A Novel Approach to Computation of the Shape of a Dot Pattern and Extraction of Its Perceptual Border

A. Ray Chaudhuri, B. B. Chaudhuri, and S. K. Parui

Computer Vision & Pattern Recognition Unit, Indian Statistical Institute, 203 B. T. Road, Calcutta 700035, India

Received October 6, 1995; accepted September 6, 1996

A novel approach to defining the external shape of a dot pattern is proposed from which the intuitive border of the set is extracted. The approach is based on a new definition called the *s*-shape, which can be generated by a data-driven procedure. The *s*-shape generates a staircase-like border. To obtain a polygonal border, an *r*-shape is defined for which the parameter *r* is found from *s*, the parameter of the *s*-shape. The main advantage of this approach is that it can be computed in $O(n)$ time for a dot pattern containing *n* points. The approach has three basic steps: (i) choice of an appropriate *s* (and corresponding *r*) from the given point set, (ii) generation of the *r*-shape, and (iii) cleaning of inconsistent parts from the *r*-shape. The diagram composed of the consistent edges of the *r*-shape is considered the perceived border of the dot pattern. A new structural basis called the dispersion matrix is evolved. Extension of the work to the digital case is discussed. The algorithm for extracting the perceptual border is fast since it is mainly composed of basic operations such as nonnegative integer addition and logical operations. Moreover, it can be implemented on parallel machines since the operations are local in the point space.

1. INTRODUCTION

Point sets with finite diameters in $\mathbb{R}^n$ are encountered in various pattern recognition and image processing problems. These points may correspond to feature vectors in feature space [1], pixels in digital images [2, 3], physical objects such as stars in the galaxy [4], or spatial data [5–7]. Depending on the problem, it is necessary to discover the structure in the point set in the form of clusters, directionality, and intrinsic dimensionality [8–13]. Registration and matching of point sets are also needed in some problems [14, 15].

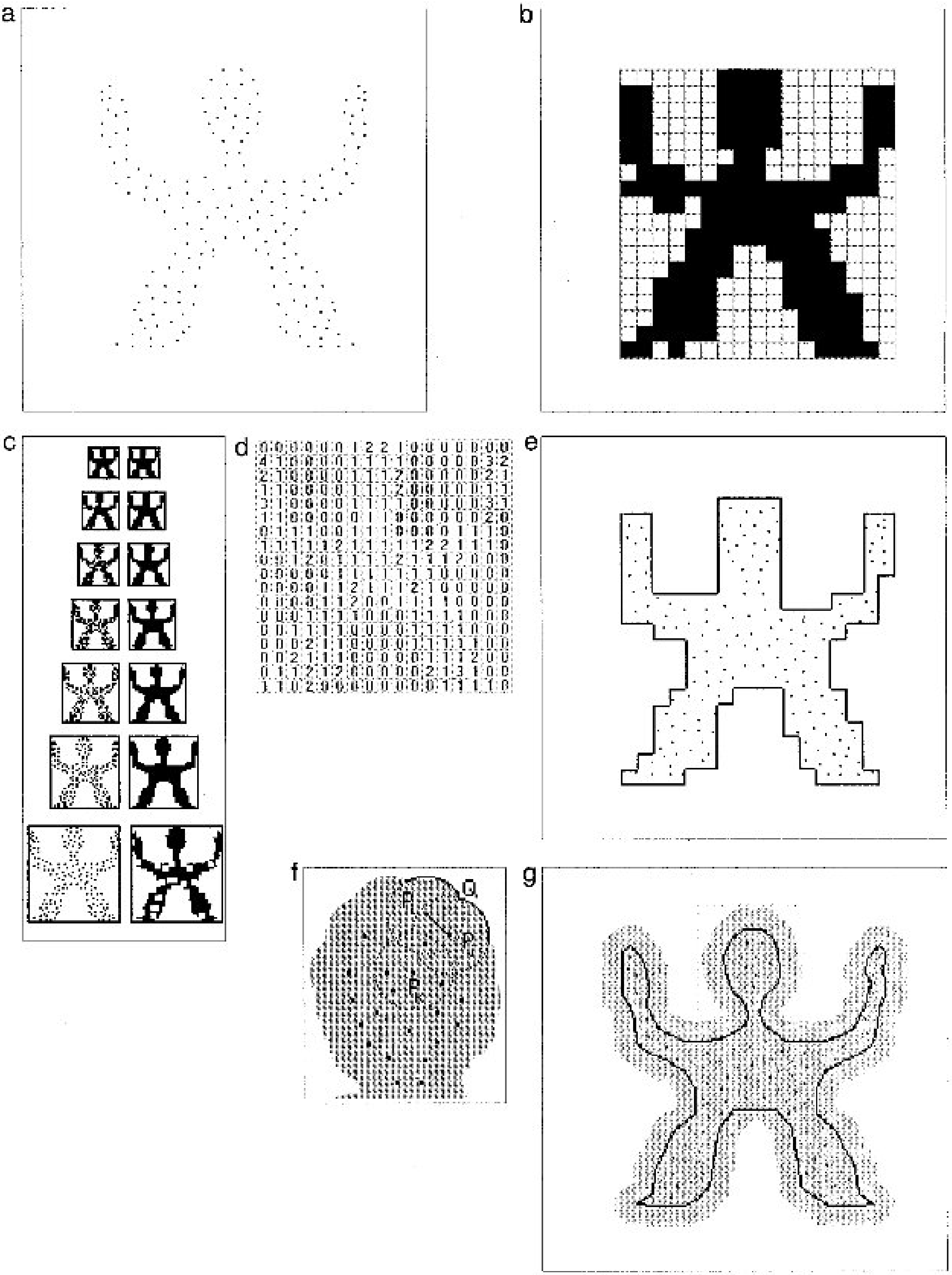
Another problem of interest is to find the border, formally known as the *external shape*, of a point set [16–25]. In $\mathbb{R}^2$ or $\mathbb{R}^3$ one can perceive the border of the point set if the points are clearly visible as well as fairly densely and more or less evenly distributed. Such a point set is referred to as a *regular dot pattern* or simply a *dot pattern* (*DP*) (Fig. 1a).

One way of defining the external shape of a dot pattern is its convex hull [16], but in many cases the underlying shape from which the points emerge is not convex. Edelsbrunner *et al.* [17] proposed a general definition of the external shape (convex or otherwise) of a dot pattern as α-shape and described an algorithm to compute it for a given α. Among others, Ahuja and Tuceryan [18] used Voronoi neighborhoods as the structural basis for extraction of the perceptual structure of a DP. A perceptual grouping is accomplished, and corrections on grouping are performed through constraint propagation using a probabilistic relaxation process. However, ad hoc thresholds are used in almost every level of decision making, and the overall processing complexity is quite high. Other notable work on external shape of a DP is that of Toussaint [19]. His definition of a *sphere-of-influence graph* is free from ad hoc thresholds and is computationally efficient. However, it is noted that when the DP is dense the *sphere-of-influence graph* in most cases do not agree with the perceived border of the DP.

The present paper also deals with the external shape of dot patterns. Our aim is to propose computationally efficient (linear, in terms of number of points) and robust unsupervised methods where the number of parameters is the only one which, when properly chosen, produces a border of the DP reasonably close to the perceived one.

We develop our approaches in $\mathbb{R}^2$ space although these can be extended in $\mathbb{R}^3$ in a rather straightforward manner.

Let the pattern plane be partitioned by a lattice of square grids. Consider the union of grids containing points of the DP. If the grid-length *s* is properly selected, the "smooth" version of this union approximates the underlying region of the pattern, and its border can be considered the border of the DP. This is the intuition behind the new shape descriptor called the *s-shape* (Fig. 1b). One can iteratively generate a finite sequence of *s*-shapes, called the *s-shape spectrum* (Fig. 1c). In Section 2, the mathematical basis of the shape spectrum is worked out. From this spectrum the *s*-shape closest to the intuitive structure of a DP can be selected using a parameter ε. A new dot pattern structural



0	0	0	0	0	0	1	2	2	1	0	0	0	0	0	0	0
4	1	0	0	0	0	1	1	1	1	0	0	0	0	0	3	2
2	1	0	0	0	0	1	1	1	2	0	0	0	0	0	2	1
1	1	0	0	0	0	1	1	1	2	0	0	0	0	0	1	1
3	1	0	0	0	0	1	1	1	1	0	0	0	0	0	3	1
1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	2	0
0	1	1	1	0	0	1	1	1	0	0	0	0	1	1	1	0
1	1	1	1	1	2	1	1	1	1	1	2	2	1	1	1	0
0	0	1	2	0	1	1	1	1	2	1	1	1	2	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	2	1	1	1	2	1	0	0	0	0	0
0	0	0	0	1	1	2	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
0	0	0	2	1	1	0	0	0	0	1	1	1	1	1	0	0
0	0	2	1	1	1	0	0	0	0	0	1	1	1	2	0	0
0	1	1	2	1	2	0	0	0	0	0	2	1	3	1	0	0
1	1	0	2	0	0	0	0	0	0	0	0	1	1	1	1	0

FIG. 1. The proposed approach. (a) A "human" shaped regular dot pattern. (b) An s-shape of the DP. (c) s_i-times scale reduced (H_i) spectrum and its smooth version (morphologically closed by unit disk). (d) The dispersion matrix of the DP ($\varepsilon = 0.4$). (e) An s-shape-based border. (f) Illustration of r-interior, r-edge, and r-adjacents (r-extreme points). (g) The r-shape with structuring radius.

basis called the *dispersion matrix* (Fig. 1d) is evolved. For s-shape-based border extraction of dot patterns (Fig. 1e), the dispersion matrix is extensively used.

To obtain a border of a DP smoother than the s-shape, another external shape descriptor called the *r-shape* is proposed. The idea behind this descriptor is as follows. Subject to the proper selection of the radius, called the *structuring radius*, the union of disks centering at points of the pattern can be considered a representation of the underlying region of the pattern. The r-shape (Fig. 1f) is the graph constructed by connecting the respective centers of each pair of intersecting disks which are partially exposed to the background. In Section 3, the r-shape and its related terminologies are formally defined. Among the existing external shapes, the r-shape is by nature closest to the α-shape. The relationship between the r-shape and the α-shape is also presented. The structuring radius for an r-shape is selected from the same sequence of positive numbers that is used for the s-shape spectrum. Based on the characterization, after partial/complete deletion of inconsistent loops from the r-shape, the remaining subgraph is considered the border of the dot pattern that is compatible with the perceived shape.

In Section 4, the implementation of our approach in digital space is presented. Here, Section 4.1 deals with an algorithm on computation of the dispersion matrix. This algorithm is used for border extraction, based on the s-shape. In Section 4.2, the digital r-shape is defined and an algorithm for its computation with the structuring radius is presented. For efficient implementation and faster processing of the r-shape, a successor-listing type of the data structure is proposed. In Section 4.3, an algorithm for detecting inconsistent edges as well as extracting the border that is compatible with the perceived shape of the input pattern is presented in detail.

