

# Morphological Skeleton Representation and Coding of Binary Images

PETROS A. MARAGOS, MEMBER, IEEE, AND RONALD W. SCHAFFER, FELLOW, IEEE

**Abstract**—This paper presents the results of a study on the use of morphological set operations to represent and encode a discrete binary image by parts of its skeleton, a thinned version of the image containing complete information about its shape and size. Using morphological erosions and openings, a finite image can be uniquely decomposed into a finite number of skeleton subsets and then the image can be exactly reconstructed by dilating the skeleton subsets. The morphological skeleton is shown to unify many previous approaches to skeletonization, and some of its theoretical properties are investigated. Fast algorithms that reduce the original quadratic complexity to linear are developed for skeleton decomposition and reconstruction. Partial reconstructions of the image are quantified through the omission of subsets of skeleton points. The concepts of a globally and locally minimal skeleton are introduced and fast algorithms are developed for obtaining minimal skeletons.

For images containing blobs and large areas, the skeleton subsets are much thinner than the original image. Therefore, encoding of the skeleton information results in lower information rates than optimum block-Huffman or optimum runlength-Huffman coding of the original image. The highest level of image compression was obtained by using Elias coding of the skeleton.

## I. INTRODUCTION

**A**UTOREGRESSIVE modeling and orthogonal transforms such as Fourier or Karhunen-Loève transforms have provided the theoretical basis for most of the research in digital image coding during the past decade. Both of these approaches exploit primarily the algebraic structure of signals. However, in the case of binary image signals, which are mainly perceived as geometrical patterns, there is a need for representations that emphasize geometric rather than algebraic structure. One such geometric representation is the *skeleton*. In general, the term skeleton has been used to describe a line-thinned caricature of the binary image which summarizes its shape and conveys information about its size, orientation, and connectivity. The skeleton has already been applied in biological shape description [1]–[3], pattern recognition [3], [9], [25], image coding [7], [18], [22], quantitative metallography [13], [14], and automated industrial inspection [15].

The skeleton of a continuous binary image was first in-

roduced by Blum [1], who originally called it the “*medial axis*” and later the “*symmetric axis*” [2]. Blum’s initial procedure for obtaining the medial axis was to set up “*grassfires*” at time  $t = 0$  along all the points of the object boundary, and to let these grassfires propagate as wavefronts toward the center of the object at uniform speed following Huygen’s principle. The medial axis points, where these wavefronts would intersect and extinguish, together with their arrival times defined the “*medial (symmetric) axis function*.” A very important property of this symmetric axis function is the ability to reconstruct the object boundary by propagating the wavefronts backward. Subsequently, a number of people developed a mathematical theory for the skeleton: Kotelly [12] and Calabi [4] (with Hartnett [5] later) for continuous images, and Rosenfeld and Pfaltz [27], [24], Mott-Smith [21], and Montanari [20] for discrete images. Influenced by all the above contributions and many others referenced in his epitomizing work [2], Blum considered two new approaches to find the medial axis. First, he used the “*symmetric point distance*” from a skeleton point to the boundary. Second, he showed that the symmetric axis is the locus of the centers of the “*maximal disks*” inscribable inside a filled-in image object. Blum’s second interpretation motivated Frank *et al.* [7] to use the concepts of a point and a *growth* in a progressive binary image transmission scheme.

Parallel to and independently from the evolution of all the above skeletonization ideas, *mathematical morphology* evolved as a set-theoretical method for image analysis whose purpose is the quantitative description of geometrical structures. Mathematical morphology, after its first introduction by Matheron [19] and Serra [28] in 1964, has found numerous applications [28], [32]. Some contributions to the morphology of graytone functions were also made by Sternberg [31], [32]. One important feature of mathematical morphology is the fact that it unifies the realization of many linear and nonlinear translation-invariant systems [16], [17]. As a particular application of this unifying power, Lantuejoul [13], [14] proved that the skeleton can be obtained by morphological set transformations. This important result provides the theory that is presented in this paper. For this reason, we refer to the skeleton as a *morphological skeleton* to distinguish it from the skeletons obtained using other approaches.

In this paper, which is the continuation of [18], we are

Manuscript received June 22, 1985; revised January 14, 1986. This work was done at the Georgia Institute of Technology, and was supported by the Joint Services Electronics Program under Contract DAAG-84-K-0024.

P. A. Maragos is with the Division of Applied Sciences, Harvard University, Cambridge, MA 02138.

R. W. Schaffer is with the School of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

IEEE Log Number 8609629.

concerned only with the use of morphological skeletons for image representation and coding. Although Serra's monograph [28] is an excellent treatment of mathematical morphology, for reasons of clarity in this paper we summarize (in Section II) the basic morphological set transformations that are necessary to understand the morphological skeleton. We also explore issues concerned with the practical implementation of skeleton transformations. In Section III, we present some new results on the morphological skeleton representation of binary images together with fast implementation algorithms for skeletal decomposition and image reconstruction, we investigate theoretical properties of the skeleton, its partial reconstructions, and fast searching algorithms for obtaining minimal skeletons, and we comment briefly on some similarities and differences between our work and related previous work. The use of the skeleton in image coding is described in Section IV. Finally, in Section V, we summarize the results of our research.

*Notation*

$R$  = set of real numbers;  $Z$  = set of all integers;  $E$  = Euclidean space  $R^2$  or  $Z^2$ .

$X, Y, A, B, C, \dots$  = subsets of  $E$ ;  $\emptyset$  = empty set.

$X^c$  = set complement of  $X$  with respect to  $E$ .

$x, y, z, a, b, c, \dots$  = elements or vector points of  $E$ .

$\{x:P\}$  = set of points  $x$  satisfying a property  $P$ .

$A \subseteq B$  = set  $A$  is a subset of  $B$ .

$U(\cap)$  = set union (intersection).

$A - B$  = set difference between  $A$  and  $B$ .

II. CONCEPTS FROM MATHEMATICAL MORPHOLOGY

A. Basic Morphological Set Transformations

Mathematical morphology extracts information about the geometrical structure of an *image object* by transforming it through its interaction with another object, called the *structuring element*, which is of simpler shape and size than the original image object. Information about size, spatial distribution, shape, connectivity, convexity, smoothness, and orientation can be obtained by transforming the image object using different structuring elements.

Mathematical morphology represents two-dimensional image objects as mathematical sets in a Euclidean space  $E$ , which can be either the continuous space  $R^2$  or the discrete space  $Z^2$ . For example, a single binary sampled image is viewed as a subset of  $Z^2$ , i.e., as a set of integer pairs. These integer pairs can be viewed as the coordinates with respect to two basis unit vectors whose length equals the sampling period in each direction. This representation of sampled binary images by subsets of  $Z^2$  is suitable both for rectangularly and hexagonally sampled images depending on whether the angle between the basis vectors is  $90^\circ$  or  $60^\circ$ , respectively. The concept, however, of a set is more general than needed to represent an image. According to Matheron's approach [19], each image object is assumed to contain its boundary and, thus, can be represented by a *closed* subset of  $E$ . In addition,

every structuring element is represented by a *compact* subset of  $E$  so that the morphological transformations are uppersemicontinuous transformations.

It is appropriate to restrict the class of morphological set transformations by imposing constraints which are consistent with goals such as efficient image representation and automatic extraction of information from the image. Therefore, a morphological transformation of an image object is said to be *quantitative* only if it satisfies four quantification constraints, which correspond to the four basic *principles* of the theory of mathematical morphology: 1) invariance under translation, 2) compatibility with change of scale, 3) local knowledge, and 4) uppersemicontinuity [28].

The simplest quantitative morphological set transformations are erosion, dilation, opening, and closing. These transformations are based on the Minkowski set addition and subtraction [8]. The *Minkowski set addition*  $A \oplus B$  of two sets  $A$  and  $B$  consists of all points  $c$  which can be expressed as an algebraic vector addition  $c = a + b$ , where the vectors  $a$  and  $b$  belong to the sets  $A$  and  $B$ , respectively. If  $A_b = \{a + b : a \in A\}$  denotes the *translate* of the set  $A$  by the vector  $b$ , then the Minkowski sum of  $A$  and  $B$  is equal to the set union of all the translates  $A_b$  of  $A$  when the vector  $b$  sweeps the set  $B$ :

$$A \oplus B = \{a + b : a \in A, b \in B\} = \bigcup_{b \in B} A_b. \quad (1)$$

The *Minkowski set subtraction* of  $B$  from  $A$ , denoted as  $A \ominus B$ , is the operation dual to Minkowski set addition with respect to complementation:

$$A \ominus B = (A^c \oplus B)^c = \bigcap_{b \in B} A_b. \quad (2)$$

Morphological erosion and dilation are defined from a geometric point of view as set transformations that shrink or expand a set. Algebraically, however, they are actually Minkowski set subtraction and addition, respectively. Let the closed set  $X$  represent a binary image and the compact set  $B$  a structuring element. The complement  $X^c$  of  $X$  represents the image background. We denote by  $B^s$  the *symmetric* set of  $B$  with respect to origin; i.e.,  $B^s$  is obtained by rotating  $B$   $180^\circ$  on the plane so that  $B^s = \{-b : b \in B\}$ . Then, the *erosion* of  $X$  by  $B$  is defined geometrically as the set of those points  $z$  such that the translate  $B_z$  is contained in the original image set  $X$ . Algebraically, the erosion of  $X$  by  $B$  is equal to the Minkowski set subtraction of  $B^s$  from  $X$ :

$$X \ominus B^s = \{z : B_z \subseteq X\} = \bigcap_{b \in B} X_{-b}. \quad (3)$$

Fig. 1 shows that erosion shrinks the object. The dual operation of erosion with respect to complementation is the dilation. *Dilation* of  $X$  by  $B$  is defined geometrically as the set of all those points  $z$  such that the translate  $B_z$  intersects  $X$ . Algebraically, the dilation of  $X$  by  $B$  is equal to the Minkowski sum of  $X$  and  $B^s$ :

$$X \oplus B^s = \{z : B_z \cap X \neq \emptyset\} = \bigcup_{b \in B} X_{-b}. \quad (4)$$

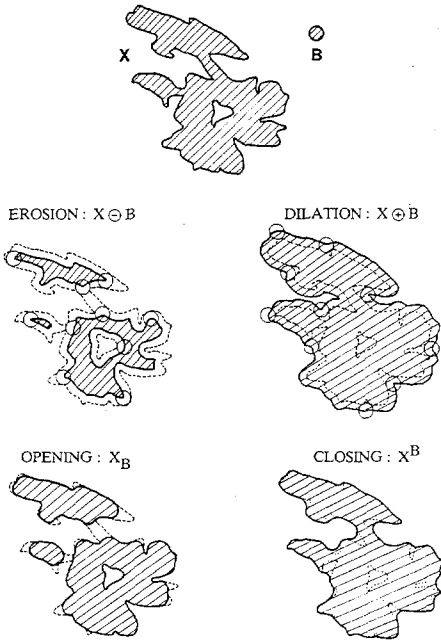


Fig. 1. Erosion, dilation, opening, and closing of  $X$  by  $B$  (the shaded areas correspond to the interior of the objects, the dark solid curve to the boundary of the transformed object, and the dashed curve to the boundary of the original object).

