although it has long been recognized that something must be done about modelling boundaries, previous approaches have led to technical problems. In this chapter, we see previous ways in which the modelling problem has been approached and what problems previous researchers have gotten into.

Traditionally, modelling boundaries has been posed as a problem of segmenting a fixed underlying space, typically $\mathbb{R}^n$, into regions. Suppose, for example, that we are representing a cup sitting on a table, as illustrated in Figure 7. The problem as traditionally posed would be to divide up $\mathbb{R}^3$ into subsets representing the cup, the table, and the background. These regions should cover all of $\mathbb{R}^3$ and be disjoint. "Boundaries" would then be places at which two regions touch.

There is a widespread belief that boundaries are, in fact, "caused by" regions. Researchers in vision often feel more comfortable explaining boundaries in images as due to "objects" in the real world (e.g. Marr 1982). Researchers in practical reasoning (e.g. Forbus 1984, Williams 1984) seem to feel more comfortable if discontinuous behavior can be explained in terms of "operating states," "processes," or "events." Situations are often described in terms of the placement of objects (as in Davis 1984a,b). As a consequence of the belief that regions induce

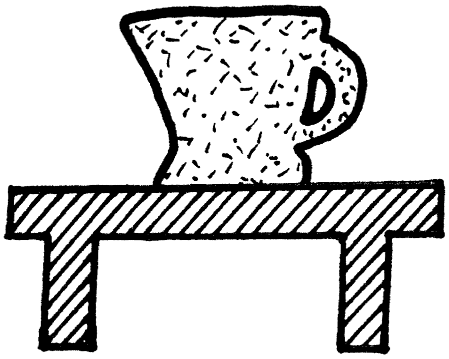


Figure 7. A cup on a table.

boundaries, researchers often state that boundaries and regions are "dual" to one another (e.g. Blake 1983, Ballard and Brown 1982, Besl and Jain 1988). However, although regions are extremely useful in explaining how people reason about the world, it is technically easier to explain regions in terms of boundaries, rather than the reverse.

In Chapter 7 (Section 10), we saw some technical problems that can occur in linguistic models of events across time. We saw that if $\mathbb{R}$ is segmented without changing its topological structure, problems arise in assigning ownership of boundary points and in accounting for the behavior of continuous functions. In high-level reasoning, the same types of temporal examples occur, together with 2D and 3D spatial examples. In this section, I concentrate on the spatial examples, because they exhibit additional types of technical problems that cannot occur in 1D representations.

The first problem with the region-based approach is that it predicts that two parts of the same region cannot touch one another across a boundary, which is false. Figure 8 shows a number of counter-examples to this. For example, a rope can be twisted so as to touch itself and a split ring touches itself along an extended border. Solid objects such as these do not merge on contact. Thus, a split ring is distinct from a normal ring, both in terms of perceived structure and in terms of its use in practical tasks. As we saw in Chapter 4, such *internal boundaries* can end abruptly in the middle of a region, both in 3D objects and in 2D projection.

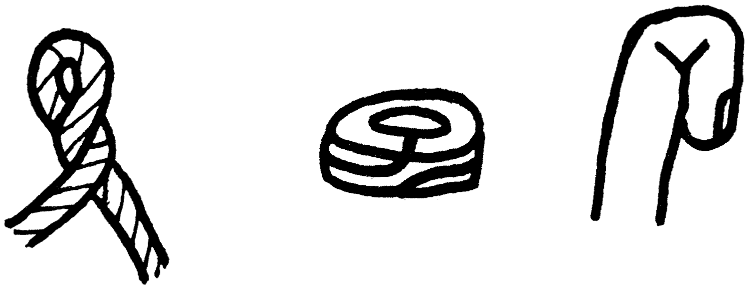


Figure 8. Examples of objects that touch themselves: a rope, split ring, and a bent finger.

The second problem with region-based models is that it is difficult to model regions that are connected to one another, such as an arm and the hand attached to it. If we define topological properties for each region separately, then the hand and arm cannot be connected. If we define topological properties using the union of the two objects, the hand and arm are connected, but so are the cup and table in Figure 7. The region-based model cannot represent the difference between these two situations except by postulating an abstract relation "connected" relating pairs of regions.

Abstract connectivity relations are often used in high-level reasoning. However, these relations are poorly developed, *ad hoc*, and involve creating two parallel theories of topology, one for within regions and one for relating pairs of regions. Furthermore, as we saw in Section 3, lack of connectivity often occurs at the same locations as abrupt changes in property values. This co-occurence is predicted by the topological model of boundaries. Segmentation models, however, provide no model for why sharp changes in property values should occur at boundaries (spatial or temporal), nor for how these changes might be related to the abstract relation "connected."

A final difficulty with the segmentation approach is that it is unclear which of two adjacent regions contains the points along their common boundary. As Figure 9 illustrates, if space is modelled using $\mathbb{R}^n$, there are three options: the boundary points belong to both regions, the boundary points belong to neither region, or the boundary points belong to exactly one of the two regions. All three of these options cause problems in practical reasoning, as as detailed by Hayes (1985a), van Benthem 1983), Allen and Hayes (1985), Pavlidis (1977), and McDermott (1982). Much of the following discussion is a cleaned-up collation of the arguments presented by these researchers.

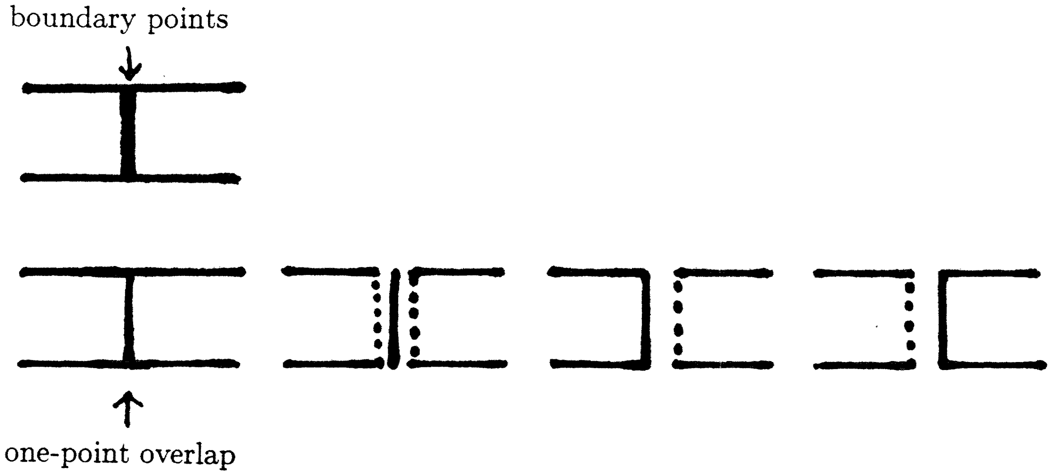


Figure 9. The two regions in the top picture share a common boundary. The bottom pictures show different ways of dividing these points between the two regions: overlap, gap, and two asymmetrical options.

Asymmetrical solutions assign each point to exactly one region, but at the cost of requiring a rule for deciding which region to assign each point to. No one has yet come up with a well-motivated rule for point assignment, in any domain. One option (Pavlidis 1977, McDermott 1982) is to base the assignment on the directions in some fixed coordinate system. For example, intervals in time might contain their earlier endpoint but not their later endpoint. Similarly, regions in 2D might contain boundary points on their left and top sides. This approach is technically unproblematic, though totally unmotivated, for intervals in time. For regions in space, however, it causes the points assigned to a region to change as the region is rotated.

The other option for assigning boundary points to one region is to develop some classification of regions and assign boundary points to certain classes of regions and not others. Hayes (1985a), for example, proposes that solid objects contain their boundary points whereas regions of empty space do not. Pavlidis (1977) proposes classifying regions in binary images on the basis of color (dark vs. light). The problem with this type of solution is ensuring that regions of the same type do not accidently come into contact. For example, the solid/empty proposal is not able to resolve the assignment when two solid objects touch one another. Classification based on color does not work if regions come in more than two colors. In 2D and higher dimensions, regions that touch themselves also cause problems for this approach.

The second option is to assign boundary points to both regions. This requires altering the definition of terms such as "overlap" so as to exclude overlap along boundaries (Davis 1984b). Unfortunately, boundaries are often created to separate two regions that bear conflicting values for some property. For example, the table in Figure 6 might be brown and the cup sitting on it red. If boundary points belong to both regions, they must bear two inconsistent values for some property function. Finally, this option produces connectivity paradoxes, as noticed by Pavlidis (1977). For example, in Figure 10, the two light areas would be connected to one another, as would the two dark areas, creating two regions that pass "through" one another.

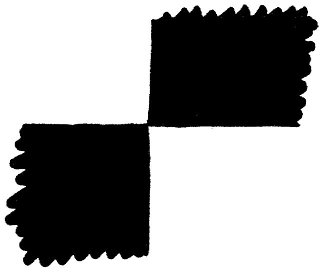


Figure 10. Are the two dark areas connected? How about the two light areas?

The third option is to assign the boundary points to neither region. The problem with this approach is that the new "boundary points" have a number of special properties that are difficult to explain. Property functions, such as color, do not assign any value to these points.[13] In discussions of temporal logic, this is sometimes called a "truth gap." Furthermore, these boundary regions do not behave like either solid objects or empty space. Intuitively, there is not any empty space in a boundary region, so you cannot put stuff there. But boundary regions cannot be moved independently as one can move real objects and their shape changes as objects around them are moved.

My open-edge model of boundaries is very similar to this third option. However, rather than endowing the boundary points with special properties, they are deleted from space. This deletion accounts for their special properties. Functions

[13] If they had a value for such a property, this could be used to assign them to one of the two regions.

cannot assign values to points that do not exist, nor can a real object occupy non-existent points of space. Furthermore, deleting these points accounts automatically for the changes in topological structure caused by boundaries.

Some researchers have attempted to avoid the problems associated with segmenting $\mathbb{R}^n$ by using models of space that are not locally like $\mathbb{R}^n$. These include Z^n (Shoham 1987), Q^n (van Benthem 1983), and the hyperreals (Weld 1988). None of these models has a pleasant topological structure for connectivity or continuity reasoning. For example no subset of Z^n with more than one point is connected and all functions from Z^n to any other space are continuous. The other two models do not handle region connectivity any better and are complicated to use.

5. Digitization

Cellular topology limits the form of regions and boundaries within any given cellular model. When digitized functions are used, they further restrict the types of situations that can be given distinct representations. Current high-level reasoning systems sometimes forbid these possibilities and sometimes allow them. In this section, I argue that the forbidden possibilities reflect unrealistic expectations about input available from either practical or scientific measurement. Re-structuring analysis using only finite-resolution representations could allow reasoning systems to handle real measurements more robustly and prevents certain technical problems encountered by researchers in this area.

Consider the problem of modelling slopes of temperatures, such as those encountered in the freezing water example of Section 3. Qualitative physics algorithms only distinguish slopes on the basis of their sign. Thus, the slopes might be modelled in cellular topology using the three-element space of values shown in Figure 11 (top).[14] This representation is similar to the second-difference labels used in the edge finder described in Chapter 4. In fact, we could use techniques similar to those in the edge finder to parse real measurements of temperature into these types of labels. That is, a combination of amplitude and duration of connected regions would be used to decide which regions have significant positive or negative slopes and which are indistinguishable from zero, given prevailing measurement errors.

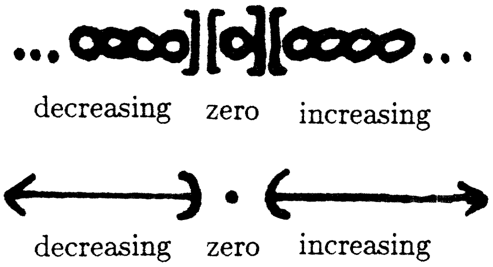


Figure 11. Top: the cellular representation of temperature slopes. Bottom: the traditional naive physics representation of temperature derivatives.

Reasoning systems handling numerical input data or numerical simulation typically make allowance for errors in the numbers (e.g. Simmons 1983, 1986, 1988, Connell 1985, 1987, Donald 1984, 1987a,b, Lozano-Pérez, Mason, Taylor 1984, Mason 1984, Brooks 1981, McDermott and Davis 1984, Davis 1986, Erdmann 1984, 1986). Qualitative reasoning systems, however, often use representations in which the value *zero* (or, equivalently, equality of certain types) is represented exactly. So, for example, the temperature slopes in the freezing example would be represented as shown in Figure 11 (bottom). Although this

[14] In many qualitative physics applications, some other property changes abruptly when the slope changes sign. In such cases, boundaries must be added to the space of slope labels, between *decreasing* and *zero* and between *zero* and *increasing*.

model looks appealingly precise, it is impractical for dealing with real measurements. There is no way we can observe a value precisely, even zero.

There are two reasons why exact values cannot be observed. First, all real measurements involve error. Laboratory conditions can reduce the magnitude of the errors, but not their existence. Secondly, physical systems only satisfy theoretical models up to some limited precision. For example, ice-water mixtures are very close to the freezing point, but individual patches of the mixture may deviate from it slightly. For the boiling water example often used in qualitative physics (e.g. Forbus 1984, Kuipers 1984), this lack of homogeneity is substantial and causes the macroscopic bubbling one observes in boiling water (Levine 1983).[15] The spurious precision of theoretical analyses in qualitative physics may be due to recognizing the existence of measurement error, but not model error.

The constraints of cellular topology not only make it impossible to create overly exact representations, but they also forbid certain possibilities that cause problems for practical reasoning algorithms. As we saw in Chapter 3, there are two distinct phenomena: infinitely dense boundaries and asymptotic function values. Figure 12 examples of these phenomena in reasoning problems. Cellular models cannot represent infinitely dense boundaries because they cannot divide a bounded region of $\mathbb{R}^n$ into more than finitely many cells.[16] Digitized functions cannot represent asymptotic function values, because the values eventually become indistinguishable from the limiting value.

These two types of infinite limits are closely related and neither one could ever occur (at least observably) in practical applications. First, when the differences from the limit value become small enough, they become indistinguishable from

[15] My understanding of these examples was improved substantially by discussions with George Fleck.

[16] I.e. without changing its topological structure.

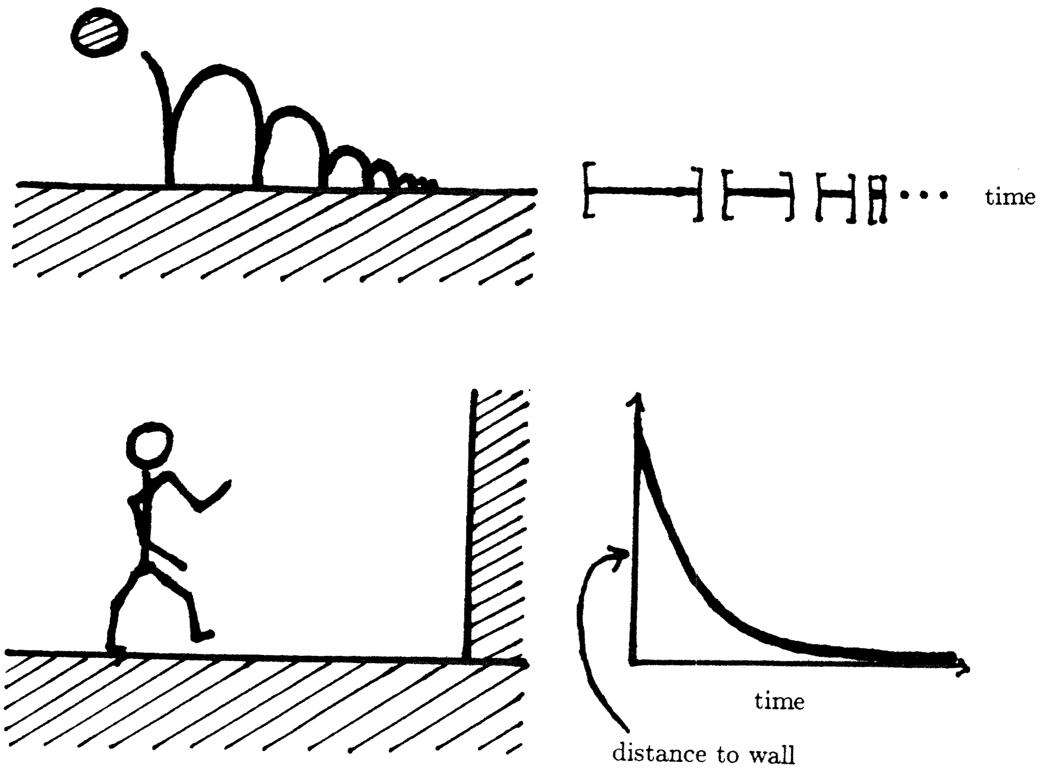


Figure 12. Top: Infinitely dense boundaries might occur with a bouncing ball, where the height of bounces decays. Bottom: Asymptotic function values might occur with a person walking slower and slower towards a wall.

zero, for any specification of measurement errors. Secondly, after some point, they also become indistinguishable from the errors in the model itself. To take an extreme case, a model of the rubber ball in Figure 12 as a solid region with some specified elasticity is no longer valid when the bounce height approaches one angstrom. Thus, we may as well assume, as cellular topology forces us to, that decaying oscillations effectively[17] stop after some finite number of repetitions and that asymptotic processes effectively reach their limit point after some finite period of time.

Infinite limits not only cannot be observed, but allowing them would cause

[17] The adverbial form of "indistinguishable."

problems for practical reasoning algorithms. Forbus (1984) prohibits asymptotic function values under the guise of a rule that forces a process moving towards a limit point to reach it in finite time. Without such a rule, even the simplest qualitative reasoning situations would generate alternative possibilities involving asymptotic values, none of which occur to naive humans. I suspect that such an assumption is made implicitly in other reasoning systems. Forbus (1987) is also forced to add an axiom specifically forcing decaying oscillations to reach their limits (Forbus 1987). McDermott (1982) also forbids the oscillation examples, on intuitive grounds. Interestingly, Shoham (1987) finds technical reasons, within his theory of causation, to forbid infinite oscillations that move backwards in time, but not similar oscillations that move forwards in time.

The limitation of cellular topology to finite resolution also affects qualitative reasoning about small perturbations. In reasoning about equilibria, a useful way to explain why the system holds a constant state is to suppose that the system was perturbed slightly and show that it must then return to its initial state. Forbus (1984) refers to this as "stutter." In infinite-resolution models, these perturbations must be infinitesimal in size. In cellular topology, the same types of explanations could be reconstructed using perturbations that are finite, but smaller than the prevailing errors.

The limitation of cellular representations to finite resolution also applies to representations of regions in space. Regions in cellular topology are represented by sets of cells. Thus, they must be the same dimension as the ambient space and cannot have any sections of lower dimension. These restrictions are often imposed by specific axioms (Davis 1984b, Ballard and Brown 1982). Some researchers, such as Hayes (1985a), do allow these types of regions. Infinitely thin regions, however, are just as unobservable as infinite limits in function values. Again, practical examples can be represented in a satisfactory way using sets of cells that are thin, but not infinitely thin.

Although a single cellular model can only represent a situation to finite resolution, it is possible to reconstruct the forbidden examples using explicit quantification. Thus, a reasoner based on cellular models could, in principle, learn calculus. However, these examples would remain more difficult to manipulate in reasoning. This seems to match human abilities. People can easily handle the finite-resolution situations found in practical situations, without any explicit teaching. However, they must be explicitly taught to manipulate infinite limit situations, facility with these examples is only acquired after years of training, and most people never learn to handle these examples successfully.

6. Support neighborhoods, scale, and texture

As we saw in the computer vision examples, the resolution of a representation is determined not only by its digitization (if any), but also by the support neighborhoods used to compute it. Wide support neighborhoods are required to avoid aliasing and drop-out in digitized functions. They are also required for representing texture and for limiting resolution, whether the function is digitized or not. In previous chapters, we have seen how these phenomena appear in linguistic semantics and low-level computer vision. In this section, I describe similar examples from high-level vision and reasoning.

The need for representations of the same situation at multiple resolutions is well-understood in practical reasoning. In fact, such representations seem to be taken for granted in fields such as Chemistry, from which many examples are taken. For example, to understand the water freezing example fully, we would need to consider both a macroscopic representation of the process in terms of phase changes and also a molecular description of how the solid structure forms. Current vision systems often produce multi-scale output. Although reasoning with representations at multiple scales is often discussed, implemented systems that actually do this, such as Patil (1981), are rare.

One recent line of research in practical reasoning has involved detecting repetitive patterns of events. Weld (1986) describes an algorithm whereby repetitions of similar events can be detected in reasoning about molecular genetics. When a repetition is detected, the pattern of events is summarized as a single continuous process. Any overall change between successive cycles is described as if it were a continuous slope in the values. The techniques of limit analysis developed for continuous processes (Forbus 1984, de Kleer and Brown 1984, Williams 1984, Kuipers 1984, 1986) can then be applied to reason about the effects of repeating the cycle many times. This type of summarization is essential in this domain, because a process may be repeated far too many times for explicitly generating the pattern to be practical. Furthermore, it allows the system to reason about the effect of multiple repetitions, even when the exact number of repetitions is not known.

Practical reasoning contains many examples of periodic patterns, as well as structures with distinctive, but non-periodic patterns. In motion planning, for example, it would be useful to describe gears as periodic, to avoid repeating motion planning computations for each tooth individually (Faltings 1987). The surface texture of objects in contact (available from both visual and tactile input) affects the friction between them. Recognizing different types of plants requires identifying periodic patterns of leaf or leaflet arrangements. Reasoning about the molecular structures of materials requires the ability to deal with both periodic structures, such as crystals, and non-periodic structures, such as liquids.

A number of researchers in computer vision have attempted to extract descriptions of textured patterns from digitized images. There are a number of texture features that might be useful for later reasoning. Vilnrotter, Nevatia, and Price (1986), Matsuyama, Miura, and Nagao (1983), Bajcsy (1972, 1973) Bovik, Clark and Geisler (1987), Zucker (1985), and Kass and Witkin (1987) concentrate on detecting periodicity in textures. Periodicity includes both repetitions occuring at discrete intervals and also continuous match of a texture against itself, often called *orientation* in computer vision. Other researchers (e.g. Voorhees 1987 and Voorhees and Poggio 1987, implementing the theory described by Julesz and Bergen 1983) have attempted to divide a texture into minimal units, called *textons* and describe the shape of these individual regions. Kjell and Dyer (1985) determine region width using inter-boundary distances, without segmentation. Laws (1979) analyzes textures by convolving the image with a range of filters.

As we saw in Chapter 5, analysis of texture properties, such as periodicity, creates properties whose support neighborhoods are much larger than single cells. As we saw in Chapters 4-6, many visual sources of input to reasoning, such as depth from stereo, also require wide support. Because the input to high-level reasoning typically involves such blurred measurements of properties, theoretical work in qualitative reasoning should use finite resolution differences, rather than the derivatives now in use (Forbus 1984, de Kleer and Brown 1984, Williams 1984, Kuipers 1984, 1986). Like other infinite-resolution representations, derivatives are not observable from real measurements, whether sensory or scientific.

A more important reason for using functions with wide-support neighborhoods is to avoid artifacts when these functions are sampled. Data used in high-level reasoning may be sampled for several reasons. First, it may come from sensors, such as those used in computer vision, that can only be packed to a finite density. Secondly, data may come from measurements taken at intervals over time, as in laboratory measurements or testing done while cooking food. Finally, representations may be sampled so that high-level reasoning algorithms can manipulate them more easily, as in the motion planning algorithms described by Lozano-Pérez (1985), Brooks and Lozano-Pérez (1985), and Donald (1984, 1987a).

Researchers in high-level reasoning often consider the possibility that data may be sampled. However, they often assume that this will be done by sampling individual points without blurring (e.g. Shoham 1987a, Forbus 1986). As we saw in Chapter 2, sampling without adequate blurring results in both aliasing and drop-out. In high-level reasoning, real gaps in data are sometimes inevitable and reasoning algorithms must be able to handle them. For example, food cooking in an oven must be taken out of the oven to be sampled and thus samples must be taken only rarely, to avoid disturbing the cooking process. However, it is important not to confuse these cases with situations where wide-support neighborhoods can be used. For example, if a dial or a moving ball is under continuous observation, it is reasonable to assume that the data can be adequately blurred before any sampling is done. When adequate blurring can be done, reasoning algorithms need not consider the possibility of sampling artifacts.

7. Conclusions

In this chapter, we have seen several things of importance to this thesis. First, we saw that connectivity and other topological properties are important in high-level vision and reasoning. These properties can occasionally be used alone, but they are more often combined with metric constraints, as in reasoning about object motion or fluid flow. We have also seen that abrupt changes in property values are important in practical reasoning, both for describing objects in space and events in time. When these abrupt changes occur, they follow the pattern predicted by the new model of boundaries. That is, multiple functions tend to have abrupt changes at a common set of locations and material connectivity tends to fail at these same locations.

In Sections 4-6, we saw that the cellular models presented in Chapter 2 can avoid technical problems faced by previous researchers. First, we saw that the new model of boundaries avoids problems with representing internal boundaries, representing connected objects, and assigning boundary points that occur in previous models. We saw that cellular models and digitized functions constrain representations so that they cannot represent infinite limit behavior or infinitely thin regions. We saw that these constraints help avoid technical problems and allow the representation to better match data available from real measurements. Finally, we saw that functions with wide support, such as texture descriptions and blurred sampling, may be useful in high-level reasoning, though they have not been extensively used by previous researchers.

Chapter 9: Testing the edge finder

1. Introduction

In this chapter, I describe a series of experiments that test the performance of the Phantom edge finder against the edge finder described by Canny (1983, 1986). There are two groups of tests. The first group evaluates the stability of edge finder output in the presence of camera noise and changes in digitization. The second group evaluates the resolution of the two edge finders, i.e. the extent to which each edge finder can handle fine detail. The results of these two tests show that the new edge finder is better at both suppressing the effects of camera noise and detecting fine detail.

The evaluations of edge finder performance presented in this chapter use an approach that is not standard in computer vision. Previous evaluations of computer vision algorithms have been based on determining the correctness of a program's output, whereas my evaluations are based on measuring its stability. Section 2 discusses how these two approaches differ and why the stability approach allows more realistic evaluations to be done with only incomplete models of reality and of later vision tasks.

Sections 3-5 discuss the details of the stability tests performed for this thesis. Section 3 shows examples of how the matcher from Chapter 5 can distinguish stable features from noise and discusses procedures used in doing the evaluations. Section 4 presents the results from the main test for stability in the presence of camera noise. Section 5 discusses the results of two smaller tests, one using images with noise of higher amplitude and one using changes in digitization.

Section 6 presents examples of qualitative differences in edge finder output. Although these differences could also be cast as differences in stability and measured quantitatively, such tests are beyond the scope of this thesis. These examples show several ways in which Canny's edge finder creates undesirable output, including rounding or breaking sharp corners, breaking boundaries near intersections, and producing spurious responses on ramps. We see that the Phantom edge finder avoids these problems, although it creates phantom boundaries on staircase patterns.

2. Stability vs. correctness

The edge finder evaluations presented in this chapter are based on determining the stability of the edge finder's output under various types of changes to the input. Previous edge finder evaluations have been based on measuring the correctness of the output, rather than its stability. In this section, we see how these two approaches differ and how the correctness paradigm has limited previous attempts to evaluate early vision algorithms.

Evaluating the stability of edge finder output uses the matcher described in Chapter 5. Stability must be measured with respect to some type of change in input. In most of the tests presented here, I am concerned with stability in the presence of camera noise. Ideally, if the edge finder is stable in the presence of camera noise, it should produce the same output on two pictures of the same scene that differ only in having different samples of random noise added by the camera system. The tests presented in Sections 3-5 determine the extent to which this is true.

Stability in the presence of camera noise is a minimal requirement for a low-level vision algorithm. For many tasks, stability under other types of changes is also required. A pilot test, described in Section 5, assesses the stability of the two edge finders under changes in digitization. This is done by comparing their output on two images of a photograph that has been translated relative to the camera. Similar tests could be done using other types of changes in the input, such as tilting the scene relative to the camera, or by assessing the extent to which boundaries from two different properties, such as stereo and texture, line up.[1] The results of the stereo algorithm, shown in Chapter 10, provide an informal measure of stability under changes in viewpoint.

The experiments presented here provide quantitative evaluations of edge finder performance on natural images containing an extensive number and variety of boundaries. Previous quantitative evaluations have been confined to simple synthetic images. This increase in coverage is due to two factors. First, the image matcher described in Chapter 5 makes it possible to compare two edge finder outputs robustly. Secondly, previous experiments have been dependent on definitions of the "correct" output for each image used, because they have attempted to measure the correctness of edge finder output, rather than its stability. (I only know of one previous evaluation based on stability, Nishihara's (1984) evaluation of the output of his stereo matching algorithm.)

If there were a generally accepted notion of what constitutes correct edge finder output, then it would make sense to compare the output of real edge finders against an idealized output. This is the approach used by previous researchers such as Haralick (1982), Sher (1987a,b), Pratt (1978), and Fram and Deutsch (1975). Unfortunately, generally accepted models for correct output only exist for

[1] Similarly, one might assess the agreement between surfaces estimated by stereo vision and surfaces estimated by tactile sensing.

extremely limited types of boundaries, chiefly minor variations on straight step edges. It is unclear how to generalize these definitions. It is frequently asserted (as in Marr 1982) that boundaries in images should be the projections of boundaries of objects in the real world. However, this does not simplify the problem of definition, because there is no generally accepted definition of what constitutes an object in 3D or, given an object, what its boundaries are. Introspective and psychophysical data provides constraints on boundary locations, but not with sufficient precision to decide between two relatively good algorithms.

Furthermore, even if we did have a definition of the correct output for each type of boundary present in natural images, it would still be necessary to obtain these correct answers for images used in evaluations. If the images are synthesized, it is difficult to ensure that they are accurate simulations of real camera images. If the images are not synthesized, it is unclear how one would determine the correct answers. Doing this by hand would be tedious and error-prone for images of any complexity. Since mechanical procedures for finding boundaries are exactly the object of study, using one edge finder to check the results of another edge finder creates a circularity.

The correctness approach makes the evaluation problem more difficult than it has to be. In order to obtain meaningful evaluations within this paradigm, it is necessary to have a complete model of reality and of all applications to which edge finder output will be put. The model of reality is required in order to generate realistic test images and the model of applications is needed to determine the correct output for each test image. Since current research does not seem close to building either type of model, this type of evaluation cannot be done.

The stability approach that I use has two advantages. First, it allows natural images to be used in evaluations, because no definition of the "correct" answer is required. This allows evaluations to be done under realistic conditions without having to build detailed models of the real world and the camera system. Because of the new image matching algorithm, evaluations can be almost completely mechanized. This allows evaluations to be done on large images with fine detail, illustrating a large variety of boundary shapes.

Secondly, stability evaluations can be done with only partial understanding of the requirements of later processing. For example, we know that applications such as stereo matching require that edge finder output be stable under small changes in viewpoint. The results of tests for this type of stability would still be valid even if we later learn that other applications require an additional type of stability, such as stability under changes in lighting. This incremental property of stability evaluations is essential, because our knowledge of how visual processing should be done is still very limited.

3. Procedures

This section discusses the procedures used for doing the stability tests. The results of these tests are described in Sections 4 and 5. In this section, I discuss how the amount of change in an image and the number of boundaries in an image were measured. We see examples illustrating how the matching procedure separates responses due to noise from those due to stable features of the scene. Finally, I describe how the output of Canny's edge finder was converted into the format used by the matching algorithm, so as to make comparative evaluation possible.

The details of the image matching process were discussed in Chapter 5. Given a pair of images and an initial alignment between them, this matcher produces a map of which parts of the image can be matched without topological changes. In edge finder evaluation, images should match where the edge finder has detected real features of the scene and they should not match where the edge finder output reflects camera noise or is heavily corrupted by camera noise. Figure 1 shows matching results for a small image, one of the translation examples discussed in Section 5. The matcher correctly marks regions of noisy responses as non-matching. Although two images never match perfectly, the difference between adequate and inadequate noise suppression is clearly indicated by the percentage of the image that is successfully matched.

Stability evaluations uses two images, reflecting views of the same scene with some small change between them, such as having different samples of random camera noise. In these tests, one of the images is viewed as the primary image and the other as a secondary image, used only for comparison. Matching results are computed using both images, yielding two numbers: the number of cells not matched successfully and the number of cells whose labels had to be altered in order to yield a successful match. Both of these numbers are stated as percentages of the total number of cells in each image. In addition, the total number of successfully matched edges is computed, using only the primary image.

Two pre-tests were run to make sure it was reasonable to use only two images for matching and only one image for computing the edge percentages. These pre-tests used a five-image sequence, containing the high-noise pair described in Section 5, together with three more images of the same scene. In the first pre-test, all five successive pairs[2] of images were matched. The percentages of the image that matched, with and without adjustment, were computed for one edge finder threshold (90), as described below. The numbers for each pair differed at most ±0.6% from the mean taken over all five pairs. The percentage of edge cells

[2] That is, images 1 and 2, 2 and 3, 3 and 4, 4 and 5, and 5 and 1.

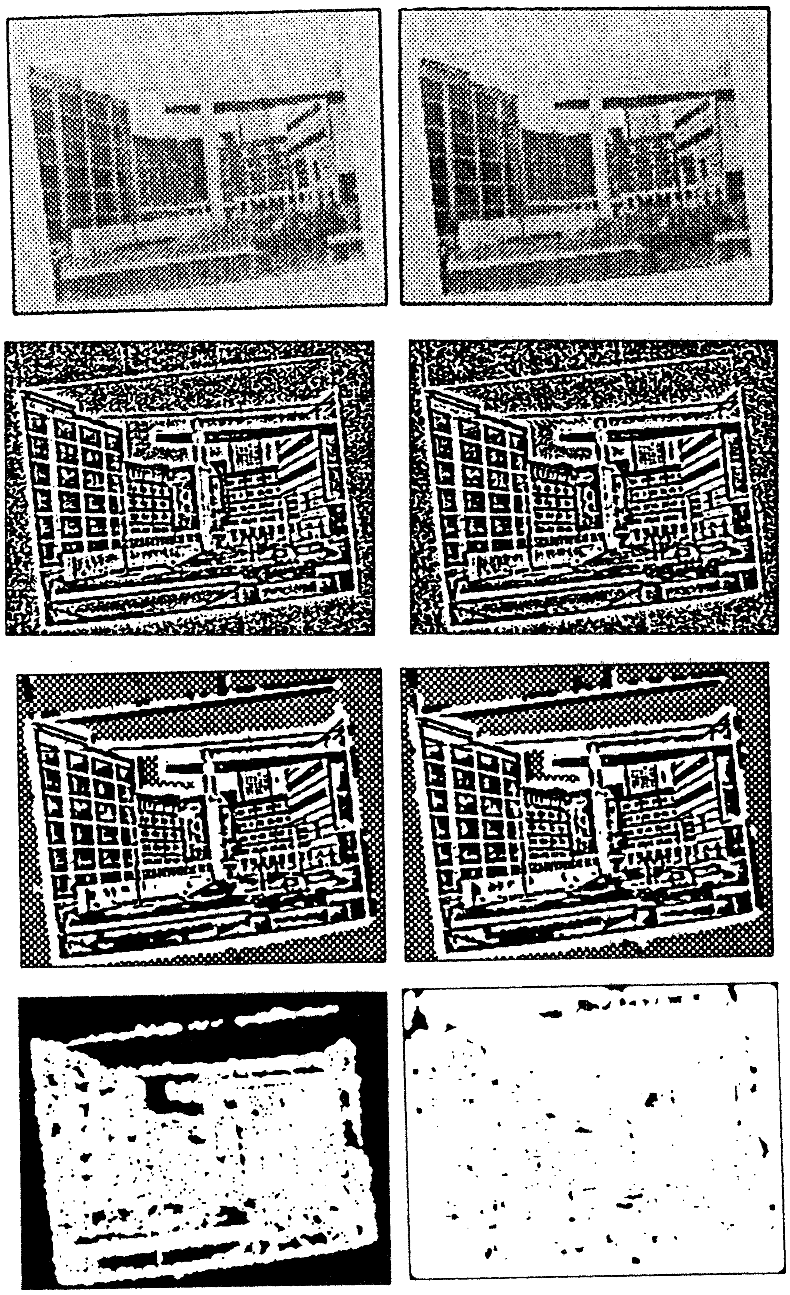


Figure 1. Top to bottom: Two images, edge finder output without adequate noise suppression, edge finder results with adequate noise suppression, and matching results. The lefthand match map shows the result of matching the noisy outputs and the righthand map shows results for the clean outputs. In both cases, matching cells are shown in white and non-matching cells in black.

was computed for each image, also according to the details given below. This was done for each of six edge finder thresholds (1, 30, 60, 90, 120, 150). For each threshold, the individual numbers differed at most ±0.5% from the mean for that threshold.

The first number computed in the stability evaluation is the percentage of the image that did not match successfully.[3] For my edge finder, this number can be computed straightforwardly from the clean match map produced by the image matcher. Canny's edge finder, however, produces results in a different format and it is necessary to convert his results into a format that the image matcher can use, before the match percentage can be computed. The difficulties come from the fact that Canny's edge finder produces on-cell, rather than inter-cell, boundaries and it does not produce dark/light labels explicitly, but rather encodes sign information in boundary orientations.

Dark/light labels for Canny's edge finder are reconstructed in two stages. Suppose that x is a cell next to a boundary cell y. The algorithm computes the orientation of x relative to y. This orientation is compared to the boundary orientation reported by the edge finder and x is labelled dark or light accordingly. This computation is done for all such pairs of cells. Because the relative orientation of two cells is quantized to 45 degrees and boundaries may be closely packed, occasional errors occur. These errors typically result in a cell being assigned both the label light and the label dark. Such cells are re-set to have no label. When all cells next to boundaries have been labelled, the labels are propagated five cells outwards, so as to create labelled regions of width similar to those produced by my edge finder. The results of this process are illustrated in Figure 2.

[3] It would be equivalent to use the percentage that did match successfully. However, the non-matching number is more convenient for graphing, because it is small for the cases of most interest, when noise suppression has been done more or less successfully.

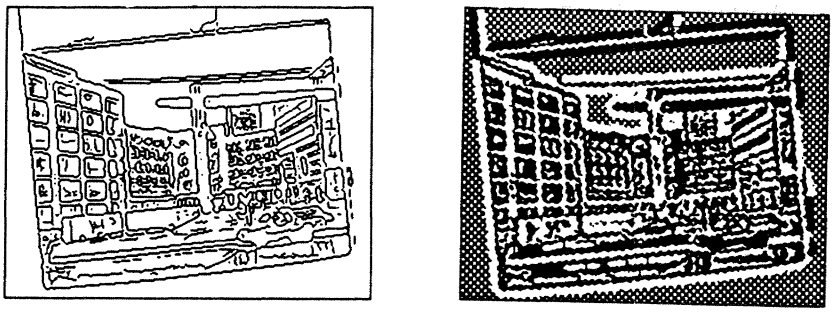


Figure 2. Dark/light labels (right) can be reconstructed from the boundaries produced by Canny's edge finder (left), using boundary orientation information provided by this edge finder.

Label reconstruction allows match maps and match percentages to be computed for both edge finders. The other two numbers used in evaluation are the number of successfully matched edges in the image, stated as a percentage of the image, and the average amount of boundary motion. Recall that an edge cell is a cell adjacent to a boundary, but not actually in the boundary. Each boundary is surrounded by two sets of edge cells, one to each side. Thus, the number of edge cells in an image is a rough measure of the total length of the boundaries in that image. Furthermore, this measure does not discriminate unfairly against either my edge finder, which places boundaries between cells, and Canny's edge finder, which places them on cells.

In order for edge counts to be as useful as possible, two niceties must be taken care of. First, edge cells are not computed on the raw edge finder output, but rather on a copy of this output enlarged by a factor of 2 in each dimension. If this were not done, thin regions and regions of high curvature would be assigned unfairly low edge counts, because a cell can border a boundary on two sides. Secondly, edges are only counted within regions that matched successfully. This is done by re-labelling non-matching cells as "zero" prior to identifying edge cells. This means that edge counts reflect the number of useful edges and do not include edges due to inadequately suppressed noise.

For Canny's edge finder, there are two distinct ways to compute the number of edge cells. In addition to those boundaries actually reported by the edge finder, other boundaries may be induced when the labels from two boundaries bump into one another during the process of spreading dark/light labels. These *staircase phantoms*, discussed further in Section 6, only sometimes represent real scene boundaries. As we see in Section 6, my edge finder consistently marks spurious boundaries in such cases, which tends to increase the total number of edges it reports. In order to assess the effect of this difference, edge counts for Canny's edge finder were computed both with (*filled*) and without (*unfilled*) these phantom boundaries. For comparative purposes, these represent over- and under-estimates of some hypothetical objective edge count.

