THE EARLY DETECTION OF MOTION BOUNDARIES

by

Anselm Spoerri

Bachelor of Science Honors in Mathematics
Bedford College (University of London), England
1982

Submitted to the Department of Brain & Cognitive Sciences
in Partial Fulfillment of the Requirements of the Degree of

MASTER OF SCIENCE IN BRAIN AND COGNITIVE SCIENCES

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 24, 1991

Signature of the author .....
Anselm Spoerri
Dept. of Brain & Cognitive Sciences
January 24, 1991

Certified by .....
Shimon Ullman
Professor of Brain & Cognitive Sciences
Thesis Supervisor

Accepted by .....
Emilio Bizzi
Chairman
Dept. of Brain & Cognitive Sciences

THE EARLY DETECTION OF MOTION BOUNDARIES

by

Anselm Spoerri

Submitted to the Department of Brain & Cognitive Sciences
on January 24, 1991 in partial fulfillment of the requirements of
the Degree of Master of Science in Brain and Cognitive Sciences

ABSTRACT

This thesis addresses the problem of how to detect boundaries on the basis of motion information alone, and its solution is performed in two stages: (i) the local estimation of motion discontinuities and the computation of the visual flow field; (ii) the extraction of complete boundaries belonging to differently moving objects. For the first stage, three new methods are presented that can independently estimate motion boundaries: the *Bimodality Tests*, the *Bi-distribution Test*, and the *Dynamic Occlusion Method*. These methods can estimate motion boundaries in a scene containing several moving objects, without prior knowledge of their shapes or motions, and they require only local computations. The motion boundary estimators have been implemented on the Connection Machine, a large parallel network of simple, locally interconnected processors. Further, it is also shown that the visual flow field can be locally estimated as a by-product of the early estimation of motion boundaries, and a mathematical formulation is provided to show that the proposed computation of visual motion is well-posed. The second stage consists of applying and modifying the *Structural Saliency Method* by Sha'ashua & Ullman to extract complete and unique boundaries from the output of the first stage, which is often broadly defined and can contain gaps. Results are presented that show that the methods can successfully segment complex dynamic images composed of random-dot patterns or natural textures. It is also shown how the methods can be used in stereopsis and surface reconstruction.

Thesis Supervisor: Dr. Shimon Ullman
Title: Professor of Brain & Cognitive Sciences

The Early Detection of Motion Boundaries

The reader may perceive the outline of a dalmation on its morning walk, but most likely she or he will experience some difficulty because of the absence of distinct intensity edges along the dalmation's outline.

If, however, the reader were to see a motion sequence of the dalmation then she or he would immediately perceive its outline, even though the intensity information in the individual frames is ambiguous.

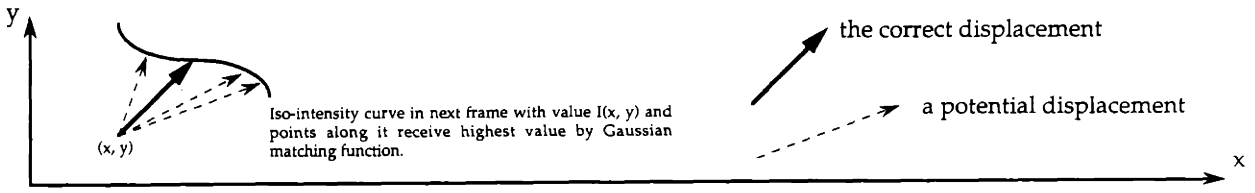
This thesis addresses the problem of how the outline of the dalmation, for example, can be computed based on motion information alone and without there being a sharp change in intensity along its outline.

- **Problem Statement**

  How to detect and group boundaries based on motion information alone and how to estimate visual motion early on ?

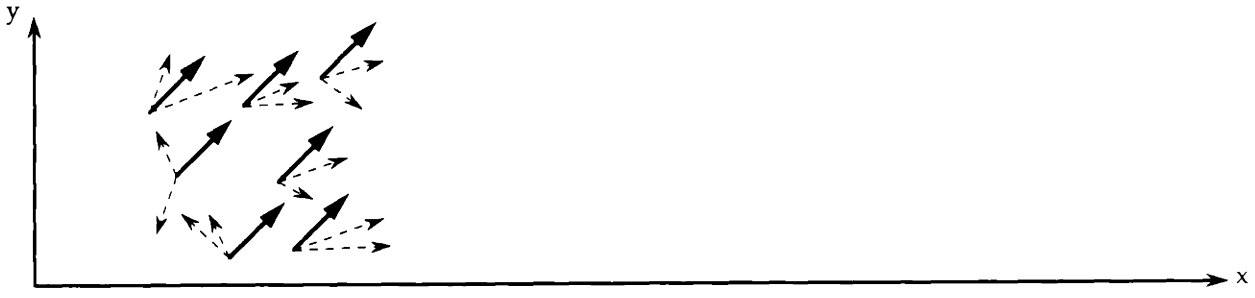
- **What can be computed early on ?** --> *Potential displacements*

  Observation & Assumption
  There is a great deal of ambiguity concerning the correct match, regardless of whether intensities or edge-tokens are used as matching primitives to compute the potential displacements. Intensity values remain roughly constant at corresponding points in subsequent frames, and we use a *Gaussian matching function*, which depends on the difference in intensity at the two points which define a potential displacement.



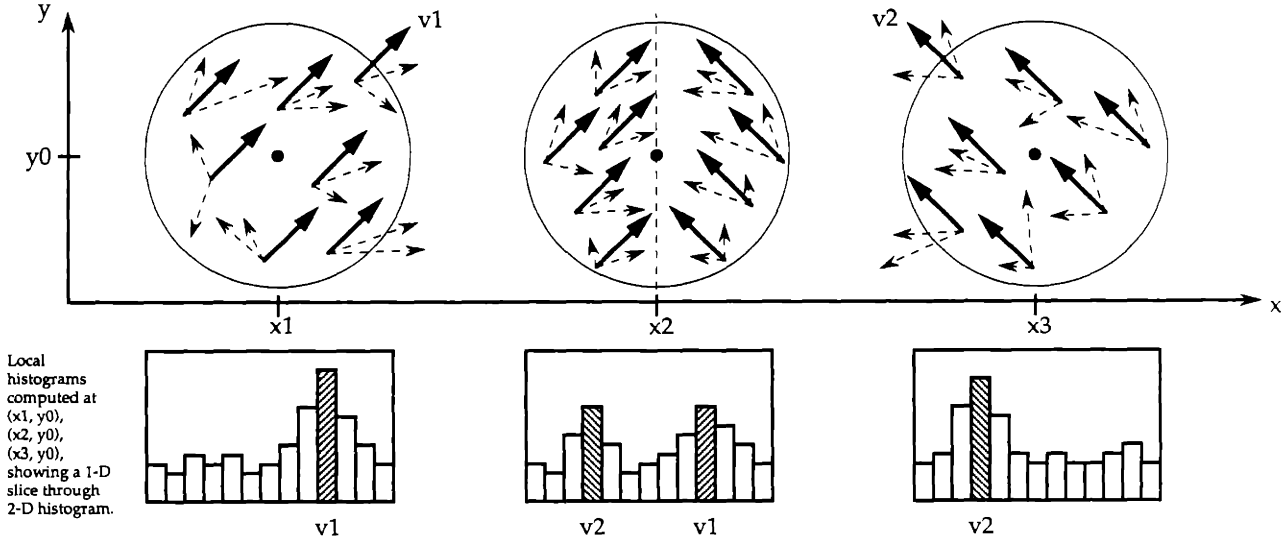
- **How to deal with the ambiguity of the potential displacements ?** --> *Use the fact that the potential displacements are unimodally distributed inside an object.*

  Observation & Assumption
  The image flow field can be approximated as locally constant. Hence, neighboring points will have a potential displacement in common and their potential displacements will cluster around a single point in a local two-dimensional histogram that collects the votes for the different possible motions.



- **How to detect motion boundaries ?** --> *Look for bimodal distributions of the potential displacements*

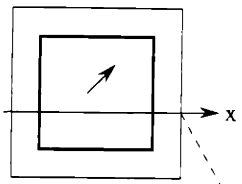
  Observation
  The potential displacements of points within a circle, whose center is in the vicinity of a motion boundary, will cluster around two different points in a local two-dimensional histogram that collects the votes for the different possible motions.

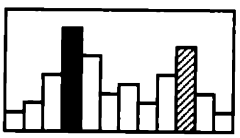


A motion boundary will cause the local histograms to be bimodal.

- **How to capture the occurring bimodality ?**

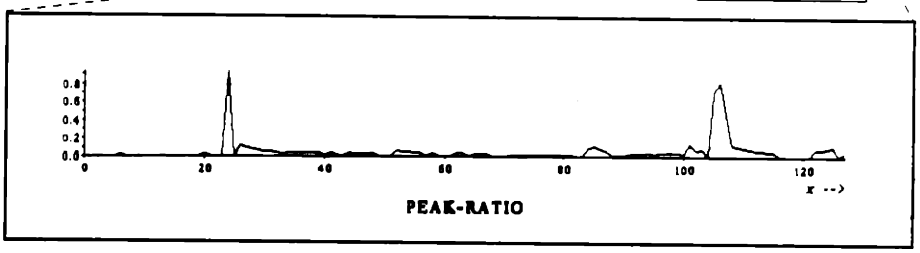
  Propose five measures that are sensitive to a motion boundary. The left column shows how they are defined and the right column shows their value along a scanline in a random-dot image containing a translating square.

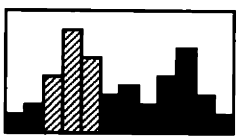




Peak-ratio

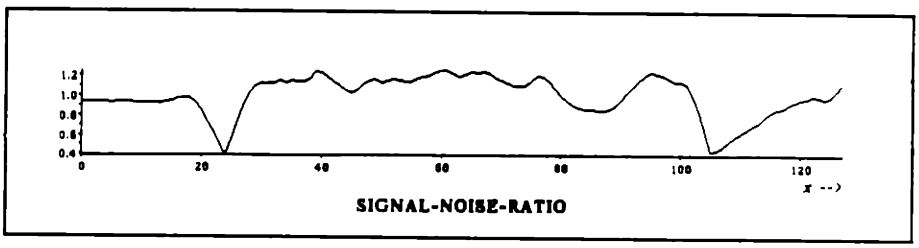
Ratio of the height of the second highest and of the highest peak in a local histogram.

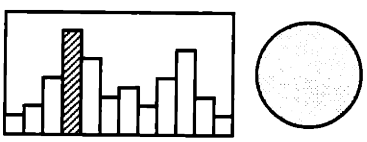




Signal-Noise-ratio

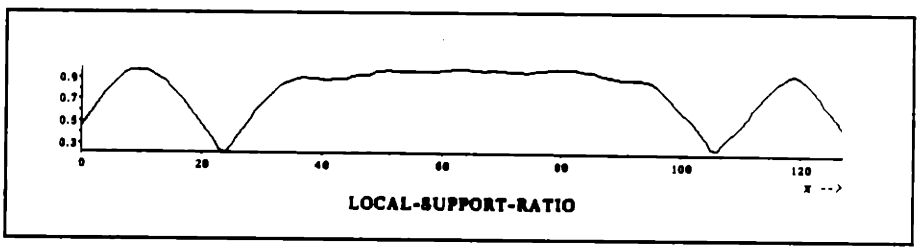
Ratio of the votes for the highest peak & its neighbors and of the votes for the remaining displacements.

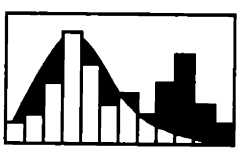




Local-Support-ratio

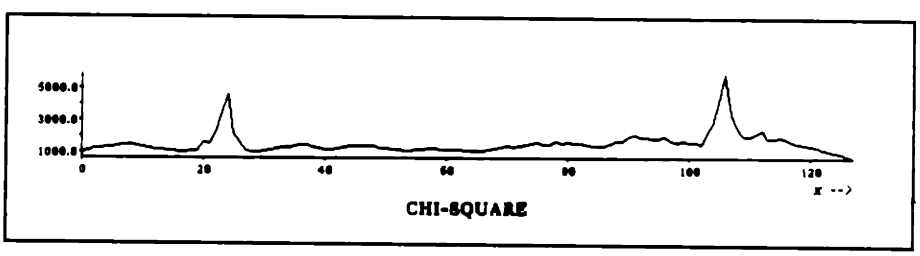
Ratio of the highest peak and the area of the circular histogram support.

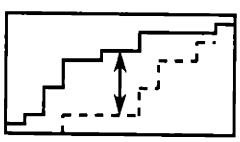




Chi-Square

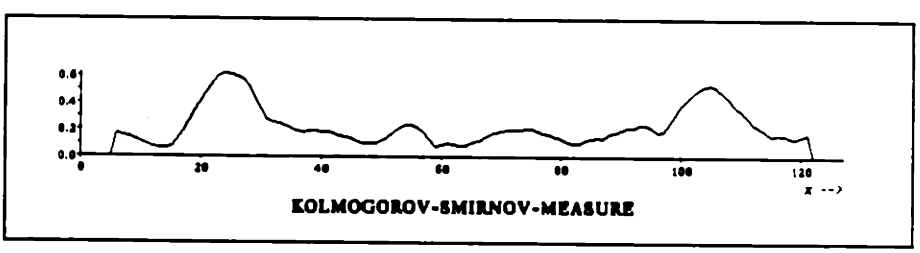
Measures how well a Gaussian distribution can be fitted to a local histogram.

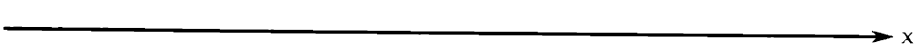




Kolmogorov-Smirnov

Measures the probability that two histograms have been created by the same population of motions.

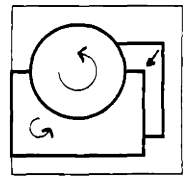




The proposed measures have a global extremum at a motion boundary.

• How to infer motion boundaries ?

The rows show the locally estimated boundaries for the case of a complex dynamic random-dot pattern that contains a rotating circle and rectangle and a translating square. The following three approaches can be used to infer the motion boundaries.



peak-ratio signal-noise-ratio local-support-ratio chi-square Kolmogorov-Smirnov

Thresholding

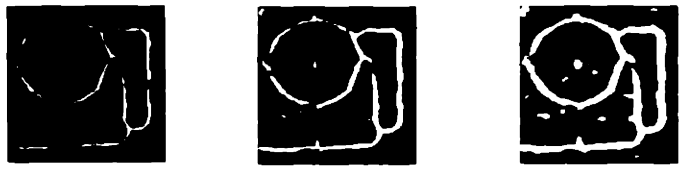
For the different measures a threshold can be derived, above/ below which a motion boundary can be asserted with high certainty.

Detecting Global Extrema

A boundary can be inferred where the first derivative of a measure crosses zero, its second derivative is of the appropriate sign, and its value is below or above a conservative threshold, which has been chosen so that any extremum below or above it can be safely excluded.

Combing the Measures

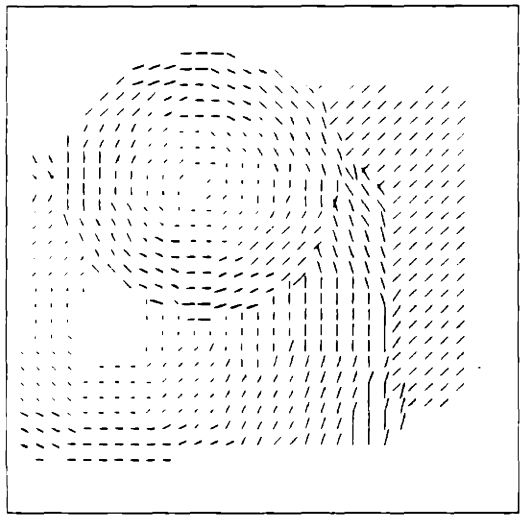
The measures have in common that they have a global extremum at a motion boundary, and that their local extrema anywhere else in the image are weakly correlated. Hence, their thickened extrema contours can be superimposed, and a motion boundary is inferred where they all intersect (thickened by 1, 2 or 3 pixels respectively). This approach has the attractive feature that it does not require the setting of a threshold.



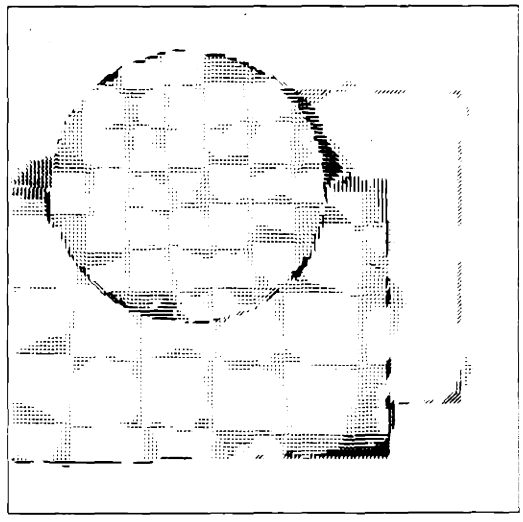
• How to locally estimate the visual flow field early on ?

The highest peak in a local histogram corresponds to the displacement with the most local support. Hence, this displacement represents an estimate of the image flow and the *peak-ratio* reflects how good the estimate is. The computation is well-posed and consistent with human psychophysics.