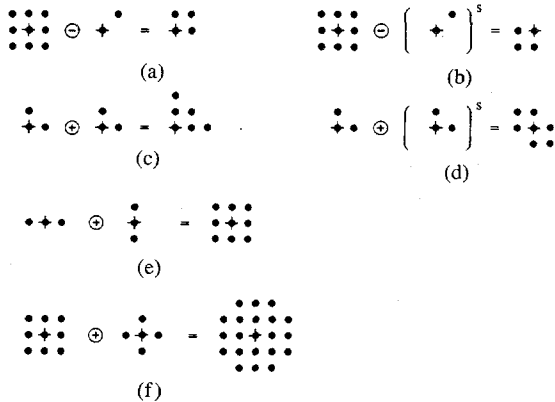


Fig. 2. Shrinking and expanding discrete sets. (a) Minkowski subtraction, (b) erosion, (c) Minkowski addition, (d) dilation, (e) and (f) illustrations of forming larger structuring elements as the Minkowski sum of two simpler sets. ( $\bullet$  = object points,  $+$  = origin).

Fig. 1 shows that dilation expands the object. Dilating the object is equivalent to eroding its background, as implied by (2). Both erosion and dilation are nonlinear operations which are generally noninvertible, and only dilation is *commutative* and *associative*. Fig. 2(a)–(d) shows examples of dilations and erosions of discrete sets. It also illustrates the difference between Minkowski addition–subtraction and dilation–erosion, respectively. Of course, there is no difference between these two pairs of operations if  $B$  is a symmetric structuring element. Therefore, whenever there is no risk of confusion, we will refer to  $X \ominus B$  or  $X \oplus B$  as simply the erosion or dilation of  $X$  by  $B$ . We next comment briefly on some selected properties of erosion and dilation.

1) *Translation Invariance*: For any vector  $z$  in  $E$ , we have  $X_z \oplus B = (X \oplus B)_z$ . In addition, erosion or dilation

by a single point is just a translation; i.e.,  $X \ominus \{b\} = X \oplus \{b\} = X_b$ .

2) *Erosion and Dilation of  $X$  by  $B$  are Increasing Transformations with Respect to  $X$* :

$$X_1 \subseteq X_2 \Rightarrow X_1 \oplus B \subseteq X_2 \oplus B. \quad (5)$$

Erosion of  $X$  by  $B$  is decreasing with respect to  $B$ ; i.e.,  $B_1 \subseteq B_2 \Rightarrow X \ominus B_2 \subseteq X \ominus B_1$ . From the above properties, if  $B$  contains the origin, then erosion is an *antiextensive* transformation, whereas the dilation is *extensive*; i.e.,  $X \ominus B \subseteq X \subseteq X \oplus B$ .

3) *Parallel Composition*: The operation of dilation distributes over set union, whereas erosion distributes over set intersection:

$$(X \cup Y) \oplus B = (X \oplus B) \cup (Y \oplus B) \quad (6a)$$

$$(X \cap Y) \ominus B = (X \ominus B) \cap (Y \ominus B) \quad (6b)$$

$$X \ominus (A \cup B) = (X \ominus A) \cap (X \ominus B). \quad (6c)$$

4) *Serial Composition*: Successively eroding (respectively, dilating) a set  $X$  first by  $A$  and then by  $B$  is equivalent to eroding (respectively, dilating)  $X$  by their Minkowski sum  $A \oplus B$ :

$$(X \oplus A) \oplus B = X \oplus (A \oplus B) \quad (7a)$$

$$(X \ominus A) \ominus B = X \ominus (A \oplus B). \quad (7b)$$

5) *Local Knowledge*: Let  $M$  be a bounded analysis mask or frame and  $X$  an image object which may exceed the mask  $M$ . Inside the mask  $M$  we can know without error only the masked set  $X \cap M$  and its transformed versions. However, we can obtain erosions or dilations of the original unmasked set  $X$  by a structuring element  $B$  without error inside a new mask  $M^* = M \ominus B^s$ .

Another pair of dual morphological transformations are the opening and the closing. If we erode  $X$  by  $B$  and then dilate the eroded set  $X \ominus B^s$  by  $B^s$ , we generally do not recover  $X$ , but rather a simplified and less detailed version of  $X$ . This new set is called the *opening*,  $X_B$ , of  $X$  by  $B$ . By duality, the *closing*,  $X^B$ , of  $X$  by  $B$  comes from dilating first and then eroding. Thus, the opening and closing are defined, respectively, as

$$X_B = (X \ominus B^s) \oplus B \quad (8a)$$

$$X^B = (X \oplus B^s) \ominus B. \quad (8b)$$

Fig. 1 illustrates that the opening transformation suppresses the sharp capes and cuts the narrow isthmuses of the object, whereas the closing transformation fills up the thin gulfs and small holes. Thus, if the structuring element  $B$  has a regular shape, both opening and closing can be thought of as being nonlinear filters which smooth the contours of the object. Both transformations are generally noninvertible. The opening is always antiextensive, whereas the closing is always extensive; i.e.,  $X_B \subseteq X \subseteq X^B$ . Both opening and closing are increasing and translation-invariant transformations. Moreover, they are *idempotent*; i.e.,  $(X_B)_B = X_B$ .

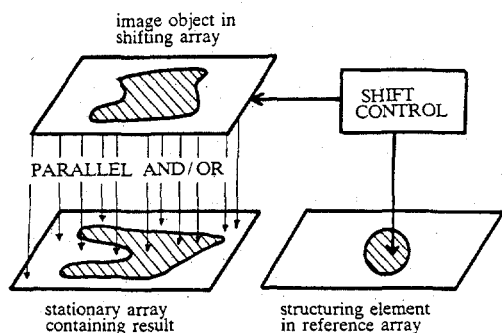


Fig. 3. (Adapted from Sternberg [31].) Parallel implementation of erosion and dilation.

The above operations may appear superficially simple, but an enormous variety [28] of image processing tasks can be performed by combining erosions, dilations, openings, and closings. A particular case is the morphological skeleton transformation of a binary image.

*B. Computational Complexity of Morphological Set Transformations*

To implement the basic morphological set operations on a conventional computer, one could simply represent the sets by binary functions, whose values are equal to one and zero at points of the object and its background, respectively, and take the local minimum and maximum inside a running window (set)  $B$  for the erosion and dilation by  $B$ , respectively. This local minimum or maximum will shrink or expand, respectively, the ones of this binary function, as investigated in [23]. However, (3) and (4) give us simple ways to practically implement erosions and dilations using just Boolean set logic. That is, erosion and dilation of  $X$  by  $B$  are algebraically interpreted as the intersection and union, respectively, of all the translates  $X_{-b}$  of  $X$  when  $b$  sweeps  $B$ . The structuring element  $B$  is a compact set, which for discrete sets means that  $B$  contains a finite number of points. Thus, the number of required translates of  $X$  is finite. The outline of a practical implementation for erosion and dilation is shown in Fig. 3. We need two bit planes for the image  $X$  and the structuring element  $B$  and a third accumulation plane for the resulting transformed image. The image plane is shifted in parallel to the accumulator plane, and the amount of shifting is controlled by the points belonging to the structuring element  $B$ . The accumulator plane holds the parallel logical AND or OR of all the shifted versions of the image plane, and after all the points of  $B$  have been spanned, it will contain the erosion or dilation, respectively, of the original image. The above implementation exploits the desired parallelism in image processing tasks [31] and is very well suited for cellular or bit-plane computers.

In addition, the parallel and serial composition properties of erosion and dilation can be exploited to speed up the practical implementation of these operations. That is, the parallel composition laws of (6) allow the formation of more complex structuring elements by taking the union or intersection of simpler elements. The serial composition laws (see (7) and Fig. 2) enable us to erode and dilate by two-dimensional (2-D) structuring elements using only

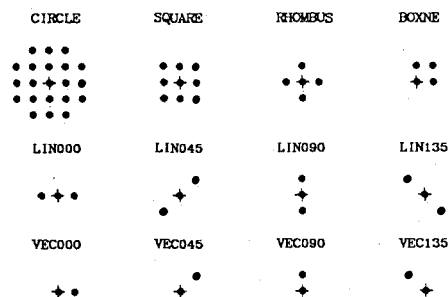


Fig. 4. Some structuring elements on the rectangular grid (● = object points, + = origin).

one-dimensional (1-D) elements. For example, the  $3 \times 3$  pixels square is the dilation of one horizontal by one vertical line segment of 3 pixels [see Fig. 2(e)]. Thus, the shape and the size of the structuring element can be programmed. Fig. 4 shows some examples of discrete symmetric and asymmetric structuring elements on the rectangular grid. The circular (21 pixels) element CIRCLE results from Minkowski sum of the square (9 pixels) element SQUARE and the rhomboid (5 pixels) element RHOMBUS as Fig. 2(f) illustrates. Similarly, the 4-pixels square element BOXNE in the NE quadrant is the Minkowski sum of the 2-pixels vectors VEC000 and VEC090. In Table I we compare the computational complexity of three different methods of implementing the erosion-dilation of an object  $X$  by a structuring element  $nB$  of size  $n$  obtained by the  $n$ -fold Minkowski sum of  $B$  with itself. Method 1 implements the operation  $X \ominus nB$  by directly using (3) and (4). Method 2 exploits the serial composition laws of (7). For example, suppose that we want to erode  $X$  by  $2B = B \oplus B$ . Then, from (7), we have

$$X \ominus 2B = X \ominus (B \oplus B) = (X \ominus B) \ominus B.$$

Thus, method 1 erodes  $X$  by  $2B$ , whereas method 2 erodes  $X$  first by  $B$  and then it erodes the result by  $B$ . Of course, both methods give the same result. Method 3 combines method 2 and further exploits (7) if  $B$  is decomposable into simpler elements. For example, let  $B$  be the SQUARE of Fig. 4; then  $B = B_1 \oplus B_2$ , where  $B_1$  and  $B_2$  are, respectively, the horizontal and vertical structuring elements LIN000 and LIN090 of Fig. 4. Thus, method 3 implements the erosion of  $X$  by  $2B$  as follows:

$$X \ominus 2B = (((X \ominus B_1) \ominus B_2) \ominus B_1) \ominus B_2.$$

Table I (compare methods 2 and 3 for size  $n = 1$ ) shows the computational savings in parallel AND/OR's of the whole image which result when we do erosions or dilations by some of the composite structuring elements of Fig. 4 by using their decomposition in simpler elements, as explained above.