In Section 5, the computational complexity of our proposed algorithms and an overall evaluation of our approach are discussed. In Section 5.1, it is shown that the complexity of border extraction based on either the s-shape or the r-shape is linear in terms of the cardinality n of the dot pattern. In this respect our algorithm is computationally cheaper than the existing approaches, most of which require $n \log(n)$. All approaches mentioned at the beginning of this section are of order $n \log(n)$. A few experimental results on some typical data sets are presented in Section 5.2. Finally, the advantages as well as the limitations of our approach, with a possible remedy, are mentioned.

2. s-SHAPE SPECTRUM AND CRUDE BORDER APPROXIMATION OF DP

Let S be a dot pattern containing n points. Let W be an optimal (in terms of minimum area) isothetic rectangular region (i.e., a rectangle whose sides are parallel to the horizontal and vertical axes of reference) containing S; i.e., $S \subset W \subset \mathbb{R}^2$. For a given grid side-length s, let $\mathscr{T}(s)$ denote a lattice of square grids, with horizontal and vertical sides on the real plane. For any grid g, let

$$G(s) = \cup\{g \mid g \in \mathscr{T}(s), g \cap W \neq \Phi\} \tag{1}$$

$$H(s) = \cup\{g \mid g \in \mathscr{T}(s), g \cap S \neq \Phi\}. \tag{2}$$

$G(s)$ denotes the set-union of grids over W, while $H(s)$ denotes the subset of $G(s)$ obtained by joining the grids which contain at least one dot each. Let $\#H(s)$ denotes the number of grids in $H(s)$. Then the area of $H(s)$ is defined as $A(H(s)) = \#H(s) \times s^2$.

Definition 2.1. $H(s)$, the hull of S induced by the lattice $\mathscr{T}(s)$ with grid-length s, is said to be an *s-shape* of S. (See Fig. 1b).

If the points of S are distributed uniformly over W, then

$$\bar{s} = \sqrt{\frac{A(W)}{n}}$$

is expected to be an optimal grid length in terms of underlying region approximation of the DP by the s-shape. Usually the points of S are distributed over only a small region of W. The induced hall $H(\bar{s})$ over S with grid-length $\bar{s}$ will give a (first) approximation of this region. We can make a better approximation iteratively as follows, where the area of the induced hull is gradually decreasing.

Consider the sequence $\langle s_i \rangle$ defined as

$$\begin{aligned} s_i &= \bar{s} && \text{when } i = 1 \\ &= \sqrt{\frac{A(H(s_i - 1))}{n}} && \text{when } i > 1. \end{aligned} \tag{3}$$

Henceforth $G(s_i)$ and $H(s_i)$ will be simply denoted as G_i and H_i, respectively.

Proposition 2.1. *The sequences $\langle s_i \rangle$ and $\langle A(H_i) \rangle$ are both monotonically nonincreasing and convergent and if $\underline{s}$ is the limiting value of $\langle s_i \rangle$ then*

$$\lim_{i \to \infty} s_i = \underline{s} \Rightarrow \lim_{i \to \infty} A(H_i) \to A(H(\underline{s})).$$

Proof. Since

$$\#H_i \leq n \quad \forall i,$$

we have

$$\begin{aligned} 0 \leq s_{i+1} &= \sqrt{\frac{\#H_i}{n}} \times s_i \\ &\leq s_i. \end{aligned}$$

Thus $\langle s_i \rangle$ is a nonnegative monotonically nonincreasing sequence. As this monotonically nonincreasing sequence is bounded from below (0 is a lower bound) it is also convergent.

Now,

$$\begin{aligned} A(H_{i+1}) &= \#H_{i+1} \times s_{i+1}^2 \\ &= \#H_{i+1} \times \left(\frac{\#H_i}{n} \times s_i^2 \right) \\ &= \frac{\#H_{i+1}}{n} \times (\#H_i \times s_i^2) \\ &= \frac{\#H_{i+1}}{n} \times A(H_i) \\ &\leq A(H_i). \end{aligned}$$

Therefore, $\langle A(H_i) \rangle$ is also a nonnegative monotonically nonincreasing sequence and hence is also convergent.

Let $\lim_{i\to\infty} s_i = \underline{s}$.

The sublattice $H(s_i)$ gradually coincides with $H(\underline{s})$ as s_i tends toward $\underline{s}$. Thus, we have by construction

$$\lim_{i\to\infty} s_i = \underline{s} \Rightarrow \lim_{i\to\infty} A(H_i) \to A(H(\underline{s})). \quad \blacksquare$$

Note that s_i in $\langle s_i \rangle$ will continue to decrease; i.e., $s_i < s_{i-1}$ until each grid in H_i contains only one dot.

Now we will show that $\langle s_i \rangle$ attains its minimum value, i.e., attains the limit, which is strictly positive after a finite number of steps.

Proposition 2.2. *The sequence $\langle s_i \rangle$ converges to $\underline{s} > 0$ after a finite number of steps.*

Proof.

$$\begin{aligned} s_{i+1} &= \sqrt{\frac{\#H_i}{n}} \times s_i \\ \Rightarrow \sqrt{\frac{\#H_i}{n}} &= \frac{s_{i+1}}{s_i}. \end{aligned}$$

By taking limits on both sides,

$$\begin{aligned} \frac{\lim_{i\to\infty} \sqrt{\#H_i}}{\sqrt{n}} &= \frac{\lim_{i\to\infty} s_{i+1}}{\lim_{i\to\infty} s_i} \\ \Rightarrow \frac{\lim_{i\to\infty} \sqrt{\#H(s_i)}}{\sqrt{n}} &= 1 \\ \Rightarrow \lim_{i\to\infty} \#H_i &= n. \end{aligned}$$

Thus, $\langle \#H_i \rangle$ is a convergent sequence of positive integers. As any convergent sequence of integers contains at most a finite number of distinct integers, $\langle \#H_i \rangle$ has only a finite number of elements which are distinct and the rest are all equal to its limiting value n. Let

$$j = \min\{i \mid \text{such that } \#H_i = n\}.$$

Note that this limiting value n is attained only when all points of S are mutually separated by grids. Then by definition, any s_i, for $i \geq j$, is constant and is equal to the limiting value; i.e.,

$$s_i = \underline{s} \quad \forall i \geq j.$$

If possible let $\underline{s} = 0$. By our assumption, since not all points of S are collinear, $A(W) > 0 \Rightarrow s_1 = \bar{s} > 0$.

Now, since $\langle s_i \rangle$ is a strictly monotonically decreasing sequence converging after a finite number of steps, there exists a positive integer j such that

$$\begin{aligned} \bar{s} > s_i > s_{i+1} &\quad \forall i < j \\ = \underline{s} &\quad \text{otherwise.} \end{aligned}$$

However, as $s_{j-1} > 0$ and #non-null grids in $H(s_{j-1}) > 0$,

$$s_j = \sqrt{\frac{\#\text{non-null grids of } H(s_{j-1})}{n}} \times s_{j-1} > 0,$$

a contradiction to the hypothesis that $\underline{s}(= s_j) = 0$. Hence, $\underline{s} > 0$ $\blacksquare$

Definition 2.2. The sequence $\langle H_i \rangle$ is said to be the s-shape spectrum of S.

The finiteness and positivity of $\langle s_i \rangle$ ensures that the s-shape spectrum is *computable.*

The s-shapes are a union of fixed sized squares. Thus, the border of an s-shape has a staircase-like appearance. Further, if the value of s_i is small, an s-shape may contain holes and its border may appear cracked. Thus, the border of an s-shape in the spectrum is considered to be a *crude border* of the dot pattern. Any hole filling as well as border smoothing algorithm (conventional or morphological) for digital images [26] gives a better appearance to the s-shapes. We proceed to make a smoother shape approximation called the r-shape. Each s-shape in the spectrum can induce an r-shape. However, it is useful for the user to know which value of s and corresponding r one should choose so that the r-shape of DP agrees well with the perceived shape. Thus, an automatic choice of an appropriate s (and hence r) from $\langle s_i \rangle$ is in order. At first, the sequence $\langle s_i \rangle$ is further analyzed for better understanding of its capability to give rise to an appropriate value of s.

Note that until all points of S are mutually separated by the grids of H_i, s_i continues to decrease. The rate of convergence of the sequence mostly depends on the homogeneity and structure of the DP. The terminal value $\underline{s}$ cannot give an appropriate grid-length unless the points of S are highly regularly distributed. It can be considered to be a lower bound of the side-length of a so-called optimal grid. However, considering the sequence $\langle s_i \rangle$ to be a spectrum of grid-lengths, we can select a suitable grid length s as follows.

For a given small $\varepsilon(>0)$, let

$$s = \max\left\{ s_k;\ \left| \frac{s_{k-1} - s_{k+1}}{s_k} \right| \le \varepsilon,\ s_{k-1} \in \langle s_i \rangle \right\}. \tag{4}$$

Definition 2.3. The quantity s is called the *ε-measure of dispersion* of the dot pattern S.

Let the grids of G_i be ordered in 2-dimensional array. Then G_i will induce a matrix $((g_{lm}))$, for example, whose (l, m)th element denotes the number of dots in the grid situated at lth row, mth column position.

Definition 2.4. The induced matrix of G_k, i.e., when the grid-length is equal to the ε-measure of dispersion, is said to be the *induced dispersion matrix* (or simply, dispersion matrix) on the dot pattern S. The dispersion matrix is denoted by $\mathrm{DMAT}(S)$.

Figure 1d illustrates the dispersion matrix of the DP of Fig. 1a where the value of $\varepsilon = 0.4$.

Definition 2.5. The *binary projection* of the dispersion matrix $((b_{lm}))$, for example, is denoted by $\mathrm{DMAT}_{\mathrm{prj}}(S)$ and defined as

$$\begin{aligned} b_{lm} &= 1 \quad \text{if } g_{lm} > 0 \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{5}$$

The dispersion matrix can be used as a structural basis of the DP. It can be considered to be a gray scale image, whereas its projection can be considered to be a binary image where each object pixel represents a nonempty grid of $H(s)$. It represents an s-times scale-reduced version of the s-shape where s is the ε-measure of the dispersion of the DP.

Consider the collection of binary projections of $((g_{lm}))$ corresponding to G_i, for $(i = 1, 2, \ldots, k)$. If the collection is ordered by i then the resulting sequence of binary images (s_i-times reduced H_i), as well as their smooth versions, show how the s-shape spectrum captures the underlying structure of the DP. Such a spectrum and its smooth version (where each projection is morphologically closed by a 3 × 3 structuring template) on the DP of Fig. 1a is presented in Fig. 1c.

In Section 4, a digital implementation of an s-shape-based algorithm (s = ε-measure of dispersion) for border extraction of a DP is presented. The computation is done over the binary projection of the dispersion matrix. Note that the cardinality of the projection is less than that of the DP.

3. r-SHAPE

Let $S = \{P_1, P_2, \ldots, P_n\}$ be a set of n points in $\mathbb{R}^2$ and let r be a positive quantity. Let $D_r(Q) \subset \mathbb{R}^2$ be the closed disk with radius r and center Q, and let $C_r(Q)$ be the boundary of $D_r(Q)$. The boundary and the interior of a closed set A will be denoted by $\mathrm{bdr}(A)$ and $\mathrm{int}(A)$, respectively.

