MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

AI Memo No. 1028 April, 1988

Symbolic Construction of a 2D Scale-Space Image

Eric Saund

Abstract

The shapes of naturally occurring objects characteristically involve spatial events occurring at many scales. It is important to make explicit the multiscale structure of a shape object in order to effectively perform shape recognition or to engage in other forms of reasoning about shape. Currently available techniques for multiscale shape analysis include image blurring and contour smoothing; each of these techniques involves uniform application of a smoothing operator to the entire image array or contour. This paper offers a new, *symbolic*, approach to constructing a primitive shape description across scales for 2d binary (silhouette) shape images. Under this approach, *grouping* operations are performed over collections of tokens residing on a Scale-Space Blackboard. Two types of grouping operations are identified that, respectively: (1) aggregate edge primitives at one scale into edge primitives at a coarser scale, and (2) group edge primitives into partial-region assertions, including curved-contours, primitive-corners, and bars. Algorithms to perform these computations are presented.

This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence Research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124. The author is supported by a fellowship from the NASA Graduate Student Researchers Program.

©Massachusetts Institute of Technology 1988

1 Introduction

The shapes of naturally occurring objects characteristically involve spatial events occurring at a multitude of scales. For example, the fish shape in figure 1 appears at a coarse scale simply as an elongated blob; at a medium scale as a somewhat more well-defined blob with smaller blobs (fins) attached; and finally, at a fine scale, as a sharply defined Anchovy complete with pronounced fin contours, pointed tail flukes, and a mouth. Shape details appearing at finer scales are situated in relation to one another by the spatial structure emergent at coarser scales. It is important to make explicit the

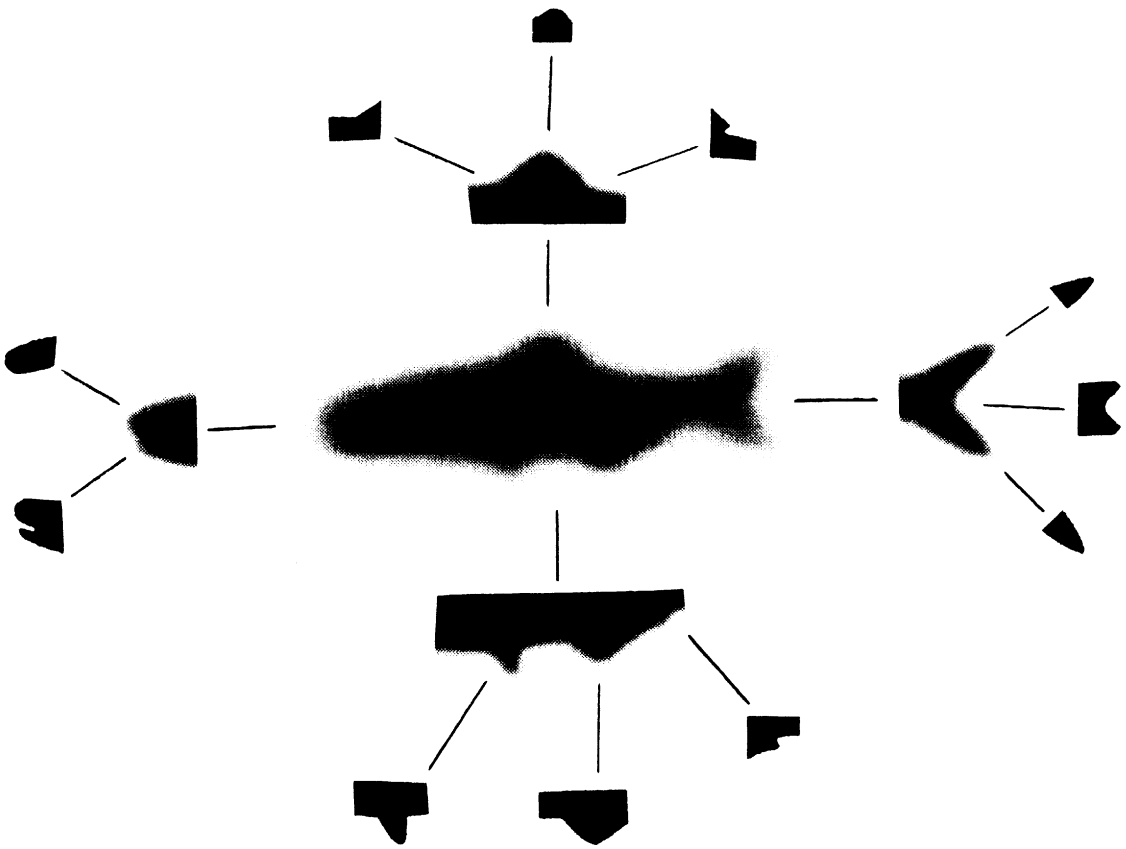


Figure 1. Important shape features occur at many scales.

multiscale structure of a shape object[1] in order to effectively perform shape recognition or to engage in other forms of reasoning about shape because important distinguishing characteristics or features may occur at any scale.

For this reason one widely cited goal for early visual shape processing is to construct a description of a shape at a variety of scales [Witkin, 1983; Mokhtarian and Mackworth, 1986; Asada and Brady, 1986; Pizer et al, 1986; Koenderink, 1984; Burt and Adelson, 1983; Crowley and Parker, 1984; Crowley and Sanderson, 1984; Sammet and Rosenfeld, 1980]. From these descriptions may be extracted important primitive shape events to be used by later stages devoted to object recognition or other visual tasks. This paper is concerned with building multiscale shape descriptions of two dimensional binary (silhouette) shape images in terms of edge and region (blob) shape primitives.

Currently available techniques for multiscale shape analysis are of two basic types: contour-based smoothing and region-based smoothing. Both of these approaches are based on the application of a numerical smoothing operator uniformly to some one-dimensional (contour-based) or two-dimensional (region-based) array of shape data. The operator is typically characterized by a size or width parameter indicating the degree of smoothing performed and hence the scale of the result. Region-based smoothing techniques may be further subdivided into isotropic smoothing operators, and oriented filters. As will be shown, at coarse scales both contour-based smoothing and isotropic region smoothing approaches fail to capture in a consistent manner important structure inherent to shape objects. The prospects for oriented filters are uncertain.

This paper describes a fundamentally different approach to extracting primitive shape descriptions at multiple scales. The approach is based on grouping of shape tokens in the style of the Primal Sketch [Marr, 1976]. Each token may bear more information than just the local magnitude of an image intensity or local orientation of a contour. The approach may be considered *symbolic* because the tokens are, conceptually, discrete entities, and because the grouping steps actually taken depend necessarily on the shape data itself. This is in contrast to uniform numeric smoothing algorithms which carry out the same arithmetic procedure everywhere regardless of the shape content of the data.

An important tool we introduce for carrying out the grouping operations

[1] We refer to a figure whose shape we are analyzing as a *shape object*.

is the *Scale-Space Blackboard.* Tokens are placed on the Blackboard according to their location, orientation, and scale. The Scale-Space Blackboard facilitates manipulation of shape information because it permits tokens to be indexed on the basis of location and scale.

The grouping procedures specify situations under which a collection of tokens should give rise to a new token. Two types of grouping operation are presented: (1) Fine-to-coarse aggregation of edge primitives generates a coarser scale edge map from finer scale edge primitives, (2) Pairwise grouping of symmetrically placed edge primitive tokens supports assertions of *curved-contour*, *primitive-corner*, and *bar* events, all of which demark partial-regions. These events are marked by partial-region type tokens placed on the Scale-Space Blackboard.

The outline of the paper is as follows: The remainder of the Introduction explores characteristics desired of a multiscale shape representation. Sections 2.1 and 2.2 briefly illustrate disadvantages of contour-based smoothing and isotropic region based smoothing approaches to identifying important coarse scale structure in shape images, while Section 2.3 shows that oriented edge filters offer some improvement over isotropic region-based smoothing operators. Section 3 introduces the *Scale-Space Blackboard* as a data structure which allows shapes to be manipulated symbolically, while preserving a pictorial quality to the organization of spatial information. Section 4 offers an algorithm for fine-to-coarse aggregation of edge primitives through token grouping. Section 5 presents rules for grouping edge primitives in order to identify more complex structures constituting partial-regions.

1.1 Objectives for Multiple Scale Shape Representation

The motivation for describing shapes at multiple scales is to separate geometric features and properties of differing size or scale, on the assumption that they are likely to reflect different parts, processes, or functional properties of objects encountered in the visual world. For example, the body and stem of an apple are related to one another by, among other things, a difference in relative size. If the early stages of visual processing can deliver object descriptions making explicit relative sizes, then later stages of processing, such as visual recognition, may be assisted in carrying out tasks such as matching

these descriptions to internal models of known objects: An apple consists of a large blob (body) with a small elongated part (stem) attached.

In evaluating the performance of a multiple scale shape description, it is important to have established, at the outset, expectations for just what sorts of geometric structure the computation is intended to segregate according to size or scale. We proceed from the following notion: size or scale corresponds to *spatial extent* in the image of a shape object. Thus, the body of an apple is considered a larger scale feature than the stem because it has greater spatial extent.

To be more precise, however, the term, "spatial extent," may be interpreted in either of two ways: as linear distance, or as area. It is clear that the body of an apple is a large scale feature relative to the stem, both because its diameter is larger than the length of the stem, and because it has greater area than the stem. But suppose the apple is hanging from a string. (See figure 2). The string may have a length comparable to the diameter of the apple, but, because of its narrow width, cover an area more similar to that

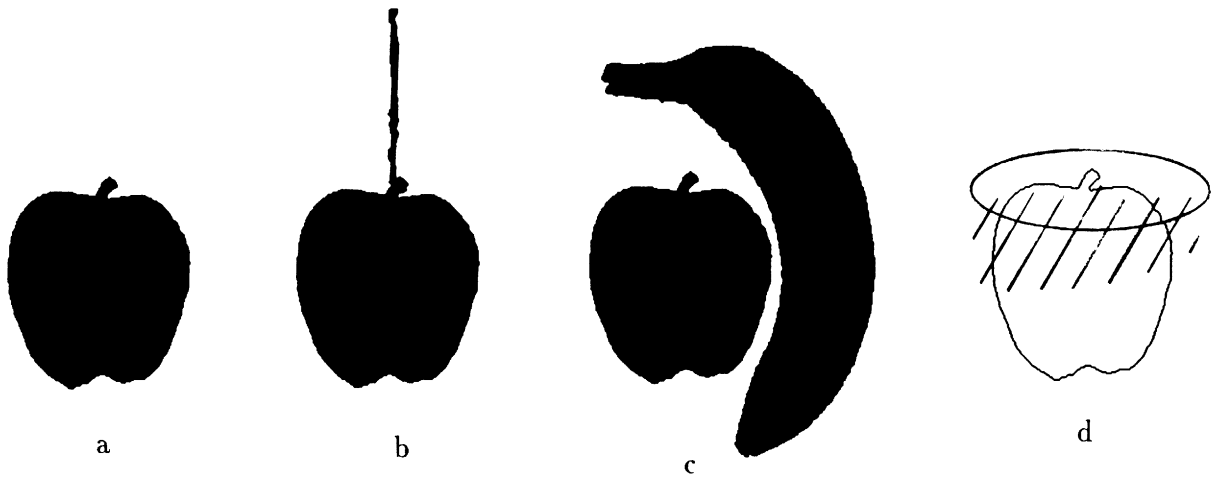


Figure 2. A two-dimensional apple shape (a) retains its fine and coarse scale structure even when the apple hangs from a string (b) and when the apple is placed near another large object (c). d. The large scale figure/ground boundary formed by the top of the apple remains unchanged under these circumstances.

of the stem. So should the string be considered a large or small scale spatial event?

This example suggests that a multiscale shape representation treat object boundaries differently from the regions they enclose. Thus, the scale assigned to a contour boundary, such as the edge of a piece of string, should depend on its linear extent, while the scale assigned to a local blob or region, such as the body of the apple or a snippet of string, should depend upon its area.

If the purpose of a multiscale shape description is to segregate features according to scale, then shape events at different scales should not interfere with one another. For example, the rounded top of an apple forms a large scale boundary between the body of the apple and the background, as shown in figure 2d. The presence of the small scale apple stem, or even the string, does not change this gross feature, and the coarse scale description of this boundary should not be affected by the presence or absence of the stem or string. Conversely, the description of smaller scale shape features or properties should remain unchanged no matter what their proximity to large features. For example, were the apple placed next to another, much larger object, the body of the apple would become, in comparison, a small scale object (figure 2c). Nonetheless, the description of the apple body should remain unaffected; the apple is still a roughly circular blob with dimples on the top and bottom.

2 Uniform Numerical Smoothing Methods

A two-dimensional region, and the one-dimensional contour enclosing this region, are complementary ways of describing a two-dimensional shape object. Accordingly, two alternative schemes are available for representing a shape object at the pixel level: as a two-dimensional array indexed by x, y spatial coordinates, or, as a one dimensional array indexed by distance along the contour, s. With each type of representation are associated natural approaches to obtaining descriptions at different scales by applying some form of numerical smoothing technique uniformly to the data.

2.1 Contour-Based Smoothing

Contour based shape representations organize the description of a shape in terms of a succession of points along an object's boundary. Several variations of contour based shape representation have been used. These include encoding of: (1) successive pixel (x, y) location, eg. [Mokhtarian and Mackworth, 1986], (2) differences in successive pixel locations $(\Delta x, \Delta y)$, eg. [Freeman, 1974], and (3) local orientation $(arctan\frac{\Delta y}{\Delta x})$, eg. [Asada and Brady, 1986]. Contour smoothing operations modify the path of the two-dimensional contour curve in space, and sometimes also its length. Here we illustrate contour based smoothing under the technique of encoding pixel (x, y) location as a function of arc length, s (measured in terms of pixel count), and smoothing the $x(s)$ and $y(s)$ functions independently:

$$x'(s) = \sum_{i=-a\sigma}^{a\sigma} G_\sigma(i)x(s-i) \tag{1}$$

$$y'(s) = \sum_{i=-a\sigma}^{a\sigma} G_\sigma(i)y(s-i), \tag{2}$$

where G is a Gaussian of width σ and the factor, a, effectively truncates the tail of the Gaussian ($a = 3$ is a suitable number). Under this scheme a closed contour is guaranteed to remain closed after smoothing, while this is not true for representations of orientation versus arc length. Figure 3 shows the contour of an apple shape under different degrees of contour smoothing obtained by using Gaussians of various widths.

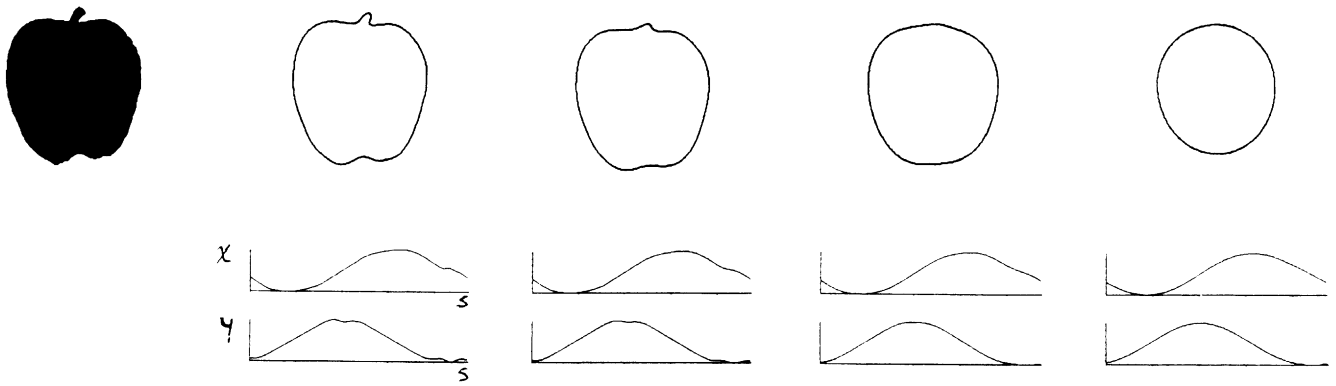


Figure 3. Apple shape encoded in terms of pixels along its bounding contour, $x(s)$ and $y(s)$. Smoothing these one-dimensional arrays yields a smoothed shape contour.

For some shape objects, contour-based smoothing does a good job of removing fine scale detail while preserving the larger scale aspects of the shape. Indeed, the apple is one example of such a case. However, many other shapes exist for which contour smoothing fails to identify important coarse scale structure, or else inappropriately suggests the presence of nonexistent coarse scale structure. Figure 4 illustrates. To the human eye, in figure 4a two parallel bars are prominent; under contour smoothing one of the bars remains at a coarse scale, while the other breaks up. In figure 4b, the apple is shown hanging from a string. Contour smoothing to a coarse scale results in misleading distortion and absurd implications about the gross shape. These effects can create hardships for any later processing stages which may seek to perform part segmentation, match to object models, or otherwise interpret coarser scale shape descriptions. A related problem arising with contour-based smoothing occurs in figure 4c. Here, a banana is placed near the apple. A very small change in shape, resulting from the banana being moved a little closer to the apple, leads to a very large change in the coarsely smoothed contour.

As these examples show, contour based representations place undue emphasis on the topology of shape boundaries. The resulting descriptive instabilities are likely to introduce insurmountable complications later on. We conclude that purely contour-based smoothing approaches do not provide an appropriate basis for constructing multiscale shape descriptions.

2.2 Isotropic Region-Based Smoothing

Region based smoothing techniques start with representations for shape consisting of two-dimensional arrays of numbers. A two-dimensional shape object (silhouette) assigns the value, (say) 1, to locations in a two-dimensional array covered by the object (figure), and 0 to the surrounding space (ground). In general, filtering a two-dimensional array of binary-valued pixels results in an array containing real numbers. Each such grey-level value may be interpreted as the "strength" of the filtering kernel response at that location.