III. MORPHOLOGICAL SKELETON IMAGE REPRESENTATION

*A. Background on Morphological Skeleton and Skeleton Function*

The skeleton  $SK(X)$  of a continuous image object  $X$ , viewed as a subset of  $R^2$ , is defined as the set of the cen-

TABLE I  
NUMBER OF PARALLEL AND/OR'S OF THE IMAGE  $X$  TO IMPLEMENT ITS EROSION-DILATION BY  $nB$ .  
( $n \geq 1$  AND  $B$  IS ONE OF THE STRUCTURING ELEMENTS OF FIG. 4)

Structuring Element $B$	Method 1 ( $X \oplus nB$ )	Method 2 ( $X \oplus B$ ) $\oplus B \cdots \oplus B$	Method 3 (Method 2 with $B = B_1 \oplus B_2$ )	Savings (percent) of Method 3 over 2
CIRCLE	$14n^2 + 6n$	$20n$	$8n$	60
SQUARE	$4n^2 + 4n$	$8n$	$4n$	50
RHOMBUS	$2n^2 + 2n$	$4n$	— <sup>a</sup>	—
BOXNE	$n^2 + 2n$	$3n$	$2n$	33
LIN000	$2n$	$2n$	—	—
VEC000	$n$	$n$	—	—

<sup>a</sup>Method 3 is feasible only if  $B$  is decomposable in simpler elements.

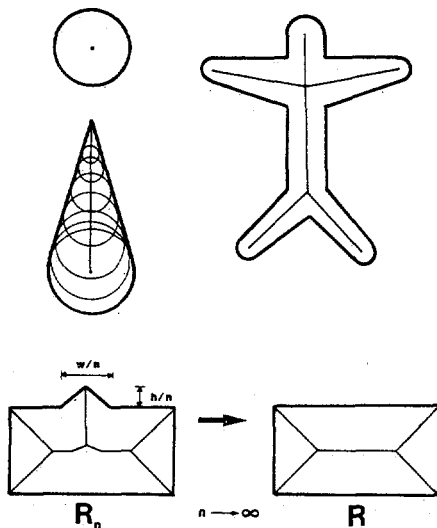


Fig. 5. Examples of skeletons of continuous images.

ters of the maximal disks inscribable inside  $X$ . A disk is *maximal* if it is not properly contained in any other disk totally included in  $X$ . Hence, a maximal disk must touch the boundary of the object  $X$  at least at two different points. Some examples of skeletons are shown in Fig. 5, where we see that the skeleton of a circle is just its center and the skeleton of the angular sector is its bisector. In the same figure (angular sector), we see that, if at each skeleton point we draw its corresponding maximal disk, then the union of all these maximal disks will be exactly equal to the original object and the envelope of the maximal disks will be the boundary of the object. Let  $S_r(X)$ ,  $r > 0$ , denote the  $r$ th skeleton subset, i.e., the set of the centers of the maximal disks whose radius is equal to  $r$ . These skeleton subsets can be obtained by using morphological erosions and openings. More precisely, assume that the class of subsets of  $R^2$  on which the skeleton is defined is the class of open sets, which is equivalent to the previously selected class of closed sets by considering their complements. Selecting open sets as the class of skeletonizable sets excludes isolated points and lines of zero thickness, which are closed sets and their own skeletons. Further, the open set  $X$  is assumed to be nonempty and to contain no half-space. Under these assumptions, Lantuejoul proved [28, p. 376] that its skeleton  $SK(X)$  exists and is equal to

$$\begin{aligned}
 SK(X) &= \bigcup_{r>0} S_r(X) \\
 &= \bigcup_{r>0} [(X \ominus rB) - (X \ominus rB)_{drB}] \quad (9)
 \end{aligned}$$

where  $rB$  denotes the open disk of radius  $r$  and  $drB$  is a closed disk of infinitesimally small radius  $dr$ . The boundaries of the eroded sets  $(X \ominus rB)$  can be viewed as the propagating wavefronts of Blum's grassfires where the propagation time coincides with the radius  $r$ . Subtracting from these eroded versions of  $X$  their opening by  $drB$  retains only the angular points, which are points of the skeleton. The original set  $X$  can be reconstructed as the union for all  $r > 0$  of the subsets  $S_r(X)$  dilated by the open disks  $rB$ , respectively.

The main topological properties of the morphological skeleton are summarized in [28] in the form of a few theorems due to Calabi and Matheron. The transformation of a binary image into its skeleton is not continuous. For example, Fig. 5 shows a sequence of rectangles  $R_n$  having on the outside of their boundary a triangular notch of width ( $w/n$ ) and height ( $h/n$ ). In the limit as  $n \rightarrow \infty$  we see that the limit of the sequence of rectangles  $R_n$  is the rectangle  $R$  without the notch. The notch of every member  $R_n$  of the sequence induces a bone in its skeleton, but the bone disappears discontinuously in the skeleton of the limit rectangle  $R$ . This inducement of a bone in the skeleton by an angle or notch in the image object boundary causes the skeleton transformation to violate the morphological principle of local knowledge and, hence, it is not a quantitative morphological transformation. However, it is still invariant under translation and scale change and is lower-semicontinuous. For more discussion about morphological skeletons of continuous images and their differences from Blum's medial axis, refer to [28]. Henceforth, we restrict our discussion only to discrete images.

Although the skeleton is not an easily digitalizable concept, we can still define the morphological skeleton of binary images sampled on a rectangular or hexagonal grid. In addition, digitization gives us a practical advantage because we can program the shape and size of the structuring element (discussed later). Let the subset  $X$  of  $Z^2$  represent a discrete binary image. It makes no difference whether  $X$  is open or closed because all subsets of  $Z^2$  are both open and closed using the Euclidean topology. Henceforth, we assume that  $X$  is nonempty and bounded.

Serra [28] provided an algorithm for the morphological skeleton  $SK(X)$  of a discrete binary image  $X$  sampled on a hexagonal grid:

$$S_n(X) = (X \ominus nB^s) - (X \ominus nB^s)_B, \quad n = 0, 1, 2, \dots, N \quad (10)$$

$$SK(X) = \bigcup_{n=0}^N S_n(X) \quad (11)$$

where  $S_n(X)$ , or simply  $S_n$  if there is no risk of confusion, denotes, henceforth, the  $n$ th skeleton subset of  $X$ . In Serra's formulation, the discrete structuring element  $B$  was the seven pixel symmetric hexagon on a hexagonal grid. This seven pixels hexagon can be said to have a "discrete radius" of one, since the distance from any of its six vertices to the center pixel (expressed in number of pixels) is equal to one. Then the element  $nB$  in (10) denotes a discrete hexagon of "radius  $n$ ." The index  $N$  denotes the maximum "radius" of a hexagonal element  $nB$  after which a further erosion erodes  $X$  down to the empty set. Equation (11) implies that the skeleton  $SK(X)$  of  $X$  is obtained as the finite union of these  $N + 1$  skeleton subsets.

An intriguing property of the morphological skeleton transformation is that it has an inverse. That is, the discrete binary image  $X$  can be *exactly* reconstructed as the finite union of its  $N + 1$  skeleton subsets dilated by the hexagonal structuring element of proper size:

$$X = \bigcup_{n=0}^N [S_n(X) \oplus nB]. \quad (12)$$

We postpone the proof of (12) until Section III-C.

### B. Some New Considerations on Morphological Skeletons

Serra's digital algorithms for skeleton decomposition and reconstruction (10)–(12) refer to hexagonally sampled binary images. Comparing (9) to (10), we see that the continuous disks were replaced by discrete hexagons. In addition, the continuously varying radius  $r$  of the maximal disks  $rB$  was replaced by the discretely varying "radius"  $n$  of the discrete hexagons  $nB$ . Finally, the opening by a disk  $drB$  of infinitesimally small radius in (9) was replaced in the discrete case by an opening by a hexagon  $B$  of "radius" one in (10).

For images sampled on a hexagonal grid, the hexagon is a natural and close approximation to a circle. However, symmetric hexagonal shapes cannot be formed on a rectangular sampling raster. In addition, although circles and hexagons are desirable as structuring elements due to their high degree of symmetry, there is no need to restrict structuring elements to be circular in shape. For this reason, we considered extensions of digital skeletonization algorithms to rectangularly sampled binary images by using symmetric convex structuring elements on the rectangular grid, such as the CIRCLE, SQUARE, and RHOMBUS elements of Fig. 4. If we consider these three elements to have a discrete radius one, then, as in the case

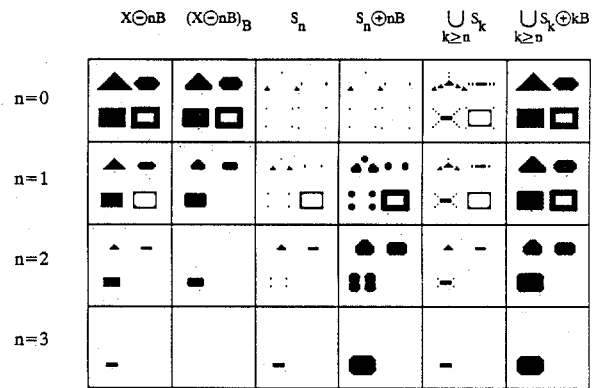


Fig. 6. Step-by-step skeletal decomposition of an image  $X$  and reconstruction of the image from its skeleton subsets (structuring element = CIRCLE).

of the discrete hexagon, we can form similarly shaped elements of discrete radius  $n$ . For example, a SQUARE element of radius  $n$  will consist of  $(2n + 1) \times (2n + 1)$  pixels. Fig. 6 illustrates in a detailed way how a rectangularly sampled binary image can be decomposed into its skeleton subsets and reconstructed from them. The structuring element  $B$  used in this example is the CIRCLE of Fig. 4. More specifically, in Fig. 6, proceeding from left to right columns, we see an image object  $X$  and its erosions by  $nB$ , the openings of these erosions by  $B$ , the skeleton subsets  $S_n(X)$ , the dilated subsets, the composition of the skeleton  $SK(X)$  as the union of the skeleton subsets (11), and the composition of the image  $X$  as the union of the dilated skeleton subsets (12). Fig. 6 shows that the  $n$ th skeleton subset is obtained by eroding  $X$  by  $nB$ , and then keeping from every eroded set  $(X \ominus nB^s)$  only those parts which consist of angular points and lines without thickness. These parts are the only ones remaining after the set difference between  $(X \ominus nB^s)$  and its opening  $(X \ominus nB^s)_B$ . The maximum skeleton subset index required for the example in Fig. 6 is equal to  $N = 3$ . In general, if the bounded object  $X$  is contained inside a square analysis mask of  $M \times M$  pixels, then  $N \leq M - 1$ . Finally, from Fig. 6 and (12), we see that the subsets of large indexes reconstruct the bulky parts in the center of the image object, whereas the small index skeleton subset are responsible for reconstructing the finer details of the image toward its boundary.