Consider the union of all disks with fixed radius r centering at points of S. If the perimeter of the disk centering at a point, say P_k, is at least partially exposed to the background then the point is said to be an *r-extreme point*. Otherwise, it is called an *r-interior point*. Mathematically, $P_k \in S$ is an r-interior point of S iff $C_r(P_k) \subset \mathrm{int}(\bigcup_{i=1}^{n} D_r(P_i))$. Let

$$E_r(P_k) = C_r(P_k) - \mathrm{int}\left(\bigcup_{i=1}^{n} D_r(P_i) \right). \tag{6}$$

The following lemma can be easily verified.

Lemma 3.1. *$E_r(P_k) = \phi$ if and only if P_k is an r-interior point of S.*

$\overline{P_i P_j}$ is said to be an *r-edge* if $E_r(P_i) \cap E_r(P_j) \neq \phi$.

Note that $\overline{P_i P_j}$ is an *r-edge* if both disks centering P_i and P_j, respectively, have a common point (say Q) that is exposed to the background with respect to the union of all disks centering points of S. (See Fig. 1e).

Any r-extreme point P_i being an end point of an r-edge is said to be an *r-vertex* of S. Thus, the set of r-vertices consists of nonisolated r-extreme points. The number of r-edges passing through an r-vertex P_i is said to be the *degree* of that vertex and the other ends of these edges are called *r-adjacent vertices* (or simply *r-adjacents*) of P_i. If P_i has more than two r-adjacents then P_i is said to be an r-branch vertex of S.

Definition 3.1. The *r-shape* of S is the planar straight line graph whose vertices are the r-vertices and whose edges are composed of r-edges. (See Fig. 1g).

An *r-path* (or, simply a *path*) in an r-shape graph is defined as a maximal alternating sequence of r-vertices and r-edges, beginning and ending with r-vertices (referred to as *end vertices*), such that each r-edge is incident with the r-vertices preceding and following it. No r-vertex appears more than once and all r-vertices except the end vertices on the path are of degree 2. A path can be expressed by a string of r-vertices. In a diagram of the r-shape, a path represents a simple curve.

Let us recall the existing definitions for the α-shape [17]. Let α be an arbitrary real number. A *generalized disk of radius* $1/\alpha$ is defined as a closed disk of radius $1/\alpha$ if $\alpha > 0$, the closed complement of a disk of radius $-1/\alpha$ if $\alpha < 0$, and a closed halfplane if $\alpha = 0$. For a set S of 2-D points, the *α-hull* of S is the intersection of all generalized disks of radius $1/\alpha$ that contain S. A point P in S is *α-extreme* in S if there exists a generalized disk of radius $1/\alpha$ containing S such that P lies on its boundary. Two α-extreme points P and Q of S are *α-neighbors* if there exists a generalized disk of radius $1/\alpha$ containing S such that both P and Q lie on its boundary. The *α-shape* of S is the planar straight line graph whose vertices are the α-extreme points and whose edges connect the respective α-neighbors. As α approaches zero, the α-shape becomes the convex hull of S.

In [20], a close relation between r-shape and α-shape was established. In all terms (e.g., r-extreme point) and transforms (e.g., E_r) defined to find the r-shape, at first the union of disks is taken and then the interior of the union is considered. On the other hand, if the interior of each disk is taken and then their union is considered, then the edge joining the points P_i and P_j satisfying $E_r(P_i) \cap E_r(P_j) \neq \phi$ is an edge of the α-shape ($\alpha = -1/r$).

By using properties of α-shape [17] and r-shape, it can be shown that the r-shape is a subgraph of its respective α-shape. The r-shape graph does not contain any point not having a neighbor within its r-distance but such an isolated point is a vertex of the respective α-shape. In addition to the isolated points, other vertices may exist in the α-shape ($\alpha = -1/r$) that do not occur in the respective r-shape. For example, consider an α-extreme point P_k such that $C_r(P_k) \not\subset \bigcup_{i=1}^{n} \mathrm{int}(D_r(P_i))$ but $C_r(P_k) \subset \mathrm{int}(\bigcup_{i=1}^{n} D_r(P_i))$. Then there exists a point Q on $C_r(P_k)$ such that if it belongs to any disk $D_r(P_i)$, its boundary $C_r(P_i)$ passes through Q. Since P_k is not an r-extreme point, at least the boundary of four disks intersects at Q so that Q is covered by the interior of the union of disks. All edges in the α-shape with one end at P_k are *weak edges* that do not occur in the respective r-shape. Here, by a weak edge we mean an edge that disappears when the value of the graph parameter is slightly changed. In this case, if α is slightly decreased from $(-1/r)$, these weak edges disappear. Thus, the r-shape is more *stable* than the respective α-shape (i.e., when $\alpha = -1/r$).

It is clear that the r-shape is different for different values of r. To get a perceptually acceptable shape, a suitable value of r should be chosen, and there is no closed form solution to this problem. We have not come across any literature on proper selection of the equivalent of r, i.e., α. In principle, one should be able to get an appropriate value of r, called the *structuring radius*, from the dot pattern itself. (The disk centered at the origin and with radius of appropriate length r is said to be the *structuring disk*.) Note that for each s_i we can find an r-shape where $r = \sqrt{2}s_i$. However, we shall consider the r-shape corresponding to the ε-measure of dispersion. The selection of the structuring radius is justified by the possibility of the existence of a single point in diagonally opposite corners of two connected grids.

Definition 3.1. The quantity $s\sqrt{2}$ is called a structuring radius of *ε-stability* (or simply structuring radius).

We would like to point out here that "perceptual structure'' of S cannot be defined uniquely. It will vary from one person to another to a small extent. The ε-factor introduces such a notion in the structuring radius. If we fix a value of ε in a suitable range we obtain an acceptable s-shape (based border) and r-shape. For more or less evenly distributed DP, ε in the range 0.3–0.5 (typically 0.4) is a good choice (see Fig. 3).

3.1. *Characterization of Inconsistency in r-Shape*

Due to local irregularity in distribution on the pattern, some edges in the r-shape (even for a structuring radius r) are redundant and inconsistent with the perceptual structure. This inconsistency can be characterized by:

(i) one or more simple closed contours of negligible length (e.g., A in Fig. 5k);

(ii) multiple paths between two branch vertices (e.g., multiple path exist between B and C in Fig. 5k).

From a graph-theoretic point of view a *successor listing* of a graph is a data structure used for the graph representation. As in the r-shape graph, where the ratio of the number of edges (e) to the number of vertices (v) is very low, this successor listing is very convenient for storage, retrieval, and manipulation of the graph. In general, the structure of a successor listing is as follows. After assigning to the vertices in any order the numbers $1, 2, \ldots v$, a linear array is constructed for each vertex k whose elements are the vertices which are adjacent to k [27].

A successor listing type of data structure, referred to as *r-shape listing*, is used to clean the border of DP generated by the r-shape by tracing over it the route(s) from one r-vertex to another. Its composition and the way of tracing by using the listing are described in detail in the section on implementation.

In the r-shape, a simple closed curve of length less than the perimeter of the structuring disk ($\approx 6 \times r$, where 6 is an approximation of 2π) is considered an inconsistent loop because of its negligible length. However, since the distribution of points in a DP is more or less even, the average length of r-edges is approximated by r. Thus, a simple closed loop in the digital implementation is considered inconsistent if the total number of r-edges on the loop is less than 6.

If more than one path is detected between two branch vertices then such a path, if not the shortest one, will be deleted if the total Euclidean length of that path and the smallest route between those branch vertices is smaller than the perimeter of the structuring disk. However, in digital implementation, as in the above case, to determine the validity of a path between the two branch vertices we consider only the number of edges in the closed loop. A path between two branch vertices is considered inconsistent if the total number of r-edges on this path and also on the corresponding shortest path is less than 6.

4. DIGITAL DOMAIN IMPLEMENTATION

Let $S = \{P_1, P_2, \ldots, P_n\}$ be a set of n discrete points (a discrete point is equivalent to a pixel in image processing terminology) where $P_i = (x_i, y_i) \in \mathbb{J}^2$ is the integer lattice and let each P_i have a label or color i. Let I be a binary image where almost all object pixels are disconnected but densely and more or less evenly distributed in a subregion of I. Then this image I can be considered an image of a dot pattern. From I, the set S can be constructed by mere scanning.

4.1. *Computation of Dispersion Matrix: Estimation of s and r*

Step 1. Find

$$a \leftarrow \min_{i=1}^{n} x_i; \quad b \leftarrow \max_{i=1}^{n} x_i;$$

$$c \leftarrow \min_{i=1}^{n} y_i; \quad d \leftarrow \max_{i=1}^{n} y_i;$$

$$A(W) \leftarrow (b - a) \times (d - c).$$

(W is the optimal isothetic rectangular region whose four vertices are respectively, top-left corner, $\omega_1(a, c)$; top-right corner, $\omega_2(a, d)$; bottom-left corner, $\omega_3(b, c)$; bottom-right corner, $\omega_4(b, d)$.)

Compute

$$s_0 \leftarrow \sqrt{\frac{A(W)}{n}};$$

$$M_0 \leftarrow \left\lceil \frac{b-a}{s_0} \right\rceil;$$

$$N_0 \leftarrow \left\lceil \frac{d-c}{s_0} \right\rceil.$$

Set $t \leftarrow 1$; $s_{-1} \leftarrow A(W)$.

Step 2. Initialize the entries of two matrices of order $M_{t-1} \times N_{t-1} : a_{lm} \leftarrow a'_{lm} \leftarrow 0$.

Set count number of nonnull buckets: $c_t \leftarrow 0$.

At each point P_i (for $i = 1, 2, \ldots, n$), compute

$$l \leftarrow \left\lceil \frac{x_i - a}{s_{t-1}} \right\rceil;$$

$$m \leftarrow \left\lceil \frac{y_i - c}{s_{t-1}} \right\rceil;$$

$$a_{lm} \leftarrow a_{lm} + 1;$$

$$\text{if } (a'_{lm} \equiv 0) \quad c_t \leftarrow c_t + 1;$$

$$a'_{lm} \leftarrow 1.$$

Step 3. Compute

$$((g_{lm})) \leftarrow ((a_{lm}));$$

$$((b_{lm})) \leftarrow ((a'_{lm}));$$

$$s_t \leftarrow s_{t-1} \times \sqrt{\frac{c_t}{n}}$$

$$M_t \leftarrow \left\lceil \frac{b-a}{s_t} \right\rceil;$$

$$N_t \leftarrow \left\lceil \frac{d-c}{s_t} \right\rceil.$$

Step 4.

If $\left(\frac{s_{t-2} - s_t}{s_{t-1}} \leq \varepsilon\right)$ then i. $s \leftarrow \lceil s_{t-1} \rceil$;

ii. structuring radius is $r \leftarrow [s_{t-1}\sqrt{2}]$;

iii. dispersion matrix is $((g_{lm}))$;

iv. its projection is $((b_{lm}))$;

else set $t \leftarrow t + 1$; goto Step 2.

Note that for estimating the structuring radius, explicit computation of $((g_{lm}))/((b_{lm}))$ is not necessary, but at each iteration the number of nonnull grids is required.

4.1.1. *Computation of an s-Shape-Based Border.*

Step 1. Find the projection of the dispersion matrix $((g_{lm}))$.

Step 2. Smooth $((b_{lm}))$ by a binary morphological closing with a unit disk. (The unit disk is a 3×3 window having center at the (1, 1) position.)

