Labeling of Curvilinear Structure Across Scales by Token Grouping

Eric Saund

EDL-92-7 May 1992 [P92-00072]

Abstract: This paper presents an algorithm for labeling curvilinear structure at multiple scales in line drawings and edge images. Symbolic CURVE-ELEMENT tokens residing in a spatially-indexed and scale-indexed data structure denote circular arcs fit to image data. Tokens are computed via a small-to-large scale grouping procedure employing a "greedy", best-first, strategy for choosing the support of new tokens. The resulting image description is rich and redundant in that a given segment of image contour may be described by multiple tokens at different scales, and by more than one token at any given scale. This property facilitates selection and characterization of portions of the image based on local CURVE-ELEMENT attributes.

XEROX

Xerox Corporation
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304

Labeling of Curvilinear Structure Across Scales by Token Grouping

Abstract

This paper presents an algorithm for labeling curvilinear structure at multiple scales in line drawings and edge images. Symbolic CURVE-ELEMENT tokens residing in a spatially-indexed and scale-indexed data structure denote circular arcs fit to image data. Tokens are computed via a small-to-large scale grouping procedure employing a "greedy," best-first, strategy for choosing the support of new tokens. The resulting image description is rich and redundant in that a given segment of image contour may be described by multiple tokens at different scales, and by more than one token at any given scale. This property facilitates selection and characterization of portions of the image based on local CURVE-ELEMENT attributes.

1 Introduction

An important class of structure occurring in images takes form as extended curvilinear contours that appear as straight or gently curving lines. Localized edge and line detectors perform initial measurements of these structures but classically are best suited to detecting image events matched to simple straight step-like or ridge-like intensity profiles. The organization of local oriented edge and line assertions into larger scale entities is an important step toward identifying semantically meaningful units reflecting object boundary contours, shadow edges, material changes, surface markings, or other physical processes that give rise to curvilinear contours.

This paper presents an algorithm for labeling extended curvilinear structure in terms of symbolic assertions, called CURVE-ELEMENT tokens, that denote poses (locations and orientations) where image data may be approximated by a circular arc. The assertion of circular arcs at different locations and sizes constitutes a useful abstraction of curvilinear structure for a number of reasons. A circular arc model is of modest complexity, allowing greater precision of approximation than a straight line, yet considerably simpler than a spline or polynomial model. Circular arc CURVE-ELEMENT tokens "glue together" collections of individual pixel-level line elements so that they may be treated as wholes. These chunks may be assigned aggregate properties of their own, such as curvature and smoothness, that summarize the original information, and CURVE-ELEMENT tokens may be queried, polled, and selected based on their individual properties and their spatial relationships.

As shown in figure 1, good circular arc approximating segments can overlap one another. Because the goal of the present computational stage is to make explicit this type of image structure *wherever* it occurs in an image, we embrace the notion of overlap and allow a given contour segment in an image in general to support the assertion of many CURVE-ELEMENT tokens. Accordingly, CURVE-ELEMENT tokens are maintained at multiple scales of approximately octave (factor of 2) intervals in magnification, and at spatial intervals along a contour equivalent to approximately half their lengths. The target representation thus resembles that of Lowe [16].

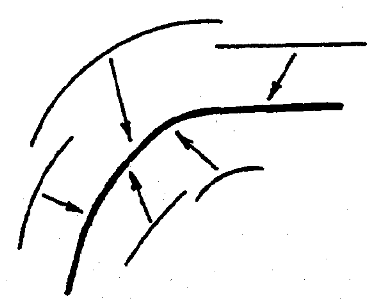


Figure 1: Circular arc approximations to a contour overlap one another.

Previous approaches to the identification of curvilinear structure in images are primarily of four types: (1) global histogram methods; (2) iterative optimization methods; (3) methods involving curve tracing, curve smoothing, and parameterized curve fitting; and (4) grouping methods.

Global histogram methods may be considered variants on the Hough Transform [4, 33]. The shortcomings of these methods for detecting imprecisely modeled data (e.g. arcs that don't form a perfect circle) in the presence of noise are well documented (see [14]). Spatially localized Hough Transforms have, however, been used successfully in conjunction with a line segment grouping strategy [28].

Existing iterative optimization methods [34, 26, 38] operate at a single (finest) spatial scale. Local assertions are influenced by spatially distal information through iteration of a local propagation step. Certain types of image structure, such as smoothly curving contours, become accentuated as specified by an optimization function, and a second step such as curve tracing or snake-like parameterized curve fitting [15] is needed to recover curves as individuated entities. These processes sacrifice self-similarity across scales (does the same output obtain when the original image is simply magnified in size?) because the optimization function is expressed in terms of target spatial structure at only the finest scale.

A number of previous methods rely on an initial curve tracing step to link strings of pixels into chain-coded contours. Then, curve smoothing or parameterized curve fitting techniques are used to obtain lower resolution descriptions or symbolic characterizations of the curve data [3, 12, 13, 17, 19, 21, 24, 25, 27, 31]. These methods have revealed that salient intermediate and large scale curvilinear structure is often only tenuously reflected in the set of purely local pixel neighbor relations, and can therefore be difficult to recover from curves generated by local pixel linking. It becomes necessary to devise methods to break chain-coded curves at potentially significant points such as inflections and curvature extrema [3, 16, 19, 31], and to link initially unrelated curves on the basis of spatial proximity [16, 21]. These latter techniques lead to grouping methods.

Token grouping methods for identifying and labeling significant image structure played a prominent role in the Primal Sketch theory of Marr [20]. Although Marr and his colleagues constructed illustrative demonstrations suggesting that spatial events such as extended contours and blobs could be computed by token grouping, the techniques were never brought to conclusive results. Since then, a number of investigators have developed techniques for small-to-large scale perceptual grouping of chain-coded curves [21, 23], line segments [5, 16, 28, 32, 35] and parameterized curve segments [11, 16]. These investigators' grouping rules rely primarily on pairwise linking and merging of their respective elements based on proximity and alignment. Under grouping approaches, curvilinear structure can be identified by purely spatially local computations even when it arises from topologically disparate image curves.

Key to any token grouping approach is means for deciding which subsets of tokens at one scale are to aggregated into new tokens at a larger scale. This paper reports a new greedy algorithm that, starting from an initial seed token, adds tokens one-by-one to the support set of sufficiently aligning tokens falling in the neighborhood of the seed, until an adaptive threshold on alignment is exceeded. This "greedy" algorithm lies at the heart of a relatively simple and conservative one-pass grouping strategy based on alignment of the arcs represented by nearby tokens. Under this strategy, larger scale CURVE-ELEMENT tokens: (1) perform a smoothing of the image contour, (2) bridge small gaps in image data, and (3) continue a circular arc segment in the presence of extraneous crossing contours. The grouping algorithm is supported by a spatially and scale indexed data structure called the *Scale-Space Blackboard.* This data structure is important because the ability to access tokens on the basis of their locations and sizes simplifies computations concerning spatial relationships among image events, and improves the efficiency of computations on local neighborhoods.

The present work makes the following contributions: First, as mentioned it offers a new grouping procedure attacking in a novel way the problem of choosing the support for new, larger scale, tokens. Second, the definition of CURVE-ELEMENT tokens plus their organization on the Scale-Space Blackboard unifies several advantageous concepts: (1) a spatially and scale-indexed data structure is used to organize data; (2) computation is hierarchical and local with respect to spatial scale, making it amenable to implementation in parallel hardware; (3) token assertions at all scales are preserved and made available for later computation; (4) a single type of descriptor labels both curving arcs and line segments (arcs with curvature = 0); (5) tokens are abstract entities capable of maintaining a great deal of information including pointers or links to neighbor, ancestor, and descendent tokens; (6) self-similarity across scales is achieved by expressing spatial relationships in terms that normalize for scale. This work draws from a previously reported method for multiscale grouping of tokens representing primitive figure/ground boundaries arising from binary shape images [30]. However, the present paper significantly departs from the previous work in two ways: (1) the present CURVE-ELEMENT tokens have different interpretations, include a curvature parameter, and are suitable for line drawings or connected curves derived from edge and ridge detector outputs, and (2) the grouping rules incorporate a greedy algorithm and are therefore different in function, simpler, and more clearly motivated.