So far, we have considered only convex symmetric structuring elements for skeletonization, because they offer a discrete version of the disk on the real plane. These convex symmetric elements have a "discrete radius  $n$ ," which assumes only integer values. However, the concept of the radius limits our potential structuring elements only to those which have an approximately circular shape. Motivated by our intuitive idea to skeletonize an image object using an arbitrarily shaped structuring element  $B$ , and by the fact that the discrete symmetric convex elements  $nB$ , considered above, can be also obtained through an  $n$ -fold dilation of  $B$  by itself, we were led to *quantify* the *size* of any discrete structuring element in the following way. If an arbitrary discrete structuring element  $B$  is considered to be of "size one," then the respective element of "size

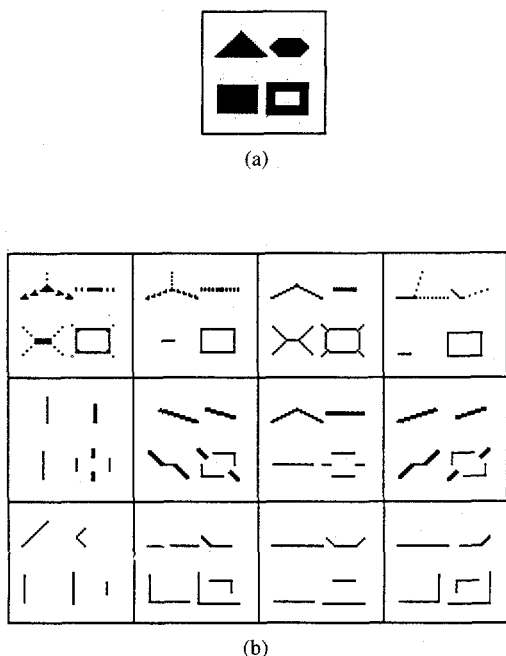


Fig. 7. (a) Test image of  $64 \times 64$  pixels. (b) Its skeletons with respect to all structuring elements of Fig. 4 (keeping the same order from top left to bottom right).

$n$ ," denoted as " $nB$ ," is defined henceforth in our analysis as the Minkowski sum of  $B$  with itself  $n$  times:

$$nB = B \oplus B \oplus \cdots \oplus B \quad (n \text{ times}). \quad (13)$$

Thus, our definition here of the "size" is used henceforth as a concept analogous to the disk radius in the continuous case, and the role of the continuous maximal disk of radius  $r$  will be played now by the maximal element  $nB$  of size  $n$ . The size of discrete element  $nB$  is quantified by the discrete index  $n$ , whereas its shape is quantified by the shape of the unit size primary element  $B$ . For  $n = 0$ ,  $nB$  is just the one-point set  $\{0\}$ . If  $B$  is convex, then  $nB$  will have the same shape as  $B$  but magnified  $n$  times. Henceforth, we assume that  $B$  is bounded and that  $B$  contains the origin of the plane. Obviously, different unit size structuring elements  $B$  will result in different skeletons, and Fig. 7 illustrates this idea. That is, Fig. 7(a) shows a test image of  $64 \times 64$  pixels, and Fig. 7(b) illustrates its skeletons with respect to all the structuring elements of Fig. 4. For a symmetric structuring element, the resulting skeleton may be disconnected but it will lie in the middle of the object as an approximate symmetric axis. However, the asymmetric structuring elements give skeletons which no longer look like a symmetric axis.

Independently of which structuring element is used for skeletonization, the resulting skeleton subsets are able to exactly reconstruct the original image using (12). Thus, the total information in the original finite image  $X$  is equivalent to that in the finite ensemble of all its skeleton subsets  $S_n(X)$  together with their corresponding index " $n$ ." In order to represent more compactly all the information in the skeleton, we defined the *morphological*

*skeleton function*  $skf(X)$  of  $X$  as the following 2-D discrete image array:

$$[skf(X)](i, j) = \begin{cases} n + 1, & (i, j) \in S_n(X) \\ 0, & (i, j) \notin SK(X) \end{cases} \quad (14)$$

where  $(i, j) \in Z^2$ . We will prove later that the skeleton subsets are disjoint sets and, hence, the skeleton function is a single-valued function. The skeleton function is a graytone function which has the same region of support as the binary skeleton  $SK(X)$ , and it is equivalent to the finite ensemble of all the binary skeleton subsets. Thus, if we know the skeleton function, then we also know the skeleton subsets; i.e., the  $n$ th skeleton subset is the set of all those points of  $Z^2$  at which the value of the skeleton function is equal to  $n + 1$ .

### C. Fast Algorithms for Skeleton Decomposition and Reconstruction

In this section we exploit some algebraic properties of the morphological set operations in order to develop two fast algorithms for skeleton decomposition and reconstruction, respectively, which reduce the complexity of Serra's algorithms from quadratic to linear.

Table I shows that erosion of  $X$  by  $nB$ , which requires  $O(n^2)$  parallel logical AND's of the whole image, can be done much faster by successively eroding  $X$  by  $B$   $n$  times, which requires  $O(n)$  parallel AND's. Thus, adopting the latter method, we see from (10) that the skeleton decomposition of  $X$  requires  $(N^2 + N)/2$  erosions by  $B$  for the erosions  $X \ominus nB$ , plus  $N + 1$  erosions and  $N$  dilations by  $B$  for the openings by  $B$ , which amounts to a total of  $(N^2 + 5N + 2)/2$  erosions-dilations of the image by  $B$ . A new algorithm which avoids this quadratic complexity of the decomposition of the image into skeleton subsets is given below. Let EROS1, EROS2, OPEN denote three accumulator sets large enough to hold the image object and its background; then,

- step 1:  $n := 0$ , EROS1 :=  $X$
- step 2: EROS2 := EROS1  $\ominus B^s$
- step 3: if EROS2 =  $\emptyset$ ,  
then  $N := n$ ,  $S_N(X) :=$  EROS1, and STOP
- step 4: OPEN := EROS2  $\oplus B$
- step 5:  $S_n(X) :=$  EROS1  $-$  OPEN
- step 6:  $n := n + 1$ , EROS1 := EROS2,  
and go to step 2

where " $:=$ " means "replaced by." The above algorithm, depicted in Fig. 8(a), requires only  $N$  erosions and  $N$  dilations of the image by  $B$  and, hence, it has linear complexity.

In the process of proving (12), we provide below a faster algorithm for exact reconstruction from the skeleton function. Let  $A$  be an accumulator set large enough to hold the image object and its background; then,

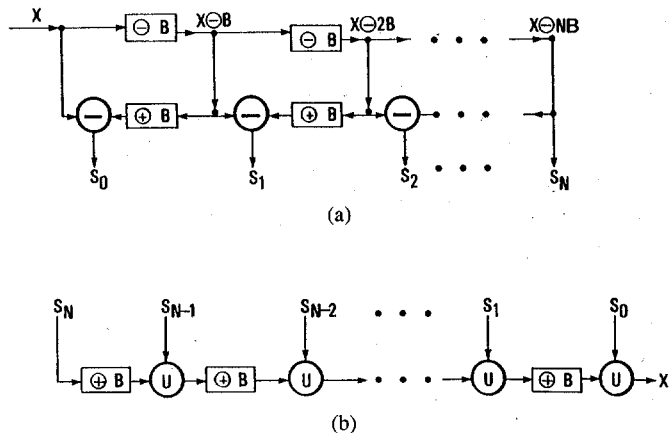


Fig. 8. Fast algorithms for (a) decomposition into skeleton subsets and (b) reconstruction from skeleton subsets.

- step 1:  $n := N, A := \emptyset$   
 step 2:  $A := A \cup S_n(X)$   
 step 3: if  $n = 0$  STOP, otherwise  $A := A \oplus B$   
 step 4:  $n := n - 1$  and go to step 2.

Let us go through some iterations of the above algorithm, which is also illustrated in Fig. 8(b).

*First Iteration:* For a bounded image  $X$ , the last subset  $S_N(X)$  must equal  $X \ominus NB^s$ , because otherwise the  $(N + 1)$ th skeleton subset would be nonempty. Thus, after step 3,  $A = (X \ominus NB^s) \oplus B = [X \ominus (N - 1)B^s]_B$ .

*Second Iteration:* After step 2,  $A = X \ominus (N - 1)B^s$ . Thus, after step 3 of the  $m$ th iteration,  $A = [X \ominus (N - m)B^s]_B$ , and after step 2 of the  $(m + 1)$ th iteration,  $A = X \ominus (N - m)B^s$ . At the end, when  $m = N$  (after  $N + 1$  iterations), the accumulator set  $A$  contains the original image  $X$ . The above algorithm requires only  $N$  dilations of the entire image by the unit-size structuring element  $B^s$ . In contrast, Serra's algorithm (12) requires  $(N^2 + N)/2$  such dilations since, for  $n = 1, 2, \dots, N$ , it involves dilation of the  $n$ th skeleton subset by  $nB^s$ . The algorithm (16) is summarized by the equation below:

$$X = [[S_N(X) \oplus B] \cup S_{N-1}(X)] \oplus B \cup S_{N-2}(X) \cdots ] \oplus B \cup S_0(X). \quad (17)$$

The algorithm (17) is equivalent to algorithm (12), because dilation distributes over set union and is associative. In (17), the reconstruction propagates from the center of the object toward its boundary. However, in (12), the reconstruction can propagate either direction, depending on whether we start the union of the dilated skeleton subsets from  $n = N$  or from  $n = 0$ . Mott-Smith [21] also proved (17), but in his approach only symmetric structuring elements were considered.

By using the decomposition, if possible, of  $B$  into a Minkowski sum of simpler elements, both algorithms (15) and (16) can be further speeded up by 33–60 percent as Table I explains.

#### D. Partial Reconstruction from Skeleton Function

Using all the skeleton subsets guarantees exact reconstruction of the original image. If some of the skeleton subsets are omitted, the image will be partially reconstructed. Our objective in this section is to *quantify* this partial reconstruction.

In the reconstruction algorithm (16), if we omit all the implied operations which take place after adding the  $(N - m)$ th skeleton subset, then we reconstruct only  $X \ominus (N - m)B^s$ . Replacing  $N - m$  by  $k$  and using distributivity and associativity of dilation yields the following:

$$X \ominus kB^s = \bigcup_{n=k}^N [S_n(X) \oplus (n - k)B] \quad (18)$$

for  $1 \leq k \leq N$ . The above formula allows us to obtain *eroded* versions of  $X$  by omitting some skeleton subsets and dilating the rest of them by elements of smaller size. Of course, using (17) and stopping the algorithm at the proper point leads to faster implementation of (18). If we dilate both sides of (18) by  $kB$ , we obtain the *openings* of  $X$  by  $kB$ , where  $1 \leq k \leq N$ :

$$X_{kB} = \bigcup_{n=k}^N [S_n(X) \oplus nB]. \quad (19)$$

By comparing (12) and (19), we see that, if we omit the first  $k$  skeleton subsets, we reconstruct the opening of  $X$  by  $kB$ . Since the opening can be viewed as a nonlinear filter for the image boundary, it may be of interest to compare this property of the skeleton to a conceptually similar property of the Fourier transform. That is, omitting or filtering out the high-frequency coefficients or portion of the spectrum of a signal corresponds to smoothing the signal. Similarly, omitting the skeleton subsets with small indexes gives smoothed versions of the object (its openings). Again, using (17) to obtain  $X \ominus kB^s$  and dilating the latter by  $kB$  gives  $X_{kB}$  much faster than the algorithm implied in (19). Finally, we can obtain *dilated* versions of the original object  $X$  and its erosions or openings by directly dilating both sides of (12), (18), and (19), respectively. A general formula results then:

$$[X \ominus kB^s] \oplus mB = \bigcup_{n=k}^N [S_n(X) \oplus (n - k + m)B] \quad (20)$$

where  $0 \leq k \leq N$  and  $0 \leq m$ . If  $m = k$ , we get openings. If  $k = 0$  or  $m = 0$ , we get dilations or erosions, respectively. If  $m > k$ , we get dilation of  $X_{kB}$  by  $(m - k)B^s$ . In short, if we omit some skeleton subsets of  $X$  and dilate the rest by elements of proper size, we obtain partial *reconstructions* of  $X$  which can be either smoothed versions (openings) or thinner and thicker versions (erosions and dilations).