Estimated Flow Field



Error Flow Field

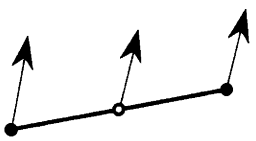


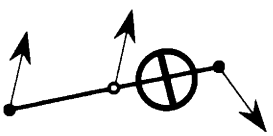
The estimated motion boundaries can be broadly defined and can contain gaps.

• How to extract complete and unique motion boundaries ? How to separate contour segments belonging to differently moving objects ?

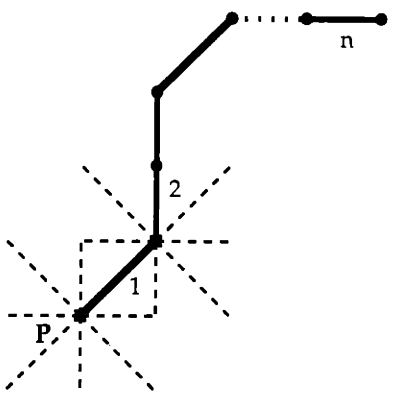
Observation
Object boundaries are generally smooth and the flow vectors along a boundary vary smoothly.

Structural Saliency Method
Extracts boundaries and closes gaps by employing a simple iterative scheme that uses an optimization approach to measure the saliency of curves of line segments in terms of their smoothness and length.
A line segment consisting of three points is created only if the estimated flow vectors associated with its points do not differ by more than two units in order to prevent curves of being formed that wander across boundaries. Each segment corresponds either to a corresponding asserted motion boundary segment or to an empty area or gap, called a virtual segment.





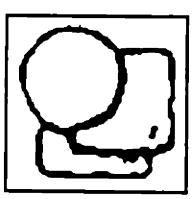
The optimization problem is formulated in terms of maximizing $\Omega(n)$ over all curves of length n starting from P. The computation becomes linear in n if Ω is an extensible function. Hence, the most salient curve of length n at P will be equal to the maxima over all segments leaving P and the maximal curves of length (n-1) starting at the respective end-points of these segments.
The saliency measure is associated with each segment and not with the entire curve.



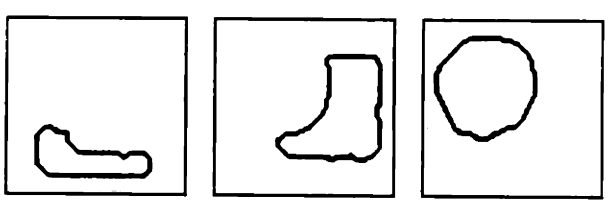
• How to extract a unique contour ?

If the area in which curves are allowed to form is broadly defined then there will be several contours growing alongside each other. To extract the most salient curve, we have to first propagate the saliency value of the most salient segment along the curve that contributed to its value. This is done iteraterively by each segment maximizing over the value of its preferred neighbor and its own. Thus, the largest value will be propagated along its curve. Finally, we perform a non-maximal suppression operation, where each segment suppresses all its neighboring segments if their saliency value is less and if they have similar motion estimates associated with them. Hence, the most salient contours belonging to differently moving objects will remain alongside each other.

Input
Estimated motion boundaries



Output
Connected contours belonging to differently moving objects



ACKNOWLEDGEMENTS

I would like to thank the following people for their contribution to this thesis :

Prof. Shimon Ullman for his guidance and continued support. He represents for me one of leading visionaries in the field of computational vision.

Prof. Whitman Richards, without whom I would not have had the opportunity to study at MIT, and I thank him for his belief in me.

Prof. Ellen Hildreth for her support over the years.

Amnon Sha'ashua for letting me use his code to run the *Structural Saliency Method* and taking the time to explain it to me.

David Clemens for his help and patience in answering my questions.

Jonathan Meyer for his help to retrieve my files and without whom my previous code would have been unaccessible.

Davi Geiger, James Mahoney, Lyle Graham-Borg for their friendship.

Janice Ellertsen for her help to secure the funding needed for me to complete my thesis.

Pamela Robertson-Pearce for her love, emotional support and encouragement as well as helping me to explore my artistic sides.

Elka Spoerri for her love and dedicated support over so many years.

In memory of my father

This thesis describes research done at the Artificial Intelligence Laboratory and the Department of Brain and Cognitive Sciences of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124. The work was also partially supported by the NSF grant IRI-8900267 and Anselm Spoerri has been supported in part by a stipend from the Erziehungs Direktion des Kantons Bern, Switzerland.

Parts of this thesis have been presented and published at the First International Conference on Computer Vision (ICCV) 1987, in London, England [38].

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

It is a major goal of vision to infer the physical properties of the objects present in a scene, such as their three-dimensional structure and motion in space. An essential first step towards this goal is the segmentation of the image into regions that are likely to correspond to different objects.

Why is it important to detect boundaries early on?
- They provide the boundary conditions required by the *early vision modules*.
- Boundaries are ideal for integrating information provided by the different *early vision modules*.
- Boundaries provide the input to *visual routines* that establish higher-order shape properties.
- Salient and grouped boundaries provide the input to *recognition processes* and reduce the combinatorial problem facing them.

This early segmentation can be used to guide and substantially facilitate the further processing of the image. Firstly, it provides the boundary conditions required by many *early vision modules*, such as optical flow, stereopsis, shape from shading, and surface reconstruction. For example, many models for these processes assume that the visible surfaces are generally smooth [14,15,17,19,28,40]. Without prior knowledge of the boundaries, however, these computations tend to impose the smoothness assumption across boundaries, leading to error in the computed motion, stereo and 3-D shape [15,17,40]. Secondly, boundaries are ideal for integrating information provided by the different early vision modules [12]. Thirdly, the early detection of boundaries provides the input to *visual routines* that establish higher-order shape properties and spatial relations among entities in the image [44]. These processes can focus the attention of higher-level modules on the edges of interest in a scene and they can preferentially allocate processing resources to these structures of interest. Fourthly, early segmentation provides the critical input to *recognition processes*, since salient and grouped edges greatly reduce the combinatorial problem facing the recognition methods, which often depend on the number of edge primitives having to be examined.

Hence, a key problem of early vision is the detection of boundaries. This problem, however, is difficult because the only information available is a large array of intensity measurements. Likewise, detection of boundaries from early 2-D or 3-D representations is difficult because they are often sparse, noisy and inaccurate, especially in the vicinity of object boundaries.

1.1 Problem Statement of the Thesis

Perceptual motion studies have shown, using random-dot displays, that the human visual system is able to segment a scene into distinct objects based on motion information alone [3,6,21]. Human observers are very sensitive to relative movement (for review, see [30]), although it appears that a large difference in direction and speed of motion may be required to localize a boundary accurately [15].

What is the role of motion information and what are its advantages?
- Motion can provide boundary and shape information in the absence of distinct intensity edges or other cues.
- Motion boundaries are sparser than intensity edges and are primarily associated with object and depth boundaries.
- Motion-based segmentation can be used for figure-ground separation.

This thesis addresses the problem of detecting boundaries on the basis of motion information alone and it deals specifically with the following questions :

- How to detect motion boundaries early on in a scene containing several moving objects, without prior knowledge of their shapes and motions ?
- How to decouple the estimation of motion boundaries from the computation of a full image flow field ?
- How to integrate the pointwise output of the developed motion boundary estimators with a process that can extract salient, complete and unique contours ?
- How to separate contour segments belonging to differently moving objects and how to group together segments belonging to the same object ?

What makes the detection of motion boundaries difficult?
- Movement of elements in an image is not given directly.
- Dilemma of computation of motion: To detect boundaries by applying existing edge detectors to the computed image flow field requires an almost error-free flow field, but a necessary condition to compute such a flow field is the knowledge of the boundaries prior to its computation.

1.2 The Advantages

Motion boundaries play four useful roles. First, motion can provide boundary and shape information in the absence of distinct intensity edges or cues from other sources of visual information. Second, motion boundaries are sparser than intensity edges, and they are primarily associated with object and depth boundaries rather than texture markings, shadows or highlights [41]. Third, motion-based segmentation offers additional information since in most cases the side of a motion boundary corresponding to the occluding object can be identified [26]. Fourth, motion boundaries can be used to facilitate the computation of the image flow field [15,17].

1.3 The Difficulties

The fundamental problem that arises in the computation of motion and its boundaries is that the movement of elements in an image is not given directly. It has to be computed from more elementary measurements. All we are given initially are the temporal changes of the intensity values at each image point, which allow us only to compute the flow component in the direction of the image gradient due to the *aperture problem* [23].

Aperture Problem
- A moving edge, seen through a circular aperture, seems to be moving normal to itself, while the transverse component of velocity can not be perceived.

One possible solution to this problem is to compute the flow field and its boundaries simultaneously, using for example a Markov Random Field model and its line processes [13,18,24]. These time consuming schemes would be greatly facilitated if the boundaries are either already known or at least estimated. A more common approach is to compute the image flow field first and then to detect motion boundaries. This approach has several inherent difficulties which will be discussed now.

Correspondence Problem
- There is a great deal of ambiguity when trying to find the unique match of a token.

The methods for computing visual motion fall in two classes: intensity-based and token-matching schemes. Intensity-based methods have to integrate the local motion measurements due to the *aperture problem* [23]. This integration problem is commonly solved by assuming that the image flow field varies smoothly in the image [2,15,17,27,28]. This constraint is valid everywhere except at object boundaries. Because of this, considerable error will occur in the vicinity of object boundaries [17,43]. A further problem is that the computed flow field is often noisy and inaccurate due to error in the initial motion measurements. As a consequence, edge detectors that locate sharp changes in the components of the computed image flow field will detect many incorrect motion boundaries [15].

- Intensity-based methods solve the aperture problem by assuming a smoothly varying image flow field to be able to integrate the dense local motion measurements.

Token-matching schemes have to solve the difficult *correspondence problem* in order to compute motion, and they usually produce a sparse flow field [42]. Such a flow field needs to be smoothly interpolated so that edge detectors can be applied to locate the motion boundaries. Without the knowledge of the boundaries, however, the interpolation scheme will cause the motion boundaries to be smoothed over to such a degree that it may become impossible to recover them, or the ones that can still be detected by the edge operator will be poorly localized.

- Token-matching methods have to solve the correspondence problem and assume smoothness to interpolate between the sparse motion estimates.

To summarize, both classes of methods for computing visual motion do not provide an image flow field from which boundaries can be detected easily and reliably. The computation of motion and the detection of motion boundaries is faced with a dilemma: in order to detect boundaries with existing edge detectors, an almost error free and densely defined image flow field is required, but a necessary condition for computing such a flow field is the knowledge of the boundaries prior to its computation.

Thus, it is necessary and desirable to be able to decouple the detection of motion boundaries from the computation of the image flow field. But what information other than the image flow field can be used to detect motion boundaries? Which quantities can be easily computed at such an early stage to compute a useful estimate of the motion boundaries ?

1.4 Detecting Motion Boundaries Early On

How are we going to solve it ?
- In two stages: (i) local estimation of motion discontinuities (ii) extraction of complete boundaries by modifying the *Structural Saliency* Method by Sha'ashua & Ullman.
- Guided by the *Constant Intensity* and the *Local Translation* assumptions, which are relaxed by using a *Gaussian matching function* and a *Gaussian spatial support function* respectively.
- Construct local histograms of the easily computable potential displacements, which will be bimodally distributed at a motion boundary.
- The highest peak in a local histogram estimates the visual motion, which can be used to separate contour segments belonging to differently moving objects.

The early detection of motion boundaries can be performed in two stages: (i) the local estimation of the motion discontinuities; (ii) the extraction of complete boundaries belonging to differently moving objects.

1.4.1 The First Stage

For the first stage, three new methods are developed that can perform the local estimation of motion boundaries: the *Bimodality Tests*, the *Bi-distribution Test* and the *Dynamic Occlusion Method*. It is also shown how visual motion can be locally estimated as a by-product of the early estimation of motion boundaries.

The first two methods make use of the fact that at a motion boundary certain quantities, which can be easily computed early on, will cluster around two different points in a local histogram. The quantities in question are (i) the potential displacements of an image point, and (ii) the flow component measured in the direction of the intensity gradient. The local histograms are constructed at every point using a circular neighborhood whose radius will range between five and eight pixels.

If a local histogram is computed in the vicinity of a motion boundary then the resulting histogram of these quantities will be bimodal, where the two peaks are of roughly equal strength. Hence, the *Bimodality Tests* detect motion boundaries by computing the degree of bimodality present in the local histograms. The *Bi-distribution Test* employs a non-parametric statistical test to detect boundaries, using the fact that the populations of motions are different on the two sides of a boundary. The *Dynamic Occlusion Method* is based on the fact that intensity edges of opposite contrast, called *thin-bars*, will be created or destroyed in the vicinity of a motion boundary. A method is developed that can locally compute the appearance and disappearance of *thin-bars* in a way that is sufficient to estimate motion boundaries, without having to solve a global and difficult correspondence problem.

The computation of the visual flow field and the detection of its boundaries can be performed in parallel, since the highest peak in a local histogram of the potential displacements corresponds to the motion with the most local support. Hence, this displacement represents an estimate of the image flow. The measures that are sensitive to degree of bimodality occurring in the local histograms will reflect how good the estimate is. A mathematical formulation is provided to show that the proposed computation of visual motion is well-posed, and it is demonstrated that the developed method is similar to the *local voting scheme* proposed by Bülthoff, Little & Poggio [7]. The approach of using local neighborhoods to find the displacement with the most local support is consistent with human psychophysics, since it exhibits several of the same "illusions" that humans perceive.

1.4.2 The Second Stage

The pointwise output of the motion boundary estimators is often broadly localized and it can contain gaps. The second stage consists of applying and modifying the *Structural Saliency Method* developed by Sha'ashua & Ullman [37,45] to extract complete and unique boundaries from the pointwise output of the first stage. Boundary segments belonging to differently moving objects are separated by using the motion estimates provided by the first stage to constrain which edge segments can be formed.

The *Structural Saliency Method* employs a simple iterative network and uses an optimization approach to produce a "saliency map", which emphasizes salient locations in the image. The saliency of curves is measured in terms of their smoothness and length, which is often sufficient to perform figure-ground separation. The main properties of the network are: (i) the computations are simple and local, (ii) globally salient structures emerge with a small number of iterations, (iii) there is little dependence on the complexity of the image, (iv) contours are smoothed, gaps are filled in and linking information between edge segments is provided.

The optimization problem is formulated in terms of maximizing a structural saliency measure $\Omega(n)$ over all curves of length n starting from P. The computation is linear in n because Ω has been constructed to be an extensible function. Hence, the most salient curve of length n at P will be equal to the maxima over all segments leaving P and the maximal curves of length $(n-1)$ starting at the respective end-points of these segments.

1.5 Organization of the Thesis

Chapter 2 discusses previous work on the detection of motion boundaries. Chapter 3 presents three new methods that can locally estimate motion boundaries early on: the *Bimodality Tests*, the *Bi-distribution Test* and the *Dynamic Occlusion Method*. It is shown how to infer a motion boundary from the computed measures and how the appropriate thresholds can be derived. Chapter 4 shows how visual motion can be locally estimated as a by-product of the early estimation of motion boundaries. A mathematical formulation is provided for the proposed computation of visual motion and it is demonstrated that the developed method is well-posed. Chapter 5 introduces the *Structural Saliency Method* by Sha'ashua & Ullman and shows how it can be modified to extract complete and unique boundaries from the pointwise output of the motion boundary estimators, whose output is often broadly localized and can contain gaps. Chapter 6 shows the results of applying the methods to image sequences composed of random-dot or natural textures. Chapter 7 shows how the methods can be applied in stereopsis and surface reconstruction. Chapter 8 provides a summary and conclusion.

CHAPTER 2

PREVIOUS WORK

2.1 Introduction

The previous work on the detection of motion boundaries can be categorized by making the following two distinctions. First, there are at least two ways to describe what takes place at an object boundary in the presence of motion. One is that regions of a more distant object will, in general, either appear or disappear from view over time at an object boundary. The other is to observe that if two adjacent surfaces undergo different motions or are separated in depth then they will give rise to a motion discontinuity along their boundary. The second distinction can be further differentiated based on the stage at which the detection of motion boundaries is performed since it can be performed either prior to, simultaneously with or following the computation of the image flow field.

2.2 Detecting Discontinuities Prior to the Computation of the Flow Field

Reichardt et al. [34] propose a method, working on the figure-ground discrimination of the house-fly, where direction selective movement detectors inhibit flicker detectors, when the same movement appears in the center and surround of the motion detectors. Hence, flicker detectors with significant activity indicate the presence of motion boundaries.

Marr & Ullman [23] and Hildreth [15] use the flow component in the direction of the intensity gradient, also called the normal flow component, to detect motion boundaries. They make use of the fact that if two adjacent objects undergo different motions v_1 and v_2, then the normal flow components, whose orientations lie between the directions of $(v_1 + 90^o)$ and $(v_2 + 90^o)$ or $(v_1 - 90^o)$ and $(v_2 - 90^o)$, will change in sign across the boundary *(see Figure 3.2)*. Therefore, a change in the sign of normal flow components with appropriate and roughly equal orientation

signals a motion boundary. This method is limited by the fact that the number of flow components, whose orientations lie between the directions of $(v_1 + 90^o)$ and $(v_2 + 90^o)$ or $(v_1 - 90^o)$ and $(v_2 - 90^o)$, will decrease as an image becomes less textured and the angle between v_1 and v_2 becomes smaller. Furthermore, the neighborhood, over which measurements are collected, will have to be large so that there will be a sufficient number of normal flow components, whose signs can be compared.