The amount of boundary motion is estimated from the total number of cells whose labels were adjusted so as to produce a successful match. This number is divided by the total number of edge cells (as described in the previous section), to yield an estimate of the number of cells moved per unit of boundary. Because each boundary is surrounded by two edge cells, you might think that this figure would be twice too large. However, remember that the edge cells were computed on an expanded version of the image. This contributes an additional, inverse, factor of two. Thus, the ratio of the number of adjusted cells to the number of edges does roughly measure the amount of boundary motion.

Again, there are a few niceties involved in computing boundary motion. First, staircase phantom boundaries move in roughly the same manner as the real boundaries generating them. Thus, they also produce cells whose labels must be adjusted in the matching process. For this reason, it is essential to use the *filled* edge cell counts when computing boundary motion for Canny's edge finder. Secondly, label transitions in the interior of large regions also produce cells whose labels must be altered during boundary adjustment. This was not taken into account in the current evaluations and thus the estimated of boundary motion reported below is high. Because the test images contain large amounts of dense texture, the effects of this over-estimation should be small.

Thus, evaluation of my edge finder yields three numbers for each pair of outputs matched: the percentage matched, the percentage of edges, and the amount of boundary motion. For Canny's edge finder, four numbers are computed, because two alternative edge percentages are given. The match percentages measure the extent to which the topology of the edge finder output is stable. The amount of boundary motion measures the extent to which boundary locations are stable, in regions of stable topology. Both of these evaluations can be improved by restrictive settings of edge finder noise thresholds, at the cost of reducing the number of boundaries detected. The percentage of edges is used to assess how much useful information is lost as stability is increased.

There are a few more points that I should note about the implementation of Canny used in these tests. I used the current MIT implementation of Canny's edge finder for the Symbolics LISP Machine. This code is similar to that used in Canny's original implementation, except that it has been adjusted for changes in Symbolics software and edge finding is done at only one scale. That is, the feature synthesis algorithm has been removed. The MIT implementation also estimates the amount of noise in each image and adjusts noise thresholds accordingly. I have disabled this estimator because it is unreliable and it would have made constructing controlled experiments more difficult.

The MIT implementation of Canny's edge finder uses two parameters for noise thresholding: a low threshold and a high threshold. These thresholds are by the hysteresis algorithm described in Canny (1983, 1986). In the following tests, the higher threshold was set consistently to twice the low threshold. Based on both the default setting used at MIT and my own experience with the edge finder, this is a good setting for this parameter. Thus, the single noise parameter mentioned in the following sections will be the low noise threshold. The experiments used a range of settings for this threshold.

4. The main noise test

The first and largest stability test measures the stability of both edge finders under moderate amounts of camera noise. This test used four large images containing dense texture. Evaluations were run for both the Phantom edge finder and Canny's edge finder, using for a range of parameter settings in each case. We see that Canny's edge finder produces a worse tradeoff between stability and number of edges, particularly for the smaller mask sizes.

The images for these tests were taken with a Panasonic WV-CD50 CCD camera and a framegrabber built at the AI lab. This system adds only low-amplitude noise to the digitized image. To a first approximation, the noise could be described as Gaussian, with $\sigma = 3$ intensity units. The images are also blurred slightly, apparently before the noise is added. This blur could be modelled as convolution with a Gaussian of $\sigma = 1$ cell. Detailed measurements of the noise and blur are not available.

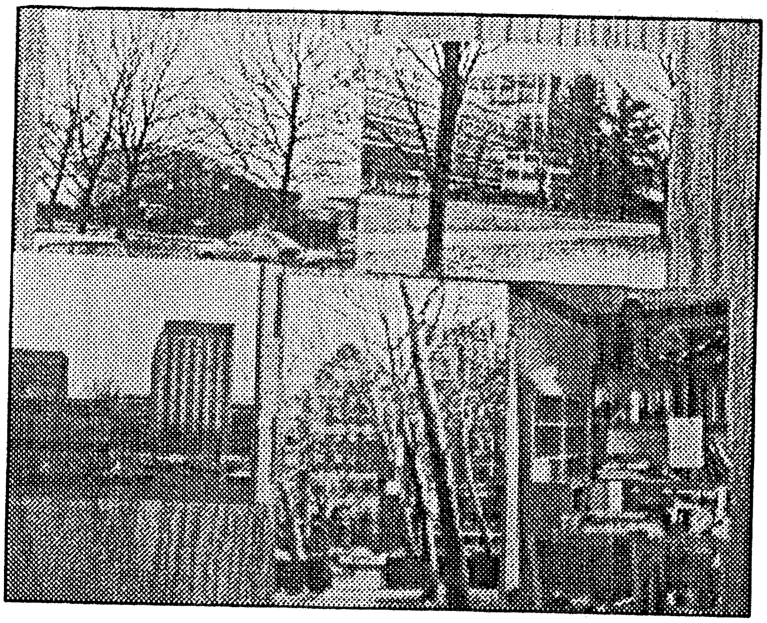
Figures 3 and 4 show the four images used in this test. Each image is 454 by 576 cells. The system supplies 8-bit intensity values and the camera aperture was adjusted so that the images cover most of this range. The ranges of intensity values in the test images vary between 209 and 243 intensity units. One scene was made up of real objects. The other three were created using black-and-white photographs, so as to pack large numbers and types of boundaries into each image. The backdrop in all cases was a wooden table surface. Two images of each scene were taken, a minute or so apart.

Figures 5-6 show the results of the Phantom edge finder for these four images and Figures 7-12 show the results of Canny's edge finder for three different mask sizes. These images were created using noise threshold settings that represent good tradeoffs between stability and number of edges detected. For Canny's edge finder, boundary cells are shown. For the Phantom edge finder, edge cells on the dark sides of boundaries are shown, in addition to boundary cells, because many boundaries contain no boundary cells.[4] Intuitively, the Phantom edge finder seems to be detecting boundaries to higher resolution with less noise. The point of this test is to obtain quantitative assessments of how noisy these outputs are. The tests described in Section 6 assess the amount of resolution delivered.

The evaluations described in Section 3 were done for a range of parameter settings of each edge finder. For the Phantom edge finder, noise threshold settings of 1,[5] 30, 60, 90, 120, and 150 were used. A seventh run with threshold 15 was later added to fill out a sparse section of the curves. Canny's edge finder was run with mask sizes 4, 8, and 12, corresponding to Gaussian smoothing with $\sigma = 0.5$, 1, and 1.5 cells respectively. The default mask size setting for the MIT implementation is 8, with occasional applications using mask size 12. For each mask size, the edge finder was run with low noise threshold settings of 0, 50,

[4] This heuristic is used in order to create boundary displays that are the same size as the original image. It has the bug that the boundaries on two sides of thin dark regions may be coelesced. However, fully detailed boundary maps, such as those used in Section 6, require expanding the image by a factor of two.

[5] A setting of 0 causes the entire image to be labelled as boundary cells, due to details of the implementation.



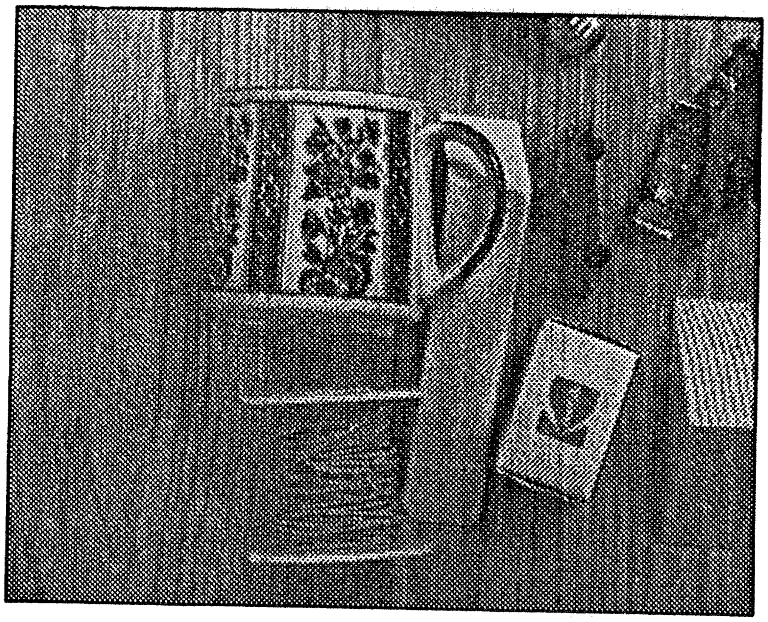


Figure 3. Two of the images used in the main stability test.

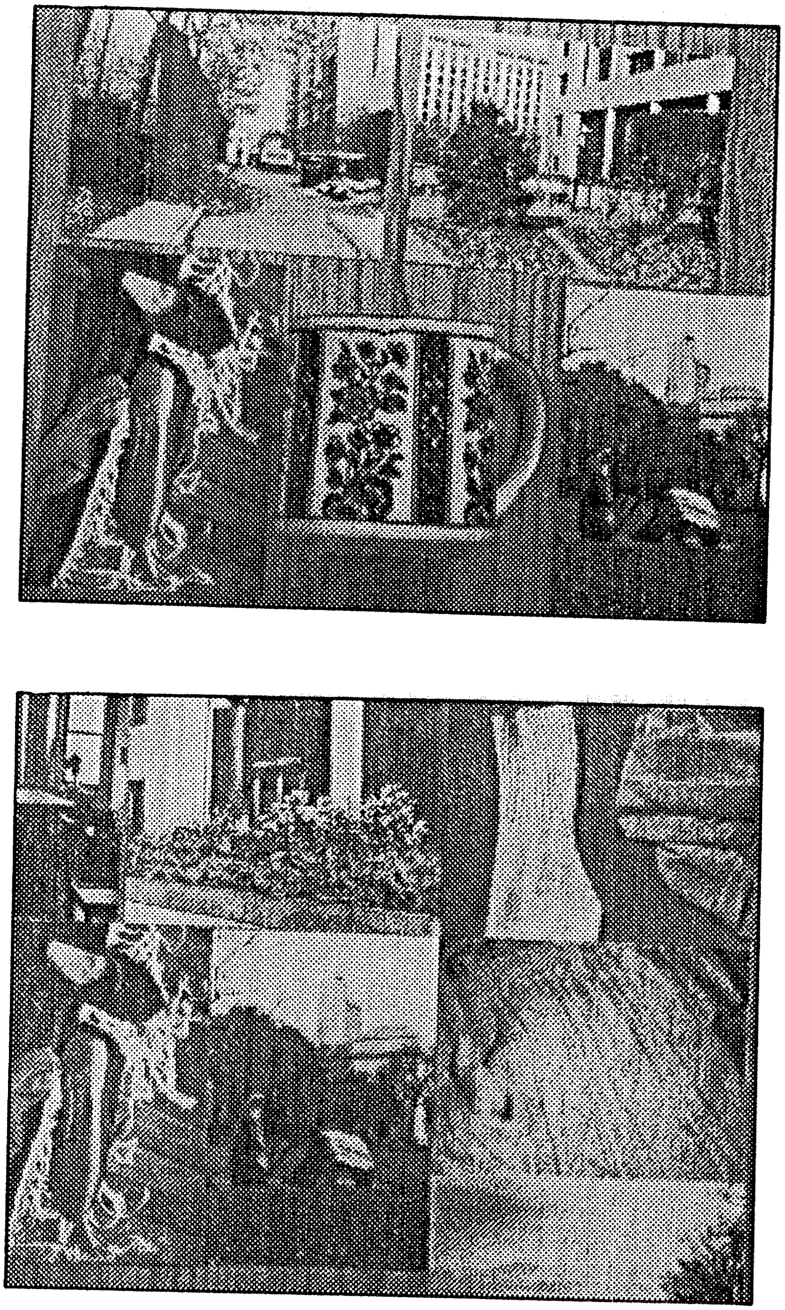


Figure 4. Two of the images used in the main stability test.

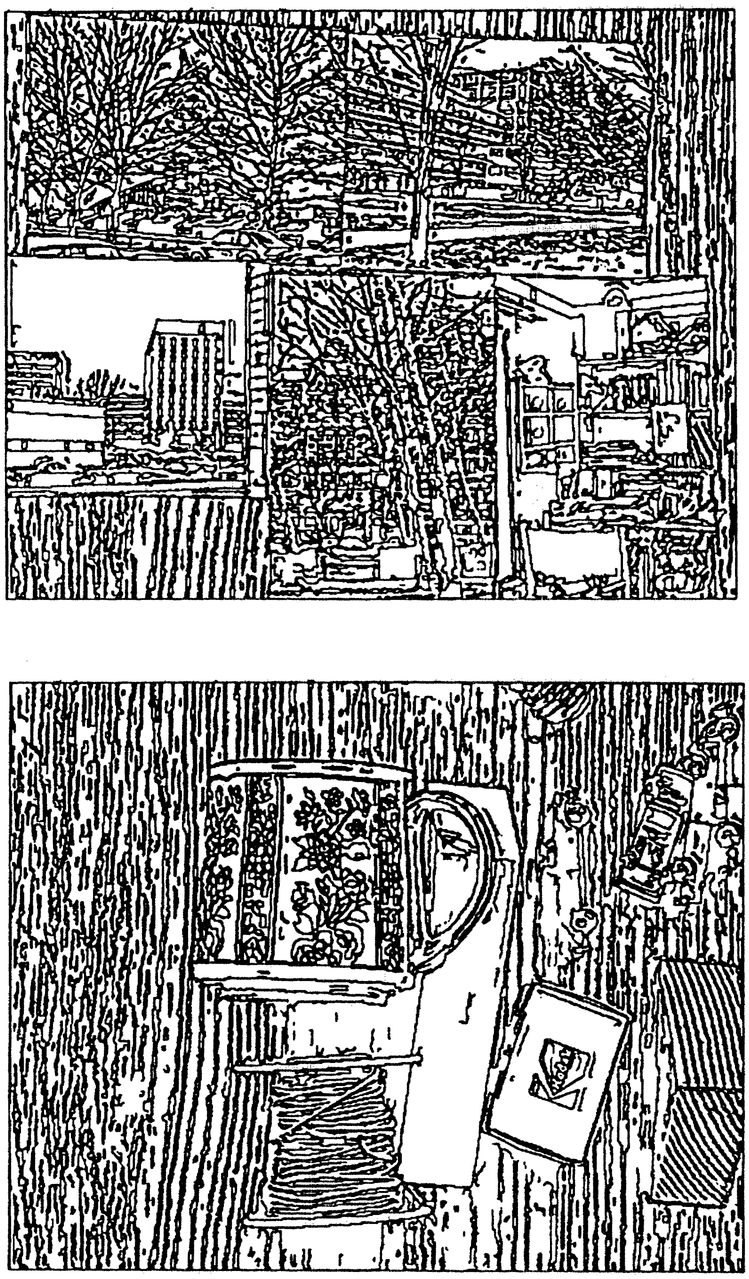


Figure 5. Phantom edge finder results for the two images in Figure 3, using noise threshold 60.

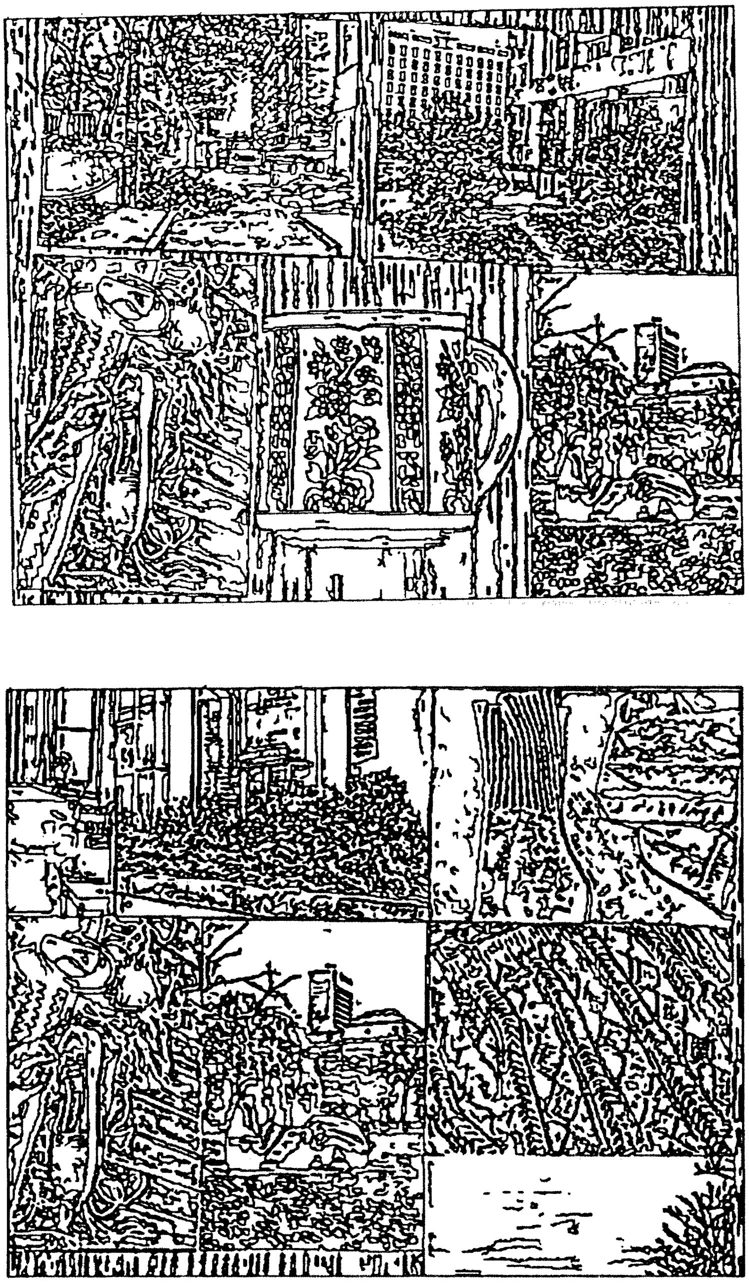


Figure 6. Phantom edge finder results for the two images in Figure 4, using noise threshold 60.

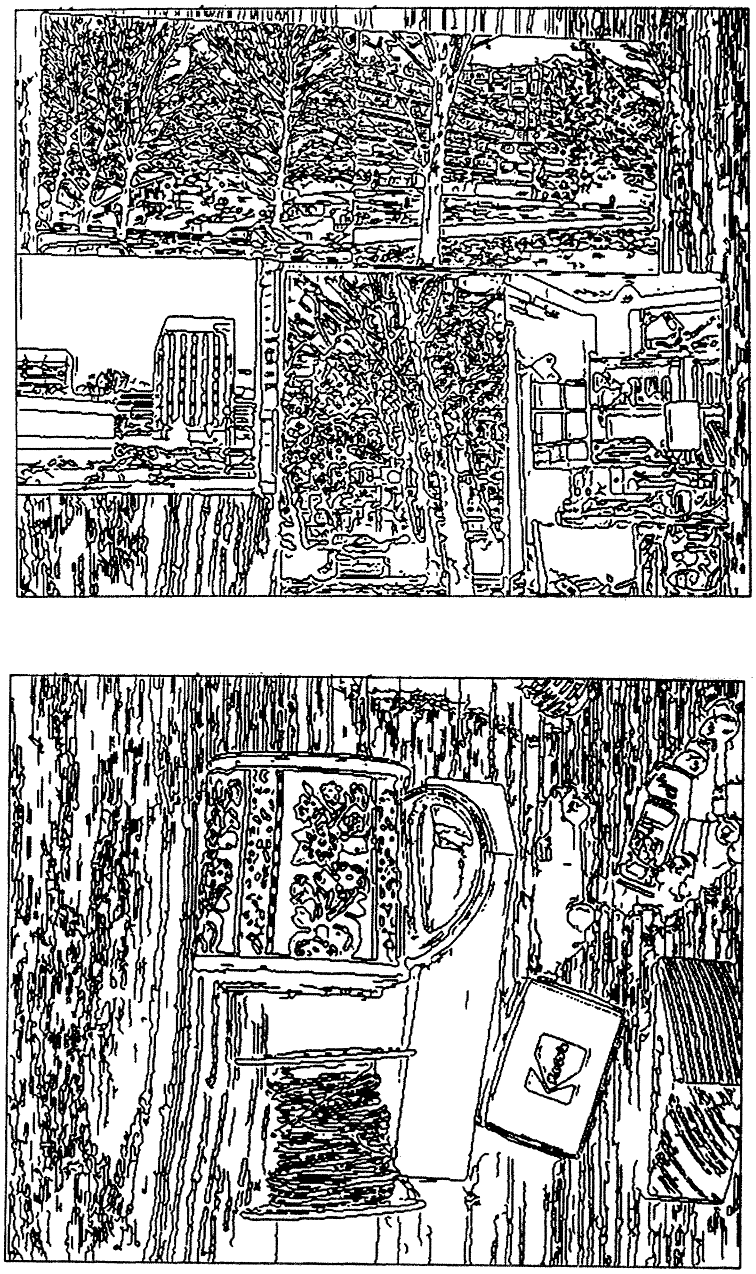


Figure 7. Canny edge finder results for the images in Figure 3, mask size 4, low noise threshold 150.

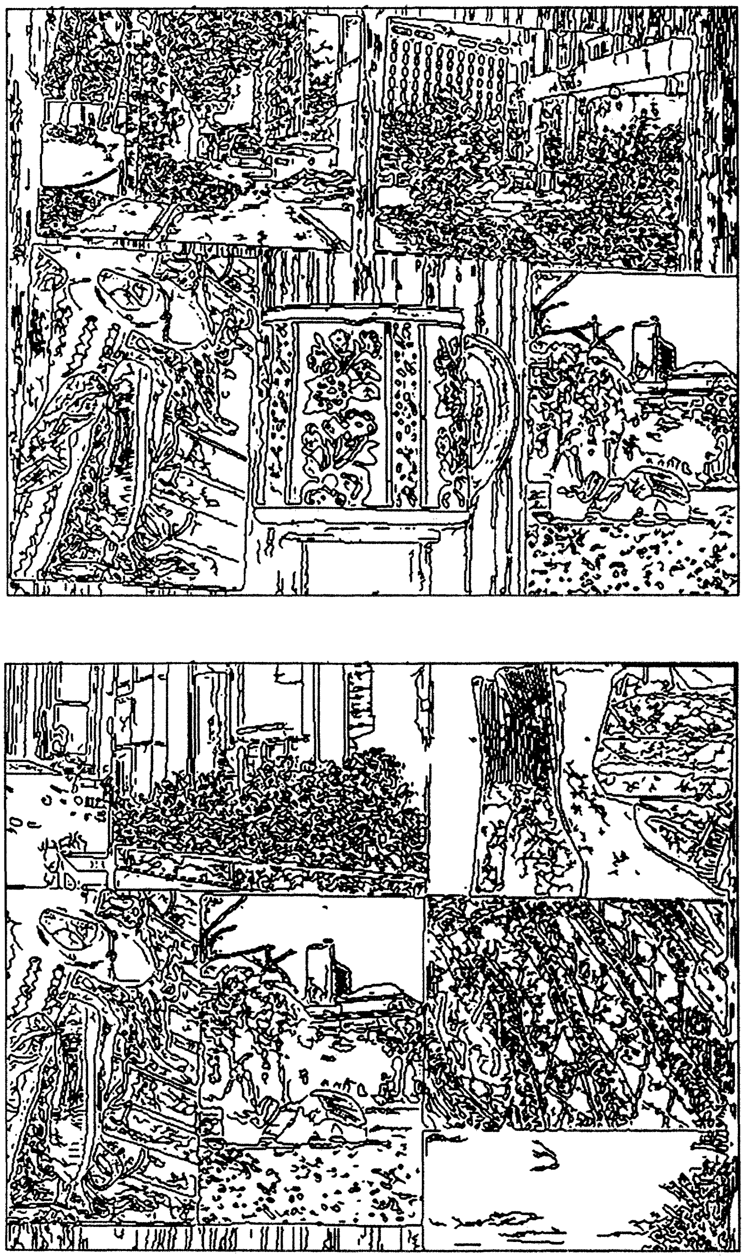


Figure 8. Canny edge finder results for the images in Figure 4, mask size 4, low noise threshold 150.

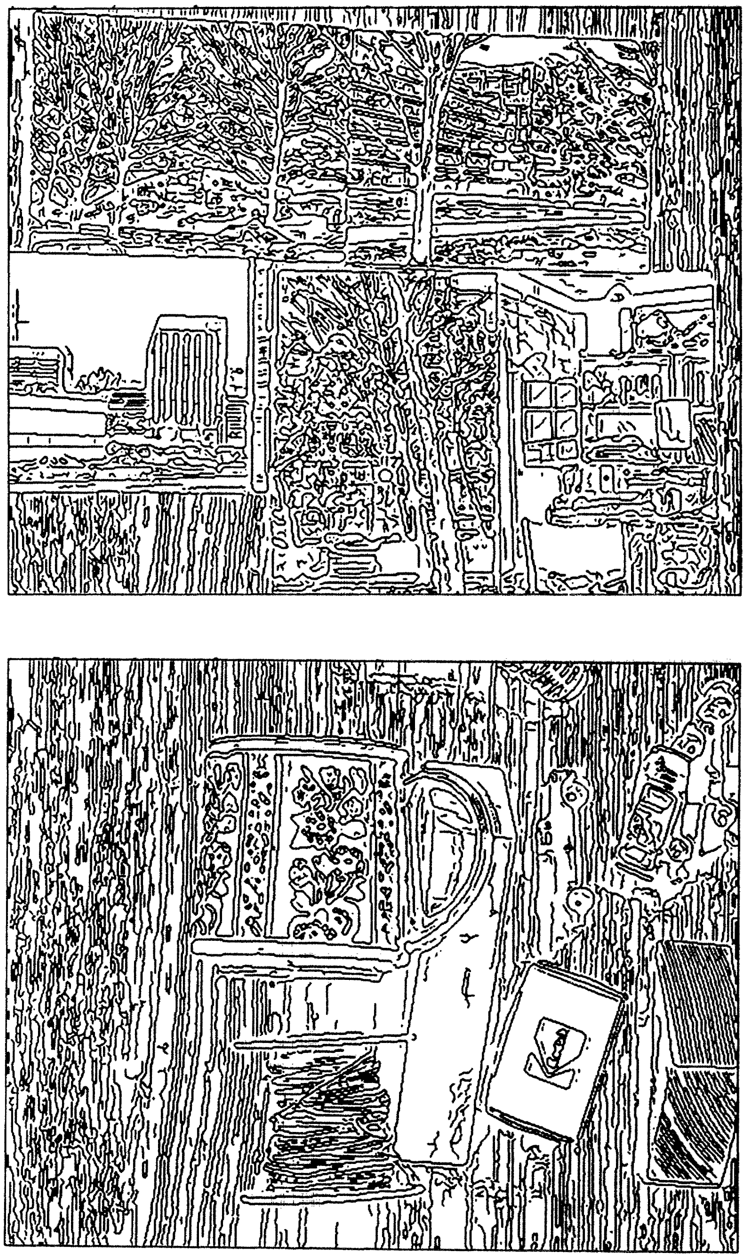


Figure 9. Canny edge finder results for the images in Figure 3, mask size 8, low noise threshold 100.

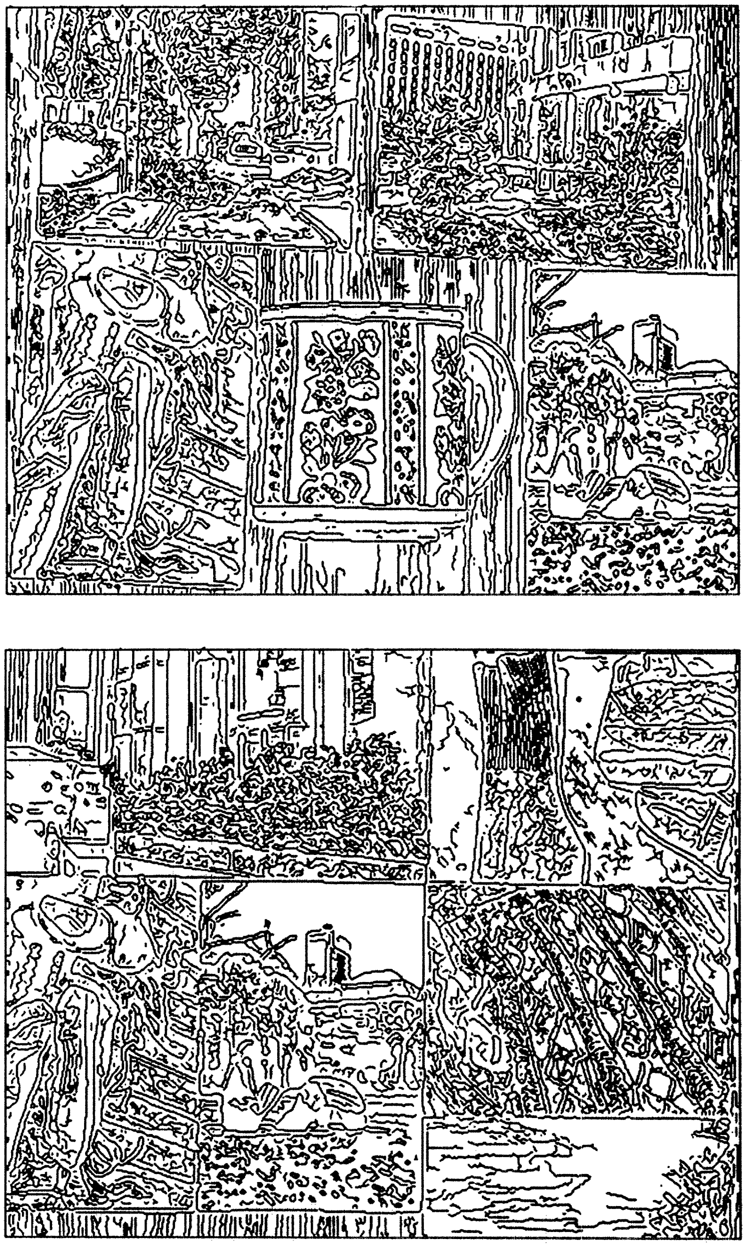


Figure 10. Canny edge finder results for the images in Figure 4, mask size 8, low noise threshold 100.

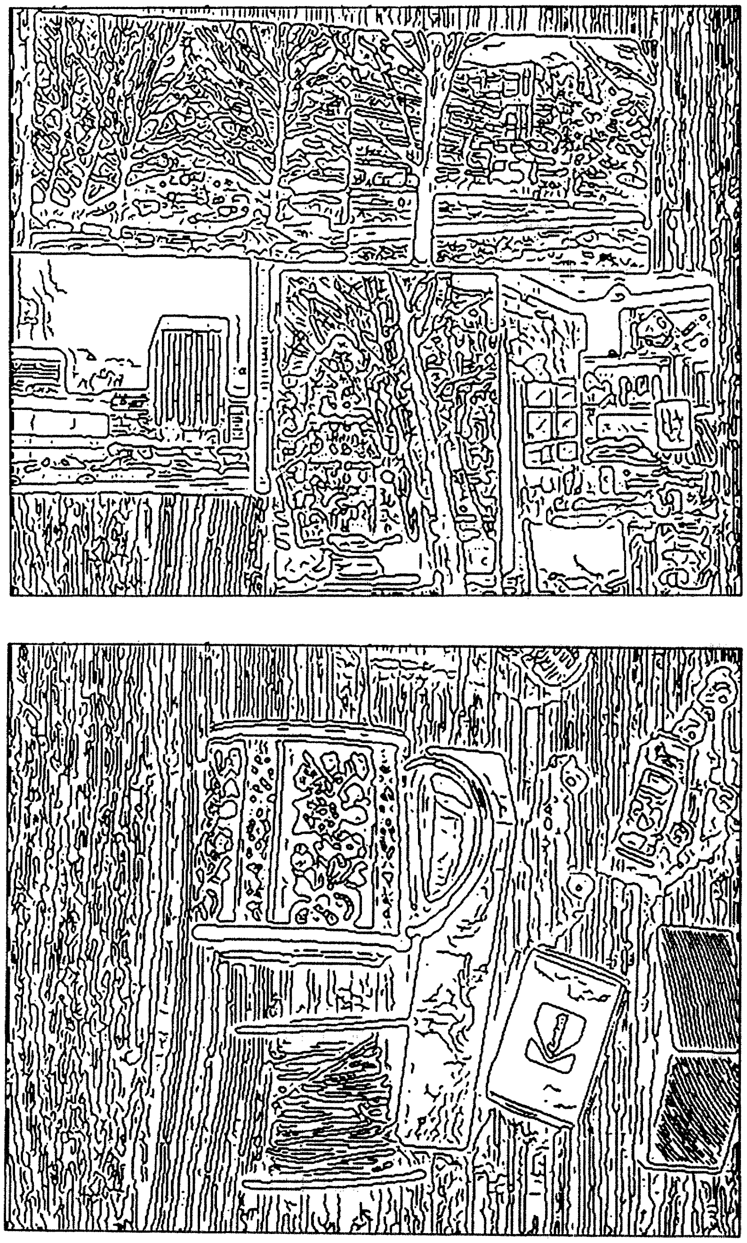


Figure 11. Canny edge finder results for the images in Figure 3, mask size 12, low noise threshold 50.

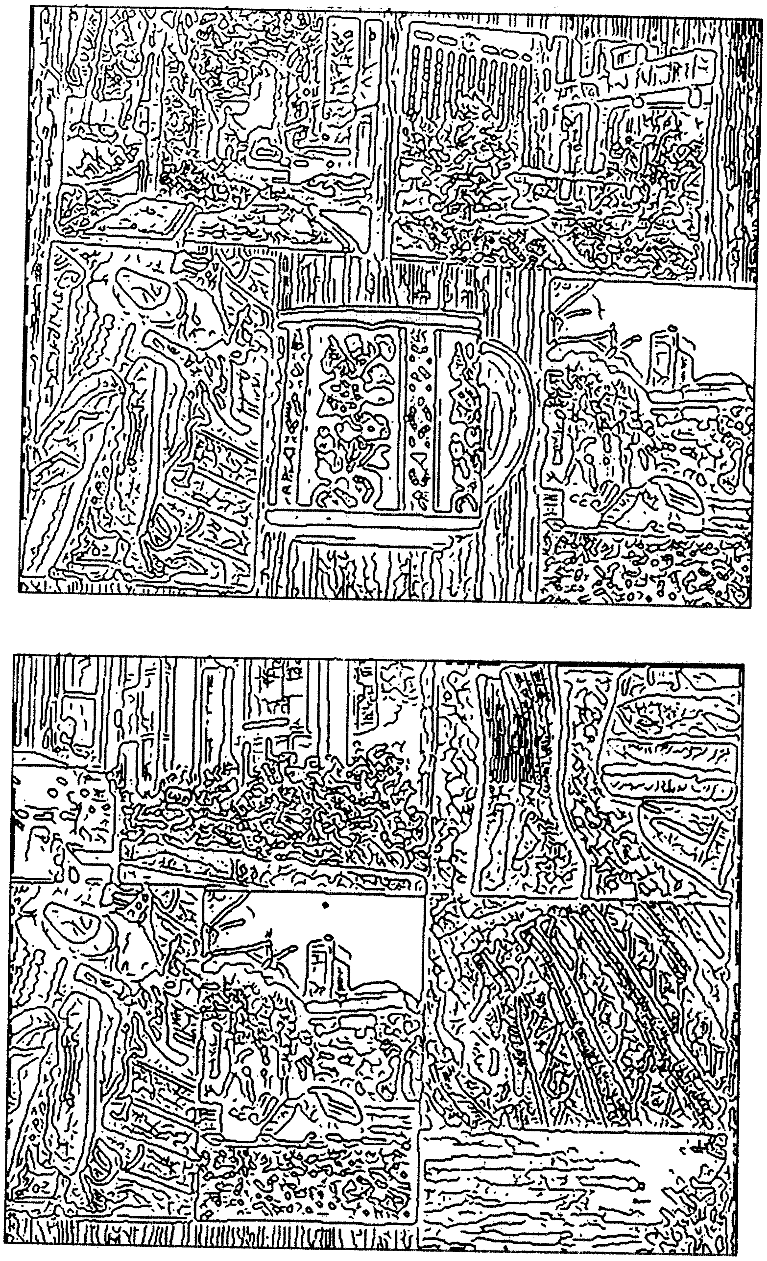


Figure 12. Canny edge finder results for the images in Figure 4, mask size 12, low noise threshold 50.

100, 150, 200, 250, and 300. The results presented are averages over all four images. The results for individual images follow the same qualitative pattern, with variations in exact values due to differences in scene content.

Figures 13-14 show graphs of the percentage of the image not matched and the percentage edges for the Phantom edge finder. As the noise threshold is increased, the stability increases steeply and then levels off, whereas the number of edges starts out level and then drops. The exact height of these curves depends on the contents of the scene. For example, if the scene contains large regions of uniform color or low-amplitude boundaries, only small percentages of the image match at low noise thresholds. If, on the other hand, most of the scene consists of high-contrast texture, the non-matching percentages are all low. Thus, the important properties of these curves are their shape and the changes in their height as parameters are varied.

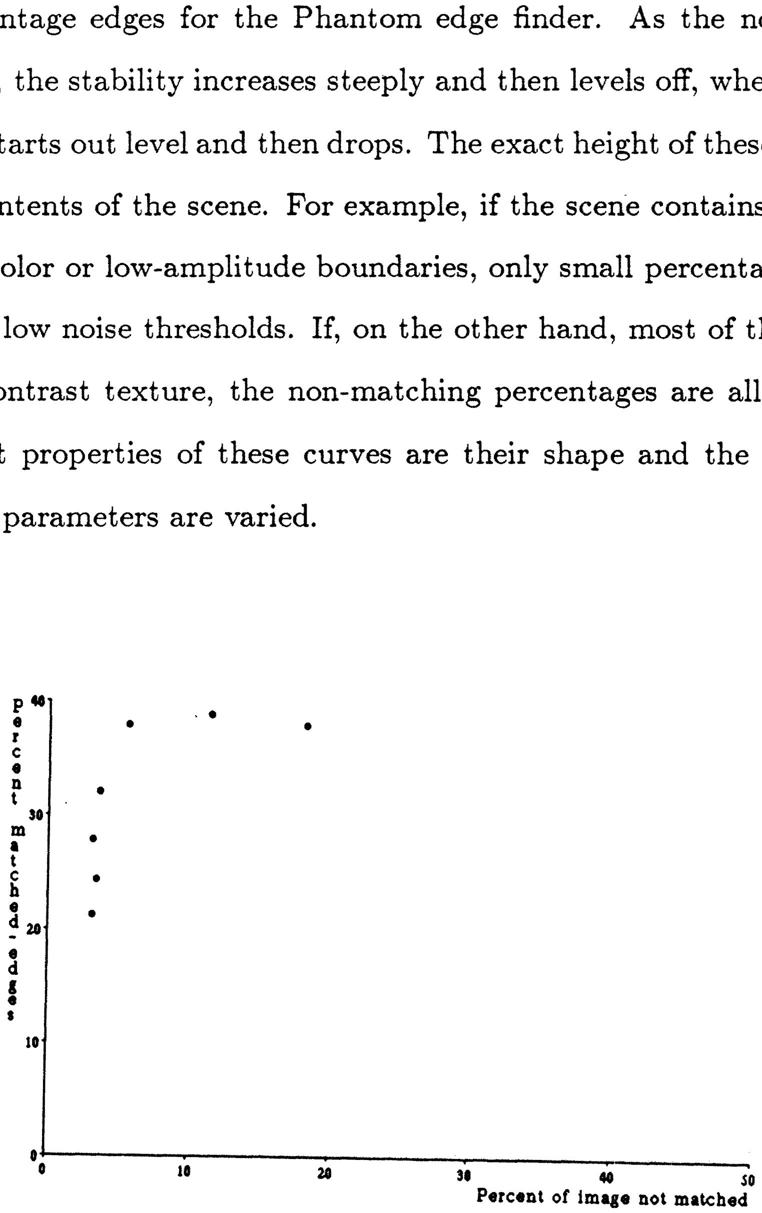
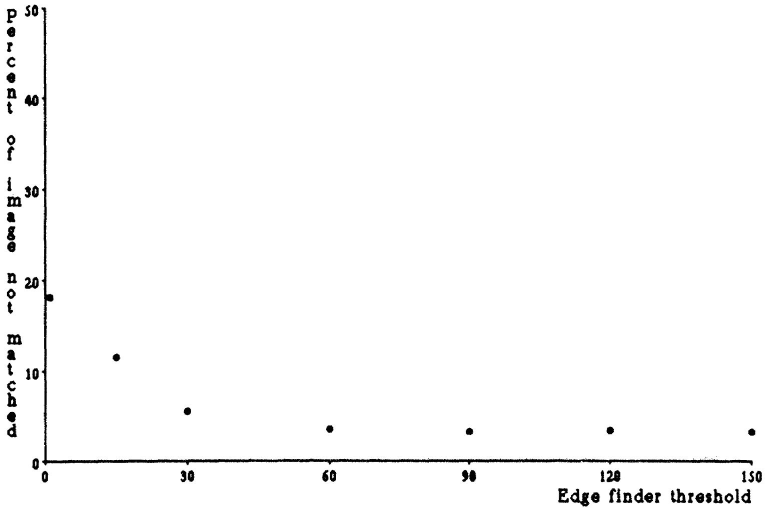


Figure 14. The matching percentages and the edge percentages from Figure 13, plotted against one another. Settings near the bend of the curve represent good compromises between stability and returning as many boundaries as possible.



