Computer Representation of Planar Regions by Their Skeletons

JOHN L. PFALTZ AND AZRIEL ROSENFELD
University of Maryland, College Park, Maryland

Any region can be regarded as a union of maximal neighborhoods of its points, and can be specified by the centers and radii of these neighborhoods; this set is a sort of "skeleton" of the region. The storage required to represent a region in this way is comparable to that required when it is represented by encoding its boundary. Moreover, the skeleton representation seems to have advantages when it is necessary to determine repeatedly whether points are inside or outside the region, or to perform set-theoretic operations on regions.

1. Introduction

There are many pictorial data processing problems which require the encoding and processing of irregularly shaped planar regions. In general it is impractical to represent such regions by explicitly enumerating their points, since the required storage capacity would be prohibitively large (though not infinite, since a digitized region contains only finitely many "points"). Instead, regions are usually described by encoding their boundaries. A boundary can be approximated piecewise by analytically simple curves, as in Sketchpad and its successors [1–3]. Alternatively, it can be approximated by a chain of segments taken from a fixed grid, as in the work of Freeman [4–5].

In this paper, an alternative approach to representing an arbitrary planar region is described. The given region is described as a union of "maximal neighborhoods" of a certain "skeleton" set of its interior points. It is shown that this approach is comparable to chain encoding in storage requirements. At the same time, it can have significant advantages for certain types of region processing problems, such as those in which it must be determined whether or not a given point is inside a given region, or in which the intersection of two or more regions must be found.

2. Maximal Neighborhoods and Skeletons

A digitized image is usually given in the form of a rectangular matrix of elements (a_{ij}) in which (i, j) are the Cartesian coordinates of a "point" and a_{ij} is the density of the digitized image at the point (i.e., the average density of the original image over the small region represented by the "point"). Other digitized image configurations are possible, for example that using a hexagonal rather than rectangular grid, which in fact seems to be preferable for some applications. However in what follows it will be assumed for simplicity that the given digital picture is in rectangular matrix form.

In order to define the concept of a maximal neighborhood, one must specify a metric on the picture matrix. Let $P_1 = a_{i_1,j_1}$ and $P_2 = a_{i_2,j_2}$ be two matrix elements (from now on: "points"), and define $d(P_1, P_2) = |i_1 - i_2| + |j_1 - j_2|$. It is easily verified that this function has the standard properties of a metric or "distance," namely

$$d(P_1, P_2) \geqq 0, \quad \text{and} = 0 \quad \text{if and only if} \quad P_1 = P_2 \qquad (1)$$

$$d(P_1, P_2) = d(P_2, P_1) \qquad (2)$$

$$d(P_1, P_3) \leqq d(P_1, P_2) + d(P_2, P_3) \qquad (3)$$

for all points P_1, P_2, P_3.

If r is a non-negative integer, the neighborhood of $P_0 = a_{i_0,j_0}$ of radius r is defined as the set of all $P = a_{ij}$ such that $d(P, P_0) \leqq r$. Evidently, this neighborhood is just the square array of points centered at P_0, oriented diagonally and with side $r + 1$ points long, as shown in Figure 1. If $r = 0$, the neighborhood reduces to P_0 itself.[1]

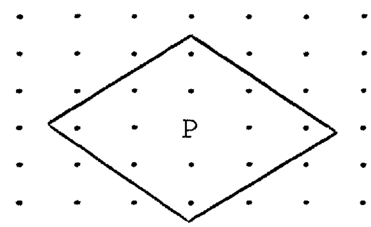

FIG. 1. Neighborhood of the point P with radius 2.

Let R be a region within the picture matrix M—in other words, R can be any subset of M. Let P_0 by any point of R. Some neighborhood of P_0 must always be contained in R, e.g., the neighborhood of radius 0. Let $\mathfrak{N}_R$ be the set of all neighborhoods (of points of R) which are contained in R. Since any point of R is contained in at least one of these neighborhoods, their union is all of R, i.e., $R = \bigcup_{N \in \mathfrak{N}_R} N$.