2 Tools: CURVE-ELEMENT Tokens in Scale-Space

The CURVE-ELEMENT description of an image consists of a set of tokens scattered in a three dimensional space consisting of two spatial dimensions plus one scale dimension. The construction of such a description, as well as its subsequent use for image interpretation, demand efficient and effective tools for computing information about the relative locations, orientations, and sizes of arcs represented by tokens, and for accessing tokens on these bases.

2.1 Self-Similarity Across Scales

Self-similarity across scales is a property by which information about spatial configurations, such as the arrangement of marks on this page, may be disjoined from information about the absolute size of their appearance in an image. For example, figure 2a*i* depicts an arrangement of marks comprising a circular arc. A description of this arc that does *not* possess self-similarity with respect to scale is curvature, because when the arc is magnified in size, the radius of curvature changes. A property that does remain invariant with respect to magnification is angular extent, which is proportional to the arc's length divided by (normalized by) the radius of curvature. Objects' identities are invariant with respect to their spatial magnification in images. That arcs of like angular extent appear more similar than arcs of like curvature supports the view that self-similarity across scales is a desirable property for a visual representation.

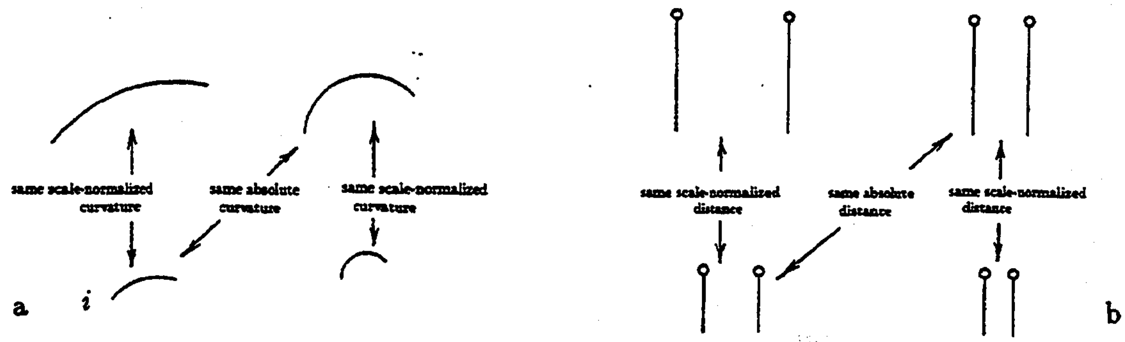


Figure 2: Scale-normalized measures of curvature and distance are invariant with respect to spatial magnification.

For this reason, in the present work information about metric curvature is expressed as a *scale-normalized curvature*, (or *sn-curvature*), which is equivalent to angular extent. By convention, positive sn-curvature is interpreted as curving in the clockwise direction while negative sn-curvature means counterclockwise turning. Similarly, *scale-normalized distance* is employed as a measure of distance between two spatial events that normalizes for the absolute sizes of the events and thereby achieves invariance with respect to absolute magnification.

These scale-normalized measures are facilitated by, following Witkin [36], graduating the scale dimension logarithmically so that equal intervals correspond to equal amounts of spatial magnification. Thus, a spatial magnification m is equivalent to a translation s in the scale dimension:

$$s = A \log m, \tag{1}$$

where A is a constant. Then, the scale-normalized curvature, ${}^{\mathrm{sn}}\kappa$, of a circular arc segment whose scale is s is given by

$${}^{\mathrm{sn}}\kappa = \kappa e^{\frac{s}{A}}. \tag{2}$$

The scale-normalized distance between two spatial events occurring at scales s_1 and

s_2, respectively, and separated by a distance $\mathbf{D}$, is given by

$$^{\mathrm{sn}}\mathbf{D} = \frac{\mathbf{D}}{\frac{1}{2}(e^{\frac{s_1}{A}} + e^{\frac{s_2}{A}})}. \tag{3}$$

Further justification for these definitions is provided in [30].

2.2 CURVE-ELEMENT Tokens

A CURVE-ELEMENT token possesses six properties of primary interest: (1) x-location, (2) y-location, (3) orientation, (4) scale or size, (5) scale-normalized curvature, (6) smoothness. See figure 3. To expedite certain computations it is also useful to cache within each token pointers to other tokens such as nearby tokens, supporting tokens, and the nearest neighbor on each end.

CURVE-ELEMENT token	
name:	ce-2854
x-location:	347.6
y-location:	93.4
orientation:	.148
scale:	4.66
sn-curvature:	.086
smoothness:	1.32

a

Figure 3: A CURVE-ELEMENT token is a packet of information about a circular arc. The token in (b) is consider smoother than the token in (c) because its support falls within a narrower window.

The scale s of a token is the location in the scale dimension corresponding to the tokens' arc length l,

$$s = A \log\left(\frac{l}{l_0}\right), \tag{4}$$

where l_0 is some distance assigned $s = 0$.

Because a CURVE-ELEMENT token asserts a circular arc without regard to the figure/ground relationship of the regions on either side of the arc, each token with orientation θ and sn-curvature ${}^{\mathbf{sn}}\kappa$ is equivalent to a token with orientation $\theta + \pi$ and sn-curvature $(- {}^{\mathbf{sn}}\kappa)$, and all computations must accord with this 180° symmetry property of CURVE-ELEMENT tokens.

The smoothness of a CURVE-ELEMENT token is a statement about the precision to which the arc model fits the underlying image data. Smoothness is expressed as a distance in the scale dimension, equivalent to asking, down to what scale of tokens does the support of the arc fall within a predetermined scale-normalized spatial distance ${}^{\mathbf{sn}}\mathbf{D}_{sm}$ from the arc? For an arc R of length l fit to a set of (dimensionless) points, p_i, smoothness H becomes:

$$H = -\log B \frac{\max_i D(R, p_i)}{l}, \tag{5}$$

where D measures the (absolute) distance between the arc and a point, and B is a constant. This is a self-similar measure, so the same smoothness is assigned to an arc with respect to a given set of points regardless of the spatial magnification of the system. Due to the logarithm, an arc whose support fits precisely is considered infinitely smooth.

2.3 Scale-Space Blackboard

CURVE-ELEMENT tokens are maintained in a data structure called the *Scale-Space Blackboard* which preserves the pictorial, spatial, organization of the original two-dimensional image. Tokens in the Scale-Space Blackboard are additionally organized by scale. In practice, this data structure consists of a stack of two-dimensional arrays, each element of which is a list of tokens falling within the spatial domain of that bin, while each level in the stack spans an octave range in the scale dimension.

A typical operation on the Scale-Space Blackboard is, "Deliver all tokens within scale-normalized distance ${}^{\mathbf{sn}}\mathbf{D}$ of the token at (x, y, s)." The property of self-similarity across scales may be used to advantage by quantizing space, at each level in the Scale-Space Blackboard, with a tesselation size proportional to a constant scale-normalized distance. This results in coarser quantization at larger scales, and the data structure therefore resembles a pyramid architecture [29].