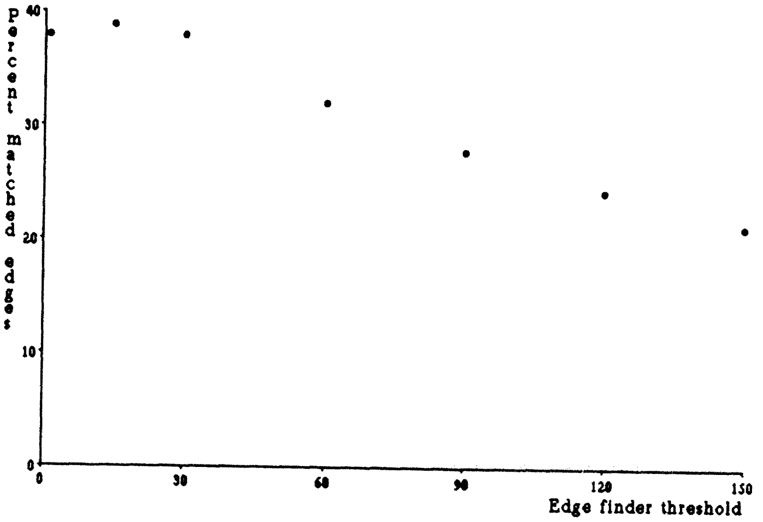
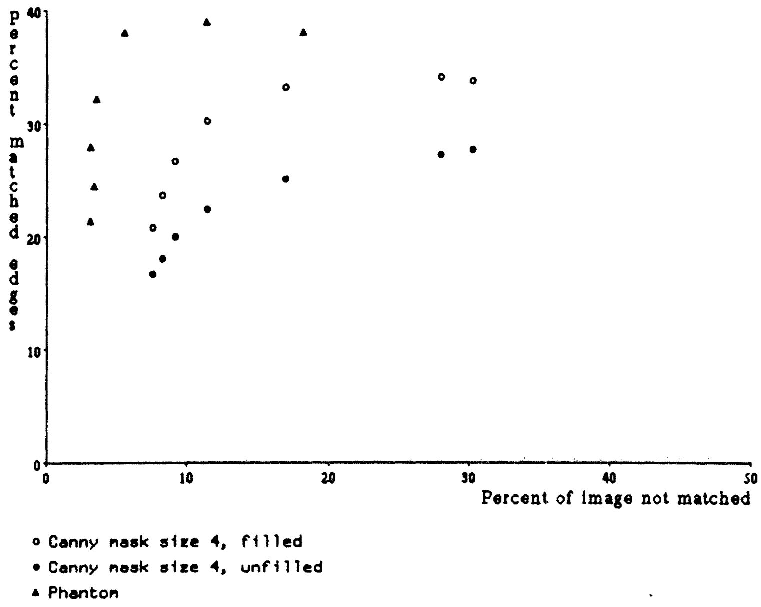


Figure 13. Top: percentage of the Phantom edge finder's output not matched (i.e. contaminated by noise), plotted as a function of the edge finder's noise threshold. Bottom: a similar plot of the number of successfully matched edges, normalized for the image size. Both of these graphs represent averages over the four images shown in Figures 3 and 4.

For comparative purposes, it is most useful to plot non-matching percentage against edge percentage, as shown in the last graph in Figure 14. This format makes explicit the tradeoff between stability and number of edges detected. It also allows results from different edge finders to be plotted in a common coordinate system. The noise threshold settings are not explicitly shown in this type of graph. However, since the points form well-behaved curves, the mapping between data points and thresholds can easily be deduced. Notice that the evaluation results for Phantom form an L-shaped curve. Noise threshold settings near the bend in this curve (about 30-60) represent good tradeoffs between stability and number of boundaries. These settings also seem the best when outputs are inspected visually.

Figures 15-16 show graphs comparing the evaluations of Phantom and Canny's edge finder. Notice that the curves for Canny's edge finder at mask sizes 4 and 8 lie entirely below the curve for the Phantom edge finder. This is true no matter which method is used for counting edge cells in Canny's output. This indicates that, for any desired degree of stability, Phantom is returning more boundaries than Canny's edge finder. If Canny's edge finder is run with mask size 12, the curves are closer, although Phantom is still performing better. The difference is most pronounced near the bends in the curves, where the best tradeoffs between stability and number of boundaries occur.

The amount of boundary motion was computed for all of these parameter settings. The average amount of motion ranged between 0.17 and 0.25 cells for the Phantom edge finder. For Canny's edge finder, it was 0.16 to 0.26. The amount of motion was lower for higher noise thresholds and (for Canny's edge finder) larger mask sizes. Since the total range of variation is so small, however, it is unclear how reliable these differences are.



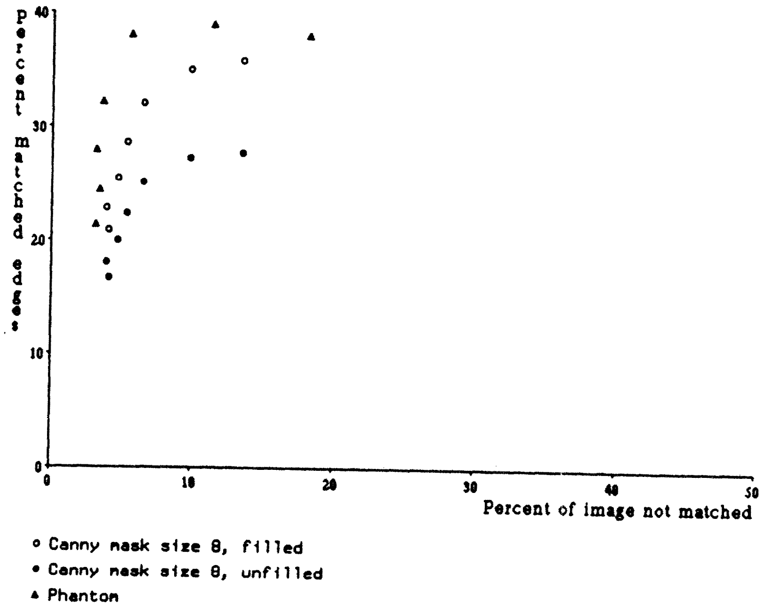


Figure 15. Performance of Canny's edge finder (mask sizes 4 and 8) compared with that of the Phantom edge finder, on the main noise test (average of the four images). These graphs show matching percentages plotted against edge percentages. Values that are higher and to the left represent better compromises between stability and returning as many boundaries as possible.

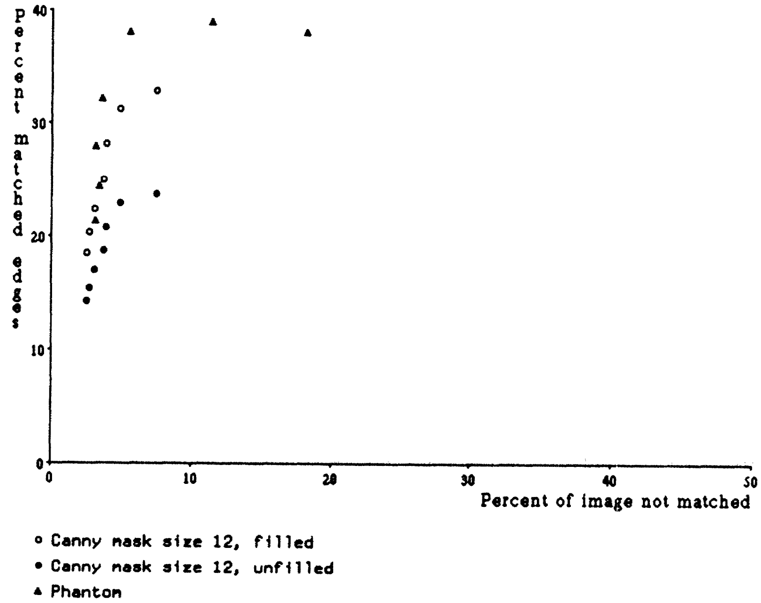


Figure 16. The same comparison as in Figure 15, but with mask size 12 of Canny's edge finder.

Thus, these tests show that the Phantom edge finder produces a better tradeoff between the number of boundaries reported and the stability of this output. Canny's edge finder performs noticably less well when mask size 4 is used. Performance is better with mask size 8 and approaches that of the Phantom edge finder for mask size 12. The average amount of boundary motion is small and is approximately the same for the two edge finders. These results are for moderate amounts of camera noise, such as one might expect from modern video camera setups.

5. Other noise tests

In addition to the main noise evaluation described in Section 4, two shorter tests were run to see how the results changed under slightly different conditions.

One test repeats the same evaluation on an image with higher levels of camera noise. The other test uses pairs of images in which the scene has been translated relative to the camera, so that the scene is digitized in different ways. Results from the translation examples are similar to those from the main noise test, though the amount of boundary motion is higher. In the high noise case, Phantom performs similarly to Canny's edge finder with mask size 8. With smoothing, Phantom performs similar to Canny's edge finder with mask size 12.

The high noise test used one pair of 454 by 576 images, similar to those described in Section 4. This image, however, was taken with a camera and digitizer that introduce higher-amplitude noise than the ones used for the main test. The constrast range in this picture was also low, only 161 intensity units. One of this pair of images is shown in Figure 17 and Figures 18-19 show edge finder output for this image. As in Section 4, these results represent good settings for the noise thresholds, somewhat higher than the best settings for the images used in the main noise test. The Phantom edge finder was run through two evaluations, once on the raw images and once after smoothing the images with a Gaussian of $\sigma = 1$ cell.

The evaluations for Canny's edge finder were run for mask sizes 8 and 12 and low noise thresholds of 0, 50, 100, 150, 200, 250, and 300. The Phantom edge finder was run with thresholds 1, 30, 60, 90, 120, and 150 for both the smoothed and unsmoothed cases. To fill in sparse sections of the graph, it was also run with threshold 75 in the unsmoothed case and threshold 45 in the smoothed case. Figure 20 shows the results of these stability evaluations. On this image, Phantom exhibits performance similar to that of Canny's edge finder with mask size 8. With smoothing, its performance is similar to that of Canny's edge finder with mask size 12. Note that the scale is different for these two graphs, because

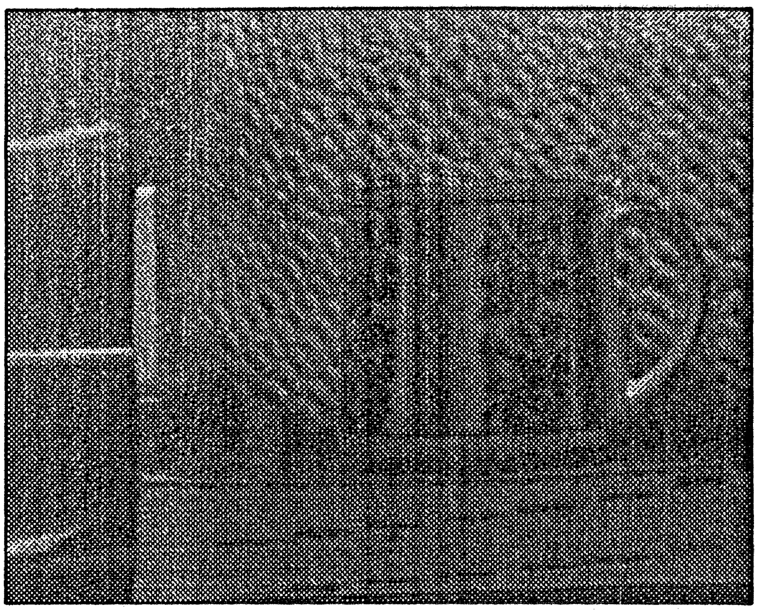


Figure 17. The image used in the high noise test.

two of the data points for the unsmoothed version of Phantom lie quite far out.

For the high noise image, boundary motions for the two edge finders are quite similar. With the exception of one measurement, boundary motions computed for the two Phantom evaluations fall in the range of 0.43 to 0.59. The odd measurement is 0.33, but it comes from the unsmoothed Phantom evaluation with threshold setting 1 and is probably corrupted by noise. Boundary motions computed for Canny's edge finder vary between 0.48 and 0.57. These numbers are noticably higher than those for the lower noise images, but, again, there seems to be no significant difference in performance between the two edge finders.

The second short test used two pairs of images in which the scene was translated relative to the camera. These images were created by placing a black-and-white photograph on a background of white paper. The photograph was moved by hand in a diagonal direction (with respect to the camera digitization), using

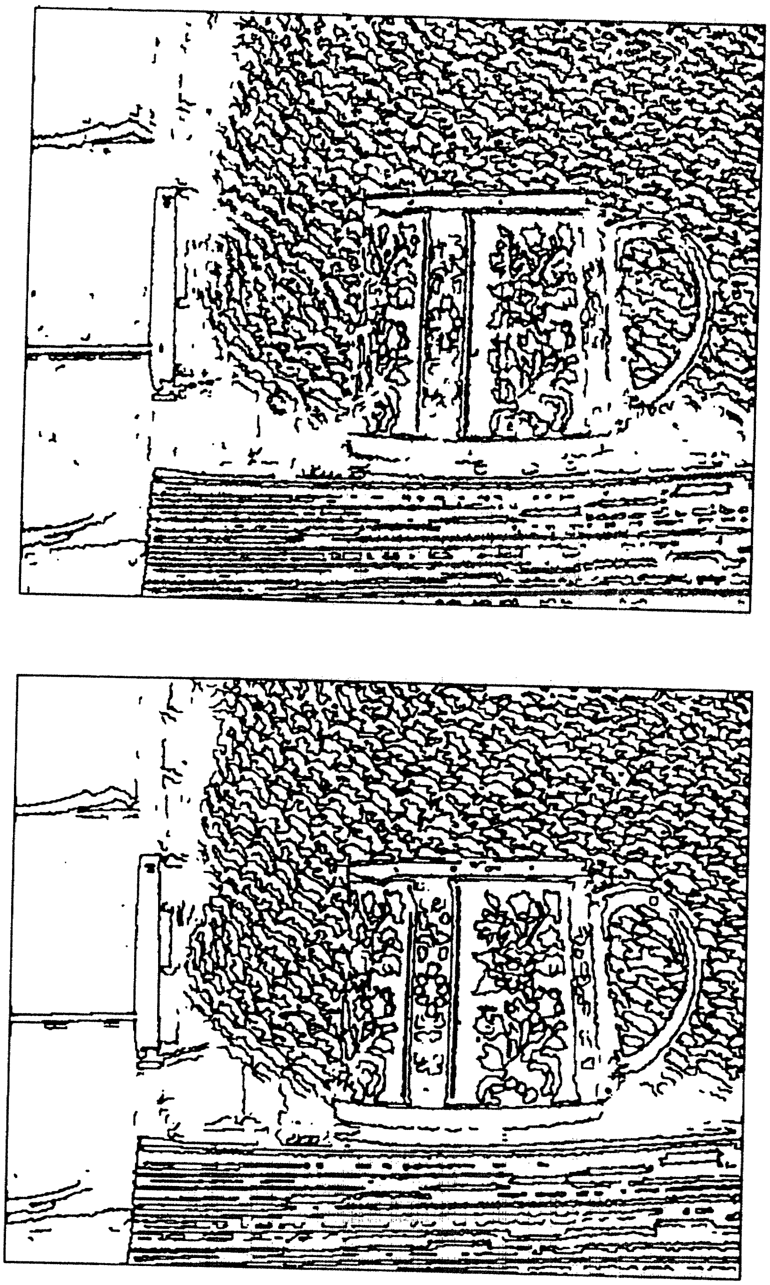


Figure 18. Edge finder results for the high noise image from Figure 17. Top: Phantom edge finder output (threshold 90). Bottom: finder with pre-smoothing (threshold 60).

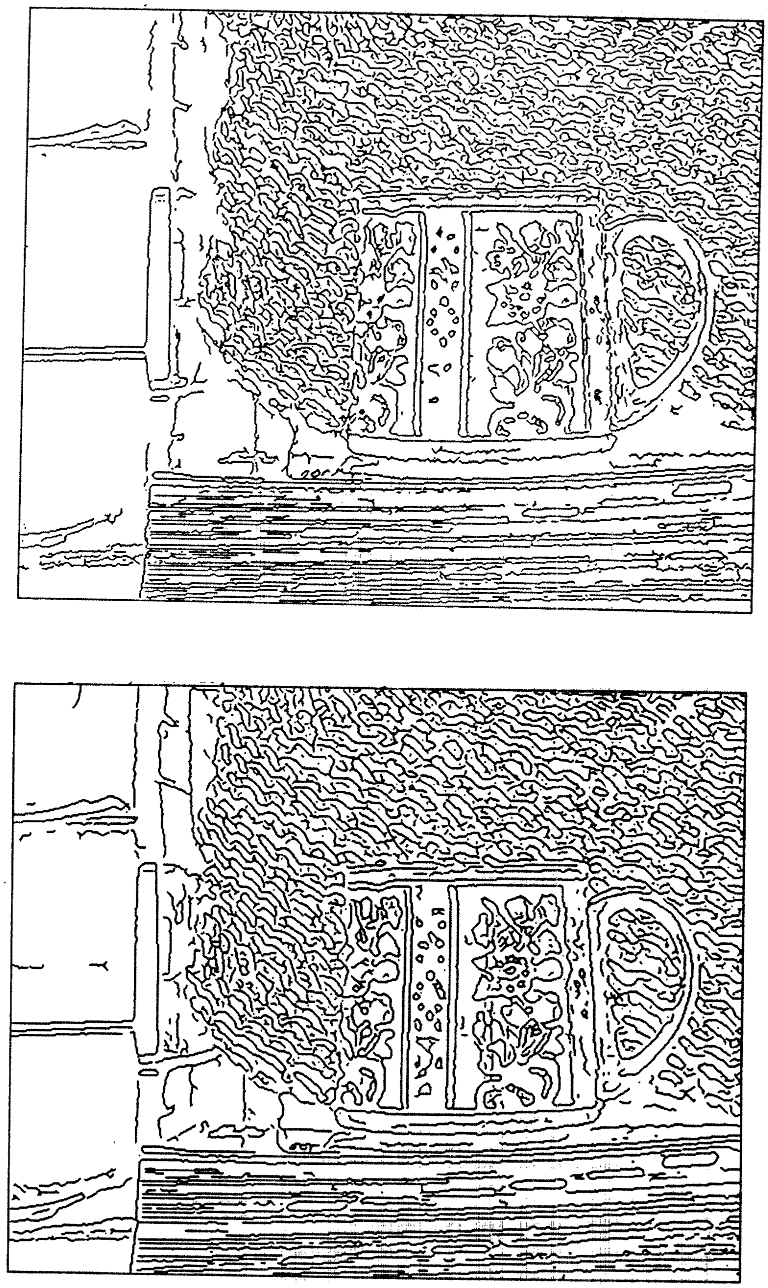


Figure 19. Edge finder results for the high noise image from Figure 17. Top: Canny's edge finder with mask size 8 (threshold 150). Bottom: Canny's edge finder with mask size 12 (threshold 100).

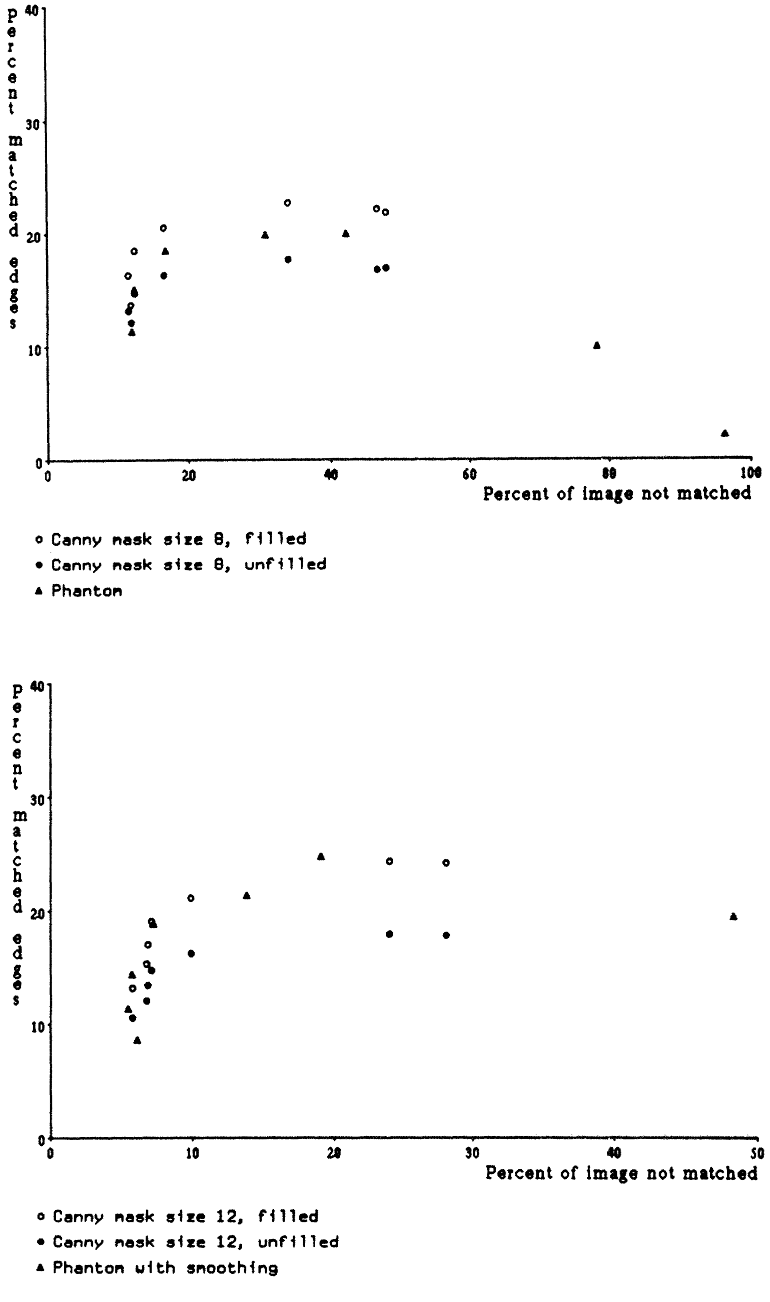


Figure 20. Performance of Phantom edge finder and Canny's edge finder on the high noise image from Figure 17. These graphs show the percentages matched against the edge percentages, as in Figures 15 and 16. The upper graph shows the results for Phantom without pre-smoothing and the lower graph shows results with pre-smoothing. Also, the scale used for matching percentages is different for the two graphs, because the results for the unsmoothed version of Phantom generate one point with a very low matching percentage.

a paper edge as a guide. Although the paper did not move, the only boundaries visible in the paper background lie parallel to the direction of translation. The images were aligned by hand (using easily identified features in edge finder output) and corresponding 224 by 288 sections extracted. Evaluation was then done on these pairs of images in the same way as in Section 4. One of each pair of images is shown in Figures 21-22, together with edge finder results.

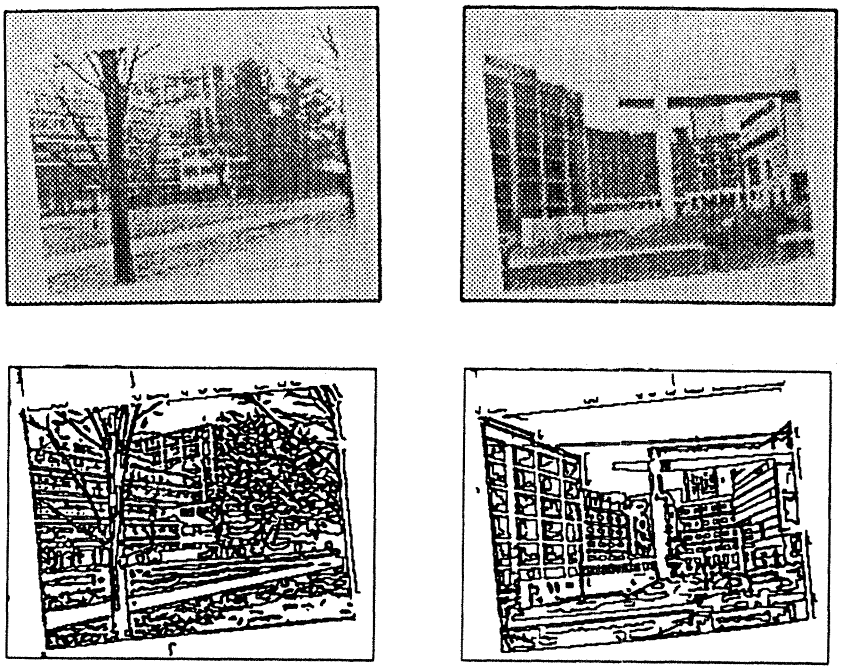


Figure 21. Top: The two images used in the translation experiment. Bottom: Phantom edge finder output (threshold 60).

Evaluations for the translated images were run for three Canny mask sizes, 4, 8, and 12, and low noise thresholds 0, 50, 100, 150, 200, 250, and 300. The Phantom edge finder was evaluated with thresholds 1, 30, 45, 60, 90, 120, and 150. The evaluation results are shown in Figures 23-24. These evaluations show roughly the same pattern as those presented in Section 4. The datapoints are

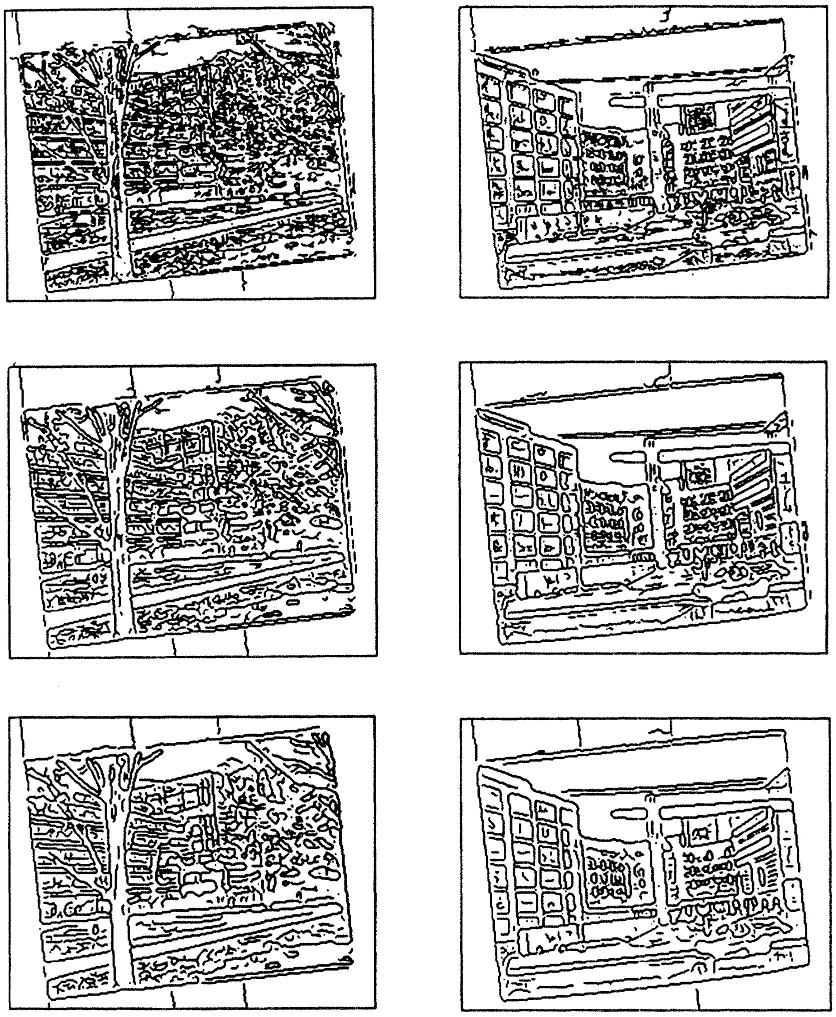


Figure 22. Canny edge finder output for the translation images. Top to bottom: mask size 4, mask size 8, and mask size 12. Noise threshold 100 was used for all these images.

somewhat more noisy, probably because less image area was used. The best threshold settings for Canny's edge finder are also slightly different. Boundary motion estimates for Phantom are between 0.24 and 0.31, and between 0.29 and 0.36 for Canny's edge finder. This is slightly higher than the motions computed for the non-translated images in Section 4, but not dramatically so.

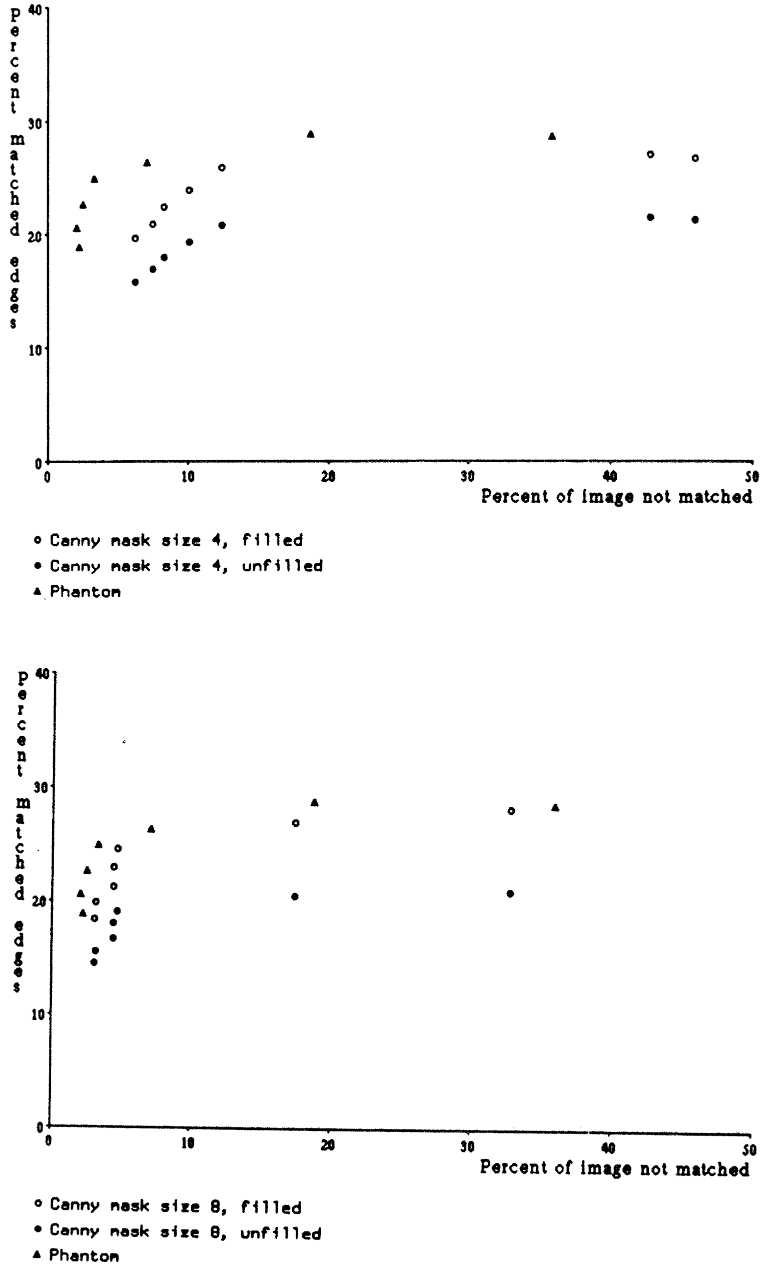


Figure 23. Plots of matching percentage vs. edge percentage for the two translation images in Figure 21. These plots compare the performance of Canny's edge finder (mask sizes 4 and 8) against that of the Phantom edge finder.

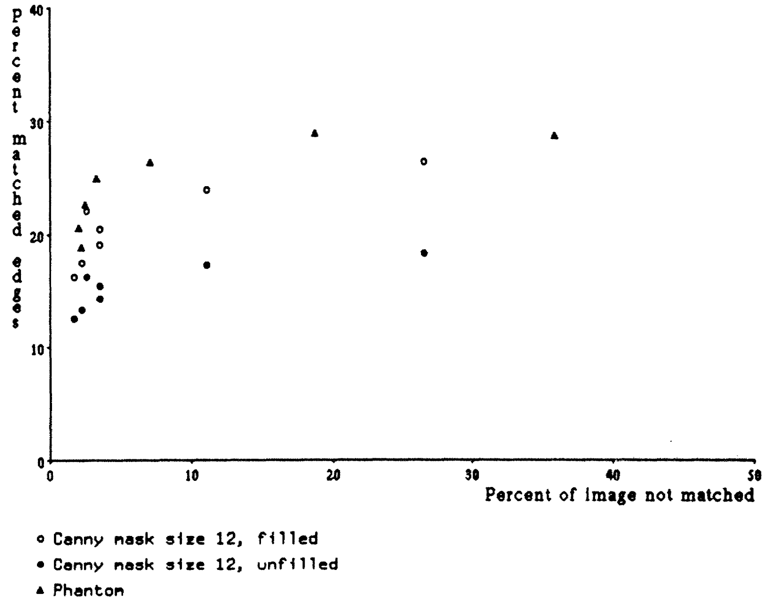


Figure 24. The same graph as in Figure 23, but using mask size 12 of Canny's edge finder.

The results for the translation and high noise conditions follow roughly the same pattern as those presented in Section 4. In the translation condition, the Phantom edge finder still performs slightly better than Canny's edge finder. In the high noise case, the Phantom edge finder loses its stability advantage, performing only as well as Canny's edge finder. Intuitively, however, there is still a difference in resolution. In the high noise example, small amounts of presmoothing improve Phantom's performance noticably. In both tests, computed boundary motions were predictably higher than those found in Section 4, but again show little difference between the two edge finders.

6. Acuity tests

In the examples shown in previous sections, there seemed to be a difference in resolution between Canny's edge finder and the Phantom edge finder, even where the two edge finders had similar stability evaluations. In this section, we see that this difference in resolution does not primarily involve differences in ability to detect small regions, but rather differences in ability to parse regions with high boundary curvature and boundary intersections. We also see several other qualitative differences in behavior between the two algorithms. The Phantom edge finder generates spurious boundaries in staircase patterns. Canny's edge finder generates spurious responses on ramps and, at small mask sizes, picks up differences between interlaced scan lines.

The examples in this section are of two types: synthetic images and natural images. Each synthetic pattern was generated at a range of contrasts (128, 103, 78, 52, and 26 intensity units difference between the darkest and the lightest values). Each image in the series was blurred with a Gaussian of $\sigma = 1$ cell and Gaussian noise of $\sigma = 3$ intensity units was added. This is a rough simulation of the effect of the camera setups used in taking the natural images. For each image, Canny's edge finder is run with mask sizes 4 (threshold 150), 8 (threshold 100), and 12 (threshold 50). The Phantom edge finder is run with noise threshold 60. The noise thresholds were chosen on the basis of the results presented in Section 4. Edge finder output is displayed using boundary maps that are twice the size of the original images, so that fine detail of the boundaries can be seen. Images are also shown at twice normal size.

The Phantom edge finder and Canny's edge finder show similar ability to detect thin regions, such as the thin-bar patterns shown in Figure 25. Both edge finders can detect regions as small as two cells wide, although these regions are beginning to be difficult for mask size 12 of Canny's edge finder. Neither edge finder can reliably detect regions one cell wide in the presence of even the moderate amount of blur and noise used in these experiments. Thus, they both have approximately the same separation acuity as humans (Marr, Poggio, and Hildreth 1980).

The real differences in resolution between the two edge finders involve boundaries with high curvature and boundary intersections. As we saw in Chapter 4, the gradient direction used by Canny's edge finder is not well-defined in these cases. I consider the high curvature problem first. On high-curvature boundaries, Canny's edge finder exhibits three failure modes: smoothing corners, breaking corners, and adding spurious boundaries. Figures 26-27 show corners of varying sharpness, illustrating all three types of failures. The results for the highest contrast example in Figure 26 are also shown enlarged, so that the small breaks are easier to see.[6] In dense texture, strange patterns of extraneous boundaries often occur, as shown in Figure 29.

Canny's edge finder also exhibits problems when boundaries intersect. Figures 29-30 show two patterns in which multiple regions meet at a common point. As you can see, the Phantom edge finder produces appropriate patterns of boundaries in these cases, whereas Canny's edge finder mangles the pattern near the intersections. Both high curvature and boundary intersections occur frequently in natural images. The difference in performance on these types of configurations accounts for the differences in apparent resolution between the two edge finders. Figures 31-33 show examples of sharp corners and boundary intersections in natural images.

The leaf image in Figure 30 illustrates another problem that occurs when Canny's edge finder is used without adequate smoothing. Our camera, like most cameras used in computer vision, uses interlaced frames. At mask size 4, Canny's

[6] Particularly after the repeated xeroxing to which this document may be subjected.

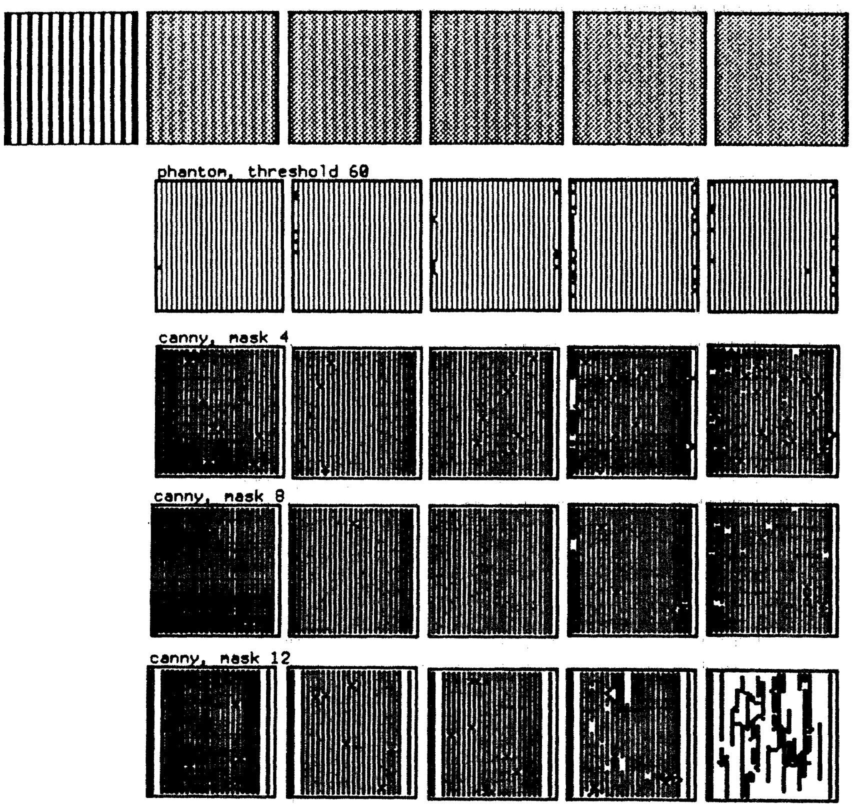


Figure 25. Performance of the Phantom edge finder and Canny's edge finder on a 50 by 50 image of thin bars (two cells wide). Both edge finders can resolve these bars, though Canny's edge finder becomes unreliable when mask size 12 is used. In this figure, and in the other figures in this section, images are enlarged by a factor of two (in each dimension) relative to the other images shown in this thesis.