Most popular among region-based smoothing operators is convolution with the circularly symmetric Gaussian. This operator is spatially isotropic, and is often followed by a differential operator such as the Gradient Magnitude or Laplacian. The latter is usually incorporated into the Gaussian smoothing step, yielding the well known $\nabla^2 G$, and its approximation, the

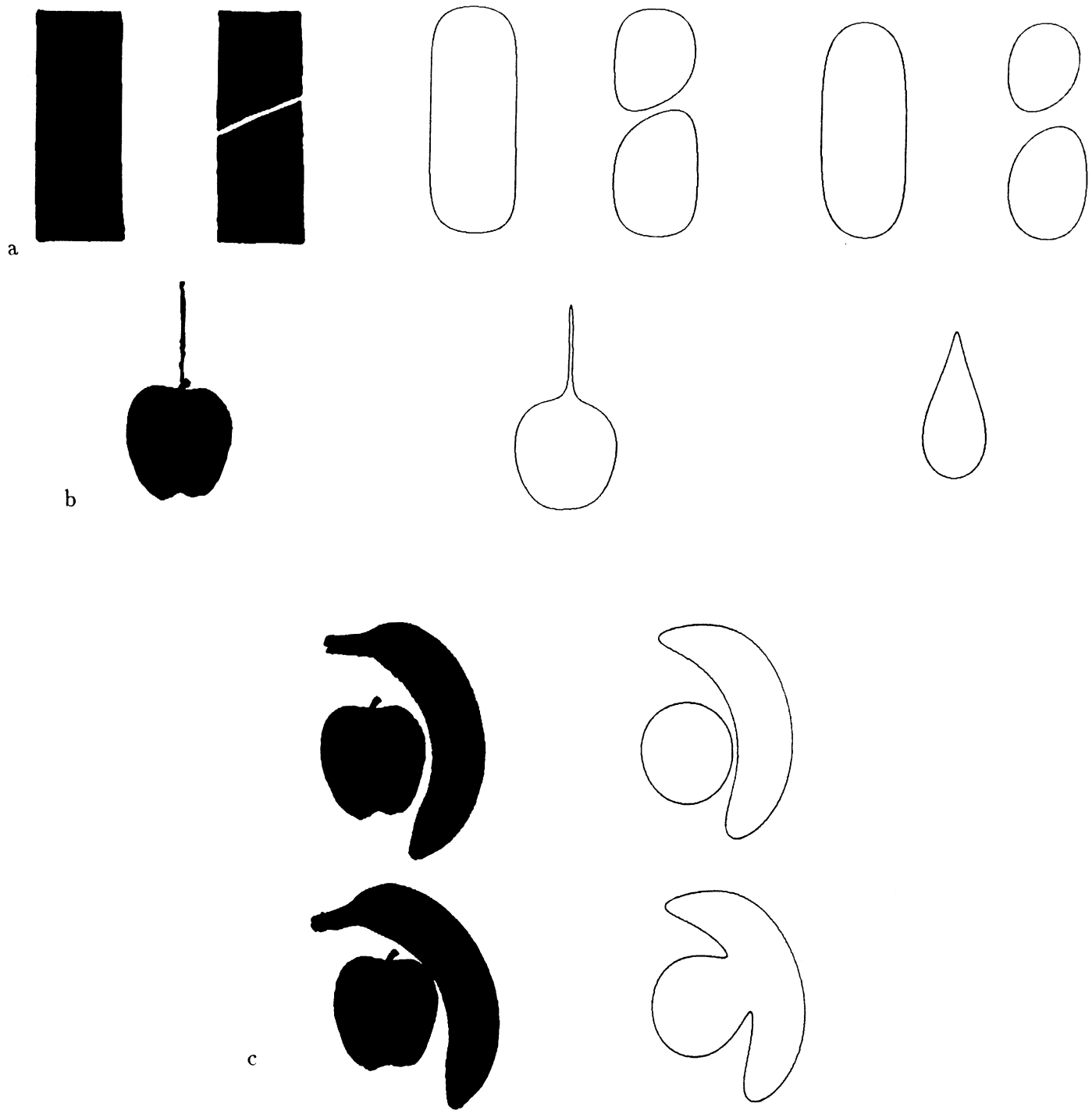


Figure 4. a. Contour smoothing fails to capture the large scale interpretation that two parallel bars are present. b. Under contour smoothing, a string tied to the apple grossly distorts the apple's shape at coarse scales. c. Moving a banana so that it just touches the apple leads to a large and discontinuous change in the coarse scale description. Contour-based smoothing methods place undue emphasis on the topology of bounding contours.

DOG (Difference of Gaussians). The outputs of these filtering operators typically feed some sort of thresholding step resulting in edge [Marr and Hildreth, 1980; Canny, 1986] or region/blob [Crowley and Sanderson, 1984; Crowley and Parker, 1984; Voorhees, 1987] assertions.

Figure 5 shows the result after Gaussian smoothing the binary silhouette of an apple with filters of various widths. Also shown are edges found by thresholding and then thinning the gradient magnitude[2]. Gaussian smoothing yields a field of numbers that may be interpreted as the "density of matter" at each spatial location, averaged in all directions. The edges found by taking peaks in the gradient magnitude of this map do a good job of removing small scale details about the apple's bounding contour, while preserving its overall, large scale shape.

Figures 6 and 7, however, show that the isotropic Gaussian blurring operation may obliterate evidence of extended edges when they occur in proximity to large yet unrelated regions or when they enclose narrow regions. In figure 6, the string tied to the apple is lost altogether under thresholding following Gaussian blurring. Because of its narrow width, it dissipates away under even moderate amounts of blurring.

The converse problem arises in figure 7, in which the apple shape is placed next to the banana. Now, the results of Gaussian smoothing and coarse scale edge detection yield an apparent coarse scale contour for the apple shape that is substantially different from the one obtained in figure 5. What happens is that, at coarse degrees of smoothing, "matter" from the banana leaks over to the region of the apple. Evidently, under Gaussian blurring, the coarse scale description of an object's shape cannot be trusted to remain stable under the presence of nearby objects, even when no object occludes any other. Again, as in the contour smoothing case, this instability effectively undermines the purpose of multiscale shape analysis.

[2]This is the foundation of the popular Canny edge detector.

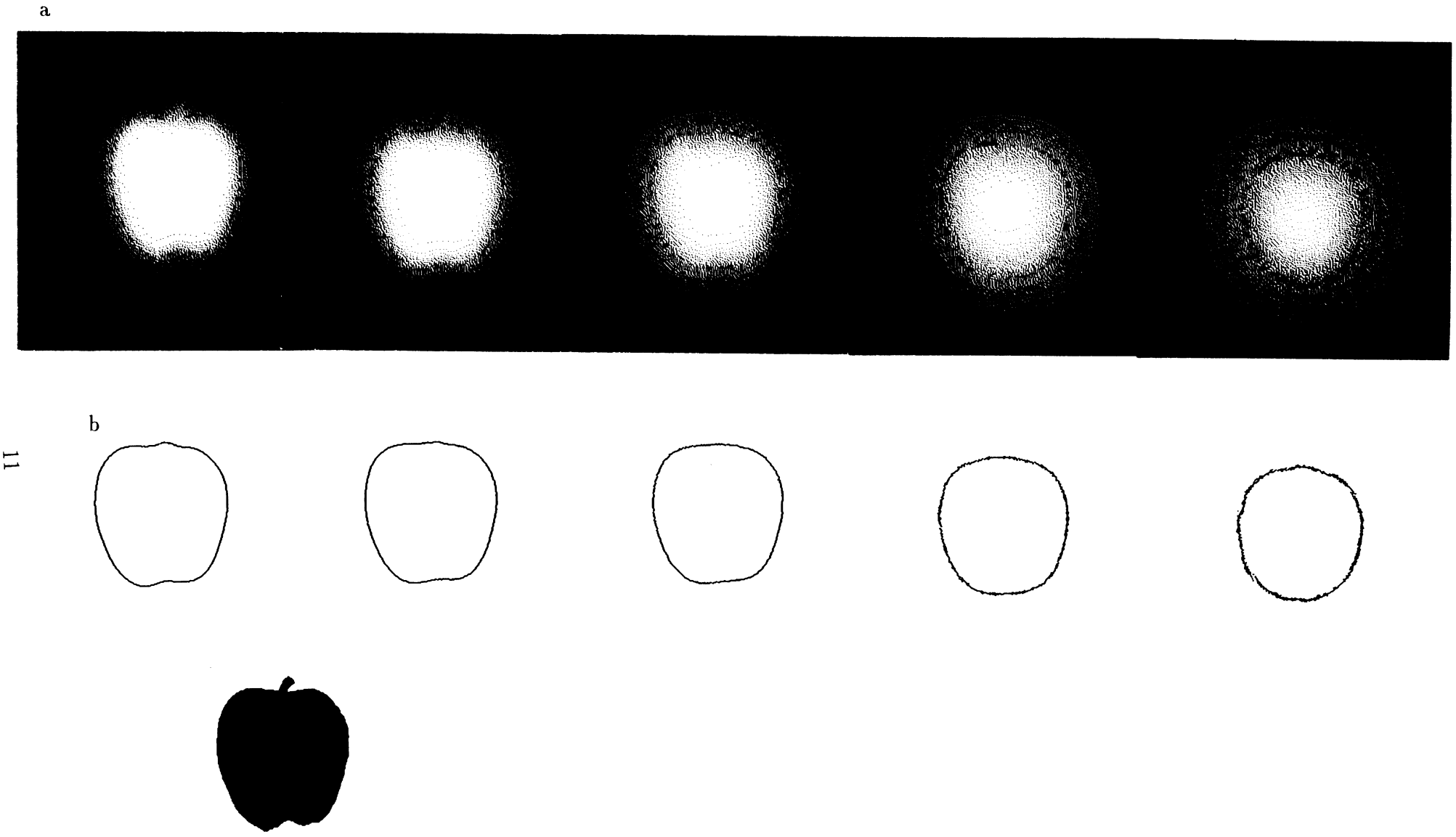


Figure 5. a. An apple shape smoothed with two-dimensional Gaussian filters of different widths (σ). b. Edges found by thresholding and thinning peaks in the gradient magnitude.

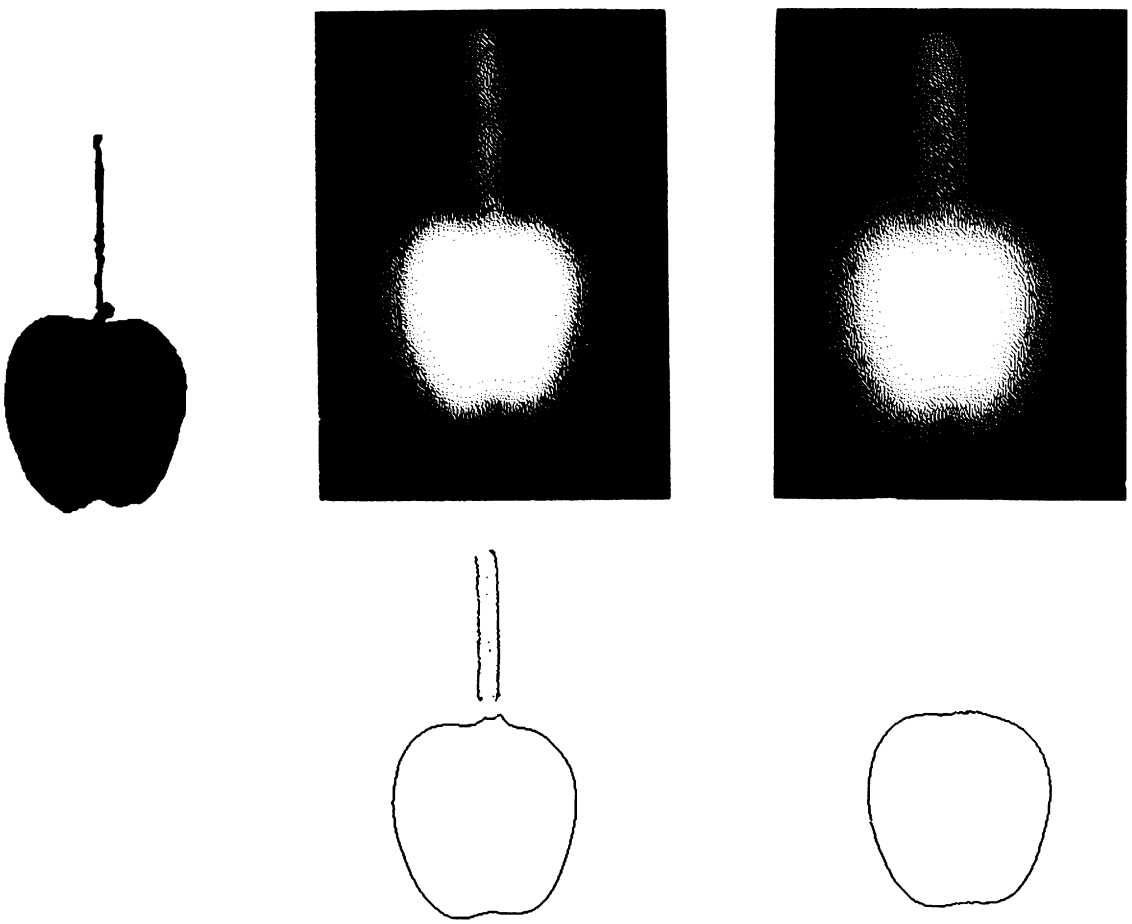


Figure 6. Under Gaussian blurring the string dissipates away even though it has large spatial extent along its length.

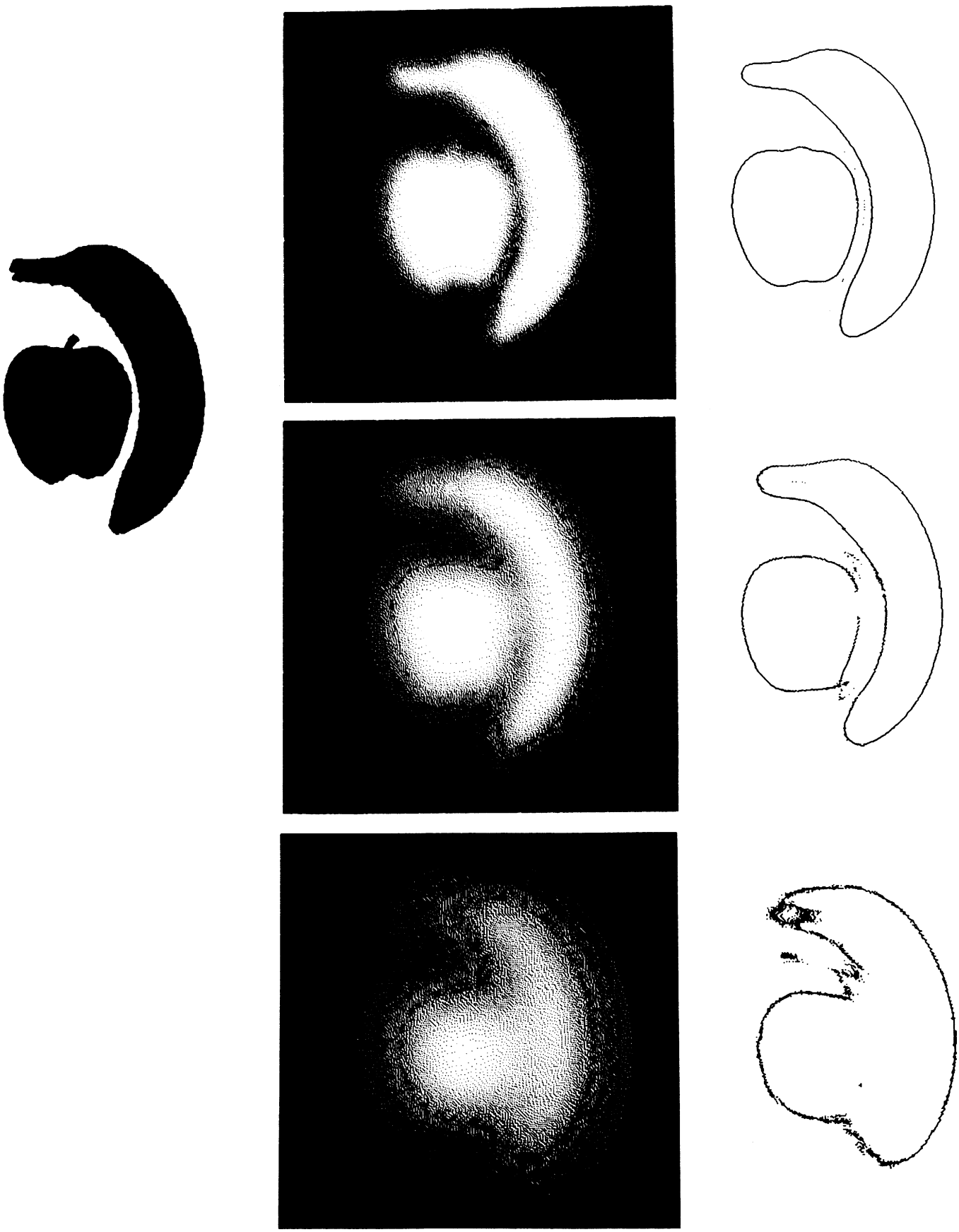


Figure 7. When the apple is placed near the banana, Gaussian blurring bleeds them together and distorts evidence of their large scale geometry.

2.3 Oriented Region-Based Filters

Another class of region based operators for extracting events at multiple scales are oriented filters, such as the Gabor filters [Daugman, 1985]. Here, we illustrate the performance of oriented edge masks consisting of a Gaussian weighting along the length of the edge, and the derivative of a Gaussian across the edge (figure 8)(see [Zucker and Iverson, 1987], who use the 2nd derivative of the Gaussian). Orientation tuning is determined by the relative widths of these profiles. Because oriented filters carry out spatial averaging non-isotropically, that is, depending upon the orientation and eccentricity of the mask, they perhaps stand a better chance of achieving smoothing along the length of a contour, while isolating regions lying on opposite sides of the contour.

Figure 9 shows the results of oriented edge detection for the apple shape. The filter mask was convolved with the original binary image at sixteen different orientations for each scale, and yields sixteen grey-level arrays for each scale. In order to facilitate presentation, it is convenient to condense this

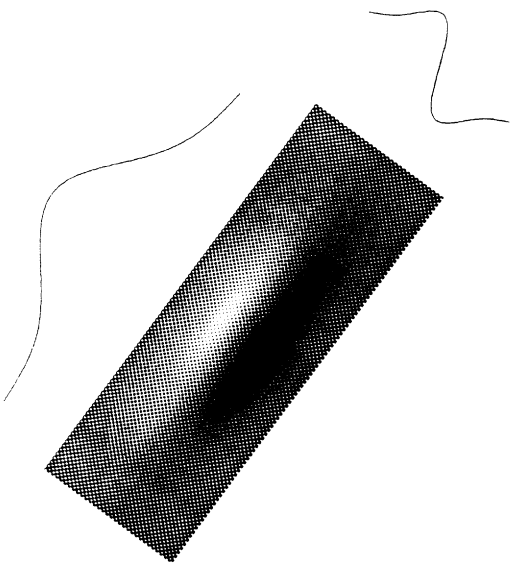


Figure 8. Oriented two-dimensional edge mask.