A neighborhood in $\mathfrak{N}_R$ will be called *maximal* if it is not (properly) contained in any other such neighborhood. Some examples of maximal neighborhoods are shown in Figure 2. Since $\mathfrak{N}_R$ is finite, any $N \in \mathfrak{N}_R$ is contained in at least one maximal neighborhood. Let $\mathfrak{M}_R \subset \mathfrak{N}_R$ be the set of maximal neighborhoods; thus $R = \bigcup_{N \in \mathfrak{M}_R} N$.

Any neighborhood is defined by specifying its center and radius. Since R is a union of maximal neighborhoods, it can thus be completely described by giving the centers and radii of these neighborhoods. This is the method of region representation which will be studied in the remainder of this paper.

The concept of representing a region by its set of maximal neighborhoods has recently been proposed and studied by Blum [6]. Since the locus of centers of maximal neighborhoods often takes the form of a centrally located stick

[1] It should be noted that other metrics could be defined on a picture matrix, which would give rise to other neighborhood systems. For example, one could define $d'(P_1, P_2) = \max(|i_1 - i_2|, |j_1 - j_2|)$ and verify that it too is a metric. For d', the neighborhood of P_0 of radius r is readily the square array of points centered at P_0, oriented horizontally and vertically, and with side $2r + 1$ points long. However, the metric d defined above seems to be the simplest for most computational purposes.

figure, the name "skeleton" has been suggested for it. We can thus speak of R as being determined by specifying its skeleton together with the maximal neighborhood radius associated with each skeleton point.

Algorithms for determining the skeleton points and their associated radii for any region in a digital picture, given the boundary of the region, are described in [7]. Figure 3 shows the skeletons of a number of different regions; each skeleton point is labeled with its radius reduced modulo 10. Other algorithms, also described in [7], will regenerate the region from the skeleton. These algorithms produce the region boundary as a distinguished point set, but not as a linearly ordered chain.

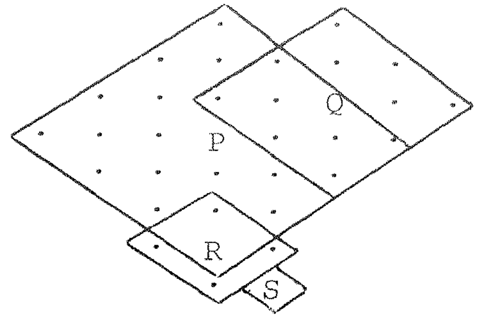

FIG. 2. A region defined by maximal neighborhoods of the points P, Q, R, S, with radii 3, 2, 1, and 0.

FIG. 3. Skeletons of three irregular regions (bounded by X's).

3. Comparison of Storage Requirements

The amounts of storage required by the skeleton and boundary techniques of region encoding will now be compared. Let R be a region on a digital picture; since the picture is discrete, the boundary of R is a polygon. Four methods of encoding R can be considered:

(a) The boundary of R is specified as an ordered sequence of straight line segments of given lengths and slopes.

(b) The same as (a), but allowing only slopes in the eight principal directions (horizontal, vertical or diagonal)

(c) The same as (b), but allowing only segments of length 1.

(d) R is specified by the set of its skeleton points and their associated radii.

Note that method (a) requires a possibly large number of bits to specify both the length and slope of each boundary segment, while methods (b)–(c) require only three bits to encode slope, and method (c) requires no encoding of length. However, methods (b)–(c) in general require successively greater numbers of boundary segments to specify R. Method (b) is used for purposes of comparison in the example which follows.