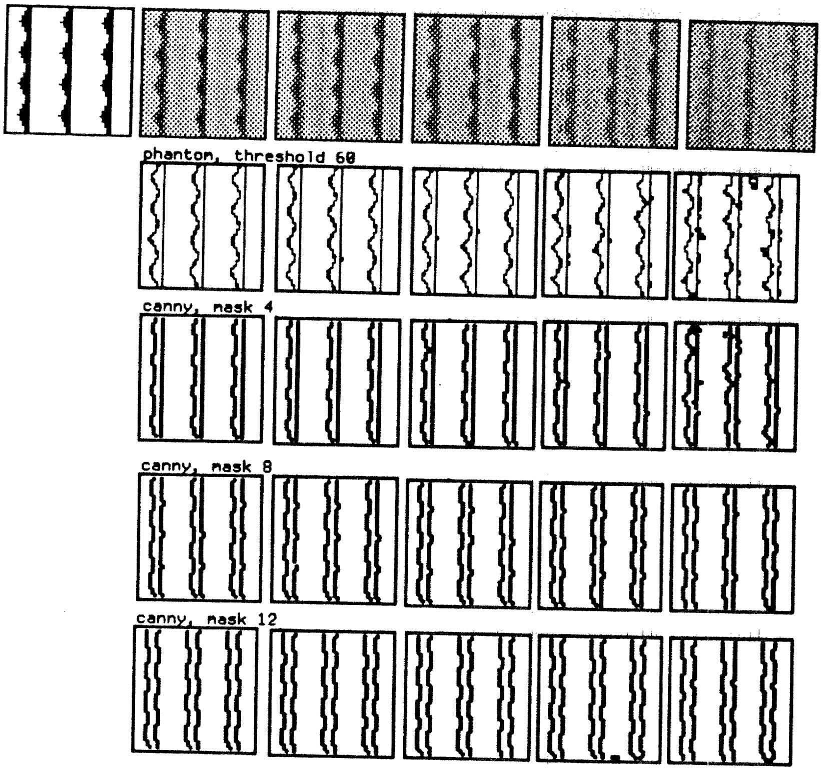


Figure 26. Canny's edge finder smooths the sharp corners in this 48 by 48 image.

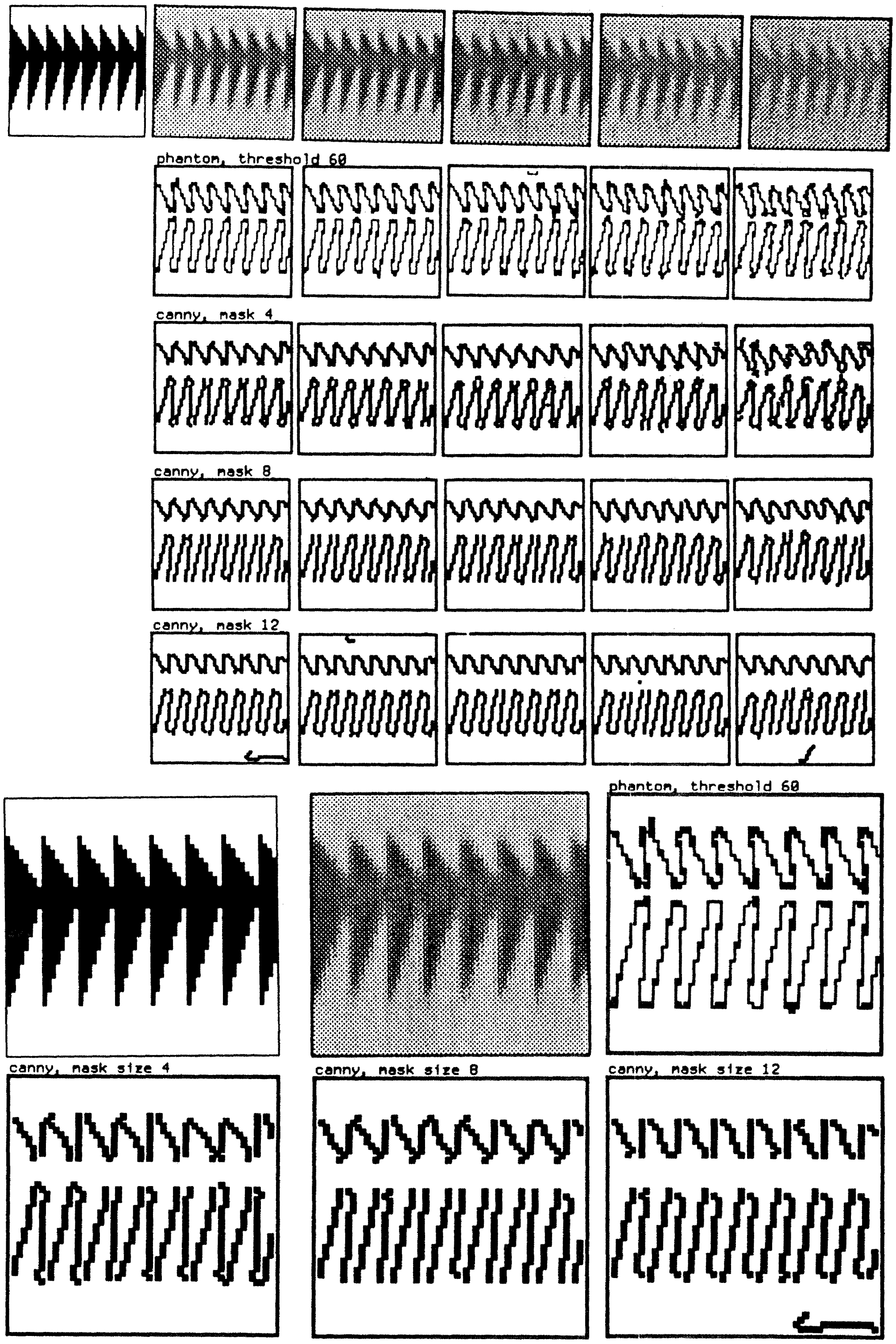


Figure 27. Canny's edge finder breaks these sharp corners and/or adds spurious boundaries to them. This image is 60 by 56 cells. The bottom figures show an enlarged version of the results for the highest contrast case (128 intensity units).

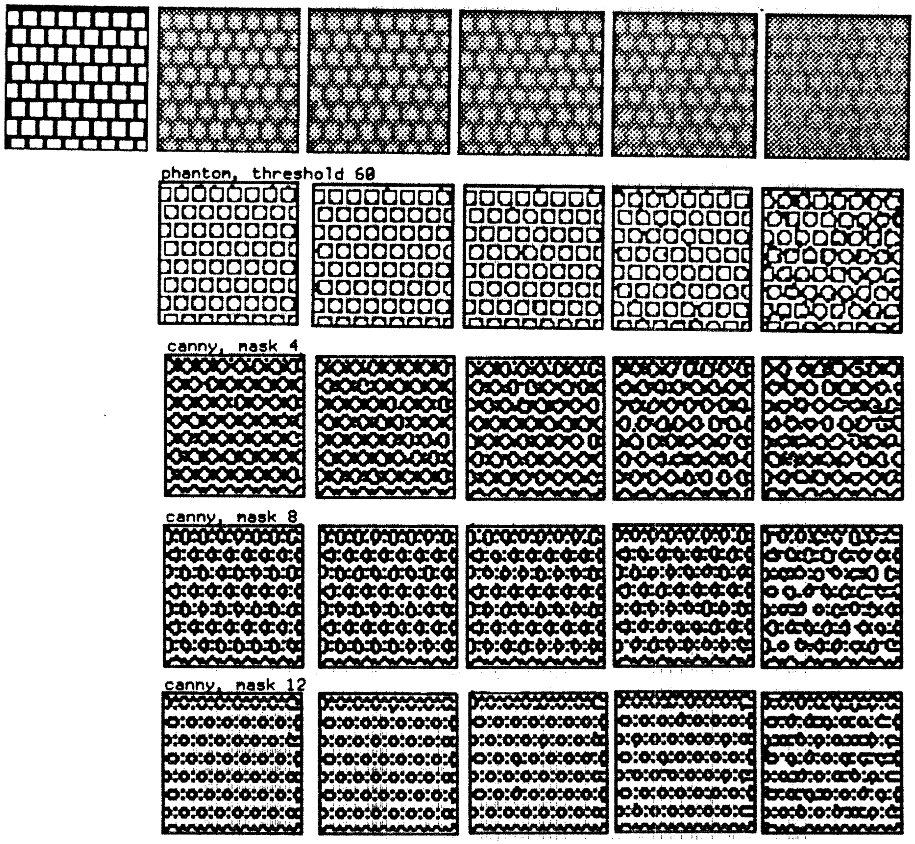


Figure 28. Performance of the two edge finders on a 54 by 54 image containing dense texture. Canny's edge finder generates spurious boundaries between regions.

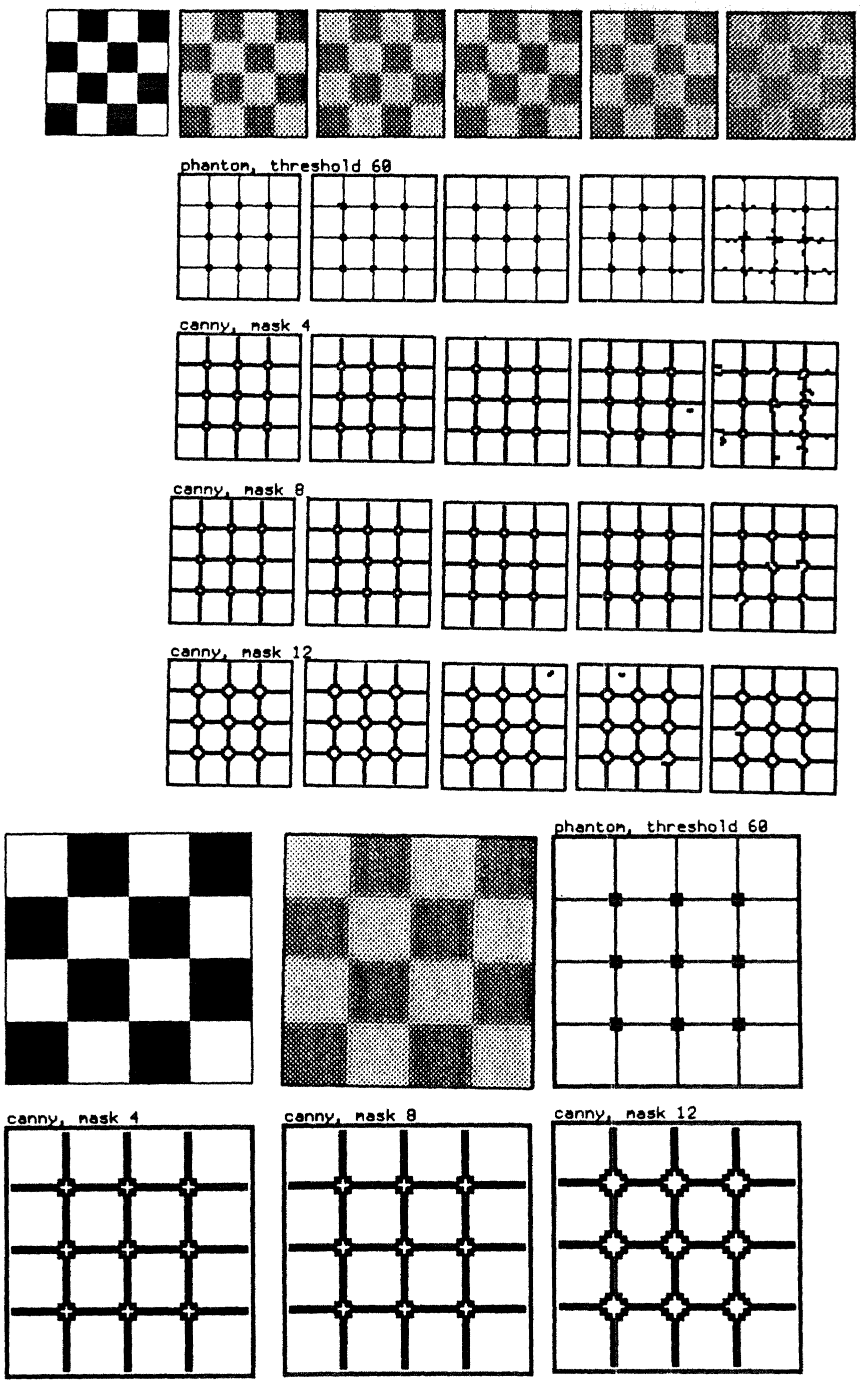


Figure 29. Canny's edge finder generates spurious regions near the boundary intersections in this 48 by 48 image. The bottom figures show an enlarged version of the results for the highest contrast case (128 intensity units).

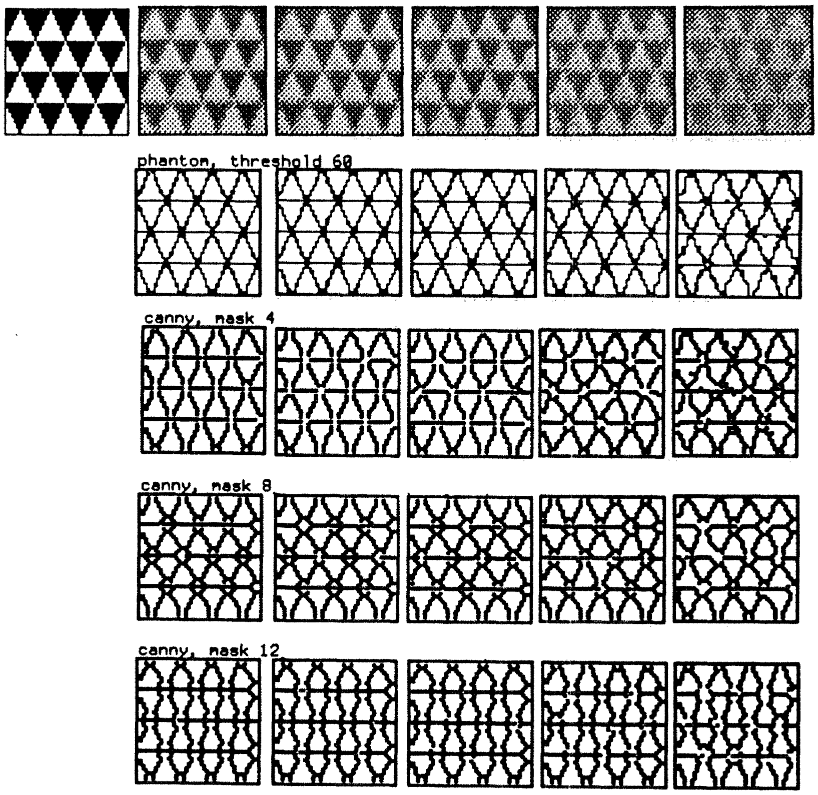


Figure 30. Another image (48 by 48) containing complicated boundary intersections. Canny's edge finder breaks the boundaries near the intersections.

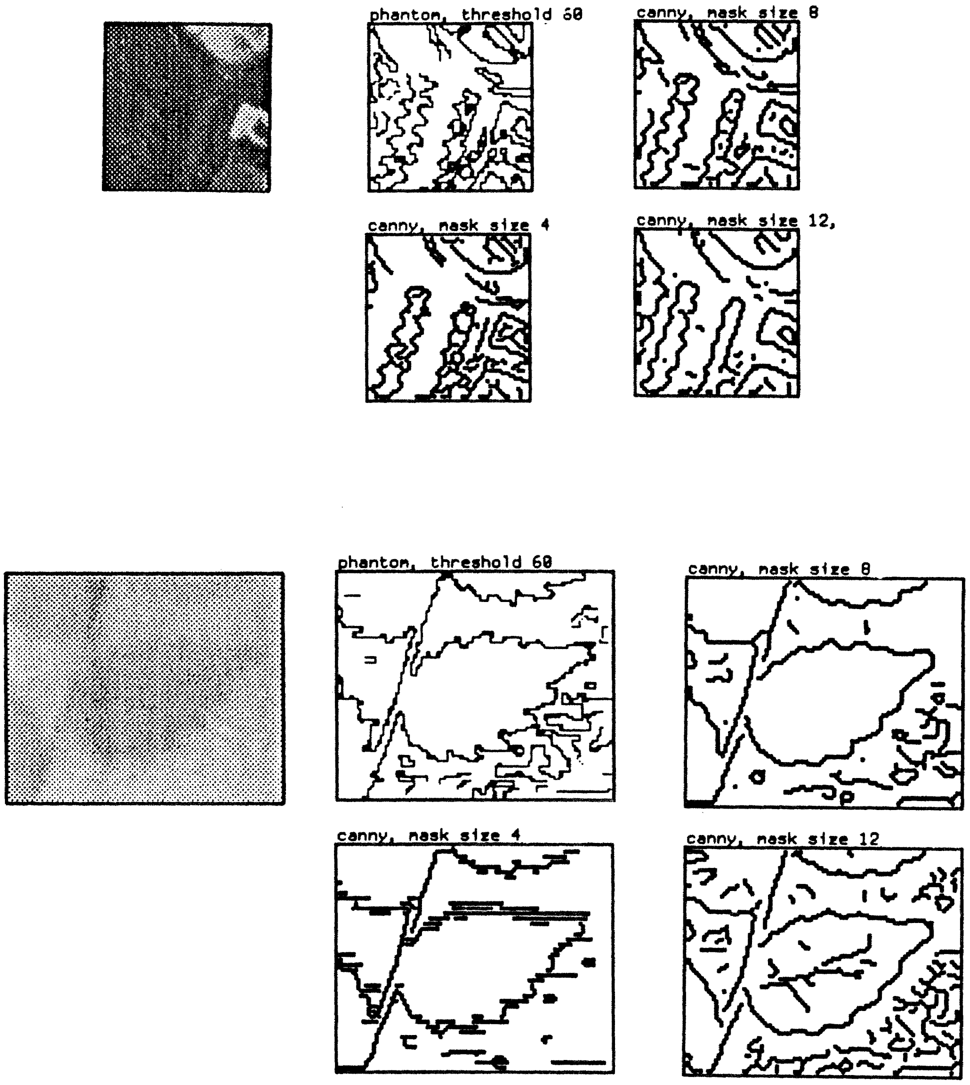


Figure 31. Two natural images containing sharp corners and boundary intersections (108 by 88 cells and 64 by 64 cells). Canny's edge finder smoothes sharp corners, breaks boundaries, and generates spurious boundaries near sharp corners and boundary intersections.

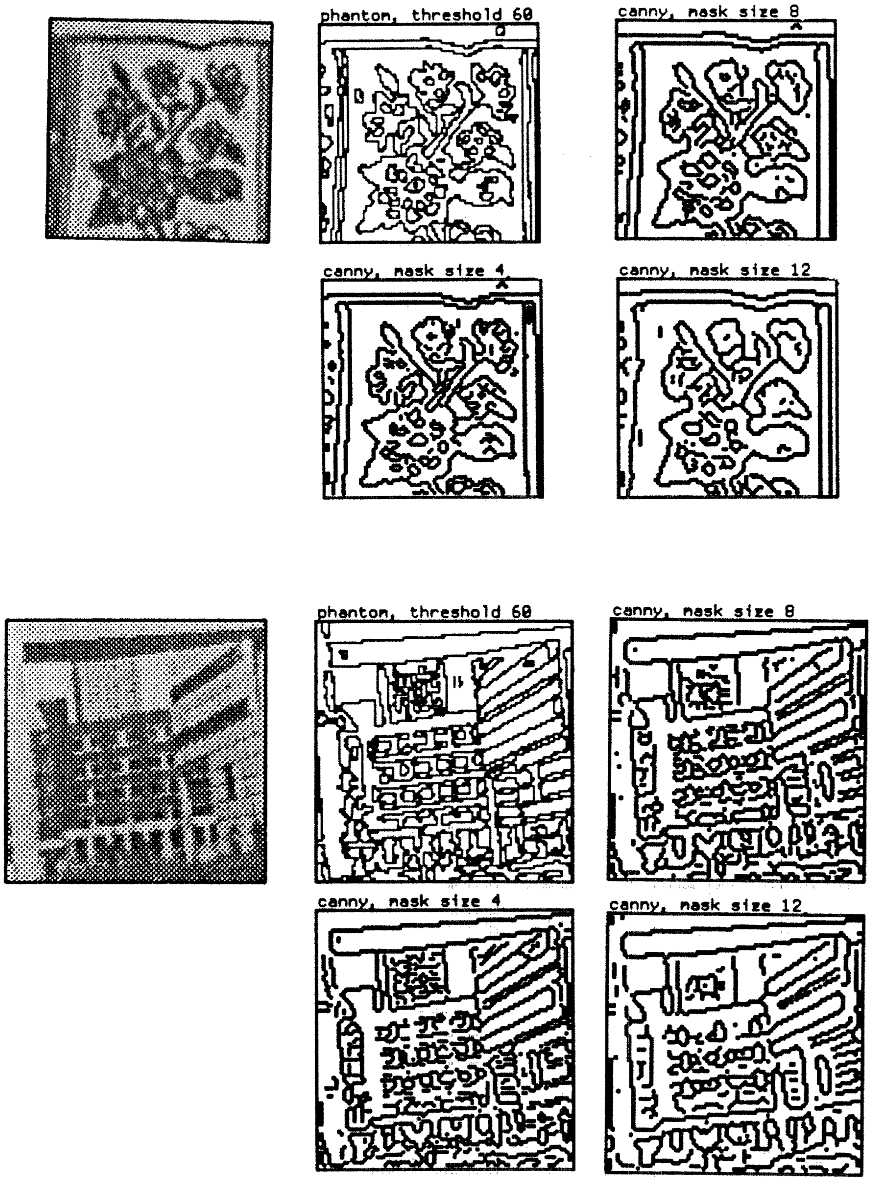


Figure 32. Performance of the two edge finders on fine texture from two real images (84 by 84 cells and 100 by 100 cells). Canny's edge finder is unable to resolve these fine regions correctly.

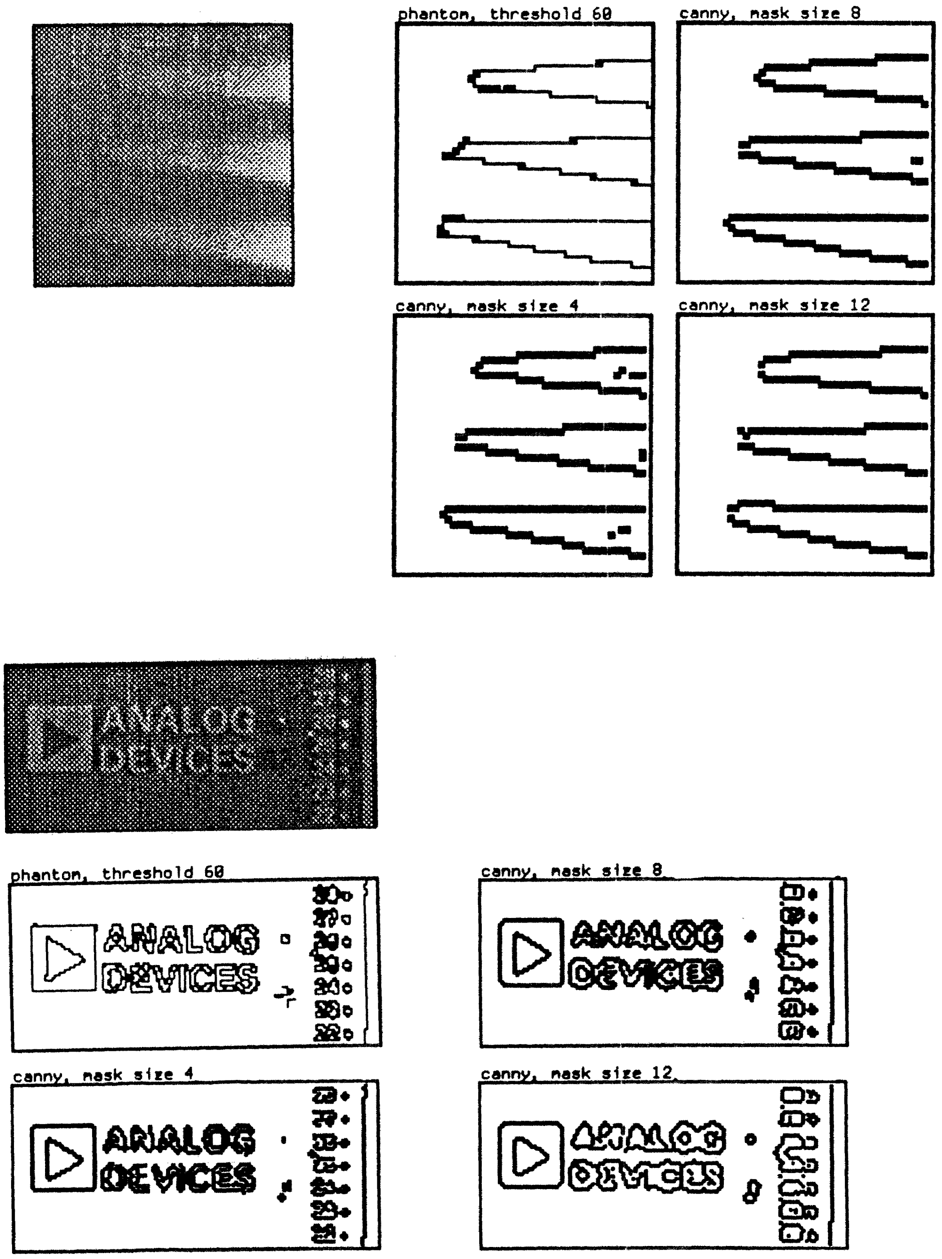


Figure 33. Top: an extract from an image of a fork (50 by 50 cells) showing how Canny's edge finder breaks boundaries at sharp corners. This image has been enlarged by a factor of 2 in each dimension, relative to the other images in this section. Bottom: another image (144 by 64 cells) containing fine texture. The Phantom edge finder resolves the boundaries of the lettering correctly and Canny's edge finder does not.

edge finder responds to differences between scan lines that are adjacent and thus come from different halves of the interlace pattern. The Phantom edge finder avoids this problem by using slightly wide second differences, $[-1,0,2,0,-1]$ rather than $[-1,2,-1]$. This type of problem would not occur in human visual processing.

Canny's edge finder can also produce similar patterns of multiple responses even when interlace effects are not involved. This is illustrated in Figure 34. In an extended region of high first difference, Canny's edge finder marks all local maxima as boundaries. These local maxima can be created by even low-amplitude noise. Canny's non-maximum suppression algorithm depends on having enough smoothing to eliminate these spurious maxima. In fact, frank multiple responses are only common for mask size 4, and rarely occur in natural images analyzed with mask sizes 8 and 12. However, many spurious contours on smoothly shaded objects, such as the eye and cup shown in Figure 35, probably result from this weakness in the algorithm.

All of the above examples illustrate situations in which the Phantom edge finder produces appropriate results and Canny's edge finder does not. The Phantom edge finder does, however, exhibit one failure mode of its own: it produces spurious boundaries on staircase patterns. The mechanism behind this was explained in Chapter 4. Figure 36-37 show examples of these spurious responses in synthetic images with staircase intensity profiles. As these figures show, the spurious responses occur only when the staircase region is relatively small. Furthermore, as discussed in Chapter 4, it may be possible to detect and eliminate these responses when they occur at coarser scales, using information from finer scales. However, I do not know of any robust method for eliminating these phantom boundaries if they occur at the finest scale of analysis. The cup image in

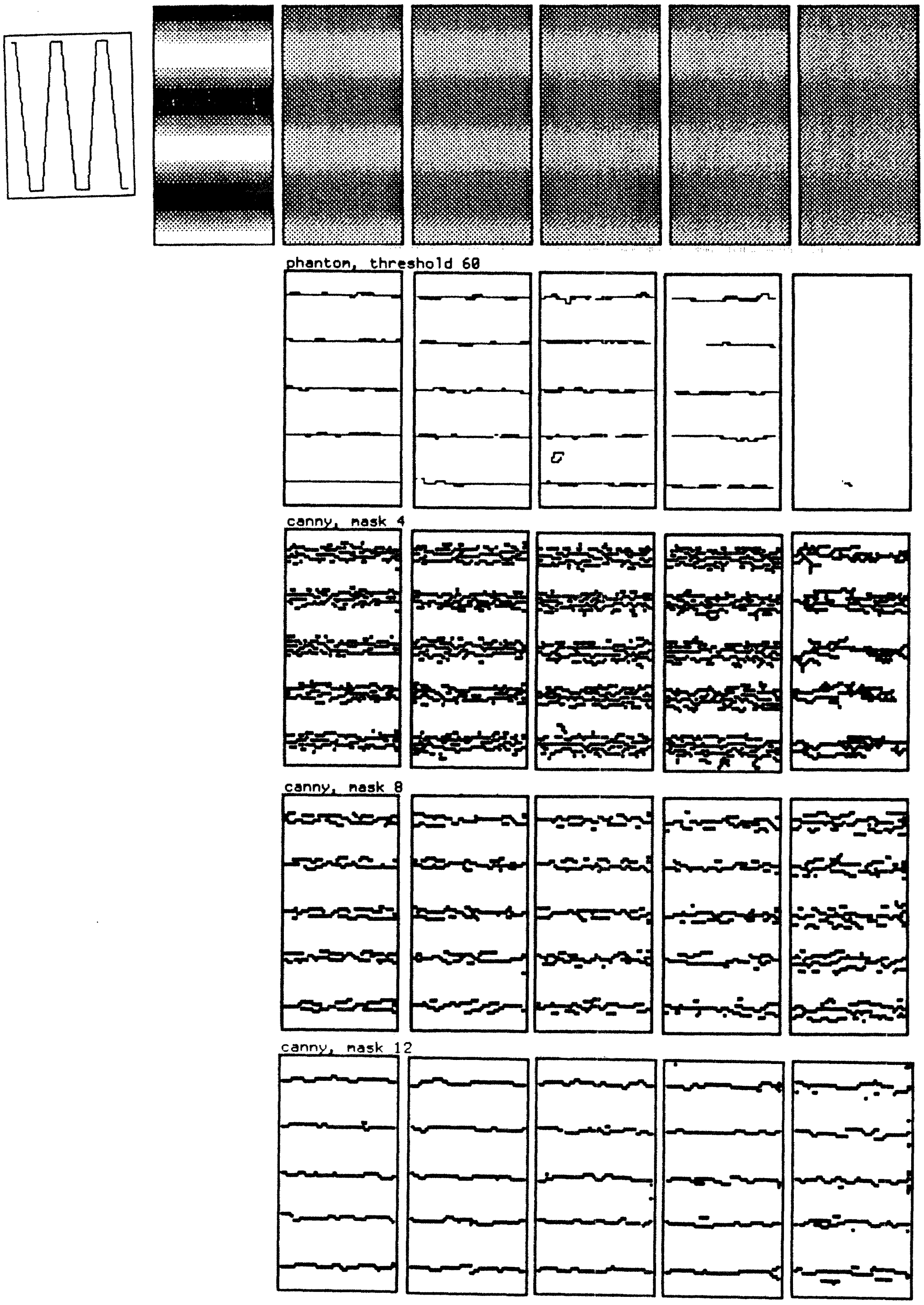


Figure 34. Performance of the two edge finders on a 50 by 100 image of intensity ramps. The graph in the upper lefthand corner shows the intensity profile for a vertical slice through this image. Canny's edge finder generates spurious multiple responses on regions with high, but constant, slopes in intensity.

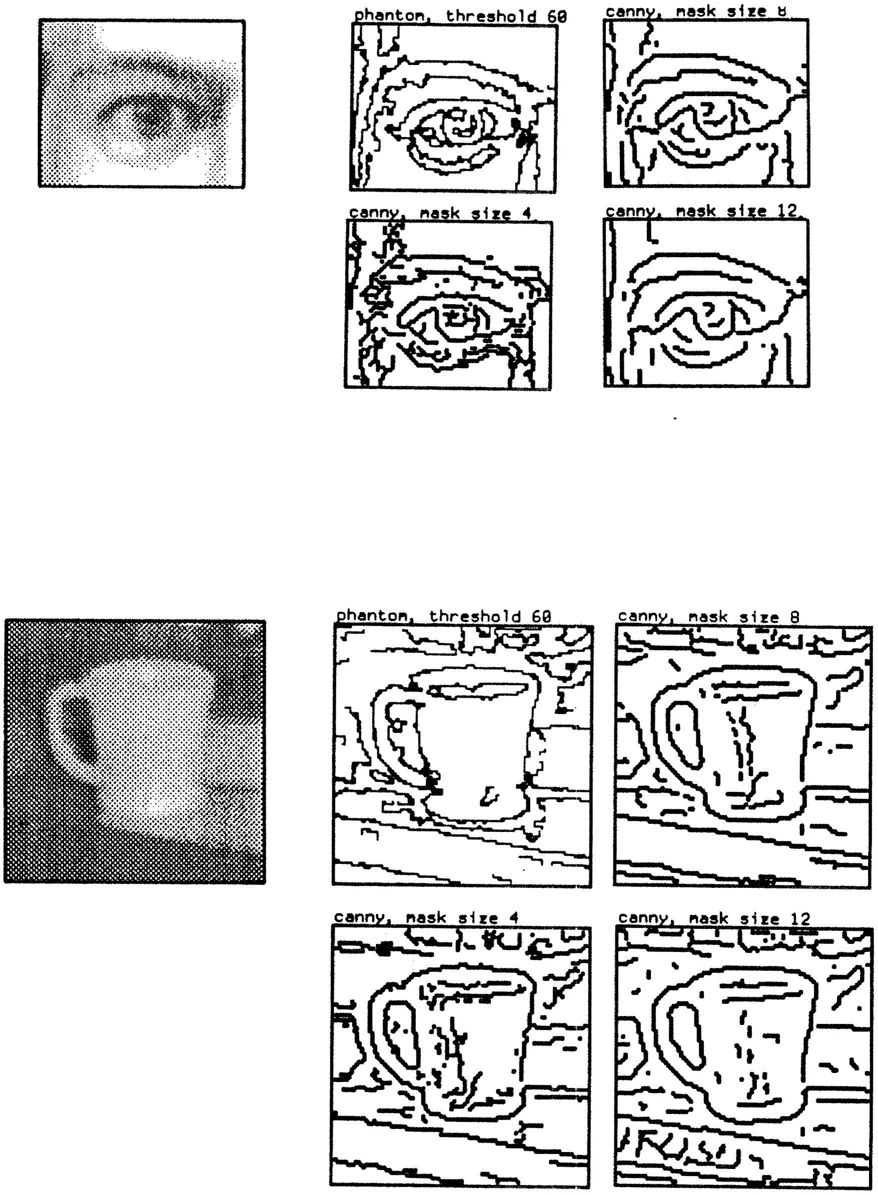


Figure 35. Examples of smooth shading from natural images (80 by 64 cells and 100 by 100 cells). As in the synthetic examples of Figure 34, Canny's edge finder generates spurious boundaries in regions of high intensity slopes.

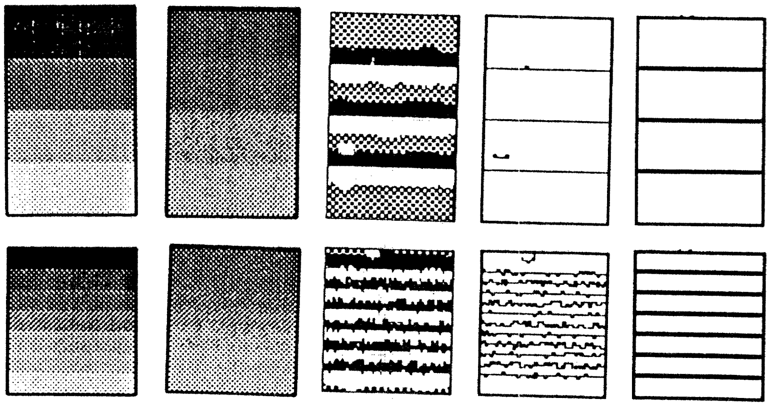


Figure 36. Results of the Phantom edge finder on two images with staircase intensity patterns, with stripes of two different widths (8 and 20 cells). Both edge finders were run on the image with contrast 128 intensity units. Left to right: the image at contrast 256, the image at contrast 128, the Phantom edge finder's response regions, the Phantom edge finder's boundaries, Canny's boundaries (mask size 8). In the image with more closely spaced boundaries, the Phantom edge finder produces spurious boundaries are generated in the middle of each stripe.

Figure 35 shows naturally occuring examples of these configurations.

In this section, we have seen a number of qualitative differences in behavior between the Phantom edge finder and Canny's edge finder. The apparent higher resolution of the Phantom edge finder is due to more accurate responses on high-curvature regions and regions where boundaries intersect. Canny's edge finder also shows problems with multiple responses on ramps, such as those due to smooth shading, and interlaced images (only for mask size 4). Although it performs better in general, the Phantom edge finder consistently marks spurious boundaries in narrow staircase patterns.

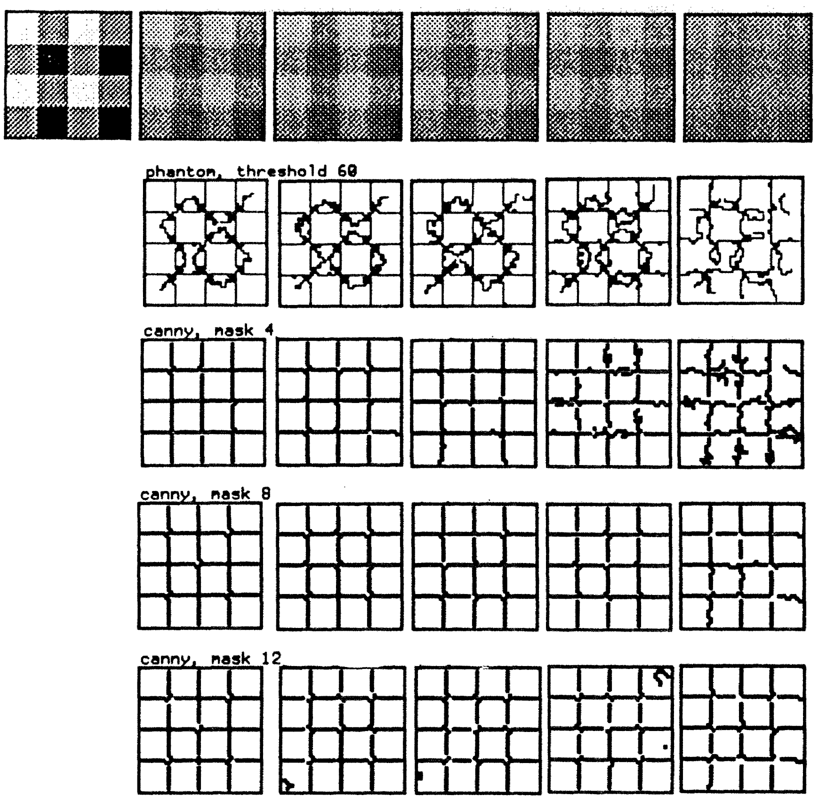


Figure 37. Performance of the Phantom edge finder and Canny's edge finder on a plaid pattern (black, white, and grey squares). As in the staircase patterns in Figure 36, the Phantom edge finder generates spurious boundaries. Canny's edge finder does not generate any spurious boundaries, but it breaks boundaries near the intersections.

7. Conclusions

The experiments presented in this chapter illustrate a number of differences in behavior between the Phantom edge finder and Canny's edge finder. These include both differences in stability and differences in qualitative behavior. The stability tests also illustrate objective methods by which noise thresholds can be chosen for the two programs. Although still incomplete, these results are more extensive than those presented in previous edge finder evaluations.

The first global conclusion that can be drawn from these experiments is that mask size 4 is a poor choice for Canny's edge finder. It performs poorly in the stability tests, produces frequent multiple responses due to smooth shading, and responds to differences between camera interlace frames. These problems with mask size 4 should come as no surprise: informal observations have led previous researchers to use mask size 8 or 12 for this edge finder. Nevertheless, we now have empirical confirmation of these problems. When confronted with differences in resolution between Canny's edge finder and other algorithms, it is tempting to suggest using a smaller mask size. These results make it clear that mask sizes below 8 are too sensitive to noise to be viable alternatives.

Secondly, we can conclude that the Phantom edge finder performs, on the whole, better than Canny's edge finder. It detects boundaries at higher resolution than Canny's edge finder. Although it may lose blurred edges as a consequence, these edges can always be recovered from coarser scales of the edge finder. When camera noise of moderate amplitude is present, Phantom's output is more stable than Canny's, for any of the three mask sizes tested. When higher-amplitude noise is present, the two perform similarly. The single most effective technique for analyzing the high noise image was a combination of Phantom's algorithm with Gaussian smoothing. This suggests that Gaussian smoothing, advocated by many previous researchers, is not a bad idea, but simply over-used.

Finally, the stability evaluations presented in Sections 3-5 provide an objective method of setting edge finder noise thresholds. Since the plots of stability vs. number of edges are roughly L-shaped, good choices for noise thresholds must lie in the bend of the curve. For the Phantom edge finder, the bend is relatively sharp and so the range of choices is quite small. For Canny's edge finder, the bends are more gradual. The requirements of particular applications may effect the exact setting of the threshold. However, these evaluations provide a good basis for making the decision.