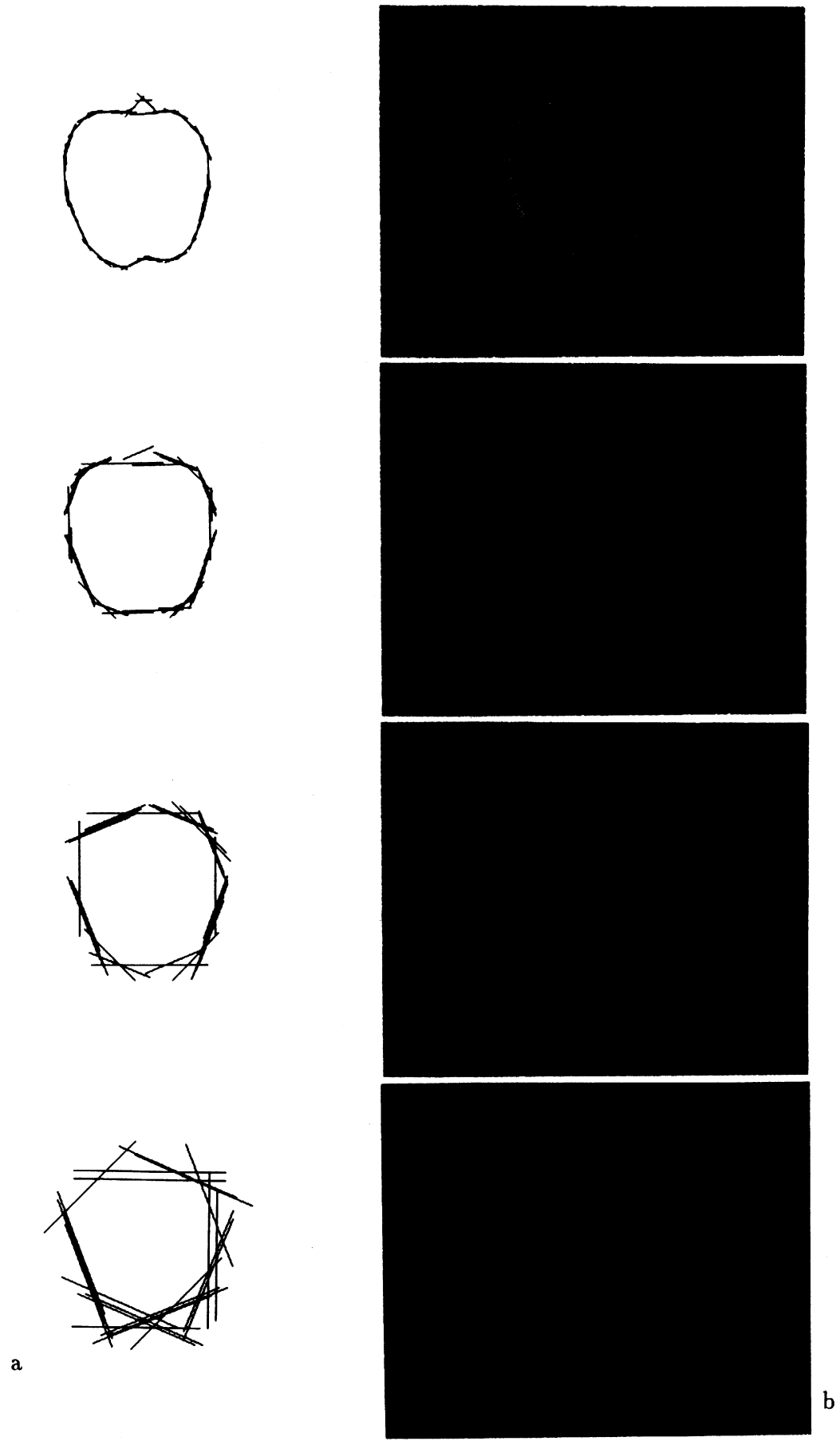


Figure 9. Apple shape under oriented edge filtering. a. Line segments denote orientations of edges after thinning and thresholding. b. Maximum filter response out of 16 orientations.

large amount of information into two arrays of numbers for each scale. One (figure 9b) depicts the strength of the maximally responding filter response, at each spatial location, the other (figure 9a) shows the orientation of the maximally responding filters for a selected subset of spatial locations, such as, for example, locations where the filter response is above a certain threshold.

Figure 10 indicates that the performance of oriented filters in identifying extended edges at coarse scales is improved over isotropic Gaussian smoothing. For example, in the absence of background clutter, the string is detected at fairly coarse scales when its boundary contour aligns with the orientation axis of the elongated mask.

However, figure 11 suggests that cases yet exist where oriented edge filters fail to identify important coarse scale edges. One source of difficulty arises from the fact that large aspect ratios may be required to detect long edges bounding an object placed very near to another object. Such greatly elongated filters by and large bring severe orientation tuning, and an inordinate number of them may be required to cover the visual field at all orientations. It is not clear to what extent this problem tarnishes the advantages of oriented filters.

Uniform numerical smoothing techniques are conceptually straightforward and simple to apply, but these in themselves amount to no sound bases for believing that they should necessarily extract the important shape properties that later visual processes can most effectively use. It seems possible, though, that oriented filters may yet offer some promise for finding large scale structure in shape images. We leave them as a subject for additional study, and turn next to a very different approach to multiscale shape analysis.

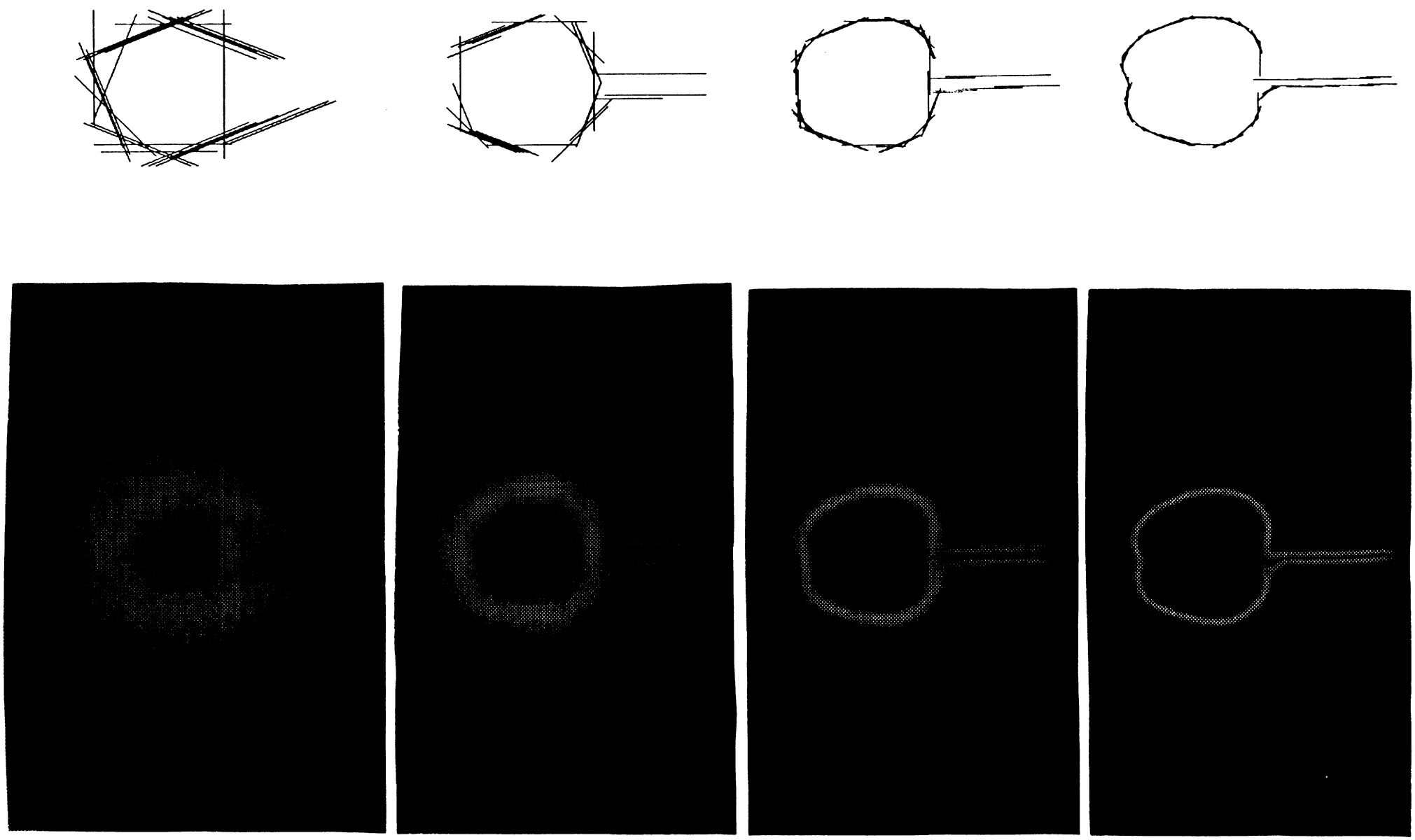


Figure 10. Under oriented edge detection the string is detected by larger masks than under isotropic blurring because the masks are more closely matched to the string's extended boundary contour.

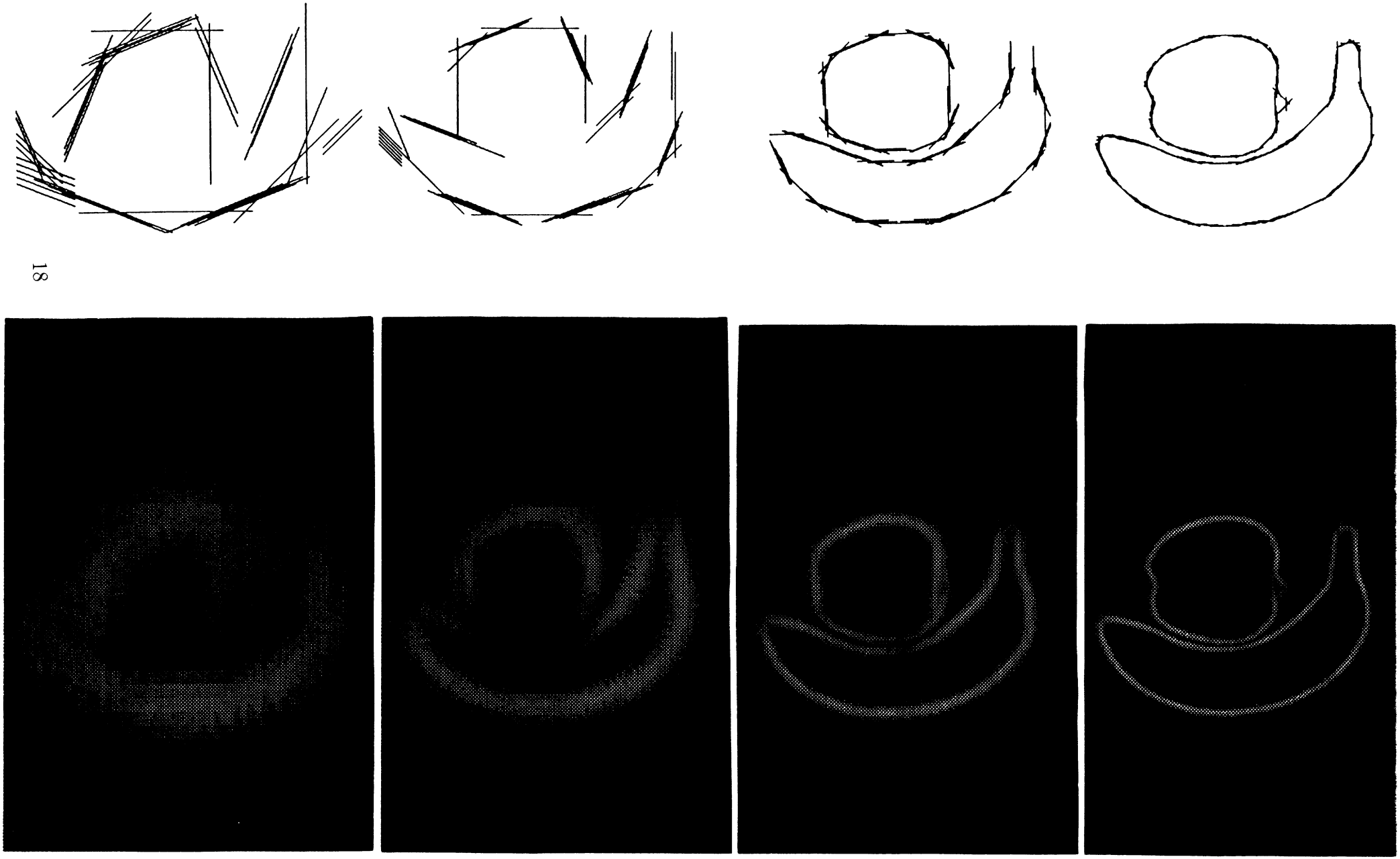


Figure 11. Using oriented edge filters, the large scale structures of the apple and banana are poorly detected. If large masks are used to identify contours bounding narrow regions, then they must be closely spaced, have high aspect ratios, and sample at many orientations.

3 The Scale-Space Blackboard

3.1 Tokens vs. Fields of Numbers

The purpose of a shape representation is to distinguish, identify, and characterize—to make explicit—certain shape properties and spatial events in the shape image that are likely to have significance either in the external world or to the system's task goals. By highlighting and naming these events, important information can be more easily manipulated by later processes carrying out pattern matching, counting, tracing, perceptual grouping, and other operations.

Alternative interpretations are available for what it takes to "make information explicit." In the case of typical region-based edge detecting filters, for example, "edgeness" is made explicit over the entire image in the form of a field of numbers describing the response strength of a convolution kernel centered at each pixel. On the other hand, edge information may also be said to have been made explicit in a list of line segments fit to edges in the image. The former representation may be called *iconic*, or *image-like* [Pylyshyn, 1973, 1981; Anderson, 1978; Kosslyn, et. al. 1979], while the latter is considered *symbolic*. Most approaches to later shape interpretation employ symbolic representations because they offer greater flexibility in assigning meaningful interpretations to parts of shape, for example, that "this edge corresponds to the stem of an apple."

This work adopts an intermediate representational format preserving the spatial character of an iconic representation while permitting symbolic tags to be attached to spatial events occurring in a shape image. The genus may be called *semi-iconic* representation. Information is made explicit via symbolic *tokens*. Tokens are symbolic in that, unlike pixel values, each token can maintain lists of properties, pointers, and other items of internal state. Yet, the pictorial aspect of spatial geometry is preserved by the assignment to each token of a location on the shape image. Furthermore, as is discussed in the next section, the tokens may be indexed by spatial location. Not every point in the image is necessarily covered by a token, however, and some locations may be associated with more than one token. The use of tokens in making explicit important image events was introduced by Marr [1976, 1982] in his proposal of the Primal Sketch as an early visual image representation, and has been applied to multiscale straight line extraction by

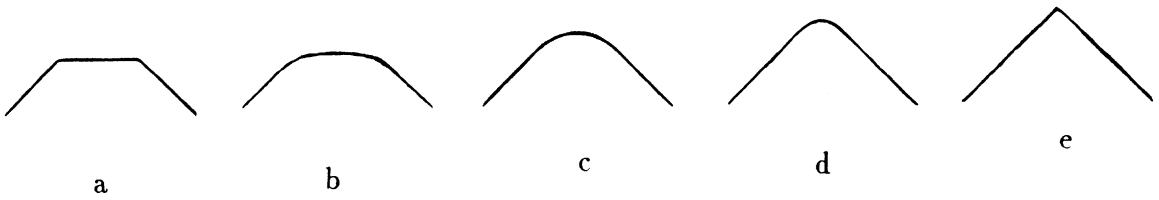


Figure 12. A sharp corner may be continuously deformed into a flattened corner. As the flattened edge gradually disappears, at some point a decision must be made that a corresponding edge token should no longer be asserted. A priori, no principled grounds exist for defining the decision criteria.

Weiss and Boldt [1986] (see also Boldt and Weiss, [1987]).

The transition from an iconic to a symbolic representation raises an issue of discretization. Shapes are fundamentally continuous things. Consider the sharp corner shape shown in figure 12e. This may be continuously deformed into a flattened corner, figure 12a. An iconic representation has no trouble describing shapes anywhere along this continuum because every location is assigned some pixel value. In contrast, a symbolic or a semi-iconic representation is inherently discrete: properties are asserted only for locations where a symbol or token has been assigned. Any time a discrete representation is to be computed from a continuous representation, qualitative decisions must be made of the form, "Should we put a token here?" Usually this decision involves the use of some threshold value, for example, "put a token everywhere an edge is present stronger than x".

It is important that later processes performing operations on discretized representations not rely upon the presence or absence of tokens that might or might not have been asserted had a threshold been slightly different. This is to say, it is desirable for a shape representation to preserve the continuous qualities that the world of naturally occurring shapes in fact displays. We attempt to abide by this principle by endowing each token with a *strength* parameter[3]. The strength parameter indicates to roughly what degree the shape property associated with a token is asserted at that token's particular location in the image. Later processes manipulating the information conveyed by shape tokens are intended to achieve independence from the instabilities of early quantization steps by modulating their computations

[3]Alternatively this may be called a *response-strength* or *activity* parameter.

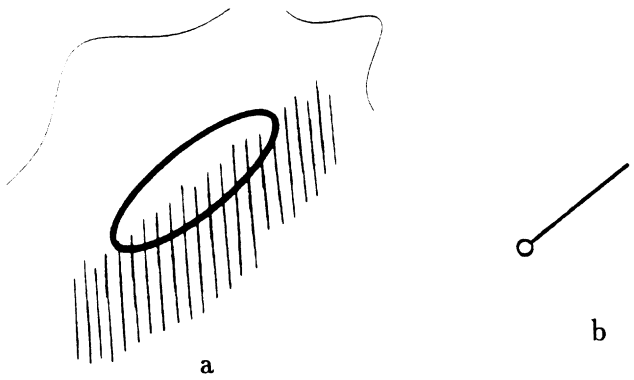


Figure 13. An edge primitive is marked by a token. The edge is viewed as having spatial extent roughly corresponding to a gaussian ellipsoid. A primitive edge token is displayed either as an ellipse (a), or as a line segment with a circle at the "front" end indicating the figure/ground orientation of the edge (b).

according to the tokens' strength parameters. As a given shape property fades from significance its later implications can have waned before its associated token disappears entirely.

The primary token employed in building multiscale shape descriptions is the *edge primitive*. In addition to strength, an edge primitive possesses the attributes of *x spatial location*, *y spatial location*, *orientation*, and *scale*. The primitive edge token denotes a boundary between figure and ground occurring approximately along its length axis, in much the same way as that measured by the oriented edge filter shown in figure 8. Though its token is assigned specific (x, y) coordinates, an edge primitive is to be interpreted as asserting information about some elongated local region as shown in figure 13. The edge assertion is to be considered strongest at the center of the region, and it diminishes with increasing distance.

3.2 Justification for Scale-Space

Despite their deficiencies in extracting coarse scale structure, contour based and region based numeric smoothing techniques deliver identical results in the limit of the finest scales of resolution. For example, were we to distribute edge-denoting tokens at nearby intervals along a very slightly smoothed ob-

ject boundary contour, these would agree with tokens located by taking the maximum gradient magnitude following slight two-dimensional Gaussian smoothing. Although we would properly label these as fine scale edges, the coarse scale structure of the shape remains implicit in the distribution of tokens about the image. Our goal is to make this coarser scale structure explicit, for example by placing appropriate additional tokens on an image.

The approach we offer to computing where such additional tokens might go is to look directly at patterns of smaller scale tokens already present. The style of computation corresponds to what is widely known as a "blackboard architecture" in the Artificial Intelligence literature: maintain a set of current assertions, as if they were written out on a blackboard. A set of rules or procedures performs pattern matching on the contents of the blackboard, and updates these contents by erasing, adding, and modifying assertions. In the present case, assertions about shape are made by placing shape tokens into the blackboard.