Step 3. The border of the resulting output of the closed version after its s-times magnification followed by a translation to the point w_1, is considered as an approximate border of the DP.

Examples of s-shape-based borders along with the inputs are shown in Figs. 1–5.

4.2. *Digital r-Shape and Its Computation*

The *discrete disk* with radius r around P_i is a set $DD_r(P_i)$ of pixels, $P = (x, y)$ so that the distance of every P from P_i satisfies the inequality $d(P, P_i) < r + \frac{1}{2}$. Moreover, the label of each pixel P in $DD_r(P_i)$ is i.

Let $\mathscr{T}_r$ be the set of ordered pairs defined as

$$\mathscr{T}_r = \{(P, L(P)) | L(P) = \max_{P \in \mathrm{DD}r(P_i)} i\}.$$

The border of $\mathscr{T}_r$, denoted by $\mathrm{bdr}(\mathscr{T}_r)$, is defined as

$$\mathrm{bdr}(\mathscr{T}_r) = \{(P, L(P))|\text{ at least one of the 4-neighbors of } P \text{ has label } 0\}.$$

The interior of $\mathscr{T}_r$, $\mathrm{int}(\mathscr{T}_r)$ is defined as

$$\mathrm{int}(\mathscr{T}_r) = \mathscr{T}_r - \mathrm{bdr}(\mathscr{T}_r).$$

Let the set of labels occurring in $\mathrm{bdr}(\mathscr{T}_r)$ be $\mathscr{L}(\mathrm{bdr}(\mathscr{T}_r)) = \{L(P)|(P, L(P)) \in \mathrm{bdr}(\mathscr{T}_r)\}$. P_i is said to be an r-extreme pixel in S iff $i \in \mathscr{L}(\mathrm{bdr}(\mathscr{T}_r))$.

Consider the 8-connected components of $\mathrm{bdr}(\mathscr{T}_r)$, each having the same label. The number of such components is greater than or equal to $\#\mathscr{L}(\mathrm{bdr}(\mathscr{T}_r))$. In general, a component with two or more pixels will have exactly two end pixels, viz., the pixels having exactly one 8-neighbor in the component.

Let $\mathscr{V}$ be the set of r-extreme pixels in S. For P_i and $P_j \in \mathscr{V}$, $\overline{P_iP_j}$ will be an r-edge of S if an end pixel of a connected component with label i in $\mathrm{bdr}(\mathscr{T}_r)$ and an end pixel of another connected component with label $j (i \neq j)$ are 8-connected. Then P_j is an r-adjacent of the r-vertex P_i. The degree of the r-vertex of P_i is the total number of r-adjacents of P_i.

The r-shape of S is expressed by a diagram consisting of r-edges.

4.2.1. *Computation of r-Shape.*

Step 1. Create a structuring disk of radius r.

Step 2. $\mathscr{T}_r$ is computed by placing the structuring disk as a mask centered at every point $P_i(x_i, y_i)$ $(i = 1, 2, \ldots, n)$ as follows. Each pixel Q in the square neighborhood (of side-length $2r + 1$) of P_i is considered. If Q is under the disk $DD_r(P_i)$ then the label i is assigned to Q.

Step 3. The square neighborhood (of side-length $2r + 1$) of P_i in $\mathscr{T}_r$ is scanned for each i. If in that neighborhood, a pixel P exists with label i such that a background pixel (i.e., pixel with label 0) exists in its 4-neighborhood then P_i is an r-extreme pixel. Further, if such a P has an 8-neighbor Q with distinct label $j (i \neq j > 0)$ such that Q has also a background pixel in its 4-neighbor then P_i is an r-vertex and $\overline{P_iP_j}$ is an r-edge of S. But if such a Q does not exist for any P, then that r-extreme pixel is considered an isolated one.

For P_i, find all the mutually distinct labels of the Qs with properties stated above and the ith row of successor listing is constructed. Let the stored labels be i_j $(j = 1, 2, \ldots, k)$ where $i_j \in \{1, 2, \ldots, i - 1, i + 1, \ldots, n\}$. Then the r-adjacents of the r-vertex P_i are $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$.

The r-shape is a combination of r-edges which can be drawn using the successor listing. In this case, the successor listing is composed of n linear arrays where the ith array contains the label of r-adjacent(s) of P_i. If the ith array of the successor listing contains $j (> i)$, then the r-edge $\overline{P_iP_j}$ is drawn. (Note that each edge in the r-shape diagram is drawn only once.)

The r-shape computation is illustrated in Figs. 2a–d. For easy understanding and representation, instead of integers, letters A–X are used in their natural orders on 24 pixels present in the pattern. The labeling of the pixels by a raster scan is shown in Fig. 2a. The output of Step 2, $\mathscr{T}_r$, is shown in Fig. 2b. The label of r-adjacent vertices detected in Step 3 are shown in Fig. 2c. The resultant r-shape is the diagram of Fig. 2d.

4.3. *Extraction of Perceived Border from the r-Shape*

To get an automatic border from the r-shape that is consistent with the intuitive shape of the dot pattern the structuring radius r is first estimated. From that r, the r-shape is computed.

4.3.1. *Listing of r-Shape.* As stated earlier, due to local irregularities, even under the structuring radius r, some edges in an r-shape may appear inconsistent with respect to the perceived boundary of the DP. A modified successor listing, referred to as an r-shape listing, is constructed to trace these edges and is denoted as $\mathrm{List}_r(S)$. This can be stored in a matrix (say, L) of order $n \times \lceil 2\pi(r + 1) \rceil$ where each of the first two entries of the ith $(i = 1, 2, \ldots, n)$ row, i.e., $L[i, 1]$ and $L[i, 2]$, contains the degree of P_i and the remaining entries are the labels of its r-adjacent vertices. The first two entries of each row are henceforth referred to, respectively, as deg and deg′ and the remaining entries constitute the *string of r-adjacents* (or simply the *adjacent string*). The initial choice of $L[i, 1] = L[i, 2] = \deg = \deg'$ for each r-vertex of the pattern is justified in the following algorithm on border cleaning.

4.3.2. *Finding the Consistent Edges in the r-Shape.* After constructing the $\text{List}_r(S)$, the algorithm for selecting the consistent r-edges and removing the inconsistent ones from the r-shape is performed through structural analysis using the following three steps: (1) tracing of primitive linear structures, (2) selection of path(s) from multiroutes in between branch vertices, and (3) removal of simple closed loops of negligible length. Note that from this r-shape listing all possible paths from an r-vertex can be found.

In step (1), a primitive linear structure means a path of r-vertices whose one end is free, i.e., a vertex of degree 1, while the other end is either another vertex of degree 1 (meaning a simple linear structure) or a branch vertex (meaning a tail-like structure). A linear structure can be found starting from an r-vertex of degree 1 and including the successive r-adjacent vertices until a vertex of degree 1 or a branch vertex is found. Let this output string be denoted by lin-str. An efficient lin-str computational approach is described below.

By checking the first column of L, one free end of a linear structure is detected (provided it exists). Let P_i be such an end vertex. Then both values $\deg(i)$ and $\deg'(i)$ are replaced by 0 and the first entry of lin-str is set to i.

Now the label (say i_1), of the adjacent vertex, P_{i_1}, of P_i on the linear structure, is the value of $L[i, 3]$ (which is the first and only non-zero value of the ith string of r-adjacents). Therefore, set lin-str$[2] = i_1$. Find the position with value i in the i_1th string of r-adjacent vertices. If $L[i_1, 2 + l]$ (for $l > 0$) is equal to i then replace its value by 0 and swap it with the content of $L[i_1, 2 + \deg'(i_1)]$. This processing has been done to avoid a redundant search because with this operation a null entry occurs only at the end of an adjacent string. This removal of traced r-vertex label from the r-adjacents string is referred to as *conditional popping.*

After the conditional popping, $\deg'(i_1)$ is reduced by 1. If $\deg(i_1)$ is equal to 2, both $\deg(i_1)$ and $\deg'(i_1)$ are set to 0 and if $L[i_1, 3] = i_2$, then set lin-str$[3] = i_2$.

This process is continued until a label, say i_k, is found whose deg is not equal to 2. Then P_{i_k} is either a branch vertex (of $\deg(i_k) > 2$) (which means the linear structure is tail-like) or another free end (if $\deg(i_k) = 1$) of the simple linear structure under consideration. Note that the *length* of the lin-str$[i, i_1, i_2, \ldots, i_k]$ minus 1 is the number of r-edges in the linear structure (which is k here). The diagram of the traced primitive linear structure, which is part of the r-shape, can be drawn by following the successor labels in lin-str$[i, i_1, i_2, \ldots, i_k]$ and joining the corresponding vertices.

After all the edge lying on the primitive linear structures are traced and removed the $\text{List}_r(S)$ contains only the edges on paths between branch vertices and the edges in the simple closed loops. Note that any remaining path from a branch vertex always ends at another branch vertex. (A loop may be formed at the same branch vertex.)

Since the perimeter of a discrete disk of radius r is of order $O(\lceil 2\pi r \rceil)$, the degree of an r-vertex is also of order $O(\lceil 2\pi r \rceil)$. Thus, all paths from a branch vertex which are not (already traced) tail-like linear structures can be stored in a matrix, say, M of size $k \times n$, where $k > \lceil 2\pi r \rceil$. For a dense pattern, we have $k \leqslant n$.

Except at the starting vertex, the tracing procedure of multipaths in between branch vertices is exactly the same as that for primitive linear structure. The deg column of $\text{List}_r(S)$ is scanned and if for the bth row we have $\deg(b) > 2$ and $\deg'(b) > 0$ then the vertex P_b is considered the starting branch vertex and is tagged by a pointer (let us call it *branch vertex pointer*). The labels of the end branch vertices to which P_b is path connected, the number of r-edges on the path (path length), and their row position in M are stored in three linear arrays, referred to as X, Y, and Z, respectively. From X, the existence of multiroutes between two branch vertices can be easily found. If b' is the label of an end branch vertex in X that repeats t times then it signifies that there exist t paths between P_b and $P_{b'}$ ($b < b'$). The path of minimum length between P_b and $P_{b'}$ can be found by checking the values in Y and corresponding Z. Consider the rest of the paths between P_b and $P_{b'}$. Any one such path and the shortest path makes a closed loop. If the total number of r-edges on this loop is less than six then the path is considered inconsistent.

The consistent paths from the branch vertex (which is tagged by the pointer at present) are identified by the labels of Z. Now each consistent path can be drawn by following the successor labels of r-adjacent vertices ordered in the row of M identified by Z.

In this way, all branch vertices and their connecting paths are detected and r-edges on these paths are removed from $\text{List}_r(S)$. Note that if there exist multiroutes between two branch vertices, the tracing over the path never repeats. For example, if for path-connected branch vertices P_b and $P_{b'}$, the label b is less than the label b' then all paths between P_b and $P_{b'}$ are processed when the branch vertex processing pointer is at the bth row of $\text{List}_r(S)$.