The ability to index information on the basis of spatial location and scale contributes substantially to the efficiency of local grouping algorithms. The combinatorics of testing subsets of tokens for, say, sufficient alignment, grows prohibitively quickly with the number of tokens describing the image unless unlikely candidate subsets can be excluded a priori through a spatial selection facility such as provided by the Scale-Space Blackboard.

3 Computing CURVE-ELEMENT Tokens

3.1 Placement of Smallest Scale Tokens

CURVE-ELEMENT tokens at the finest, initial, scale may be asserted by any of a variety of methods, for which the subsequent grouping algorithm works equivalently. The main requirements are that tokens falling along a common contour align with one another reasonably well, and that they are are placed at appropriate intervals along image curves. The representation is designed for tokens' arcs to overlap one another by approximately 50% of their lengths.

The intial tokens for figures 6 and 9 were generated by first thinning the original image to 4-connectivity, then creating chain-linked curve segments by tracing curves (without particular regard to the behavior at junctions), and finally placing CURVE-ELEMENT tokens at intervals of every four pixels along the curve segments. Token location, orientation, and sn-curvature were determined by least squares fit to nine pixels, these being the center pixel and the nearest four in either direction.

For fitting a circular arc to a set of points, we employ a simple two-stage procedure. Bookstein's [6] efficient method for fitting a circle to points works well for points forming deep arcs ($> \sim 90°$), but breaks down for noisy shallow arcs. Conversely, shallow arcs are adequately fit by a parabolic model. Therefore, we first perform a least squares parabolic fit through the centroid of the points; if the arc is sufficiently shallow, these fitting parameters are adopted for the CURVE-ELEMENT token; if not, pose and curvature parameters from Bookstein's method are used.

An alternative method for computing finest scale CURVE-ELEMENTS was employed for figure 7. This involved detecting intensity ridges by filtering the original image with a cosine phase Elementary Gabor Filter (see e.g. [10]) at 22.5° orientation intervals, performing non-maximum suppression, then thresholding at a threshold set automatically by means similar to that described by Canny [8]. A 0-curvature CURVE-ELEMENT token could then be placed at every surviving pixel with orientation equal to that of the most strongly responding filter. To prune excess tokens until neighboring tokens were spaced at an appropriate scale-normalized distance, a subsampling procedure was used similar to that described in section 3.3.

3.2 Small-to-Large Scale Grouping

The crux of the computation lies in the grouping operation by which a set of CURVE-ELEMENT tokens at one scale may give rise to a new CURVE-ELEMENT token at a scale approximately one octave larger in size. The ideal case is shown in figure 4a. Here, tokens T_1, T_2,and T_3 occurring at, say, scale $s = 2$, are well aligned with one another and may support the assertion of token T_4 occurring at scale $s = 3$. The pose and curvature of token T_4 are determined by least squares fit, as described above, to points sampled along the lengths of the arcs represented by its supporting tokens.

While this ideal case does occur often in real images, the real challenges to grouping methods are found in more complex geometric configurations. The remaining examples of figure 4 present a number of other, more difficult, situations that may be observed to occur in images possessing curvilinear structure readily apparent to human observers. These include crossing contours, gaps, tangential junctions or Y-junctions, and surrounding clutter.

The problem faced by a token grouping approach may be viewed in the following way: suppose we adopt a strategy analogous to the ideal case, that is, a fitting

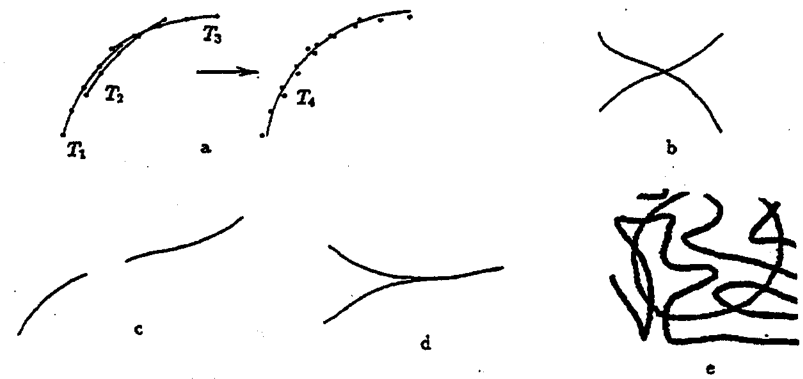


Figure 4: (a) In the ideal CURVE-ELEMENT grouping case, all of the tokens in the vicinity of a "seed" token (T_2) contribute points from which the parameters of a larger scale CURVE-ELEMENT can be computed. More difficult situations arise when contours cross one another (b), have gaps (c), form Y-junctions (d), or are found among the clutter of nearby contours (e).

algorithm will compute arc parameters for a token at some larger scale based on points derived from tokens occurring at a smaller scale range. Then, which smaller scale tokens are to serve as support (contribute points for fitting) for each larger scale token? Figure 5 illustrates the issue for situation 4d. Figure 5b demonstrates that fitting an arc to *all* of the CURVE-ELEMENTS in a spatial neighborhood can result in an incorrect assertion about the larger scale curvilinear structure. We are led to the strategy of selecting the support for new, larger scale, CURVE-ELEMENTS by choosing among subsets of smaller scale tokens.

If at least two smaller scale tokens are required to support a larger scale token, then combinatorially, $\sum_{k=2}^{n} \binom{n}{k}$ subsets of n tokens may be constructed. Figure 5c shows the twenty-six arcs resulting from fits to points derived from all possible such subsets of at least two of the five tokens in figure 5a. A suitable criterion for selecting the good arcs, that is, the arcs that agree well with their support and capture the larger scale curvilinear structure, is offered by the smoothness measure presented in

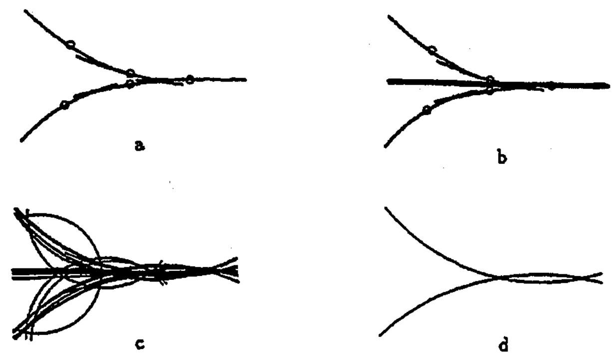


Figure 5: (a) Five CURVE-ELEMENT tokens. (b) The least-squares arc fit to all of these tokens. (c) The 26 arc fits to subsets of at least two tokens. (d) The two arcs from (c) having the greatest smoothness parameter.

section 2.2. This is illustrated in figure 5d, which displays just those arcs in figure 5c whose smoothness falls above a threshold.

Thus the goal of CIRCULAR-ARC token grouping is to efficiently identify collections of tokens that yield larger scale tokens of high smoothness when a circular arc fit is made. For the bulk of the small-to-large scale token grouping operations, this can be accomplished by an algorithm based on the following strategy: Each token occurring in a given scale range serves as a seed that attempts to recruit nearby CURVE-ELEMENTS to support a token approximately one octave larger in scale. The algorithm is "greedy" in that nearby tokens are recruited one-by-one, best first, on the basis of their contribution to the smoothness of the larger scale arc. More formally,

For each seed token T_i:

[1]Throughout the paper free parameters of the computation such as thresholds are denoted by the symbol, ρ. All reported results employed the same fixed settings for these constants, the choices for which are presented in Appendix B.