These evaluations are related in three ways to the topological ideas presented in this thesis. First, they show that the noise suppression algorithm based on star-convex neighborhoods reliably suppresses the effects of camera noise, performing better than the Gaussian smoothing used in Canny's edge finder. Secondly, the new edge finder takes account of the differences between the behavior of derivatives in infinite resolution spaces and differences in digitized spaces. This helps the edge finder perform more reliably on corners and region intersections. Finally, the topological matcher made it possible to conduct quantitative evaluations of edge finder performance on images of real scenes, which has not been possible before.

Chapter 10: Stereo testing

1. Introduction

In this chapter, I will present some tests of the new stereo matching algorithm, on both synthetic and natural stereo pairs. These examples illustrate that the new algorithm can tolerate its large search area without becoming confused. They show that it can successfully match images with substantial vertical displacements and rotation, recovering plausible horizontal disparities. They also show that the new algorithm can reconstruct sharp depth discontinuities without blurring depth values across the discontinuity.

Section 1 will discuss general procedures used in all of the stereo tests and the method of displaying stereo results. Section 2 discusses the synthetic stereo examples and Section 3 discusses the natural stereograms. Section 4 presents a brief example showing how the same matching technique might be applied in analyzing motion sequences.

2. Procedures

The new stereo algorithm runs quite slowly, due to a combination of the slowness of star-convex sum and the wide search neighborhoods considered. Each of the examples presented here took from a couple days to a week to run, depending on the size of the image and the range of disparities present in it. Thus, careful choice of examples was essential in order to achieve as much coverage as possible. The examples were deliberately chosen to be difficult to match.

The images used are described in detail in Sections 3 and 4. Most of the images have some vertical disparity (up to 16 cells). Horizontal disparities occur with both signs and with magnitudes up to 35 cells. Several of the natural images are rotated. Although the angular rotations are small (up to 5 degrees), they create significant additional vertical displacements near the edges of the images. As described in Chapter 6, the algorithm initially assumes that the images are correctly aligned and adjusts the alignment as it computes the stereo correspondence. Except for the motion example (discussed in Section 4), the search parameters and adjustment algorithm are as described in Chapter 6. As in most other examples in this thesis, the edge finder was run with noise threshold 60.

Most of the images presented could not be run to the finest scale possible, due to time constraints. Thus, as indicated in the individual image descriptions, the results presented are from sub-sampled versions of the images. As noted in Chapter 4, the effects of camera noise are less for sub-sampled images. However, since the edge finder's noise suppression algorithm is quite robust, I do not think that this significantly affects the stereo results.

Figure 1 shows the output for a synthetic stereogram, discussed more fully in Section 3. Anticipating that many readers will see only xeroxed versions of these results, I have designed the display format to make as much use of binary images as possible. All of the stereo pairs are presented in the same format and I will explain some details of it here. In the interests of space, I have only presented the stereo results from the point of view of one of the two images. Except for two images, for which both sets of results are shown, results for the other half of the computations were similar. Results are presented for the finest scale to which each computation was run.

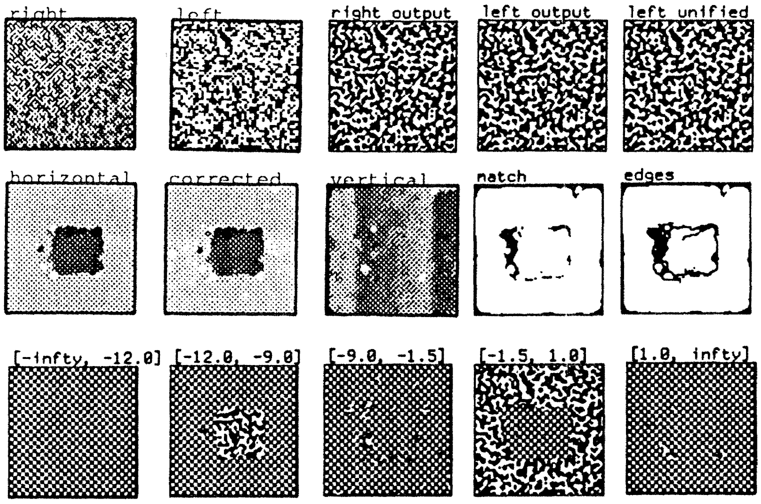


Figure 1. Matching results for a stereogram with a change in contrast. The top row shows the stereogram, edge finder results, and a cartoon derived from the edge finder results for the lefthand image. The middle row show computed disparities, match maps, and edges. The bottom row show which regions of the lefthand image matched at various disparity ranges. (Non-matching parts of the image are shown in a checkerboard pattern.) The computed adjustment to alignment was (-1.7, -2.0) cells of translation and a rotation of 0.9 degrees.

The stereo pairs are presented for crossed-eye viewing. In the top row of Figure 1, I have shown both the original images and the cartoon version of the edge finder output for the finest scale at which stereo matching was done. The final item in this row is a binary cartoon created by filling gaps in the finest scale cartoon (left image) with values from coarser scales. This process was described in Chapter 4, Section 8. For this image, it has little effect, because the finest scale responses are rarely zero. The images presented in Figure 11 show the difference more clearly.

The second and third rows show the matching results. Three disparity fields are shown: horizontal, corrected horizontal, and vertical. Remember that image rotation can be estimated using the vertical disparities. The corrected horizontal disparities are the horizontal disparities corrected for the effects of this estimated rotation. The displays are linear, but the intensity range reflects only the main[1] disparity responses. Outliers beyond this range are normalized to the edges of the range. These outliers represent only small numbers of points, as you can see from the range displays discussed below. All of these results are from the point of view of the lefthand image.

The match map shows which cells (in the lefthand image) were assigned stereo disparities. Cells marked in black did not receive any acceptable stereo match. The edge map shows edges computed from the corrected horizontal disparities using the Phantom edge finder (noise threshold 240). Boundary cells and cells to the dark sides of boundaries are shown in black, as are non-matching cells from the match map. These edge finder results are experimental and are primarily presented to give the reader another source of information about the computed disparities. More experimentation would be needed to design a robust adaptation of the edge finder to this domain.

The third row of the output display shows which parts of the left image had disparities within specified ranges. The disparity ranges are specified in cells (measured in the finest scale version of the image that was matched) and are given above each image. Regions of the image within the range are shown in cartoon form, using the filled-in binary cartoon (last item from the top row). Areas outside the range are shown in a checkerboard pattern. The disparity ranges were adjusted by hand, to produce as informative a display as possible. Finally, the computed adjustments to the image alignment are given. The translation is reported as a vector, horizontal component first.

[1] In terms of numerical distribution, judged subjectively.

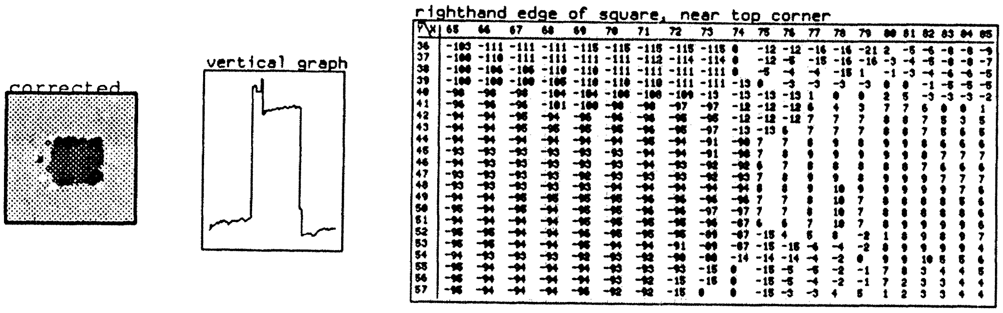
3. Synthetic images

The stereo algorithm was run on five random-dot stereograms, illustrating a variety of conditions. The images include a stereogram with large difference in contrast, one with a large vertical offset, an example of Panum's limiting case, an example with steep gradients in disparity, and a sparse stereogram. All stereograms were created as binary images with intensity values 0 and 255. They were then blurred with an approximation to a Gaussian of $\sigma = 1$ cell and Gaussian noise of $\sigma = 3$ intensity units was added. This procedure simulates the effects of camera noise and is the same as that used in Chapter 9.

The stereogram presented in Figure 1 is a 50% random dot stereogram showing a raised square. The image is 100 cells square and the dots are 2 cells on a side. The square has a horizontal disparity of 10 cells and the whole image (square and background) has a vertical disparity of 2 cells. The contrast of the righthand image was reduced to half that of the lefthand image.

As you might expect, the stereo matcher performs well on this image. There are some mis-matches, particularly near the occluded region and a strip along the top of the square has disparities consistently slightly high (by perhaps 2 cells). Figure 2 shows some intensity values near the edge of the square and a vertical slice though the disparity map. As you can see, the stereo algorithm has correctly reconstructed a sharp change in disparity between the square and the background.

Disparity values for this image, and the other synthetic images, fluctuate 1-1.5 cells from the correct disparity. This probably reflects errors in the boundary motion calculation, rather than incorrect matches. Although the algorithm computes the correct adjustment to the vertical alignment of the images, it incorrectly reconstructs a slight (0.9 degree) rotation. This error may be due to



righthand edge of square, near top corner

Y\X	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85
36	-103	-111	-111	-111	-115	-115	-115	-115	-115	0	-12	-12	-16	-16	-21	2	-5	-6	-8	-8	-9
37	-100	-110	-111	-111	-111	-111	-112	-114	-114	0	-12	-5	-15	-16	-16	-3	-4	-5	-8	-8	-7
38	-100	-106	-106	-110	-110	-111	-111	-111	-111	0	-5	-4	-4	-15	1	-1	-3	-4	-6	-6	-5
39	-100	-100	-100	-105	-110	-110	-110	-111	-111	-13	0	-3	-3	-3	-3	0	0	-1	-5	-5	-5
40	-98	-98	-98	-104	-104	-106	-106	-109	-13	-13	-13	-13	1	0	0	2	5	-3	-3	-3	-2
41	-96	-96	-96	-101	-100	-98	-98	-97	-97	-12	-12	-12	6	4	3	7	7	6	0	0	1
42	-94	-94	-95	-94	-96	-96	-96	-95	-96	-12	-12	-12	7	7	7	8	8	7	5	3	5
43	-94	-94	-95	-95	-95	-96	-96	-95	-97	-13	-13	6	7	7	7	8	8	7	5	6	5
44	-94	-94	-94	-94	-94	-94	-95	-94	-91	-90	7	7	8	9	8	9	9	8	6	6	6
45	-93	-93	-93	-93	-93	-93	-94	-94	-91	-90	7	8	9	9	9	9	9	8	7	7	7
46	-94	-93	-93	-93	-93	-93	-94	-93	-92	-92	6	7	8	8	8	8	8	7	6	6	6
47	-93	-93	-93	-93	-92	-93	-93	-93	-92	-93	7	8	9	9	8	9	9	9	7	7	7
48	-93	-93	-93	-93	-93	-94	-94	-94	-94	-94	8	8	9	10	9	9	9	9	9	7	8
49	-94	-94	-95	-95	-95	-95	-96	-96	-96	-96	7	7	8	10	7	8	8	8	8	5	6
50	-95	-94	-95	-94	-95	-95	-96	-96	-97	-97	7	7	8	10	7	8	8	8	8	8	6
51	-94	-94	-94	-94	-95	-95	-95	-95	-96	-87	6	6	7	10	7	8	9	9	9	9	6
52	-96	-95	-94	-94	-96	-95	-95	-95	-89	-87	-15	4	5	8	-2	1	8	9	8	9	7
53	-96	-96	-94	-94	-96	-94	-94	-91	-89	-87	-15	-15	-5	-4	-2	8	9	9	9	9	4
54	-94	-93	-93	-93	-92	-93	-92	-90	-88	-14	-14	-14	-4	-2	0	9	9	10	5	5	6
55	-95	-94	-94	-94	-94	-93	-93	-93	-15	0	-15	-5	-5	-2	-1	7	8	3	4	4	5
56	-96	-94	-94	-94	-94	-93	-92	-15	-15	0	-15	-5	-4	-2	-1	7	2	3	3	4	4
57	-96	-94	-94	-94	-96	-92	-92	-15	0	0	-15	-3	-3	4	5	1	2	3	3	4	4

Figure 2. Disparity values for part of the image in Figure 1 and a vertical slice through the middle of the same image. The reconstructed disparity values change abruptly at the boundary between the square and the background, in agreement with the subjective judgements of human observers.

the errors in disparity values and/or to the rather primitive estimation technique. Despite such small errors, the reconstructed rotations for all of these images were close enough to enable successful match.

The second synthetic stereogram is shown in Figures 3-4. This stereogram was created by averaging two 50% random-dot stereograms, one with 2 by 2 dots and one with 16 by 16 dots. The stereogram is 200 cells square and depicts a raised square with a disparity of 4 cells. The whole images (both square and background) have been shifted 16 pixels vertically, relative to one another. This stereogram may take some effort to fuse. Since the top and bottom of the image are rivalrous, fusion is easier to obtain if the center of the image is fixated. Again, the stereo matcher matches this image with few errors.

The third and fourth synthetic stereograms are shown in Figure 5. They are both 200 cells square, but were only run to the second finest scale (100 cells square). The top stereogram shows a set of ramps with a peak-to-peak spacing of 100 cells and peak disparities of ±10 cells (i.e. ±5 cells at this scale). It is a

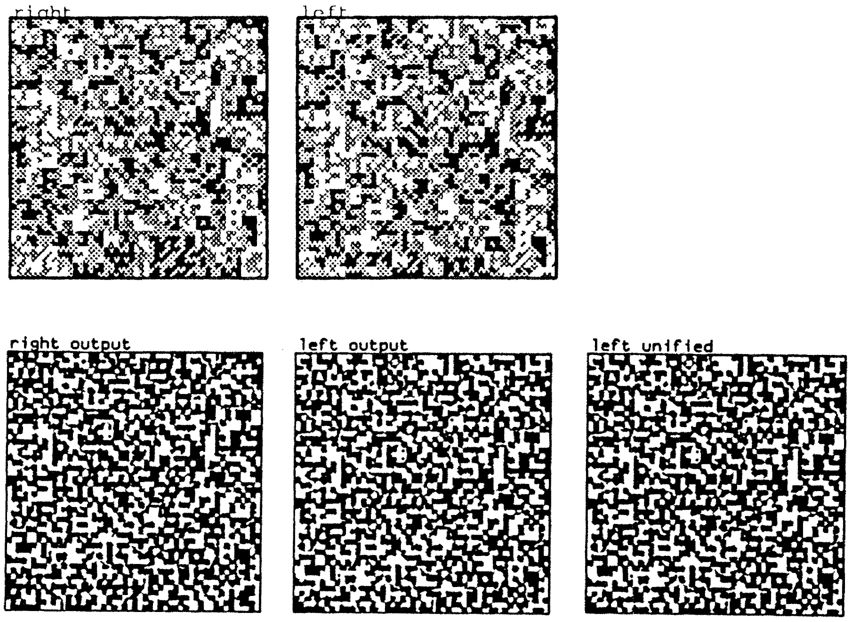


Figure 3. A stereogram with a large vertical displacement.

10% random-dot stereogram with dots 2 cells on a side. The bottom stereogram shows a raised square, but using only sparse (5%) dots. Each dot is 4 cells on a side. The square has a horizontal disparity of 20 cells (i.e. 10 at this scale) and a vertical disparity, relative to the background, of 4 cells (i.e. 2 cells at this scale).

Both of the stereograms in Figure 5 were slightly more difficult to fuse and caused more errors. The tops and bottoms of the ramps were often smoothed or unmatched. There is a large patch of incorrect disparity in the sparse image. As you can see, this area has few boundaries. Thus, the disparities in this area are largely inherited from coarser scales, at which the problem arose. The shape of the square in the sparse stereogram is only poorly recovered, but this also seems to be true for human perception.

The final stereogram shows an example of Panum's limiting case. The right-hand image is a normal 5% random-dot stereogram, 100 cells square, with dots

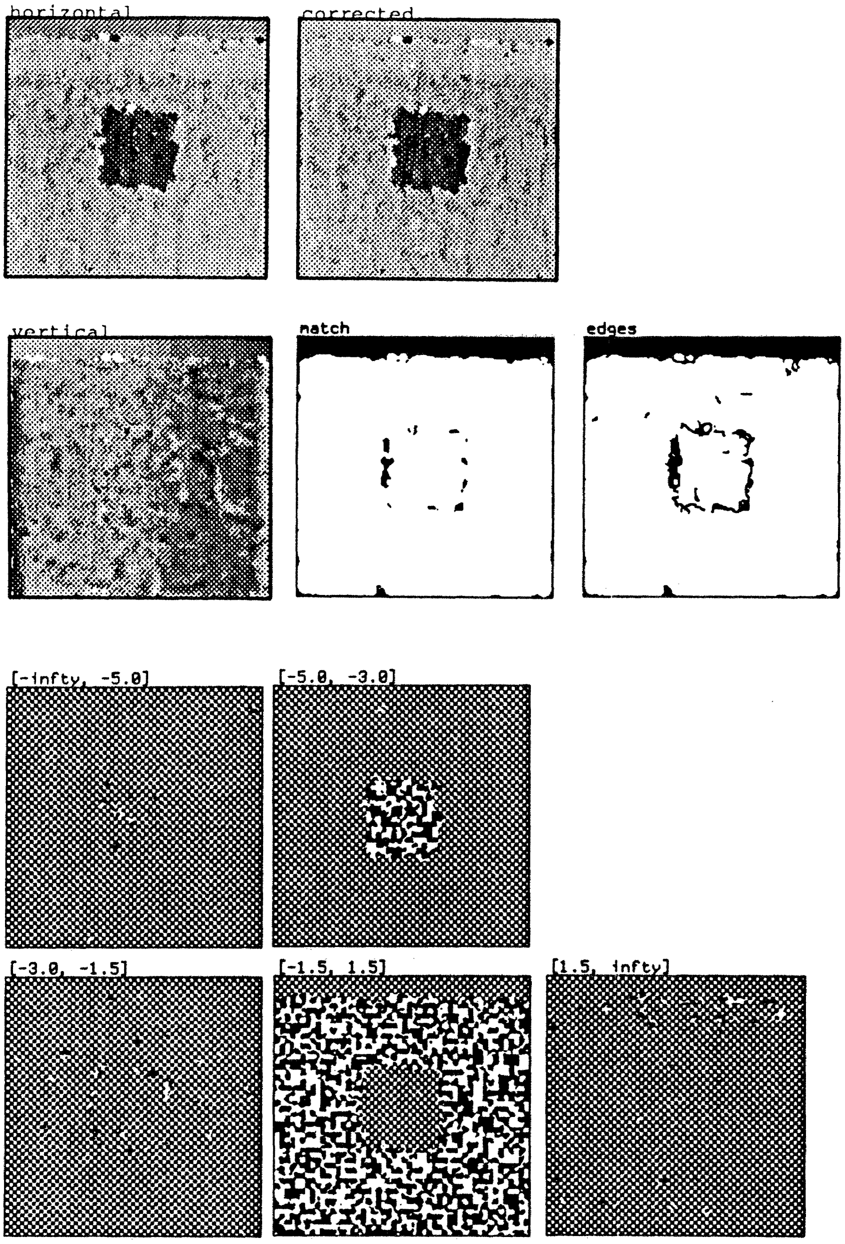


Figure 4. Matching results for the stereogram in Figure 3. Computed adjustment to alignment was (-0.4, -16.0) cells of translation and a rotation of -0.2 degrees.

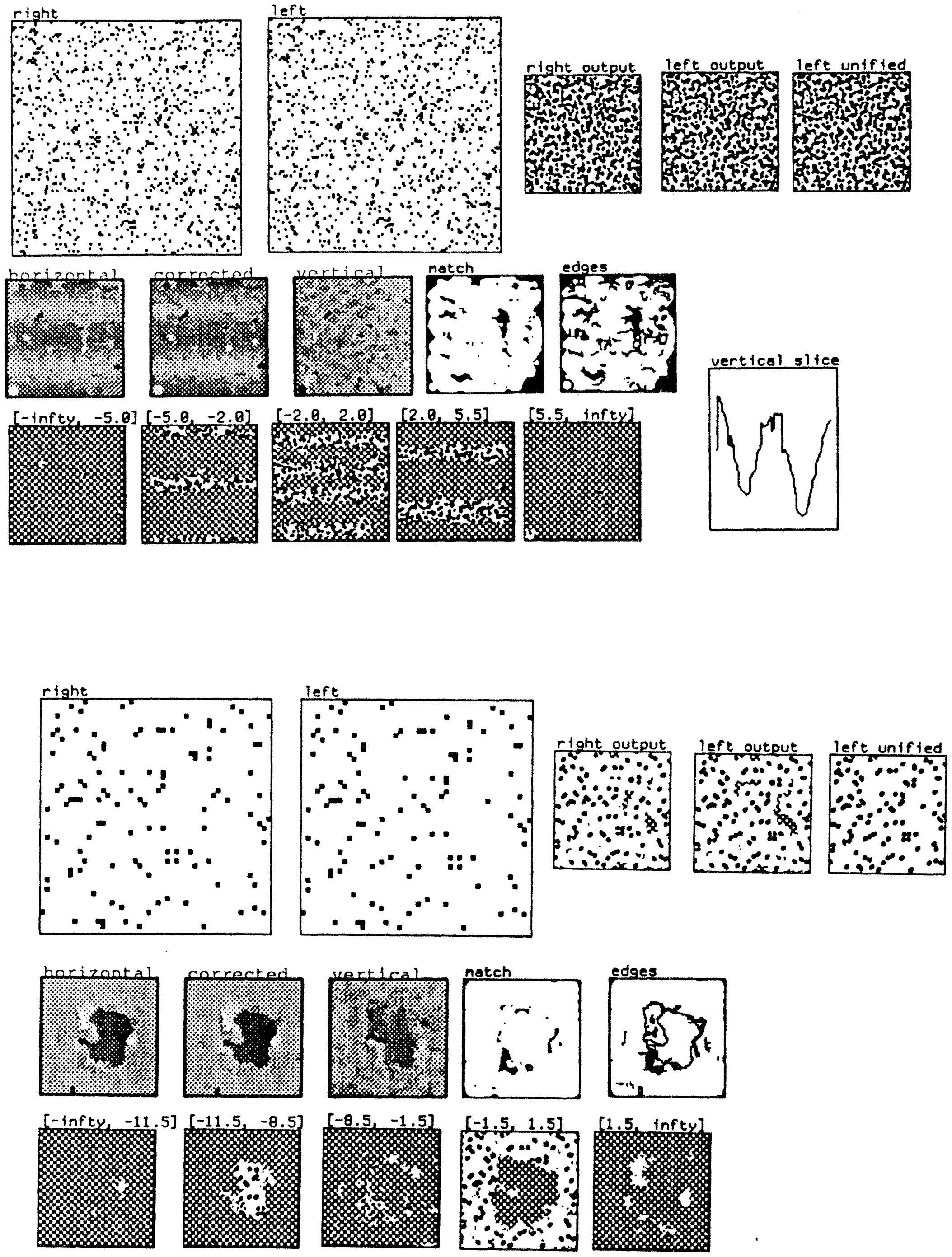


Figure 5. Matching results for two synthetic stereograms. Top: A stereogram of representing ramp-like surfaces. The computed translation was (0.7, -0.4) cells and the computed rotation was 0.1 degrees. In addition to the usual matching results, a vertical slice through the reconstructed disparities is shown. Bottom: A sparse stereogram. The computed translation was (-1.4, -0.8) and the computed rotation was 0.5 degrees.

2 cells on a side. The lefthand image was created by displacing the righthand image horizontally by 4 cells and super-imposing it on itself. Humans interpret this stereogram as representing parts of two parallel flat surfaces. The stereo algorithm recovers many of the correspondences for this interpretation, although it does generate a number of intermediate values between the two surfaces.

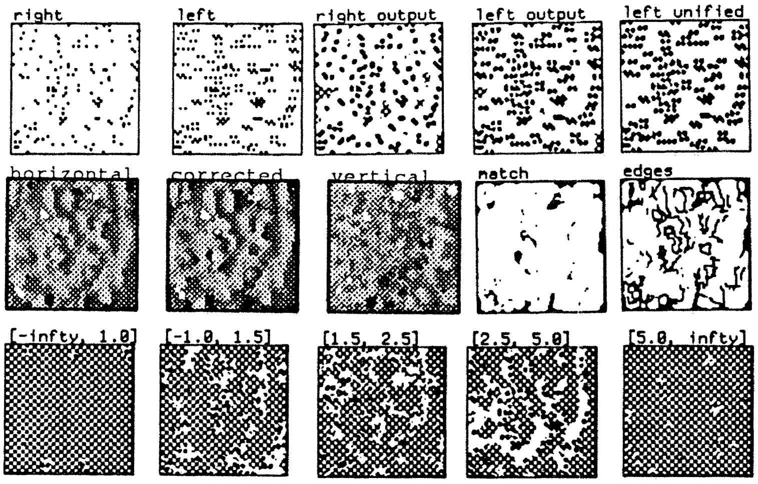


Figure 6. Matching results for a stereogram of Panum's limiting case. This stereogram depicts two parallel planes. Each dot in the righthand image must be matched to two dots in the lefthand image. The computed translation was (-2.4, 0.3) cells and the computed rotation was 0.3 degrees.

For stereograms such as this, where a patch of one image might match two distinct patches of the other image, results from the two halves of the stereo computation are not guaranteed to be the same. Figure 8 shows the results from the perspective of the righthand image. They are more similar to the lefthand results than one might expect. The algorithm has managed to recover many of the same depth differences, despite having to split dots into two halves in creating the depth boundaries.

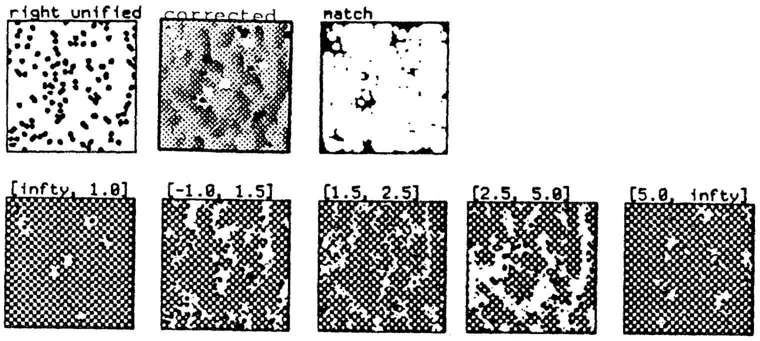


Figure 7. Results for the stereogram in Figure 6, seen from the perspective of the righthand image.

4. Natural images

The stereo algorithm was also run on five natural stereograms. Because of time limitations, it was not possible to run these examples to the finest scale. In all cases, the images were sub-sampled to $\frac{1}{4}$ the area of the original image. In one case (noted below) this was repeated twice. Note that calibration data was available for some of these images and other researchers have used versions of these images which have been adjusted to remove alignment problems. In such cases, I have used the original camera images, without normalization.

The first image in the set, an image of buildings from the air, is shown in Figure 8. It has very little vertical displacement or rotation. Although the new algorithm makes some scattered errors, it successfully detects the height difference between the buildings and the background. In comparing these results to those of previous algorithms, remember that the new algorithm must consider the whole (large) vertical and horizontal search area even though this image happens to have only small displacements.

The next two examples, shown in Figure 9, show two vision researchers in

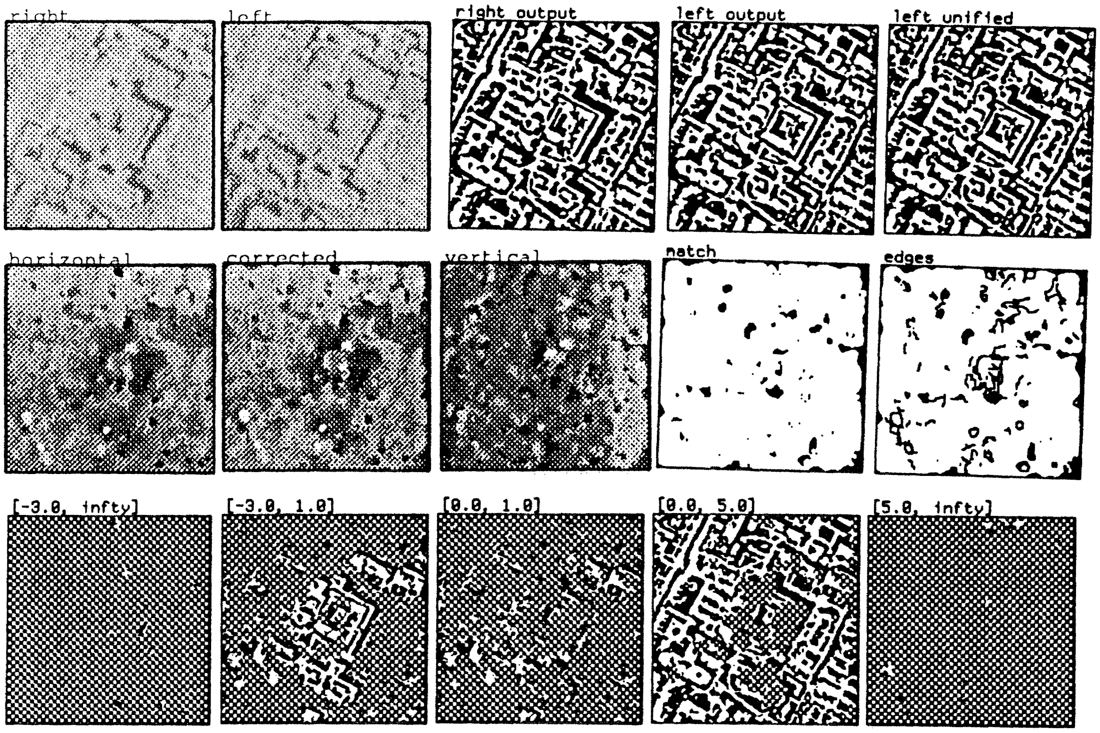


Figure 8. Matching results for an image of buildings from the air. The computed translation was (-1.3, 0.6) cells and the computed rotation was 0.3 degrees.

front of a textured background. In the original images, the two people were roughly the same size. In the top pair, a portion of the images was extracted for computation[2] and sub-sampled once. In the second pair, the images were sub-sampled twice. These stereo pairs were taken from a camera system that was not precisely calibrated and there is a noticable rotation between the two images in each pair. In both cases, the researcher has been successfully separated from the background.

The fourth stereo pair (Figures 10-11) shows a teddy bear, a newspaper, and a metal part on a wooden table. The change in depth between the objects

[2] In this example, and in the other examples in this section, sections with the same coordinates were extracted from both images. Thus, any overall translation between the two images was preserved in the cropped versions.

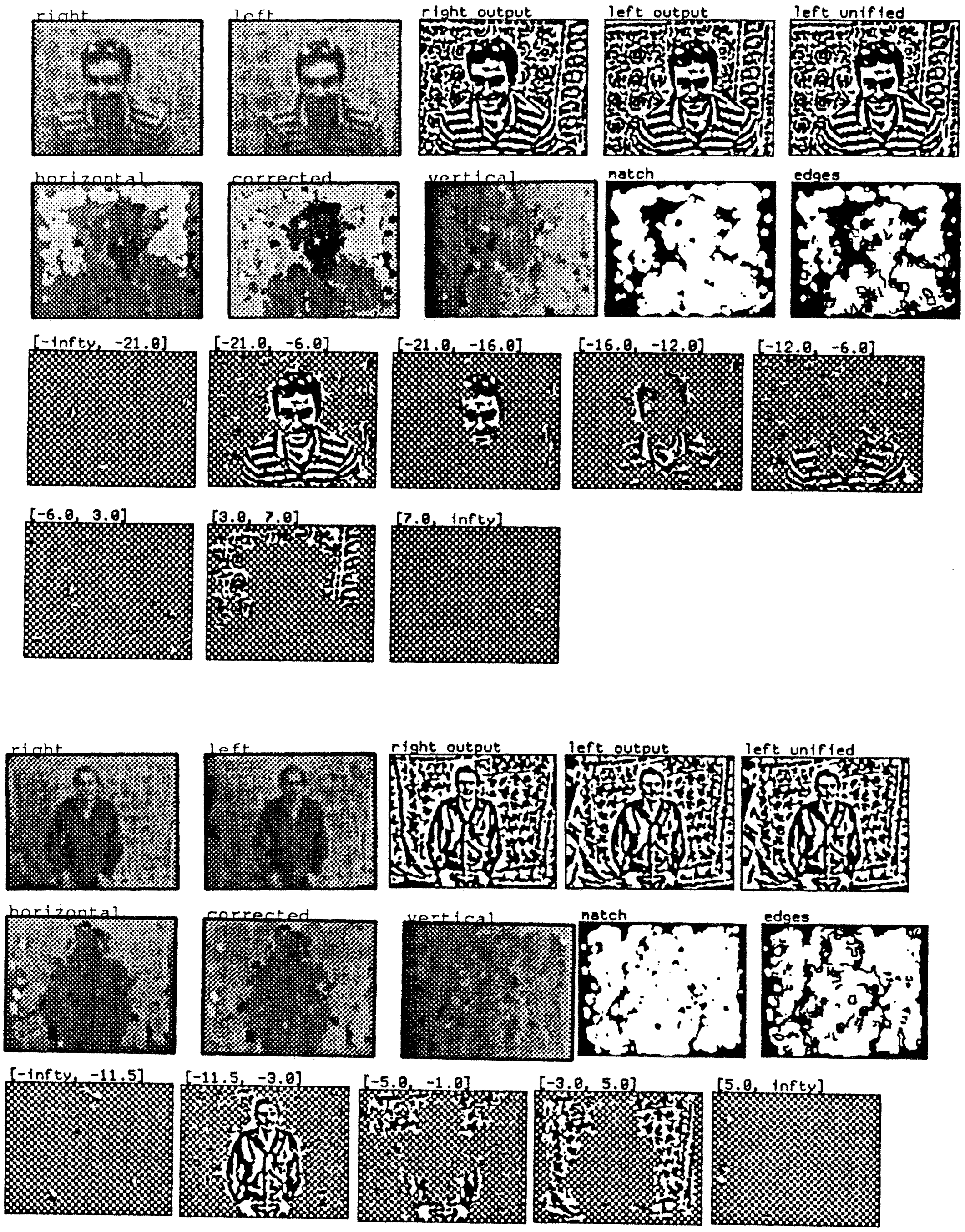


Figure 9. Matching results for two stereograms of people in front of a textured background. For the top pair, the computed translation was (-6.2, -0.2) cells and the computed rotation was 3.8 degrees. For the bottom pair, the computed translation was (2.9, -0.1) cells and the computed rotation was -5.0 degrees.

and the background is less than it was for the researcher images, because the objects are lying directly on the table. Thus, although the algorithm separates the newspaper neatly from the background, there is a region at the edges of the bear where intermediate depth values are produced. It is not clear whether these values are correct or reflect blurring. In this image, the disparity differences between the objects and the background are small relative to the amount of rotation. Thus, the objects are much easier to identify in the corrected disparity map than in the uncorrected map.

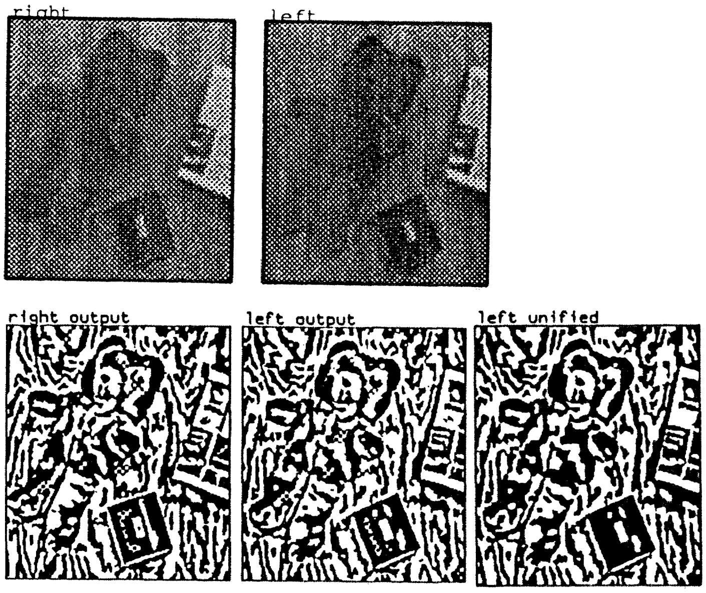


Figure 10. A stereogram of a teddy bear, a newspaper, and a metal part on a wooden table.

The final natural stereogram, in Figure 12 shows a view down a laboratory corridor. This image, although sub-sampled, still has a significant vertical displacement, large horizontal disparities, and substantial occlusion. However, the matcher still succeeds in fusing much of the image. The major errors involve the strip along the lefthand side of the left image. This region is off the edge of the

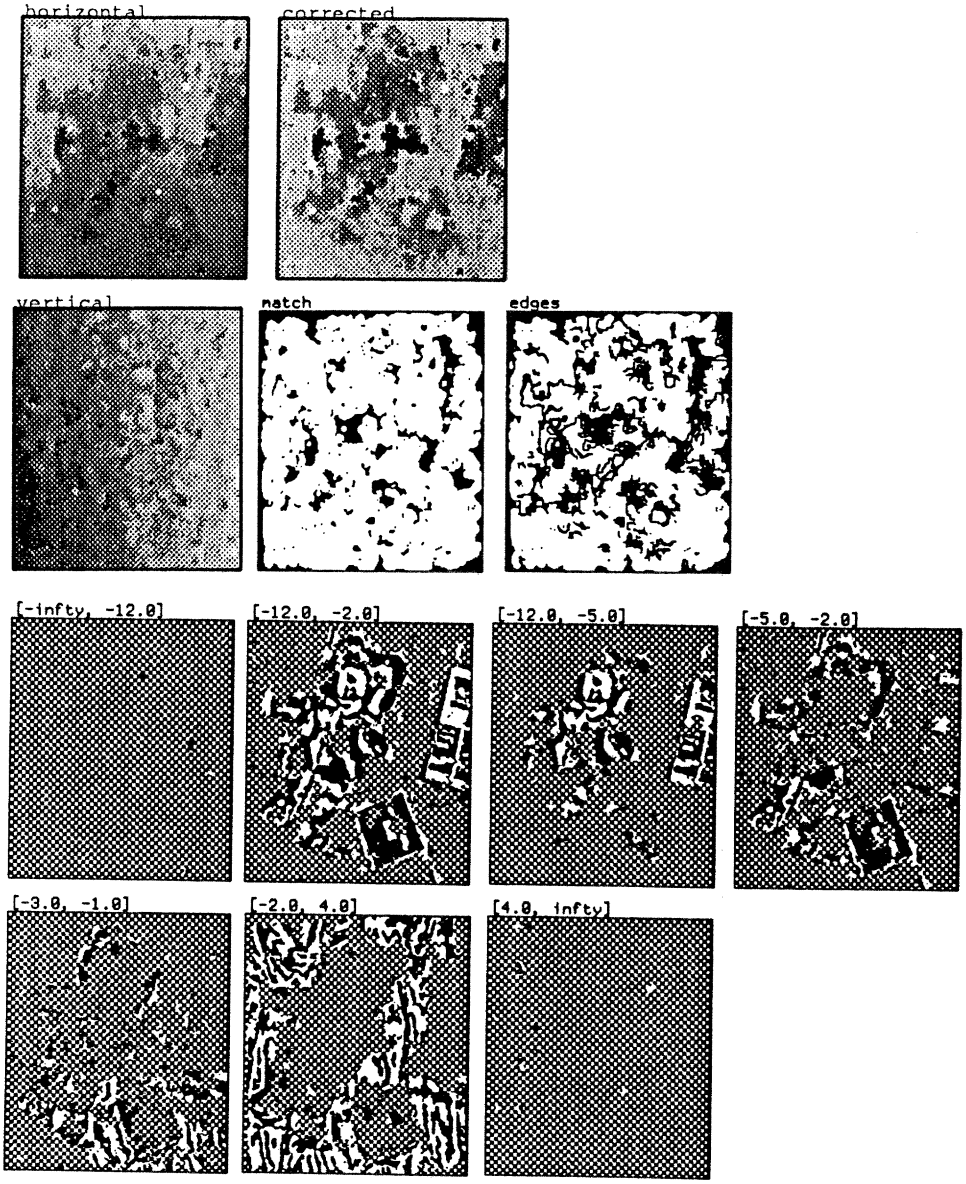


Figure 11. Matching results for the stereogram in Figure 10. The computed translation was (-2.3, -2.1) cells and the computed rotation was 4.1 degrees.

right image, but the matcher mistakenly assigns disparity values to it. In fact, this region is matched to the *same* part of the image as the strip immediately to its right, which it closely resembles.