3.2.1 Indexing Spatial Information in a Blackboard

A number of important design choices are available as to just where and how various aspects of shape information are to be stored and organized, using a blackboard architecture. Note that having two-dimensional (as in a physical blackboard) or n-dimensional spatial arrangement is only an optional component to the organization of blackboard architectures as they are classically viewed.

The most crucial set of issues revolves around the means provided for *indexing* into the blackboard, that is, for addressing and accessing the shape information it contains. The following question arises: To what degree is information viewed as residing "inside" a token, and to what degree in terms of the token's location in some coordinate system defined on the blackboard. To illustrate, the information borne by each edge token could be written on a scrap of paper tossed in a heap; one examines symbols written on the scraps to read off tokens' location in space, orientation, and other properties. The blackboard becomes then the heap of paper. Alternatively, a physical blackboard on a wall may easily be assigned a two-dimensional coordinate system making explicit horizontal and vertical distance from an origin; a shape token might correspond to a dot drawn on the blackboard, this token expressing information only by virtue of its location on the board's surface.

Obviously, each scheme has its advantages and disadvantages. The token-as-scraps-of-paper scheme permits each token to maintain a large number of properties about itself, such as location, orientation, strength, time of day that it was created, and so forth, but this scheme offers no efficient way of attacking the heap to find a token possessing a given set of properties. Conversely, the coordinate-system scheme provides a handy means for indexing information on the basis of content—is there an edge at location $(4,5)$?, just go there and look—but it requires that the blackboard have as many dimensions as independent pieces of information denoted by each token.

For the present purposes, we adopt an intermediate course: tape scraps of paper to the blackboard. Tokens are localized on the blackboard in terms of a coordinate system organizing along a few crucial properties, but each token possesses internal state maintaining additional useful information. The interesting design choice arising is, which information is important enough to merit its own coordinate dimension on the blackboard?

In the world of two-dimensional shape objects, four leading candidates present themselves. These are, *x spatial location*, *y spatial location*, *orientation*, and *scale*. These are the four geometric parameters fixing an edge primitive in the representation: Where is it?, What is its orientation?, and How big is it? Because shape silhouettes are by definition two-dimensional images, x, y coordinates are obvious choices for structuring the blackboard. As for the other two candidates, Walters [1987] has argued in favor of *rho-space*, in which a third, ρ, dimension makes explicit the *orientation* of features, and Witkin [1983] suggests creating a *scale-space* by establishing a separate *scale* dimension[4].

Scale-space segregates spatial events of different sizes, that is, it provides a handle for indexing information on the basis of scale. The size of an edge primitive, for example, is indicated by the placement, along a separate scale (σ) dimension, of a token corresponding to that edge. This organization simplifies the sequence of operations required to query a shape description as to whether certain properties are true of the object under observation. If a pattern matching rule needs to know whether a medium scale edge at location $(5,6)$ and orientation $32°$ is present in order to decide that an object

[4]Witkin's original presentation of scale-space dealt with the evolution across scales of zero-crossings of a DOG-filtered one-dimensional signal, as the width of the Gaussian filter increases. Here, we forbear zero crossings and instead refer only to the use of an independent dimension denoting size or scale.

has parallel sides, then under a scale-space organization it may more rapidly narrow down the set of tokens that must be examined than if it had to check through tokens representing all scales. Depending upon the degree to which algorithms for analyzing shape regard scale as an important shape property, this gain in efficiency may be as significant as that obtained by ruling the blackboard with x, y spatial coordinates.

Similar gains in efficiency may be obtainable, for some purposes, with blackboard organizations making explicit a separate orientation dimension. However, given the stated purpose of identifying the multiscale structure of shapes, and because of the difficulties in managing high-dimensional spaces, the present work sacrifices the possibility of indexing shape information directly on the basis of orientation, and instead employs a Scale-Space Blackboard consisting of two spatial dimensions plus one scale dimension.

3.3 Behavior of Scale-Space

Scale-space possesses a number of useful and interesting properties whose examination clarifies what it means for a shape event to be "at a certain scale." The maintenance of these desirable properties may depend upon the enforcement of certain definitions and conventions over the computational operations that act upon the scale-space data structure.

3.3.1 Self-Similarity Across Scales

The principle quality offered by scale-space is *self-similarity across scales* [Burt and Adelson, 1983]: it is most convenient that a computation performed on any shape of a given size yields the same results as the same computation performed on an identical shape that has been uniformly magnified (or reduced) in size. For example, the tests establishing whether four line segments are arranged as a square—adjacent edges perpendicular, opposite edges lie at a distance equal to their lengths, ratio of diagonal to edge length equals $\sqrt{2}$, and so forth—should be the same no matter how large or small the square is.

The most important implication of the self-similarity principle is that computations on scale space should be defined so that magnifications in the spatial dimensions correlate with uniform translations in the scale dimension. Figure 14 illustrates in the case of a simplified scale-space consisting of

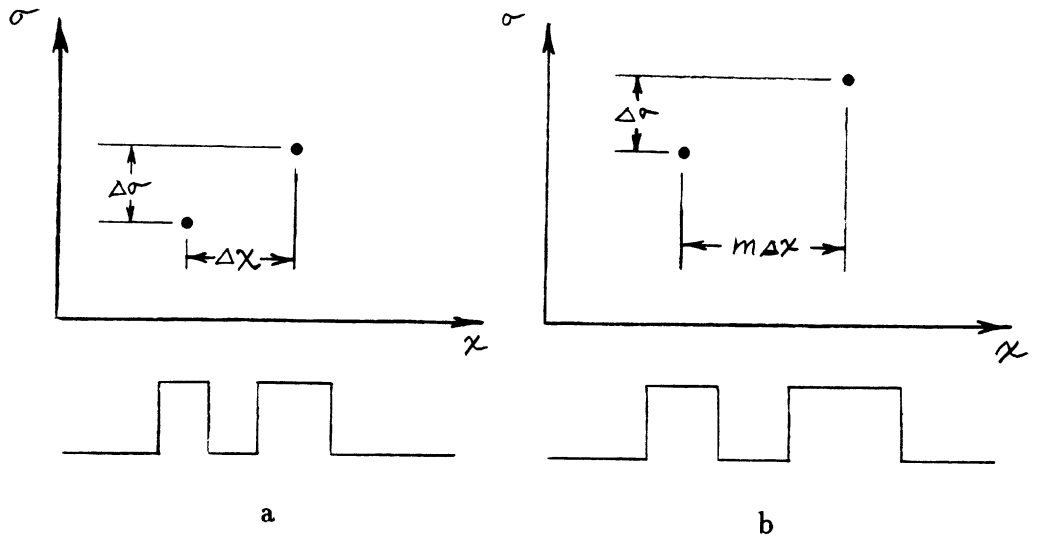


Figure 14. a. A one-dimensional figure composed of two binary pulses. b. The same figure magnified in the spatial dimension by a factor, m. Scale-space images of these shapes are shown above. Each pulse is depicted as a dot, and the width of the pulse determines the dot's placement along the scale (σ) dimension. The principle of self-similarity across scales dictates that when the relative distance of shape features is preserved, their distance along the scale dimension ($\Delta\sigma$) is also preserved.

a scale dimension and only one spatial dimension. Two shape features possessing different sizes and spatial locations are represented as tokens placed at different scales and spatial locations in scale space. Call their proximity in scale space, $(\Delta x, \Delta\sigma)$. Now, take the original shapes and simply magnify the picture by a factor, m. Obviously, the features each grow in size, and the distance between them increases by this factor, but, their *relative distance* (distance relative to size) does not change. Under the self-similarity principle, the scale space image of this new picture places tokens in proximity to each other, $(m\Delta x, \Delta\sigma)$; the shape features' preserved relative sizes becomes manifest as a preserved distance along the scale dimension.

In order to enforce this property the scale dimension is graduated on a logarithmic scale [Witkin, 1983; Schwartz, 1980]. Consider a shape event, for example, an edge primitive, occurring at some reference scale, $\sigma = 0$. The placement along the scale dimension of another edge primitive which is identical to the first, but uniformly magnified by a factor, m, is given by:

$$\sigma = A \log m, \tag{3}$$

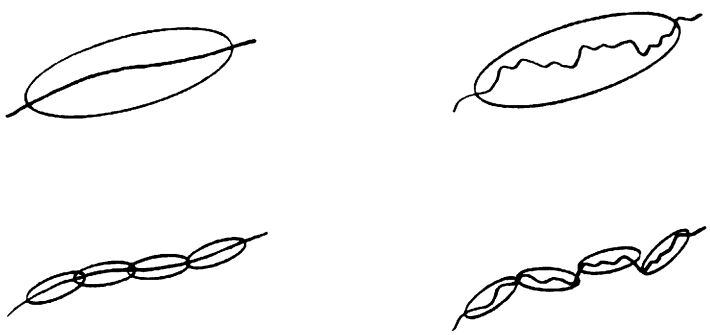


Figure 15. At coarse scales a long smooth edge and a long jagged edge appear identical. Only at finer scales do edge primitives obtain sufficient resolution to distinguish smaller scale detail.

where A is a constant.

Another significant consequence of the self-similarity principle is that precision in the specification of a spatial event's spatial location depends upon the scale of that event. Suppose that some tolerance is associated with stating the exact placement, in x and y, of a token denoting a primitive edge. This tolerance region may for convenience be considered equivalent to the region of space described by a shape token (figure 13). Then self-similarity implies that this tolerance region grows proportionally with the size of the edge primitive. This is to imply that a large scale edge primitive alone does not precisely localize the boundary of the shape object that gave rise to it.

Further implications arise concerning the meaning contained by the assertion of a primitive shape event occurring "at scale σ". As illustrated in figure 15, a long, well defined edge, and a long jagged edge, appear at coarse scales as identical in terms of edge primitives. It is only when one examines medium and finer scale information that descriptive edge primitives obtain sufficient precision to discriminate between these two shape events. Thus,

a complete description of even a geometrically simple shape object must involve analysis of information across a wide range of scales. For example, the description of a long, straight contour boundary, in terms of tokens denoting edge primitives placed on a Scale-Space Blackboard, will be comprised of a collection of tokens lying all along the boundary, *and* at various depths in the scale dimension.

The Scale-Space Blackboard leaves open the possibility of inventing more complex types of tokens that integrate shape information occurring over several scales.

3.3.2 Scale-Normalized Distance

The measurement of distance plays an integral role in the analysis and interpretation of shape. In order to conform to the principle of self-similarity across scales, it is necessary that computations involving distance measurements among shape tokens in the Scale-Space Blackboard be able to take into account the relationship between distance and scale. Just stating that two edge tokens are parallel and lie at 2cm distance from one another does not complete the story, for if they are both fine scale tokens then they could have arisen from opposite ends of an object, while if they are both coarse scale tokens they must by necessity be asserting virtually the same information (see figure 16). Relative distance (distance relative to scale) is the important property, not actual distance.

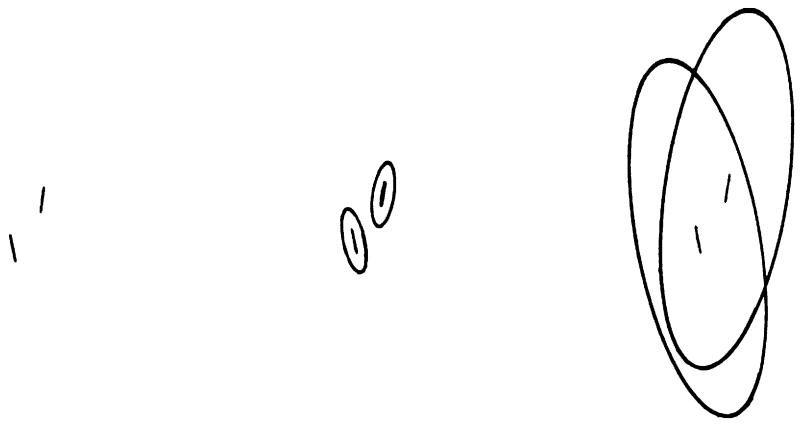


Figure 16. Whether or not the contours described by two edge primitive tokens are fact the same contour depends upon the tokens' scales as well as their relative distance and orientation.

For this reason we define *scale-normalized distance* with the property that the scale-normalized distance between a pair of tokens remains constant as the configuration undergoes uniform magnification. By taking this step, whenever computations take place involving relative distances between shape tokens, scale is automatically taken into account. Some leeway is afforded in the selection of the scale-normalized distance measure. We choose the following:

Definition: The **Scale Normalized Distance (sn-distance)** *between two tokens occurring at scales σ_1 and σ_2, respectively, and separated by a distance* **D**, *is given by*

$$^{\mathrm{sn}}\mathbf{D} = \frac{\mathbf{D}}{\frac{1}{2}(e^{\frac{\sigma_1}{A}} + e^{\frac{\sigma_2}{A}})} \tag{4}$$

The justification for this definition is as follows: If a unit distance is measured at scale $\sigma = 0$, then this distance is magnified at scale σ by a factor, $e^{\frac{\sigma}{A}}$ (inverse of equation (3)). Sn-distance adjusts for the scale of two tokens by dividing the spatial distance between them by the average of their associated magnification factors.

It is instructive to consider the behavior of the sn-distance between two tokens occurring at different scales. Imagine three tokens, A, B, and C, positioned colinearly and as shown in figure 17. Their pairwise distances obey the relationship,

$$\mathbf{D}(A,B) + \mathbf{D}(B,C) = \mathbf{D}(A,C) \tag{5}$$

When the tokens all occur at the *same* scale, their pairwise scale-normalized distances also obey this relationship:

$$^{\mathrm{sn}}\mathbf{D}(A,B) + {}^{\mathrm{sn}}\mathbf{D}(B,C) = {}^{\mathrm{sn}}\mathbf{D}(A,C) \tag{6}$$

But consider what happens when token B increases in scale. Then, by equation (4), the sn-distances distances between tokens A and B, and between tokens B and C decrease, while the sn-distance between tokens A and C remains unchanged. In general, the laws of Euclidian distances as expressed by equation (6) do not hold for scale-normalized distance.

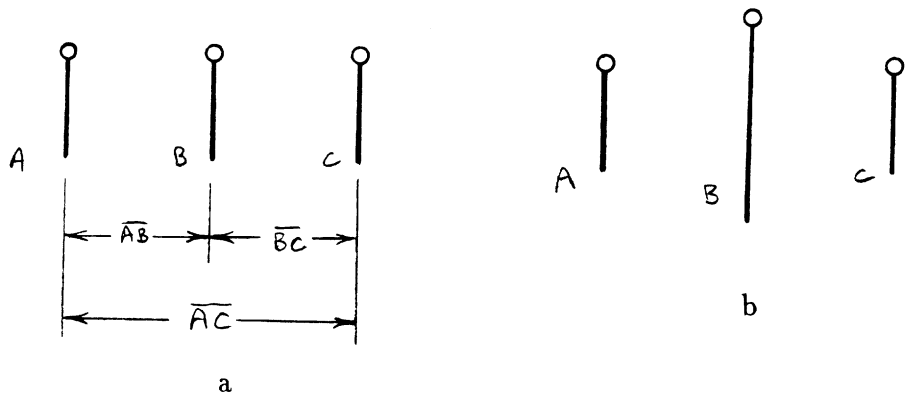


Figure 17. a. When colinear tokens occur at the same scale, then scale-normalized distances behave according to the law, ${}^{\mathbf{sn}}\mathbf{D}(A,B)+{}^{\mathbf{sn}}\mathbf{D}(B,C) = {}^{\mathbf{sn}}\mathbf{D}(A,C)$. b. However, when token B is moved to a coarser scale this relationship no longer holds.

3.3.3 Quantization and Sampling

The x-y-σ Scale-Space Blackboard data structure permits algorithms to index into a shape description on the basis of spatial location and scale. This is conceptually a continuous space. However, for purposes of implementing the Scale-Space Blackboard on a computer, it becomes necessary to quantize the space so that, for example, points in scale-space may be assigned to elements of an array. As a purely practical matter, how might we go about tesselating scale-space?

First, note that as long as shape tokens behave as scraps of paper on which may be written down any information desired, then an appropriate strategy is to include among this list of properties a token's pose in scale-space (spatial location, orientation and scale). Computations involving a token's pose should use this information rather than the quantized array indices specifying the token's address in the Scale-Space Blackboard. This tactic ensures that whatever array quantization scheme is used, its effects may be confined to the efficiency of computation but not the results.

The array quantization issue separates into two: quantization along the spatial coordinates, and quantization along the scale coordinate. Quantization of the scale coordinate will depend in part on how closely spaced along the scale dimension two different shape tokens, specifying different proper-

Figure 18. At a given spatial location, the jagged contour can give rise to edge primitives with different orientations at different scales.

ties, yet occurring at the same spatial location, might be placed. To illustrate the question more clearly, figure 18 shows a figure whose local orientation at a coarse scale is quite different from its local orientation measured at a fine scale. Over how small a distance in the scale dimension might such a phenomenon occur? We present no theoretical analysis but simply relate empirical experience suggesting that a magnification of about a factor of two (one octave) characterizes the rapidity with which the information asserted at one scale can differ from the information asserted at another scale. Thus, scale quantization at steps in the neighborhood one octave or slightly less seem about right.