To compare the relative amounts of storage required by these methods in practical situations, an outline map of southeast Asia (Figure 4) was manually digitized on a 200×250 grid. Table I gives the number of straight line boundary segments [as in method (b)] and the number of skeleton points for each country on this map. Note that except for China and Burma, which have long straight lines as major parts of their boundaries, there are always somewhat fewer skeleton points than boundary segments. (Note also that a skeleton point radius cannot exceed half the diameter of the picture, while a straight boundary segment can be as long as the picture diameter; thus an arbitrary radius can be specified using one bit less than required to specify the length of a boundary segment. However, if long straight boundary segments are very rare, appreciable savings can be achieved by using Shannon-Fano techniques to encode boundary segment lengths.) On the other hand, specifying the slope of a boundary segment in method (b) requires only three bits,

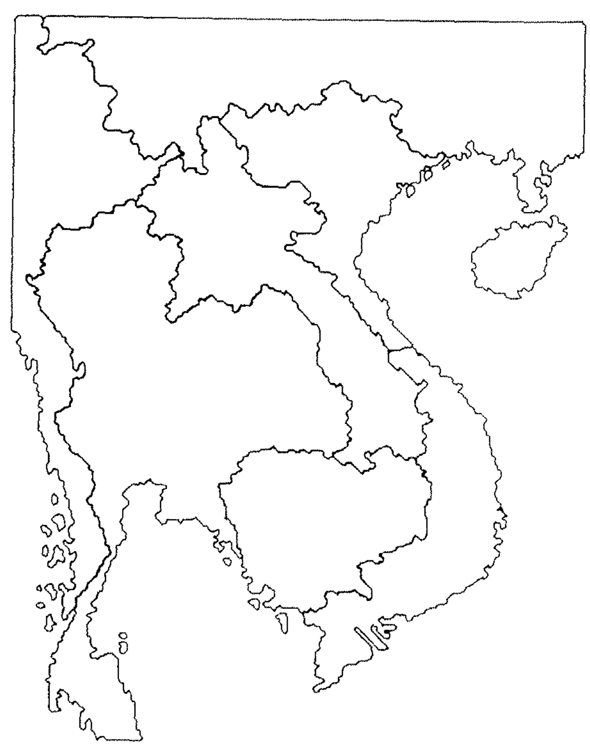

FIG. 4. Outline map of Southeast Aisa

TABLE I. Boundary Segments and Skeleton Points for Figure 4

Region	*Number of straight line boundary segments*	*Number of skeleton points*
Burma	338	371
Cambodia	178	135
China	231	407
Laos	286	272
Thailand	440	402
N. Viet Nam	197	183
S. Viet Nam	262	244

while specifying the position of a skeleton point requires a number of bits which depends on the picture diameter (in the case of Figure 4, roughly 16 bits).

It can thus be concluded that the skeleton method of encoding a region requires somewhat more storage than do the boundary methods, as exemplified by method (b). However, if the picture is not too large, the storage required is of the same order of magnitude. In particular, if the region has many connected components or is multiply connected, the skeleton representation may actually be more economical. Note, in fact, that to represent the boundary of such a region, special coding schemes may be needed to link the disconnected parts of the boundary, whereas the skeleton method can be used without modification. In any case, for certain applications, the additional storage requirement may be more than offset by gains in processing speed.

4. Comparison of Processing Requirements: Shading

The standard method of determining whether a point lies inside or outside a region, given the boundary of the region, is to draw a straight line from the point to the border of the picture and count the number of times it intersects the boundary. If this number is odd, the point is inside; if even, outside—provided that the line is never tangent to the boundary. To carry out this procedure, each segment of the boundary must be compared with the line in order to determine whether or not they intersect.

If a region is given by specifying its skeleton, the procedure for determining if a given point P lies inside it is analogous. The coordinates i, j of the point must be compared with the coordinates i_k, j_k and radius r_k of each skeleton point P_k. The point lies inside the region if and only if $| i - i_k | + | j - j_k | \leqq r_k$ for some k. Note that in general it is necessary to make all of these comparisons only for points outside the region; if a point is inside, that fact is established as soon as the first P_k satisfying the above relation is found.