2.3 Detecting Discontinuities After the Computation of the Flow Field

Nakayama et al. [29] propose to detect boundaries by using a center-surround operator that signals image flow differences between the center and surround, but their method has not been implemented and tested. Potter [31] employs region growing techniques to group features of similar velocity, assuming that the image flow field is due to translation. Clocksin [9] shows that object and depth boundaries give rise to discontinuities in the magnitude of flow created by an observer translating in a static environment.

For the more general case of unconstrained motion, Thompson et al. [41] show that object boundaries give rise to discontinuities in the image flow field. In principle, these sharp changes could be detected as zero-crossings in the Laplacian of the components of the flow field. In a preceding paper, Thompson et al. [1982] computed the image flow field using a token-matching method. Because the resulting flow field was sparse, they had to smoothly interpolate between the feature points at which the flow field was defined. Without the knowledge of the location of the object boundaries, their interpolation scheme smoothed over the boundaries. As a result, the motion boundaries that could still be detected by the Laplacian operator were poorly localized.

Schunck [35] computes the image flow field using a motion constraint line clustering algorithm. He assumes that the flow field is due to the translation of objects in the scene under orthographic projection. Object boundaries are detected by using an edge detector that locates the sharp changes in the components of the flow field. Schunck utilizes an iterative

procedure that interleaves the application of an edge detector with a smoothing of the computed flow field, in order to reduce the noise that is causing the erroneously detected boundaries.

Adiv [1] first partitions a flow field into connected segments, where each segment is consistent with a rigid motion of a roughly planar surface. A global, multipass Hough transform is used to determine the parameters describing the motion and the plane. The segments are then grouped under the hypothesis that they are created by a single, rigidly moving object, by searching for the motion parameters that are compatible with all the segments in the corresponding group.

Terzopoulos [40] proposes to detect discontinuities in sparse surface representations by marking locations where the thin plate used to interpolate between the sparse data points has an inflection point and its gradient is above some threshold. To overcome the shortcoming that the smoothing thin plate tends to obscure boundaries, a cost is also introduced for the placement of a boundary, leading to a non-convex cost functional that has to be minimized.

2.4 Detecting Dynamic Occlusion After the Computation of the Flow Field

An example of the approach that also detects boundaries after the flow field computation, but uses the fact that dynamic occlusion occurs at object boundaries, is the work of Mutch & Thompson [26]. They use a relaxation technique to compute the flow field. Areas in the image with a high percentage of features that do not have a match in the previous or subsequent frame are identified as regions that have appeared or disappeared, respectively.

2.5 The Simultaneous Computation of the Flow Field and its Discontinuities

Wohn & Waxman [47] suggest a scheme where the motion segmentation is performed by detecting "boundaries of analyticity", that is where an approximation of the local flow field by second order polynomials breaks down. The boundaries are located within the process that models the local flow field.

Hutchinson, Koch, Luo & Mead [18] and Gamble & Poggio [12] propose that binary line processes, first introduced in the Markov Random Field method developed by [13], can signal boundaries. At locations where such a line process is set, an edge is postulated ensuring that the smoothness assumption is not imposed across them. The computation of the image flow field and the activation of the binary line processes is then performed so as to minimize a non-convex energy functional.

Hutchinson et al. and Gamble et al. restrict the location of motion boundaries to coincide with the location of intensity edges. This strategy effectively prevents motion boundaries from forming at locations where no intensity edges exist, unless strongly suggested by motion data. Conversely, however, intensity edges by themselves will not induce the formation of discontinuities in the absence of sharp changes in motion.

Hutchinson et al. introduce the following procedure to cope with the different velocity gradients that are generally present in a scene. The formation of lines is initially strongly penalized, encouraging a smooth image flow field everywhere except at very steep velocity gradients. A smaller price has to be paid subsequently, and the image flow field will break at smaller flow gradients. The final state of the network is independent of the limiting flow gradient, and their method has been successfully applied to motion sequences.

CHAPTER 3

THE EARLY ESTIMATION OF MOTION BOUNDARIES

3.1 Introduction

Observation
- Object and depth boundaries give rise to discontinuities in the visual flow field and cause dynamic occlusion.

In this chapter, we will describe three new methods that can estimate motion boundaries at an early stage in the processing of visual information, using only motion and no intensity boundary information. The methods make use of the following two facts. First, object boundaries give rise to discontinuities in the flow field, i.e., the velocities on the two sides of a boundary cluster around two different points in a velocity histogram. Second, dynamic occlusion occurs at an object boundary in the presence of motion, and therefore spatial relationships between simple image features change most dramatically in the vicinity of motion boundaries.

This chapter consists of four parts. First, we will describe the *Bimodality Tests* that estimate motion boundaries by computing the degree of bimodality present in the local histograms of the potential displacements or normal flow components. Second, we will introduce an application of the *Kolmogorov-Smirnov Test* in the *Bi-distribution Test*, that detects boundaries by measuring the probability that two histograms have been created by the same population of motions. Third, we will discuss how to infer the presence of a motion boundary from the measures computed by the *Bimodality Tests* and the *Bi-distribution Test*. Fourth, we will describe the *Dynamic Occlusion Method* that makes use of the fact that thin-bars are created or destroyed at a motion boundary.

Before describing the methods in detail, we will discuss the matching primitives used, and why either the local histograms of the potential displacements or the normal flow components contain sufficient information to estimate motion boundaries. We will also outline the constraints that can be used to filter the local histograms and how to handle images that contain only little texture and are sensitive to the effects of noise.

3.1.1 Matching Primitives

The *Bimodality Tests* and the *Bi-distribution Test* are flexible in terms of the matching primitives used, since either intensities, zero-crossings or other edge features can be used.

Flexible in terms of the matching primitives used:
- Smoothed intensity values are densely defined and we use a *Gaussian matching function* to account for occurring changes in intensity between frames.
- Edge-tokens are more robust, but they are sparse and they can be non-uniformly distributed.

Intensity values have the advantage that the potential displacements can be computed at almost every point. Hence, the density of matching primitives will be uniform across a boundary, and the methods will be more robust, because there will be more contributors to the local histograms. The intensity values are also smoothed by convolving them with a Gaussian filter to increase their reliability.

We are, however, implicitly assuming that the intensity values at corresponding points do not change greatly, although they are sensitive to noise and, more importantly, to changes in illumination. These effects will be minor as long as there is sufficient texture in the image. The problem will be more serious in parts of the image where intensity changes slowly. To account for these gradual changes in intensity, we use a *Gaussian matching function*, which depends on the difference in intensity at the two points which define a particular displacement, to weigh the possible displacements of a point. The smaller the difference in intensity, the greater the weight that is assigned to a particular displacement. The spread of the *Gaussian matching function* can be chosen to reflect the estimated noise in the intensity measurements.

Using zero-crossings or other edge features as matching primitives has the advantage that they are more likely to be tied to a physical event in the scene, and are therefore more stable with respect to noise and changes in illumination[1]. These primitives, however, have the disadvantage that they tend to be sparse, and their density can be non-uniform across the image. In particular, the less textured the image, the greater the size of the histogram neighborhood needs to be for there to be sufficient contributors to the local histograms. This increase in the size of the histogram neighborhood, however, can decrease the robustness of the developed

[1] It has been noted that methods superficially so different as edge-based and intensity-based flow field computations give very similar results and are to a certain degree equivalent [7].

methods because it increases the likelihood that the image flow field changes too rapidly over the spatial support used to compute the histograms.

3.1.2 Input Representation

The input representation used by the *Bimodality Tests* and the *Bi-distribution Test* is a local histogram constructed at each image point. The matching primitives that lie within a circular neighborhood will contribute either the match scores for all the possible displacements or their normal flow component to the local histogram. The radius of the spatial support used to compute the histograms will typically range between five and eight pixels.

The local histograms of the potential displacements contain sufficient information to infer the presence of motion boundaries, because, in a region that is translating locally, all the matching primitives will have one potential displacement in common, namely, the one which corresponds to the translation of the region. Thus, there will be a single strong peak at the location in the histogram that corresponds to the local translation[2]. In the vicinity of an object boundary, the local histogram will have two peaks of roughly equal height because the matching primitives in one half of the histogram neighborhood will have one displacement in common, whereas the other half will have a different displacement in common. Hence, motion boundaries give rise to local histograms that have a bimodal distribution *(see Figure 3.1)*.

• The potential displacements of points in the vicinity of a motion boundary will cluster around two different points in a local histogram, which collects the votes for the different possible motions.

As previously noted, the local motion measurements provide only the normal flow components. These components, however, provide sufficient information to detect motion boundaries for the following reason. Normal flow components that have the same orientation will have both the same sign and roughly equal magnitude in a region that is locally translating. If, however, two adjacent objects move differently then the normal flow components of most orientations will have different magnitudes across the boundary *(see Figure 3.2)*.

---

[2] provided the motion primitives are not arranged in a regular pattern, as would be the case for an image composed of stripes, causing the resulting histogram to contain ridges.

Figure 3.1 **The Information Provided by the Potential Displacements.**
Shows a 1-D slice through the two-dimensional local histograms that collect the potential displacements of the points that lie within a circle centered at the locations (x_1, y_0), (x_2, y_0), (x_3, y_0), respectively. The solid vectors represent the correct local displacements and the dashed vectors represent the other, but spurious potential displacements.

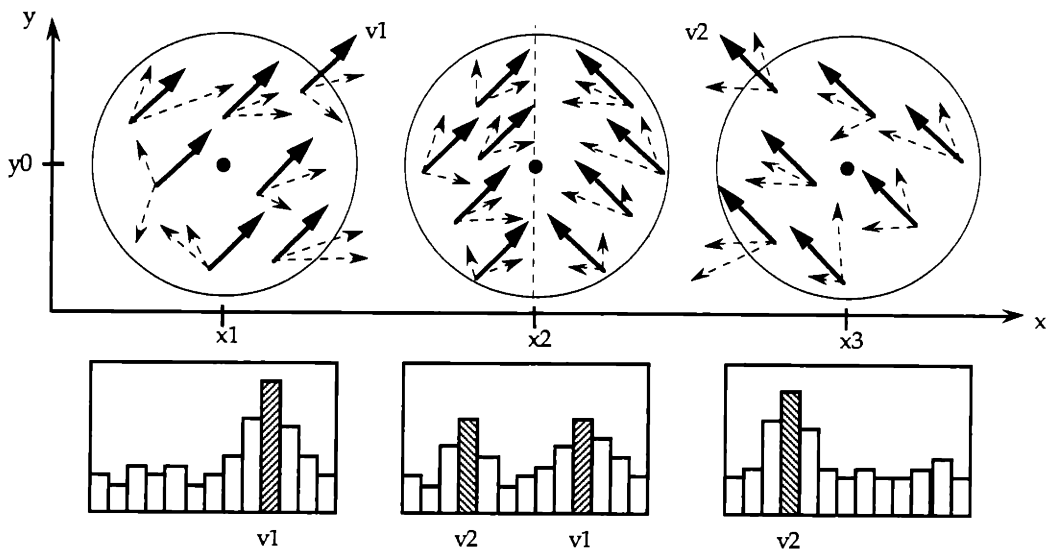
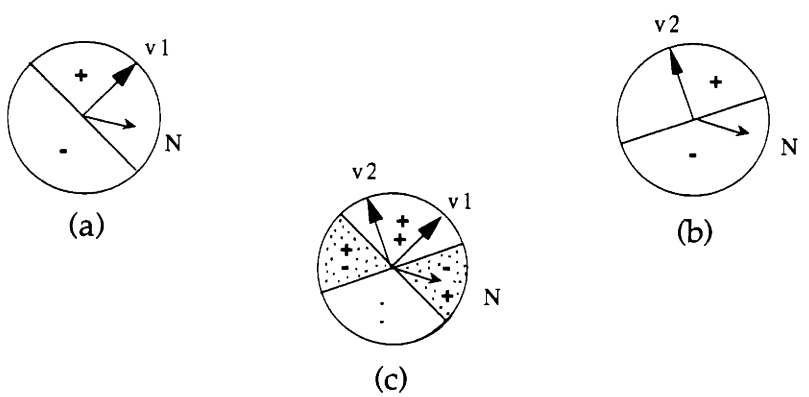


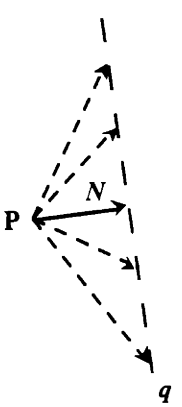
Figure 3.2 **The Information Provided by the Normal Flow Vectors.**
(a) and (b) show for which orientations of the normal flow vector N the sign of its component will be positive or negative with respect to v_1 and v_2, respectively. (c) Combines results of (a) and (b) and the textured areas show for which orientations the component of the normal flow vector N will be of opposite sign across a motion boundary.



Hence, a histogram of the normal flow components that lie in the same narrow orientation range will be bimodal at a motion boundary. The distance between the two modes will be a function of the angle α between the normal flow vector N and the bisector of v_1 and v_2 as well as the resolution of the histogram. The smaller the angle α, the greater the distance between the two peaks will be. The resolution of the histograms can be chosen arbitrarily, but there is the following trade-off: the coarser the resolution, the more robust the histograms. But, the number of the orientation ranges that will display bimodality at a motion boundary will be less, and the flow difference across a boundary will have to be larger, in order for there to be two distinct modes in the histogram. We will choose the resolution to be equal to the one used for the potential displacements. This should ensure that the histograms will be robust and that there will be a sufficient number of disjoint orientation ranges that are sensitive to motion boundaries. We will detect motion boundaries by computing the local histograms for a number of disjoint orientation ranges and analyzing them, using the methods that will be described below.

The use of the normal flow components to segment a scene extends the work by Marr & Ullman [23] and by Hildreth [15] in two ways. First, it uses the magnitude as well as the sign of the components to detect motion boundaries. Second, the flow components at any point where the matching primitives of our choice are defined will contribute to the local histogram, instead of just the normal flow components that can be measured along contours.

The information provided by the measured normal flow vector N could also be used in another way to detect motion boundaries. The normal flow vector at a point P defines a line q on which its corresponding point P' in the next frame has to lie, *(see Figure 3.3)*. Hence, in a region that is locally translating, all lines defined by the normal flow components will intersect roughly at the location in a velocity histogram that corresponds to the local translation of the region. At points in the vicinity of a motion boundary, the local histogram will have two peaks of roughly equal height, because the lines defined by the normal flow components in one half of the neighborhood will intersect at one particular point, whereas the ones from the other half will intersect at a different point.



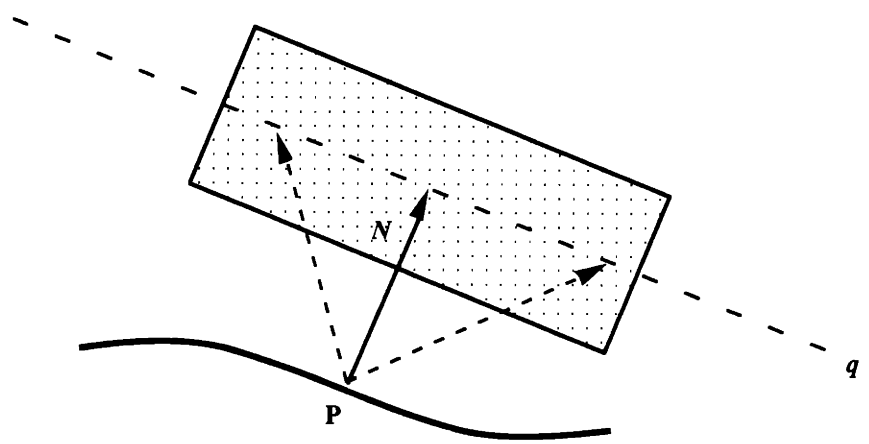
3.1.3 Ways to Filter the Histograms

In this part, we will describe the constraints that can be used to remove some of the incorrect potential displacements of a motion primitive. This filtering reduces the noise in the local histograms and it sharpens the peaks.

Matching Constraints
Corresponding motion primitives have :
- Same contrast
- Match lies on the line defined by normal flow component.
- Information measured at different scales can be combined.

The first constraint is that corresponding matching primitives must have the same sign of contrast, i.e. the scalar product of their intensity gradients must be positive. Similarly, the angle between the intensity gradients at corresponding primitives should be within a certain bound for small rotations. The second constraint is that the normal flow vector N at a point P defines a line q on which the corresponding point in the subsequent frame has to lie. Hence, a rectangular window can be specified within which the corresponding motion primitive must lie, where the dimensions of this window are chosen to account for errors in the measured flow components. This constraint greatly reduces the number of potential displacements *(see Figure 3.3)*. The third constraint is that a match must lie in the intersection of the bands defined by the normal flow components, which have been measured at different scales.

Figure 3.3 **The Normal Flow Constraint.**
The normal flow component N at a point P defines a line q on which the corresponding point in the subsequent frame has to lie. A rectangular window can be specified within which the corresponding motion primitive must lie, and its dimensions are chosen to account for measurement errors and the maximal expected displacement.



3.1.4 Ways to Handle Images with Sparse Texture

- Use magnitude of intensity gradient or its local average to suppress false alarms in regions with little texture.

Motion sequences that have sparse texture are very sensitive to the effects of noise. Hence, the intensity values at corresponding points will most likely not be the same, and the potential displacements that have been computed using the *Gaussian matching function*, which favors constant intensity, will assign the highest weight to the wrong displacements.