As for the spatial dimensions, coordinate quantization should accord with the purposes of the algorithms that consult the Scale-Space Blackboard. One of the most common operations is likely to be a query of the form, "Is there a token at pose P?". The purpose in making this query is of course really to discover whether the shape object under analysis displays some spatial event such as an edge at pose P, under the assumption that this spatial event will be represented by a token (or tokens) in the Scale-Space Blackboard. It would therefore seem reasonable to choose a tesselation size in the neighborhood of the range of poses that a token might take in describing a given single localized spatial event, i.e. choose array bin sizes to cover about the same spatial extent as the spatial localization tolerance of a shape primitive (figure

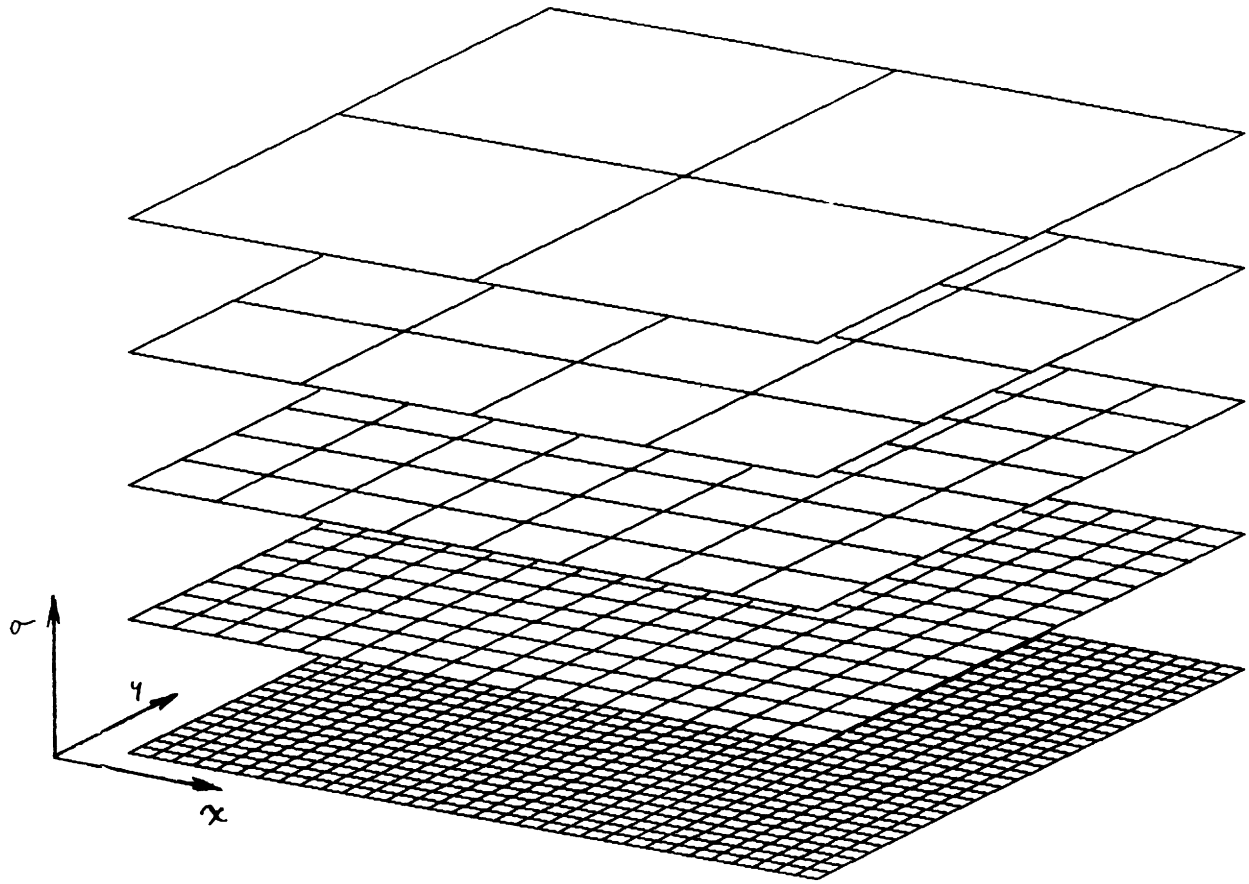


Figure 19. A stack of two-dimensional arrays for implementing the scale-space blackboard. Each array bin holds a list of tokens falling within its domain of scale-space. Coarser tesselation at coarser scales gives resemblance to a pyramid data structure.

13).

Note that individual elements or bins in the array maintaining the contents of the Scale-Space Blackboard may contain not just one but several tokens. Note also that appropriate spatial quantization changes with scale, so that many fewer array elements need be provided per unit area at coarse scales than at fine scales. A suitable picture is of a collection of two-dimensional arrays stacked at octave distances along the scale dimension, as shown in figure 19. This data structure closely parallels pyramid style image representations [Sammet and Rosenfeld, 1980; Burt and Adelson, 1983].

4 Multiscale Description by Fine-to-Coarse Aggregation

We are now equipped to offer a procedure for building a multiscale shape description one scale at a time, from fine scales to coarse. A shape is at this early stage described in terms of edge primitives possessing the attributes of location, orientation, scale, and strength. A token's strength attribute indicates something like "how good" an edge is present at the token's pose. The objective for the fine-to-coarse aggregation procedure is to place "good" edges at successively coarser scales, starting with primitive edge tokens placed at intervals along the shape object's boundary contour at some initial (finest) scale. The aggregation procedure iterates, proceeding from fine scales to coarse, until a desired coarseness of description is reached.

The design of a fine-to-coarse aggregation procedure is motivated by considering configurations of edge primitives that give rise to good coarser scale edges. A sampling of prototypical situations is presented in figure 20.

Figure 20a is the simplest case. A collection of finer scale edges that align with one another give rise straightforwardly to a coarser scale edge. Note in

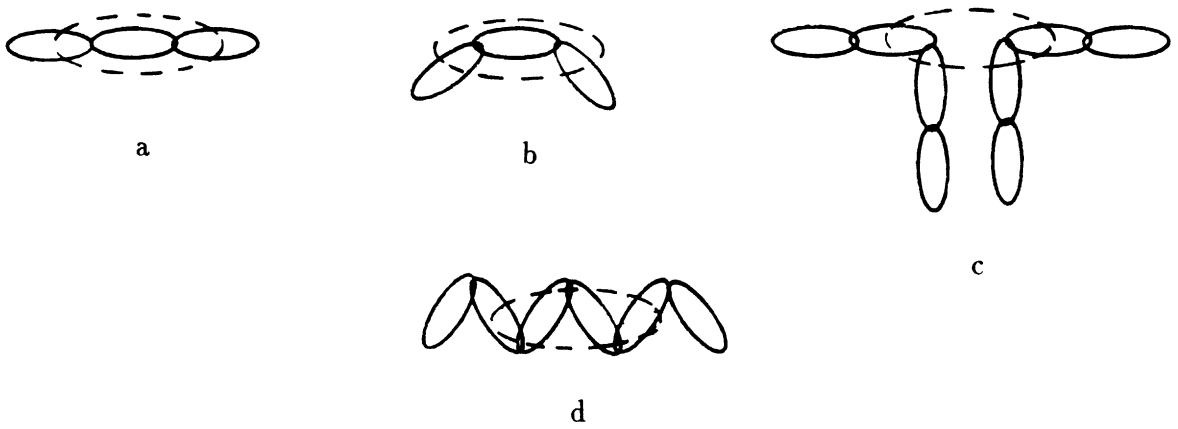


Figure 20. Configurations of finer scale edge primitives (solid ellipses) supporting assertions of edge primitives one octave coarser in scale (dashed ellipses).

this figure that the portion of the image that a given edge token describes may overlap with that of other edge tokens. The spacing of primitive edge assertions along a contour is a free parameter of the representation. For reasons elaborated below, we find it useful for one edge primitive to overlap the next by about 50% of its length.

Figure 20b shows that a section of curved contour gives rise to edge tokens very well aligned with one another at fine scales, but with increasing orientation difference at coarser scales. We suggest that coarser scale primitive edges associated with curved contours be considered weaker than edge primitives associated with straight contours, in much the same way that a coarse scale oriented edge filter would give a weaker response to a curved contour than to a straight edge.

Figure 20c illustrates that a broken contour appearing at a fine scale as two aligned yet disparate portions of a shape may nevertheless be described by a single edge primitive at a coarser scale. This is to say, the pattern matching methods deciding where coarse scale edges are to be placed must be able to identify pairs of finer scale edges aligning with one another across a gap or protrusion.

Finally, 20d shows that, when appropriately configured, a collection of fine scale edges may individually have very different orientations from the coarser scale edge that the collection generates. The algorithm described in this paper omits explicit consideration of this type of situation.

4.1 Fine-to-Coarse Aggregation Procedure

The basic step of the fine to coarse aggregation procedure takes as input a set of primitive edge tokens occurring at a single scale, σ_i, in the Scale-Space Blackboard, and it returns a set of new edge primitives at scale σ_c. Let us refer to scale σ_i as the current "input" scale, and scale σ_c as the "coarser" scale. As implemented, the new tokens delivered are one octave coarser in scale than the input tokens, though the algorithm does not depend upon this rate of aggregation. The basic step proceeds in four smaller steps:

I. Identify seed poses for new coarser scale tokens.

II. Starting from the seeds, refine the placement of new coarser scale tokens based on primitive edge tokens occurring at the input scale.

III. Determine the strengths of these coarser scale tokens.

IV. Prune redundant coarser scale tokens.

These steps are discussed in turn.

4.1.1 Step I. Identify Seed Poses for Coarser Scale Tokens

A seed pose is an initial guess as to where a coarser scale token might be well placed. Observing figure 20, we introduce seed poses at every primitive edge token at the input scale, and at locations where two primitive edge tokens approximately align with one another across an sn-distance (scale-normalized distance) approximately equal to the twice the length of a token. Call the latter case, "gap-jumping" seeds. The orientation of a gap-jumping seed is taken to be the average orientation of the two input tokens that gave rise to it.

The detection of gap-jumping seeds requires checking of input tokens pairwise to determine whether or not they fulfill the seeding qualifications, i.e. proper distance and alignment (and no other token aligned in between). This operation is assisted enormously by the spatial and scale indexing provided by the Scale-Space Blackboard, as this data structure greatly facilitates the inspection of only tokens lying within some spatial neighborhood.

4.1.2 Step II. Refine the Placement of Coarser Scale Tokens

The second step is, for each seed, to determine the best pose for a new coarser scale token suggested by this seed. Selecting the "best pose" originating from a given seed involves finding a pose that tends to maximize the strength of the resulting coarser scale token while tethering the new pose so that it still "belongs" to the seed.

The general approach of the fine-to-coarse grouping procedure is that a coarser scale description is to be aggregated from the information contained in the finer scales. Accordingly, the algorithm computes a coarser scale token's pose as a weighted average of pose information over some support set of input tokens in the neighborhood of the seed (see figure 21). A question immediately arises as to how each supporting input token associated with a given new coarser scale token is to be weighted relative to the other supporting tokens. The factors influencing this weighting are: (1) the spatial

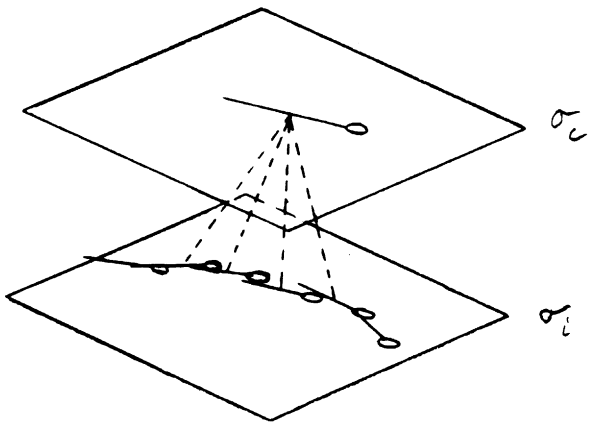


Figure 21. A token at scale σ_c is placed by taking a weighted average of information contained in a set of support tokens occurring at scale σ_i.

relationship between the seed pose and the pose of the supporting input scale token, (2) the proximity of other nearby, possibly redundant, supporting input scale tokens, and (3) this supporting input scale token's strength. These factors are dealt with as follows:

1. Spatial relationship between seed pose and supporting input scale token. Figure 22a shows several possible configurations among a seed pose and the pose of an input-scale token that will have some influence on the placement of a new, coarser scale token initially placed at the seed pose. How should this influence, or weight, be assigned, say, as a number between 0 (low influence) and 1 (high influence)? From figure 22 we reason that influence should: (1) decrease with distance from the seed pose, (2) decrease with distance faster across the orientation of the seed pose than along its orientation, (3) decrease as the relative orientation of the seed pose and the supporting token differ, but (4) less so as their sn-distance decreases. These factors translate into the following expression for calculating the *raw-influence-weight*, W_i', of a token, T_i, occurring at scale σ_i, on the pose of a token, T_c, at the next scale, σ_c, which has been initially placed at its seed

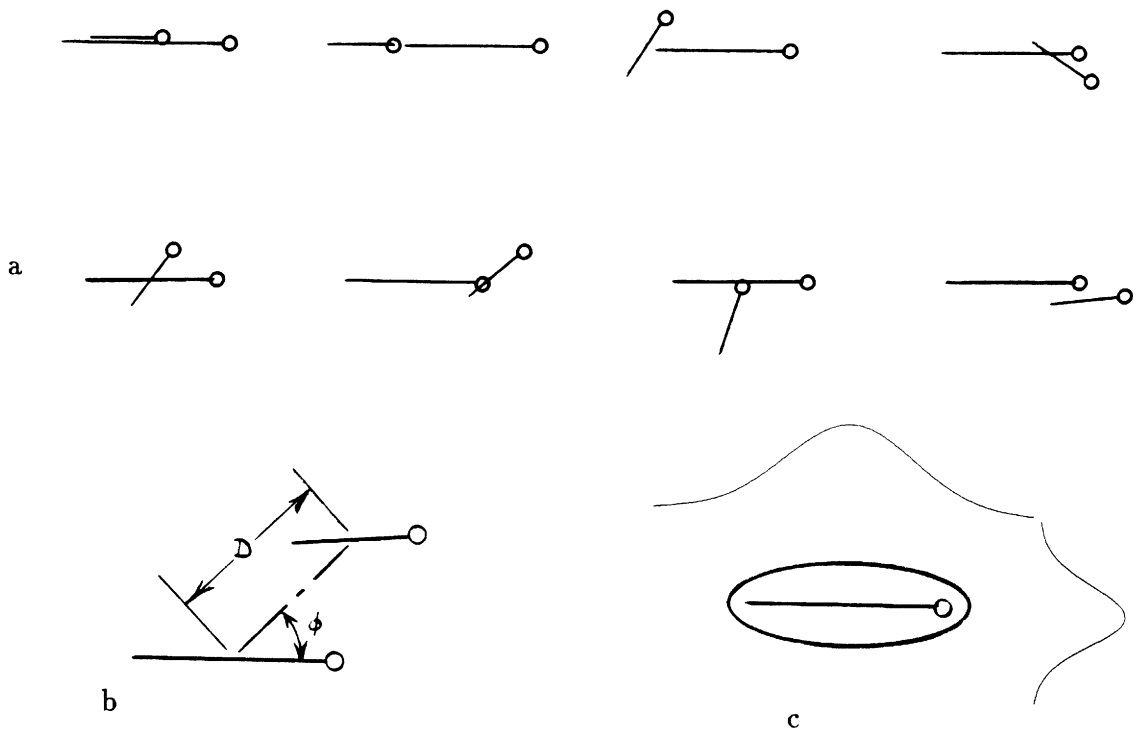


Figure 22. a. A number of possible spatial relationships between a coarser scale token placed at its seed pose (larger line segment) and one of its supporting finer scale tokens (shorter line segment). The supporting token's influence is considered greater when it is near to and aligned with the seed pose. b. The distance, D, and angle, ϕ, entering into the Gaussian weighting ellipsoid, $G({}^{\mathrm{sn}}\mathbf{D}, \phi_{c,i})$, shown in c.

pose:

$$W_i' \leftarrow G({}^{\mathrm{sn}}\mathbf{D}, \phi_{c,i})[1 - \min(1, B\ {}^{\mathrm{sn}}\mathbf{D}^p)|\sin \Delta\theta_{c,i}|], \tag{7}$$

where ${}^{\mathrm{sn}}\mathbf{D}$ is the sn-distance between the seed and the supporting input scale token, $\phi_{c,i}$ is the direction from token T_c to token T_i, $\Delta\theta_{c,i}$ is their relative orientation, and $G(\mathrm{D}, \phi)$ is an ellipsoidal two-dimensional Gaussian weighting function with major axis aligned with $\phi = 0$ (see figures 22b and c). B and p are positive constants. The ellipsoidal Gaussian weighting function has maximum value 1 when $G = 0$, and it trails off to 0 at infinity. This ellipsoid's aspect ratio is a free parameter, for which the value 4 : 1 has been found to serve acceptably. The term in brackets drops below 1 only when tokens are relatively distant and have substantially different orientations.

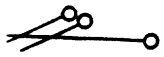


Figure 23. The two smaller scale support tokens supply redundant pose information.

2. The proximity of nearby, possibly redundant, supporting input scale tokens. Figure 23 presents a situation in which two input scale tokens are very near to one another, and would contribute similar influence on the pose of a coarser scale token initiated at the seed pose shown. The information that these two tokens offer about the underlying finer scale shape is redundant, and these two tokens should not both share equal weight with other tokens providing very different information. Some scheme is required causing the information from input tokens located very near one another to saturate in their collective influence upon the pose of the coarser scale token under construction. This effect is achieved by the following procedure:

I. Sort supporting input tokens by decreasing *raw-influence-weight*, W'.

II. For input token T_i, identify the supporting input token, T_j, that: 1. has greater or equal *raw-influence-weight*, and 2. is most similar in pose. Pose similarity, L, may be estimated by the following expression:

$$L(T_i, T_j) = G({}^{\mathbf{sn}}\mathbf{D}, \phi_{i,j}) \cos \Delta\theta_{i,j} \tag{8}$$

III. Choose the value of the *modified-influence-weight*, W_i'', for token T_i in such a manner that it decreases according to its degree of similarity to its most similar stronger neighbor, T_j:

$$W_i'' \leftarrow W_i'(1 - L(T_i, T_j)) \tag{9}$$

3. Strength of this supporting input scale token. The *influence-weight* of a supporting input scale token on the pose of a coarser scale token should be proportional to the primitive edge strength, S_i, of that input token.

Thus, finally, the *influence-weight*, W_i, of an input scale token T_i on a given coarser scale token is expressed by

$$W_i \leftarrow S_i W_i'' \tag{10}$$

Once the *influence-weights* of all of its supporting input scale tokens have been established, then the pose of each new coarser scale token may be determined. The new token's (x, y) location can simply be taken as the weighted average of the (x, y) locations of supporting tokens, and its orientation as that providing best alignment with the locations of the supporting tokens, in the least-squares sense. If desired, it is possible to devise formulas assigning the coarse scale token's orientation on the basis of the aggregate orientations of the supporting tokens as well as their locations.

4.1.3 Step III. Determine Coarser Scale Token Strength

Under the Scale-Space Blackboard representation, the qualitative presence or absence of a descriptive token such as, for example, an edge primitive, is to be modulated with an indication of how strongly the token asserts that its attribute is actually present, at a corresponding pose, in the shape object under observation. This is the token's strength parameter. Every seed generated in step I leads to the placement of a coarser scale shape token in step II. However, some of these coarser scale tokens represent better primitive edges than others. Figure 24 presents a few examples of situations in which the assertion of a coarser scale edge is more strongly or more weakly supported by the finer scale edges present. Step III assigns a strength, S, $0 \leq S \leq 1$, to every newly created coarser scale primitive edge token.