1. Initialize the variables, *support-list* and *potential-recruit-pool*. *Support-list* is the set of tokens to which the larger scale arc will be fit. *Potential-recruit-pool* is a set of tokens T_j that are eligible to participate with seed token T_i in the support of a larger scale token. To be included in the *potential-recruit-pool*, a token must be in the same scale range as T_i, it must be located within a threshold scale-normalized distance of the seed token which is approximately equal to a tokens' arc length, and it must fall above a threshold measure of spatial relationship with respect to T_i, called *alignment affinity*, and denoted $F(T_i, T_j)$. Two tokens' alignment affinity is large when they are near to one another and cocircular. Our particular mathematical expression for this relation is presented in Appendix A.[1]

$$\begin{aligned} &\textit{support-list} \leftarrow \{T_i\} \\ &\textit{potential-recruit-pool} \leftarrow \{T_j : |s(T_i) - s(T_j)| < \rho_1 \\ &\qquad\qquad\qquad\qquad\qquad \text{AND } {}^{\mathbf{sn}}\mathbf{D}(T_i, T_j) < \rho_2 \\ &\qquad\qquad\qquad\qquad\qquad \text{AND } F(T_i, T_j) > \rho_3\} \end{aligned}$$

2. Attempt to recruit first the two tokens in the *potential-recruit-pool* that are nearest to the seed token on each end (these are called the *forward-neighbor* token, $T_{i,fw}$, and the *rearward-neighbor* token, $T_{i,rw}$).

Let h_0 *be the smoothness of the circular arc fitting the tokens,* $T_i, T_{i,fw}$, *and* $T_{i,rw}$,

$$\begin{aligned} &\textit{if} \quad h_0 > \rho_4 \\ &\textit{then support-list} \leftarrow \{T_i, T_{i,fw}, T_{i,rw}\} \\ &\textit{else support-list} \leftarrow \{T_i\} \end{aligned}$$

3. Iteratively recruit the best candidate token from *potential-recruit-list* until the smoothness of the resulting arc falls below an adaptive threshold, σ.

3a. Initialize σ.

$$\begin{aligned} &\textit{if} \quad h_0 > \rho_4 \\ &\textit{then } \sigma \leftarrow \max(\rho_5, \rho_6 h_0) \\ &\textit{else } \sigma \leftarrow 0 \end{aligned}$$

3b. For each token T_j in the *potential-recruit-pool* and not already in *support-list*, measure the smoothness h_j resulting for the circular arc fit to the tokens that are already members of *support-list*, plus token T_j. Set $h_{best} \leftarrow \max(h_j)$, and set the variable T_{best} to be the corresponding best-fitting token.

3c. Either accept the best-fitting token and continue adding to *support-list*, or else exit.

$$
\begin{array}{ll}
\textit{if} & h_{best} \geq \sigma, \\
\textit{then} & \textit{support-list} \leftarrow \{T_{best}\} \cup \textit{support-list} \\
 & \sigma \leftarrow \max(\rho_5, \rho_6 h_{best}) \\
 & \textit{proceed to step 3b.} \\
\textit{else} & \textit{proceed to step 4.}
\end{array}
$$

4. Assign larger scale token parameters based on least squares fit to the tokens in *support-list*, as described in section 3.1. Assign token smoothness as described in section 2.2, and add to the Scale-Space Blackboard all tokens whose smoothness exceeds a threshold value ρ_7.

The effect of this algorithm is to recruit tokens that lie along a common contour. If the local contour is fit by a circular arc very precisely, then the threshold σ will hold potential recruits to a tight tolerance. If the arc forms a less than perfect circular arc, more slop is allowed in the support tokens that can be recruited into *support-list.*

The procedure is first carried out for seed tokens in the scale range, $s = 0$, delivering tokens in the scale range $s = 1$, or at approximately twice the size of the initial seed CURVE-ELEMENTS. After a subsampling step, described below, these tokens in turn give rise to tokens at scale $s = 2$, and so forth. The grouping step is spatially local and may be carried out across all seed tokens at a given scale range in parallel.

A second, supplementary, mechanism for introducing larger scale CURVE-ELEMENTS is based on bridging gaps between aligned tokens. Any pair of tokens T_i and T_j meeting the following conditions is also allowed to spawn a new token: (1) $|s(T_i) - s(T_j)| < \rho_8$ (that is, T_i and T_j are in the same scale range); (2) $\rho_9 < {}^{\mathbf{sn}}\mathbf{D}(T_i, T_j) < \rho_{10}$; (3) $G(T_i, T_j) > \rho_{11}$ (G is a measure for the degree to which two arcs lie on the same circle (see Appendix A); (4) No other token lies between and in alignment with T_i and T_j.

These conditions are spatially local and the detection of qualified "gap-bridging" pairs is facilitated by the spatial indexing properties of the Scale-Space Blackboard. The resulting CURVE-ELEMENTS are asserted on the Scale-Space Blackboard along with those found by the primary grouping mechanism, and the circular arc parameters are determined as usual by least squares fit, in this case there being only two support tokens.

3.3 Subsampling Larger Scale Tokens

Because a larger scale token is spawned at each seed token, the result of the above procedure leaves larger scale tokens more densely packed than the target 50% overlap of neighboring like-scale CURVE-ELEMENTS along the length of a contour. Therefore, the following procedure is used to subsample tokens:

1. For each larger scale token, compute a *redundancy-cost* based on the degree of overlap with and alignment with its *forward-neighbor* and *rearward-neighbor* tokens.
2. Remove from the Scale-Space Blackboard every token whose *redundancy-cost* is greater than that of both its *forward-neighbor* and *rearward-neighbor*, and greater than a threshold ρ_{12}.
3. Proceed to Step 1., and repeat the procedure until no token is removed from the Scale-Space Blackboard at step 2.

Due to subsampling, each octave range in the Scale-Space Blackboard will contain approximately half as many tokens as the next smallest scale range.