Shortcuts can be devised to reduce the number of comparisons actually required in both the boundary and skeleton cases, by using special methods of indexing the boundary segments or skeleton points. (For the boundary case see, e.g., [8].) One can, for example, (a) enclose the region in a set of rectangles, and determine analytically which rectangle(s) contain the skeleton points to be searched; (b) use a sweeping line technique which limits search to those skeleton points whose maximal neighborhoods intersect the line; (c) arrange the skeleton points in various types of lexicographic order. However, it is not difficult to exhibit cases in which these shortcuts fail to yield significant savings.

It can be concluded from the foregoing that since there are typically fewer skeleton points than boundary segments, and the comparison operations required in the skeleton case are considerably simpler, the skeleton representation has significant advantages if it is necessary to repeatedly determine whether points are inside or outside the given region.

A specific application which does require many such determinations is that of shading a region, for example with parallel straight lines. If the region is given in boundary form, the method described in the first paragraph of this section can be used to determine, for any given line, the segments of it which lie inside the region. Repeating this process for other lines parallel to the given line will systematically generate the desired set of shading segments. Note, however, that virtually the entire process must be repeated for each line.

An algorithm for parallel line shading of a region given in skeleton form can proceed as follows: For any one line L, the distance d_k and direction θ_k from any point on L to each skeleton point P_k is first determined. These distances and directions can then be computed very easily for the other points on L, and for points on lines parallel to L, by systematically incrementing the d_k and θ_k appropriately. (Similar algorithms can be devised for shading a region with any of a wide variety of other regular textures.)

A Fortran routine has been written which outputs a shaded drawing of any region which has been stored in skeleton form. Specifically, this routine shades the specified region with straight lines of any orientation and density. Examples of the output of this routine for the map of Figure 4 are shown as Figures 5 and 6. These shaded maps were drawn by a Calcomp Model 565 Digital Incremental Plotter.

Figures 5–6 also show region boundaries which were generated from the stored skeletons. This was done by the following procedure: A point known to be inside the region (e.g., a skeleton point) is picked, and a straight line is drawn from it until a point is found which is no longer inside the region; this point must be on the boundary. With the direction of the straight line as a reference, the neighboring points are examined in a clockwise sequence until another boundary point is found. Repeating this process will systematically generate the successive boundary points, thus providing directly a chain-encoded representation of the boundary.

5. Set-Theoretic Operations on Regions

A frequently encountered problem in computer process-

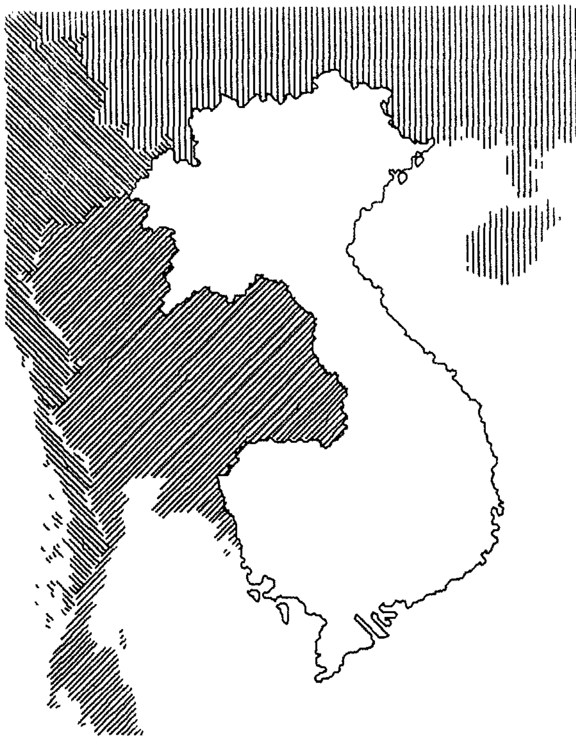

FIG. 5. Shaded map produced from Figure 4, with four regions combined.