The rest in the $\text{List}_r(S)$ correspond only to those edges that lie on a simple closed loop where each r-vertex is of degree 2. Now to search an r-vertex on such a closed loop, the deg′ column is scanned from the top. If P_i is the lowest labeled r-vertex on a simple loop then $\deg'(i) = 2$. This label i is placed on the first position of the output string. Both $\deg(i)$ and $\deg'(i)$ are set to zero. The first member of the string of adjacents, say, i' is tagged as the label of the end r-vertex. If the label of the second adjacent vertex of P_i is i_1, then i_1 is stored in the second position of the output string. After condition popping at the i_1th adjacent string, the other adjacent vertex of P_{i_1} (say P_{i_2}) is found. Until an r-vertex of label i' is found, the same procedure,

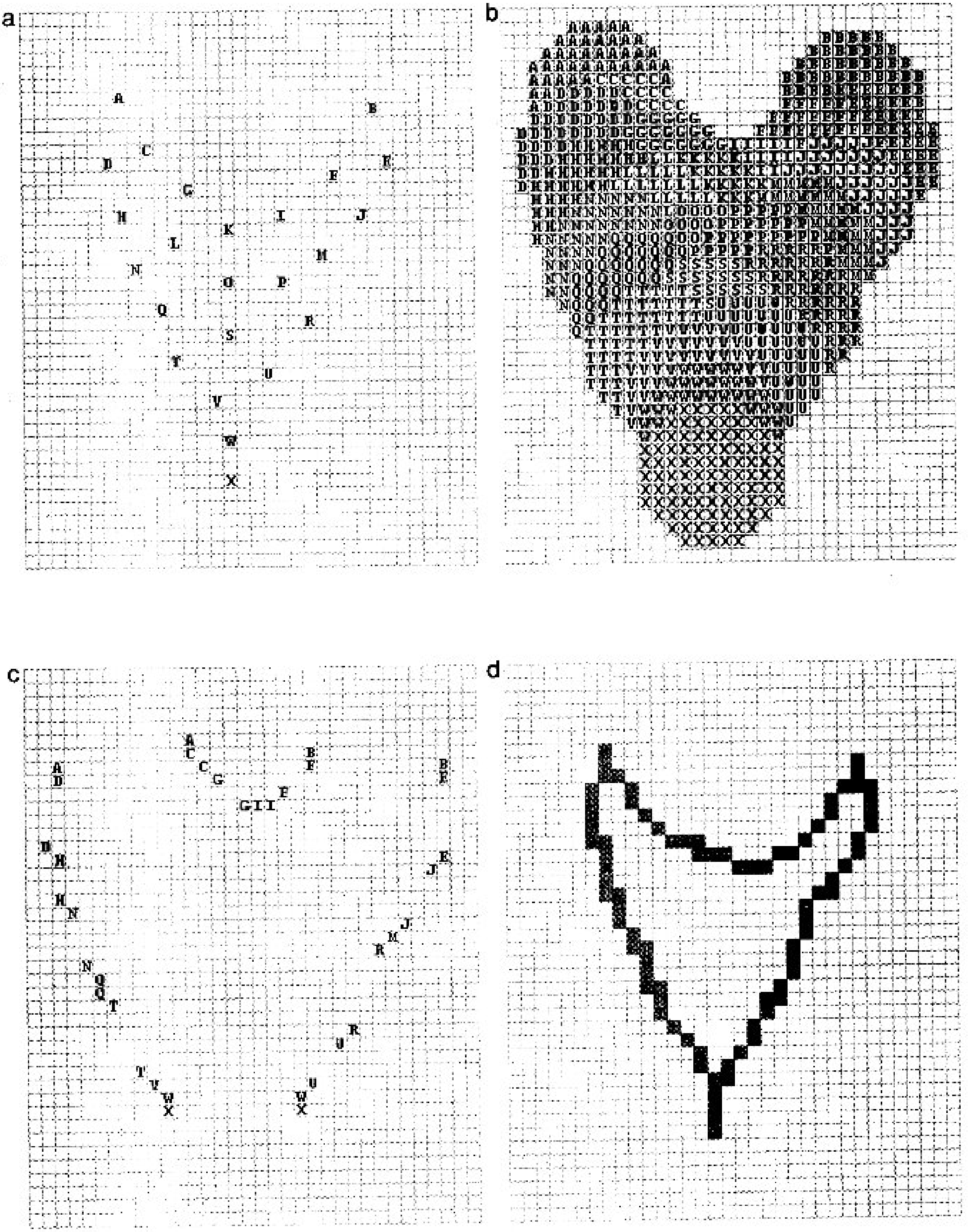
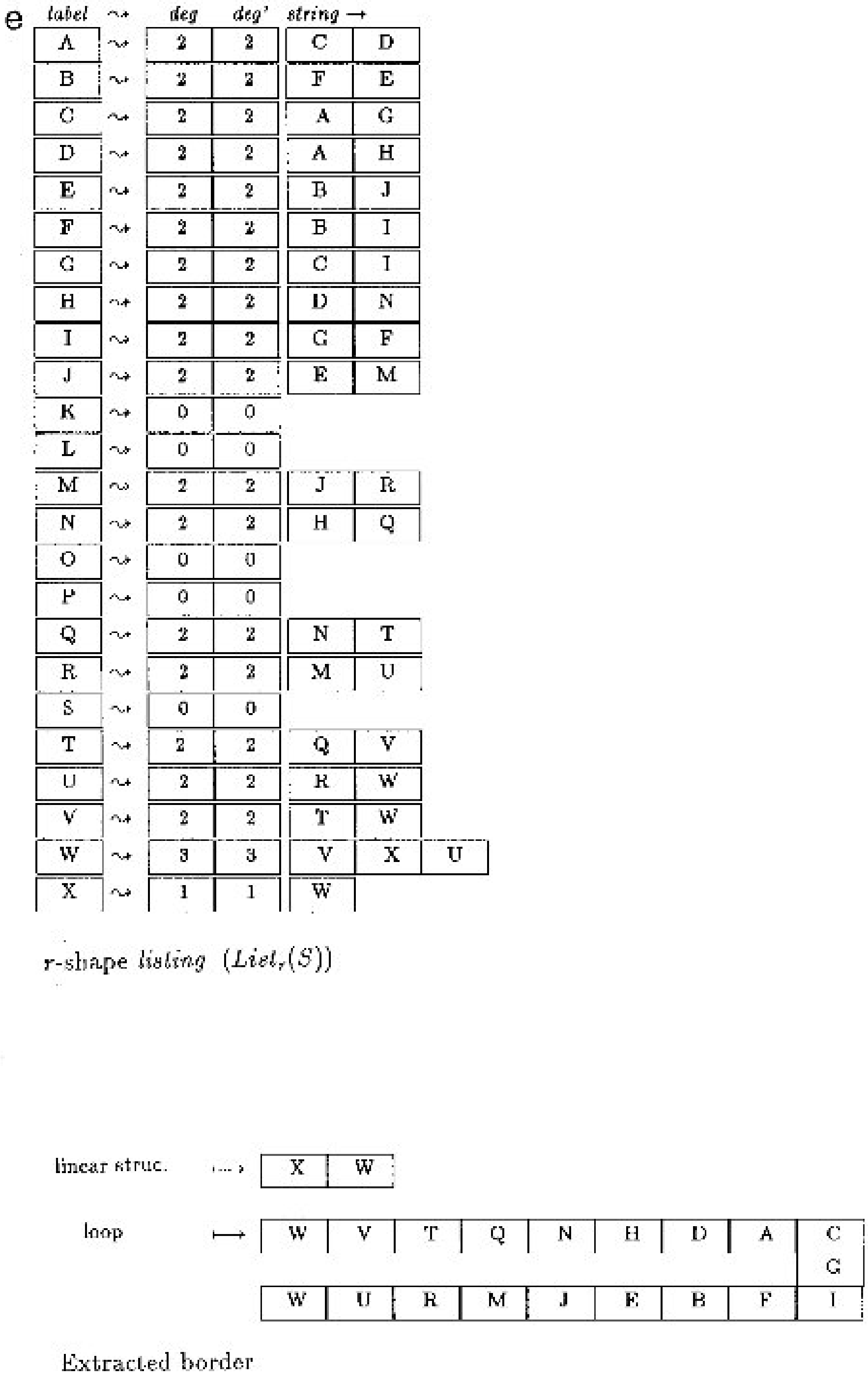


FIG. 2. Computation of the *r*-shape. (a) The labeling of the pixels by a raster scan (Step 1). (b) $\mathcal{T}_r$, the union of labeled digital disks (Step 2). (c) The label of detected *r*-adjacent vertices (Step 3). (d) The resultant *r*-shape. (e) The *r*-shape listing and extracted border in the form of strings.



label	⤳	deg	deg'	string →		
A	⤳	2	2	C	D	
B	⤳	2	2	F	E	
C	⤳	2	2	A	G	
D	⤳	2	2	A	H	
E	⤳	2	2	B	J	
F	⤳	2	2	B	I	
G	⤳	2	2	C	I	
H	⤳	2	2	D	N	
I	⤳	2	2	G	F	
J	⤳	2	2	E	M	
K	⤳	0	0			
L	⤳	0	0			
M	⤳	2	2	J	R	
N	⤳	2	2	H	Q	
O	⤳	0	0			
P	⤳	0	0			
Q	⤳	2	2	N	T	
R	⤳	2	2	M	U	
S	⤳	0	0			
T	⤳	2	2	Q	V	
U	⤳	2	2	R	W	
V	⤳	2	2	T	W	
W	⤳	3	3	V	X	U
X	⤳	1	1	W		

r-shape *listing* $(List_r(S))$

linear struc.	⟶	X	W

loop	⟼	W	V	T	Q	N	H	D	A	C
										G
		W	U	R	M	J	E	B	F	I

Extracted border

FIGURE 2—*Continued*

i.e., conditional popping and storing at the output string, is repeated. If the number of r-edges is less than six, the r-edges on the loop are considered inconsistent. Otherwise, by tracing the successive labels in the output string, the corresponding r-edges can be joined by line segments. In this way all existing simple closed loops are traced out.

The diagram, that is composed of these so called consistent r-edges, is considered the perceived border of S.

The r-shape listing and the extracted linear structure, as well as the loop (at a branch vertex) existing in the r-shape of the dot pattern of Fig. 2a, are presented in Fig. 2e.