4 Results

4.1 Performance

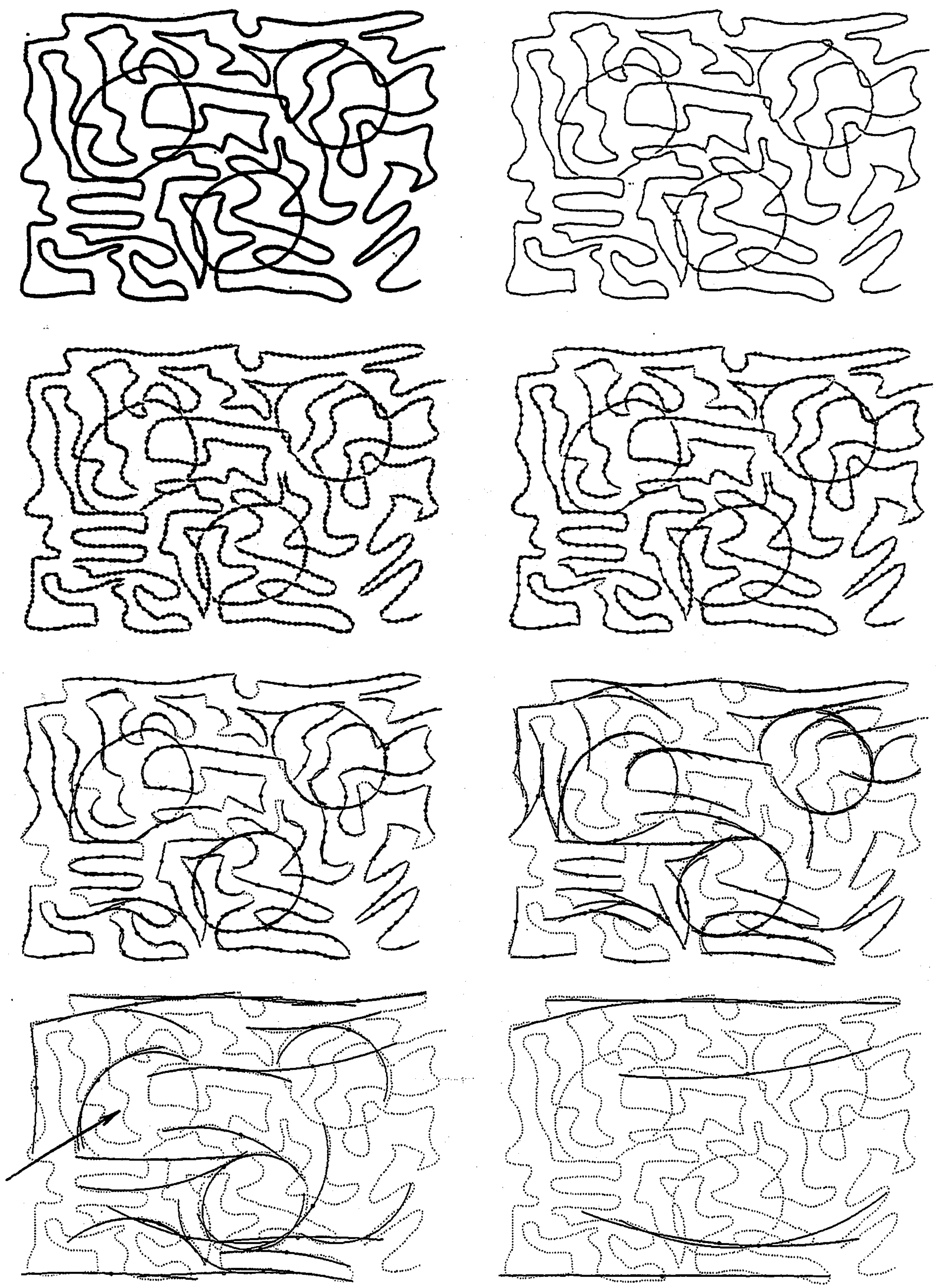
Figure 6 presents CURVE-ELEMENTS found by the grouping procedure for a line drawing including three closed circular contours and many winding paths [18]. At each scale range, tokens are depicted by the circular arc they represent. A small circle is drawn at the midpoint of each arc in order to better show the density of tokens along contours. This multiscale description employs a total of 4,647 tokens distributed across scales as follows: Scale range 0: 2335 tokens, 1: 1266, 2: 626, 3: 282, 4: 102, 5: 30, 6: 6 (Figure 6 displays a subset of these based on tightening the threshold on token smoothness.) Overall computation time is approximately one second per token on a Symbolics 3650 serial computer, or about 80 minutes for figure 6.

The major strengths and weaknesses of the grouping algorithm are evident in this figure. First, note that performance is correct in the ideal situation. That is, where the contour is relatively smooth, unbroken, and isolated from clutter, CURVE-ELEMENT tokens are fit to the contour at intervals of roughly half an arc length. These situations occur mostly at the smaller scale ranges.

At scale ranges 3 (6e) and above, disjoint contour fragments become near to one another in relation to their lengths, and it is at these scales that the procedure for selecting the support of larger scale CURVE-ELEMENTS becomes crucial. Note in figures 6e and 6f that tokens are found corresponding to contours that cross one another, including those of the prominent circles. Also, at these scales the fact that tokens represent circular arcs becomes manifest in that the token description no longer fits the original data precisely, but becomes something of a smooth approximation.

At the largest scales some CURVE-ELEMENTS are asserted that do not correctly correspond to human perception of curvilinear structure. Many of these arise from the bridging of gaps between aligning contour fragments for which, if they appeared in isolation, it would otherwise be appropriate to assert a gap-bridging token. A weakness of the present grouping algorithm is therefore that it does not take into account sufficient information about the local environment of potentially groupable tokens. Although it is not visible in the illustration, many of these questionable tokens are, however, betrayed by having a lesser smoothness parameter than more perceptually apparent contour fragments.

A second shortcoming of the algorithm is indicated by the arrow in figure 6g. Here, the contour is smooth at the smallest scales, roughly straight at a very large scale,



a b c d e f g h

Figure 6: (a) Original image (from [18]). (b) Finest scale CURVE-ELEMENT tokens (see text). (c) through (h) CURVE-ELEMENTS at successive octave scale ranges. Circles denote midpoint of circular arc models and indicate density of tokens asserted along contours.

but very wavy at an intermediate scale. The algorithm is not capable of detecting emergent large scale curvilinear structure where a contour is very broken, jagged, or wavy because every token assertion must be supported by tokens at the next smallest scale range. At scale range 4 (6f) the indicated section of contour is fit acceptably by no circular arc, and consequently no large scale token can be asserted at scale 5 (6g).

The strength of the gap-jumping component of the algorithm is exhibited in Figure 7. Note that the loop is broken at the smallest scales, yet large scale tokens cover its entire length. Note also that contours at small or intermediate scales merge and are described by single arcs at larger scales. (However, the onological issue by which closely spaced parallel lines may be interpreted at larger scales as unitary curvilinear events does merit further consideration.)

4.2 Tokens as Indicators of Significant Image Events

CURVE-ELEMENT tokens serve as useful units or chunks of information for image analysis because they make explicit important image structure that is only latent in edge or line detector output. Information about the visual image may be accessed either by querying the tokens directly or by selecting for further processing just those tokens possessing certain properties.

Figure 8a displays a subset of CURVE-ELEMENT tokens appearing in figure 6 selected on the basis of scale-normalized curvature: selected tokens have absolute value of sn-curvature falling above a threshold. 8b shows the results of a similar selection step, in which in addition to falling above a sn-curvature threshold, tokens must align with at least one neighboring token along a common circle in order to be selected. The selection of tokens in this way simplifies the task of establishing the presence and locations of circles or partial circles in a complex image.

Figure 9 shows that the CURVE-ELEMENT token description of an image of handwritten text reflects our perception of the slash through the word as a single coherent entity. A single mouse click invokes a trivial program to remove from the blackboard this "slash" token at the largest scale, plus all smaller scale tokens which support it,