Reasoning from the examples in figure 24, a coarser scale edge is strongly supported when finer scale edges are aligned all along its length. Strength decreases when: (1) the orientations of supporting finer scale edges deviate from that of the coarser scale edge, and when (2) supporting tokens fail to span its entire length. A mathematical expression reflecting these criteria is:

$$S \leftarrow \min\{1, [\min(V_{sum}, C) + \min(V_{front}, C) + \min(V_{rear}, C)]\}, \tag{11}$$

where C is a positive constant. V_{sum} is a sum over all supporting tokens, T_i, of each supporting token's contribution to the strength of the new coarser scale token.

$$V_{sum} = \sum_i V_i \tag{12}$$

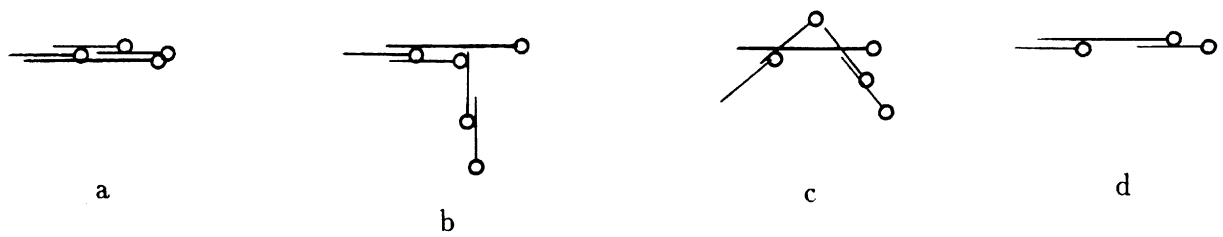


Figure 24. A coarser scale token is assigned a strength according to whether finer scale tokens are aligned with it all along its length. The situation in a. receives greater strength than in b., c., or d.

$$V_i = W_i^p \cos^q \Delta\theta_{c,i}, \tag{13}$$

where p and q are positive constants, and $\Delta\theta$ is the difference between the orientation of the coarse scale token and that of the supporting finer scale token, T_i. The use of the *influence-weight*, W_i, ensures that redundant supporting tokens do not unduly influence the strength computation. The terms, V_{front} and V_{rear} in equation (11), weigh support at the two ends of the coarser scale edge, as follows:

$$V_{front} = \sum_{i_{front}} V_i |{}^{\mathbf{sn}}\mathbf{D}_{proj}| \tag{14}$$

$$V_{rear} = \sum_{i_{rear}} V_i |{}^{\mathbf{sn}}\mathbf{D}_{proj}| \tag{15}$$

${}^{\mathbf{sn}}\mathbf{D}_{proj}$ is the scale-normalized distance between supporting token T_i and the new coarse scale token, projected onto the length axis of the coarse scale token (see figure 25). Equation (11) is constructed so that in order for a token to receive a maximum strength of 1, it must receive substantial support along its entire length.

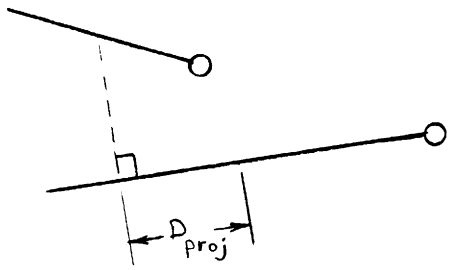


Figure 25. D_{proj} is the distance from a token to a reference token, projected onto the reference token's length axis.

4.1.4 Step IV. Subsample the Coarser Scale Description

By the principle of self-similarity, coarser scale edge primitives describe larger portions of a shape image than do edge primitives occurring at finer scales. Also, they are proportionately less precise in specifying absolute spatial location. Therefore, the coarse scale description of a shape employs tokens more sparsely distributed across the shape image than does a fine scale description. This is analogous to the case in signal processing, in which the sampling required to reconstruct a signal depends upon its bandwidth.

The procedure for generating coarse scale tokens creates a new token at every seeded location. When the jump in scale is one octave, approximately twice as many coarse scale tokens are generated as are necessary. While this should not be harmful to later computations for any fundamental reasons, it is wasteful, and it adversely affects the perspicuity of the coarse scale shape description. For this reason the fourth step in the fine-to-coarse aggregation procedure is to prune the coarse scale shape description so that tokens overlap one another by approximately 50% of their length.

The design of a procedure for subsampling the coarser scale description follows three guidelines: (1) prune tokens of weaker strength first, (2) prune a token lying very near another token in location and orientation, (3) prune a token closely sandwiched between and aligned with two other tokens. See

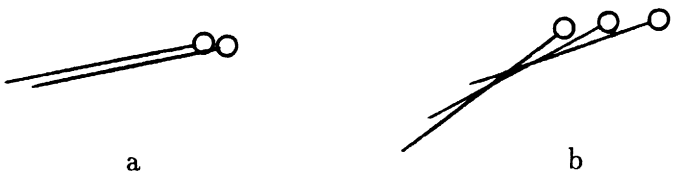


Figure 26. Tokens are pruned, weakest first, when they: a. lie very near in pose to another token, or b. are sandwiched between other tokens.

figure 26. A satisfactory algorithm is the following:

I. Sort tokens by decreasing strength, S.

II. In three passes through the sorted list of all tokens, remove tokens falling under criteria 2. and 3.

The three passes are taken with increasingly stringent bounds on how near to another token a given token may not be. Taking several increasingly severe passes has been found helpful in ensuring that weaker tokens which may perhaps yet describe important nuances in shape are not prematurely stomped out by stronger tokens.

4.2 Results

Performance of the fine to coarse edge primitive aggregation procedure is illustrated in figures 27 though 30. As seen in figure 27, the coarse scale description of the apple survives well even when the contour is interrupted by the protrusion of a string (figure 27d), and when other large objects are in proximity (figure 27b). In figure 27c, when the banana moves close enough to occlude part of the apple's contour, much of the apple's boundary in the vicinity of the banana is nonetheless detected at coarser scales.

Figure 28 helps to illustrate the fact that as scale increases, primitive

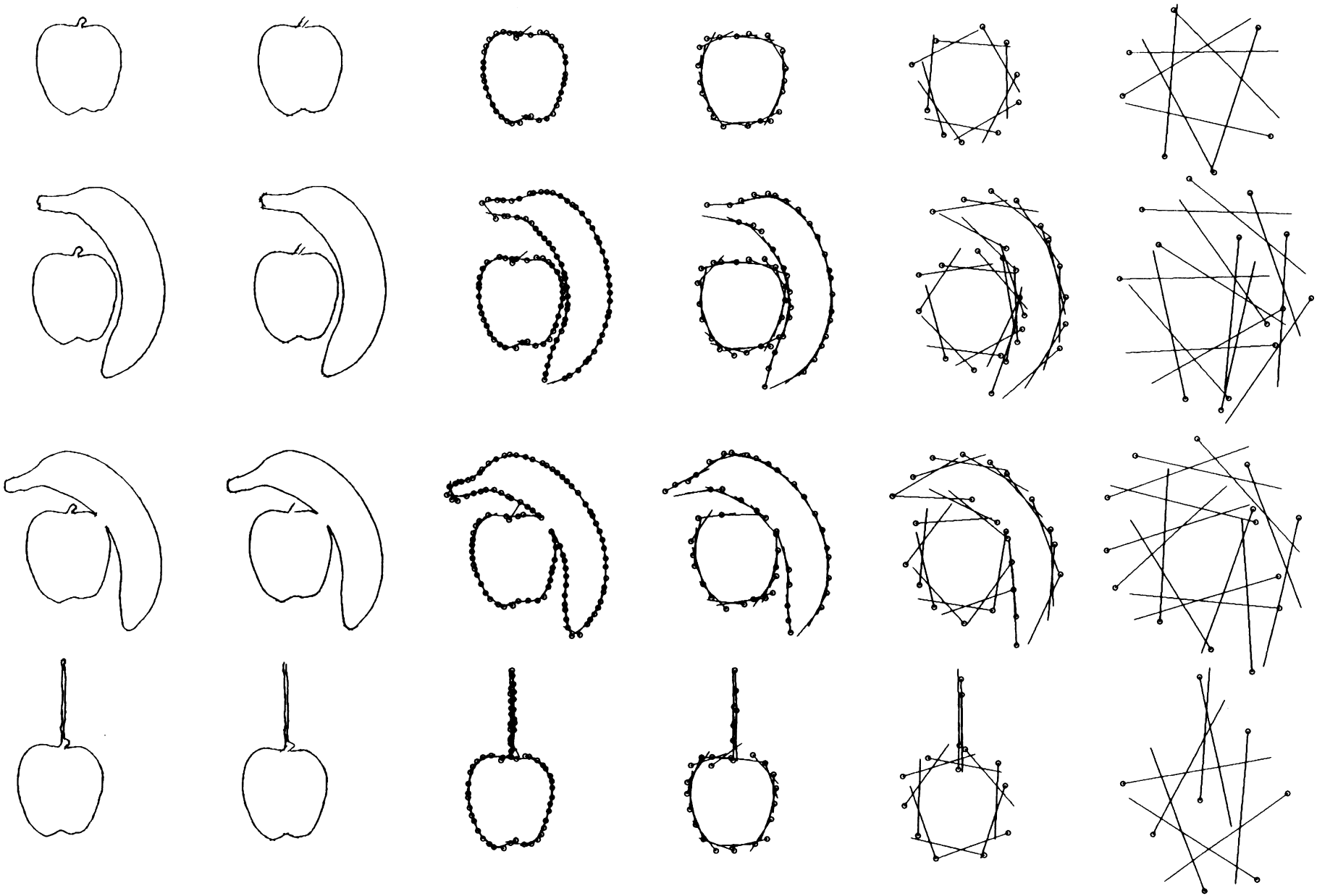


Figure 27. Primitive edge tokens at six scales found by fine-to-coarse aggregation.

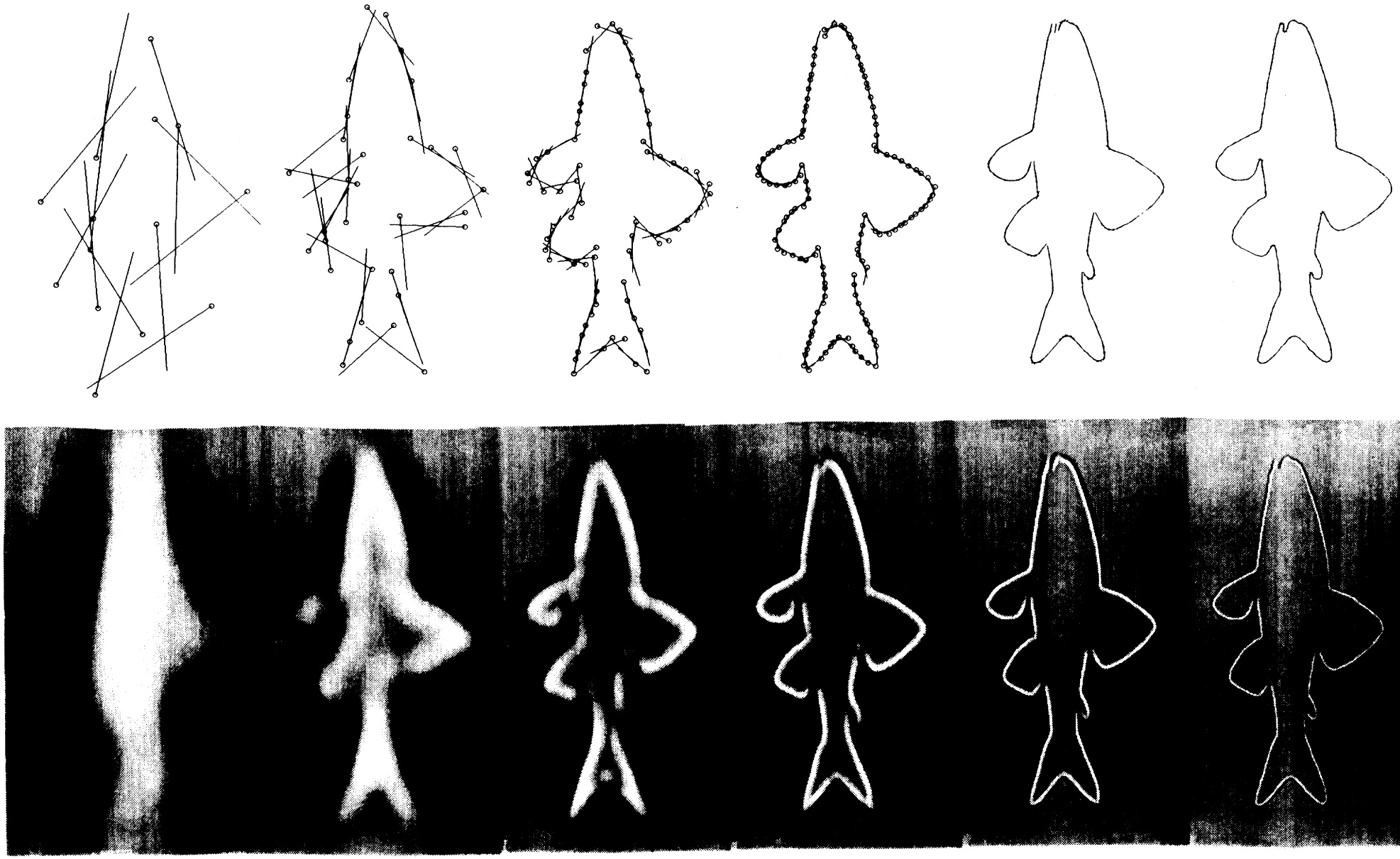


Figure 28. A grey-level image "reconstructed" from the tokens found across scales for a Trout-Perch shape. A light and dark region were colored into the array on the figure and ground side of each token, respectively.

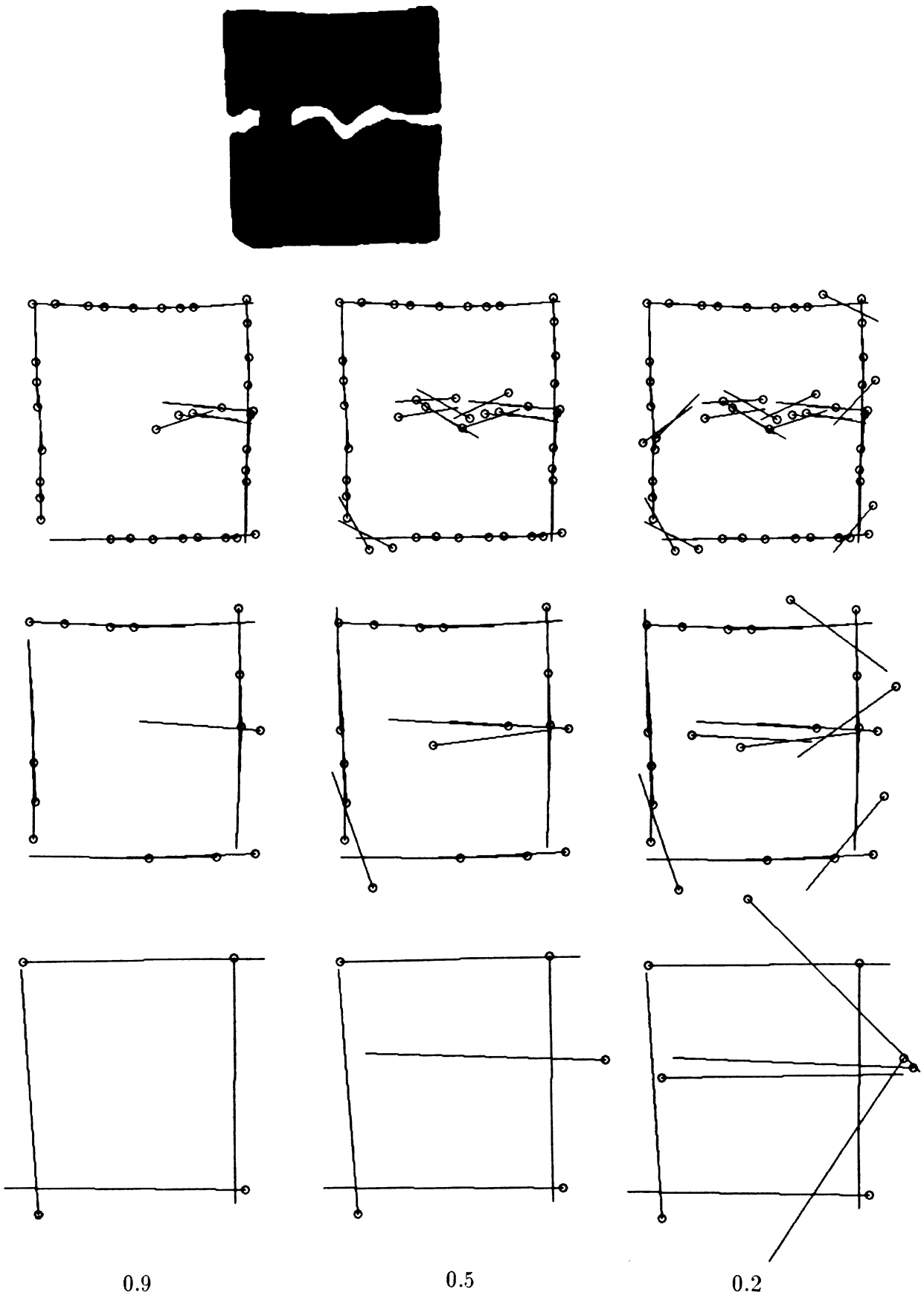


Figure 29. Edge primitives are assigned a strength between 0 and 1. Tokens stronger than a threshold are displayed at three scales, for threshold values 0.2, 0.5, and 0.9. Tokens aligning with well defined figure/ground boundaries are stronger.

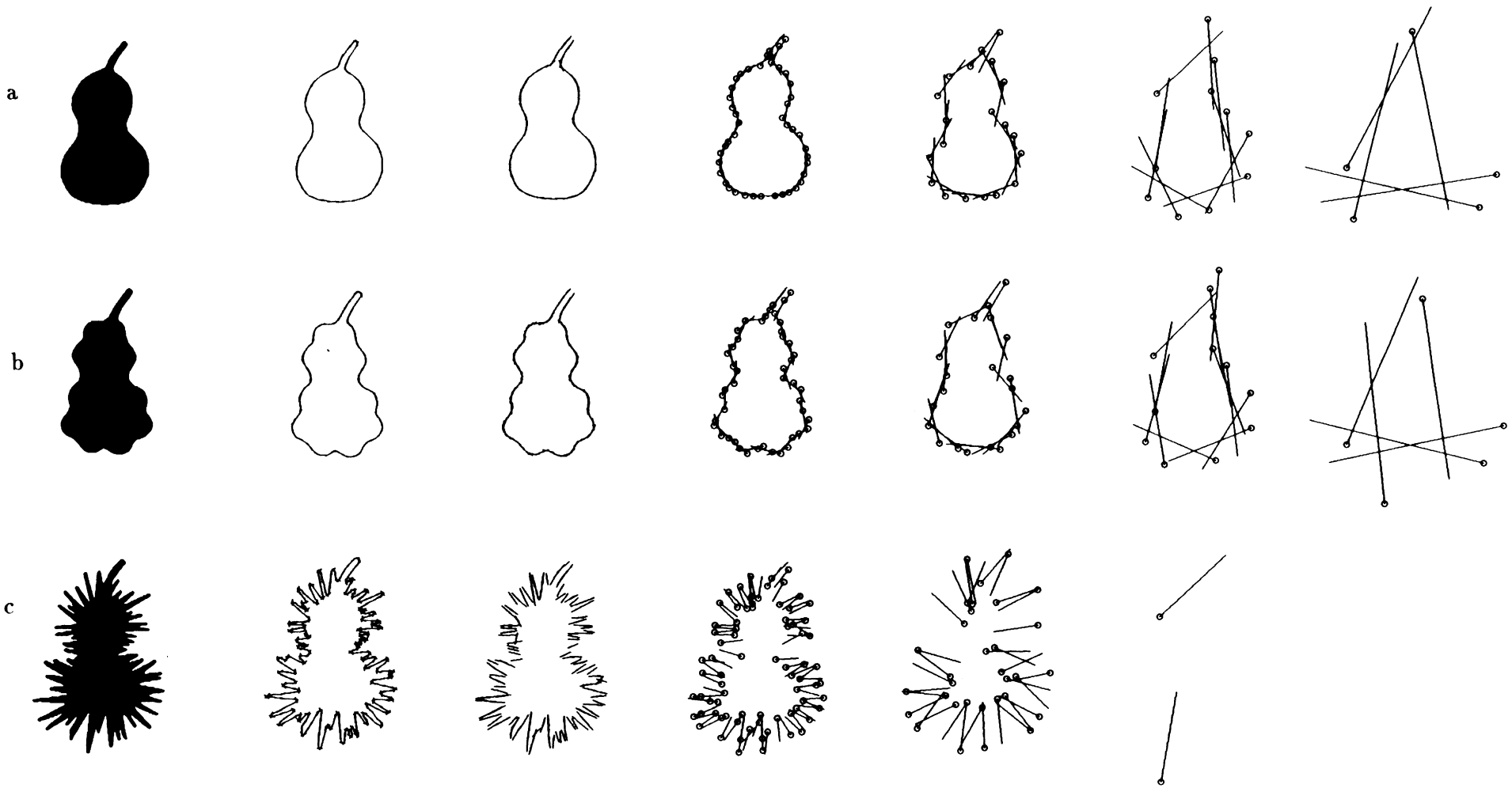


Figure 30. Edge tokens found by fine-to-coarse aggregation for a smooth, rippled, and spiny pear (from [Richards, et al, 1986].) The algorithm described does not identify the coarse scale structure of the spiny pear because it has no rules for grouping tokens aligned perpendicular to their orientations.

edge tokens demark figure/ground boundaries of decreasing spatial resolution. This figure depicts grey-level images "reconstructed" from the tokens residing in each of six slices of the Scale-Space Blackboard. For each token, a lightened region (figure) and a darkened region (ground) were colored into an 8-bit image on either side of each token. For convenience, the light/dark colored region for each token takes the form of the oriented filter mask shown in figure 8. As the pseudo-blurred images show, at coarser scales the primitive edge information describes figure/ground boundaries of greater spatial extent while smaller details of the object's boundary are smoothed over.