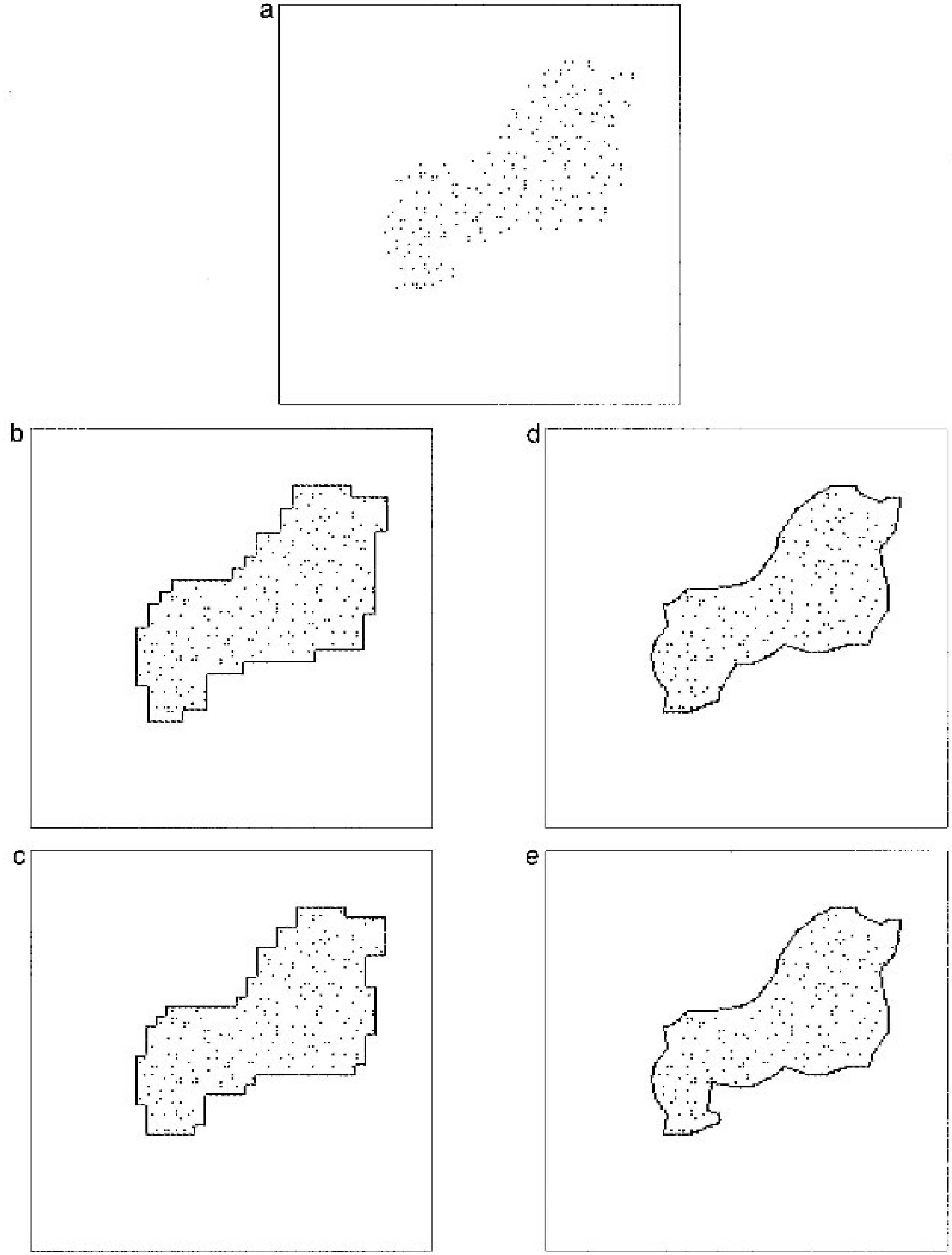


FIG. 3. The perceptual compatibility of border (ε lies in the range 0.3–0.5). (a) A "polar bear" shaped dot pattern. (b) The s-shape-based extracted border along with the DP (for $\varepsilon = 0.5$). (c) The s-shape-based extracted border with the DP (same for both $\varepsilon = 0.3, 0.4$). (d) The r-shape along with the DP ($\varepsilon = 0.5$). (e) The r-shapes for the DP are same for $\varepsilon = 0.3, 0.4$.

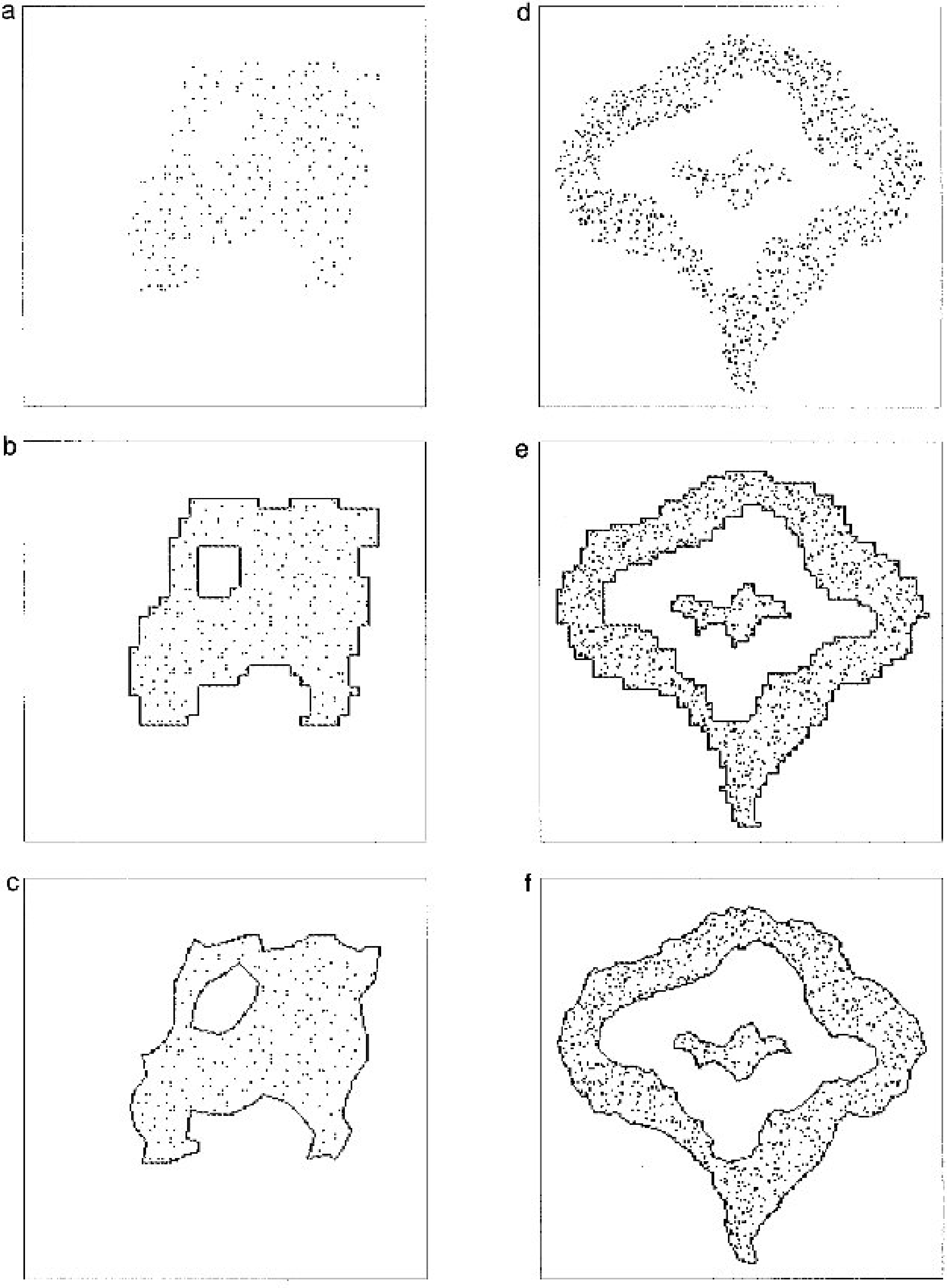


FIG. 4. Two more examples of dot patterns where *r*-shapes are free from inconsistent edges ($\varepsilon = 0.4$). (a) The input pattern, (b) the *s*-shape-based border, and (c) the *r*-shape, (d) the input pattern, (e) the *s*-shape-based border, and (f) the *r*-shape.

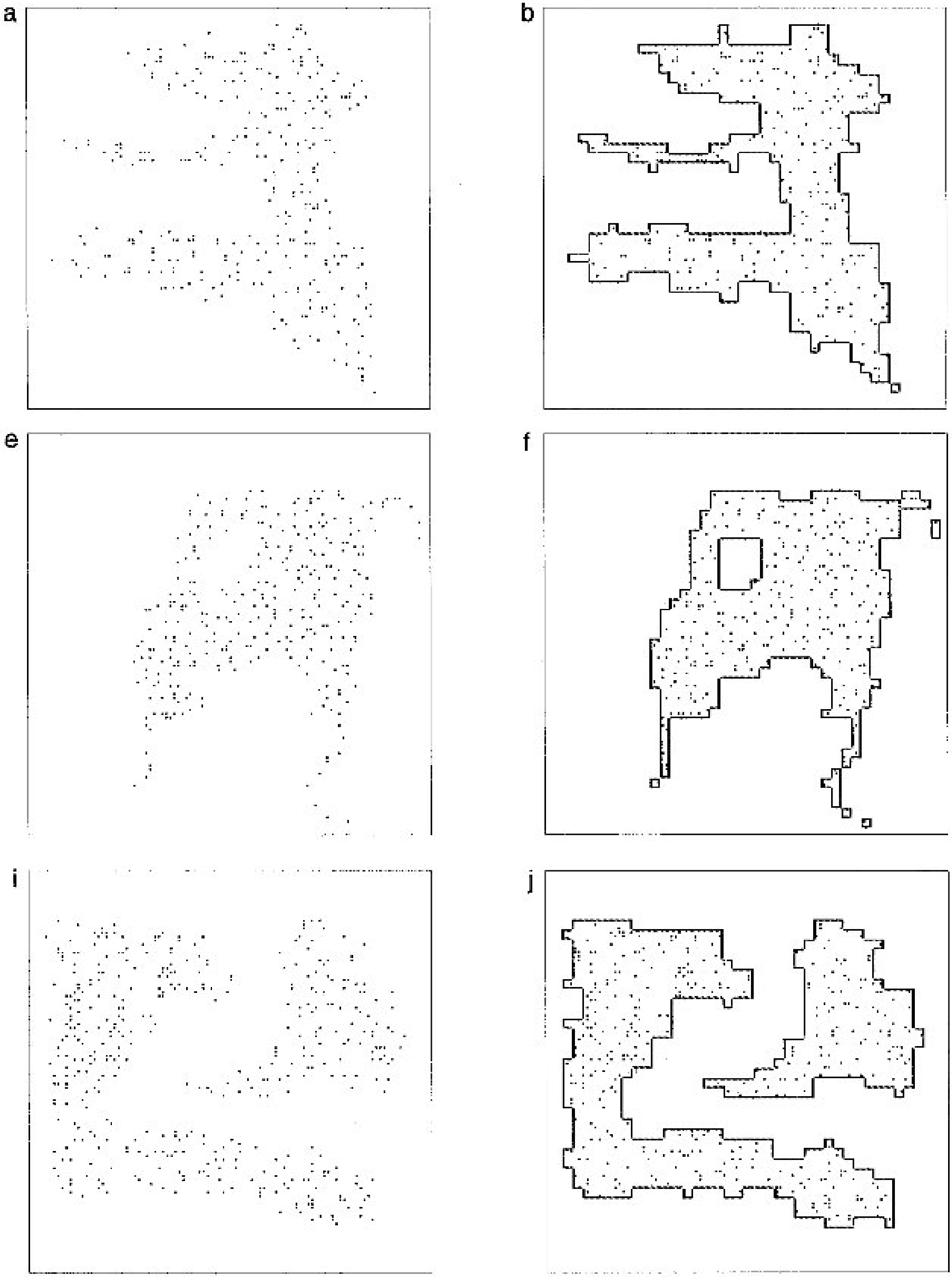


FIG. 5. Examples of dot patterns where *r*-shapes are not free from inconsistent edges ($\varepsilon = 0.4$). (a) The input pattern, (b) the *s*-shape-based border, (c) the *r*-shape, and (d) the final extracted border; (e) the input pattern, (f) the *s*-shape-based border, (g) the *r*-shape, and (h) the final extracted border; (i) the input pattern, (j) the *s*-shape-based border, (k) the *r*-shape, and (l) the final extracted border.