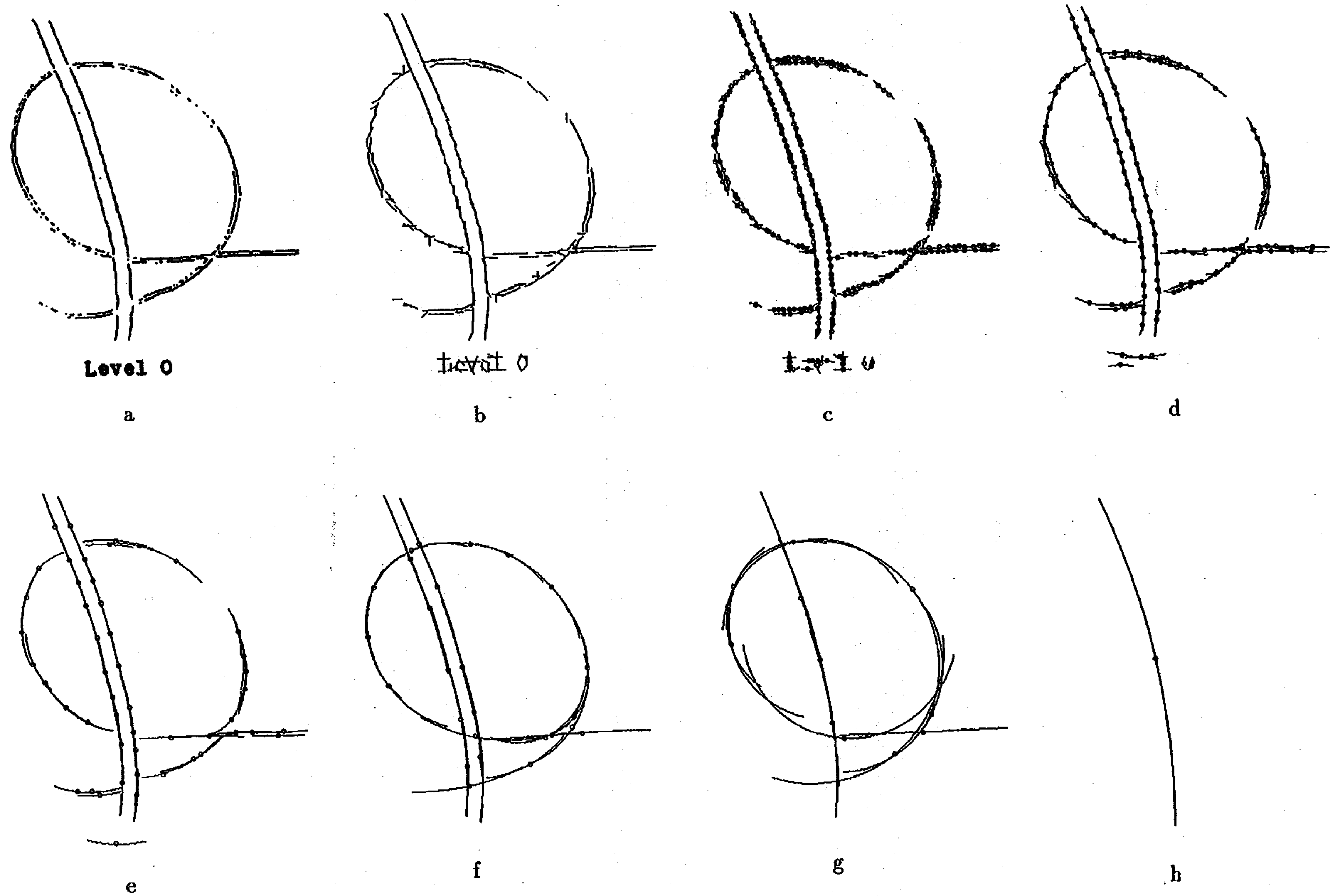
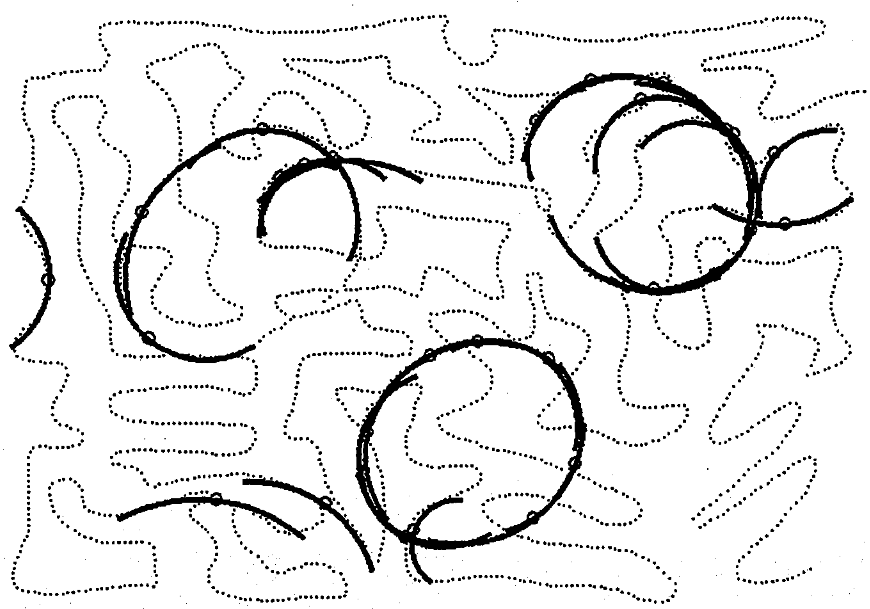
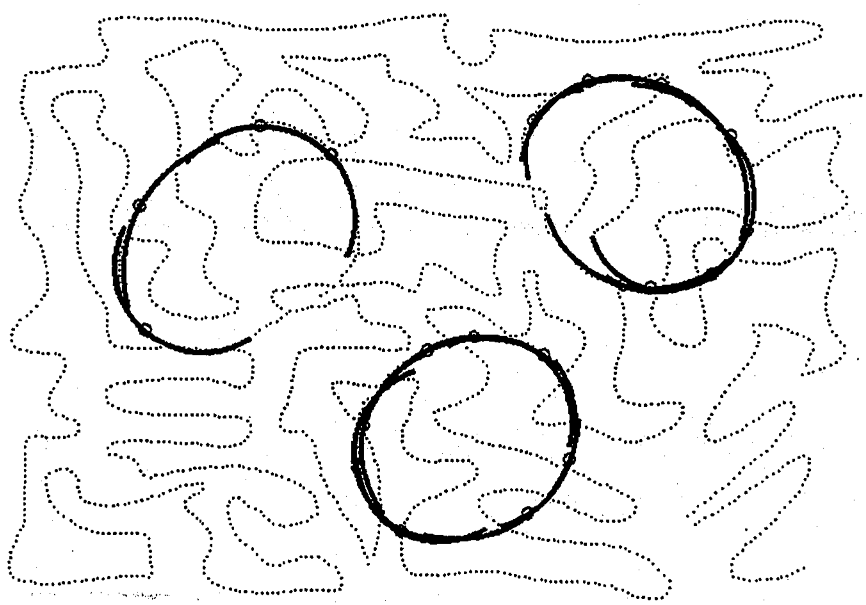


Figure 7: (a) Original image (from [21]). (b) Finest scale CURVE-ELEMENT tokens (see text). (c) through (h) CURVE-ELEMENTS at successive octave scale ranges.



a



b

Figure 8: (a) CURVE-ELEMENTS from figure 6f whose scale-normalized curvature lies above a threshold value. (b) Tokens from 6f selected on the basis of scale-normalized-curvature, smoothness, and cocircular alignment with a nearby token.

leaving behind the (Scale 0) tokens shown in figure 9d. Note that no realistic spatial filter would detect this line in the presence of the clutter of the surrounding lettering, and the set of chain coded contours found by linking small scale detector outputs would yield only an incoherent basketful of fragments.

5 Conclusion

A representation for curvilinear structure consisting of overlapping CURVE-ELEMENT tokens denoting circular arc models constitutes a useful stage for intermediate level visual processing because it strikes a good balance between simplicity, accuracy, abstraction, and richness. CURVE-ELEMENTS make explicit important aspects of image structure in themselves, and they are of the right "grain size" to be combined in simple ways into more complex structures. The assertion of tokens in an overlapping fashion at multiple scales leads to richness: to a first approximation, a token is available at the right location and scale to describe explicitly and exclusively *every* section of contour; any of these could potentially play an important role in later visual analysis such as queries, selection, or further grouping.

The grouping algorithm presented for computing CURVE-ELEMENTS achieves good performance under the restriction that curvilinear structure identified must be relatively smooth and free of very large gaps (relative to the extents of the contours forming the gap). Crossings, Y-junctions, gently wavy contours, small to moderate size gaps, and the presence of moderate clutter are all handled appropriately by the algorithm in nearly all cases observed. Figures 6 through 9 offer representative results. Because of the algorithm's neglect of certain cases, however, further work is needed if CURVE-ELEMENT tokens are to be asserted correctly in images containing dashed or dotted lines [1], jagged contours, or fuzzy (grey-level) contours. In these cases it may be appropriate to combine grouping methods with other techniques such as oriented linear or nonlinear filtering.