Clearly, this duplicate match cannot be correct. This would be a good situation in which to apply ordering constraints, as discussed in Chapter 6. Such constraints, however, can only determine that one of the two matches is incorrect. Some additional information is needed to distinguish between them. For this image, it is unclear how to decide which match is correct, since the two strips of image are so similar. Figure 13 shows the results of the computation from the perspective of the right image. Although this computation has assigned much of the strip to the correct disparity, some points have been given the other candidate disparity. Thus, the match evaluations have not been able to robustly assign a higher evaluation to the correct match.

5. A motion example

Because motion analysis and stereo matching are such similar tasks, I ran a modified version of the stereo algorithm on two successive frames of a motion sequence, shown in Figure 14. This sequence shows a hand holding a cup and rotating it. The hand holding the cup is moving non-rigidly, with the arm still and the fingers following the motion of the cup. The frames shown were sub-sampled to $\frac{1}{4}$ the area of the original image.[3] Although this was not a fast motion and the image has been sub-sampled, displacements are still as large as 7 cells for some parts of the cup, i.e. larger than the width of many texture regions in this image.

[3] The original image was smoothed slightly, since it was taken under high noise conditions. However, this probably matters little to the results presented, because of the sub-sampling.

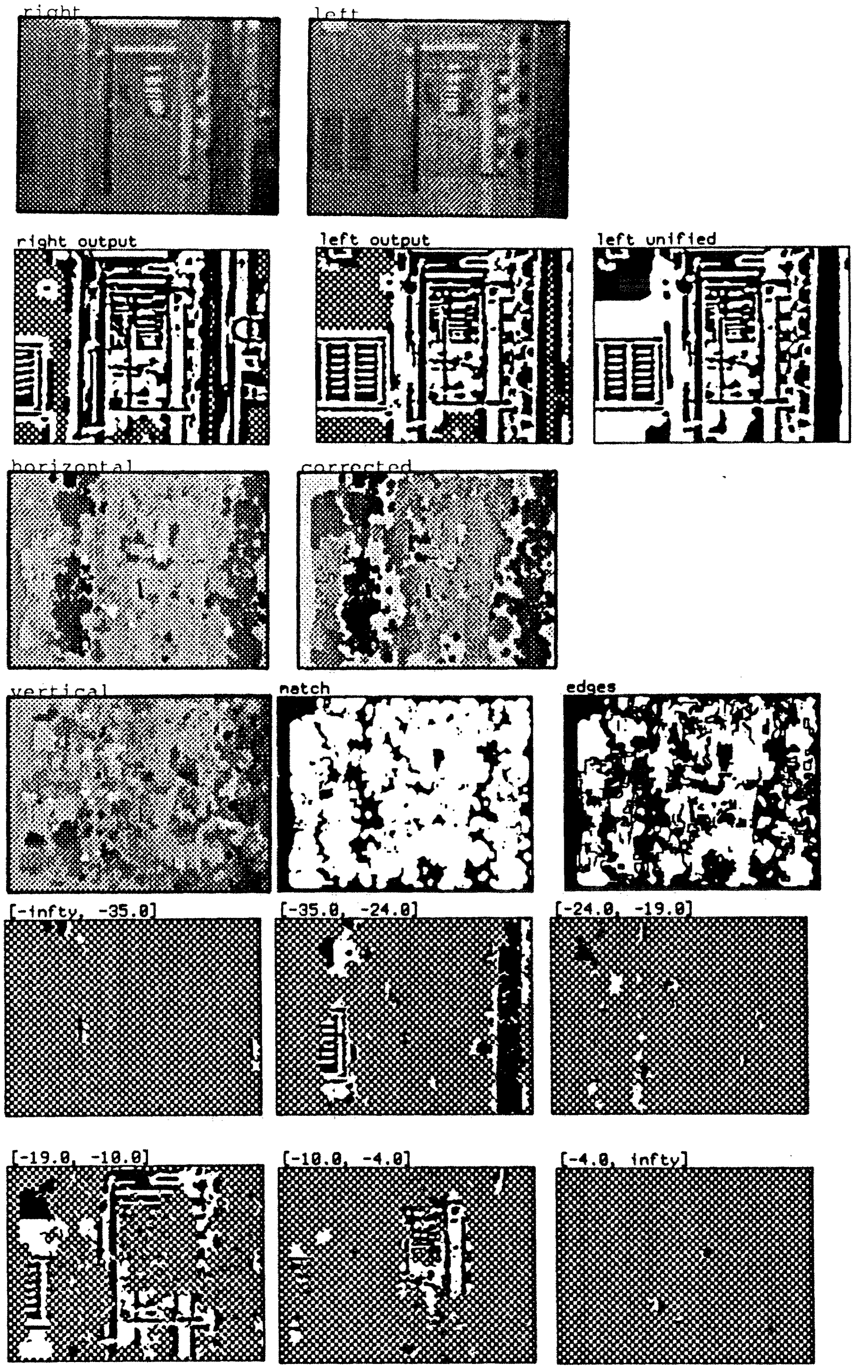


Figure 12. Matching results for a view down a corridor. The computed translation was (-17.2, 3.2) cells and the computed rotation was 0.2 degrees.

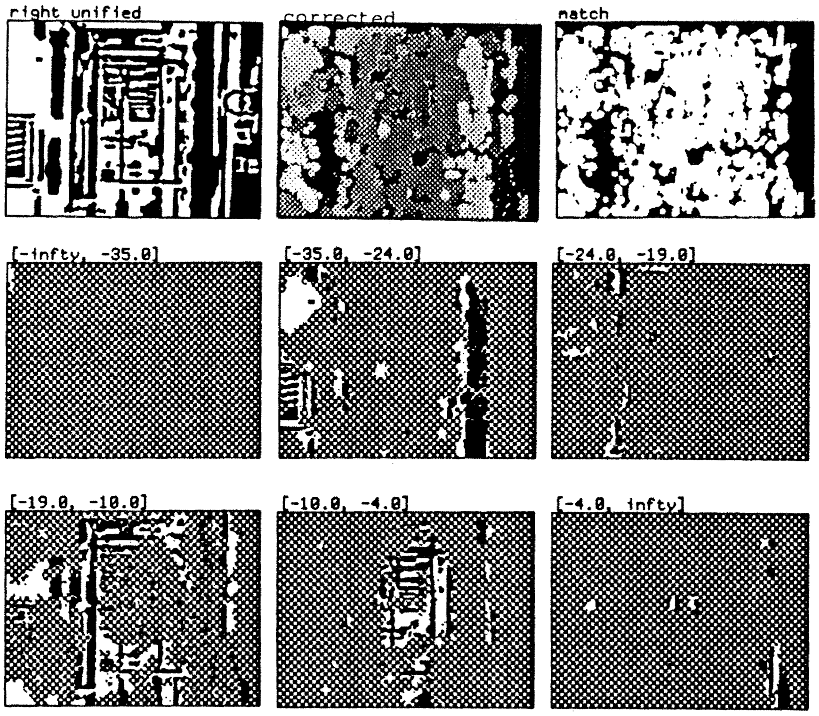


Figure 13. Matching results for the stereo pair in Figure 12, computed from the perspective of the right image.

In motion matching, the method of adjusting image alignments used in stereo analysis is no longer applicable. Although this technique could be used to correct for the effects of camera motion, different objects in the scene may be moving independently, both vertically and horizontally. A full motion analysis algorithm would probably need to make the adjustments only for a limited part of the image, determined on the basis of the reasoner's current interest. This seems to be how people handle the problem of independently moving objects.

The stereo algorithm was adapted in two ways for the motion experiment. First, image adjustment was eliminated. Coarse-scale suggestions were accepted without pruning or other alteration. However, to avoid excessive computation time, a smaller search area was specified. Since motion analysis has no favored

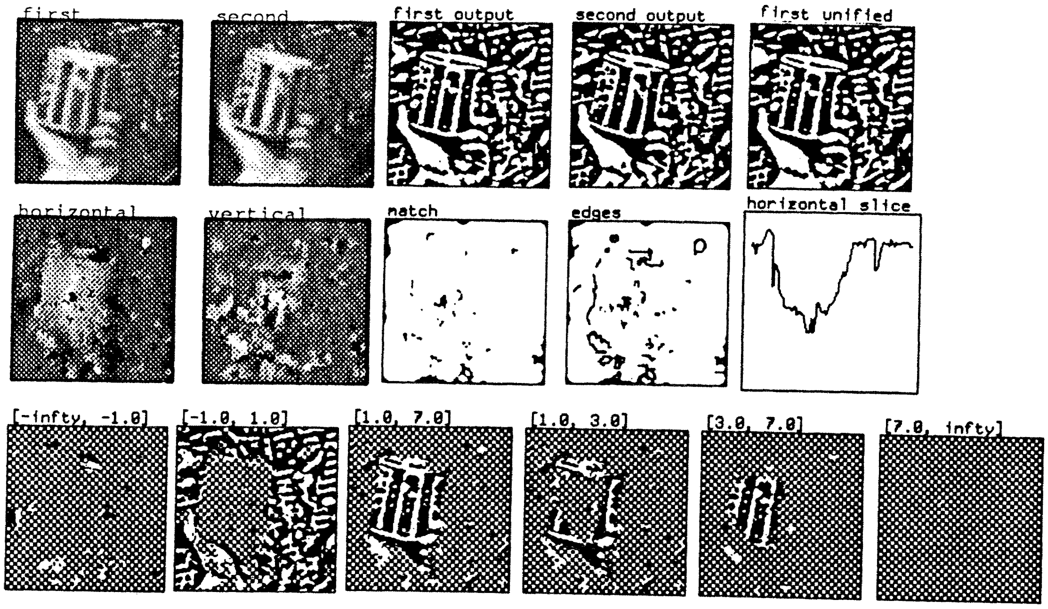


Figure 14. Matching results for two frames from a motion sequence, computed using a modified version of the stereo matcher. The motion involved rotation of the cup about a vertical axis. Although the motion is almost entirely horizontal, the matcher was not informed of this. The horizontal slice through the horizontal disparity field shows that the reconstructed motions are plausible.

direction of motion, the analysis at each scale considered both horizontal and vertical displacements up to ±2 cells from coarser-scale estimates. Although the motion in this image happens to be horizontal, the program had no biases favoring this direction.

As you can see from Figure 14, the algorithm performs successfully on the motion images. It has correctly reconstructed no motion for the background and some motion for all of the cup. The computed motion increases smoothly towards the center of the cup, which is consistent with a rotation about the center of the cup. This is most visible in the horizontal slice, shown in the second row of the figure. This slice was taken about $\frac{1}{3}$ of the way from the top of the image and cuts across the middle of the cup, avoiding the unmatched patch in the middle of the handle.

6. Conclusions

In this chapter, we have seen that the new stereo algorithm can successfully match both synthetic and natural stereograms, as well as successive frames from a motion sequence. It can successfully handle stereo pairs with large vertical disparities (up to 16 cells) and rotation (up to 5 degrees) without becoming confused. It can reconstruct disparities when the stereo correspondence cannot be one-to-one and when the image features are sparse. Where there are sharp changes in disparity, the algorithm correctly reconstructs the changes as sharp.

These results confirm the two expected benefits of using topological structure in the image matcher. First, the constraint that matches preserve topological structure, together with the strength assessments based on star-convex sum, results in robust evaluations of matching strength. Secondly, because strength assessments and boundary motion computations are confined to regions that match at approximately the same disparity, results for cells near depth discontinuities are not contaminated by values from cells on the other side of the discontinuity.

Chapter 11: The main mathematical proofs

In this chapter, I provide formal definitions for all mathematical concepts used in the rest of the thesis, as well as formal proofs for facts required in discussion or algorithm analysis. The main theorem proved in this chapter is that a structure-preserving mapping between the adjacency structures of two cell complexes determines a homeomorphism between their underlying spaces. This result allows me to show that two cell complexes are homeomorphic by comparing their adjacency structures, which is typically simpler than building homeomorphisms directly.

Section 1 gives some basic defintions and lemmas. Sections 2 and 3 develop the two combinatorial representations for cell complexes and prove that they fully represent the topological structure of the underlying spaces. Section 4 defines the open-edge model of boundaries and discusses the definitions of path and region connectivity. Sections 4 and 5 develop methods for proving that two cell complexes are homeomorphic. Section 5 also defines the closed-edge model of boundaries. Finally, Section 7 compares my representations to those proposed by previous researchers.

1. Notation and preliminary definitions

Notational conventions vary somewhat. Some of my conventions, largely borrowed from Munkres (1984), are as follows:

- B^m is the closed unit m-ball.
- Int X is the topological interior of X.

- $\overline{X}$ is the topological closure of X.
- The boundary of X, Bd X, is $\overline{X}$ – Int X.
- S^n is the unit n-sphere.

The starting point for my discussion of space-filling cells is regular cell complexes. These complexes offer the advantage of being well understood, well behaved topologically, and not substantially more general than the cases I consider. The following definition is paraphrased from Munkres (1984:214,216):

> **Definition**: A *regular cell complex* is a Hausdorff[1] space X and a collection of disjoint open cells e_α whose union is X such that:
>
> (1) For each open m-cell e_α of the collection, there exists a homeomorphism $f_\alpha : B^m \rightarrow X$ that carries Int B^m onto e_α and Bd B^m onto the union of finitely many open cells, each of dimension less than m, and
>
> (2) A set A is closed in X if $A \cap \overline{e}_\alpha$ is closed in $\overline{e}_\alpha$ for each α.

The following useful facts and definitions relating to regular cell complexes are taken from Munkres (1984:214-221) and Massey(1980:76-104):

(a) $\overline{e}_\alpha = f_\alpha(B^m)$.

(b) Bd $e_\alpha = f_\alpha(\text{Bd } B^m)$.

(c) For any n-cell e_α, Bd e_α is the union of closures of (n-1)-cells.

(d) e_α is a *face* of e_β if $e_\alpha \subseteq \overline{e}_\beta$. e_α is a *proper face* of e_β if $e_\alpha \subseteq \overline{e}_\beta$ and $e_\alpha \neq e_\beta$.

(e) If e^n is an n-cell and e^{n+2} is an (n+2)-cell such that e^n is a face of e^{n+2}, then there are exactly two (n+1)-cells e^{n+1} such that e^n is a face of e^{n+1} and e^{n+1} is a face of e^{n+2}.

[1] A space is *Hausdorff* if for each pair of distinct of distinct points x and y, there exists an (open) neighborhood U_x of x and an (open) neighborhood U_y of y such that U_x and U_y are disjoint.

(f) The *dimension* of a regular cell complex is the largest dimension of any of its cells. If no largest dimension exists, the complex is said to have infinite dimension.

(g) The *n-skeleton* of X, written X^n, is the regular cell complex consisting of all cells of X whose dimension is less than or equal to n.

For later proofs, I need a few more simple facts:

Lemma 1:

(i) Every n-cell e_n has at least one face of each dimension less than or equal to n.

(ii) All proper faces of an n-cell have dimension strictly less than n.

(iii) The face relationship is a partial order (i.e. reflexive and transitive).

(iv) The face relationship is anti-symmetric, i.e. if e_α is a face of e_β and e_β is a face of e_α, then $e_\alpha = e_\beta$.

(v) If e_n is an n-cell and e_r is an r-dimensional face of e_n, then there is a sequence of cells $e_n = e^n, e^{n-1}, ..., e^r = e_r$ such that e^i has dimension i for $r \leq i \leq n$ and e^i is a face of e^{i+1} for $r \leq i < n$.

Proof:

(i), (ii), and (v): use property (c) and induction.

(iii) follows directly from the definition of *face*.

(iv) follows from (ii).

EOP

2. Incidence structures

Although metric structure (e.g. cell shape, cell area, inter-cell distances) may be required for some algorithms, the metric information available in practical applications rarely has the precision needed to deduce the topological structure.

I therefore develop a representation for cell complexes that is independent of their metric structure. This representation is useful for manipulating topological properties, such as connectedness.

I first describe a combinatorial representation that handles all regular cell complexes and then develop a second one that is more convenient in form but handles only a restricted class of complexes. The first representation is defined as follows:

Definition: The *incidence structure* of a regular cell complex X consists of a list of all cells in X together with an incidence relation Face on this set of cells such that $\text{Face}(x, y)$ if and only if x is a face of y.

The incidence structure of a regular cell complex specifies that complex up to homeomorphism. The proof consists of several pieces. First:

Lemma 2: The dimensions of all cells in a regular cell complex can be deduced from its incidence structure.

Proof: The proof is by induction on the dimensions of the cells. By Lemma 1, parts (i) and (ii), the zero-dimensional cells are exactly those cells e_α such that $\{e_\beta | \text{Face}(e_\beta, e_\alpha)\} - \{e_\alpha\}$ is empty.

Suppose that we have identified all the cells of dimension less than n, where $n > 0$. By the same two parts of Lemma 1, the n-dimensional cells must be exactly those cells e_α such that $\{e_\beta | \text{Face}(e_\beta, e_\alpha)\} - \{e_\alpha\}$ contains only cells of dimension less than n and at least one cell of dimension $(n - 1)$. This criterion can be used to identify all n-cells because all cells of dimension less than n have already been identified. EOP

The construction of a homeomorphism between two regular cell complexes depends crucially on the following property of n-cells:

Lemma 3: A homeomorphism between the boundaries of two n-cells can be extended to a homeomorphism between the two n-cells.

Proof: Let X and Y be two n-cells and let $f : BdX \rightarrow BdY$ be a homeomorphism. The crucial point to note is that X and Y are both homeomorphic to B^n, by the definition of a regular cell complex. B^n is homeomorphic to the cone C^n formed as a quotient space from $S^n \times [0,1]$ by identifying all points $\{(x,1)\}$ to a single point. So, let $h_X : X \rightarrow C^n$ and $h_Y : Y \rightarrow C^n$ be homeomorphisms. f induces a homeomorphism $f' :$ Bd $C^n \rightarrow$ Bd C^n, namely $f' = h_Y \circ f \circ {h_X}^{-1}$. See Figure 1. f' can be extended to a homeomorphism $g' : C^n \rightarrow C^n$ by defining $g'((x,i)) = (f(x,0),i)$. g' then induces a homeomorphism $g = {h_Y}^{-1} \circ g' \circ h_X$ from X to Y. EOP

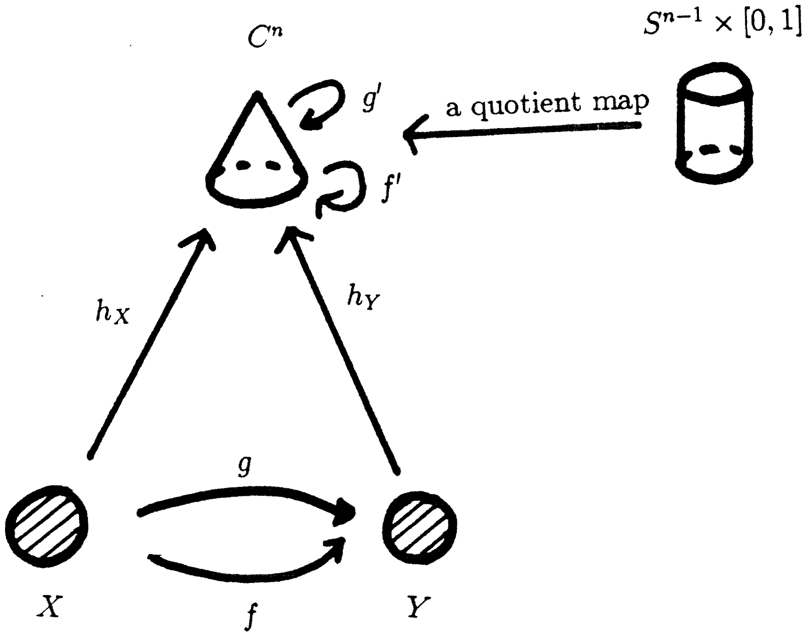


Figure 1. A picture of the functions involved in the proof of Lemma 3.

These pieces can now be assembled into a proof of the final result:

Theorem 1: Let X and Y be regular cell complexes and let Face_X and Face_Y be the incidence relations on their cells. Suppose further that f is a bijection from the cells of X onto the cells of Y that preserves the incidence structure, i.e. such that $\text{Face}_Y(f(e_\alpha), f(e_\beta))$ holds if and only if $\text{Face}_X(e_\alpha, e_\beta)$ holds, for any cells e_α and e_β. Then there is a homeomorphism between X and Y that maps every cell e_α onto $f(e_\alpha)$.

Proof: Because f preserves the incidence structure and this determines cell dimensions (Lemma 2), $f(e_\alpha)$ and e_α must have the same dimension, for each cell e_α. For each dimension $n \geq 0$, I construct a homeomorphism $F_n : X^n \to Y^n$. The construction is by induction on n and F_n is equal to F_{n-1} on the (n-1)-skeleton of X. If X and Y have finite dimension r, F_r is the required homeomorphism from X to Y. If X and Y have infinite dimension, we can define $F_\infty : X \to Y$ such that $F_\infty(x) = F_r(x)$ whenever $x \in e_\alpha$ for some r-cell e_α. Since the topology of a regular cell complex is coherent with its n-skeletons, f_∞ must be a homeomorphism (see Munkres 1984:10,220-221).

Constructing F_0 is trivial, since each 0-cell has only one point. If $\{x\} = e_\alpha$, then $F_0(x)$ is the point y such that $\{y\} = f(e_\alpha)$.

Suppose the homeomorphism F_{n-1} has been constructed, for some n, and that F_{n-1} maps each cell e_α of X^{n-1} onto $f(e_\alpha)$. Because f preserves the incidence structure, F_{n-1} must map the boundary of each n-cell e_α onto the boundary of $f(e_\alpha)$. By Lemma 3, F_{n-1} can be extended to a homeomorphism of all of e_α onto $f(e_\alpha)$. Since the n-cells are all disjoint, this extension can be done independently for all n-cells, yielding a function $g^{e_\alpha} : e_\alpha \to f(e_\alpha)$, for each n-cell e_α, which agrees with F_{n-1} on the boundary of e_α. F_{n-1} and the g^{e_α} can be pasted together into one bijective function F_n, since they agree in value where their domains intersect. Since a function on a regular cell complex is continuous if its restriction to each closed cell is continuous (Munkres 1984:215) and F_n is a homeomorphism on X^{n-1}, F_n must be a homeomorphism. Furthermore, F_n maps each cell e_α of X onto $f(e_\alpha)$. EOP

Although the incidence structure of a regular cell complex determines it up to homeomorphism, not all incidence structures determine regular cell complexes. If the incidence structure *is* legitimate, the cell complex can be reconstructed from it inductively, by attaching n-cells to its (n-1)-skeleton, for each n. If the complex has infinite dimension, it can be constructed as the coherent union of its n-skeletons, for all $n \geq 0$, as in Munkres (1984:220-221). Two conditions must be satisfied in order for such a construction to succeed:

(1) The procedure described in Lemma 2 must assign exactly one dimension to each cell (e.g. the face relation cannot have any ordering loops), and

(2) If e_α is an m-cell, the subset of the newly-constructed (m-1)-skeleton that is supposed to form the boundary of e_α, i.e. those cells designated in the incidence structure as proper faces of e_α, must be homeomorphic to an (m-1)-sphere.

I do not know of a succinct way to express the second condition combinatorially.

Luckily, in this thesis, I only need to use 2D cell complexes. For these complexes, the conditions have a simple form:

Fact 1: The following conditions define when a 2D incidence structure determines a regular cell complex:

(1) A 0-cell has no proper faces.

(2) A 1-cell has exactly two proper faces.

(3) The proper faces of a 2-cell consist of a number of 1-cells, call them $\{e_1 ... e_r\}$, together with the same number of 0-cells, $\{v_1 ... v_r\}$, such that (i) for

all i, $0 \leq i < r$, v_i is a face of e_i and e_{i+1}, (ii) v_r is a face of e_r and e_1, and (iii) there are no other face relationships.

Unfortunately, the possibilities rapidly get more complicated in higher dimensions.

3. Adjacency structures

In the incidence structure of a regular cell complex, cells of different dimensions are treated similarly. In practical applications, however, representations of space are often viewed as a set of space-filling cells of some uniform dimension, together with a description of how they are connected to one another. For example, in computer vision, the domain of a function is the set of pixels in an image, not the set of pixels and pixel faces. The goal in designing the *adjacency structure* representation is for the combinatorial representation to reflect this distinction between cells and connections (regions and boundaries).

I first define the adjacency structure for an arbitrary regular cell complex and then establish conditions under which the full incidence structure can be reconstructed from it. The main definition is:

Definition: If X is a regular cell complex, the *n-adjacency map* is the map Adj_n : {cells of X} $\rightarrow$ {sets of n$-$cells of X} such that $\mathrm{Adj}_n(e_\alpha)$ is the set of all n-cells of X of which e_α is a face. The *n-adjacency structure* of X is the image of Adj_n and an element of the n-adjacency structure is called an *n-adjacency set.*

When the intended dimension n is clear from context or does not matter, I drop the prefix or subscript "n" from this family of terms. In this thesis, I always consider spaces that are the union of N-cells, for some fixed dimension N, and use the N-adjacency structure. Examples of these structures were presented in Chapter 2.

An immediate consequence of the definitions is the following:

Fact 2: For any n, if e_α is a face of e_β, then $\mathrm{Adj}_n(e_\beta) \subseteq \mathrm{Adj}_n(e_\alpha)$.

Proof: Because the face relationship is transitive (Lemma 1, part iii). EOP

This suggests how the two representations are related. When the converse holds as well, the two representations are equivalent. Specifically:

Lemma 4: If Adj_n is the n-adjacency map of a regular cell complex and e_α is a face of e_β whenever $\mathrm{Adj}_n(e_\beta) \subseteq \mathrm{Adj}_n(e_\alpha)$, then the incidence structure of the complex can be deduced from its adjacency structure.

Proof: We are given that e_α is a face of e_β if and only if $\mathrm{Adj}_n(e_\beta) \subseteq \mathrm{Adj}_n(e_\alpha)$. This, together with Lemma 1 (iv), implies that Adj_n is one-to-one and thus it maps the set of cells bijectively onto the adjacency structure. So the adjacency sets are effectively a list of the cells in the complex. Furthermore, inclusion relationships among them determine which cells are faces of one another and so specify the incidence relation. EOP

Directly checking the condition in Lemma 4 may be difficult. The following lemma specifies a more practical set of conditions:

Lemma 5: If X is a regular cell complex and Adj is its N-adjacency map and

(1) X is the union of a set of closed N-cells,

(2) Each (N-1)-cell is a face of at least two N-cells, and

(3) The intersection of any set of closed N-cells is exactly one closed cell or empty,

then, for any two cells e_α and e_β in X, $\text{Adj}(e_\alpha) \subseteq \text{Adj}(e_\beta)$ implies that e_β is a face of e_α.

Proof: For any cell e_α, define $\text{MAdj}(e_\alpha)$ to be the cell such that $\overline{\text{MAdj}(e_\alpha)} = \bigcap_{e_i \in \text{Adj}(e_\alpha)} \overline{e_i}$, which exists because of conditions (1) and (3). In an arbitrary regular cell complex, $\text{MAdj}(e_\alpha)$ can be different from e_α. Figure 2 shows an example of a 1-cell B that belongs to only one 2-cell A in a 2D complex. In this example, $\text{MAdj}(B)$ is A, rather than D, because A is the only 2-cell to which B belongs. The proof of Lemma 5 largely consists of showing that the three conditions force $\text{MAdj}(e_\alpha)$ to be e_α.

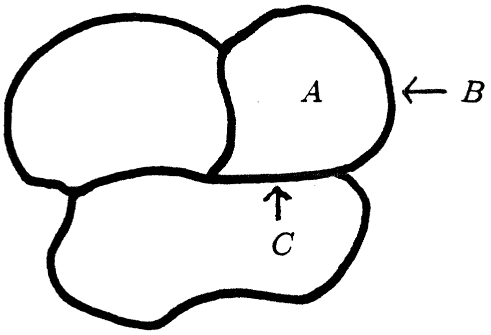


Figure 2. In this complex, $\text{MAdj}(C) = C$ but $\text{MAdj}(B) = A$. Thus, in an arbitrary regular cell complex, $\text{MAdj}(e_\alpha)$ is not necessarily e_α.

First, note the following useful facts:

(i) e_α is a face of $\text{MAdj}(e_\alpha)$.

(ii) $\text{Adj}(e_\alpha) \subseteq \text{Adj}(e_\beta)$ implies that $\text{MAdj}(e_\beta)$ is a face of $\text{MAdj}(e_\alpha)$.

(iii) Every cell e_α of dimension n, $n \leq N-1$, is a face of at least two cells of dimension $n+1$.

Property (i) holds because $e_\alpha \subseteq \overline{e_i}$ for all $e_i \in \text{Adj}(e_\alpha)$ and thus $e_\alpha \subseteq$

$\bigcap_{e_i \in \mathrm{Adj}(e_\alpha)} \overline{e_i}$. Property (ii) holds because

$$\overline{\mathrm{MAdj}(e_\beta)} = \bigcap_{e_i \in \mathrm{Adj}(e_\beta)} \overline{e_i} \subseteq \bigcap_{e_i \in \mathrm{Adj}(e_\alpha)} \overline{e_i} = \overline{\mathrm{MAdj}(e_\alpha)}$$

Property (iii) holds for all cells of dimension less than $N - 1$ by Condition (1), Lemma 1(v), and Property (e) of regular cell complexes. Condition (2) stipulates it directly for (N-1)-cells.

Now, if we can show that MAdj(e_α) = e_α for any cell e_α, the conclusion of the lemma follows, by Property (ii). The proof that MAdj(e_α) = e_α is by reverse induction on the dimension of e_α.

Base:

If e_α is an N-cell, then Adj(e_α) = $\{e_\alpha\}$. Thus, the definition of MAdj directly implies that MAdj(e_α) = e_α.

Inductive step:

Suppose that MAdj(e_α) = e_α for all cells e_α of dimension larger than some n, $0 \leq n < N$. Let e_α be an n-cell.

The cell e_α is the face of two distinct (n+1)-cells e_β and e_γ by Property (iii). If e_α is a face of e_β then Adj(e_β) $\subseteq$ Adj(e_α) by Fact 2, which implies that MAdj(e_α) is a face of MAdj(e_β) by Property (ii). Similarly, MAdj(e_α) is a face of MAdj(e_γ) But, by the inductive hypothesis, MAdj(e_β) = e_β and MAdj(e_γ) = e_γ. Thus, MAdj(e_α) is a face of both e_β and e_γ.

But, by Property (i), e_α is a face of MAdj(e_α). Thus, we have the face relationships shown in Figure 3. By Lemma 1 (ii), either MAdj(e_α) is an n-cell or an (n+1)-cell. Continuing to use the implications of Lemma 1 (ii), if MAdj(e_α) is an (n+1)-cell, it must be equal to both e_β and e_γ, which is impossible. Thus, it must be an n-cell and equal to e_α.

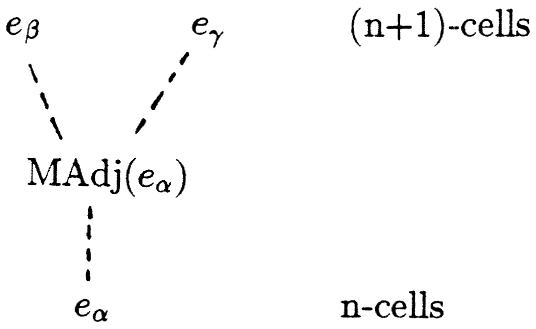


Figure 3. The n-cell e_α is a face of MAdj(e_α), which in turn must be a face of both the (n+1)-cell and e_β and the (n+1)-cell e_γ.

EOP

Since the regular cell complexes meeting the conditions of Lemma 5 are useful in my applications, I define a name for them:

Definition: An *N-space structure* is a regular cell complex such that

(1) It is the union of a set of closed N-cells,

(2) Each (N-1)-cell is a face of at least two N-cells, and

(3) The intersection of any set of closed N-cells is exactly one closed cell or empty.

Unless there is some explicit statement to the contrary, adjacency maps for N-space structures are always N-adjacency maps.

We can summarize most important consequence of Theorem 1, Lemma 4, and Lemma 5 as:

Theorem 2: If X and Y are N-space structures and if there is a bijection between the N-cells of X and the N-cells of Y that preserves the adjacency relation, then X and Y are homeomorphic.

Finally, notice that the same constructions can also be used to specify regular cell complexes that are subsections of N-space structures. Such a *partial N-space structure* is specified by (1) a list of N-cells in it, (2) a list of all N-cells that share an adjacency set with the N-cells in list (1), and (3) a list of all adjacency sets from the N-space structure that contain cells in list (1). I refer to the union of the closed N-cells in list (1) as the *region* represented by the partial structure and union of the closed N-cells in list (2) as the *border* of the structure. Representing the topology of the underlying space of the region requires naming all N-cells in the border in addition to the N-cells actually in the region. Specifically, we have:

> **Corollary 1**: Suppose that X and Y are partial N-space structures and f is a bijection between the N-cells in both the region and the border of X and the N-cells in both the region and border of Y. If f preserves the (partial) adjacency relation, then the regions represented by X and Y are homeomorphic.

To see why this is true, first note that the conclusions of Lemma 4 and Lemma 5 hold for the partial structure because they hold for the full structure. Secondly, if f preserves the partial adjacency relations given in the specifications of X and Y, then it induces a map between partial incidence structures that preserves the incidence relation. Finally, if the adjacency set of a cell is listed in (3), then the adjacency sets for all of its faces must be in list (3) also. Thus, list (3) defines a regular cell complex. Therefore, the construction in Theorem 1 will succeed.

4. Boundaries and connectivity

In the previous sections, we built a formal model for space or subsets of space. In Chapter 2, I argued that we must be able to add topological boundaries to space in order to represent locations of sharp changes in properties or lack of material connectivity. In this section, I specify how to add such boundaries to cell complexes and to their combinatorial representations, using an "open-edge" model of boundaries. This definition is then used to formalize the notions of region connectivity and structure-preserving functions.

Boundaries are added to combinatorial representations of cell complexes as follows:

> **Definition**: A regular cell complex *with boundaries* is a regular cell complex together with a list of cells called the *boundaries* of the complex.

If the cell complex is an N-space structure, its boundaries can be specified as a list of adjacency sets, since these are in one-to-one correspondence with the cells. I refer to either type of list as "the boundaries" of a structure. In the discussion that follows, I largely limit myself to N-space structures, though analogs of some definitions and lemmas may hold for more general cell complexes.

Reasoning about cell complexes primarily uses this combinatorial definition of boundaries. However, in order to relate combinatorial definitions of concepts such as "connectedness" or "continuity" to the standard mathematical definitions, we need a model of how the underlying space is changed when boundaries are added. In this section, I use the "open-edge" model of boundaries, defined as follows:

> **Definition**: If X is a regular cell complex and B is a set of cells of X designated as boundaries, the *open-edge model* of X with boundaries B is the set of points in X that do not belong to any boundary cell, with the topology it inherits as a subspace of X.

In other words, we delete the boundary cells from space, as shown in Figure 4. I use the open-edge model in this section, because it is easy to construct for arbitrary regular cell complexes. For the types of cell complexes used in practical reasoning, other models are possible. These alternatives are discussed in Section 6.

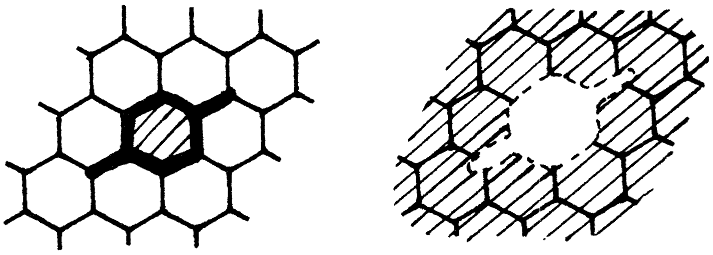


Figure 4. In the open-edge model, when a cell is added to the list of boundaries, all points in it are deleted from the underlying space.

Now that we know what boundaries are, we can define connectivity of paths and regions:

Definition: Two N-cells in an N-space structure with boundaries are *adjacent* if they belong to a common adjacency set.

Definition: Two adjacent N-cells in an N-space structure with boundaries are *connected* if they belong to a common non-boundary adjacency set.

Definition: A *connected path* in an N-space structure with boundaries is a finite ordered list of N-cells $\{A_1, ..., A_r\}$, such that A_i and A_{i+1} are connected, for $1 \leq i < r$. The path is also said to *connect* A_1 and A_R.

Definition: A set of N-cells X in an N-space structure with boundaries is *connected* if there is a path connecting every pair of cells in X, all of whose elements belong to X.

For the open-edge model of boundaries, these combinatorial definitions of connectivity are equivalent to the standard topological definitions. Specifically:

Lemma 6: Two N-cells A_1 and A_r in an N-space structure are connected by some combinatorial path $\{A_1, ..., A_r\}$ if and only if any pair of points $a_1 \in \overline{A}_1$ and $a_r \in \overline{A}_r$ can be connected by a path in the underlying space.

Proof: Building a path in the underlying space given an N-cell path is trivial. If a path in the underlying space is given, note that it can only intesect finitely many N-cells, because it is compact. These N-cells form the N-cell path and their ordering can be deduced from their intersections with the given path. EOP

Lemma 7: If a set of N-cells in an N-space structure is connected combinatorially, the union of the closures of these cells is connected in the underlying topological space.

Proof: These spaces are locally path connected, so path-connectedness and connectedness are equivalent. EOP

In practical reasoning, boundaries in an N-space structure must be deduced from sharp changes in the value of some function, e.g. image intensity or edge finder labels. This information typically comes in the form of a test for whether a pair of adjacent cells "contrast." An algorithm must be given a way to convert these pairwise contrasts into a set of boundaries for the structure. I do this as follows:

Definition: An adjacency set $\mathcal{B}$ belongs to the boundaries *induced by* a pairwise contrast relation R exactly if there exist two cells A and B in $\mathcal{B}$ such that $R(A, B)$ holds.

Boundaries induced by a pairwise contrast relation have restricted form. Specifically:

Definition: An N-space structure with boundaries is said to satisfy the

Subset Condition if an adjacency set belongs to the boundaries whenever any of its subsets does.

Fact 3: The boundaries induced by any pairwise contrast relation satisfy the subset condition.

This method of constructing boundaries from pairwise contrasts avoids connectivity paradoxes found in previous models, such as those discussed in Chapter 8.

In addition to pairwise boundaries, practical reasoning algorithm can sometimes deduce that certain entire cells belong in the boundaries. Uses for boundary cells were discussed in Chapters 2 and 4. If a cell X is specified as a boundary cell, I treat X as contrasting with itself. If a cell constrasts with itself, all adjacency sets containing that cell must belong to the boundaries induced by that contrast relation.

In the open-edge model of boundaries, the Subset Condition imposes interesting restrictions on the form of the underlying space when boundaries are present. First, note that the Subset Condition implies that if a cell is in the boundaries, so are its faces. Thus, the set of points belonging to boundary cells is closed and so the sub-space remaining after they have been removed is an open subset of empty space. If the original cell complex is an N-manifold, then the open-edge model of that complex with any set of boundaries must also be an N-manifold.

5. Structure-preserving functions and subdivision

For reasoning about topological structure, a combinatorial definition of homeomorphism is also required. This concept is somewhat difficult to express in terms of the cell structure. The natural definition for a cell structure is:

Definition: A function from the N-cells of one N-space structure onto the N-cells of another is *structure-preserving* if (1) it is bijective, (2) it preserves the adjacency structure, and (3) it preserves the labelling of adjacency sets as boundaries or not boundaries.

Using the results of Section 3, we can show the following:

Lemma 8: If there is a structure-preserving function between the N-cells of one N-space structure and the N-cells of another N-space structure, then the underlying spaces of the two structures are homeomorphic.

Proof: Use the construction from Theorem 2 to construct a homeomorphism f between the two complexes assuming there are no boundaries. f maps each cell onto the corresponding cell. Thus, if the construction of the open-edge boundary model deletes a point x in one cell complex, it must delete $f(x)$ in the other complex, and vice versa. EOP

The problem with structure-preserving functions is that they only exist when the two spaces have the same cell structure, in addition to being homeomorphic. However, I do not believe that the general problem of determining whether two spaces are homeomorpic is computationally tractable. Certainly people cannot determine easily whether two regions are homeomorphic, unless their shapes are similar. Therefore, I develop classes of modifications to the cell structure that are guaranteed to preserve the topology of the underlying space. It may not be possible to relate all homeomorphic cell structures using these techniques, but they cover a range of cases that are useful in designing practical algorithms.

The simplest way two cell complexes can be related is if one is a subdivision of the other:

Definition: A regular cell complex X is a *subdivision* of another regular cell complex Y if every cell of X is contained in a cell of Y and every cell of Y is the union of finitely many cells of X.