We try to solve this problem by, firstly, weighing the contributions to the local histograms based on the magnitude of their intensity gradient or by allowing points to contribute only if their gradient is above a certain threshold. This places our scheme midway between area- and edge-based approaches. Edge locations are favored because the gradient is high, but other places contribute as well. Secondly, we compute the average of the gradient over the neighborhood used to compute the histograms and we suppress the output of the methods that estimate motion boundaries if the average is not above a chosen threshold.

3.2 The Bimodality Tests

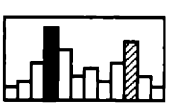
We will now present two methods that locate motion boundaries by detecting the resulting bimodality in the local histograms computed at a boundary. In the first method, three measures are computed that are sensitive to the degree of bimodality in the histograms. We will discuss the assumption of *local translation* that underlies these three measures, and introduce a *Gaussian spatial support function* as a way to relax this assumption. The second method detects bimodality by applying the chi-square test. In this discussion, we will consider the case where the potential displacements are the input to the local histograms, but what will be said applies equally well to the normal flow components.

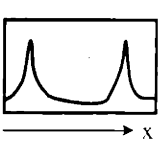
3.2.1 The Ratio Measures

This method consists of three measures that each capture and monitor a different characteristic of a motion boundary. The local histograms must contain two modes of roughly equal height at a boundary, assuming local translation. This is captured by the *peak-ratio*. At a motion boundary the votes will not just cluster around the correct displacement, the "signal", but will be more spread out due the votes from the other side of the boundary. This is measured by the *signal-noise-ratio*. Finally, the displacement receiving the most votes should receive minimal local support at a motion boundary, which is measured by the *local-support-ratio*. These three ratios all have a global extremum at a motion boundary, and their local extrema anywhere else in the image are weakly correlated with each other.

• The *Ratio Measures* have a global extremum at a motion boundary, and their local extrema anywhere else are weakly correlated.

The **Peak-Ratio** measures the degree of bimodality by comparing the heights of the two highest peaks in a local histogram. It is equal to the ratio of the height of the second highest and of the height of the highest peak. Hence, the height of the peaks is used to represent the strength of the peaks. This is a reasonable approximation to make as long as the local flow field can be assumed to be constant over the spatial support used to compute the local histogram.

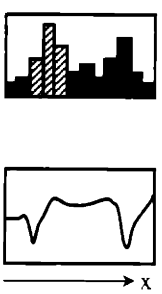




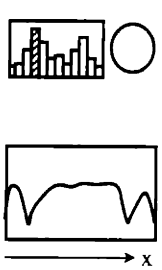
When we compute the two highest peaks, we require that their respective neighboring displacements received strictly less votes. This

ensures that the two highest peaks are separated by at least two displacement units and furthermore, that no motion boundaries are asserted within a moving object whose image flow field is composed of patches of uniform motion that differ by one displacement unit.

The *peak-ratio* will be small in a region that is locally translating. This is because there will be one strong peak at the location corresponding to the local translation, while the second highest peak, which will be due to the incorrect potential displacements, will be small in comparison. At a boundary the two highest peaks will be of roughly equal height, because the matching primitives in one half of the spatial support will have one particular displacement in common, whereas the matching primitives in the other half will have another displacement in common that receives the highest matching score. Thus, the *peak-ratio* will generally have a global maximum close to 1.0 at a motion boundary *(see Figure 3.4)* .



The **Signal-Noise-Ratio** is equal to the ratio of the number of votes for the highest peak and its neighbors and of the number of votes for the remaining displacements in the histogram. In a region that is locally translating, all the points in the histogram, other than the one corresponding to the local translation, will receive some votes due to the incorrect potential displacements. We will refer to these votes as the noise activity in the histogram. The *signal-noise-ratio* will have a global minima at a motion boundary because the heights of the highest peak and its neighbors, the "signal", will decrease, whereas the noise activity will increase due to the votes from the other side of the boundary *(see Figure 3.4)*.



The **Local-Support-Ratio** measures how many of the contributors to the local histogram have supported the displacement with the most votes. This measure is equal to the ratio of the height of the highest peak and the maximal possible local support (which is equal to the area of the neighborhood used to compute the histogram, provided that all points are weighted equally, see also section 3.2.2). The *local-support-ratio* will be close to 1.0 in a region that is locally translating because almost all the points will have a potential displacement that receives the highest matching score and is equal to the local translation. It will have a global minimum below 0.5 at a boundary, because at least half of the matching primitives will not have a potential displacement that votes most strongly for the highest peak *(see Figure 3.4)*.

3.2.2 The Local Translation Assumption and Ways to Relax it

Assumption
- Visual flow field is *locally constant* over the spatial support used to compute the local histograms.

The assumption that underlies the computation of the above three measures is that the visual flow field is *locally constant* over the spatial support used to compute the local histograms. This assumption is strictly only true for the projected flow field of a 3D planar surface patch, translating parallel to the image plane under orthographic projection. It is, however, a satisfactory local assumption, and it is sufficient to just use the height of the peaks to compute the degree of bimodality present in the histograms. It is also assumed that the matching primitives are not arranged in a regular pattern; as would be the case for an image composed of stripes, which would cause the resulting histogram to contain ridges.

- The smaller the size of the histogram neighborhood, the steeper the magnitude of the flow gradient can be, but the less robust the three measures, because there will be fewer contributors to the local histogram.

The size of the histogram neighborhood imposes an upper limit on the magnitude of the flow field gradient that can be tolerated, so that the local translation assumption still holds. In general, there is also the following trade-off between the size of the histogram neighborhood, how much the flow field can change locally and the robustness of the histogram method: the smaller the size of the histogram neighborhood, the steeper the slope of the flow gradient can be. The smaller the neighborhood, the less robust the three measures, because there will be fewer contributors to the local histogram. We employ a circular neighborhood for the construction of the histograms, with a radius between five and eight pixels. This range of radii has proved sufficient to estimate motion boundaries reliably.

- Relax *Local Translation Assumption* by using *Gaussian spatial support function* that weighs points less that are farther away from the point at which the histogram is computed, since points farther apart are less likely to move equally in a smoothly varying flow field.

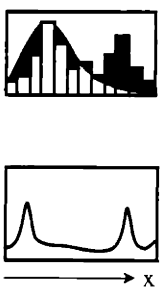
There are at least three ways to handle the situation where the assumption of local translation should be relaxed and the local flow field changes too quickly over the spatial support used to construct the local histograms. First, we can use a *Gaussian spatial support function* that weighs contributors to a local histogram less that are farther away from the point at which the histogram is computed. This will account for the fact that the flow vectors at points farther apart are less likely to be equal in a smoothly varying flow field. It will also sharpen the response of the ratio measures, as is shown in section 3.4.1.1. Second, the flow field can be "slowed down" by using a coarser resolution for the histogram. Third, a measure of the broadness of the peaks could be computed and incorporated in the analysis.

3.2.3 A Statistical Test

Due to noisy intensity measurements, the potential displacements of many of the matching primitives receiving the highest weight may not contain the correct displacement. This could cause the peaks to be broadly or ill-defined. It could also have the effect that the second highest peak is just a major sub-peak of the highest peak. These concerns lead us to consider the following statistical method.

Chi-Square Test
- Measures how well a Gaussian distribution can be fitted to a local histogram.

The **Chi-Square Test** will measure how well a Gaussian distribution can be fitted to a local histogram. Motion boundaries cause the distribution in the local histograms to be bimodal, whereas anywhere else the histograms will be unimodal. Due to noise and errors in the intensity measurements, the peaks of the histograms might not be well defined, but their unimodal or bimodal nature will be preserved. We estimate the parameters of the Gaussian distribution by requiring that it be centered at and pass through the highest peak of the local histogram. Hence, the error of trying to fit a Gaussian distribution to the histogram will be maximal in the vicinity of a boundary *(see Figure 3.4)*.



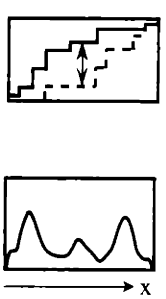
3.3 The Bi-distribution Test

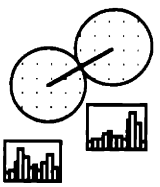
The input to this method is also a local histogram of the potential displacements or normal flow components. The difference, however, is that it attempts to detect motion boundaries by comparing histograms computed at different image points, rather than by analyzing the individual histograms.

Kolmogorov-Smirnov Test
- Compares two local histograms by computing the maximal absolute difference between their cumulative density functions.

3.3.1 A Non-Parametric Statistical Test

The **Kolmogorov-Smirnov Test** measures the probability that two local histograms have been created by the same population of motions. It does this by computing the maximal absolute difference between the cumulative density functions of the two histograms. The *Kolmogorov-Smirnov measure* will be maximal in the vicinity of a motion boundary because the histograms on either side of the boundary are created by different populations of displacements *(see Figure 3.4)*.

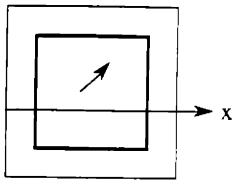


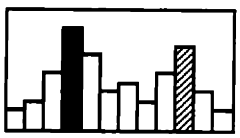


At each image point the *Kolmogorov-Smirnov measure* is computed by comparing the histograms constructed at two points, whose connecting line passes through the point in question, and which are separated by twice the radius of the histogram neighborhood. Several orientations of this connecting line are used to detect motion boundaries of all orientations. The *Kolmogorov-Smirnov measure* that is assigned to a point is the maximum of the measures that have been computed for each of the chosen orientations.

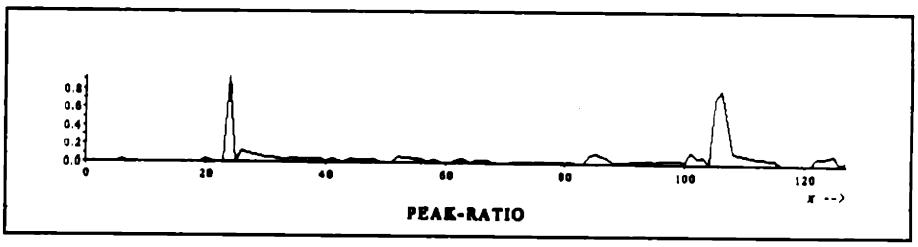
This test has the advantage that it does not depend on the form of the histograms that are being compared. Also not a great deal needs to be known about the nature of the two histograms. There are, however, the following limitations and trade-offs when comparing the histograms constructed at two different points: the more the spatial supports used to compute the two histograms overlap, the less the two histograms will differ. The greater, however, the distance between the two points, the more likely it will be that two histograms have been created by different populations of motions, although the two points might still belong to the same object. For example, if the *Kolmogorov-Smirnov measure* is computed in the center of a rotating object then it will be maximal there, because any two histograms that are being compared will have their peaks at different locations (e.g. notice the high *Kolmogorov-Smirnov measure* at the center of the rotating circle in *Figure 6.1*).

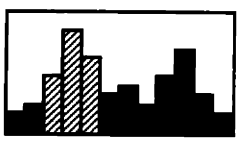
Figure 3.4 **The Developed Measures to Estimate Motion Boundaries.**
The left column shows the definition of the five measures that are sensitive to a motion boundary, and the right column displays their value along a scanline in a random-dot image containing a translating square.



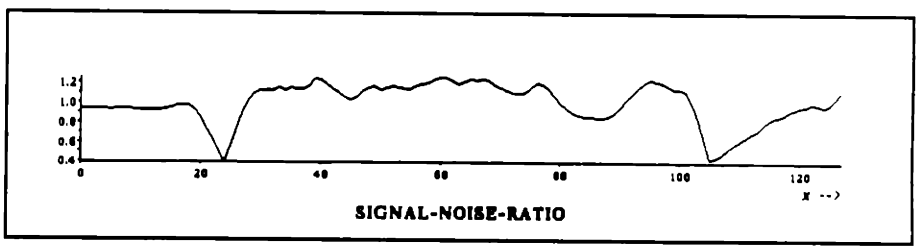


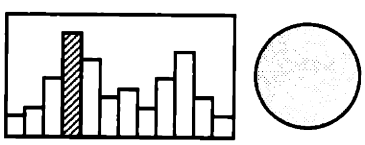
Peak-ratio
Ratio of the height of the second highest and of the highest peak.



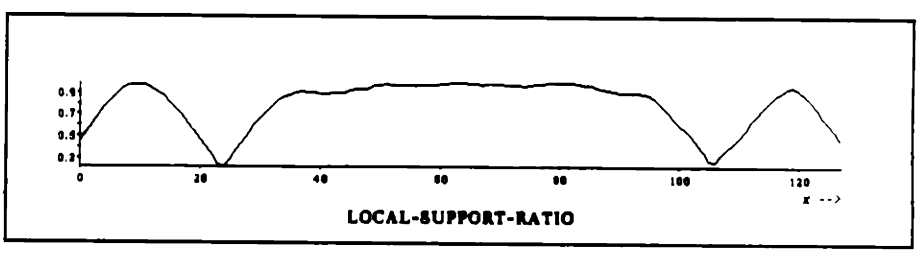


Signal-Noise-ratio
Ratio of the votes for the highest peak & its neighbors and of the votes for the remaining displacements.

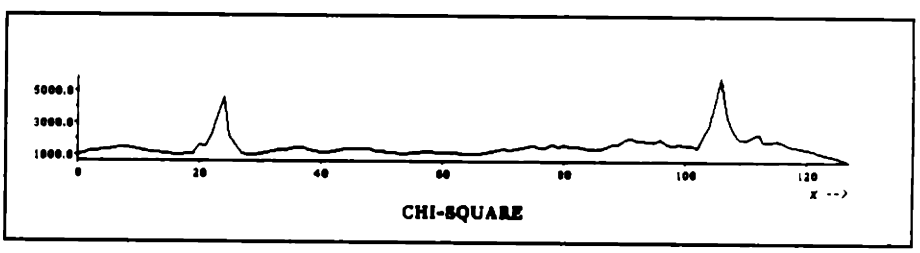


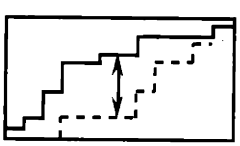


Local-Support-ratio
Ratio of the highest peak and the area of the circular histogram support.

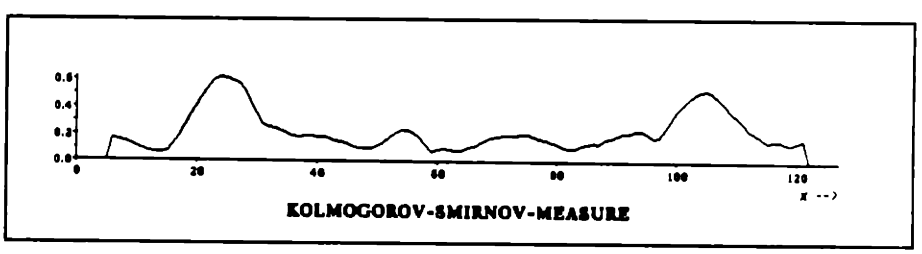


Chi-Square
Measures how well a Gaussian distribution can be fitted to a local histogram.





Kolmogorov-Smirnov
Measures the probability that two histograms have been created by the same population of motions.

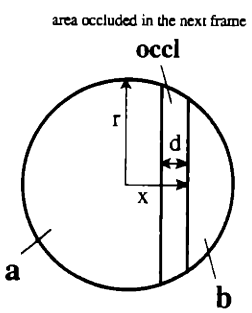


3.4 Inferring Boundaries

We have introduced five measures that are sensitive to the presence of motion boundaries, because they each capture and monitor a different characteristic of a motion boundary. The question arises of how and when to infer a motion boundary so that few actual boundaries are being missed and few spurious ones are being accepted. We will consider thresholding and the detection of global extrema as ways to infer motion boundaries. We will also address how well the detected boundaries are localized.