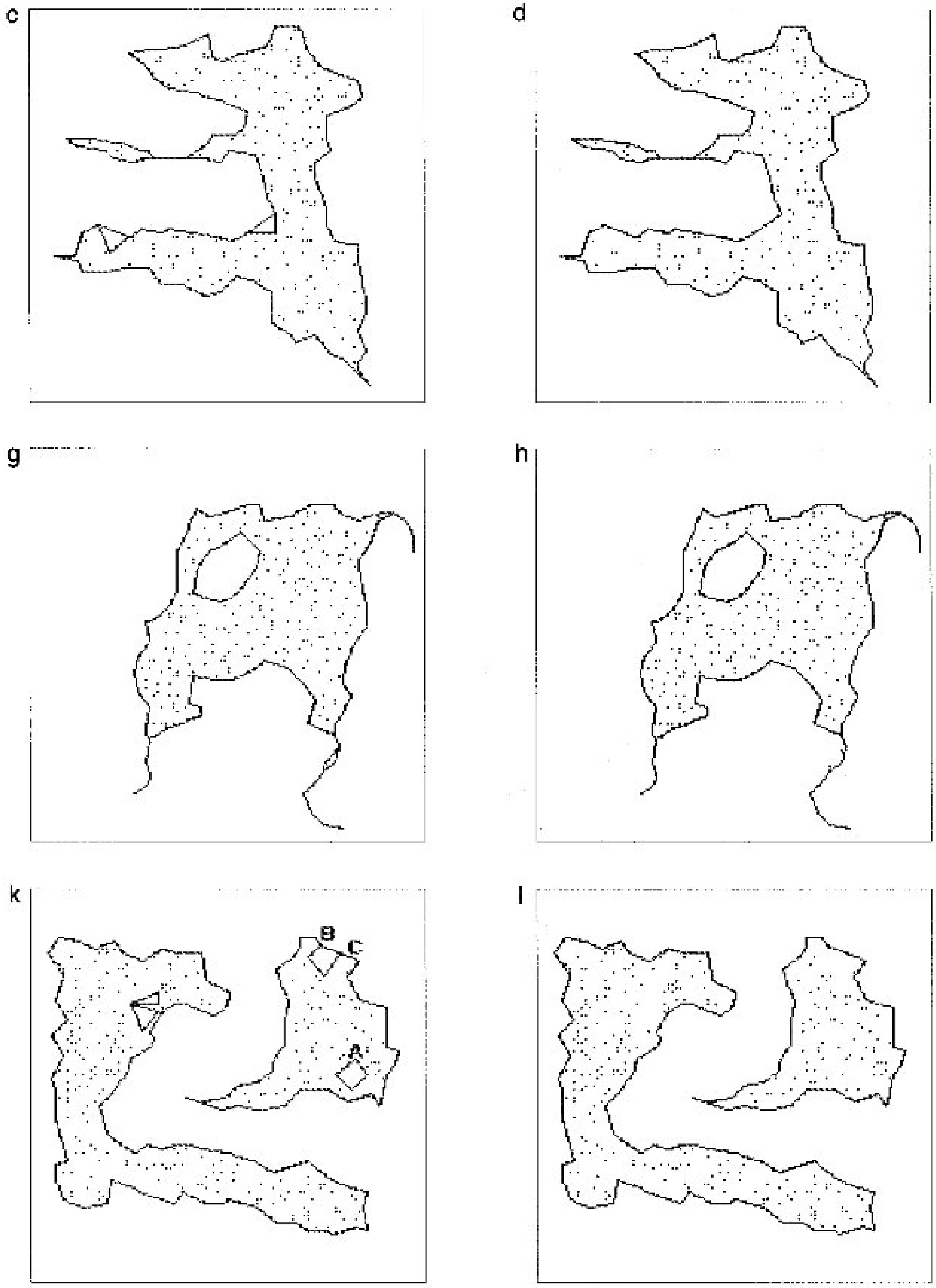


FIG. 5—*Continued*

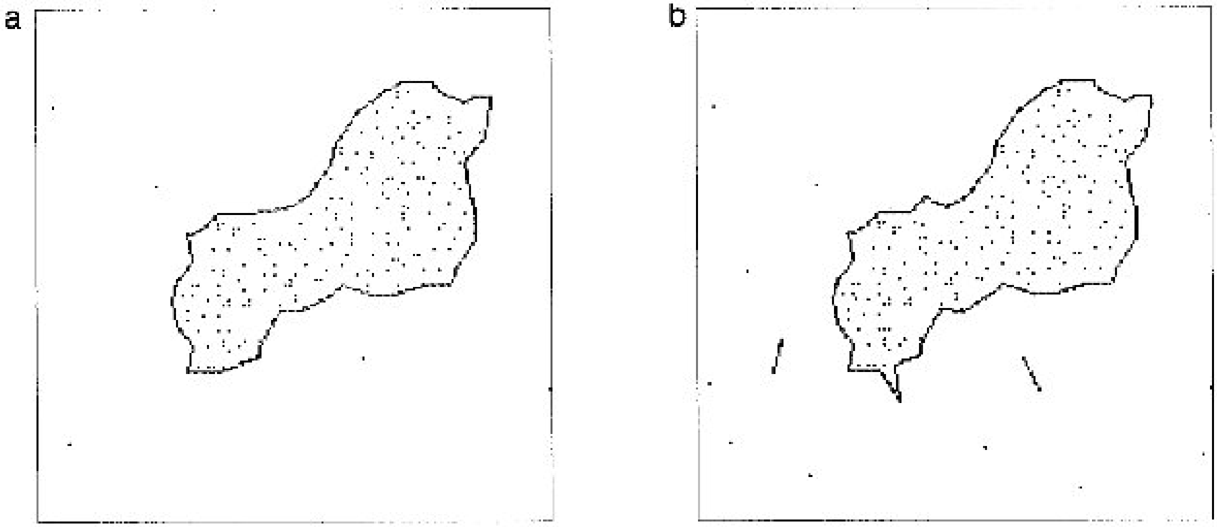


FIG. 6. Robustness in presence of noise. (a) The r-shape-based extracted border along with the input which is a corrupted version of Fig. 3a by 15-dB SNR noise. (b) The r-shape-based extracted border along with the input which is a corrupted version of Fig. 3a by 10-dB SNR noise.

5. EVALUATION OF PROPOSED APPROACHES

5.1. *Computational Complexity*

From the construction it is clear that for a given s, the border based on the s-shape is linear with respect to the cardinality of DP. We will show that the computation of the r-shape as well as the extraction of the perceived border, as a whole, is possible in linear time. First, the complexity of the r-shape computation is examined.

5.1.1. *r-Shape Computation Complexity.* Since generation of a discrete disk with radius r at each point of DP requires a constant time ($O(r^2)$), $\mathcal{T}_r$ can be computed in $O(n)$ time. As described in the algorithm on r-shape computation, for each point P_i, pixels in its square neighborhood with sides of length $2r + 1$ are scanned and processed. If P is a pixel in $\mathcal{T}_r$ with label i in the square neighborhood of P_i then its 8-neighbors are also traced. If in the 8-neighborhood of P, there exists a background pixel as well as a pixel Q with a label distinct from i, then the 4-neighborhood of Q is scanned to detect whether Q has any background pixel. Therefore the worst case computation for each pixel in square neighborhood of P_i requires 32 ($= 8 \times 4$) comparisons. Thus, checking whether a pixel is an r-interior, an isolated r-extreme pixel, or an r-vertex, as well as finding its r-adjacents, requires a constant time. Therefore, the r-edges of an r-shape can be detected in $O(n)$ time.

5.1.2. *Perceived Border Extraction Complexity.* For eximation of the structuring radius, it is evident that in each iteration, the computation of $((b_{lm}))$ is linear with respect to n.

In Proposition 3.1, we have shown that in the analog case, the limiting size of the bucket is attained only after a finite number of steps. However, in the digital domain, since the structuring radius is an integer, in the worst case $2(\lfloor s_2 \rfloor + 1)$ can be considered as an upper bound of iteration number, where s_2 is the estimated length of the cell after the second bucketing of S. Since for a dense pattern $s_2 \ll n$, the structuring radius estimation can be considered linear with respect to n.

As discussed earlier in the case of r-shape computation, the time required for computing the listing, $\text{List}_r(S)$, is also linear with n. Since the degree of an r-vertex is $O(\lceil 2\pi r \rceil)$, in a worst case situation, the total number of r-edges is $O(\lceil 2\pi r \rceil n) \approx O(n)$. Now recall that each r-edge of the r-shape is traced over $\text{List}_r(S)$ only once. It can be easily verified from the detailed description of the algorithm that each of the three modules, namely tracing over the linear structure, selection of path(s) from multiroutes between two branch vertices, and finally the selection from simple closed loops, requires only $O(n)$ computation. Thus, finding the consistent edges of the r-shape requires $O(n)$ time.

From the above discussion it is concluded that the time complexity of extracting the border (which is reasonably close to the intuitive shape) is linear with respect to the cardinality of the pattern.

5.2. *Experimental Results and Discussion*

We have tested the proposed algorithms on several synthetic dot patterns. The results are quite satisfactory. We have experimentally found that if ε lies in the range 0.3–0.5, the extracted border is compatible with the perceptual