#### E. Properties of the Morphological Skeleton

Assume everywhere that  $B$  contains the origin; i.e.,  $\{0\} \subseteq B$ . The nontrivial proofs of the following properties and theorems are given in the Appendix.



*Property 1:* a)  $B = \{0\} \Rightarrow SK(X) = \emptyset$ . b)  $X \ominus B^s = \emptyset \Rightarrow SK(X) = X$ .

*Property 2:* The skeleton subsets are disjoint sets.

*Property 3:* The morphological skeleton is a set transformation which is a) *translation-invariant*, b) *antiextensive* and c) *idempotent*.

*Definition:* The structuring element  $(nB)_z$  of size  $n$  located at point  $z$  is said to be *maximal* in  $X$  provided that  $(nB)_z \subseteq X$  and there does not exist any other element  $(mB)_y$  in  $X$  with  $m > n$  such that  $(nB)_z \subseteq (mB)_y \subseteq X$ .

*Theorem 1:* If the skeleton of  $X$  is obtained with respect to a bounded and convex structuring element  $B$ , then a point  $z$  belongs to the  $n$ th skeleton subset of  $X$  if and only if the element  $(nB)_z$  is maximal in  $X$ .

*Theorem 2:* If the skeleton of a convex image object  $X$  is obtained with respect to a bounded and convex structuring element  $B$ , then  $X$  is equal to its opening  $X_{kB}$  by  $kB$  if and only if its first  $k$  skeleton subsets  $S_n(X)$ ,  $n = 0, 1, 2, \dots, k-1$ , are empty.

Theorem 2 is important because it explicitly relates the skeleton subsets of  $X$  with the smoothness of the boundary of  $X$ . That is, since  $B$  is convex, the openings  $X_{kB}$  will have a smoothing effect on the boundary of  $X$  for any size  $k$ . In addition, the larger the size  $k$  of  $kB$ , the more severe the smoothing. Thus, an object  $X$  is "smoothed to a degree  $k$ " (i.e.,  $X = X_{kB}$ ) if and only if its first  $k$  skeleton subsets are empty. Therefore, the smaller the index  $n$  of  $S_n(X)$ , the greater its contribution to the roughness of the boundary of  $X$ .

### F. Globally and Locally Minimal Skeleton

Can we find some skeleton points which, if removed from the original skeleton, still leave the remaining skeleton function with the property of exact reconstruction? Mott-Smith [21], Rosenfeld and Kak [26], and Frank *et al.* [7] speculated that it may be possible to remove some points of the skeleton and still reconstruct the image exactly, obviously by exploiting the overlapping of the dilated skeleton subsets; but they did not indicate how. Of course, this idea applies only for 2-D structuring elements because we cannot remove any points from skeletons obtained using 1-D structuring elements and still have exact reconstruction. Motivated by the above questions and speculations, we tried to find minimal subsets of the original morphological skeleton.

First, we define a *minimal subset* of the skeleton to be a part of the original skeleton whose points are sufficient for exact reconstruction but removal of just one of these points will result in partial reconstruction. A minimal subset may have points belonging to different skeleton subsets. Note that the word "subset" in our definition of a "minimal subset" means only a subcollection of skeleton points and it is not to be confused with the "skeleton subsets" as defined in (10). A minimal subset always exists since, in the worst case (no redundancy in the skeleton), the original skeleton is the minimal subset. The most straightforward way to find a minimal subset would be to sequentially remove each skeleton point, one at a time,

and use (12) or (17) to reconstruct the image from the remaining skeleton. If exact reconstruction were obtained, then the skeleton point in question could be removed and we could proceed to the next skeleton point. However, the above search procedure would require the use of the reconstruction algorithm (12), or the faster (17), as many times as the total number of skeleton points, and hence, it would be very time consuming. Therefore, we developed two algorithms which find, if it exists, a minimal subset of the original skeleton in a more efficient (computationally faster) way.

The first algorithm finds a *globally minimal* skeleton, which is a subset of the original skeleton guaranteeing the exact reconstruction of the entire image but not its partial reconstructions as quantified by (19). Let  $X$  be the original image and let  $S_n(X)$ ,  $n = 0, 1, \dots, N$ , be its skeleton subsets with respect to a structuring element  $B$ . For each skeleton subset index  $n$ , we create a *binary* function  $k_n(i, j)$ ,  $(i, j) \in \mathbf{Z}^2$ , whose value is equal to one at points  $(i, j) \in nB$  and zero everywhere else; i.e.,  $k_n$  is the *characteristic function* of the set  $nB$ . Then, for each  $n$ , we shift the function  $k_n$  to all the points of  $S_n(X)$ , and add algebraically all these contributions for all the points of  $S_n(X)$  and for all  $n$ . In the end, we will have created a pseudograytone function,  $\text{pgf}(X)$ , whose region of support is identical with the original binary image  $X$  and whose value at every point of the image is  $\geq 1$ ; this pseudograytone function is equal to

$$[\text{pgf}(X)](i, j) = \sum_{n=0}^N \sum_{(r, t) \in S_n(X)} k_n(i-r, j-t) \quad (21)$$

for any  $(i, j) \in \mathbf{Z}^2$ . Now, to decide whether a certain point  $(r, t) \in S_n(X)$  can be removed, we check first whether the value of the pseudograytone function  $\text{pgf}(X)$  at all the points of the region of support of the respective shifted characteristic function  $k_n(i-r, j-t)$  is  $\geq 2$ . If so, then we remove this skeleton point  $(r, t)$ , and we subtract algebraically  $k_n(i-r, j-t)$  from  $[\text{pgf}(X)](i, j)$ . Otherwise, we proceed to the next skeleton point until all the skeleton points have been searched. The question of how to scan the skeleton is still open. The method that we used was to successively search each skeleton subset beginning from  $S_1(X)$  or  $S_N(X)$  and continuing, respectively, in ascending or descending order of their indexes. Note that the points of the 0th subset  $S_0(X)$  cannot be removed because they do not belong to any of the openings of  $X$ ; hence, there is no structuring element  $nB$  of nonzero size  $n$  included in  $X$  that contains any of the points of  $S_0(X)$ . Both scanning methods resulted in approximately the same minimal skeleton. The only difference was that, in a few cases, scanning the skeleton subsets in ascending order resulted in a minimal skeleton with slightly fewer points. One intuitive reason for this may be the fact that the points of the skeleton subsets with large indexes are associated with structuring elements of larger size, and, hence, a smaller number of those is able to remove the redundancy of the skeleton.

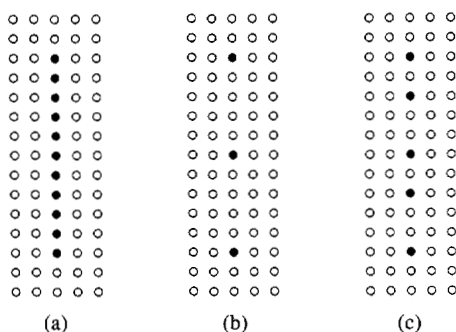


Fig. 9. (a) Original skeleton, (b) globally minimal skeleton, (c) locally minimal skeleton (structuring element = SQUARE, ○, ● = image points, ● = skeleton points).

TABLE II  
NUMBER OF SKELETON POINTS [64 × 64 PIXELS TEST IMAGE OF FIG. 7(a)]

Structuring Element	Original Skeleton	Globally Minimal Skeleton	Locally Minimal Skeleton
CIRCLE	175	77	88
SQUARE	121	61	68
RHOMBUS	149	149	149
BOXNE	100	49	66

The above-defined globally minimal skeleton does not guarantee that the partial reconstructions of the image will be quantified as in (19). Therefore, we introduced the concept of the *locally minimal* skeleton defined as follows. Consider the fast reconstruction algorithm (17). When we add the  $n$ th skeleton subset  $S_n(X)$  and dilate the partial sum by  $B^s$ , we reconstruct  $[X \ominus (n - 1)B^s]_B$ , i.e., the opening of the  $(n - 1)$ th erosion. Using the latter condition as a criterion, we created for each  $n$  a pseudogray-tone function for the opening  $[X \ominus (n - 1)B^s]_B$ , as in (21), and we searched to find which points of  $S_n(X)$  can be removed in the same way as done for the globally minimal skeleton. This algorithm provides a minimal skeleton whose modified skeleton subsets satisfy the partial reconstruction (19). That is, if we do not use the first  $n$  modified skeleton subsets, then we reconstruct  $X_{nB}$ . Obviously, the locally minimal skeleton guarantees exact reconstruction, but it will have more points than the globally minimal skeleton because it was obtained by imposing an additional constraint.

Fig. 9(a) shows an image and its original morphological skeleton with respect to the SQUARE element. Fig. 9(b) and (c) shows, respectively, its globally and locally minimal skeleton. We see that the original skeleton consists of 11 points, where the globally and locally minimal skeletons consist of only 3 and 5 points, respectively. Table II shows the redundancy removed globally or locally from the skeleton of the test image of Fig. 7(a) using different structuring elements. The globally minimal skeletons were found by searching the skeleton subsets in ascending order. The results in Table II indicate that the globally minimal skeleton may have more than 50 percent fewer points than the original skeleton. Fig. 10 shows two original images of different resolution (the 64 × 64 pixels

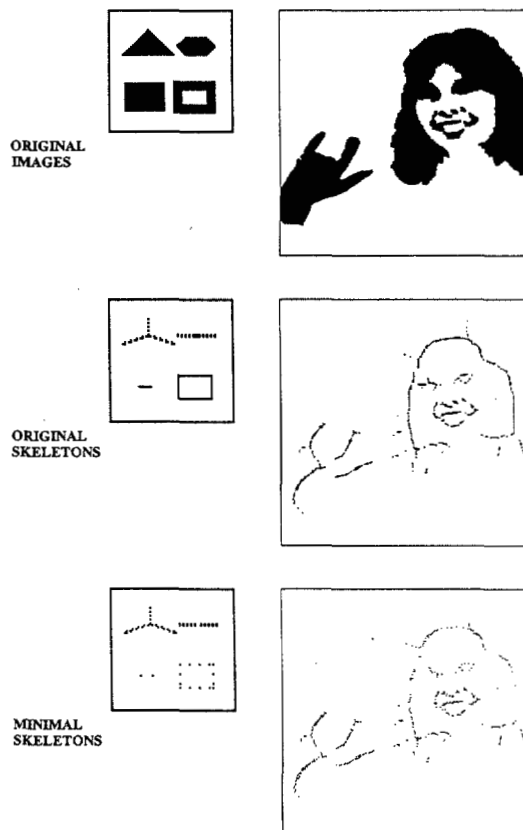


Fig. 10. Original images, original skeletons, and globally minimal skeletons (structuring element = SQUARE).

test image and a 256 × 256 pixels image showing a hand and a face shape), but magnified differently for display purposes. Fig. 10 also shows their original skeletons with respect to the SQUARE element and their globally minimal skeletons. The reduction in skeleton points can be

clearly seen in the minimal skeletons, although, for these two images, the original skeletons are more aesthetically pleasing.