3.4.1 Thresholds and their Derivation

For the *Ratio Measures*, a threshold can be derived by calculating their expected value as a function of the histogram neighborhood radius r and the distance x from the boundary at which the local histogram is computed. We assume that the correct flow field is given and we consider shearing[3] and occluding motion, where d denotes the width of the area occluded in the subsequent frame. Hence, the height of the two highest peaks is equal to areas a and b of the circular support used to compute the local histograms.



$$peak\text{-}ratio = \frac{height\ of\ 2^{nd} highest\ peak}{height\ of\ highest\ peak} = \frac{b}{a}$$

$$local\text{-}support\text{-}ratio = \frac{height\ of\ highest\ peak}{maximal\ local\ support} = \frac{a}{c}$$

where

$$c = a + b + occl = area\ of\ circle$$

$$a = \pi\ r^2 - \left\{ r^2 \cdot acos\left(\frac{x-d}{r}\right) - (x-d)\sqrt{r^2-(x-d)^2} \right\},\ where\ \frac{d}{2} \le |x-d| \le r$$

$$b = r^2 \cdot acos\left(\frac{x}{r}\right) - x \cdot \sqrt{r^2 - x^2},\ where\ \frac{d}{2} \le |x| \le r$$

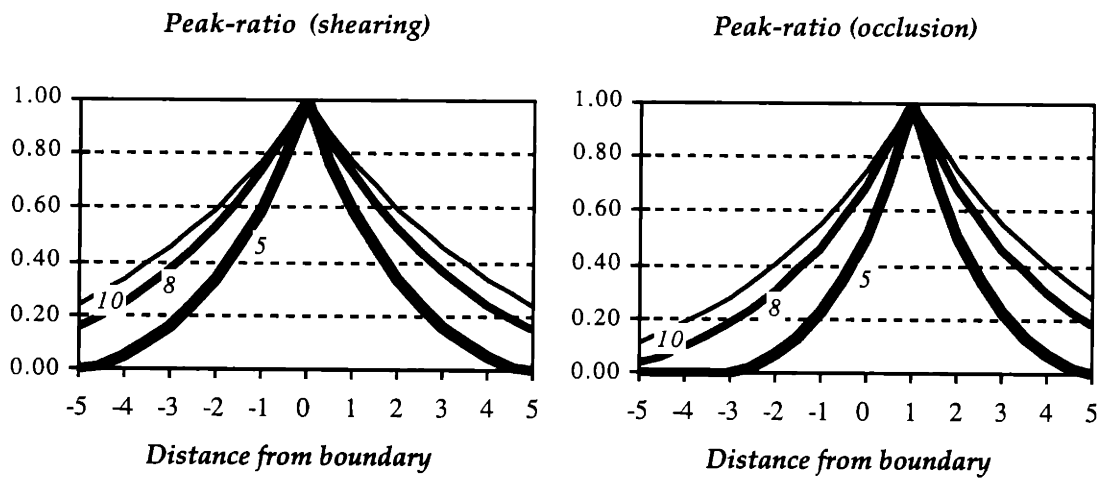
$$\text{if } occl = 0 \text{ then } peak\text{-}ratio = \frac{1 - local\text{-}support\text{-}ratio}{local\text{-}support\text{-}ratio}$$

[3] Shearing motion occurs when the relative movement between two objects is in the direction of their boundary and hence no occlusion occurs.

For shearing motion and radii 5, 8 and 10 pixels, the *peak-ratio* will be 0.34, 0.52 and 0.60, respectively, two pixels away from the boundary, (*see Figure 3.5*). For occluding motion, the *peak-ratio* will be maximal one pixel away from the boundary (for explanation see section 3.4.3), and it will be 0.23, 0.46 and 0.55, respectively, three pixels away from the boundary. This leads us to use a threshold of 0.8 for the *peak-ratio*, because this ensures that few actual boundaries are being missed and few spurious ones are being accepted. We have obtained good results with this threshold, regardless of the type of display or motion.

• Use a threshold of 0.8 for the *peak-ratio*.

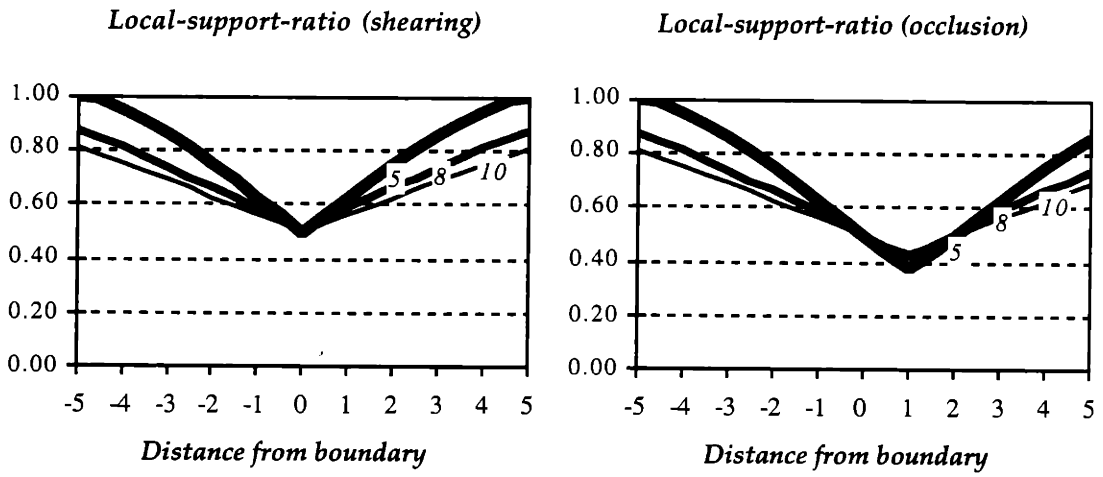
Figure 3.5 **The Derivation of a Threshold for the *Peak-Ratio*.**
The right and left panels show the expected value of the *peak-ratio* for the case of shearing and occluding motion, respectively, and its value has been computed as a function of the radius r = 5, 8, 10 of the circular neighborhood used to construct the histogram and as a function of the distance x from the boundary at which the local histogram has been computed. For the case of occluding motion, the width d of the area occluded in the next frame is assumed to be equal to two.



Similar graphs can be computed for the *local-support-ratio*, (*see Figure 3.6*). For occluding motion and radii 5, 8 and 10 pixels, the *local-support-ratio* will be minimal at one pixel away from the actual boundary, and it will be 0.63, 0.58 and 0.56, respectively, three pixels away from the boundary. This leads us to use a threshold that ranges between 0.45 and 0.6, where any value below it will be considered.

• Use a threshold between 0.45 and 0.6 for *local-support ratio*.

Figure 3.6 **The Derivation of a Threshold for the *Local-Support-Ratio*.**
The right and left panels show the expected value of the *local-support-ratio* for the case of shearing and occluding motion, respectively, and its value has been computed as a function of the radius r = 5, 8, 10 of the circular neighborhood used to construct the histogram and as a function of the distance x from the boundary at which the local histogram has been computed. For the case of occluding motion, the width d of the area occluded in the next frame is assumed to be equal to two.



For the *signal-noise-ratio*, a threshold can be derived by using the following approximation. The *signal-noise-ratio* has been defined to be equal to the ratio of the local support for the highest peak and its neighbors, referred to as the "signal", and the total number of votes minus the "signal". If we assume that the total of votes is a multiple of the area of the histogram neighborhood[4], and that the "signal" is a multiple of the height of the highest peak, then the *signal-noise-ratio* will be equal to :

$$signal\text{-}noise\text{-}ratio = \frac{signal}{total\ votes\ -\ signal}$$

if $total\ votes = \alpha \cdot c$ and $signal = \beta \cdot a$ then

$$signal\text{-}noise\text{-}ratio = \frac{\beta \cdot a}{\alpha \cdot c - \beta \cdot a} = \frac{a}{\delta \cdot c - a}, \text{ where } \delta = \frac{\alpha}{\beta}$$

if $\delta = 1$ then

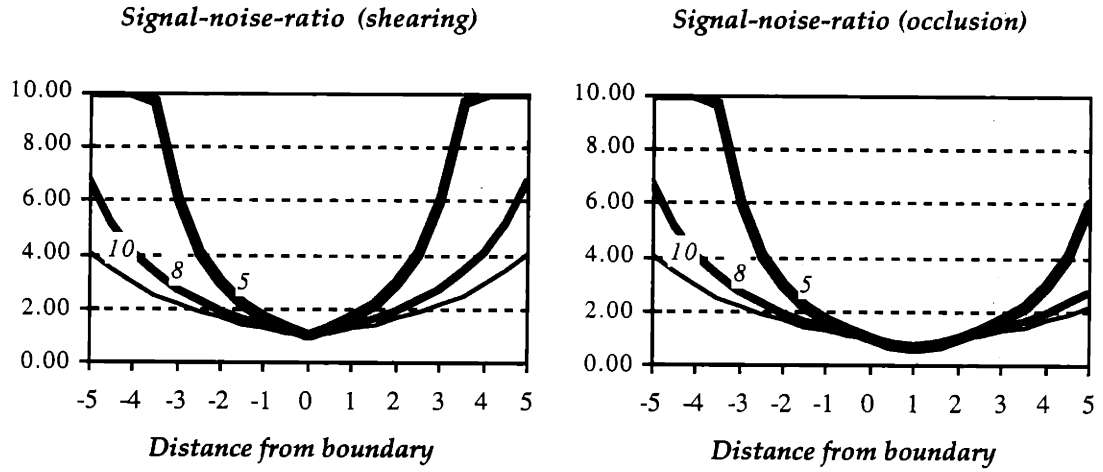
$$signal\text{-}noise\text{-}ratio = \frac{a}{c - a} = \frac{local\text{-}support\text{-}ratio}{1 - local\text{-}support\text{-}ratio}$$

[4] which is equivalent to assuming that each point has a certain number of potential displacements on the average.

For occluding motion and radii 5, 8 and 10 pixels, the *signal-noise-ratio* will be minimal at one pixel away from the actual boundary and it will be equal to 0.60, 0.73 and 0.77, respectively, and it will be 1.68, 1.38 and 1.29, respectively, three pixels away from the boundary, *(see Figure 3.7)*. These values represent the upper bounds for the *signal-noise-ratio*, and we will use, in general, a threshold of 0.6.

• Use a threshold 0.6 for *signal-noise-ratio*.

Figure 3.7 **The Derivation of a Threshold for the *Signal-Noise-Ratio*.**
The right and left panels show the expected value of the *signal-noise-ratio* for the case of shearing and occluding motion, respectively, and its value has been computed as a function of the radius r = 5, 8, 10 of the circular neighborhood used to construct the histogram and as a function of the distance x from the boundary at which the local histogram has been computed. For the case of occluding motion, the width d of the area occluded in the next frame is assumed to be equal to two.



For the *chi-square* and the *Kolmogorov-Smirnov measure*, a confidence level can be derived. For example, the confidence level for the *Kolmogorov-Smirnov measure* will be roughly 0.1. This confidence level, however, is too low to be used to localize the motion boundaries for the following reason. The *Kolmogorov-Smirnov measure* can be above this confidence level even for points that lie in a translating region because the matching scores of the potential displacements associated with the matching primitives can be sufficiently different. We will therefore use a threshold between 0.4 and 0.6 to detect and localize motion boundaries, and reasonable results have been obtained with this choice.

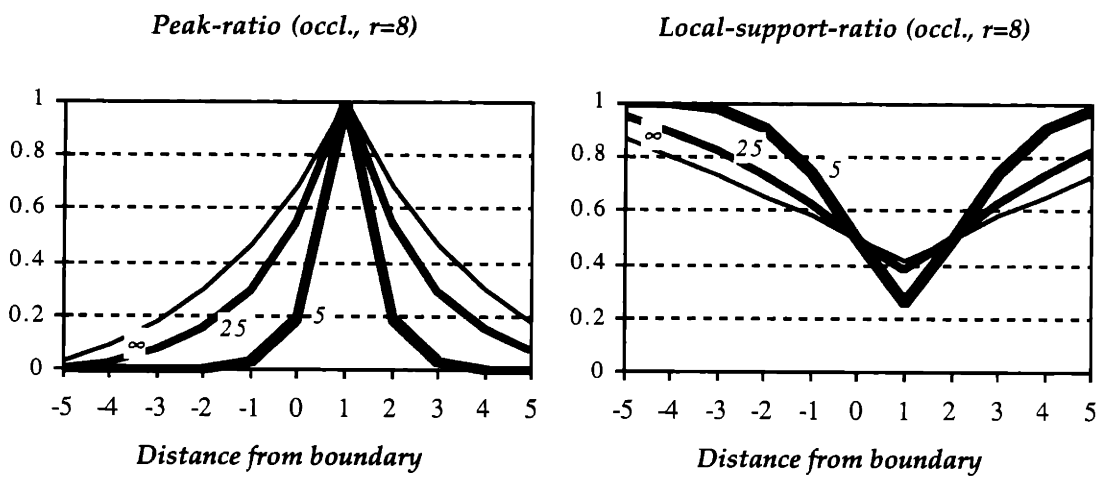
• Use a threshold between 0.4 - 0.6 for the *Kolmogorov-Smirnov measure*.

3.4.1.1 How to Sharpen the Response of the Ratio Measures

• *Gaussian spatial support function* sharpens response of ratio measures.

If we use, as mentioned in section 3.2.2, a *Gaussian spatial support function* with sigma δ that weighs contributing points less that are farther away from the point at which the histogram is computed, then this will cause the response of the ratio measures to be sharpened, *(see Figure 3.8)*. The smaller sigma δ, the sharper the response, and the next figure shows the resulting responses for sigma $\delta = 5$, 25 and ∞ (which is equivalent to weighing all contributing points equally), where $r = 8$ and we consider occluding motion.

Figure 3.8 **Sharpening the Response of the *Ratio Measures*.**
The right and left panels show the expected value of the *peak-ratio* and *local-support-ratio*, respectively, if a *Gaussian spatial support function* with sigma = 5, 25 or ∞ is used to weigh the contributing points less that are farther away from the point at which the histogram is computed. Occluding motion is assumed and the radius r of the circular neighborhood used to construct the histogram is equal to eight. The smaller sigma, the sharper the response of the two measures.



3.4.2 The Detection of Global Extrema

As *Figure 3.4* has shown, all the proposed measures have a global extremum in the vicinity of a motion boundary. There are at least two ways in which the presence of motion boundaries can be inferred via these global extrema. First, a boundary can be inferred where the first derivative of the *peak-ratio*, for example, is zero, its second derivative is negative, and where this ratio is above some minimal threshold. This minimal threshold is chosen so that any extremum below it can be safely excluded.

• A boundary is inferred where the first derivative of a measure is zero, its second derivative is of the appropriate sign, and where a ratio measure is above/below some threshold.

Second, the measures have in common that they have a global extremum at a motion boundary, and that their local extrema anywhere else in the image are weakly correlated with each other. Hence, the extrema contours can be used in the following way to locate the motion boundaries, without having to use any thresholding. First, the extrema contours are computed by differentiation. These contours are then thickened by some number of pixels because the extrema of the different measures are not perfectly localized and can be shifted with respect to each other at a motion boundary. Finally, these thickened contours are superimposed, and a motion boundary is inferred where they all intersect *(see Figure 6.2)*. This approach of combining the extrema contours to detect boundaries has the attractive feature that it does not require the setting of a threshold. The motion boundaries are inferred by corroborating the information provided by the different measures, and good results have been obtained.

• Combining and intersecting the thickened extrema contours to detect the boundaries has the attractive feature that it does not require the setting of a threshold.

3.4.2.1 Hysteresis

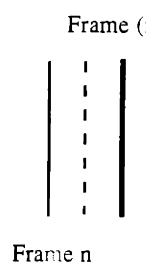
The problem with setting a fixed threshold is that it can cause the detected boundaries to streak. Streaking occurs when the *peak-ratio*, for example, fluctuates above and below the threshold of our choice along a motion boundary. To reduce the likelihood of streaking, the thresholding approach could be improved by using hysteresis [8]. We could use two thresholds, a high and a low one. A high threshold of 0.9 is chosen so to ensure that any point on a local maxima contour of the *peak-ratio* above this threshold is with a high probability a motion boundary.

A low threshold of 0.6 is chosen so that the probability is low that a motion boundary is missed. If any point of a local maxima contour is above the high threshold, then that point is immediately accepted, as is the entire connected segment of the contour which contains the point and lies above the low threshold. The likelihood of streaking could thereby be greatly reduced, because for a contour to be broken it must now fluctuate above the high and below the low threshold. Also the probability that false motion boundaries are marked is reduced because the high threshold can be raised without risking streaking. If streaking still occurs then these gaps can be filled by the methods introduced in Chapter 5.

3.4.3 Localization

The localization of a motion boundary is affected, firstly, by the curvature of the boundary with respect to the size of the neighborhood used to compute the local histograms; corners, for example, will get rounded. Secondly, regions occluded in the next frame will cause the estimated boundary to lie midway between the location of the actual object boundary in the first frame and its location in the next frame (as shown in derivation for the thresholds). This is because the occluded matching primitives will not have a match in the next frame and only midway between the locations of the actual object boundary in the first and second frame are the consistent contributions from the two sides of the boundary roughly equal. The detected motion boundary should however coincide with the actual boundary, if a region appears next to it in the subsequent frame, because the matching primitives on either side will have a match in the next frame.

• Corners get rounded and the estimated boundary can lie midway between the locations of the actual object boundary in the first and second frame respectively.



3.4.3.1 Figure-Ground Separation

The fact that the estimated boundary can lie midway between the actual object boundary in the first and second frame could be used to infer the side of a motion boundary that corresponds to the occluding object. If the order of the frames is reversed then the regions, which disappeared previously, will come into view now, and the estimated boundary will be correctly localized there. Similarly, the estimated motion boundaries, where previously regions came into view, will now be shifted in the

direction of the relative motion between the occluding object and the background. Hence, we can compute to which side a boundary has moved by comparing where the estimated boundary happens to lie with respect to the boundary that was estimated by reversing the order of the frames. We refer to this displacement of boundary as v_B. We have to consider the velocities on the two sides of a boundary, in order to be able to infer which side of the motion boundary is closer to the viewer. As will be outlined in the Chapter 4, the highest peak in the local histograms of the potential displacements estimates the image flow at each point. Now, the occluding object will move in the same direction as the motion boundary, i.e. the scalar product of their flow vectors has to be positive. Hence, if the scalar product of v_B and the difference vector between the velocity to the right, v_R, and to the left of the boundary, v_L, is positive, i.e. $v_B.(v_R - v_L) > 0$, then the occluding object is to the right of the detected boundary. Similarly, a negative scalar product implies that the side to the left of the detected motion boundary is closer to the viewer. If there is no dynamic occlusion occurring, then v_B will be zero, and the local inference of which side corresponds to the occluding object becomes difficult.