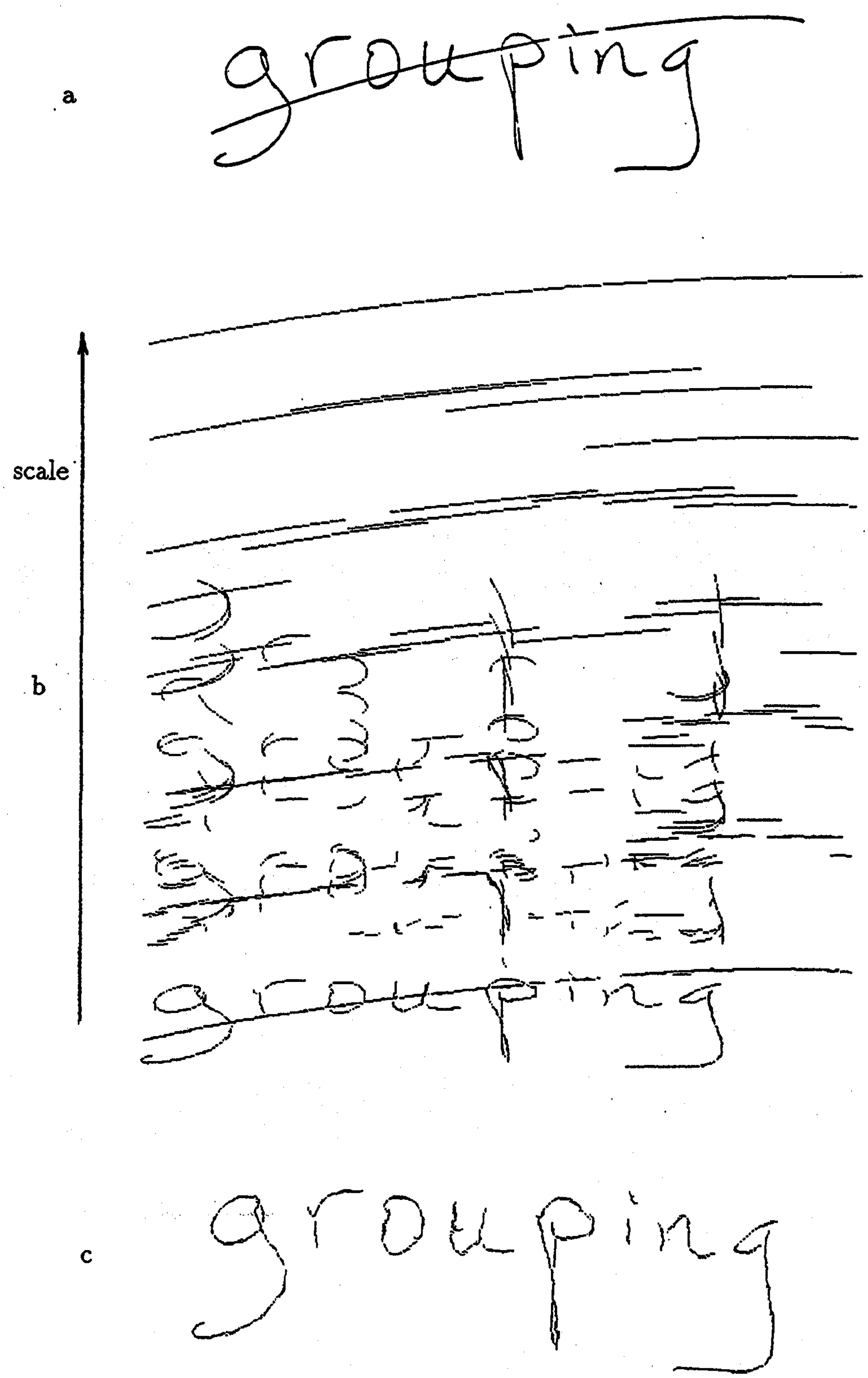


Figure 9: (a) Original image. (b) CURVE-ELEMENT tokens across all scales, viewed in oblique projection. (c) Circular arc tokens at Scale 0 remaining after removal of the large scale token representing the slash, plus those tokens supporting it down through successively smaller scales.

Appendix A: Spatial Proximity Measurement Functions

A function F assessing the *alignment-affinity* of two CURVE-ELEMENT tokens T_1 and T_2 operates on the relative locations and orientations of the tokens, independent of their curvatures. This function is engineered to approach the value 1.0 when the tokens are either moderately near one another and cocircular or else very near one another and parallel, but approach the value −1.0 when the tokens are moderately near one another and not cocircular.

$$F(T_1, T_2) = F_{nf}(2F_c(1 - F_{nn}F_{ss}) - 1)$$

where

$$F_{nf}(T_1, T_2) = B(\max(0, ({}^{\mathbf{sn}}\mathbf{D}(T_1, T_2) - \rho_{13})), \rho_{14})$$

$$F_c(T_1, T_2) = 1 - (F_{c1}(1 - F_{cn}) + F_{cn}F_{c2})$$

$$F_{c1} = \frac{1}{\pi} 2\min(|\alpha_1 + \alpha_2|, (\pi - |\alpha_1 + \alpha_2|))$$

See figure 10.

$$F_{cn} = B({}^{\mathbf{sn}}\mathbf{D}(T_1, T_2), \rho_{15})$$

$$F_{c2} = 2|\frac{\theta}{\pi}|$$

$$F_{nn} = 1 - B(\max(0, ({}^{\mathbf{sn}}\mathbf{D}(T_1, T_2) - \rho_{16})), \rho_{17})$$

$$F_{ss} = (|\frac{\alpha_1}{\pi}| + |\frac{\alpha_2}{\pi}| - |\frac{\alpha_1}{\pi} - \frac{\alpha_2}{\pi}|)$$

$$B(x, x_0) = \begin{cases} 0 & \text{if } x > x_0 \\ 1 - ax^2 & \text{if } x < \frac{x_0}{2} \\ a(x - x_0)^2 & \text{otherwise} \end{cases}$$

A function G assessing the degree to which two arcs lie on the same circle is engineered to return a maximum value of 1 when two arcs are precisely cocircular in their spatial proximity, relative orientation, and curvatures.

$$G(T_1, T_2) = \rho_{18}G_n + \rho_{19}G_o + \rho_{20}G_c$$

where

$$G_n(T_1, T_2) = {}^{\mathrm{sn}}\mathbf{D}(T_1, P) + {}^{\mathrm{sn}}\mathbf{D}(T_2, P)$$

See figure 11.

$$G_o(T_1, T_2) = \min(|\beta_1 - \beta_2|, |\beta_1 - \beta_2 - \pi|)$$

$$G_c(T_1, T_2) = |\Delta\kappa_{1,2}| \min(.1, |\theta_1 - \theta_2|)|$$

These functional forms were arrived at through ad-hoc experimentation to have properties required for the CURVE-ELEMENT grouping algorithm, and alternative formulations having similar behavior would be entirely appropriate.

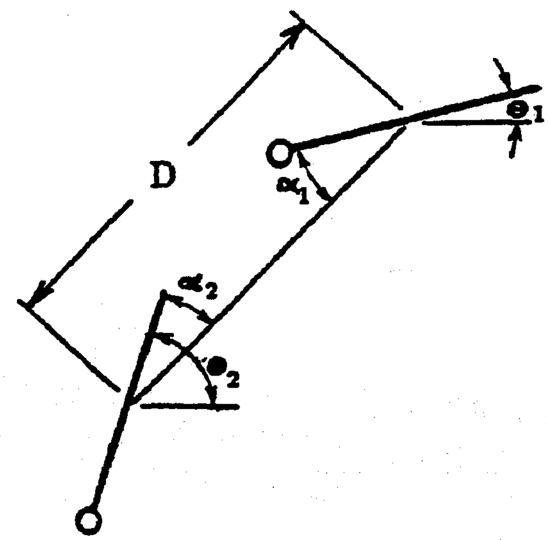


Figure 10: Geometry for alignment-affinity measure.