### G. Related Previous Work and Discussion

To explain the similarities and differences of our work with previous work, we will interpret parts of previous research using morphological concepts and definitions, although the respective researchers have used quite different notation and mathematical algorithms to obtain their results. Let  $D_n = [X \ominus nB^s] - [X \ominus (n + 1)B^s]$  for  $n = 0, 1, \dots, N - 1$ . Then, Rosenfeld and Pfaltz [27] find the skeleton by first computing a "distance transform" image whose value at points of  $D_n$  is equal to  $n + 1$ . The implied structuring element is either the SQUARE or the RHOMBUS of Fig. 4, which correspond to the "chess-board" or "city-block" [26] distance metric, respectively, on discrete images. The skeleton points are then found as the points of local maxima of this distance transform image, whose restriction to the skeleton points is equal to the skeleton function in (14). In Pfaltz and Rosenfeld [24], the image is represented as the union of maximal neighborhoods, which are equivalent to the dilated skeleton subsets in (12) using the RHOMBUS element. In [26, pp. 213 and 217], the two parallel algorithms for skeletal decomposition and reconstruction from the distance transform image have comparable complexity with our two fast algorithms (15) and (16); the other two algorithms in [26, pp. 214, 217] require only three scans of the image, but they are strictly *serial* and need some modifications to handle general asymmetric structuring elements. The advantage, however, of the morphological approach is that it is simpler and faster (for parallel algorithms) since it requires only parallel AND/OR's on binary images, whereas the distance transform approach requires local minima and maxima on graytone transform images. In [24], [26], and [27], the structuring elements are limited only to those related to some distance metric, and there is no explicit use of the partitioning of the skeleton into subsets. However, Mott-Smith [21] discriminated among the different skeleton subsets and used symmetric "neighborhoods" for structuring elements. He defined an "expansion" operation (Minkowski sum), but not an erosion or opening. Hence, he peels off the successive layers  $D_n$  from the image object by dilating the background and then intersecting it with the object. As a result, his skeletal decomposition algorithm requires two additional parallel operations on the whole image per each skeleton subset when compared to our algorithm (15). The full generality offered by selecting an arbitrarily shaped structuring element in the morphological skeleton is conceptually found only in the work by Frank *et al.* [7]. They implicitly used skeletonization ideas when they talked about "growth patterns" (structuring elements), "seed pixels" (skeleton points), "number of growth stages" (index of skeleton subset), and "increasing or decreasing growth" (dilation or erosion).

The morphological skeleton, and our work in this area,

is different from the above approaches because it allows us to view the original image as a set and to quantify, using a compact notation and mathematical formalism, the required shrinking and expanding of the image by using erosion and dilation of sets. Moreover, these morphological set operations can be implemented with simple parallel computer architectures and are the prototypes for a large class of systems [16], [17]. In addition, the morphological skeleton has the following advantages. 1) It can be defined for both continuous and discrete images. 2) The morphological set operations are inherently separable, and this results in a significant computational savings. 3) The morphological structuring element frees the skeleton from the limitations of a distance or a symmetric neighborhood and, thus, it unifies and generalizes previous approaches. 4) The partial reconstruction (openings, erosions, and dilations of the original) can be addressed and quantified by simply omitting a few skeleton subsets. 5) The global and local minimality of the skeleton can be defined and investigated. 6) By using a different structuring element for the successive erosions of the image, we can further generalize the morphological skeleton as explained in Maragos [16, p. 191].

### IV. SKELETON CODING OF BINARY IMAGES

The most prominent methods for coding binary images are *block* coding and *runlength* coding, as summarized by Huang [10]. Intuitively, the compression efficiency of both of these methods is dependent upon how "thin" (sparse) the binary image is. In Section III we saw that the skeleton  $SK(X)$  of  $X$  is guaranteed to be thinner than  $X$  (antiextensive transformation). Moreover, the skeleton subsets form a partition of the skeleton and, hence, each skeleton subset is even thinner than  $SK(X)$ . Each skeleton subset is a binary image by itself. Thus, it is natural to consider techniques for coding the skeleton rather than the image itself. In this section we describe our study of a binary image coding method in which each skeleton subset is coded individually using block, runlength, or Elias [6] coding (explained later and suitable to code very sparse binary signals), and then all or some skeleton subsets are used to fully or partially reconstruct the image. The motivation for the above method is the intuitive idea that, because of the thinness of the skeleton subsets, the sum of the bit rates required to code all of them may be smaller than the bit rate required to code the original image directly using block or runlength coding. In addition, the information of the original binary image is totally contained in its skeleton function, which is a graytone image. Hence, we can alternatively code this graytone skeleton function using an efficient scheme. Coding the skeleton subsets or coding the skeleton function have the same result with respect to reconstructing the image at the decoder (receiver). In both cases, the image will be reconstructed progressively, either starting from the inside "bulky" parts and propagating toward the image boundary, or starting from the outside fine details and reconstructing toward the center of the image. This aestheti-

TABLE III  
NUMBER OF BITS TO ENCODE ORIGINAL IMAGE  $X$  AND ITS GLOBALLY MINIMAL SKELETON [ $B =$   
STRUCTURING ELEMENT BOXNE,  $X = 64 \times 64$  PIXELS TEST IMAGE OF FIG. 7(a)]

Image Coded	Coding Skeleton Subsets				
	Block-Huffman ( $P = 2 \times 4$ )	Runlength-Huffman		Elias ( $m = 3$ )	Image Reconstructed
		Common <sup>a</sup>	Separate		
$X$	946	837	714	4080	$X$
$\{S_N(X), S_{N-1}(X), \dots, S_{k+1}(X), S_k(X)\}$					
$k = 0$	5712	1153	999	496	$X$
$k = 1$	5195	1063	920	462	$X_B$
		Coding Skeleton Function			
$SK(X)$	625	479	420	344	$SK(X)$
$skf(X)^b$	779	633	574	498	$X$

<sup>a</sup>“Common” = same Huffman code for black and white runlengths, as opposed to “separate.”

<sup>b</sup>The Huffman rate for the amplitudes of  $skf(X)$  is the difference between the rates for  $SK(X)$  and  $skf(X)$ .

cally pleasing progressive image reconstruction may be desirable for transmission of images at low bit rates, as emphasized in [7]. Thus, the decoder can wait to receive all the skeleton subsets and then start the reconstruction from index  $n = N$  exploiting the fast algorithm (17). Alternatively, the incoming  $n$ th skeleton subset (transmitted in real time while the image is progressively decomposed into skeleton subsets of larger indexes) could be dilated by a structuring element  $nB$ . This latter approach could be speeded up if the structuring elements  $nB$  for different sizes  $n$  are prestored as templates at the decoder (an idea suggested by Frank *et al.* [7]). Coding the skeleton subsets leads to a faster real-time transmission of the image at the encoder (transmitter) than coding the skeleton function, since each skeleton subset can be encoded and transmitted while the next one is being obtained, whereas the skeleton function can be encoded only after all the subsets have been obtained. Thus, coding the skeleton subsets seems better suited for real-time image transmission, whereas coding the skeleton function may be better for storage purposes since the skeleton function is a more compact representation. We next determine, for each approach, the best technique to code the skeleton information of test images, and then we compare the best result of skeleton coding to directly coding the original image. To conclude, we give an application of skeleton image coding. At this point we emphasize that we will henceforth deal only with the globally minimal skeleton. One reason is that it guarantees exact reconstruction of the original image and one of its partial reconstructions, i.e., its first opening. In addition, since it consists of a minimal subset of skeleton points, the globally minimal skeleton will obviously lead to a higher compression efficiency than coding the original skeleton.

#### A. Coding the Skeleton Subsets

The skeleton subsets of an  $M \times M$  pixels binary image will also be finite binary images of  $M \times M$  pixels each. To code the original image or its skeleton subsets, we

tried three coding techniques: block coding, runlength coding, and Elias coding, as explained below.

**Block-Huffman Coding [10]:** According to Shannon's theory of discrete source coding [29], we consider the discrete binary image  $X \cup X^c$  (image object and its background) as a sample function of a 2-D stochastic process characterized by joint probability distributions of all orders. We further partition the image in 1-D or 2-D blocks of  $P$  pixels each, which cover the entire image without overlap. Clearly, there are  $2^P$  different such blocks and we consider them as our source messages. Let  $p_i$  denote the probability (in practice, we measure frequency of occurrence) of the  $i$ th block. Then, the  $P$ th-order joint entropy  $H_P$  (in bits/pixel) of the binary image is equal to

$$H_P = -(1/P) \sum_{i=1}^{2^P} p_i (\log_2 p_i). \quad (22)$$

As is well known,  $H_P$  is a nonincreasing function of  $P$ , and the limit as  $P \rightarrow \infty$  is the entropy of the stochastic source. We can employ Huffman [11] or other suboptimum codes [10] to code these  $2^P$  different blocks (messages) at rates very close to  $H_P$ . Specifically, if  $R_{\text{HUF}}$  denotes the rate (in bits/pixel) achieved by Huffman encoding of the blocks, then  $H_P \leq R_{\text{HUF}} < H_P + (1/P)$  [10]. We henceforth refer to Huffman encoding of blocks as “block-Huffman” coding. Since for some classes of images the skeleton is a much thinner binary image than the original, then its  $P$ th-order joint entropy (and hence, the Huffman rate to encode it) will be lower too. Table III (look at the case of coding the skeleton function, as explained in Section IV-B) verifies the above intuition for the  $64 \times 64$  pixels test image of Fig. 7(a). Henceforth, all our results concerning block-Huffman coding will refer to coding 2-D blocks whose size is  $P = 2 \times 4 = 8$  pixels. For larger values of  $P$ , the number of Huffman codewords exceeds 256 and, hence, it becomes impractical for table lookup.