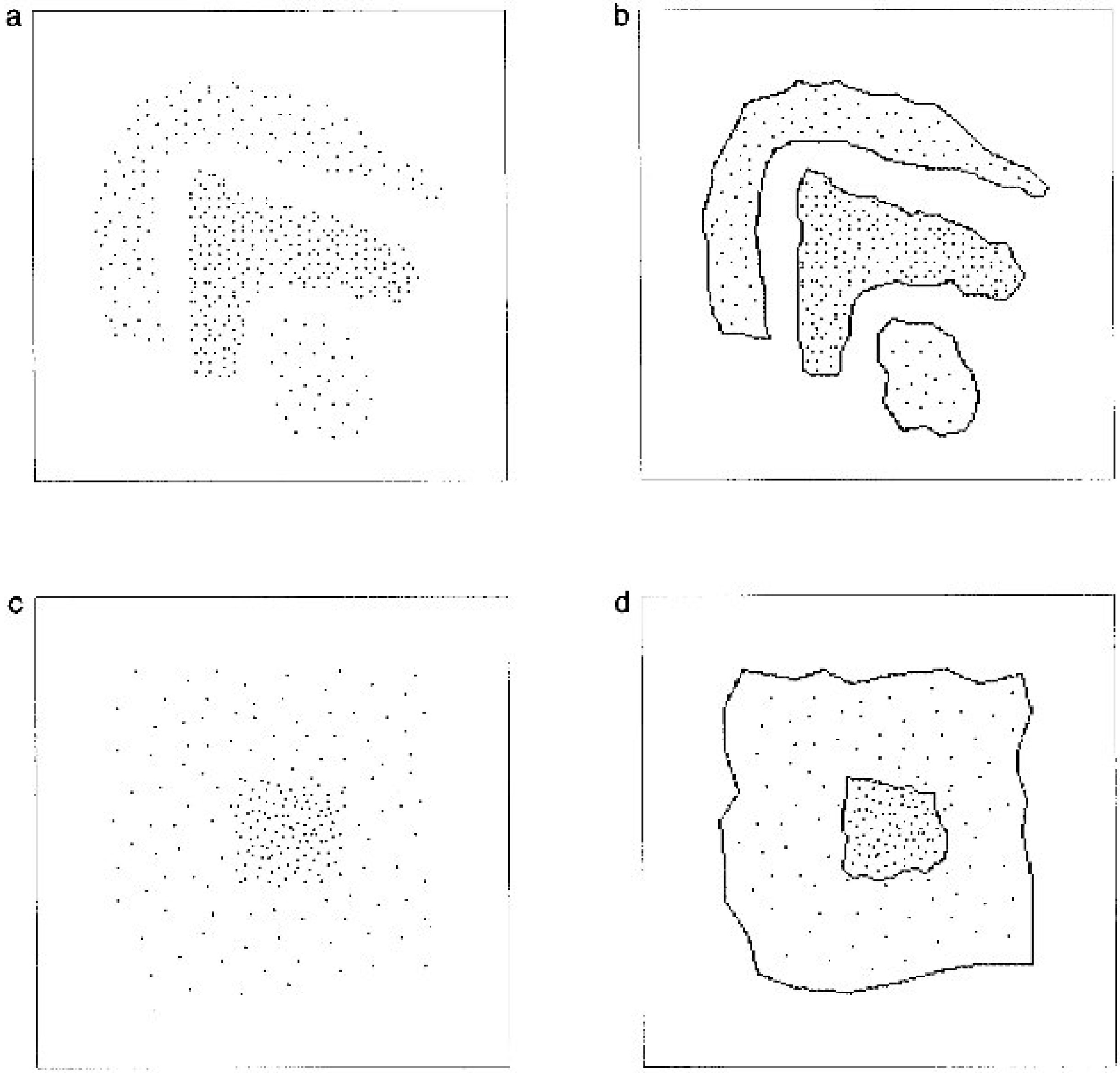


FIG. 7. Situation for mixed point patterns. (a) The components of the input are disconnected. (b) The r-shape-based extracted border with the input. (c) The denser component is embedded in lighter pattern. (d) Extracted border with the input.

border of the dot pattern. For a given DP (Fig. 3a) the s-shape-based extracted borders for $\varepsilon = 0.3, 0.4, 0.5$ (the shapes for $\varepsilon = 0.3, 0.4$ are in fact the same) are shown in Figs. 3b, c, respectively. For the same DP, the r-shapes (along with the input) are given in Figs. 3d, e.

Over a few other typical dot patterns the borders based on the s-shape, r-shapes, and finally extracted borders (these are proper subgraphs of respective r-shapes) are presented in Fig. 5. In all these cases ε has been fixed at 0.4.

Our proposed algorithm is the first of its kind to extract the perceived border for a given DP in $O(n)$ time. On the other hand, any algorithm for shape extraction based on the Voronoi diagram, Delaunay triangulation, the minimum spanning tree [28], or the sphere of influence graph needs at least $O(n \log n)$ computation [29]. In a digital case, the number of iterations required for getting the structuring radius of a DP is very low. For example, in all the above listed figures of the r-shape the number of iterations required are less than four. Except for a very few operations, the computation is based on integer addition, subtraction, and comparison. Thus, our method of border extraction of dot pattern is very fast and efficient.

The proposed algorithms for the structuring radius estimation and the r-shape (r-shape listing) are locally computable and can be directly implemented on parallel machines. In [20], where the concept of *r*-shape was first introduced, we proposed an algorithm for *r*-shape computation that could be implemented in single instruction multiple data computers viz. on the CRCW–SM (concurrent read, concurrent write, shared memory) computer model.

Our method can be easily implemented on hardware where the proposed border extractor can be treated as a low-level vision operator.

Though the proposed method is designed for a single set of dots pattern, the boundary extraction algorithm is directly applicable also for *disconnected dot patterns* provided the dispersion measure of individual components is more or less the same. By disconnected dot patterns, we mean that the perceptual structure contains topologically distinct components and the minimum distance between any two DPs is more than twice of structuring radius. For example, in Fig. 4d and Fig. 5i ($\varepsilon = 0.4$), the number of such disconnected dot patterns is 2.

Our proposed border extractor of dot patterns is robust. Consider the case of noisy dot patterns, where we assume that some points that are not of the original S are introduced (let these points constitute the set S') and a few points are deleted from the original DP (let S'' be the collection of these deleted points). Then $\overline{S} = S' \cup (S/S'')$ can be considered a noisy version of the dot pattern S. Such a situation can arise when a binary image of a DP is degraded by salt and pepper noise. If this noise level is low, the proposed perceived border extractor works well. Two results are shown in Fig. 6. Figures 6a, and 6b illustrate the borders (along with the respective inputs) when the pattern of Fig. 3a is corrupted by 15 dB and 10 dB of SNR noise, respectively.

To find the shape of *mixed point patterns*, where it consists of dot patterns of different ε-dispersion measures, a cut-one-out approach is outlined below.

- Find the dispersion matrix for the mixed point patterns and find the most dense component in it.
- Trace back the induced pattern of this most dense component of the matrix in the original mixed pattern and extract its border separately.
- Remove the extracted part from the mixed patterns and repeat the whole process.

This approach should be modified when the patterns are not disconnected (i.e., when the distance between two DPs is less than the sum of their respective structuring radii). At the intersecting zone, the border of the denser pattern may be considered the common separator and the border of the less dense pattern should be modified accordingly. Two cases of mixed dot patterns are shown in Figs. 7a and 7c. In the first case (Fig. 7a) patterns are disconnected but in the latter (Fig. 7c) a denser pattern is embedded in another DP. The results are shown in Figs. 7b and 7d, respectively. The details of the above approach will be discussed in a forthcoming report.

Note that our approach on external shape computation can be extended to data in higher dimensions. This approach can also be applied to data clustering where the final output in the form of a set of edges will give the boundary of a cluster.

ACKNOWLEDGMENTS

One of the authors (A. R. Chaudhuri) acknowledges the Council of Scientific and Industrial Research (CSIR), India, for providing him a research fellowship in pursuing his work. The authors are also thankful to the anonymous referee for constructive comments which greatly helped in revising the manuscript.

REFERENCES

1. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
2. O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint,* MIT Press, Cambridge, MA, 1993.
3. J. M. Chassery and A. Montanvert, Geometric representation of shapes and objects for visual perception, in *Geometric Reasoning for Perception and Action,* (C. Laugier, Ed.), Selected Papers Presented in a Workshop Held at Grenoble, France, September 16–17, 1991, pp. 163–182, Springer-Verlag, Berlin/New York, 1991.
4. H. Ogawa, Labeled pattern matching by Delaunay triangulation and maximal cliques, *Pattern Recognit.* **19**(1), 1986, 35–40.
5. R. Laurini and D. Thompson, *Fundamental of Spatial Information Systems,* The A.P.I.C. Series, Vol. 37, Academic Press, London, 1992.
6. P. J. Taylor, *Quantitative Methods in Geography: An Introduction to Spatial Analysis,* Houghton Mifflin, Boston, 1977.
7. A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams,* Wiley, New York, 1992.
8. M. R. Anderberg, *Cluster Analysis for Applications,* Academic Press, New York, 1973.
9. D. Chaudhuri and B. B. Chaudhuri, Multi-seed non-hierarchical clustering technique, *Int. J. Syst. Sci.* **26**(2), 1995, 375–385.
10. K. V. Mardia, *Statistics of Directional Data,* Academic Press, London, 1972.
11. E. Yodogawa, Quantitative measure of perceived orientation strength of dot patterns, in *IEEE Proc. of the International Conference on Systems, Man, and Cybernetics*, Tucson, Arizona, November, 1985, 584–588.
12. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data,* Prentice Hall, Englewood Cliffs, NJ, 1988.
13. P. J. Verveer and R. P. W. Duin, An evaluation of intrinsic dimensionality, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(1), 1995, 81–86.
14. S. Ranade and A. Rosenfeld, Point pattern matching by relaxation, *Pattern Recognit.* **12,** 1980, 269–275.
15. U. Shinji, Parameterized point pattern matching and its application to recognition of object families, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(2), 1993, 136–144.
16. C. Ronse, A bibliography on digital and computational convexity (1961–1988), *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(2), 1989, 181–190.
17. H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, On the shape of

a set of points in the plane, *IEEE Trans. Inf. Theory* **IT-29,** 1983, 551–559.

18. N. Ahuja and M. Tuceryan, Extraction of early perceptual structure in dot patterns: Integrating region, boundary, and component gestalt, *Comput. Vision Graphics Image Process.* **48,** 1989, 304–356.
19. G. T. Toussaint, A graph-theoretical primal sketch, in *Computational Morphology*, (G. T. Toussaint, Ed.), pp. 229–260. North-Holland, Amsterdam, 1988.
20. A. Ray Chaudhuri, S. K. Parui, and B. B. Chaudhuri, Computing the shape of a dot pattern in digital images in parallel environment, in *Proc. of the 3rd. International Conf. on Automation, Robotics, and Computer Vision, Singapore, November, 1994*, Vol. 1, pp. 474–478.
21. A. Ray Chaudhuri, S. K. Parui, and B. B. Chaudhuri, Extraction of perceptual dot pattern, *Proc. of the 2nd Asian Conference on Computer Vision*, Vol. III, pp. 62–65, National Technological University Press, Singapore, 1995.
22. B. B. Chaudhuri, Application of quadtree, octree, and binary tree decomposition techniques to shape analysis and pattern recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **7**(6), 1985, 652–661.
23. S. K. Parui, S. Sarkar, and B. B. Chaudhuri, Computing the shape of a point set in digital images, *Pattern Recognit. Lett.* **14,** 1993, 89–94.
24. J. D. Radke, On the shape of a set of points, in *Computational Morphology*, (G. T. Toussaint, Ed.), pp. 105–136. North-Holland, Amsterdam, 1988.
25. M. Tuceryan, A. K. Jain, and N. Ahuja, Supervised classification of early perceptual structure in dot patterns, in *Proc. of the 11th IAPR International Conference on Pattern Recognition, The Hague, The Netherlands, August 1992*, Vol. 2, pp. 88–91.
26. R. M. Haralick, and L. Shapiro, *Computer and Robot Vision*, Addison–Wesley, Reading, MA, Chap. 5, pp. 157–262, 1992.
27. N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Chap. 11, pp. 270–273, Prentice–Hall International, Englewood Cliffs, NJ, 1990.
28. C. T. Zahn, Graph theoretical methods for detecting and describing Gestalt clusters, *IEEE Trans. Comput.* **C-20,** 1971, 68–86.
29. F. P. Prepareta and M. I. Shamos, *Computational Geometry: An Introduction*, Chap. 5, pp. 185–225, Springer-Verlag, Berlin/New York, 1985.