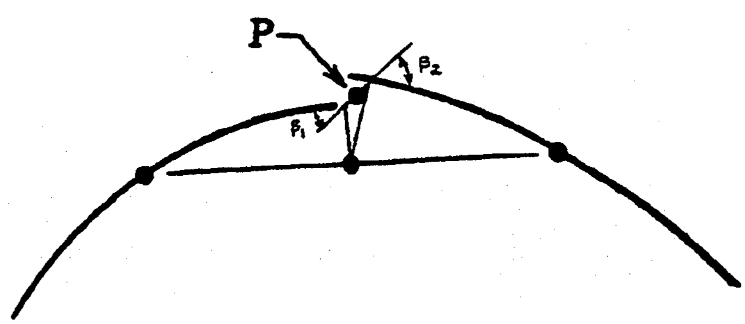


Figure 11: Geometry for cocircularity measure.

Appendix B: Values of Free Parameters

The following are values of the free parameters of the perceptual salience criteria and curve segment identification algorithm used in the current implementation.

parameter	value
ρ_1	1.0
ρ_2	9.0
ρ_3	.3
ρ_4	.4
ρ_5	.1
ρ_6	2.0
ρ_7	.5
ρ_8	1.0
ρ_9	8.0
ρ_{10}	14.0
ρ_{11}	.3
ρ_{12}	.7
ρ_{13}	12.0
ρ_{14}	6.0
ρ_{15}	5.0
ρ_{16}	.5
ρ_{17}	1.5
ρ_{18}	.2
ρ_{19}	5.0
ρ_{20}	100.0

References

[1] Ahuja, N., and Tuceryan, M.; [1989]; "Extraction of Early Perceptual Structure in Dot Patterns: Integrating Region, Boundary, and Component Gestalt," *CVGIP*, **48**, 304-356.

[2] Anderberg, M., [1983], "Cluster Analyis for Applications," Academic Press, New York.

[3] Asada, H., and Brady, M., [1984], "The Curvature Primal Sketch," *IEEE Trans. PAMI*, **8**:1, 2-14.

[4] Ballard, D., [1981], "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Rec.*, **13**:2, 111-122.

[5] Boldt, M., Weiss, R., and Riseman, E., [1990], "Token-Based Extraction of Straight Lines," *IEEE SMC*, **19**:6, 1581-1594.

[6] Bookstein, F.; [1979]; "Fitting Conic Sections to Data," *CGIP*, **9**, 56-71.

[7] Burnes, J., Hanson, A., and Riseman, E.; [1986]; "Extracting Straight Lines," *IEEE TPAMI*, **8**:4, 425-455.

[8] Canny, J., [1986], "A Computational Approach to Edge Detection," *IEEE TPAMI*, **8**:6, 679-698.

[9] Chen, D.; [1989], "A Data-Driven Intermediate Level Feature Extraction Algorithm," *IEEE Trans. PAMI*, **11**:7, 749-758.

[10] Daugman, J., [1985], "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am.*, **2**:7, 1160-1169.

[11] Dolan, J., and Weiss, R.; [1989]; "Perceptual Grouping of Curved Lines," *Proc. DARPA IU Workshop*, 1135-1145.

[12] Dudek, G., and Tsotsos, J.; [1990]; "Recognizing Planar Curves Using Curvature-Tuned Smoothing," *Proc. 10th International Conf. Pattern Recognition*, Atlantic City, June, 1990, 130-135.

[13] Grimson, W.E.L., [1989], "On the Recognition of Curved Objects," *IEEE TPAMI*, **11**:6, 632-642.

[14] Grimson, W.E.L., and Huttenlocher, D., [1990], "On the Sensitivity of the Hough Transform for Object Recognition," *IEEE TPAMI*, **12**:3, 255-274.

[15] Kass, M., Witkin, A., and Terzopoulos, D., [1987], "Snakes: Active Contour Models," *Proc. ICCV*, London, 259-268.

[16] Lowe, D.; [1985]; *Perceptual Organization and Visual Recognition*, Kluwer, Boston.

[17] Lowe, D.; [1988]; "Organization of Smooth Image Curves at Multiple Scales," *Proc. ICCV*, Tampa, Fl., 558-567.

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

[19] Marimont, D., [1984], "A Representation for Image Curves," *Proc. National Conf. on Artificial Intelligence,* Austin, Texas, 237-242.

[20] Marr, D., [1976], "Early Processing of Visual Information," *Phil. Trans. R. Soc. Lond.* B **275**, 483-519.

[21] Meer, P., Sher, A., and Rosenfeld, A., [1989], "The Chain Pyramid: Hierarchical Contour Processing," *IEEE TPAMI*, **12**:4, 363-376.

[22] Merlin, P., and Farber, D., [1975], "A Parallel Mechanism for Detecting Curves in Pictures," *IEEE Trans. Computer* **24**, 96-98.

[23] Mohan, R., and Nevatia, R.; [1989]; "Segmentation and Description Based on Perceptual Organization," *Proc. IEEE CVPR*, 333-341.

[24] Mokhtarian, F., and Mackworth, A., [1986], "Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," *IEEE TPAMI*, **8**:1, 34-43.

[25] Nalwa, V., and Pauchon, E.; [1986], "Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," *IEEE TPAMI*, **8**:1, 34-43.

[26] Parent, P., and Zucker, S.; [1989]; "Trace Inference, Curvature Consistency, and Curve Detection," *IEEE TPAMI*, **11**:8, 823-839.

[27] Perkins, W., [1978], "A Model-Based Vision System for Industrial Parts," *IEEE Trans. Computers*, **27**:2, 126-143.

[28] Princen, J., Illingworth, J, and Kittler, J.; [1989]; "A Hierarchical Approach to Line Extraction" *Proc. IEEE CVPR*, 92-97.

[29] Rosenfeld, A; [1986]; "Pyramid Algorithms for Perceptual Organization," *Behav., Res, Methods, Instr., and Computers*; **18**:6, 595-600.

[30] Saund, E., [1990], "Symbolic Construction of a 2D Scale-Space Image," *IEEE TPAMI*, **12**:8, 817-830.

[31] Shirai, Y.; [1978]; "Recognition of Real-World Objects Using Edge Cues," in Hanson, A., and Riseman, E., *Computer Vision Systems*, Academic Press.

[32] Shneier, M., [1982], "Extracting Linear Features from Images Using Pyramids," *IEEE SMC*,**12**:4, 569-572.

[33] Sklansky, J., [1978], "On the Hough Transform for Curve Detection," *IEEE Trans. Computers*, **27**, 923-926.

[34] Ullman, S., and Shaashua, A.; [1988]; "Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network," MIT AI Memo 1061.

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

[36] Witkin, A., [1983], "Scale-Space Filtering," *Proc. IJCAI* 1019-1022.

[37] Witkin, A. and Tenenbaum, J., [1983], "On the Role of Structure in Vision," in Beck, J., Hope, B., and Rosenfeld, A., eds, *Human and Machine Vision*, Academic Press, New York.

[38] Zucker, S., David, C., Dobbins, A., and Iverson, L.; [1988]; "The Organization of Image Curve Detection: Coarse Tangent Fields and Fine Spline Coverings," *Proc. ICCV*, Tampa, Fl., 568-577.