**Runlength-Huffman Coding [10]:** The  $M \times M$  pixels binary images of 0's and 1's (with 0 = white and 1 =

black) is scanned line by line, and blocks of  $L$  consecutive black (respectively, white) pixels are found, where  $1 \leq L \leq M$ . These 1-D blocks are called black (respectively, white) *runs*, and their length  $L$  is called the *runlength*. If we know the runlengths of all the black and white runs, which obviously alternate, together with the first column of the image (for identification of the first run on each image line), then we know the whole image. Thus, we can code the image by encoding the runlengths, either by using Huffman codes (which are optimum among all variable-length codes), or other suboptimum codes [10] to avoid some disadvantages of the Huffman codes such as their high sensitivity to a change of the source statistics. In using Huffman codes for the runlengths, we can either use the same Huffman code for both black and white runlengths or, reaching the upper limit of optimality [10], use two separate Huffman codes distinguishing between black and white runlengths. In this paper, we investigated both ways of Huffman encoding of the runlengths for both the original image and its skeleton subsets. Table III shows the bit rates for runlength coding of the skeleton information, where we see that the runlength coding rate for the skeleton function is smaller than the rate resulting from runlength coding of the original image. In the entries of Table III corresponding to runlength coding, we added a constant equal to  $M$  bits (in the case of coding the subsets we added this constant only on the first subset), because of the required image column to identify black or white runs at the beginning of each line of an encoded image. However, we did not add the extra bit rate required either for end-of-line synchronization, or for a symbol separating successive skeleton subsets.

*Elias Coding* [6]: Suppose that a binary image is very sparse consisting mainly of 0's with a few 1's. We can concatenate all the image lines to form an  $M^2$ -pixels image vector and we find all the runs of 0's of length  $L \geq 0$ . If we know all these runlengths and separate them by some symbol marking the end of a run (equivalently, the presence of a 1), then we know the original image exactly. Elias's idea was to code these runlengths  $L$  of 0's (viewed as decimal numbers) by representing them using the  $m$  symbols of a  $m$ -ary arithmetic system and a  $(m + 1)$ th symbol (referred to henceforth as "comma") to represent the 1's separating the consecutive runs of 0's. For  $m = 1$ , the Elias code is the same as the original binary signal. For  $m = 2$ , we have a binary system to represent the runlengths  $L$ . Since, however, the comma is an extra symbol, the coded sequence will be in ternary digits. For  $m = 3$ , the runlengths are represented in a ternary system and Elias code requires  $m + 1 = 4$  different symbols which can be obtained using a 2-bit code. Specifically, we chose the 4 required symbols "0, 1, 2, comma" to correspond to the 2-bit codes 00 = "comma," 01 = "0," 10 = "1," 11 = "2." Consider, for example, the 40-bit binary sequence

00000000001000000100000011001000000001.

The sequence of runlengths  $L$  is 11, 6, 7, 0, 2, 8 and the Elias code ( $m = 3$ ) is

100111 00 1101 00 1110 00 01 00 11 00 1111.

The coded sequence is shorter (32 bits) than the original, and this compression will increase with the average runlength. By using larger values of  $m$ , provided that the binary input sequence is very sparse, we can further increase the compression efficiency up to certain limits [6].

We coded the skeleton subsets of the globally minimal skeleton using Elias codes ( $m = 3$ ) and, for each skeleton subset, after its last point had been encoded, we transmitted two consecutive commas to signal the end of this subset and to separate successive subsets. In this way, the presence of an empty skeleton subset manifests itself as four consecutive commas. By looking at the fast reconstruction algorithm (17) we see that, before we add the  $n$ th skeleton subset  $S_n(X)$ , we have already reconstructed  $(X \ominus nB^s)_B$ . By its definition,  $S_n(X)$  is disjoint from this previous partial sum and surrounds it from the outside since the reconstruction in (17) has an outward flow. Hence, we computed the runlength between two consecutive points of  $S_n(X)$  by skipping over all the intervening points of the partial sum  $(X \ominus nB^s)_B$ . This technique, which we call *masking*, results obviously in shorter runlengths and, consequently, in smaller bit rates using Elias codes, since the number of bits in Elias codes decreases logarithmically with the runlength. Table III contains some results which show that Elias coding of the skeleton subsets performed better than any other method of coding the skeleton subsets or the original image itself, mainly because the skeleton subsets are very sparse images for which Elias coding performs very well. Elias coding of the original image performed very poorly, as expected, because of the presence of too many consecutive 1's which require a lot of commas.

### B. Coding the Skeleton Function

The skeleton function  $skf(X)$  of a binary image  $X$  conveys two types of information. First, the location of skeleton points, and, second, its values at such points which are related to the indexes of the respective skeleton subsets. Thus, we coded the skeleton function by coding first the whole skeleton  $SK(X)$ , which is a binary image, using block, runlength, or Elias coding, and then the graytone information (the values  $n + 1$  for  $n = 0, 1, \dots, N$ ) using a Huffman code. The results for the original test image of Fig. 7(a) are shown in Table III. There, we see that coding the skeleton function using block or runlength coding of the whole skeleton performs better than coding the skeleton subsets or the original image. However, in the case of using Elias codes, the criterion for choosing between coding the skeleton subsets or coding the skeleton function should be other than their compression efficiency, because both methods achieve approximately the same compression ratio. Again, Elias coding was the best method to code the binary image of the whole skeleton.

### C. An Application

In Fig. 11(a) we see a binary image of  $256 \times 256$  pixels showing a person forming a symbol in a sign language.

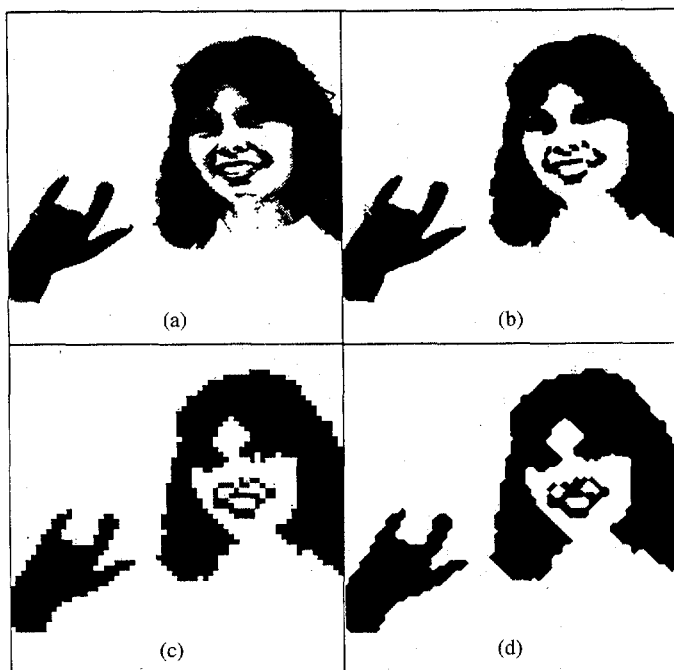


Fig. 11. (a) Original binary image of  $256 \times 256$  pixels, (b) opening of (a) (65536 bits; after skeleton coding 5734 bits), (c) image in (a) decimated  $4 \times 4$  times and then interpolated (4096 bits; after skeleton coding 1246 bits), (d) closing-opening of interpolated image in (c) by a RHOMBUS element of size 2.

This image was obtained by thresholding a low-detailed graytone version. The interest in this kind of image arose from the need to develop a videophone for visual communication between deaf persons over the public telephone network [30]. The opening of the image in Fig. 11(a) by a SQUARE element is shown in Fig. 11(b). The original information in the image of Fig. 11(b) is 65.5 Kbits. Elias coding of the globally minimal skeleton (using a SQUARE element) of the image in Fig. 11(b) compressed it down to 5.7 Kbits; i.e., a compression ratio of 11.4:1. Runlength-Huffman and block-Huffman coding of the images in Fig. 11(b) gave compression ratios of 8.3:1 and 5.1:1, respectively. If the quality criterion is only the ability to recognize the hand and lip shape information in this sign image, we can further decimate the image of Fig. 11(b) by a factor of  $4 \times 4$  producing an image of  $64 \times 64$  pixels image whose interpolated version (repeating rows and columns 4 times) is shown in Fig. 11(c). Furthermore, Elias coding of the skeleton information (using VEC090 as structuring element) on the  $64 \times 64$  pixels image of Fig. 11(c) further compressed the information by a factor 3.3:1. Also, omitting a few skeleton subsets results in a compression ratio of about 4:1 for smoother versions of the image (openings). Note that block-Huffman and runlength-Huffman coding of the same uninterpolated image gave a compression ratio of 3.5:1 and 2.9:1, respectively. Thus, for thin images containing isolated lines and points, as the  $64 \times 64$  pixels image of Fig. 11(c), the superior compression efficiency of the skeleton drops down and it becomes comparable with block and runlength coding. The interpolated image of Fig. 11(c), which has edges like "stair steps," was

smoothed using a morphological closing-opening. The smoothed image is shown in Fig. 11(d). In going from the image of Fig. 11(a) to the images of Fig. 11(c) and (d), the information has been compressed  $16 \times 3.3 \approx 53$  times. The images of Fig. 11(c) or (d) after skeleton coding can be encoded using 1.2 Kbits. Hence, by using a frame rate of 8–9 frames per second, which was judged as sufficient in [30], one could transmit moving images of this kind or other low-detailed scenes over telephone lines at a rate of 9.6 Kbits/s. Moreover, the skeleton representation of these images facilitates image filtering or enhancement, since it can provide us with eroded, dilated, opened, and closed versions of the original images.

#### D. Discussion on Skeleton Coding

The selection of a structuring element is critical in skeleton coding as Table IV shows. Generally, smaller structuring elements give thinner skeletons, but a larger number of skeleton subsets. The question of how to choose the "optimum" structuring element to achieve the highest compression of the image still remains open. Assuming that the structuring element that minimizes the number of points in the skeleton was the best, we found the skeletons of the image with respect to all elements of Fig. 4, and then selected the element giving the smallest number of skeleton points. Of course, the coding efficiency of the skeleton depends not only on the number of its points, but also on their relative location. However, for practical purposes and for a fast decision rule, the above criterion is relatively good; e.g., Table IV shows that, searching for the skeleton with the fewest points, the element BOXNE is chosen which gives the skeleton function and the skeleton subsets with the smallest bit rate. For coding the skeleton, we must also transmit a negligible overhead information consisting of one bit for signaling whether the skeleton was scanned horizontally or vertically, and four bits for the choice of one (out of up to 16) structuring element.