3.5 The Dynamic Occlusion Method

In this section, we show how dynamic occlusion can be used to estimate motion boundaries at a stage prior to the computation of visual motion. Specifically, we want to develop a method that can locally compute the appearance and disappearance of simple features in a way that is sufficient to estimate boundaries, without having to solve a global and difficult correspondence problem.

3.5.1 Dynamic Occlusion of Thin-Bars

Certain spatial relationships between simple image features change most dramatically in the vicinity of a boundary in the presence of motion. In particular, zero-crossings[5] of opposite contrast will move closer together or farther apart. They may even disappear or come into view. Hence, pairs of zero-crossings of opposite contrast will be created or destroyed in the vicinity of a boundary. We will refer to these pairs as thin-bars because they can correspond to thin bars of constant intensity in the image. We define a pair of zero-crossings of opposite contrast to constitute a thin-bar if they are separated by less than 3 sigma, where sigma refers to the spread of the Gaussian used to smooth the image. The appearance or disappearance of the thin-bars can be used to construct a method that locally estimates motion boundaries.

Definition
- A *thin-bar* is a pair of zero-crossings of opposite sign separated by less than 3 sigma, where sigma refers to the spread of the Gaussian used to smooth the image.

- Thin-bars consisting of close pairs of zero-crossings of opposite contrast are created or destroyed at a motion boundary.

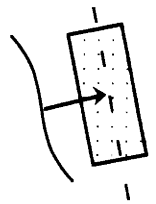
When tracking a thin-bar, we do not attempt to solve completely the correspondence problem since we will only check for the existence of a matching thin-bar, instead of trying to determine the correct and unique match. The disappearance of a thin-bar will only be concluded if no corresponding thin-bar can be found in the next frame that satisfies the constraints outlined below. As *Figure 6.5* shows, this is sufficient to estimate motion boundaries. The appearance of a thin-bar is detected by using the fact that the appearance of a thin-bar is equivalent to the disappearance of a thin-bar, when the order of the frames is reversed.

- We do not attempt to solve the correspondence problem since we only check for the existence of a matching thin-bar.

[5] Zero-crossings correspond to sharp changes in intensity detected by filtering the image with the Laplacian of a Gaussian.

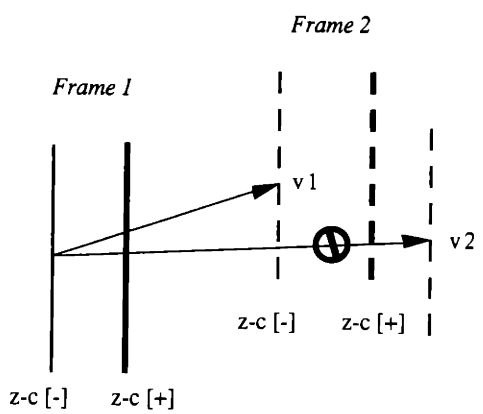
A matching thin-bar has to satisfy the following constraints. First, corresponding zero-crossings must have the same contrast, i.e., the scalar product of the intensity gradients at the locations of the zero-crossings must be positive. Similarly, the angle between the intensity gradients at the locations of corresponding zero-crossings should be within a certain bound for small rotations. Second, the direction of the measured normal flow component constrains the motion of a zero-crossing within 180°. More specifically, the direction and magnitude of the measured normal flow component defines a band within which the matching thin-bar has to lie *(see Figure 3.3)*. The dimensions of the band are chosen to account for measurement errors. This constraint reduces greatly the number of potentially matching thin-bars. Third, we can define a spatial ordering for a thin bar, since either the first zero-crossing will be to the right or left of the second zero-crossing, and vica versa. This spatial relationship or ordering is not likely to change as a thin-bar moves, because the two partners are spatially close and their flow vectors are therefore roughly equal *(see Figure 3.9)*.

Matching Constraints
Corresponding zero-crossings have:
- Same contrast
- Match lies on the line defined by normal flow component.



- Spatial ordering preserved.

Figure 3.9 **The Spatial Ordering Constraint.**
If the zero-crossing with a negative contrast, z-c [-], moves with v_1 then the spatial ordering between the two zero-crossings of opposite contrast will remain intact in Frame 2. But if it were to move with v_2, then the spatial ordering between z-c [-] and z-c [+] would be violated.



The *Dynamic Occlusion Method* requires that the image be finely textured, because otherwise thin-bars will appear or disappear only in a few places. Furthermore, this method can have false alarms, when a surface rotates in depth or, for perspective projection, when a plane moves towards or away from the viewer. This will cause the distance between zero-crossings to increase or decrease, and it can thereby accidently create or destroy thin-bars. In the case of a rotating cylinder, dynamic occlusion and effects due to rotation in depth are confounded, but the thin-bars are still being created or destroyed only in the vicinity of the boundary of the cylinder. Despite these shortcomings, the reason for developing this method has been to show that the dynamic occlusion of these simple features can be computed locally in a way that is sufficient to estimate boundaries at a stage prior to the computation of visual motion, without having to solve a global correspondence problem (for results see Chapter 6).

CHAPTER 4

THE LOCAL ESTIMATION OF VISUAL MOTION

In this chapter we show how visual motion can be locally estimated as a by-product of the early estimation of motion boundaries. The local histograms of the potential displacements can be used to compute a dense image flow field, because the local histograms have their highest peak at the displacement that received the most local support. Hence, this displacement represents an estimate of the image flow. Furthermore, the ratio of the two highest peaks or "strongest contenders" reflects how good the estimate is. A low *peak-ratio* implies a good estimate, whereas a *peak-ratio* close to one implies the presence of a motion boundary and, likewise, that the estimated image flow might be inaccurate.

- The highest peak in a local histogram corresponds to the displacement with the most local support. Hence, this displacement represents an estimate of the image flow.
- The estimation is well-posed and consistent with human psychophysics.

It is worth noting the following: firstly, the estimated motion boundaries are not incorporated in the computation of visual motion discussed here. These early estimates of the image flow field and its discontinuities could then be integrated in a later computation. Secondly, the local estimation of visual motion will be difficult in image regions with only little texture, as is the case for the early estimation of motion boundaries.

Local support or voting schemes have been used by, for example, Stevens (1977) [39], Fennema & Thompson (1979) [12], Prazdny (1984) [33], Bandopadhay & Dutta (1986) [4] and Bülthoff, Little & Poggio (1989) [7] to compute disparity and displacements fields. These methods, however, do not compute and analyze the full histogram of the possible displacements to detect the presence of boundaries.

In this chapter we show that the method proposed in this thesis for computing visual motion, using the local histograms of the potential displacements, is well-posed. Furthermore, we show that the proposed method is similar to the *local voting scheme* developed by Bülthoff, Little & Poggio. The two methods might appear to be different because of nomenclature and more importantly because of what their main goal is.

Bülthoff et al. are primarily interested in estimating the image flow field and assume that motion boundaries should be detected a later stage, whereas we are primarily interested in demonstrating that the detection of motion boundaries can be decoupled from the computation of the image flow field and that it can be performed using no intensity boundary and only motion information.

Both methods assume that the image flow field can be approximated locally as constant and both use a small circular neighborhood at each point to determine the displacement with the most votes. The main difference is that the votes for each possible displacement is recorded in a local histogram by our method, whereas Bülthoff et al. are only interested in the displacement with the most votes. Hence, our method computes a more general representation, which can be used to detect motion boundaries and estimate visual motion in parallel. Another difference lies in the comparison function used to determine the pointwise match between intensities in subsequent frames.

4.1 Mathematical Formulation

The computation of the visual flow field is locally underconstrained and in order to make it well-posed we need to add a constraint to compute the smoothest flow field which matches the data [7]. When the projected motion of objects is small relative to the image size, we can restrict the search for corresponding points to small regions in the image. Using a formulation similar to the one used by Bülthoff et al. [7], we look for a discrete image flow field $V(x,y) = (u(x,y),v(x,y)) \in (-/+\mu,-/+\mu)$ to minimize:

$$\int [\Omega(E_t(x,y),\ E_{t+\Delta t}(x+u\Delta t,y+v\Delta t)) + \beta\ (d^2u/dx^2 + d^2u/dy^2 + d^2v/dx^2 + d^2v/dy^2)]\ dx\ dy \qquad (1)$$

where $E_t(x,y)$ denotes the image brightness or intensity at (x,y) at time t, Ω is a comparison function which measures the pointwise match between subsequent frames, and μ denotes the maximal expected displacement in the x and/or y dimension.

We construct the image flow field pointwise, since for each displacement, every point evaluates a comparison function Ω at that displacement, and it then sums the match scores over the circular neighborhood C_r. Each point chooses the displacement with maximal support out of the finite set of possible displacements. The resulting image flow field is the union of these pointwise displacements.

We simplify and approximate equation (1) by using the constraint that the image flow field can be assumed to be *locally constant* in the small neighborhood C used at each point to compute the local support for the different possible displacements. We choose the neighborhood C_r to be circular with a radius r that is dependent on the distance to the objects in the scene and their expected size in the image. The choice of μ depends on the maximal expected velocities of objects in the scene, their distances from the camera, and the time separation Δt between frames. The time separation Δt is small and therefore the resulting image displacements will be small with respect to the image size. Hence, we are dealing with short range motion.

The second-order term of equation (1) vanishes, because of the local translation assumption. The simplified and approximated equation (1) minimizes now, in each overlapping circular neighborhood $C_r(x,y)$ with radius r :

$$\sum_{(x,y) \in C_r} \Omega\big(E_t(x,y),\, E_{t+\Delta t}(x+u\Delta t, y+v\Delta t)\big). \qquad (2)$$

As mentioned in section 3.1.1, we use a *Gaussian matching function,* which depends on the difference in intensity to measure the pointwise match between subsequent frames to account for the occurring changes in intensity. The smaller the difference in intensity, the larger the weight

that is assigned to a particular displacement. The spread β of the *Gaussian matching function* can be chosen to reflect the estimated noise in the intensity measurements. Hence, in our case the comparison function Ω is equal to:

$$\Omega(E_t(x,y),\, E_{t+\Delta t}(x+u\Delta t, y+v\Delta t)) = -\, e^{-\beta\,(E_t(x,y) - E_{t+\Delta t}(x+u\Delta t, y+v\Delta t))} \qquad (3)$$

whereas Bülthoff et al. use $\Omega(E_t(x,y),\ E_{t+\Delta t}(x+u\Delta t,y+v\Delta t)) = (E_t(x,y) - E_{t+\Delta t}(x+u\Delta t,y+v\Delta t))^2$.

We can substitute equation (3) into equation (2) and absorb the minus sign by turning the minimization into a maximization. Hence, the visual flow vector of a pixel is computed by maximizing for all $(u,v) \in (-/+\mu, -/+\mu)$:

$$\sum_{(x,y) \in C_r} e^{-\beta (E_t(x,y) - E_{t+\Delta t}(x+u\Delta t,y+v\Delta t))^2} \qquad (4)$$

The local neighborhoods used to estimate the image flow field are overlapping from pixel to pixel. Each pixel, surrounded by its neighborhood C_r with radius r, independently chooses the image flow vector to maximize matching in its neighborhood. We do not match intensities directly, since the presence of noise makes the process unstable. We rather choose the displacement whose intensity value maximizes (4), which in turn regularizes the solution of the matching computation [7].

4.2 Advantages and Relationship to Human Psychophysics

Like the method by Bülthoff et al. [7], this way of estimating the image flow field has several attractive features. First, noise is reduced by the local neighborhoods used to find the displacement with the most local support. Second, it does not rely on the numerical precision of derivatives, making it therefore more robust. Third, this approach computes a dense image flow field, removing the necessity of interpolating or smoothing the estimated flow field.

Bülthoff et al. [7] have demonstrated that the approach of using local neighborhoods to find the displacement with the most local support is consistent with human psychophysics, since it exhibits several of the same "illusions" that humans perceive, such as the "barbepole-", the "non-rigidity-", the "motion-capture-" and the "Wallach's aperture-illusion".

CHAPTER 5

EXTRACTING COMPLETE AND UNIQUE CONTOURS

5.1 Introduction

The pointwise output of the motion boundary estimators is often broadly localized and it can contain gaps. Hence, we have to find a way to extract single and unique boundaries without gaps. We apply and modify the *Structural Saliency Method* developed by Sha'ashua & Ullman [37,45] to achieve this goal.

Sha'ashua and Ullman have proposed two different kinds of saliency measures: *local saliency* and *structural saliency*. An edge's local saliency is determined by attributes of that edge alone, and in our case local saliency is equal to the magnitude of the output of the motion boundary estimators. Structural saliency refers to more global properties of an edge - its relationships with other edges - and often this saliency is a property of the structure as a whole, whereas the parts of the structure are not necessarily salient in isolation.

5.2 The Structural Saliency Method

The Structural Saliency Method employs a simple iterative network and uses an optimization approach to produce a "saliency map", which emphasizes salient locations in the image. The saliency of curves is measured in terms of their smoothness and length, which is often sufficient to perform a figure-ground separation. The main properties of the network are: (i) the computations are simple and local, (ii) globally salient structures emerge with a small number of iterations, (iii) there is little dependence on the complexity of the image, (iv) contours are smoothed, gaps are filled in and linking information between edge segments is provided.

How does it work?
- It extracts complete boundaries by computing their saliency in terms of their smoothness and length.
- The optimization is linear in terms of the length of the contour because an extensible functional is used.

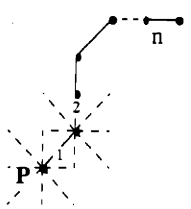
5.2.1 Detailed Description

A structural saliency measure Ω is computed by a locally connected network of processing elements. The image is represented by a network of $n \times n$ grid points, where each point represents a specific (x, y) location in the image. At each point P there are k orientation elements coming into P from neighboring points, and the same number of orientation elements leaving P to nearby points (in the current implementation k is equal to 16, providing a reasonable angular resolution). Each orientation element p_i responds to the output of the motion boundary estimators by signalling the presence of the corresponding motion boundary in the image, so that those elements that do not have an underlying line segment are associated with an empty area or gap in the image. We refer to a connected sequence of orientation elements $p_{i+1}, \ldots, p_{i+n}$, each element representing a line segment or a gap (called a virtual element), as a curve of length n. The optimization problem is formulated as maximizing $\Omega(n)$ over all curves of length n starting from p_i.

An exhaustive enumeration of all combinations of $p_{i+1}, \ldots, p_{i+n}$ would require an exponential search space of size k^n for each element in the network. The computation becomes linear in n if we use an *extensible function* Ω to measure saliency:

$$\max_{\delta^n(p_i)} \Omega_n(p_i, \ldots, p_{i+n}) = \max_{p_{i+1} \in \delta^1(p_i)} \Omega_1(p_i, \max_{\delta^{n-1}(p_{i+1})} \Omega_{n-1}(p_{i+1}, \ldots, p_{i+n}))$$

where $\delta^n(p_i)$ is the set of all possible curves of length n starting from p_i.



Hence, the maximal curve of length n at P will be equal to the maxima over all possible segments leaving P and the maximal curves of length $(n-1)$ starting at the respective end-points of these segments.

It is worth noting that the optimal contour through P does not necessarily extend itself as the iterations proceed. In fact, the optimal curve at stage $n+1$ can be different from the optimal curve at stage n. Further, the saliency measure is associated with each element, not with the entire curve.

The structural saliency E_i is equal to the weighted contributions of the local saliency values along the curve. Each weight is a product of two factors. The first factor is inversely related to the number of virtual elements (i.e. gaps) along $p_i, \ldots, p_j$, and the second factor is inversely related to the total curvature of the curve. Curves that have a high structural saliency value are long curves that are as straight as possible and have the least number of gaps (for an in-depth description, see Sha'ashua 1988 and Sha'ashua & Ullman 1989 [36,37,45]).

- Structural saliency is equal to the weighted contributions of the local saliency values along the curve.

The structural saliency E_i is updated by the following computation:

$$E_i^{(0)} = \sigma_i$$

$$E_i^{(n+1)} = \sigma_i + \rho_i \max_{p_j \in \delta(p_i)} E_j^{(n)} f_{i,j}$$

and it can be shown by induction on the length of the curve that

$$E_i^{(n)} = \sum_{j=i}^{i+n} C_{i,j} \rho_{i,j} \sigma_j$$

where

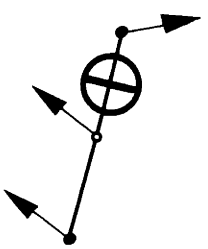
$$C_{i,j} = \prod_{k=i}^{j-1} f_{k,k+1}, \text{ where } f_{k,k+1} = e^{-\frac{2a_k \tan\frac{a_k}{2}}{\Delta s}}$$

and

$$\rho_{i,j} = \prod_{k=i+1}^{j} \rho_k, \text{ where } \rho_k = \begin{cases} 1 \text{ if } p_k \text{ is active} \\ <1 \text{ if } p_k \text{ is virtual} \end{cases}.$$

5.2.2 Extending the Structural Saliency Method

We incorporate the motion estimates to separate boundary segments belonging to differently moving objects. The three points that constitute an oriented segment have each a motion estimate associated with them. We allow only points to form an oriented segment whose motion estimates do not differ by more than two displacement units. We want to prevent contours from being formed that wander across motion boundaries, and thereby violate the constraint that a flow field varies smoothly along a boundary. The effectiveness of this constraint hinges on how well the qualitative aspects of the motion field are estimated.