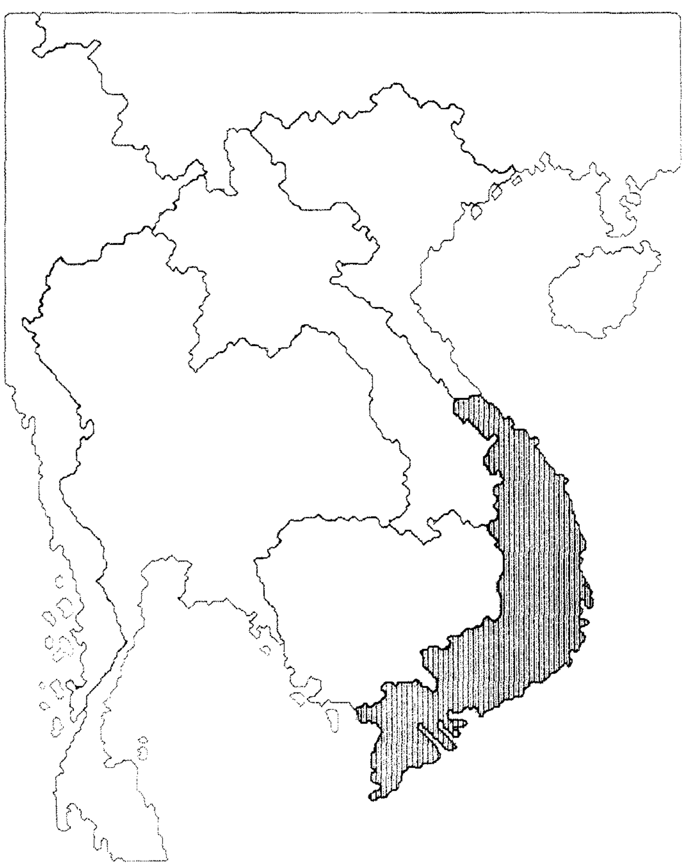

FIG. 6. Shaded map produced from Figure 4, rescaled

ing of map data is that of determining the intersection of two given regions—or, more generally, determining any given set-theoretic composite of a given collection of regions. If the regions are represented analytically, as in Sketchpad and its descendants, this type of manipulation can be carried out in the analytical domain. However, as the regions become complex, their analytical representation becomes uneconomical.

The following is an algorithm for directly determining the skeleton representation of the intersection $A \cap B$ of two regions, given the skeleton representations of A and of B.

Let P_i be a skeleton point for A, and r_i the corresponding radius; let Q_j, s_j be defined analogously for B. If P is a point in $A \cap B$, then there exist i and j such that

$$d(P, P_i) \leqq r_i; \qquad d(P, Q_j) \leqq s_j.$$

For any point P, let r be the largest integer such that

$$d(P, P_i) + r \leqq r_i \quad \text{for some } i;$$

similarly, let s be the largest integer such that

$$d(P, Q_j) + s \leqq s_j \quad \text{for some } j.$$

Let t be the smaller of r and s. If $t < 0$, P is not in $A \cap B$; while if $t \geqq 0$, P is in $A \cap B$, and the neighborhood of P of radius t is the largest neighborhood of P contained in $A \cap B$.

Order the set of such (P, t) in descending order of their t's. If t is maximal, P must be a skeleton point. If t is not maximal, P is a skeleton point if and only if it has no horizontal or vertical neighbor P' such that the corresponding t' is greater than t. (The number of operations required to implement this algorithm can be reduced by an order of magnitude by applying various shortcuts, as indicated in Section 4.)