In order to illustrate the significance of a token's strength parameter, figure 29 displays edge tokens at three scales using three different thresholds on token strength. As may be observed, coarser scale edges that bridge gaps and cut corners are assigned lesser strength than edges falling along a line of smaller scale edges.

Figure 30 shows a situation in which the aggregation procedure fails to identify coarse scale structure. Note that the smooth pear and rippled pear give rise to nearly identical coarse scale descriptions. However, when the contour texture of the pear is extremely jagged, finer scale edge tokens lie nearly perpendicular to the large scale figure/ground boundary, and are not successfully grouped into coarse scale tokens falling along the boundary. Detection of this sort of contour may be addressed by the development of additional grouping rules, or else by some form of numeric smoothing operation.

We have shown that symbolic processes operating on collections tokens in a Scale-Space Blackboard are able in most cases to construct successively coarser shape descriptions in terms of a simple vocabulary in which tokens denote edge primitives. The Scale-Space Blackboard also supports other interesting grouping operations making explicit more complex shape entities.

5 Pairwise Grouping of Edge Primitives

Symbolic tokens denoting edge primitives are extremely simple, possessing only the attributes of pose (location, orientation, and scale) and strength. Let us refer to these as *Type 0* tokens. This section introduces another class of shape token, called *Type 1* tokens, possessing one additional parameter of internal state. Type 1 tokens are constructed from pairs of Type 0 tokens. The spatial configurations (*Type 1 configurations*) subsumed by this class

of tokens form a continuum which includes shapes that might be called, "curved contour segments," "primitive-corners," and "bars." These terms are elaborated below. In analogy to the fine-to-coarse aggregation procedure, we construct pattern matching procedures to identify Type 1 configurations occurring in the Scale-Space Blackboard, and then mark these occurrences by placing Type 1 tokens appropriately.

5.1 Definition of Type 1 Configurations

Two tokens in scale-space are spatially related to one another by four numbers. These numbers must collectively specify the tokens' relative x and y location, relative orientation, and relative scale. Type 1 tokens possess one internal parameter whose range generates a one-dimensional family of configurations, in other words, a one-dimensional constraint-curve in the four-dimensional space of a pair of Type 0 tokens' relative configuration (see [Saund, 1987]). The definition for Type 1 tokens must therefore constrain or otherwise account for three remaining degrees of freedom.

Type 1 configurations are defined by specifying three constraints on the relative poses of the two component Type 0 tokens: (1) The Type 0 tokens must occur at the same scale, (2) The Type 0 tokens must be symmetrically placed, (3) The Type 0 tokens must lie at a fixed, prespecified, scale-normalized distance from one another.

The first condition, that two Type 0 tokens satisfying a Type 1 configuration must occur at the same scale, is straightforward.

The second requirement states that a Type 1 configuration must be comprised of Type 0 tokens that are symmetrically placed. This condition is illustrated in figure 31; the relative orientations between each token and the line segment joining them must be equal. This specification of angular equality lies behind the definition of the Smoothed Local Symmetries shape representation [Brady and Asada, 1984; Connel, 1985, Fleck, 1985], and has also been called "co-circularity" by Parent and Zucker [1985].

Strictly speaking the first two conditions allow no tolerance for the tokens to differ in scale or to deviate from symmetrical placement by even a slight amount. Obviously, some tolerance is desirable. A potential question arising is then, how much tolerance is acceptable? We handle this question by appealing to a token's strength parameter. The closer to identical scale and perfectly symmetrical alignment a pair of Type 0 tokens are placed, the closer

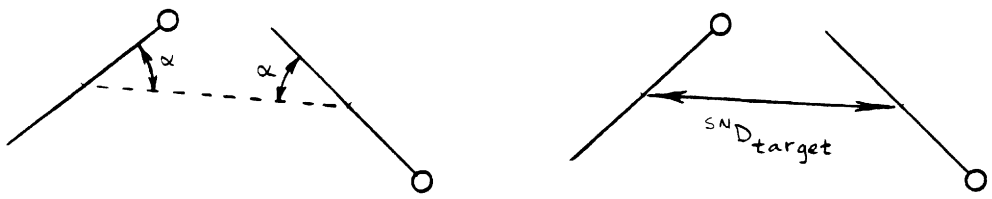


Figure 31. Constraints on the spatial relationship of a pair of Type 0 tokens (edge primitives) if they are to satisfy the Type 1 configuration conditions: a. symmetric placement (co-circularity) b. fixed, predetermined scale-normalized distance. An additional condition is that the Type 0 tokens must occur at the same scale.

to 1 can be the strength of the Type 1 token naming the pair. As the Type 0 tokens stray, the Type 1 token strength must drop to 0.

The third condition suggests that two Type 0 tokens satisfying the conditions of a Type 1 configuration must lie at a characteristic predefined sn-distance, ${}^{\mathbf{sn}}\mathbf{D}_{target}$, from one another. See figure 31. Now, a pair of Type 0 tokens may certainly lie at virtually any (true) distance from one another, depending upon the geometry of the shape object giving rise to it. By equation (4), a given true distance ($\mathbf{D}$) corresponds to another given scale-normalized distance (for example, ${}^{\mathbf{sn}}\mathbf{D}_{target}$) *only* at one particular scale. However, the fine-to-coarse aggregation procedure places Type 0 tokens only at octave intervals in the scale dimension. We cannot guarantee that Type 0 tokens will have been placed precisely where needed along the scale dimension in order to satisfy condition 3 of the definition of a Type 1 configuration.

The resolution to this matter is to note that a shape description does not change rapidly across scales. In other words, the orientation and strength attributes computed for a primitive edge token at one scale would be almost identical to those of a primitive edge positioned at a closely nearby scale. Therefore it is fair to adopt the following tactic: pretend that a Type 0 token placed at a given scale generates a virtual set of Type 0 tokens possessing the same (x, y) location and orientation, but placed at *all* surrounding scales within, say, a one-half octave range. Then, Type 1 grouping takes place on

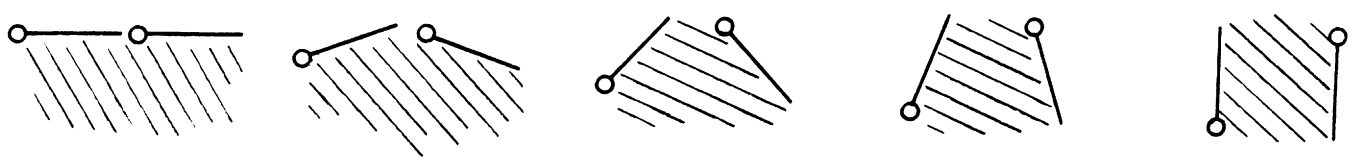
just the pair of virtual tokens required to satisfy condition 3. The resolution amounts to this: place a Type 1 token in scale-space at a scale coordinate depending upon the measured sn-distance between the two component Type 0 tokens. Specifically,

$$\sigma_{T1} = \sigma_{T0} + A \log \frac{^{sn}\mathbf{D}_{T1}}{^{sn}\mathbf{D}_{target}}, \tag{16}$$

where σ_{T1} is the placement of the Type 1 token along the scale dimension, σ_{T0} and $^{sn}\mathbf{D}_{T0}$ are respectively the scale of and scale-normalized distance between the constituent Type 0 tokens, and $^{sn}\mathbf{D}_{target}$ is the characteristic sn-distance defined for the Type 1 configuration.

5.2 The Class of Type 1 Configurations

The internal parameter of a Type 1 token makes explicit one remaining degree of freedom in the spatial configuration of two Type 0 tokens. This degree of freedom is equivalent to the relative orientation of the Type 0 tokens. Figure 32 illustrates the range of configurations generated as this parameter varies. Intuitive interpretations of several of these shapes come readily to mind. When the Type 0 tokens' orientations are roughly aligned, the parameter makes explicit the local curvature of a *curved-contour* segment. When the



curved-contour primitive-corner bar

Figure 32. Members of the class of Type 1 configurations. Each member defines the open boundary of a partial-region.

relative orientation is more or less 90°, the parameter describes the vertex angle of a *primitive-corner*.[5] Finally, when the Type 0 tokens are oriented approximately 180° with respect to one another, the parameter describes the taper of a *bar*. Bars, primitive-corners and to a lesser extent, curved-contours demark local *partial-regions*, as shown by the shaded areas in figure 32. Note that the Type 1 parameter may take either positive or negative values. Parameter values of opposite sign are related by reversal of the figure/ground relationship.

Computation of Type 1 tokens from Type 0 tokens is quite straightforward. Pairs of Type 0 tokens satisfying the three criteria are easily found by virtue of the spatial indexing and scale indexing afforded by the Scale-Space Blackboard data structure. Wherever a Type 1 configuration is found, a Type 1 token is placed at some suitable pose on the Blackboard, such as midway between the constituent Type 0 tokens.

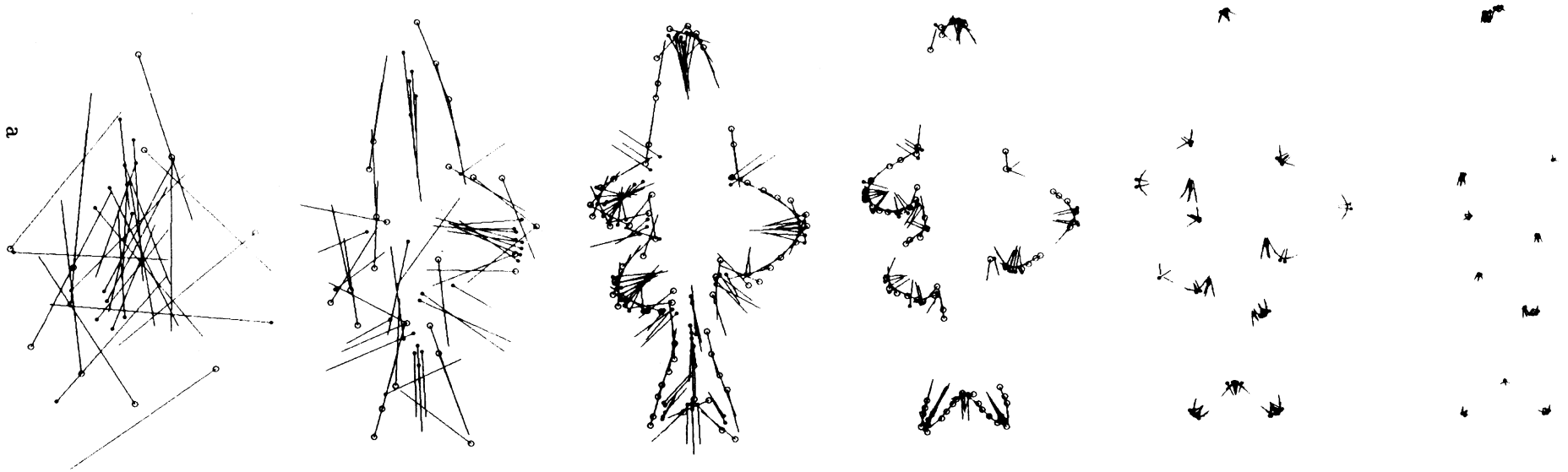
5.3 Results

Figures 33 through 35 present the results of Type 1 token grouping for several shape objects. Each Type 1 token is displayed as a line segment placed at the token's pose in the image, with a small circle at one end indicating its orientation. In addition, the two Type 0 tokens supporting this Type 1 token are also drawn. For clarity, those Type 1 tokens are omitted which describe a gently curved section of contour; only primitive-corners and bars are shown.

Figure 33 shows partial-regions found for a Trout-Perch shape. Note that Type 1 tokens make explicit salient negative or background partial regions, such as the fork of the tail, as well as regions forming parts of the figure itself. These are distinguished by the sign of the Type 1 parameter within each Type 1 token (although this number is not displayed). Figures 34 and 35 show that large scale partial-region description of the body of an apple is not fazed by a radical alteration in the bounding contour formed when the apple is hung from a string, nor by the presence of a nearby object such as a banana.

Figures 33 through 35 also show that the Type 0 and Type 1 grouping rules interpret the scale of regions and the scale of contours in a different

[5]The term, "primitive-corner" is used to emphasize that the Type 1 shape description occurs independently at different scales. The term, "corner" is reserved for future descriptors of corner shapes integrating information across several scales.



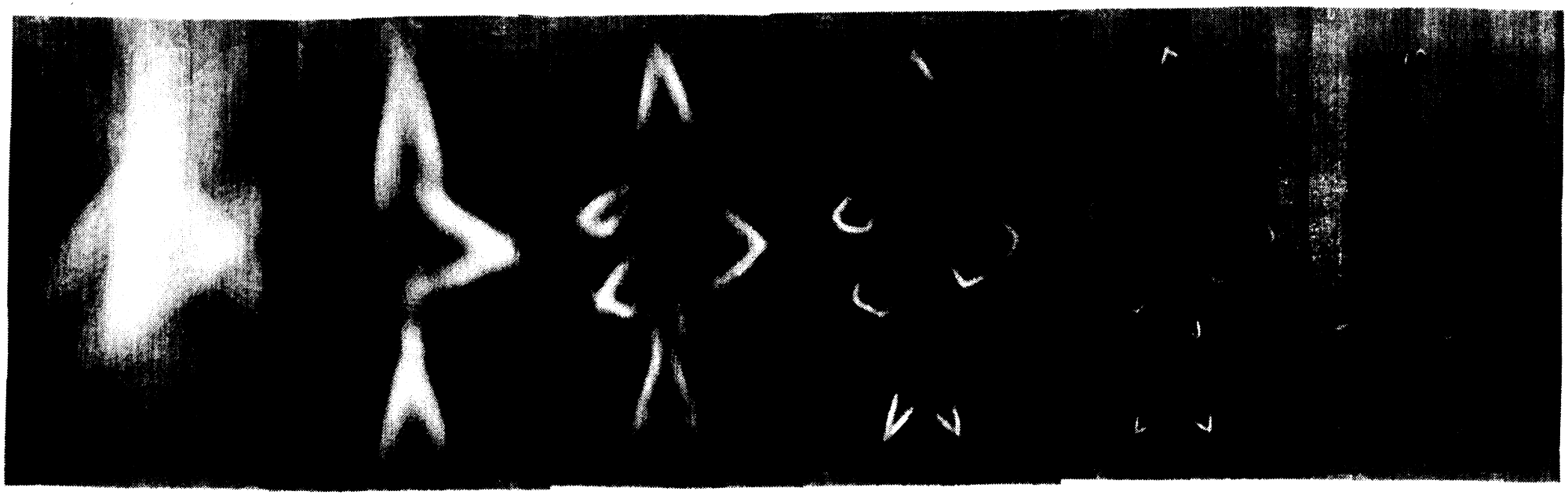


Figure 33. a. Partial-regions (except partial-regions describing curved-contours) found on a Trout-Perch shape. These were computed from the primitive edges shown in figure 27. For each partial region are displayed the Type 1 token itself, plus the supporting Type 0 tokens. b. Grey-level image reconstructed from the Type 0 tokens displayed in 33a. A light and dark region were colored into a grey-level array on the figure and ground side of each Type 0 token, respectively.

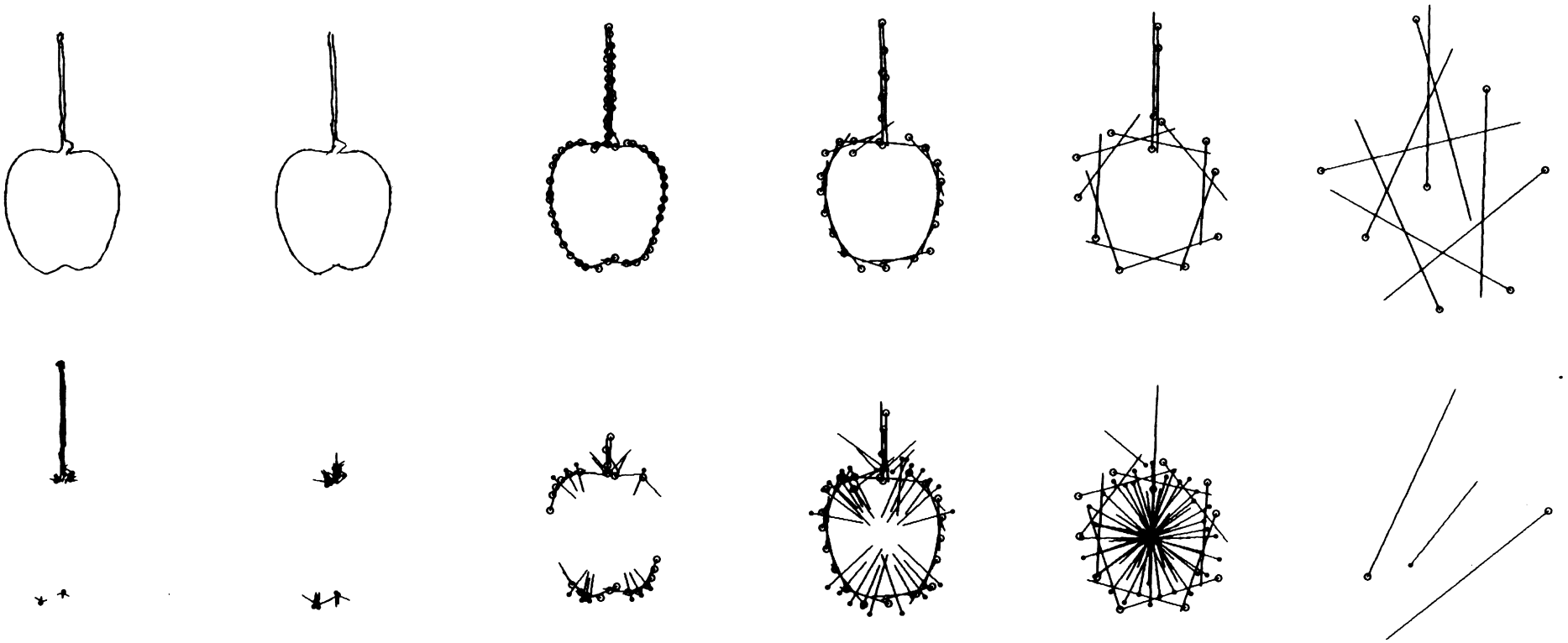


Figure 34. Primitive edges (Type 0 tokens) for the apple-with-string shape, plus partial-regions computed from these primitive edges.