5.2.3 Extracting a Unique Contour

If the area in which curves are allowed to form is broadly defined, then there will be several contours growing alongside each other, as is the case in our examples. To extract the most salient curve, we have to first propagate the structural saliency value of the most salient segment along the curve that contributed to its value, because the saliency measure is associated with each element and not with the entire curve. The propagation is done iteratively by each segment maximizing over the value of its preferred neighbor and its own [Sha'ashua in prep.]. Thus, the largest value will be propagated along its curve.

Finally, we perform a non-maximal suppression operation [Sha'ashua in prep.], where each segment suppresses all its neighboring segments if their structural saliency value is less and if they have similar motion estimates associated with them. Hence, the most salient contours belonging to differently moving objects will remain alongside each other[6] *(see Figure 6.7)*.

[6] At the locations where the differently moving objects occlude each other, there will be two boundary segments extracted that lie alongside each other, but where one of them is an artifact of the occlusion. A next step could be to label the boundary segments that lie alongside each other so that they receive a lower priority than boundary segments that do not have a boundary segment belonging to another object close by, when the extracted boundaries are the input to a recognition process.

CHAPTER 6
RESULTS

In this chapter, we present results, where the developed methods have been applied to motion sequences containing several moving objects composed of either random-dot or natural textures. The methods for estimating the motion boundaries have been implemented on the Connection Machine, a massively parallel network of simple, locally interconnected processors [16]. The smoothed intensity values are used as the matching primitives and the histograms of the potential displacements are used as the input representation.

6.1 The Estimation of Motion Boundaries

The early detection of motion boundaries is performed in two stages: (i) the local estimation of the motion discontinuities; (ii) the extraction of complete boundaries belonging to differently moving objects.

The methods for estimating the motion boundaries make use of the fact that the potential displacements of image points in the vicinity of a motion boundary will cluster around two different points in a local velocity histogram. The local histograms are constructed at every point using a circular neighborhood with a radius of eight pixels. The potential displacements are quantized and they are measured in terms of pixels.

6.1.1 The Bimodality Tests and the Bi-distribution Test

The *Bimodality Tests*, consisting of the *peak-ratio*, *local-support-ratio*, *signal-noise-ratio* and the *chi-square measure*, estimate motion boundaries by computing the degree of bimodality present in the local histograms of the potential displacements. The *Bi-distribution Test* detects boundaries by applying the *Kolmogorov-Smirnov Test* to measure the probability that two histograms have been created by the same population of motions.

6.1.1.1 Complex Dynamic Random-Dot Display

Figure 6.1 shows the estimated boundaries for a complex random-dot motion sequence which contains a rotating circle and rectangle, and a translating square in the image plane. The first row displays the estimated boundaries using thresholding. The second row displays the inferred boundaries by detecting the global extrema and using a minimal threshold.

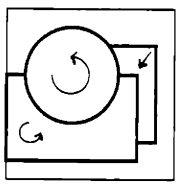
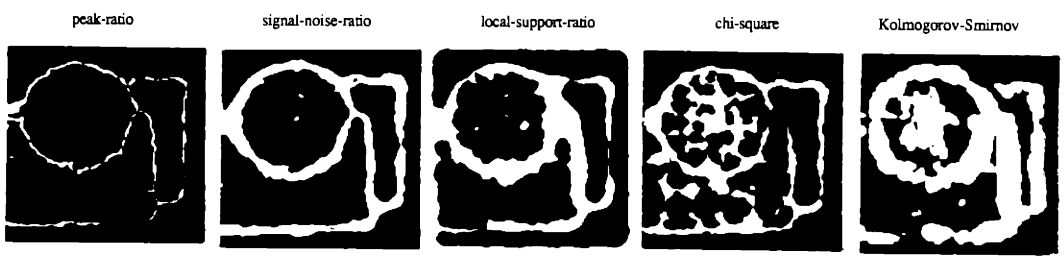
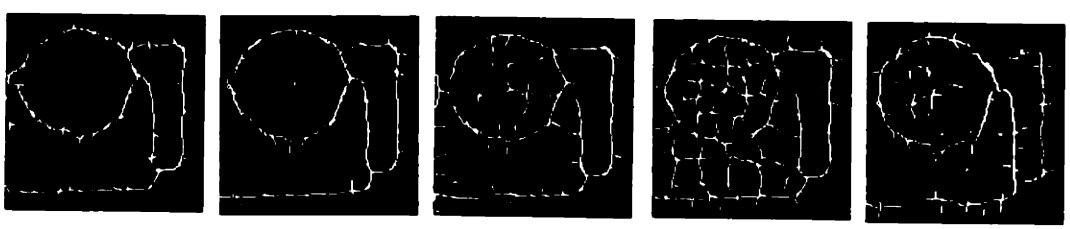


Figure 6.1 **Estimating Motion Boundaries in a Complex Dynamic Random-Dot Display.**

Thresholding



Detecting Global Extrema

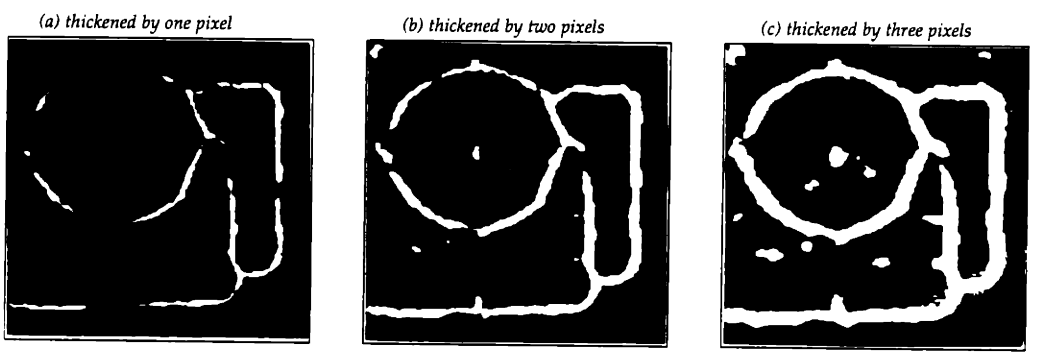


For the above example, the *peak-ratio* and the *signal-noise-ratio* successfully estimate all the motion boundaries, and they mark very few false boundaries. The reason these two measures perform so well is that they directly measure the degree of bimodality occurring in the local histograms, whereas the other measures do it indirectly. The *local-support-ratio*, the *chi-square measure* and the *Kolmogorov-Smirnov measure* also successfully infer where motion boundaries are present, but they mark more incorrect boundaries. The *chi-square measure* has a high false alarm rate inside the two rotating objects, because the highest peak is

broadly defined at the borders between regions of constant displacement that differ only by one displacement unit. The *Kolmogorov-Smirnov measure* has a high false-rate at the center of the rotating circle, because any two histograms that are being compared will have their peaks at different locations.

The false alarms can be ruled out by overlapping the thickened extrema contours of several of the measures, because these measures have a global extrema at a motion boundary, whereas their local extrema elsewhere in the image are weakly correlated with each other. Figure 6.2 shows the results of intersecting the thickened extrema contours of the *peak-ratio*, *signal-noise-ratio* and *local-support-ratio* to infer the motion boundaries. (a), (b) and (c) display the intersections of the extrema contours thickened by one, two and three pixels, respectively. This approach has the attractive feature that it does not require the setting of a threshold and it can be used to rule out false alarms. Figure 6.2 demonstrates that the measures are highly correlated at a motion boundary, whereas elsewhere in the image they are weakly correlated with each other.

Figure 6.2 **Intersecting the Extrema Contours of the Developed Measures to Estimate Motion Boundaries.**

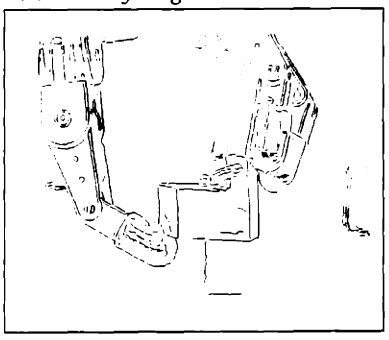


6.1.1.2 Natural Motion Sequence

Figure 6.3 (a) shows the Canny edges of the Salisbury Robot Hand; (b) displays the estimated motion boundaries when the hand is lifting the object that it is holding, where the *peak-ratio* has been thresholded and its output has been suppressed where the average intensity gradient was not sufficiently large; (c) shows the detected global maxima of the *peak-ratio*.

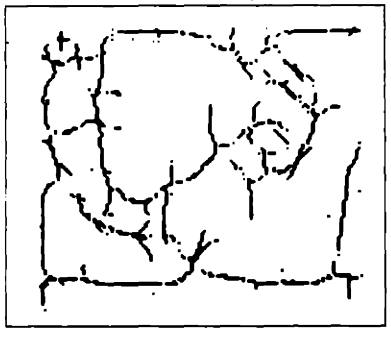
Figure 6.3. **Estimating Motion Boundaries in a Natural Image Sequence.**

(a) Canny Edges



(b) Peak-ratio thresholded

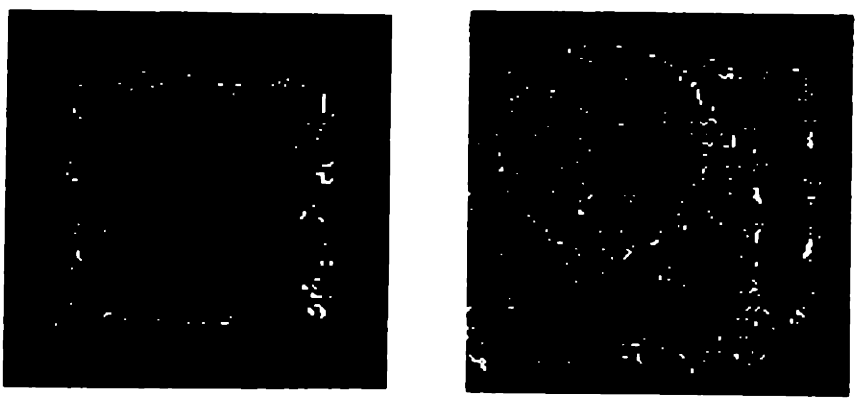
(c) Global maxima of peak-ratio



6.1.2 Dynamic Occlusion Method

Figure 6.4 (a) shows where the Dynamic Occlusion Method estimated the appearance or disappearance of *thin-bars* in a random-dot display of a translating square. This method gives a rough sense of the motion boundary, although it does not provide complete boundaries. (b) Shows the output of this method for the same motion display as in Figure 6.1. The marked locations provide a sense of the boundaries for this more complex display, although there are false alarms in the rotating regions.

Figure 6.4 **Estimating Motion Boundaries using the *Dynamic Occlusion Method*.**



(a) (b)

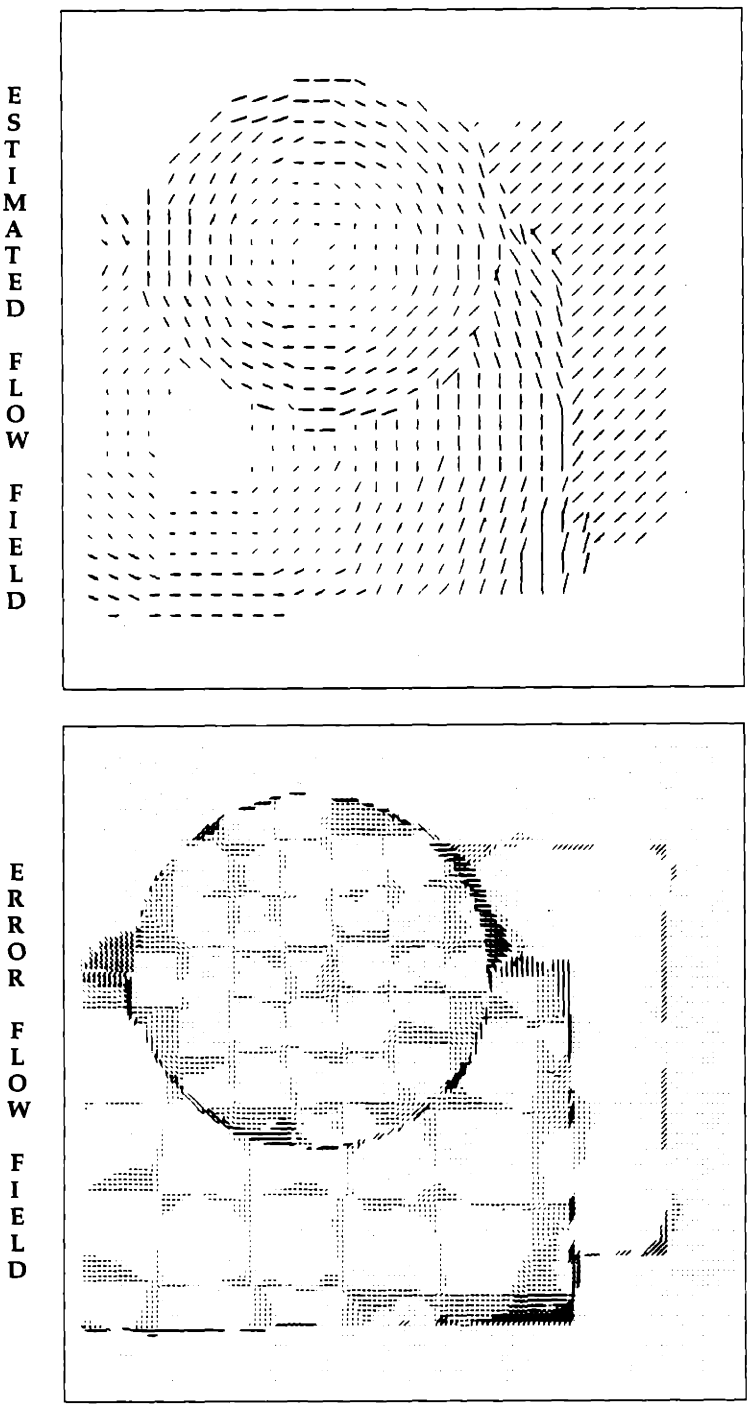
6.2 The Estimation of Visual Motion

A local histogram of the potential displacements has its highest peak at the displacement that received the most local support. Hence, this displacement represents an estimate of the image flow.

6.2.1. Complex Dynamic Random-Dot Display

The first panel in Figure 6.5 shows the estimated image flow field for a complex random-dot motion sequence which contains a rotating circle and rectangle, and a translating square. The second panel displays the error in the computed flow field. As expected, the error is largest in the vicinity of the motion boundaries. In the interior of the rotating objects, there are also small errors at the borders between the regions of constant displacement that differ only by one displacement unit.

Figure 6.5 **Estimating the Image Flow Field.**
The first panel shows the estimated image flow field for a complex random-dot motion sequence which contains a rotating circle and rectangle, and a translating square. The second panel displays the error in the computed flow field. As expected, the error is largest in the vicinity of the motion boundaries. In the interior of the rotating objects, there are also small errors at the borders between the regions of constant displacement that differ only by one displacement unit.



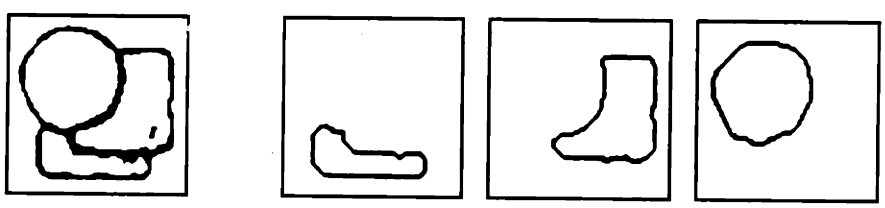
6.3 Extracting Complete & Unique Motion Boundaries

The pointwise output of the motion boundary estimators is often broadly localized and it can contain gaps. We apply and modify the *Structural Saliency Method* developed by Sha'ashua & Ullman to extract single and unique boundaries without gaps.

6.3.1. Complex Dynamic Random-Dot Display

Figure 6.6 (a) shows the estimated motion boundaries for a random-dot motion sequence which contains a translating circle, rectangle and square, where the *peak-ratio* has been used and thresholded to provide the estimate. (b) Displays the three most salient structures extracted by the *Structural Saliency Method*, where the motion estimates are used to ensure that the contours do not wander across motion boundaries. The purpose of the second stage is to extract complete boundaries from an input that can be noisy.

Figure 6.6 **Extracting Complete & Unique Motion Boundaries.**



Input
Estimated motion boundaries

Output
Connected contours belonging to differently moving objects

CHAPTER 7
APPLICATIONS

7.1 Stereopsis

Stereopsis computes relative depth by using the differences, also called disparities, in the projection of points in space onto the two eyes or cameras, which view the scene from two slightly different vantage points. The key problem of stereopsis is how to match points in the two images that correspond to the same point in space. This correspondence problem is inherently underdetermined and constraints are needed to solve it. As for the computation of the image flow field, the assumption is typically made that the surfaces of objects are generally smooth, i.e., that the disparity varies smoothly almost everywhere in the image. This constraint is not valid across depth boundaries, and so far, most stereo algorithms not only do not directly detect discontinuities in depth but also perform badly precisely at these locations [10,46]. The methods developed for the early detection of motion boundaries are relevant to stereopsis in the following ways.

- The disparity fields of stereopsis are equivalent to image flow fields created by a restricted class of motions.

- Advantageous to detect depth or motion boundaries prior to the stereo computation, because they make explicit where the smoothness assumption is not valid, and they could be used to simplify the correspondence problem.