This definition is adapted from the definition of subdivision for simplicial complexes given by Munkres (1984, p. 83). It forces the two cell complexes to share the same underlying points. Furthermore, the finiteness condition forces them also to share the same topological structure. Thus, if we can find a structure-preserving map between subdivisions of two cell complexes, that is enough to establish that the two complexes are homeomorphic.

The corresponding combinatorial notion is that of a subdivision mapping between two cell complexes:

Definition: If Y is an N-space structure and X is a subdivision of Y, then the *subdivision mapping* is the map Sub from the N-cells of X onto the N-cells of Y such that Sub(A) contains A for every N-cell A in X.

Suppose that we are given a map Sub from the N-cells of an N-space structure X onto the N-cells of another N-space structure Y. We would like to be able to determine whether Sub could be a subdivision mapping between these structures (or structures homeomorphic to them) by examining its combinatorial properties. Sub induces a mapping SubAdj from the adjacency sets of X onto the adjacency sets of Y. What we need to verify is that the underlying space corresponding to SubAdj$^{-1}(A)$ is homeomorphic to an r-ball, for each r-cell A in Y. Since each cell can be subdivided into at most finitely many cells, the legitimacy of any subdivision can be checked by using the following fact and induction:

Fact 4: An open r-ball is homeomorphic to two open r-balls together with an open (r-1)-ball that is their common face, for any $r > 0$.

The definitions of subdivision that I have just given were for cell complexes with no boundaries. When boundaries are added, the boundaries in the cell complex and its subdivision must correspond in order for their underlying spaces to be homeomorphic. Specifically:

Definition: A subdivision mapping Sub from the N-cells of an N-space structure X to the N-cells of an N-space structure Y *preserves the boundaries* if an adjacency set $\{A_i\}$ is marked as a boundary in X if and only if $\{f(A_i)\}$ is marked as a boundary in Y.

If a subdivision relationship preserves the boundaries, then it induces a homeomorphism of the underlying spaces produced by the open-edge model of boundaries. Thus, we can now prove two complexes to be homeomorphic if we can find subdivisions of them that preserve the boundaries and that can be related to one another via a structure-preserving mapping.

6. Thickening boundaries and closing them

Many pairs of complexes representing homeomorphic spaces can be related via common subdivisions. However, there are pairs of homeomorphic cell complexes that cannot be related by subdivision alone. An example of one type is illustrated in Figure 5. Since certain algorithms in this thesis must be able to relate such pairs of complexes, we need an additional technique for establishing homeomorphism, which I call the *thick boundary property.*

The thick boundary property is defined as follows:

Definition: If X is an N-space structure and $\mathcal{R}$ is an adjacency set of X, then CLOSURE($\mathcal{R}$) is the set of all adjacency sets of which $\mathcal{R}$ is a subset.

Definition: An N-space structure X satisfies the *thick boundary property* if, for any boundary adjacency set $\mathcal{R}$ in X, the following modifications to the adjacency structure yield a legitimate N-space structure with an underlying space (in the open-edge model of boundaries) homeomorphic to that of X:

(1) Add a new N-cell C.

(2) Remove all adjacency sets in CLOSURE($\mathcal{R}$).

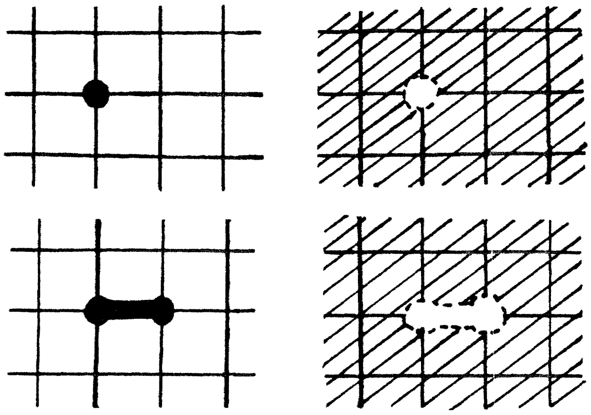


Figure 5. A 2D cell complex with one vertex deleted and a similar cell complex with one edge deleted are homeomorphic under the open-edge model, but they cannot be related via boundary-preserving subdivision alone.

(3) For each adjacency set $\mathcal{S}$ in X, if $\mathcal{S}$ is a subset of any element of CLOSURE($\mathcal{R}$), add $\mathcal{S} \cup \{C\}$ as a boundary adjacency set.

In other words, if the structure satisfies the thick boundary condition, a cell can be added "in the middle" of a boundary that did not formerly have any thickness. This construction is illustrated in Figure 6.

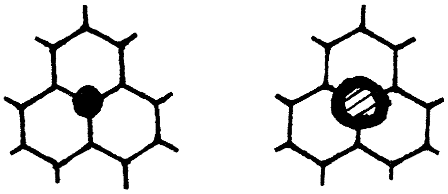


Figure 6. Thickening a boundary in an N-space structure.

N-space structures do not, in general, satisfy the thick boundary condition. For example, a 2-space structure can have three 2-cells joined along a single edge. If this edge is replaced by a 2-cell, the resulting adjacency structure will not represent any 2D regular cell complex. I do not know of any convenient combinatorial way to decide when a structure satisfies this condition. It requires, at least, a solution to the problem of determining when a cell complex is a sphere, posed at the end of Section 2.

As in Section 2, I content myself with describing when the condition holds for 2D cell complexes. These conditions are as follows:

Lemma 9: A 2-space structure satisfies the Thick Boundary Condition if

(1) Its boundary assignment satisfies the Subset Condition,

(2) The 1-cells meeting at any fixed 0-cell can be arranged into a finite list $(E_1, ..., E_r)$ in which E_i and E_{i+1} belong to a common 2-cell for every i and E_1 and E_r also belong to a common 2-cell, and

(3) Each 1-cell is a face of exactly two 2-cells.

Proof: The details of the construction depend on the dimension of the boundary cell $\mathcal{R}$ that is to be thickened. If $\mathcal{R}$ is a 0-cell, the construction is as shown in Figure 7. First, pick a point on each 1-cell of which $\mathcal{R}$ is a face and put a new 0-cell there, subdividing the 1-cell. For each 2-cell $\mathcal{S}$ of which $\mathcal{R}$ is a face, subdivide $\mathcal{S}$ with a 1-cell joining the two marked points. By condition (2), these new 0-cells and 1-cells must form a circle. The new 2-cell C is the union of all the wedge-shaped 2-cells inside this circle. The homeomorphism from the old underlying space to the new one pulls the points in C out into the area just outside the boundary of C.

Suppose now that $\mathcal{R}$ is a 1-cell. By the Subset Condition, we know that the endpoints of $\mathcal{R}$ are also in the boundaries. To construct C, first thicken the boundaries at the two endpoints of $\mathcal{R}$, adding new cells D and E. The situation is then as shown in Figure 8. Just as in the 0-cell case, we can slice off sections of the 2-cells of which $\mathcal{R}$ is a face. Since there are only two

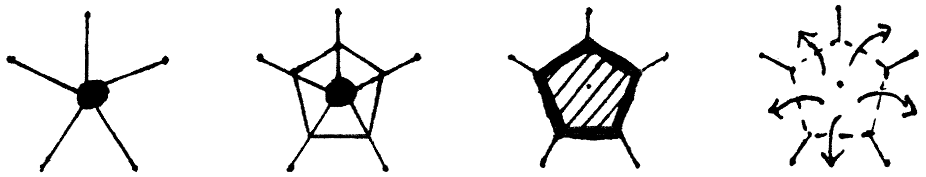


Figure 7. Thickening the boundary at a 0-cell. Left to right: the boundaries, dividing the adjoining regions, the new boundary cell, and the mapping from the old space to the new one.

such 2-cells, the two cut-off pieces can be merged to form a new cell F. The desired cell C is then the union of D, E, and F.

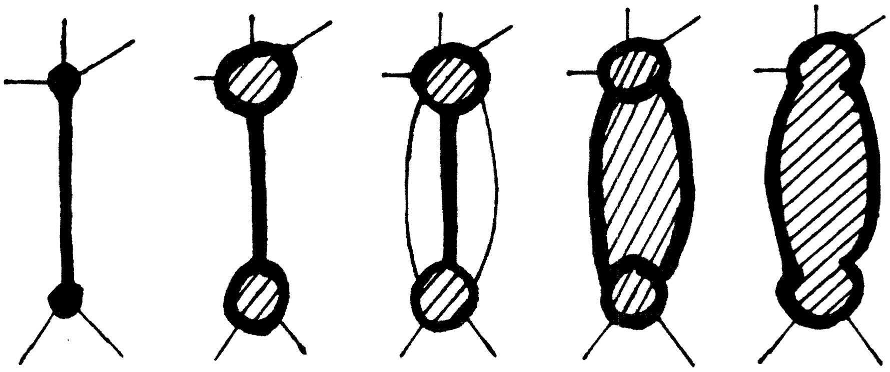


Figure 8. Thickening the boundary at a 1-cell. Left to right: the boundaries, thickening the endpoints, dividing the adjoining regions, three pieces of the new boundary cell, and the new boundary cell.

Thickening the boundary around a 2-cell requires only that its 0-cell and 1-cell faces can be thickened. C is then the union of original 2-cell and these new cells.

EOP

The boundary thickening construction also provides a second model for boundaries, the *closed-edge* model, if the cell complex and boundary assignment satisfy the Thick Boundary Condition. In order to add boundaries in this model, one first thickens all boundary cells. Then, as in the open-edge model, delete all boundary cells, but then add back in all faces of non-boundary cells. This is illustrated in Figure 9. The resulting space is just like the open-cell model, except for these extra boundary points. In fact, all of the discussion in these last three sections works as well for the closed-edge model as for the open-edge one. The only exception is that one-cell regions are no longer homeomorphic to cellular representations of $\mathbb{R}^n$ in the closed-edge model.

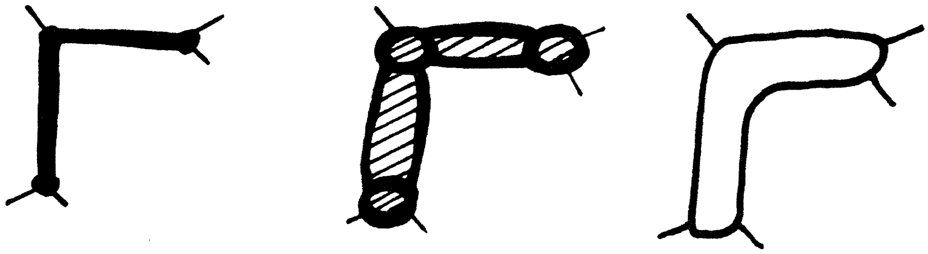


Figure 9. Constructing the closed-edge model of boundaries.

In practical reasoning, it is often useful to be able to thicken boundaries repeatedly. Thus, the following property may be useful:

Fact 5: The construction in Lemma 9, applied to a complex satisfying the Thick Boundary Condition, yields a complex that also satisfies the Thick Boundary Condition.

In particular, complexes for which the closed-edge model of boundaries can be constructed will satisfy the Thick Boundary Property, using the closed-edge model of boundaries rather than the open-edge one.

The combination of subdivision and boundary thickening is sufficient to handle the practical examples in this thesis. However, it does not relate all pairs of cell complexes with homeomorphic spaces. For example, in the open-edge model, a region consisting of one N-cell surrounded by boundaries is homeomorphic to an N-ball, and thus homeomorphic to $\mathbb{R}^n$, which might be represented by an infinite cell complex. However, subdivision only allows a cell to be divided into finitely many pieces. Thus, the one-cell region and $\mathbb{R}^n$ cannot be proved homeomorphic by the techniques I have described. I do not, however, know of any practical applications that require this ability.

7. Comparisons to previous work

In this section, I briefly review previous methods of representing the topology of digitized spaces. We see that pairwise representations are inadequate to completely specify the topological structure of a space. I also mention other work related to representing cell complexes.

Adjacency structures are similar in form to the pairwise connectivity relations used by some researchers, e.g. Pavilidis (1977), Lee and Rosenfeld (1986), and Poston (1971; based on work by Zeeman 1962). In Poston's case, the representation is not only pairwise, but also "fuzzy." He also introduces a more structured notion of a "local matroid structure," but it unclear that even this pins down the topological structure. Standard, non-fuzzy pairwise representations describe the topology of a set of N-cells by specifying which pairs of N-cells share a common face. There are two variants, one in which only pairs of N-cells sharing an (N-1)-face are included and one that includes pairs of cells sharing a face of any dimension.

Adjacency structures fully specify the topology of a set of cells, whereas pairwise connectivity relations do not. The first problem with pairwise representations is that they do not uniquely specify the dimensionality of the cell complex. For example, the two cell complexes in Figure 10 have the same pairwise structure.

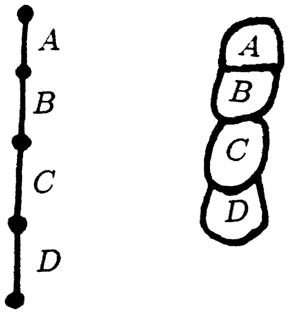


Figure 10. Two cell complexes of different dimensionality can have the same pairwise connectivity structure.

Pairwise connectivity relations are also inadequate even within one dimension. For example, consider the cell complex shown in Figure 11 (left). This cell complex consists of four 2-cells arranged in a square. If only cells sharing a 1-face are considered connected, then this cell complex has the same connectivity structure as the ring shown in Figure 11 (middle). If cells sharing any face are considered connected, then it has the same connectivity structure as the tetrahedral cell arrangement shown in Figure 11 (right). This indeterminacy in structure makes it difficult to reason about the topology of these sets of cells.

Cellular topology, on the other hand, can represent each of these three cell complexes uniquely. For example, Figure 12 shows the same sets of cells as in Figure 11 (left) and Figure 11 (middle), with appropriate border cells added. In each case, the cell complex formed by A, B, C, and D can be reconstructed exactly from the adjacency sets involving them, using the constructions given earlier in this chapter. As you can easily verify, these structures are different for

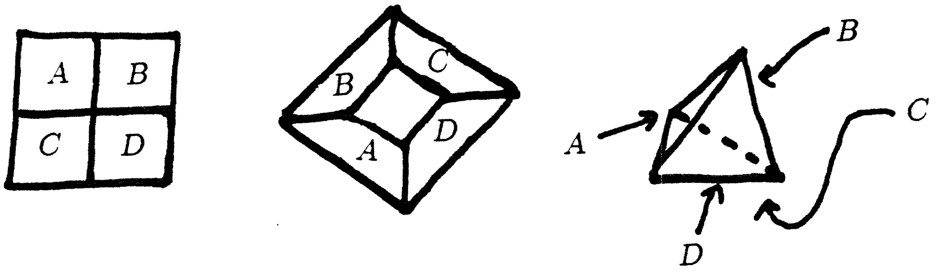


Figure 11. The set of cells on the left has the same pairwise connectivity structure as the set of cells in the middle or the set of cells on the right, depending on which definition of pairwise connectivity is used.

the two cell complexes. The same holds for the tetrahedral complex, but drawing it and its border cells is beyond my ability.

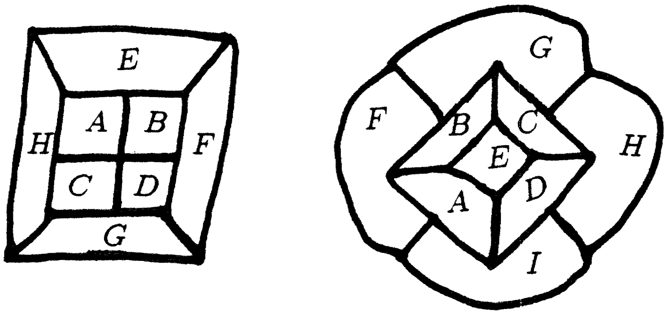


Figure 12. The two 2D complexes from Figure 11, with border cells added.

Adjacency structures are, to the best of my knowledge, original. Incidence structures seem a relatively obvious idea, given the standard development of regular cell complexes. I have not, however, seen a previous, explicit proof of their sufficiency. Gruňbaum and Shephard (1987) establish a number of related results for tilings of the plane. Their discussion was useful in formulating my proofs, even though they take advantage of the metric structure of the plane and

I use only topological properties. Baumgart's (1972) "winged edge polyhedron" representation is also similar to incidence structures, but he supplies no proof of its sufficiency, even for the 3D cases he considers.

8. Conclusions

In this chapter, I have developed all the formal machinery needed to analyze the examples presented in the rest of the thesis. I have modelled space using a slightly restricted class of regular cell complexes, called N-space structures. Two models for adding boundaries to space were presented. The open-edge model can be defined for all cell complexes and closed-edge model for a more restricted class of spaces. Definitions were presented for path and region connectivity that are equivalent to the standard definitions. The Subset Condition, useful in assigning boundaries in practical reasoning, was defined.

I have also developed methods of determining when two cell complexes have homeomorphic underlying spaces. For either model of boundaries, subdivision and structure-preserving mappings preserve the topological structure of the underlying space. When additional conditions on cell structure are met, a boundary thickening operation also preserved the topological structure. Though they do not cover all cases of homeomorphic cell structures, these techniques are sufficient for analyzing the algorithms presented in this thesis.

Chapter 12: Re-cap, conclusions, and future work

1. Introduction

In this chapter, I summarize the results presented in this thesis. Three types of results were presented. First, I developed a mathematical framework, called *cellular topology*, that makes it possible to manipulate topological structure using finite descriptions and that makes it easier to handle digitized functions. Secondly, I presented uses for topological concepts in a wide range of domains and showed also how the cell structure is useful in reasoning. Finally, I showed how these ideas translate into algorithms that can robustly process digitized camera images. I presented output from these programs and detailed testing of the edge finder implementation.

Section 2 summarizes the mathematical formalism briefly and re-caps the mathematical results from Chapters 2 and 11. Section 3 outlines the uses for topology that we have seen and Section 4 summarizes the uses for cell structure and digitized functions. Section 5 re-caps the main experimental results presented in this thesis. Finally, in Section 6, I discuss possibilities for future work extending this research.

2. Mathematical groundwork

The main hypothesis of this thesis is that boundaries change the topological structure of space. Exploring this hypothesis requires a concrete model of what changes they induce. The framework of cellular topology was developed to simplify manipulation of space and boundaries. It also provides useful assistance in managing digitized functions, such as those found in computer vision.

Cellular topology models space using regular cell complexes, a standard construct from topology. In Chapter 11, I developed two combinatorial representations for these complexes: incidence structures and adjacency structures. I proved that incidence structures fully represent the topological structure of a regular cell complex. Adjacency structures, closer to the form of representations used in practical reasoning, also fully specify topological structure under specific conditions, typically satisfied in practical applications. These representations are useful in relating data structures used in applications to the mathematical structure of the underlying space.

Using the cellular models of space, I then developed models of how the topological structure of space is changed when boundaries are added. The cellular models make this task easier, because they prevent pathological situations, such as Cantor sets, from developing and because they provide finite representations for continuous spaces. Two models of boundaries were developed, the open-edge model and the closed-edge model. There are only small practical differences between the two models. The open-edge model can be constructed under more general conditions, but the closed-edge model seems more appropriate for modelling certain phenomena, such as state changes in linguistic semantics.

In Chapter 11, I also developed techniques for proving that two cellular representations (with boundaries) are homeomorphic. These techniques were used in Chapter 5 to develop operations for moving boundary locations without changing the topological structure that they induce. These operations are crucial to building the image matcher, which is central to the implementations in this thesis. The operations cannot relate any two homeomorphic spaces, but they seem to cover the full range of case required by low-level vision algorithms.

Cellular topology avoids technical problems encountered by previous formalisms used in Artificial Intelligence research. Unlike previous pairwise representations, it fully represents the topological structure of situations. The boundary models avoid problems assigning function values to boundary points, as discussed in Chapters 7 and 8. Cellular topology also constrains the form of space and boundaries in ways that are useful in practical reasoning, without forbidding useful possibilities.

Finally, the cell structure provides a finite-resolution notion of minimal-sized moments. This is clearly useful in computer vision algorithms, which must deal directly with digitized data. Moreover, we saw that this notion was useful in modelling data for linguistic semantics and for high-level reasoning. Cellular representations for these areas allow them to use data from real sources of input, whether sensory data or experimental measurements. Paradoxes involving minimal-sized intervals about state changes, encountered by previous linguistic proposals, can be resolved using cells or, alternatively, using points and the closed-edge model of boundaries. Digitized functions also rule out certain infinite limit situations that cause problems for formal analysis of both practical reasoning algorithms and data from linguistic semantics.

3. Uses for topology

At the beginning of this thesis, I asserted that topological structure is useful for a wide range of reasoning. We have seen that this is true, although the form of the examples varies from domain to domain. Examples were presented from three areas: low-level vision, natural language semantics, and high-level vision and reasoning. These areas cover much of the current research in Artificial Intelligence, except for low-level control of manipulation and low-level language parsing (speech and syntactic analysis).

The most intuitively appealing uses for topological structure come from high-level vision and reasoning. There we saw that topological structure is useful in reasoning about flows (electrical, fluid, causal, and applied forces) and material connectivity. It is also widely used, together with metric information, in representing the shape of objects. Boundaries are crucial to the analysis of changes over time. In both time and space, we find that cellular topology correctly predicts the correlation between boundary locations and connectivity. It also provides succinct descriptions for situations in which many properties have abrupt changes at a common location, as is often the case in practical applications.

Data from linguistic semantics provides useful parallels to the data from high-level reasoning. We saw that topological boundaries are useful in modelling state changes and in distinguishing activities from achievements. The behavior of temporal connectives also provides support for the claim of cellular topology that boundaries hypothesized to account for sharp changes in property values should also interrupt region connectivity. We also saw that connectivity provides a formal explanation for the meaning of perfect aspect verb forms in English and makes it easier to explain the conditions under which the progressive aspect can be used. Finally, topological structure was crucial to the operation of the low-level vision algorithms.

Another phenomenon that we saw in all domains was that properties must often be computed using wide support neighborhoods. This is required in order to avoid aliasing and other artifacts in sampled data. It is also necessary, even assuming perfect data, for analysis of textured patterns and stereo matching.

Textured patterns occur across all domains, including textured events over time (linguistic and high-level reasoning), 2D texture in images (computer vision), and 3D texture of materials (high-level reasoning). We saw that these support regions must often be restricted so that they do not cross certain boundaries. This type of restriction, formalized using topological connectivity, is used in the stereo matcher to avoid blurring of disparity values.

4. Experimental results

Two implementations were built for this thesis: an edge finder and a stereo matching algorithm. The stereo algorithm is built around an image matcher that has other applications. In this thesis, it was also used to match images in testing the stability of edge finder output. I also presented preliminary examples indicating that the matcher may also be useful for analysis of motion sequences and for detecting texture periodicity and orientation.

Both the edge finder and the image matcher take advantage of the topological structure of images. The edge finder uses the topology induced by second difference responses to decide which responses represent real features and which are due to camera noise. Evaluation of the response at each cell is confined to a star-convex (and, *a fortiori*, connected) neighborhood about that cell, containing only responses of the same sign. This prevents evaluation of the response from being corrupted by other, nearby, responses.

The image matcher takes advantage of the full topological structure of an image, induced by boundaries marked by the edge finder, to constrain matching. Matching is done by adjusting the boundaries in one image so that they are as close as possible to those in the other image, without changing the topological structure they induce. This adjustment makes matching insensitive to small changes in boundary location caused by camera noise, changes in viewpoint, and the like. The matcher also uses the star-convex neighborhood algorithm developed for the edge finder to evaluate match strengths and suppress noise in the output disparity fields.

The edge finder implementation was extensively tested and its performance compared, in Chapter 9, to that of Canny's (1983, 1986) edge finder. The new edge finder produced results that were more stable under camera noise and had higher resolution, particular near sharp corners and region intersections. There are two novel aspects to the testing procedure. First, the tests were based on stability, rather than accuracy. This made it possible to assess edge finder performance on natural image data, rather than simple synthetic images. Secondly, the image matcher made it possible to process large amounts of image data automatically and robustly. Thus, the evaluations are based on substantially larger amounts of data and substantially more realistic conditions than previous comparative evaluations.

The stereo implementation was tested on a range of natural and synthetic stereograms. Because the implementation is slow, it was not possible to conduct quantitative tests, like those done for the edge finder. Nevertheless, I established that the matcher produces two improvements on past performance. First, its more robust measure of match strength allows it to tolerate larger search neighborhoods without becoming confused. In particular, it can successfully match images with vertical disparities (up to 16 cells) and rotation (up to 5 degrees), which has not previously been possible. Secondly, match assessments and disparity computations are confined to connected neighborhoods, which keeps them from crossing sharp changes in disparity. This prevents blurring of values near these boundaries.

5. Future work

There are several obvious directions in which this work could be extended. The existing implementations could be made to run faster. Implementations could be built for other types of vision analysis, such as texture, and for the other domains considered. Finally, the mathematical development could be extended, both to fill in gaps in the current theorems and to explore other types of structure. In this section, I discuss each of these ideas briefly.

The existing edge finder and stereo implementations are very slow. In the case of the stereo matcher, some of the blame rests on its large search neighborhoods. However, the main bottleneck is the speed of the star-convex sum operation, used extensively thoughout the algorithms. This operation is very local, and thus parallel, and not particularly complicated. Thus, there is considerable hope for improvement.

I intend to explore several approaches to making this operation faster. First, I hope to implement the algorithm on some type of specialized image processing hardware. Secondly, it may be possible to build a faster serial implementation, perhaps by scanning through the image and using moving averages. Finally, preliminary results suggest that the current algorithms uses more paths than are necessary to achieve good results. Making the algorithms faster will allow me to build more ambitious applications and do more extensive testing of existing ones (such as the stereo matcher).

If the basic operations can be made faster, it will be possible to explore more types of vision applications in more detail. I am particularly interested in analyzing object motion and image texture. Motion of 3D objects is important, because it could potentially allow a computer vision system to build 3D models of objects using only visual input. It is easiest to interpret motion data when the motions are under computer control, e.g. the object is held in a robot arm. Judging from human performance, it should also be possible to obtain useful information even when the motion is not under computer control.

A second area of interest to me is analysis of periodicity and orientation in textures. I have preliminary results suggesting that the matcher can be used to determine in which regions a textured image matches itself at a specified alignment of the images. I hope to build a control structure to search a range of image translations and analyze the results to determine texture periodicity, orientation, and perhaps region width. It might also be possible to extend this to other symmetries of the plane, such as reflections and rotations, and to relate it to work on local symmetry representations of shape. Finally, it has been suggested to me that this technique might be useful in analyzing textured data from the physical sciences.

Another set of possibilities involve building applications, or at least more detailed analyses, in high-level vision and non-vision domains. For example, it might be possible to use the topological matching framework to convert the ideas of Koenderink and van Doorn (1982) into usable algorithms. It might also be possible to build more robust versions of the accretion/deletion detector of Mutch and Thompson (1985) for motion and stereo analysis. It may also be possible to use techniques based on the Phantom edge finder to analyze input data into a form suitable for qualitative physics reasoning, along the lines suggested by Forbus (1986).

A final source of future work is extending the mathematical development. Recall that several lemmas, involving boundary thickening and deciding when a proposed cell boundary was a sphere, were only stated and proved for the 2D case. One problem for future research is either to extend these proof to higher dimensions or to show that they are, in some provable way, intractable. The formal development could also be extended to include metric or metric/topological properties. One open question, for example, is whether the space of approximately straight paths about each cell, out to some fixed radius, can be given some nice structure and, if so, what that structure is. Such structure might be useful in relating cellular topology to standard theories of differential geometry.

Appendix A: Viewing stereo pairs

The stereo pairs shown in this thesis have been arranged for crossed-eye viewing. In this appendix, I explain how to learn how to view such pairs of images so as to see depth. Alternatively, stereo viewers can be used, if available. However, depths will appear inverted if a viewer is used. That is, a depressed square will appear in place of a raised one. For some images, where the inverted depths conflict with other cues (e.g. a person appearing concave), this may seem bizarre and may affect the fusion.

Figure 1 shows a simple random-dot stereogram. If you fuse it successfully, you will see a square raised above a background, both textured with random dot patterns. The square is half the width of the stereogram, in the middle. In order to see this, you need to cross your eyes, so as to put your left eye's view of the image marked "left" on top of your right eye's view of the image marked "right, as shown in Figure 2.

A good way to learn to fuse these images is to hold a pencil partway between your eyes and the images. Look at the pencil, so it looks clear. Behind it, the images will have moved towards the right alignment for fusion, but they will look blurry. The first step in learning to fuse the images is to get the middle copies of the two images to lie exactly on top of one another, by moving the pencil forward or backwards. Keep focused on the pencil while you do this.

The second step in fusing the two images is to get them in focus, without moving them. In other words, you want your eyes to point in the right directions

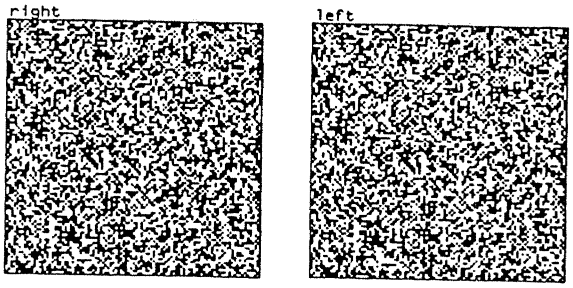


Figure 1. A simple random-dot stereogram.

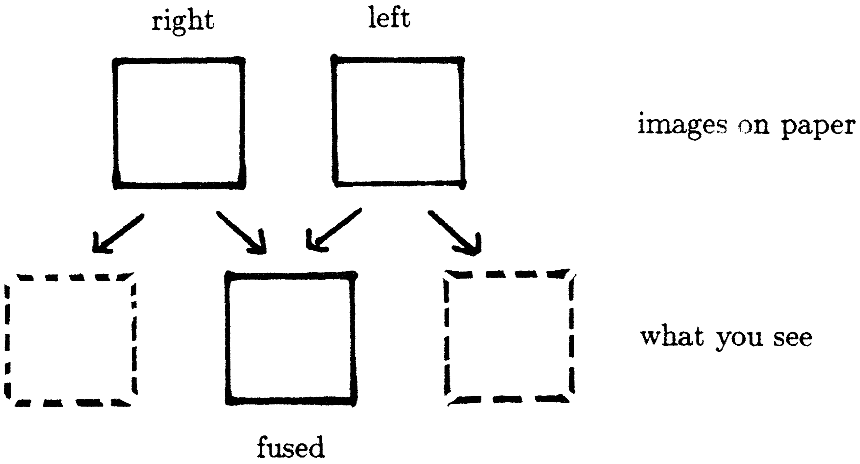


Figure 2. What you want to see.

for looking at the pencil (i.e. so it is not doubled) but make the focus adjustment on your eyes appropriate to the sheet of paper that the images are on. This is the frustrating part, because whenever the images start to focus, they also will try to separate. It takes some experimentation. Relax and try to look at the texture in the images while making sure the pencil does not separate into two pencils. Relax. Keep thinking alternately about the pencil and the image you are trying to bring into focus (the middle of the three apparent images in your field of view). Relax.

Fusing crossed-eye stereograms takes some practice. I heard one vision researcher, whose anonymity I will protect, refer to the process as "breaking one's eyes." For some of us, that is about how it feels for the first few attempts. Your visual system has spent decades learning how to make sure that vergence and focus work together, and it takes some effort to undo the effects of that practice. If you feel frustrated or your visual system feels "broken," break off, take a couple aspirin, and try again the next day. It took me several days of trying before I succeeded in fusing such stereograms.

Appendix B: Implementing boundary adjustment operations

Chapter 5, Section 2 described a set of operations for adjusting boundary locations. In the current implementation, the four adjustment operations are not implemented directly. Implementing these operations directly, for all possible rotations and reflections, would be painful to implement and would run slowly. Instead, all these possibilities are collected into one *boundary test*, which can be implemented more easily. This appendix describes this test briefly, for those who may be interested in reproducing my implementation. It also gives details of how boundary adjustment operations are applied in the matcher.

The boundary test determines whether some non-boundary cell x in the image can be changed into a boundary cell without affecting the topology. Both thickening and thinning relate an input configuration to an output configuration. In thickening, the input configuration must pass the boundary test. In thinning, the output configuration may involve changes to the boundary structure induced by changes in labels, as discussed in Section 3 of Chapter 5. This new output cell structure must pass the boundary test.

In order to determine whether the configuration about some cell x meets the conditions for one of the adjustment operations, the boundary test considers which edges and vertices of x are marked as boundaries. These edges and vertices form some number of line segments and isolated vertices. If x is entirely surrounded by boundaries, they form a circle. Since the boundaries are induced by label contrasts, an edge of x can only be a boundary if its vertices are also boundaries (see Chapter 11, Section 4). The boundary test counts the total number of "ends" in these boundaries.

Specifically, each vertex of a cell is counted as:

0 ends if it is not a boundary,

0 ends if both edges leading into it are boundaries,

1 end if one edge leading into it is a boundary, and

2 ends if it is a boundary, but neither edge leading into it is a boundary.

A cell x passes the boundary test if and only if the total count of ends (over all four vertices) is exactly 2. Proving that this test is equivalent to the operations given above is simply a matter of enumerating all possible configurations of boundaries for a square cell. There are not many distinct ones.

As described in Chapter 5, boundary adjustment consists of two phases: thickening and thinning. The thickening phase of boundary adjustment is a straightfoward application of the adjustment operations. The thinning phase is slightly more interesting, because the matcher must assign new, non-boundary labels to cells. When the matcher assigns a new label to a cell x, it must re-calculate the boundaries about that cell, taking into account any label contrasts introduced by x's new label. This calculation is done using exactly the same rules as the edge finder used for assigning the original boundaries. Only after the boundaries have been re-calculated can the algorithm check that the output configuration matches one of the four adjustment operations. Thus, the thinning operates by hypothesizing its desired labelling for x and then retracting the new label if it would change the topological structure.

The thinning phase also re-labels cells in the interior of regions of uniform color. Remember that cells in the middle of uniform-color regions may bear the label *zero*, in addition to *dark* and *light*. Thus, even if the boundaries in an image match, cell labels may still disagree near transitions to and from *zero*. This additional case in the re-labelling algorithm cleans up these mis-matches. As in re-labellings near boundaries, boundaries must be re-calculated after the new label is assigned. A cell x is re-labelled only if it is not next to a boundary in either the input or the output configuration. The cell structure is not changed at all in this type of re-labelling.

Since each boundary adjustment operation examines only a 3 by 3 neighborhood of the cell of interest, the operations can be applied in parallel. However, only one cell in four can be examined on each parallel application. Thus, each pass consists of four parallel operations, one considering cells with coordinates $(2n, 2m)$, the second considering cells $(2n+1, 2m)$, and so forth. This order of application reduces the possibilities for "runs," in which a region many cells wide is moved into or out of a boundary because it is aligned with the scanning direction. This is important even in a serial implementation. Using this order of application, the maximum amount of thickening or thinning generated by three passes varies between three and six cells, depending on the orientation of the boundary.

Appendix C: Other verbal properties

Chapter 7 presents a classification of situations in time based on their temporal structure. In addition, English makes at least two other distinctions in situation type that seem to be related more to questions of causality than to temporal structure. One distinguishes agentive from non-agentive verbs and the other distinguishes actions in which there is change over time from those not involving change over time. As Dowty (1979) noticed, there is a tendency to conflate these distinctions with distinctions in temporal structure, because there is some statistical correlation. Such a conflation, however, makes the data difficult to understand.

The first additional distinction that Dowty draws is between situations caused or controlled by an animate agent (agentive situations) and those that are not (non-agentive situations). For example, Sentence 1 is agentive, whereas Sentences 2-3 are not.

(1) Ken walked to Tech Square.

(2) The rock rolled down the hillside.

(3) Brian noticed the poster on the playroom wall.

In Sentence 2 the subject of the sentence is not (except in certain science fiction novels) capable of exercising voluntary control over the action. In Sentence 3, although the subject is animate, the action in question is not under voluntary control.

Dowty lists a number of tests (relatively well-known in linguistics) for whether a verb phrase is agentive or not. For example, only agentive verb phrases can occur in constructions of the form "persuade so-and-so to do X," as illustrated by Sentences 4-6:

(4) I persuaded Ken to walk to Tech Square.

(5) #I persuaded the rock to roll down the hillside.

(6) #I persuaded Brian to notice the poster on the playroom wall.

This distinction is important not only for linguistic semantics, but also for causal reasoning (as in Allen 1984) in which it is often important to sort out what agent is responsible for some course of events.

Dowty also draws a second distinction between actions like the one described in Sentence 7, which describes a static situation, and actions in which the world varies over time, as in Sentence 8:

(7) Marvin stood in the playroom.

(8) Marvin ate lunch in the playroom.

Some combination of this distinction and agentiveness seems to be required to explain certain "do" constructions. Consider, for example, Sentences 9-12:

(9) What Marvin did was roll down the hill.

(10) What Marvin did was lie in the grass.

(11) ?What the rock did was roll down the hill.

(12) #What the rock did was lie in the grass.

This *Pseudo-cleft* construction is best with an animate subject. It seems to be at least marginally acceptable with an inanimate subject when the situation is not static, but it is totally bad if the situation is static and the subject is inanimate.

I should underscore the fact that this static/non-static distinction is different from the state/action distinction presented in the Chapter 7. Sentences like Sentence 7 do not pass the tests for states given in that chapter. For example, they can occur in the progressive, as in Sentence 13:

(13) Marvin was standing in the playroom.

Thus, there seem to be two different distinctions involved. If they are conflated, it is very difficult to account for some of the data.

Both of these distinctions are tangential to the issues discussed in this thesis. I have brought them up for several reasons. First, readers already familiar with the four-way verb classification in linguistics may have been surprised not to see some of these properties incorporated into the tests for different verb classes. Secondly, it is easy to make the mistake of incorporate either agentiveness or staticness into the definition of the state/action distinction, because the properties do tend to co-occur statistically. Finally, making these other distinctions explicit may help clarify the situation for readers attempting to extend this research, particularly those who may have little or no background in linguistics.

Appendix D: Coercion in natural language data

One potential source of confusion in analyzing verb class data is the freedom with which verbs that normally represent one type of situation can be reinterpreted as representing another type of situation. A simple metaphor for understanding these phenomena is that there are a number of *coercion* rules, like type coercions in programming languages, that specify how to change one type of situation into another. These are typically brought into play when the standard interpretation cannot result in a well-formed semantic structure. For example, when verbs representing actions appear in the simple present tense, they are regularly interpreted as refering to states describing habitual properties of the subject, because states can occur in the simple present and actions cannot.

Alternatively, one can think of coercions as part of a more general pattern of operations that change a constituent of one type into a constituent of another type. For example, the result of adding a measure phrase to an activity is an accomplishment. Coercions are similar to this type of combination, except that there is no overt marking in the sentence to indicate that coercion rules have been applied. An advantage to grouping these two types of processes together is that some semantic changes that have no overt marking in English, such as inceptive uses of state or iterative uses of actions, are marked by special forms in other languages. In this appendix, I discuss some of the more common types of coercions and I also mention several interesting cases of how constituents combine.

I start by describing the rules for noun (and related constituents), because the phenomena are simpler for them than for verbs.[1] We saw in Section 6 that the determiner "a" can only occur with count nouns (those refering to objects) and that only count nouns can occur in the plural. When a noun that usually refers to a stuff occurs in one of these forms, it must somehow be coerced into a reference to an object. There are at least two standard ways of doing this, which are best illustrated by example, as in Sentences 1-2:

(1) There were three wines at the wine tasting.

(2) Steve came back from the bar with six beers.

In Sentence 1, the mass noun "wine" is re-interpreted as refering to a kind of wine. In Sentence 2, on the other hand, "beer" is re-interpreted as refering to some fixed (but unspecified) quantity of beer.

These two tactics for coercing descriptions of stuffs into descriptions of objects can also be done overtly, as in Sentences 3-4:

(3) There were three kinds of wine at the wine tasting.

(4) Steve came back from the bar with six mugs of beer.

Sentence 3 is almost equivalent in meaning to Sentence 1. Sentence 4 contains an overt measure phrase and, unlike Sentence 2, makes explicit what quantity of beer is intended. Sentence 4 would typically be used when the intended quantity can be infered from context.

In a similar way, count nouns can be re-interpreted as refering to stuffs when they occur in syntactic contexts where a mass noun is ordinarily required. For example:

[1] These facts have been relatively well known for some time. See, for example, Bach (1986) and Allan (1980), for summaries of recent research on the semantics of noun classes.

(5) After tasting it, I added more missionary to the stew.

Given a suitable context, almost any object can be imagined as broken down into the stuff that it is made of. I don't know of any way to re-express this shift in meaning with an overt marker. Plural marking also seems to convert count nouns into descriptions of stuffs (see Carlson 1977a,b), but with a slightly different meaning. For example, Sentence 6 is more likely to be used when the missionaries are added whole, whereas Sentence 5 suggests that they are ground up:

(6) After tasting it, I added more missionaries to the stew.