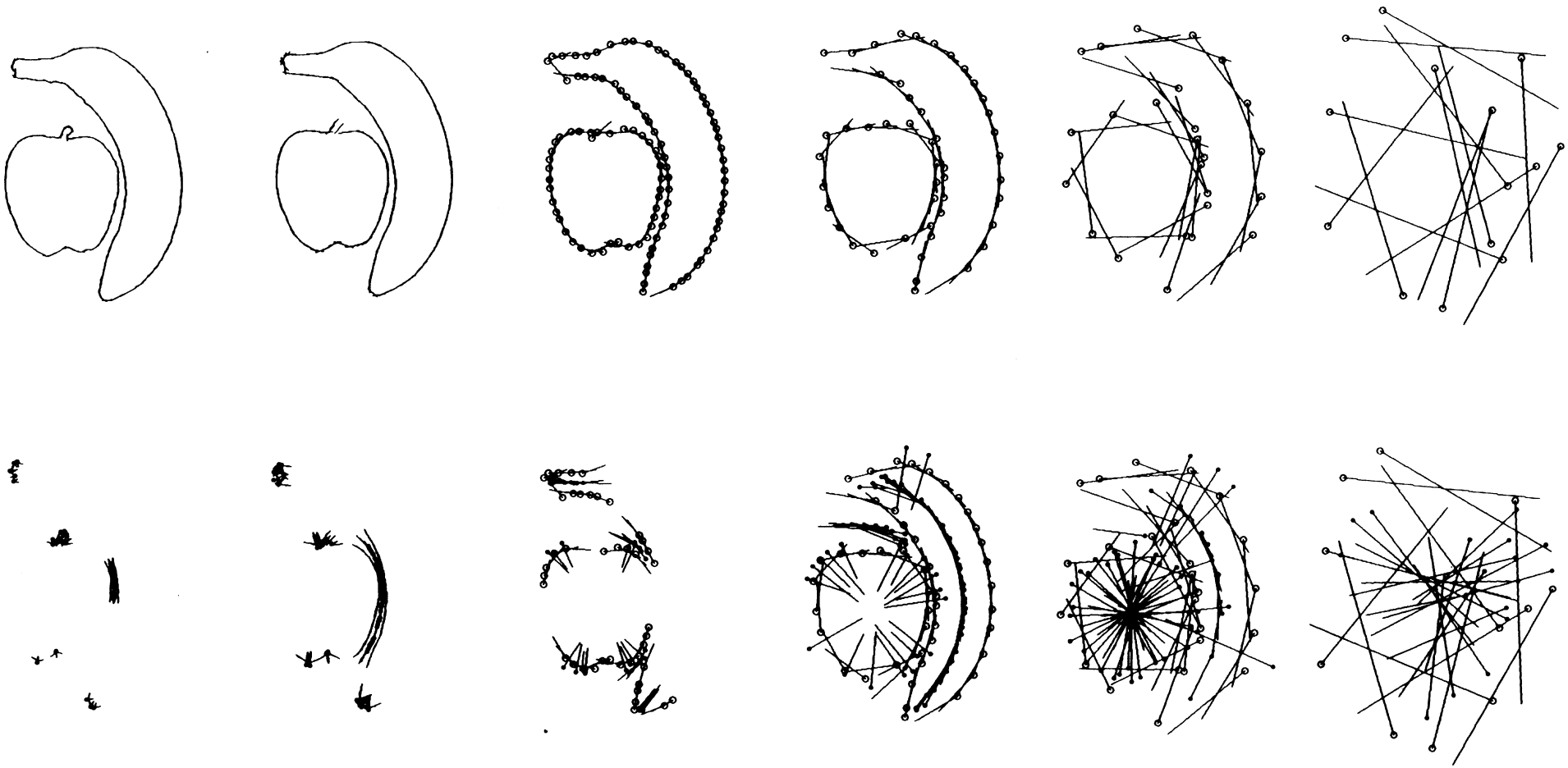


Figure 35. Primitive edges (Type 0 tokens) for the apple-near-banana shape, plus partial-regions computed from these primitive edges.

manner. Type 0 fine-to-coarse aggregation places figure/ground boundaries at a coarse scale if they are of large *linear* (one-dimensional) extent. Thus, the string tied to the apple generates coarse scale Type 0 tokens. In contrast, Type 1 partial-region grouping places shape features at a coarse scale according to their *two-dimensional* spatial extent, or area. Therefore the string, which is of locally small area because of its narrow width, appears only at fine scales in the Type 1 representation.

It is worth noting that one aspect of shape structure not sought by the Type 1 grouping rules is nonlocal symmetry. This is to say, structure is found only at distances commensurate with the scale of the tokens being grouped. In particular, at this early stage no attempt is made to identify configurations such as shown in figure 36, where fine scale tokens form a symmetrical pair but are spaced remotely with respect to their scale. This attitude bounds the complexity of the Type 1 grouping operation because it limits the neighborhood within which to search for other Type 0 tokens forming a Type 1 configuration with any given Type 0 token. The spatial and scale indexing provided by the Scale-Space Blackboard provides the substrate mechanism supporting this spatially limited search. Because the neighborhood of a Type

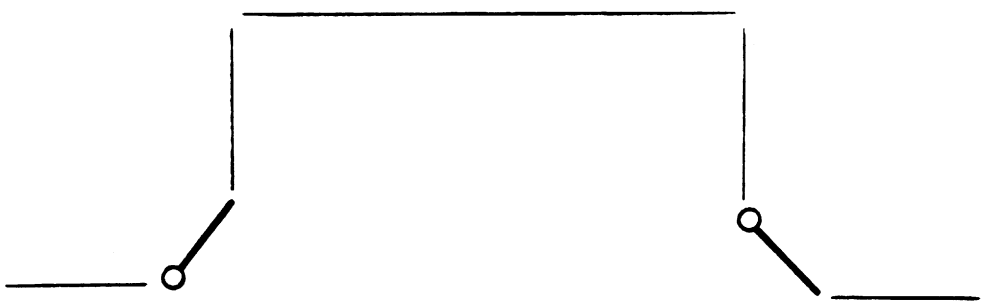


Figure 36. Type 1 grouping does not attempt to group pairs of edge primitives located remotely with respect to their scale.

0 token is defined in terms of scale-normalized distance, that is, that it's absolute size depends upon the scale of the Type 0 token itself, symmetrical configurations spanning large distances *are* identified by the Type 1 grouping rules, but only when their component Type 0 tokens are themselves of a large scale. This scale-relative quality of the computation arises naturally from the property of self-similarity across scales supported by the scale-space representation.

6 Conclusion

This paper has presented an alternative to numerical smoothing or blurring approaches to building multiscale shape descriptions. By performing grouping operations on symbolic shape tokens, coarse scale structure is made explicit based on information present at finer scales of description. Unlike numerical blurring, however, the symbolic grouping rules afford substantial control over just what kinds of coarser scale structure is and is not identified. As a result, the multiscale description of an object's shape retains stability under the presence of other nearby objects, such as when an apple is placed near a banana, and under disruptions of perceptually salient contours, such as when an apple is hung from a string. We acknowledge the importance of treating *regions* and *contours* as complementary aspects of shape geometry, and therefore have designed distinct operations for extracting multiscale contour and region information.

In the course of developing the symbolic grouping approach to multiscale shape representation, we have introduced the Scale-Space Blackboard as a tool for maintaining and accessing spatial information. Shapes are represented in terms of symbolic tokens placed on the Blackboard. This strategy serves as a step toward bridging the gulf between the iconic or image-like representation of a shape implicit in an array of pixels, and later stages of representation making use of purely symbolic data structures. The tokens placed on the Scale-Space Blackboard are symbolic in that they may contain not just a grey-level value, but frame slots, numbers, lists, and pointers, yet the representation is image-like in that the Scale-Space Blackboard provides for indexing of tokens based on location and scale. The use of symbolic tokens, spatially arranged, was first suggested by Marr [1976] in his discussion of the Primal Sketch. Although Marr recognized the significance of scale,

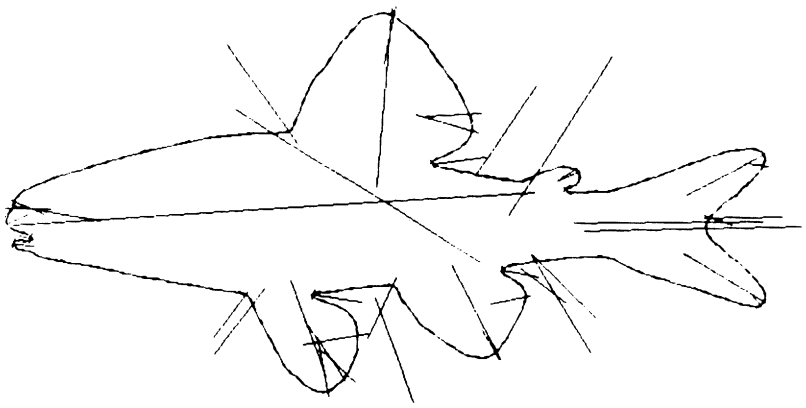


Figure 37. "Spine" axes computed from the Type 1 tokens in figure 33 by a very simple clustering algorithm.

possibility of interpreting scale as a distinct dimension in addition to the spatial dimensions was not elaborated until some years later by Witkin [1983]. This work unites these two ideas. A similar approach to finding extended straight lines in grey-level images is adopted by [Weiss and Boldt, 1986] and [Boldt and Weiss, 1987].

The stage is now set to construct additional procedures operating over the contents of the Scale-Space Blackboard in order to identify more complex and more abstract geometric events and shape properties. These procedures may write new tokens onto the Blackboard, with token types corresponding to the properties they identify. For example, one commonly sought shape description is a listing of an object's "spines," or part axes. Figure 37 shows axes found by performing a very simple clustering operation on the Type 1 tokens of figure 33. These spines are only an illustration that the multi-scale shape description delivered does indeed support the extraction of more

complex shape entities; the proper design of a "spine token" making explicit taper, spine curvature, and so forth is a subject for further work.

Because the Scale-Space Blackboard retains a pictorial quality while the symbolic tokens it contains may represent extended spatial events, or "chunks" of shape, it is not unlikely that this approach to shape representation may also serve as a suitable substrate for elemental visual operations supporting Visual Routines [Ullman, 1983; Mahoney, 1987].

Acknowledgements

This work was carried out under the supervision of Professors Eric Grimson and Whitman Richards. I thank these gentlemen, plus John Connell, Aaron Bobick, and Margaret Fleck for their comments on an earlier draft of the paper.

References

Anderson, J. R., [1978], "Arguments Concerning Representations for Mental Imagery," *Psychological Review*, Vol. 85, no. 4, 249-277.

Asada, H., and Brady, M. [1984], "The Curvature Primal Sketch," *IEEE T. PAMI*, Vol. 8, no. 1, 2-14.

Audubon Society Field Guide to North American Fishes, Whales, and Dolphins, [1983], A. A. Knopf, New York.

Boldt, M., and Weiss, R, [1987], "Token-Based Extraction of Straight Lines," COINS Tech. Report 87-104, UMass, Amherst, MA.

Brady, M., and Asada, H., [1984], "Smoothed Local Symmetries and their Implementation," *Int. J. Robotics Research.* Vol. 3, no. 3, 36-61.

Burt, P., and Adelson, T., [1983], "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Communications*, Vol. 31, no. 4, 532-540.

Canny, J., [1986], "A Computational Approach to Edge Detection," *IEEE T. PAMI*, Vol. 8, no. 6, 679-698.

Connell, J., [1985] "Learning Shape Descriptions: Generating and Generalizing Models," AI Lab TR 853, MIT.

Crowley, J., and Parker, A., [1984], "A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform," *IEEE T. PAMI*, Vol. 6, 156-170.

Crowley, J., and Sanderson, A., [1987], "Multiple Resolution Representation and Probabilistic Matching of 2D Grey Scale Shape," *IEEE T. PAMI*, Vol. 9, no. 1, 113-121.

Daugman, J., [1985], "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am.*, Vol. 2, no. 7, 1160-1169.

Fleck, M., [1985], "Local Region Symmetries," AI Lab TR 852, MIT.

Freeman, H., [1974], "Computer Processing of Line Drawing Images," *Computer Surveys*, Vol. 6 no. 1, 57-98.

Koenderink, J., [1984], "The Structures of Images," *Biol. Cybern.* 50, 363-370.

Kosslyn, S., Pinker, S., Smith, G., and Schwartz, S., [1979], "On the Demystification of Mental Imagery," *Behavioral and Brain Sciences*, Vol. 2, 535-581.

Mahoney, J., [1987], "Image Chunking: Defining Spatial Building Blocks for Scene Analysis," AI Lab TR 980, MIT.

Marr, D., [1982], *Vision*, Freeman & Co., New York.

Marr, D., [1976], "Early Processing of Visual Information," *Proc. Roy. Soc. Lond.* Vol. 275 B, 483-519.

Marr, D., and Hildreth, E., [1980], "Theory of Edge Detection," *Proc. Roy. Soc. Lond. B.* Vol. 207, 187-217.

Mokhtarian, F., and Mackworth, A., [1986], "Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," *IEEE T. PAMI*, Vol. 8, no. 1, 34-43.

Mackworth, A., and Mokhtarian, F., [1984], "Scale-Based Descriptions of Planar Curves," *Proc. 5th Bien. Conf. Canadian Soc. for Computational Studies of Intelligence.*, 114-118.

Parent, P., and Zucker, S., [1985], "Trace Inference, Curvature Consistency, and Curve Detection," CIM-86-3, McGill Univ.

Pizer, S., Koenderink, J., Lifshitz, L., Helmink, L., and Kaasjager, A., [1986], "An Image Description for Object Deformation Based on Extremal Regions in the Stack," in Bacharach, S., ed., *Information Processing in Medical Imaging*, 24-37, Martinus Nijhoff, Boston.

Pylyshyn, Z., [1973], "What the Mind's Eye Tells the Mind's Brain," *Psychological Bulletin*, Vol. 80, no. 1, 1-24.

Pylyshyn, Z., [1981], "The Imagery Debate: Analog Media versus Tacit Knowledge," in Block, ed., *Imagery*, MIT Press, Cambridge, MA.

Richards, W., Dawson, B., and Whittington, D., [1986], "Encoding Contour Shape by Curvature Extrema," *J. Opt. Soc. Am.* Vol. 3, 1483-1492.

Sammet, H., and Rosenfeld, A., "Quadtree Representations of Binary Images" *Proc. Fifth Int. Conf. Pattern Rec.*, 815-818.

Saund, E., [1988], "Dimensionality-Reduction Using Connectionist Networks," IEEE T. PAMI, in press.

Schwartz, E., [1980], "Computational Anatomy and Functional Architecture of Striate Cortex: A Spatial Mapping Approach to Perceptual Coding," *Vision Research*, Vol. 20, 645-669.

Ullman, S., [1983], "Visual Routines," *Cognition*, Vol. 18, 97-159.

Voorhees, H., [1987], "Finding Texture Boundaries in Images," AI Lab TR 968, MIT.

Walters, D., [1987], "Rho-Space: A Neural Network for the Detection and Representation of Oriented Edges," *Proc 9th Conf. Cog. Sci. Soc.*, 455-460.

Weiss, R., and Boldt, M., [1986], "Geometric Grouping Applied to Straight Lines," *Proc. IEEE Conf. CVPR,* 489-495.

Witkin, A., [1983], "Scale-Space Filtering," *Proc 8th Int. Joint Conf. Artificial Intelligence*, 1019-1022.

Zucker, S., and Iverson, L. [1987], "From Orientation Selection to Optical Flow," *CVGIP*, Vol. 37, 196-220.

This blank page was inserted to preserve pagination.

CS-TR Scanning Project
Document Control Form

Date : 4/27/95

Report # AIM-1028

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 59 (65 IMAGES)
Not to include DOD forms, printer instructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:_____

Check each if included with document:
☒ DOD Form (2) ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages (by page number):_____

Photographs/Tonal Material (by page number): 11-15, 17-18, 43, 51

Other (note description/page number):

Description :	Page Number:
IMAGE MAP (1) UNNUM'ED TITLE PAGE	
(2-59) PAGES NUM'ED 2-59	
(60-62) SCANNING TARGET, DOD (2)	
(63-65) TRGTS (3)	

Scanning Agent Signoff:
Date Received: 4/27/95 Date Scanned: 5/2/95 Date Returned: 5/4/95

Scanning Agent Signature: Michael J. Cook

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

completed 8-26-88

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AIM 1028	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: AD-A195926	
4. TITLE (and Subtitle): Symbolic Construction of a 2D Scale-Space Image	5. TYPE OF REPORT & PERIOD COVERED: memorandum
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Eric Saund	8. CONTRACT OR GRANT NUMBER(s): N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Artificial Inteligence Laboratory 545 Technology Square Cambridge, MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209	12. REPORT DATE: April 1988 13. NUMBER OF PAGES: 59
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Office of Naval Research Information Systems Arlington, VA 22217	15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

scale-space
symbolic token grouping
shape representation
scale-space blackboard

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The shapes of naturally occurring objects characteristically involve spatial events occurring at many scales. It is important to make explicit the multiscale structure of a shape object in order to effectively perform shape recognition or to engage in other forms of reasoning about shape. Currently available techniques for multiscale shape analysis include image blurring and contour smoothing; each of these techniques involves uniform application of a smoothing operator to the entire image array or

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 cont.

contour. This paper offers a new, *symbolic*, approach to constructing a primitive shape description across scales for 2d binary (silhouette) shape images. Under this approach, *grouping* operations are performed over collections of tokens residing on a Scale-Space Blackboard. Two types of grouping operations are identified that, respectively: (1) aggregate edge primitives at one scale into edge primitives at a coarser scale, and (2) group edge primitives into partial-region assertions, including curved-contours, primitive-corners, and bars. Algorithms to perform these computations are presented.

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

Scanned

Date: 5/2/1995

M.I.T. Libraries
Document Services