Another algorithm can be used to obtain the skeleton of the set-theoretic difference of A and B. With notation as above, define

$$\delta(P, A) = \min_i [d(P, P_i) - r_i].$$

It is easily seen that if $\delta(Q_j, A) > s_j$, then Q_j is a skeleton point of B–A; whereas if $\delta(Q_j, A) \leqq s_j$, the neighborhood of Q_j of radius s_j is not contained in B–A. Moreover, if P is a skeleton point of B–A which is not a skeleton point of A, its associated radius must be $\delta(P, A) - 1$, and its neighborhood of this radius must be contained in the maximal neighborhood of some skeleton point of B which is not a skeleton point of B–A. These necessary conditions are not sufficient; however, the set of points which they define can be reduced to the true skeleton of B–A by the procedure described in the preceding paragraph.

The union of the skeletons of A and B gives a skeleton-type representation for $A \cup B$. This set is not the skeleton of $A \cup B$ unless A and B are disjoint; however, it still completely defines $A \cup B$, even if somewhat redundantly,

(*Continued on page 125*)

For tape-oriented systems, the READ and PRINT statements were changed to READ INPUT TAPE or READ (i, j) and WRITE OUTPUT TAPE or WRITE (i, j). (2) For some systems, CALL EXIT was changed to STOP. (3) For some systems, the F was dropped from library function names.

A translation program has been written to translate the source decks from FORTRAN II-D to FORTRAN IV.

Acknowledgments. The authors are indebted to Dr. Carl F. Kossack for advice concerning the programs and to the Referee for advice on this paper.

The name Stat-Pack is a short name for Biostatistical Programming Package and grew out of usage and the fact that all the programs are stored on an IBM 1316 disk pack.

Since the paper was written, several other similar packages with the same name have been called to the attention of the authors, namely, one at the University of California at Berkeley and one distributed by the Univac Division of Sperry Rand Corp.

RECEIVED MAY, 1966; REVISED SEPTEMBER, 1966

REFERENCES

1. DIXON, W. J. Biomedical computer programs. Health Sci. Comput. Facility, U. of California, Los Angeles, 1964.
2. STERLING, T., AND POLLACK, S. Medcomp—Statistical Systems. Medical Comput. Center, U. of Cincinnati, Ohio 1963.
3. BUHLER, R. P-Stat, a system of statistical Programs for the 7090/7094. Computer Center, Princeton U., Princeton, N.J., 1964.
4. SHANNON, S. Stat-Pack, biostatistical programming package. Goddard Computer Science Lab., Wadley Research Institute, Dallas, Texas, 1966.

PFALTZ AND ROSENFELD—*cont'd from p. 122*

and so can still be used to represent $A \cup B$. In Figure 5, this method was used to determine the union of Cambodia, Laos, and North and South Viet Nam.

RECEIVED APRIL, 1966

REFERENCES

1. SUTHERLAND, I. E. Sketchpad: A man-machine graphical communication system. Proc. AFIPS 1963 Spring Joint Comput. Conf., Vol. 23, pp. 324–346.
2. JOHNSON, T. E. Sketchpad III: A computer program for drawing in three dimensions. Proc. AFIPS 1963 Spring Joint Comput. Conf., Vol. 23, pp. 347–353.
3. ROBERTS, L. G. Machine perception of three-dimensional solids. *Optical and Electro-Optical Information Processing.* M.I.T. Press, Cambridge, Mass., 1965, pp. 159–197.
4. FREEMAN, H. On the encoding of arbitrary geometric configurations. *IEEE Trans. EC-10* (June 1961), 260–268.
5. FREEMAN, H., AND GARDER, L. Apictorial jigsaw puzzles, the computer solution of a problem in pattern recognition. *IEEE Trans. EC-13* (April 1964), 118–127.
6. BLUM, H. A transformation for extracting new descriptors of shape. *Proceedings, Symposium on Models for the Perception of Speech and Visual Form,* AFCRL, Boston, November 1964, in press.
7. ROSENFELD, A., AND PFALTZ, J. L. Sequential operations in digital picture processing. *J. ACM 13*, 4 (Oct. 1966), 471–494.
8. LOOMIS, R. G. Boundary networks. *Comm. ACM 8* (Jan. 1965), 44–48.