A pattern of data similar to these changes in noun class occurs also with verbs and verb phrases, but the possibilities are more complicated. Some common changes are:

- iteratives,
- habituals,
- states re-interpreted as activities, and
- inceptives (overt or implied).

In the rest of this appendix, I discuss each of these types of changes in meaning in turn.

In Chapter 7, Section 11, we saw that verbs describing accomplishments or state changes change into descriptions of activities when they have a plural or mass noun direct object. In these cases, the action is interpreted as iterated over time. Under certain contexts, non-activities can be interpreted as iterated over time even when no overt plural is present. For example, in Sentence 7 the bare verb describes an accomplishment that happens instantaneously and thus could not normally take the progressive. An iterative reading is, however, possible.

(7) Patrick was noticing the new couch for weeks after it appeared.

Another construction that seems to be distinct from iteration, but may be confused with it, involves habitual readings of verb phrases. Consider Sentences 8 and 9, given by Woisetschlaeger (1976):

(8) Sam drives a truck for the ABC Company.

(9) Sam is driving a truck for the ABC Company.

Sentence 9 makes an empirical observation about what Sam is up to at the moment. Sentence 8, on the other hand, makes a claim about Sam's station in life. While empirical observations may have led us to conclude Sentence 8, the sentence itself describes a theory of how the world works, not an observation. In fact, Sentence 8 can reasonably be uttered if Sam is not, at this moment, driving, or even if Sam is newly hired and has not yet been out on the road.

To account for such sentences, Woisetschlaeger (1976) and Goldsmith and Woisetschlaeger (1976) postulate a distinction between "structural" vs. "phenomenal" readings of sentences. Phenomenal readings describe what is happening in the world and are the default types of readings under most conditions. Structural readings describe situations that are habitual or typical. Sentences given structural readings behave like states, perhaps because they describe properties of the world or of objects in it. Thus, a simple way to force a structural reading in English is to put the sentence in the simple present, as in Sentence 8, because actions cannot occur in the simple present. Conversely, a phenomenal reading can be forced by putting the sentence in the progressive, as in Sentence 9, because states cannot occur in the progressive. Carlson (1977a,b) uses a similar analysis in his discussion of the meaning of bare plurals.

Most verbs in English refer primarily to phenomenal situations and only have structural readings when that is forced by the context. There are, however, exceptions. For example, the verb "own" describes a situation that is structural, perhaps because it is created entirely by human convention. This is illustrated by Sentences 10:

(10) Margaret owns two acoustic guitars.

(11) #Margaret is owning two acoustic guitars.

This verb is one of a restricted group of verbs that are traditionally classed as descriptions of states. Other examples include "love,"[2] "have," "know," and "believe." Most states involve the verb "to be," together with a noun, adjective, or prepositional phrase.

There are also types of situations where a structural reading is more common than a phenomenal one. For example, the location of large statues, mountains, rivers, and other such objects is typically viewed as a structural property of the world. Thus, to paraphrase an example from Dowty (1979), Sentence 12 is a relatively neutral description of geography, whereas Sentence 13 describes a flood:

(12) The Thames flows through the center of London.

(13) The Thames is flowing through the center of London.

The progressive forces a phenomenal reading of the sentence and thus creates a presupposition that the situation described is prone to change.

Under appropriate conditions, even the verb "to be" can be forced into a phenomenal reading. For example, Sentence 14 can be acceptable in theatrical contexts:

(14) Jody is being a plant.

[2] Though this word is acquiring a secondary sense in which it is phenomenal.

Perhaps this reading represents a re-interpretation of "be a plant" as an activity. Alternatively, perhaps "to be" is like "stand," in that it describes an activity but one that is structural with most arguments. This would account for the fact that "to be" can behave more like an activity than a state with certain adjectives, as in Sentences 15-16:

(15) Willie is being noisy.

(16) Willie is noisy.

Not only is Sentence 15 good, despite being in the progressive, but these two sentences seem to display a difference in meaning parallel to Sentences 8-9.

When a verb or verb phrase describes a situation that can persist over a prolonged interval, i.e. anything but a state change, verbs such as "stop" and "start" can be used to create new constituents that refer to the state change at which the action or state stops or starts. This is illustrated by Sentences 17-18:

(17) Norman stopped being a student.

(18) Anita started running along the river.

Occasionally, such an interpretation is possible without overt modification to the verb phrase, as in Sentence 19:

(19) Mike played squash at 3:00.

(20) Suddenly, the cat was on the table!

The adverb "suddenly" requires a change of state and would thus not make sense applied directly to a state. The prepositional phrase "at 3:00" requires a state or activity that can hold over a very short interval of time and thus would not normally make sense modifying an accomplishment that takes a prolonged period of time. Thus, in both cases, the situation is coerced into a state change. In such cases, it is the start of the state or action that is referred to.

In this appendix, we have seen a number of patterns of coercions and rules for combination of constituents. Without a clear understanding of these rules, linguistic data on temporal and spatial structures become obscured by apparent counterexamples.

Bibliography

Allen, James F. (1983) "Maintaining knowledge about temporal intervals," *Communications of the ACM* 26/11, pp. 832–843.

______ (1984) 'Towards a General Model of Action and Time," *Artificial Intelligence* 23/2, pp. 123–154.

Allen, James F. and Patrick J. Hayes (1985) "A Common-Sense Theory of Time," *International Joint Conference on Artificial Intelligence* 1985, pp. 528–531.

Allan, Keith (1977) "Classifiers," *Language* 53/2, pp. 285–311.

______ (1980) "Nouns and Countability," *Language* 56/3, pp. 541–567.

Anderson, Lloyd B. (1982) "The 'Perfect' as a Universal and as a Language-Specific Category," pp. 227–264 in Paul J. Hooper, ed., *Tense-Aspect: Between Semantics and Pragmatics,* Typological Studies in Language, vol. 1, John Benjamins Publishing Co., Amsterdam.

Argyle, Edward (1971) "Techniques for Edge Detection," *Proceedings of the IEEE* 59/2, pp. 285–286.

Asada, Haruo and J. Michael Brady (1986) "The Curvature Primal Sketch," *PAMI* 8/1, pp. 2–14.

Ayache, Nicholas and Bernard Faverjon (1987) "Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments," *International Journal of Computer Vision* 1/2, pp. 107–131.

Bach, Emmon (1986) "The Algebra of Events," *Linguistics and Philosophy* 9/1, pp. 5–16.

Bajcsy, Ruzena (1972) "Computer Identification of Textured Scenes," Ph.D. thesis, Stanford University, Computer Science Department, distributed by the Artificial Intelligence Laboratory as AIM–180.

Bajcsy, Ruzena (1973) "Computer Identification of Visual Surfaces," *Computer Graphics and Image Processing* 2/2, pp. 118–130

Baker, Henry Harlyn (1982) "Depth from Edge and Intensity Based Stereo," Ph.D. thesis, Stanford University, Computer Science Department, distributed by the Artificial Intelligence Laboratory as AIM–347.

Baker, Henry Harlyn and Thomas O. Binford (1981) "Depth from Edge and Intensity Based Stereo," *International Joint Conference on Artificial Intelligence* 1981, pp. 631–636.

Ballard, Dana H. (1981) "Generalizing the Hough Transform to Detect Arbitrary Shapes" *Pattern Recognition*, 13/2, 111–122, reprinted in Fischler and Firschein (1987).

Ballard, Dana H. and Christopher M. Brown (1982) *Computer Vision,* Prentice-Hall, Englewood Cliffs, NJ.

Barnard, Stephen T. (1986) "A Stochastic Approach to Stereo Vision," *Proceedings of the National Conference on Artificial Intelligence* 1986, pp. 676–680, reprinted in Fischler and Firschein (1987).

Barnard, Stephen T. and Martin A. Fischler (1982) "Computational Stereo," *Computing Surveys* 14/4, pp. 553–572.

Barnard, Stephen T. and William B. Thompson (1980) "Disparity Analysis of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2/4, pp. 333–340.

Baumgart, Bruce G. (1972) "Winged Edges Polyhedron Representation," Artificial Intelligence Project, Computer Science Department, Stanford University, Memo AIM–179 = STAN–CS–320.

Bennett, Michael and Barbara Partee (1978) "Toward the Logic of Tense and Aspect in English," distributed by the Indiana University Linguistics Club.

van Benthem, J.F.A.K. (1983) *The Logic of Time*, D. Reidel, Dordrecht.

Bergholm, Fredrik (1987) "Edge Focusing," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9/6, pp. 726–741.

Berzins, Valdis (1984) "Accuracy of Laplacian Edge Detectors," *Computer Vision, Graphics, and Image Processing* 27/2, pp. 195–210.

Besl, Paul J. and Ramesh C. Jain (1988) "Segmentation through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10/2, pp. 167–192.

Binford, Thomas O. (1981) "Inferring Surfaces from Images," *Artificial Intelligence* 17, reprinted as J. M. Brady, ed., *Computer Vision,* North-Holland, Amsterdam, pp. 205–244.

Blake, Andrew (1983) "Parallel Computation in Low-level Vision," Ph.D. thesis, University of Edinburgh.

Blum, Harry (1973) "Biological Shape and Visual Science (Part I)," *Journal of Theoretical Biology* 38, pp. 205–287.

Blum, Harry and Roger N. Nagel (1978) "Shape Description using Weighted Symmetric Axis Features," *Pattern Recognition* 10, pp. 167–180.

Boie, Robert A., Ingemar J. Cox, Pavel Rehak (1986) "On Optimum Edge Recognition using Matched Filters," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1986, pp. 100–108.

Bolles, Robert C., H. Harlyn Baker, and David H. Marimont (1987) "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion," *International Journal of Computer Vision* 1/1, pp. 7–55.

Bovik, Alan C., Marianna Clark, and Wilson S. Geisler (1987) "Computational Texture Analysis using Localized Spatial Filtering," *Proceedings of the IEEE Computer Society Workshop on Computer Vision* 1987, pp. 201–206.

Boyer, K. L. and A. C. Kak (1988) "Structural Stereopsis for 3-D Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10/2, pp. 144–166.

Brady, J. Michael and Haruo Asada (1984) "Smoothed Local Symmetries and Their Implementation," *International Journal of Robotics Research* 3/3, 36–61.

Brady, Michael, Jean Ponce, Alan Yuille, and Haruo Asada (1985) "Describing Surfaces," *Computer Vision, Graphics, and Image Processing* 32/1, pp. 1–28.

Brooks, M. J. (1978) "Rationalizing Edge Detectors," *Computer Graphics and Image Processing* 8/2, pp. 277–285.

Brooks, Rodney A. (1981) "Symbolic Reasoning Among 3-D Models and 2-D images," *Artificial Intelligence* 17, reprinted as J. M. Brady, ed., *Computer Vision,* North-Holland, Amsterdam, pp. 285–348.

Brooks, Rodney A. and Tomás Lozano-Pérez (1985) "A Subdivision Algorithm in Configuration Space for Findpath with Rotation," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol SMC–15/2, pp. 224–233.

Bruce, Bertram C. (1972) "A Model for Temporal References and its Application in a Question Answering Program," *Artificial Intelligence* 3/1, pp. 1–25.

Bülthoff, Heinrich H. and Hanspeter A. Mallot (1987) "Interaction of Different Modules in Depth Perception," *Proceedings of the International Conference on Computer Vision* 1987, pp. 295–305.

Burt, Peter and Bela Julesz (1980a) "Modifications of the Classical Notion of Panum's Fusional Area," *Perception* 9, pp. 671–682.

——— (1980b) "A Disparity Gradient Limit for Binocular Fusion," *Science* 208, pp. 615–617.

Buxton, B.F. and H. Buxton (1984) "Computation of Optic Flow from the Motion of Edge Features in Image Sequences," *Image and Vision Computing* 2/2, pp. 59–75.

Callahan, James and Richard Weiss (1985) "A Model for Describing Surface Shape," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1985, pp. 240–245.

Canny, John F. (1983) "Finding Edges and Lines in Images," M.S. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–720.

______ (1986) "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8/6, pp. 679–698, reprinted in Fischler and Firschein (1987).

Carlson, Gregory N. (1977a) "Reference to Kinds in English," Ph.D. thesis, Department of Linguistics, University of Massachusetts, Amherst, destributed by the Graduate Linguistic Student Association of that department.

______ (1977b) "A Unified Analysis of the English Bare Plural," *Linguistics and Philosophy* 1, pp. 413–457.

Cavanagh, Patrick (1987) "Reconstructing the Third Dimension: Interactions between Color, Texture, Motion, Binocular Disparity, and Shape," *Computer Vision, Graphics, and Image Processing* 37/2, pp. 171–195.

Chen, Lin (1985) "Topological Structure in the Perception of Apparent Motion," *Perception* 14/2, pp. 197–208.

Chen, J. S. and G. Meioni (1987) "Detection, Localization and Estimation of Edges,' *Proceedings of the IEEE Computer Society Workshop on Computer Vision* 1987, pp. 215–217.

Clark, James J. (1986) "Authenticating Edges Produced by Zero Crossing Algorithms," unpublished manuscript.

Connell, Jonathan H. (1985) "Learning Shape Descriptions," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, TR–853.

Connell, Jonathan H. and J. Michael Brady (1987) "Generating and Generalizing Models of Visual Objects," *Artificial Intelligence* 31/2, pp. 159–183.

Comrie, Bernard, *Aspect*, Cambridge University Press, Cambridge, 1976.

______ (1985) *Tense*, Cambridge University Press, Cambridge.

Davis, Ernest (1986) *Representing and Acquiring Geographic Knowledge,* Research Notes in Artificial Intelligence, Morgan Kaufmann, Los Altos, CA.

——— (1984a) "An Ontology of Physical Action," Technical Report #123, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, NY.

——— (1984b) "Shape and Function of Solid Objects: Some Examples," Technical Report #137, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, NY.

Davis, Larry (1982) "Hierarchical Generalized Hough Transforms and Line-Segment Based Generalized Hough Transforms," *Pattern Recognition*, 15, 277–285.

Deriche, Rachid (1987) "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector," *International Journal of Computer Vision* 1/2, pp. 167–187.

Donald, Bruce R. (1984) "Motion Planning with Six Degrees of Freedom," M.S. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–791.

——— (1987a) "A Search Algorithm for Motion Planning with Six Degrees of Freedom," *Artificial Intelligence* 31/3, pp. 295–353.

——— (1987b) "Error Detection and Recovery for Robot Motion Planning with Uncertainty," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–982.

Dowty, David R. (1979) *Word Meaning and Montague Grammar*, D. Reidel, Dordrecht.

——— (1986) "The Effects of Aspectual Class on the Temporal Structure of Discourse: Semantics or Pragmatics?" *Linguistics and Philosophy* 9/1, pp. 37–61.

Drumheller, M. and T. Poggio (1986) "On Parallel Stereo," *Proceedings of the IEEE Conference on Robotics and Automation* 1986, pp. 1439–1448.

Duwaer, A.L. and G. van den Brink (1981) "Diplopia Thresholds and the Initiation of Vergence Eye-Movements," *Vision Research* 21, pp. 1727–1737.

Erdmann, Michael A. (1984) "On Motion Planning with Uncertainty," M.S. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed bythe Artificial Intelligence Laboratory as TR–810.

——— (1986) "Using Backprojections for Fine Motion Planning with Uncertainty," *International Journal of Robotics Research* 5/1, pp. 19–45.

Erdmann, Michael A. and Tomás Lozano-Pérez (1987) "On Multiple Moving Objects," *Algorithmica* 2, pp. 477–521.

Faltings, Boi (1987) "Qualitative Kinematics in Mechanisms," *International Joint Conference on Artificial Intelligence* 1987, pp. 436–442.

Fischler, Martin A. and Oscar Firschein (1987) *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms,* Morgan Kaufmann, Los Altos, CA.

Fleck, Margaret M. (1985) "Local Rotational Symmetries," M.S. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–852.

______ (1986) "Local Rotational Symmetries," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1986, pp. 332–337.

______ (1988) "Representing Space for Practical Reasoning," *Image and Vision Computing* 6/2, pp. 75–86.

Fram, Jerry R. and Edward S. Deutsch (1975) "On the Quantitative Evaluation of Edge Detection Schemes and their Comparison with Human Performance," *IEEE Transactions on Computers* C-24/6, pp. 616–628.

Forbus, Kenneth D. (1984) "Qualitative Process Theory," pp. 85–168 in *Artificial Intelligence* 24, reprinted as Daniel G. Bobrow, ed., *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge, Massachusetts.

______ (1986) "Interpreting Measurements of Physical Systems," *Proceedings of the National Conference on Artificial Intelligence* 1986, pp. 113–117.

______ (1987) "The Logic of Occurrence," *International Joint Conference on Artificial Intelligence* 1987, pp. 409–415.

Funt, Brian V. (1980) "Problem-solving with Diagramatic Representations," *Artificial Intelligence* 13, pp. 201–230, reprinted in Fischler and Firschein (1987).

Gamble, Ed and Tomaso Poggio (1987) "Visual Integration and Detection of Discontinuities: The Key Role of Intensity Edges," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 970.

Geman, Stuart and Donald Geman (1984) "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, pp. 721–741, reprinted in Fischler and Firschein (1987).

Gennery, Donald B. (1977) "A Stereo Vision System for an Autonomous Vehicle," *International Joint Conference on Artificial Intelligence* 1977, pp. 576–582.

Gennert, Michael A. (1986) "Detecting Half-Edges and Vertices in Images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1986, pp. 552–557.

Gillett, Walter Eisner (1988) "Isses in Parallel Stereo Matching," M.S. thesis, Massachusetts Institute of Technology, Department of Brain and Cognitive Sciences.

Goldsmith, John and Erich Woisetschlaeger (1979) "The Logic of The Progressive Aspect," distributed by the Indiana University Linguistics Club.

Grimson, W. Eric L. (1981a) *From Images to Surfaces: A Computation Study of the Human Early Visual System,* MIT Press, Cambridge, Massachusetts.

——— (1981b) "A Computer Implementation of a Theory of Human Stereo Vision," *Philosophical Transactions of the Royal Society of London B* 292, pp. 217–253.

——— (1985) "Computational Experiments with a Feature Based Stereo Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7/1, pp. 17–34.

Grimson, W. Eric L. and Theo Pavlidis (1985) "Discontinuity Dectection for Visual Surface Reconstruction," *Computer Vision, Graphics, and Image Processing* 30, pp. 316–330.

Grünbaum, Branko and G.C. Shephard (1987) *Tilings and Patterns*, W.H. Freeman and Co., New York.

Hamblin, C.L. (1972) "Instants and Intervals," pp. 324–331 in J.T. Fraser, F.C. Haber, and G.H. Müller, eds., *The Study of Time*, Proceedings of the First Conference of the International Society for the Study of Time, Springer-Verlag, Berlin.

Hannah, Marsha Jo (1980) "Bootstrap Stereo," *Proceedings of the DARPA Image Understanding Workshop* 1980, pp. 201–208.

Haralick, Robert M. (1980) "Edge and Region Analysis for Digital Image Data," *Computer Graphics and Image Processing* 12/1, pp. 60–73.

——— (1984) "Digital Step Edges from Zero Crossings of Second Directional Differences," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6/1, pp. 58–68, reprinted in Fischler and Firschein (1987).

Haralick, Robert M., Layne T. Watson, and Thomas J. Laffey, "The Topographic Primal Sketch," *International Journal of Robotics Research* 2 (1983) 50–72.

Hayes, Patrick J. (1985a) "Naive Physics 1: Ontology for Liquids," pp. 71–107 in Jerry R. Hobbs and Robert C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex Publishing Corporation, Norwood, NJ.

______ (1985b) "The Second Naive Physics Manifesto," pp. 1–36 in Jerry R. Hobbs and Robert C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex Publishing Corporation, Norwood, NJ.

Heeger, David J. (1987) "Optical Flow from Spatiotemporal Filters," *Proceedings of the International Conference on Computer Vision* 1987, pp. 181–190.

Hildreth, Ellen (1983) "The Detection of Intensity Changes by Computer and Biological Vision System," *Computer Vision, Graphics, and Image Processing* 22, 1–27.

______ (1984) *The Measurement of Visual Motion,* MIT Press, Cambridge, Massachusetts.

Hinrichs, Erhard (1986) "Temporal Anaphora in Discourses of English," *Linguistics and Philosophy* 9/1, pp. 63–82.

Hoff, William and Narendra Ahuja (1987) "Extracting Surfaces from Stereo Images: An Integrated Approach," *Proceedings of the International Conference on Computer Vision* 1987, pp. 284–294.

Horn, Berthold K. P. and Brian G. Schunck (1981) "Determining Optical Flow," *Artificial Intelligence* 17, reprinted as J. M. Brady, ed., *Computer Vision,* North-Holland, Amsterdam, pp. 185–203.

Horn, Berthold K. P. and R. J. Woodham (1978) "Destriping Satellite Images," *Proceedings of the DARPA Image Understanding Workshop* 1978, pp. 56–63.

Huang, Kai, David Lee, and Theo Pavlidis (1987) "Edge Detection through Two-Dimensional Regularization," *Proceedings of the IEEE Computer Society Workshop on Computer Vision* 1987, pp. 225–227.

Hueckel, Manfred H. (1971) "An Operator which Localizes Edges in Digitized Pictures," *Journal of the Association for Computing Machinery* 18/1, pp. 113–125.

______ (1973) "A Local Visual Operator which Recognizes Edges and Lines," *Journal of the Association for Computing Machinery* 20/4, pp. 634–647.

Huertas, Andres and Gerard Medioni (1986) "Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8/5, pp. 651–664.

Huttenlocher, Daniel P. (1988) "Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

Huttenlocher, Daniel P. and Shimon Ullman (1987) "Recognizing Rigid Objects by Aligning them with an Image," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 937.

——— (1988) "Recognizing Solid Objects by Alignment," *Proceedings of the DARPA Image Understanding Workshop* 1988, pp. 1114-1124.

Jackendoff, Ray (1983) *Semantics and Cognition,* MIT Press, Cambridge, Massachusetts.

Johnson, Marion R. (1981) "A Unified Temporal Theory of Tense and Aspect," in *Syntax and Semantics, Volume 14: Tense and Aspect*, Philip J. Tedeschi and Annie Zaenen, eds., Academic Press, New York, pp. 145–175.

Julesz, B. and J. R. Bergen (1983) "Textons, The Fundamental Elements in Preattentive Vision and Perception of Textures," *Bell System Technical Journal* 62/6 Part II, pp. 1619–1645, reprinted in Fischler and Firschein (1987).

Kamp, Hans (1979) "Events, Instants and Temporal Reference," pp. 376–417 in R. Bäuerle, U. Egli, and A. von Stechow, eds., *Semantics from Different Points of View,* Springer-Verlag, Berlin.

Kass, Michael (1983) "A Computational Framework for the Visual Correspondence Problem," *International Joint Conference on Artificial Intelligence* 1983, pp. 1043–1045.

——— (1983) "Computing Visual Correspondence," *Proceedings of the DARPA Image Understanding Workshop* 1983, pp. 54–60.

Kass, Michael and Andrew Witkin (1985) "Analyzing Oriented Patterns," *International Joint Conference on Artificial Intelligence* 1985, pp. 944–952, reprinted in Fischler and Firschein (1987).

——— (1987) "Analyzing Oriented Patterns," *Computer Vision, Graphics, and Image Processing* 37/3, pp. 362–385.

Kjell, Bradley P. and Charles R. Dyer (1985) "Edge Separation and Orientation Texture Measures," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1985, pp. 306–311.

de Kleer, Johan and John Seely Brown (1984) "A Qualitative Physics Based on Confluences," pp. 7–83 in *Artificial Intelligence* 24, reprinted as Daniel G. Bobrow, ed., *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge, Massachusetts.

Koenderink, J. J. and A. J. van Doorn (1976) "The Singularities of the Visual Mapping," *Biological Cybernetics* 24, pp. 51–59.

Koenderink, J.J. and A.J. van Doorn (1982) "The Shape of Smooth Objects and the Way Contours End," *Perception* 11, pp. 129–137.

Krol, Jodi D. and Wim A. van de Grind (1980) "The Double-Nail Illusion: Experiments on Binocular Vision with Nails, Needles, and Pins," *Perception* 9, pp. 651-669.

Kuipers, Benjamin (1984) "Commonsense Reasoning about Causality: Deriving Behavior from Structure," pp. 169–203 in *Artificial Intelligence* 24, reprinted as Daniel G. Bobrow, ed., *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge, Massachusetts.

——— (1986) "Qualitative Simulation," *Artificial Intelligence* 29, pp. 289–338.

Laws, Kenneth I. (1979) "Texture Energy Measures," *Proceedings of the DARPA Image Understanding Workshop* 1979, pp. 47–51.

Lawton, Daryl (1983) "Processing Translational Motion Sequences," *Computer Vision, Graphics, and Image Processing* 22, pp. 116–144.

Leclerc, Yvan (1985) "Capturing the Local Structure of Image Discontinuities in Two Dimensions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1985, pp. 34–38, reprinted in Fischler and Firschein (1987).

Lee, Chung-Nim and Azriel Rosenfeld (1986) "Connectivity Issues in 2D and 3D Images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1986, 278–285.

Lee, David, Theo Pavlidis, and Kai Huang (1988) "Edge Detection through Residual Analysis," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1988, pp. 215–222.

Levine, Ira N. (1983) *Physical Chemistry,* McGraw-Hill, New York.

Levine, Martin D., Douglas A. O'Handley, and Gary M. Yagi (1973) "Computer Determination of Depth Maps," *Computer Graphics and Image Processing* 2/2, pp. 131–150.

Li, Charles N., Sandra A. Thompson, and R. McMillan Thompson (1982) "The Discourse Motivation for the Perfect Aspect: The Mandarin Particle *le*," pp. 19–44 in *Tense-Aspect: Between Semantics and Pragmatics,* Typological Studies in Language, vol. 1, John Benjamins Publishing Co., Amsterdam.

Little, James, Heinrich Bültoff, and Tomaso Poggio (1987) "Parallel Optical Flow Computation," *Proceedings of the DARPA Image Understanding Workshop* 1987, pp. 915–920.

Lozano-Pérez, Tomás (1981) "Automatic Planning of Manipulator Transfer Movements," *IEEE Transactions on Systems, Man, and Cybernetics* 11/10, pp. 681–698.

——— (1985) "Motion Planning for Simple Robot Manipulators," *Proceedings of the International Symposium on Robotics Research* 1985, 231–238.

Lozano-Pérez, Tomás, M.T. Mason, and R.H. Taylor (1984) "Automatic Synthesis of Fine-Motion Strategies for Robots," *International Journal of Robotics Research* 3/1, pp. 3–24.

McDermott, Drew (1982) "A Temporal Logic for Reasoning about Processes and Plans," *Cognitive Science* 6, pp. 101–155.

McDermott, Drew and Ernest Davis (1984) "Planning Routes through Uncertain Territory," *Artificial Intelligence* 22, pp. 107–156.

Macleod, I. D. G. (1972) "Comments on 'Techniques for Edge Detection,' " *Proceedings of the IEEE* 60/3, p. 344.

Marr, David (1982) *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information,* W.H. Freeman and Co, San Francisco, CA.

Marr, David and Ellen Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London B* 207 (1980) 187–217.

Marr, David and Tomaso Poggio (1976) "Cooperative Computation of Stereo Disparity," *Science* 194, pp. 283–287.

——— (1979) "A Computational Theory of Human Stereo Vision," *Proceedings of the Royal Society of London B* 204, pp. 301–328.

Marr, David, Tomaso Poggio, and Ellen Hildreth, "Smallest Channel in Early Human Vision," *Journal of the Optical Society of America* 70 (1980) 868–870.

Marroquin, J. L. (1984) "Surface Reconstruction Preserving Discontinuities," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 792.

Mason, Matthew T. (1984) "Automatic Planning of Fine Motions: Correctness and Completeness," *International Conference on Robotics Research* 1984, pp. 492–503.

Massey, William S. (1980) *Singular Homology Theory,* Graduate Texts in Mathematics 70, Springer-Verlag, New York.

Matsuyama, Takashi, Shu-Ichi Miura, and Makoto Nagao (1983) "Structural Analysis of Natural Textures by Fourier Transformation," *Computer Vision, Graphics, and Image Processing* 24, pp. 347–362.

Mayhew, John E.W. and John P. Frisby (1980) "The Computation of Binocular Edges," *Perception* 9, pp. 69–86.

——— (1981) "Psychophysical and Computational Studies towards a Theory of Human Stereopsis," *Artificial Intelligence* 17, reprinted as J. M. Brady, ed., *Computer Vision,* North-Holland, Amsterdam, pp. 349–385

Medioni, Gerard and Ramakant Nevatia (1985) "Segment-Based Stereo Matching," *Computer Vision, Graphics, and Image Processing* 31, pp. 2–18.

Minsky, Marvin (1975) "A Framework for Representing Knowledge," pp. 211–277 in Patrick Winston, ed., *The Psychology of Computer Vision,* Mc-Graw-Hill, New York.

Mohan, Rakesh, Gerard Medioni, and Ramakant Nevatia (1987) "Stereo Error Detection, Correction and Evaluation," *Proceedings of the International Conference on Computer Vision* 1987, pp. 315–324.

Moravec, Hans P. (1977) "Towards Automatic Visual Obstacle Avoidance," *International Joint Conference on Artificial Intelligence* 1977, p. 584 [sic: only one page].

——— (1981) "Rover Visual Obstacle Avoidance," *International Joint Conference on Artificial Intelligence* 1981, pp. 785–790.

Mori, Ken-ichi, Masatsugu Kidode, and Haruo Asada (1973) *Computer Graphics and Image Processing* 2/3–4, pp. 393–401.

Mowforth, Peter, John E.W. Mayhew, and John P. Frisby (1981) "Vergence Eye Movements Made in Response to Spatial-Frequency-Filtered Random-Dot Stereograms," *Perception* 10, pp. 299–304.

Mourelatos, Alexander P. D., "Events, Processes, and States," in *Syntax and Semantics, Volume 14: Tense and Aspect*, Philip J. Tedeschi and Annie Zaenen, eds., Academic Press, New York, 1981, pp. 191–212.

Munkres, James R. (1975) *Topology: A First Course*, Prentice-Hall, Englewood Cliffs, NJ.

——— (1984) *Elements of Algebraic Topology,* Addison-Wesley, Menlo Park, CA.

Mutch, Kathleen M. and William B. Thompson (1985) "Analysis of Accretion and Deletion at Boundaries in Dynamic Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7/2, pp. 133–138.

Nalwa, Vishvjit S. and Thomas O. Binford (1986) "On Detecting Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8/6, pp. 699–714.

Nevatia, Ramakant (1976) "Depth Measurement by Motion Stereo," *Computer Graphics and Image Processing* 5/2, pp. 203–214.

Nevatia, Ramakant and K. Ramesh Babu (1980) "Linear Feature Extraction and Description," *Computer Graphics and Image Processing* 13/3, pp. 257–269.

Nielsen, K. R. K. and Tomaso Poggio (1983) "Vertical Image Registration in Stereopsis," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 743.

Nishihara, H. K. (1984) "Practical Real-Time Imaging Stereo Matcher," *Optical Engineering* 23/5, pp. 536–545.

Noble, J. Alison (1987) "Finding Two Dimensional Image Structure," *Proceedings of the IEEE Computer Society Workshop on Computer Vision* 1987, pp. 222–224.

Ohta, Yuichi and Takeo Kanade (1983) "Stereo by Intra- and Inter-scanline Search using Dynamic Programming," Department of Computer Science, Carnegie-Mellon University, technical report CMU–CS–83–162.

______ "Stereo by Intra- and Inter-scanline Search using Dynamic Programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7/2, pp. 139–154.

Parvin, B. and G. Medioni (1987) "Adaptive Multiscale Feature Extraction from Range Data," *Proceedings of the IEEE Computer Society Workshop on Computer Vision* 1987, pp. 23–28.

Patil, Ramesh S. (1981) "Causal Representation of Patient Illness for Electrolyte and Acid-Base Diagnosis," Ph.D. thesis, Dept. of Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Laboratory for Computer Science as TR–267.

Pavlidis, Theo (1977) *Structural Pattern Recognition*, Springer-Verlag, Berlin.

Pearson, Don E. and John A. Robinson (1985) "Visual Communication at Very Low Data Rates," *Proceedings of the IEEE* 73/4, pp. 795–812.

Persoon, Eric (1976) "An New Edge Detection Algorithm and its Applications in Picture Processing," *Computer Graphics and Image Processing* 5/4, pp. 425–446.

Pollard, Stephen B., John E.W. Mayhew, and John P. Frisby (1985) "PMF: A Stereo Correspondence Algorithm using a Disparity Gradient Limit," *Perception* 14, pp. 449–470.

Ponce, Jean and J. Michael Brady (1987) "Toward a Surface Primal Sketch," pp. 195–240 in Takeo Kanade, ed., *Three-Dimensional Visual Systems*, Kluwer Academic Publishers, Dordrecht.

Poggio, Gian F. and Tomaso Poggio (1984) "The Analysis of Stereopsis," *Annual Review of Neuroscience* 7, pp. 379–412.

Poggio, Tomaso et. al. (1988) "The MIT Vision Machine," *Proceedings of the DARPA Image Understanding Workshop* 1988, pp. 177-198.

Poston, Tim (1971) "Fuzzy Geometry", Ph.D. thesis, Univ. of Warwick.

Pratt, William K. (1978) *Digital Image Processing,* John Wiley and Sons, New York.

Prazdny, K. (1985) "The Detection of Binocular Disparities," *Biological Cybernetics* 52, pp. 93–99, reprinted in Fischler and Firschein (1987).

Quam, Lynn H. (1984) "Hierarchical Warp Stereo," *Proceedings of the DARPA Image Understanding Workshop* 1984, pp. 149–155, reprinted in Fischler and Firschein (1987).

Reichenbach, Hans, *Elements of Symbolic Logic,* The Free Press, New York, 1947.

Richards, Whitman and Donald D. Hoffman (1985) "Condon Constraints on Closed 2D Shapes," *Computer Vision, Graphics, and Image Processing* 31/2, pp. 156–177.

Rom, Hillel and Shmuel Peleg (1988) "Image Representation using Voronoi Tessellation: Adaptive and Secure," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1988, pp. 282-285.

Rosenfeld, Azriel (1970) "A Nonlinear Edge Detection Technique," *Proceedings of the IEEE* 58/5, pp. 814–816.

Ryle, Gilbert, *The Concept of Mind*, Barnes and Noble, London, 1949.

Schunck, Brian G. (1987) "Edge Detection with Gaussian Filters at Multiple Scales," *Proceedings of the IEEE Computer Society Workshop on Computer Vision* 1987, pp. 208–210.

Scott, Guy L. (1986) "Local and Global Interpretation of Moving Images," D.Phil thesis, University of Sussex.

Sher, David B. (1987a) "A Probabilistic Approach to Low-Level Vision," Ph.D. thesis, Dept. of Computer Science, University of Rochester, distributed by that department as TR–232.

——— (1987b) "Tunable Facet Model Likelihood Generators for Boundary Pixel Detection," *Proceedings of the IEEE Computer Society Workshop on Computer Vision* 1987, pp. 35–40.

Shoham, Yoav (1987a) "Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence," Ph.D. thesis, Yale University, Department of Computer Science, 1987, distributed by that department as technical report YALEU/CSD/RR #507.

——— (1987b) "Temporal Logics in AI: Semantical and Ontological Considerations," *Artificial Intelligence* 33/1, pp. 89-104.

Simmons, Reid G. (1983) "Representing and Reasoning about Change in Geologic Interpretation," M.S. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–749.

——— (1986) " 'Commonsense' Arithmetic Reasoning," *Proceedings of the National Conference on Artificial Intelligence* 1986, pp. 118–124.

——— (1988) "Combining Associational and Causal Reasoning to Solve Interpretation and Planning Problems," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–1048.

Spacek, L. A. (1985) "The Detection of Contours and the Visual Motion," D.Phil thesis, University of Essex.

Taylor, Barry (1977) "Tense and Continuity," *Linguistics and Philosophy 1*, 199–220.

Tenny, Carol Lee (1987) "Grammaticalizing Aspect and Affectedness," Ph.D. thesis, Massachusetts Institute of Technology Department of Linguistics and Philosophy.

Thompson, William B., Kathleen M. Mutch, and Valdis A. Berzins (1985) "Dynamic Occlusion Analysis in Optical Flow Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7/4, pp. 374–383.

Tichý, Pavel (1985) "Do we need Interval Semantics?" *Linguistics and Philosophy* 8/2, pp. 263–282.

Torre, Vincent and Tomaso A. Poggio (1986) "On Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8/2, pp. 147–163.

Trivedi, Harit P. and Sheelagh A. Lloyd (1985) "The role of Disparity Gradient in Stereo Vision," *Perception* 14/6, p. 685-690.

Ullman, Shimon (1984) "Visual Routines," *Cognition* 18, pp. 97–106, reprinted in Fischler and Firschein (1987).

——— (1986) "An Approach to Object Recognition: Aligning Pictorial Descriptions," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo 931.

Ulupinar, Fatih and Gérard Medioni (1988) "Refining Edges Detected by a LoG Operator," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1988, pp. 202–207.

Voorhees, Harry (1987) "Finding Texture Boundaries in Images," M.S. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–968.

Voorhees, Harry and Tomaso Poggio (1987) "Detecting Textons and Texture Boundaries in Natural Images," *Proceedings of the International Conference on Computer Vision* 1987, 250–258.

Vendler, Zeno (1967) *Linguistics in Philosophy*, Cornell University Press, Ithaca NY.

Vilnrotter, Felicia M. (1981) "Structural Analysis of Natural Textures," Ph.D. thesis, University of Southern California, available as USCISG 100.

Vilnrotter, Felicia M., Ramakant Navatia, and Keith E. Price (1986) "Structural Analysis of Natural Textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8/1, pp. 76–89.

Watt, R. J. and M. J. Morgan (1983) "The Recognition and Representation of Edge Blur: Evidence for Spatial Primitives in Human Vision," *Vision Research* 23/12, pp. 1465–1477.

Watt, R. J. and M. J. Morgan (1984) "Spatial Filters and the Localization of Luminance Changes in Human Vision," *Vision Research* 24/10, pp. 1387–1397.

______ (1985) "A Theory of the Primitive Spatial Code in Human Vision," *Vision Research* 25/11, pp. 1661–1674.

Weld, Daniel D. (1986) "The Use of Aggregation in Causal Simulation," *Artificial Intelligence* 30/1, pp. 1–34.

______ (1988) "Theories of Comparative Analysis," Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, distributed by the Artificial Intelligence Laboratory as TR–1035.

Williams, Brian C. (1984) "Qualitative Analysis of MOS Circuits," pp. 281–346 in *Artificial Intelligence* 24, reprinted as Daniel G. Bobrow, ed., *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge, Massachusetts.

Witkin, Andrew P. (1983) "Scale-Space Filtering," *International Joint Conference on Artificial Intelligence* 1983, pp. 1019–1022, reprinted in Fischler and Firschein (1987).

Woisetschlaeger, Erich (1976) "A Semantic Theory of the English Auxiliary System," Ph.D. thesis, Department of Linguistics and Philosophy, Massachusetts Institute of Technology, distributed by the Indiana University Linguistics Club.

Yellott, John I., Jr., Brian A. Wandell, and Tom N. Cornsweet (1984) "The Beginnings of Visual Perception: the Retinal Image and its Initial Encoding," pp. 257–316 in John M. Brookhart, Vernon B. Mountcastle, and Ian Darian-Smith, eds., *Handbook of Physiology, Section 1: The Nervous System, Volume*

III. Sensory Processes, Part 1, American Philosophical Society, Bethesda, MD.

Yeshurun, Yehezhel and Eric L. Schwartz (1987) "Cepstral Filtering on a Columnar Image Architecture: A Fast Algorithm for Binocular Stereo Segmentation," Technical Report #286, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, NY

Young, Richard A. (1986) "Locating Industrial Parts with Subpixel Accuracies," *Proceedings of the Society of Photo-Optical Instrumentation Engineers* 728: *Optics, Illumination, and Image Sensing for Machine Vision,* pp. 2–9.

Zeeman, E. C. (1962) "The Topology of the Brain and Visual Perception," pp. 240–256 in M. K. Fort, Jr., ed., *Topology of 3-Manifolds and Related Topics,* Prentice-Hall, Englewood Cliffs, NJ.

Zucker, Stephen W. (1985) "Early Orientation Selection: Tangent Fields and the Dimensionality of their Support," *Computer Vision, Graphics, and Image Processing* 32/1, pp. 74–103, reprinted in Fischler and Firschein (1987).