First, stereopsis is a special case of general motion, because its disparity fields are equivalent to image flow fields created by a restricted class of motions and all the motion boundaries are due to depth discontinuities. Hence, these depth boundaries can be detected by the methods developed for general motion at a stage prior to the depth computation, where the two images do not need to be registered.

Second, motion boundaries can be used as stereo matching features and there is psychological evidence that the human visual system is able to do this [21,22,32]. The motion boundaries can be matched using the ordering constraint, i.e., if a motion discontinuity is to the left of another motion discontinuity in the left image then this ordering will be

preserved in the right image, and vice versa. Further, the figural continuity and edge connectivity constraints can be applied, because the motion boundaries will form continuous contours [5,26].

Third, the detected motion and depth boundaries can be used as pointers to the regions in the two images that do not possess a match in the other image due to occlusion. In particular, these occluded regions will always be to the right (left) of a depth discontinuity in the left (right) image, for perspective projection. A search could be performed in the neighborhood of a detected motion or depth boundary to determine the extent of an occluded region. Finally, the corresponding points, that are visible in both eyes, could be then matched using the ordering constraint, thereby simplifying the correspondence problem.

Fourth, a stereo algorithm can be devised that simultaneously computes depth and its discontinuities, because the highest peak in the local histogram of the potential disparities estimates the disparity, and the depth boundaries can be inferred where the *peak-ratio*, for example, is close to one.

To summarize, it is advantageous to detect depth or motion boundaries prior to and use them in the stereo computation, because they make explicit where the smoothness assumption is not valid, and they could be used to simplify the correspondence problem.

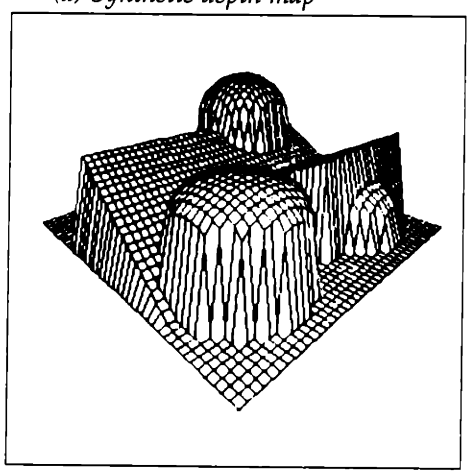
7.2 Surface Reconstruction

In most models of stereopsis, disparity is initially computed at specific locations, such as where intensity changes sharply. The surface reconstruction from this sparse and noisy data can be formulated in terms of minimizing an energy functional [14,20,40]. In particular, the surface reconstruction should be performed as a piecewise smooth interpolation to account for the existence of several surfaces within a scene. Without the knowledge of the locations of depth discontinuities, the information about the shape of one surface can affect the shape of an adjacent surface, i.e., the surface reconstruction scheme will smooth over the boundaries.

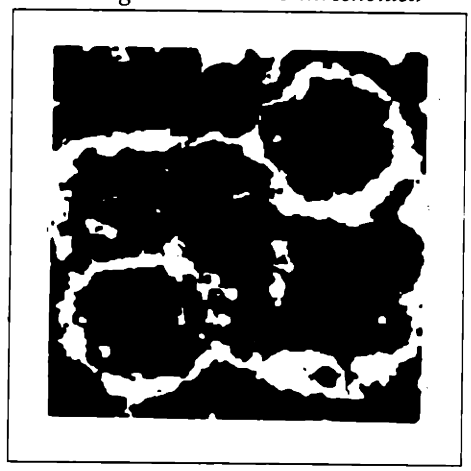
Hence, the early detection of depth boundaries is of special importance because it makes explicit where not to smoothly interpolate the sparse depth map. The methods described in this thesis can be used to segment sparse and noisy depth maps *(see Figure 7.1)*.

Figure 7.1 **Detecting Boundaries in a Sparse Depth Map.**
(a) Shows the synthetic depth map used as the test input. The depth map has a depth range of 200 units. The resolution is reduced by a factor of 10, because the depth gradient is too large and changes too rapidly over the spatial support used to construct the local histograms of the depth estimates. (b) Displays the depth boundaries detected by thresholding the *signal-noise-ratio*, where the sparseness of the data is 10% and Gaussian noise has been added.

(a) Synthetic depth map



(b) Signal-noise-ratio thresholded



CHAPTER 8
SUMMARY & CONCLUSION

This thesis has shown, firstly, that a useful segmentation can be performed on the basis of motion information alone at an early stage of visual processing. Secondly, it has been demonstrated that the estimation of motion boundaries can be decoupled from the computation of a full image flow field and how it can be performed in parallel. Thirdly, this thesis has shown how to integrate the pointwise output of the developed motion boundary estimators with a process that can extract salient, complete and unique contours, where contour segments belonging to differently moving objects are separated and segments belonging to the same object are grouped together. The detection of motion boundaries has been performed in two stages: (i) the local estimation of the motion discontinuities and of the visual flow field; (ii) the extraction of complete boundaries belonging to differently moving objects.

8.1 The First Stage

For the first stage, three new methods have been presented that can independently estimate the presence and location of motion boundaries: the *Bimodality Tests*, the *Bi-distribution Test*, and the *Dynamic Occlusion Method*. These methods require only local computations. They have been implemented on the Connection Machine, a parallel network of simple, locally interconnected processors.

The *Bimodality Tests* and the *Bi-distribution Test* make use of the fact that at a motion boundary certain quantities, which can be easily computed from an image sequence, will cluster around two different points in a local histogram. The quantities in question are (i) the potential displacements of an image point, or (ii) the flow component measured in the direction of the intensity gradient. The local histograms are constructed at every point using a circular support, whose radius ranges between five and eight pixels.

We use a *Gaussian matching function*, which depends on the difference in intensity at the two points defining a displacement, to compute the match score of a possible displacement. This matching function has been chosen to account for the fact that the intensity values at corresponding points can change due to noise and changes in illumination. Further, we use the magnitude of the intensity gradient or its local average to suppress false alarms in regions with little texture.

We assume that the image flow field can be approximated as *locally constant*. Hence neighboring points will have a potential displacement in common. We can relax this assumption by using an *Gaussian spatial support function* that weighs contributors less that are farther away from the point at which the histogram is computed. This will account for the fact that the flow vectors at points farther apart are less likely to be equal in a smoothly varying flow field. It will also cause the response of the *Ratio measures* to be sharpened.

The *Bimodality Tests* consist of four measures that monitor the degree of bimodality present in the local histograms of either the potential displacements or the normal flow components. The *peak-ratio*, the *local-support-ratio* and the *signal-noise-ratio* can be computed from the local histograms directly, and each of them captures a different characteristic of a motion boundary. The *chi-square measure* estimates bimodality by measuring how well a Gaussian distribution can be fitted to a local histogram. Of these four measures, the *peak-ratio* and *signal-noise-ratio* estimate motion boundaries most accurately and reliably, because they directly measure the degree of bimodality present in the local histograms. It was also found that more than one of these measures can be combined to detect boundaries and to rule out false alarms by intersecting the thickened extrema contours of several of these measures.

The *Bi-distribution Test*, which uses the non-parametric statistical *Kolmogorov-Smirnov* test, can compare any two distributions. But this method often does not perform as well as the *Bimodality Tests*, because the local histograms used in the detection of motion boundaries can be sufficiently different even for nearby points belonging to the same moving object.

The reason why we have developed five different measures is because they each capture and monitor a different characteristic of a motion boundary. We have shown that these measures have a global extrema at a motion boundary, whereas their local extrema elsewhere in the image are weakly correlated with each other. Thresholds have been derived for the different measures, and we have shown how to use thresholding and the detection of global extrema as ways to infer the presence of motion boundaries. In particular, the approach that combines and intersects the thickened extrema contours to estimate the boundaries has the attractive feature that it does not require the setting of a threshold. The motion boundaries are inferred by corroborating the information provided by these measures, and good results have been obtained.

The *Dynamic Occlusion Method* uses the fact that *thin-bars* are created or destroyed at a motion boundary. Dynamic occlusion of these simple features can be computed locally in a way that can estimate boundaries prior to the computation of motion without having to solve global correspondence.

It has also been shown that the visual flow field can be locally estimated as a by-product of the early estimation of motion boundaries. The highest peak in a local histogram of the potential displacements estimates the local image flow. The measures that are sensitive to degree of bimodality present in the local histograms reflect how good the estimate is. It was noted that the developed method to compute visual motion is well-posed and that it is similar to the *local voting scheme* proposed by Bülthoff, Little & Poggio [7].

8.2 The Second Stage

We have applied and modified the *Structural Saliency Method* developed by Sha'ashua & Ullman [37,45] to extract complete and unique boundaries from the pointwise output of the first stage, which is often broadly defined and can contain gaps. Boundary segments belonging to differently moving objects have been separated by using the motion estimates provided by the first stage to constrain which edge segments can be formed.

The *Structural Saliency Method* extracts boundaries and closes gaps by employing a simple iterative scheme that uses an optimization approach to measure the saliency of curves of line segments in terms of their smoothness and length. The optimization problem is formulated in terms of maximizing a structural saliency measure $\Omega(n)$ over all curves of length n starting from P.

The computation is linear in n because Ω has been chosen to be an *extensible* function. Hence, the most salient curve of length n at P will be equal to the maxima over all segments leaving P and the maximal curves of length $(n-1)$ starting at the respective end-points of these segments. The saliency measure is associated with each segment and not with the entire curve.

Because the area defined by the first stage is broadly defined, there will be several contours growing alongside each other. To extract the most salient curve, we propagate the saliency value of the most salient segment along the curve that contributed to its value. This is done iteratively by each segment maximizing over the value of its preferred neighbor and its own. Thus, the largest value is propagated along its curve. Finally, we perform a non-maximal suppression operation, where each segment suppresses all its neighboring segments if their saliency value was less and if they had similar motion estimates associated with them. Hence, the most salient contours belonging to differently moving objects remain alongside each other.

Finally, we have presented results that show that the developed methods can successfully segment scenes with several independently moving objects, without prior knowledge of the shape and motion of the objects. We have also shown that the developed methods can segment sparse depth maps.

Bibliography

[1] **Adiv, G.** (1985) "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, July.

[2] **Anandan, P. & Weiss, R.** (1985) "Introducing a smoothness constraint in a matching approach for the computation of optical flow," Proc. IEEE Workshop on Computer Vision: Representation and Control, Belaire, MI, October, 186-194.

[3] **Anstis, S. M.** (1970) "Phi movement as a subtraction process," Vision Research 10, 1411-1430.

[4] **Bandopadhay, A. & Dutta, R.** (1986) "Measuring image motion in dynamic images," IEEE Workshop on Motion: Representation and Analysis.

[5] **Baker, H. H. & Binford, T. O.** (1981) "Depth from edge and intensity based stereo," Seventh International Joint Conference on Artificial Intelligence August.

[6] **Braddick, O. J.** (1974) "A short-range process in apparent motion." Vision Research 14, 519-527.

[7] **Bülthoff, H.; Little, J. & Poggio, T.** (1989) "A parallel motion algorithm consistent psychophysics and physiology," Proc. IEEE Workshop on Visual Motion, Irvine, CA, March, 165 - 172.

[8] **Canny, J.F.** (1983) "Finding edges and lines," MIT Artificial Intelligence Technical Report 310.

[9] **Clocksin, W. F.** (1980) "Perception of surface slant and edge labels from optical flow: a computational approach." Perception 9, 253-269.

[10] **Drumheller, M. & Poggio, T.** (1986) "On Parallel Stereo," IEEE International Conference on Robotics and Automation.

[11] **Fennema, C. I. & Thompson, W. B.** (1979) "Velocity Determination in Scenes Containing Several Moving Objects," Computer Graphics and Image Processing 9, 301-315.

[12] **Gamble, E. & Poggio, T.** (1987) "Visual integration and detection of discontinuities: the key role of intensity edges," MIT Artificial Intelligence Memo, 970.

[13] **Geman, S. & Geman, D.** (1984) "Stochastic relaxation, Gibbs distribution, and the Bayesian Restoration of Images,"IEEE Trans. on Pattern Analysis and Machine Intelligence,Vol. PAMI-6, 721-741.

[14] **Grimson, W.E.L.** (1981) From Images to Surfaces (MIT Press, Cambridge).

[15] **Hildreth, E.** (1984) The Measurement of Visual Motion, (MIT Press, Cambridge).

[16] **Hillis, D.** (1985) "The Connection Machine," MIT Department of Electrical Engineering and Computer Science Ph.D. Thesis and MIT Press.

[17] **Horn, B.K.P. & Schunk, B.** G.(1981) "Determining Optical flow," Artificial Intelligence 17, 185-204.

[18] **Hutchinson, J.; Koch, C.; Luo, J & Mead, C.** (1988) "Computing motion analog and binary resistive networks," IEEE Computer, March, 1988.

[19] **Ikeuchi, K. & Horn, B.K.P.** (1981) "Numerical Shape from Shading and Occluding Boundaries," Artificial Intelligence 17, 141-184.

[20] **Koch, C.; Marroquin, J. & Yuille, A.** (1986) "Analog `neural' networks in early vision," Proc. Natl. Acad. Sci., Vol. 83.

[21] **Julesz, B.** (1971) Foundations of Cyclopean Perception. (University of Chicago Press, Chicago).

[22] **Lee, D.N.** (1970) "Binocular stereopsis without spatial disparity," Perception & Psychophysics, 9, 216-218.

[23] **Marr, D. & Ullman, S.** (1981) "Directional Selectivity and its Use in Early Visual Processing," Proceedings of the Royal Society of London B. 211, 151-180.

[24] **Marroquin, J.** (1984) "Surface Reconstruction Preserving Discontinuities," MIT Artificial Intelligence Memo, 792.

[25] **Mayhew, J.E.W. & Frisby, J.P.** (1981) "Psychological and computational studies towards a theory of human stereopsis" Artificial Intelligence 17.

[26] **Mutch, K. M. & Thompson, W. B.** (1985) "Analysis of accretion and deletion at boundaries in dynamic scenes," IEEE Trans. on Pattern Analysis and Machine Intelligence, March.

[27] **Nagel, H.-H.** (1984) "Recent advances in image sequence analysis." Proc. Premier Colloque Image -- Traiment, Synthese, Technologie et Applications, Biarritz, France, May, 545-558.

[28] **Nagel, H.-H. & Enkelmann, W.** (1986) "An Investigation of Smoothness Constraints for the Estimation of Displacements Vector Fields from Image Sequences," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, 565-593.

[29] **Nakayama, K. & Loomis, J. M.** (1974) "Optical velocity patterns, velocity sensitive neurons, and space perception: a hypothesis," Perception, Vol.3, 63-80.

[30] **Nakayama, K.** (1985) "Biological image motion processing: a review," Vision Research, Vol. 25, No. 5, 625 - 660.

[31] **Potter, J. L.** (1975) "Velocity as a cue to segmentation using motion information," IEEE Trans. Systems, Man, Cybernectics SMC-5, 390-394.

[32] **Prazdny, K.** (1984) "Stereopsis from kinetic and flicker edges," Perception & Psychophysics 36 (5), 93-99.

[33] **Prazdny, K.** (1985) "Detection of binocular disparities," Biological Cybernectics, Vol. 35, 81 -100.

[34] **Reichardt, W. & Poggio, T.** (1980) "Figure-ground discrimination by relative movement in the visual system of the fly. Part I: Experimental results," Biological Cybernectics, Vol.35, 81-100.

[35] **Schunck, B.** (1986) "The motion constraint equation for optical flow," Proc. Int. J. Conf. Pattern Recognition, 20-22.

[36] **Sha'ashua, A.** (1988) "Structural Saliency: the detection of globally salient structures using a locally connected network," Master's Thesis, Dept. of Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, CS88-18.

[37] **Sha'ashua, A. & Ullman, S.** (1988) "Structural Saliency: the detection of globally salient structures using a locally connected network," Proc. Int. Conf. on Computer Vision, 321 - 327.

[38] **Spoerri, A. & Ullman, S.** (1987) "The Early Detection of Motion Boundaries," Proc. Int. Conf. on Computer Vision, 209 - 218.

[39] **Stevens, K. A.** (1977) "Computation of locally parallel structure," MIT Artificial Intelligence Memo 392.

[40] **Terzopoulos, D.** (1986) "Regularization of Inverse Visual Problems Involving Discontinuities," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, May.

[41] **Thompson, W. B.; Mutch, K. M. & Berzins, V. A.** (1985) "Dynamic occlusion analysis in optical flow fields," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 4.

[42] **Ullman, S.** (1979) The Interpretation of Visual Motion, MIT Press, Cambridge.

[43] **Ullman, S.** (1981) "Analysis of visual motion by biological and computer systems," IEEE Computer, August, 57-69.

[44] **Ullman, S.** (1983) "Visual Routines," MIT Artificial Intelligence Memo, 723.

[45] **Ullman, S.** & Sha'ashua, A. (1989) "Structural Saliency: the detection of globally salient structures using a locally connected network," MIT Artificial Intelligence Memo 1061.

[46] **Wildes, R.** (1989) "On interpreting stereo disparity," MIT Artificial Intelligence Technical Report 1112.

[47] **Wohn, K. & Waxman, A. M.** (1985) "Contour evolution, neighborhood deformation and local image flow: curved surfaces in motion," Univ. Maryland Center for Automation Research Tech. Rep. 134, July.