Table III indicates that Elias coding is the best way to code the skeleton, and this combination achieves a higher compression than "optimum" block-Huffman or runlength-Huffman coding of the original image. By "optimum" we mean a Huffman code optimized for one specific image and not based on the average statistics of a large class of images. Therefore, in practice, the bit rates achieved by block or runlength coding will be higher than the results presented in this paper. In contrast, Elias coding is a very simple code to implement, and its implementation does not depend on the source statistics, as the Huffman code does. Previous researchers, however, speculated that coding the skeleton [24] or the medial axis [22] will not perform better than boundary or runlength encoding, respectively, of the image. The discrepancy between our results favoring skeleton coding and their speculations seems to lie in the fact that their arguments were solely based on the number of skeleton points. In addition to the standard techniques for binary image coding to which skeleton coding was compared in this paper, there are also

TABLE IV  
CODING EFFICIENCY OF GLOBALLY MINIMAL SKELETON VERSUS STRUCTURING ELEMENT [ $64 \times 64$  PIXELS TEST IMAGE OF FIG. 7(a)]

Structuring Element	Max Skeleton Subset Index	Number of Skeleton Points	Number of Bits for <sup>a</sup> Skeleton Subsets	Number of Bits for <sup>a</sup> Skeleton Function
CIRCLE	3	77	568	662
SQUARE	7	61	510	548
RHOMBUS	7	149	1008	1310
BOXNE	14	49	496	498
LIN000	13	83	672	631
LIN090	8	135	782	1047
VEC000	26	63	592	618
VEC090	16	102	696	826

<sup>a</sup>Using Elias code ( $m = 3$ ).

other techniques which have recently attracted interest. One such technique is *quadtree representation and coding* of binary images (briefly reviewed in [26]). Although a comparison between skeleton and quadtree coding is beyond the scope of this paper, we point out two differences between quadtrees and skeletons. The quadtree representation of a binary image is shift-varying, whereas the morphological skeleton is shift-invariant. In addition, quadtrees are not redundant, whereas skeletons generally contain some redundancy because of the overlapping between maximal disks. Of course, the globally minimal skeleton is not redundant with respect to the skeleton points needed for exact reconstruction.

Instead of using a fixed structuring element over the whole image, we also tried to vary the structuring element by segmenting the image and adaptively finding different elements that performed better over different segments of the image. However, the results in terms of coding efficiency were not good enough to justify the above adaptation.

## V. CONCLUSIONS

Morphological set erosions and dilations can be implemented simply and fast using only parallel shifting and logical AND/ORing of the image. Using erosions and dilations, which are the prototypes of any increasing and translation-invariant system as shown in [16], and [17], and the concept of a structuring element, which generalizes the concept of distance and neighborhood, we can obtain the morphological skeleton of a binary image, which unifies many previous approaches to skeletonization. The morphological skeleton is a translation-invariant, antiextensive, idempotent set transformation which admits an inverse. Thus, a finite image can be decomposed into a finite number of skeleton subsets and exactly reconstructed from them. We developed fast algorithms for skeleton decomposition and reconstruction which reduce the original quadratic complexity to linear. By omitting some skeleton subsets, we quantified the partial reconstructions (smoothed, shrunk, or expanded versions) of the original image. Some theoretical properties of the skeleton were investigated; specifically, theorems 1 and 2 in Section III-E shed some light in the use of the skel-

eton as a set transformation that quantifies shape and size in binary images. The concepts of a globally and locally minimal skeleton were introduced and investigated. By developing appropriate fast algorithms, we found that the minimal skeleton may have more than 50 percent fewer points than the original skeleton and still guarantee exact reconstruction.

Exploiting the fact that the skeleton is thinner than the original image, we coded the skeleton subsets and the skeleton function using block-Huffman, runlength-Huffman, and Elias coding. By using each of these three methods for binary image coding, encoding of the skeleton information gave higher compression (except for block and runlength coding of the subsets) than directly coding the original image. The best method to encode the location of skeleton points was Elias coding combined with a technique that we developed called *masking*; this combination gave compression ratios of 8.3:1 (0.12 bits/pixel) for regularly shaped images [test image of Fig. 7(a)] and 11.4:1 (0.09 bits/pixel) for irregularly shaped images [sign image of Fig. 11(b)] containing blobs. Omitting a few skeleton subsets results in higher compression. For the above two images, the compression achieved by skeleton coding represents a respective 90 and 124 percent increase over optimum block coding of the original images, which gave compression ratios of 4.3:1 and 5.1:1, respectively. Similarly, the above compression of skeleton coding represents a respective 69 and 37 percent increase over optimum runlength coding of the originals, which gave compression ratios of 4.9:1 and 8.3:1, respectively. Similar results favoring the skeleton coding were obtained for images of cells containing big areas. However, the superior compression efficiency of the skeleton drops for thin images consisting mainly of lines and points without thickness, and becomes comparable to optimal block-Huffman and runlength-Huffman coding. Two other advantages of skeleton coding are the inherent ability of the skeleton for partial reconstructions and for a progressive image transmission scheme (see also [7]) that is aesthetically pleasing. Clearly, block or runlength coding offer only exact reconstruction and are not at all progressive in reconstructing the image at the receiver. Coding the skeleton subsets is faster and better suited for real-



time image transmission, whereas the skeleton function is a more compact representation for image storage. The selection of the best (for coding purposes) structuring element among a finite menu was done by choosing the element giving the skeleton with the fewest points, although this criterion is not optimum.

## APPENDIX

## PROOFS OF THE PROPERTIES AND THEOREMS IN SECTION III-E

*Proof of Property 2:* Since  $\{0\} \subseteq B$ , from (10),  $S_{n+1} \subseteq X \ominus (n+1)B^s \subseteq (X \ominus nB^s)_B$ . Since  $(X \ominus nB^s)_B \cap S_n = \emptyset$ , we have  $S_n \cap S_{n+1} = \emptyset$  for all  $n$ . Q.E.D.

*Proof of Property 3:*

a)  $[SK(X_z)] = [SK(X)]_z$  because of the translation invariance of the set operations involved.

b)  $S_n \subseteq X \ominus nB^s \subseteq X$ . Hence,  $SK(X) \subseteq X$ .

c) Known is that  $\{0\} \subseteq B \iff z \in B_z$  for all  $z$ . Let  $z \in S_n(X)$  with  $n > 0$  (assume  $N > 0$ , because otherwise  $SK(X) = X$ ). Then,  $\{z\} \oplus B = B_z \subseteq (X \ominus nB^s) \oplus B = [X \ominus (n-1)B^s]_B$ . Thus,  $B_z \cap S_k = \emptyset$  for all  $k \leq n-1$ . Also, since  $z \notin X \ominus (n+1)B^s$ , there is a point  $a \in B_z$  such that  $a \notin X \ominus nB^s$ . Hence,  $a \notin S_m$  for all  $m \geq n$ . Therefore,  $B_z$  cannot be totally contained in  $SK(X)$  for any  $z$ . Thus,  $SK(X) \ominus B^s = \emptyset$ . Hence, because of property (1b),  $SK[SK(X)] = SK(X)$ . Q.E.D.

*Lemma 1:* If  $(nB)_z$  is maximal in  $X$ , then  $z \in S_n(X)$ .

*Proof:* If  $(nB)_z$  is maximal, then  $(nB)_z \subseteq X \iff z \in X \ominus nB^s$ . Suppose that  $z \notin S_n$ . Then,  $z \in (X \ominus nB^s)_B \iff$  there exists a point  $y$  such that  $z \in B_y \subseteq X \ominus nB^s$ . Thus,  $\{z\} \oplus nB = (nB)_z \subseteq B_y \oplus nB = [(n+1)B]_y \subseteq X \ominus nB^s \oplus nB = X_{nB} \subseteq X$ . Hence,  $(nB)_z \subseteq [(n+1)B]_y \subseteq X$ . This contradicts the assumption that  $(nB)_z$  is maximal in  $X$ . Hence,  $z \in S_n$ . Q.E.D.

*Lemma 2:* If  $B$  is bounded and convex, then  $z \in S_n(X)$  (skeleton subset obtained with respect to  $B$ ) implies that  $(nB)_z$  is maximal in  $X$ .

*Proof:* Let  $z \in S_n \implies z \notin (X \ominus nB^s)_B$ . Suppose  $(nB)_z$  is not maximal in  $X$ . Then, there exists a point  $y$  and  $m > n$  such that  $(nB)_z \subseteq (mB)_y \subseteq X$ . Since  $B$  is convex and bounded (see Matheron [19, p. 21]),  $(mB)_y \ominus nB^s = [(m-n)B]_y^{nB^s}$  is equal to  $[(m-n)B]_y$ . Hence,  $z \in [(m-n)B]_y \subseteq X \ominus nB^s$  which implies that  $z \in (X \ominus nB^s)_{(m-n)B} \subseteq (X \ominus nB^s)_B$  since  $(m-n) \geq 1$  (see Serra [28, p. 54]). However, this contradicts the fact that  $z$  does not belong to  $(X \ominus nB^s)_B$ . Hence,  $(nB)_z$  is maximal in  $X$ . Q.E.D.

*Lemma 3:* If  $S_0 = S_1 = \dots = S_{k-1}(X) = \emptyset$ , then  $X_{kB} = X$ .

*Proof:* Results from (12) and (19). Q.E.D.

*Lemma 4:* If  $B$  is bounded and convex and  $X$  is convex, then  $X_{kB} = X$  implies that  $S_n(X) = \emptyset$  for  $n = 0, 1, 2, \dots, k-1$ .

*Proof:* Since  $X$  and  $B$  are convex, then  $X \ominus nB^s$  and  $(X \ominus nB^s)_B$  are convex sets. Since  $X = X_{kB}$  then  $X_{(n+1)B} = X_{nB}$  for all  $n$  such that  $0 \leq n < k$  (see [28, pp. 53, 54]). Then  $(X \ominus nB^s) \oplus nB = (X \ominus nB^s)_B \oplus nB$ . Hence, because of the dilation property of convex sets [19, p.

21], we have that  $X \ominus nB^s = (X \ominus nB^s)_B$ , and thus,  $S_n(X) = \emptyset$  for all such  $n$  such that  $0 \leq n < k$ .

Q.E.D.

By combining lemmas 1, 2, and 3, 4 we obtain, respectively, the proofs of theorems 1 and 2 of Section III-E.

## ACKNOWLEDGMENT

We wish to thank Dr. T. R. Hsing for kindly providing us with the original handsign image from which the image of Fig. 11(a) came, and for exposing us to this application of real-time binary image transmission.

## REFERENCES

- [1] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Forms*, W. Wathen-Dunn, Ed. Cambridge, MA: M.I.T. Press, 1967.
- [2] —, "Biological shape and visual sciences (Part I)," *J. Theoret. Biol.*, vol. 38, pp. 205-287, 1973.
- [3] H. Blum and R. N. Nagel, "Shape description using weighted symmetric axis features," *Pattern Recogn.*, vol. 10, pp. 167-180, 1978.
- [4] L. Calabi, "A study of the skeleton of plane figures," Parke Mathematical Labs, Carlisle, MA, Rep. SR2-60429, June 1965.
- [5] L. Calabi and W. E. Hartnett, "Shape recognition, prairie fires, convex deficiencies and skeletons," *Amer. Mathem. Monthly*, vol. 75, pp. 335-342, 1968.
- [6] P. Elias, "Predictive Coding—Part I," *IRE Trans. Inform. Theory*, vol. IT-2, pp. 16-33, Mar. 1955.
- [7] A. J. Frank, J. D. Daniels, and D. R. Unangst, "Progressive image transmission using a growth-geometry coding," *Proc. IEEE*, vol. 68, pp. 897-909, July 1980.
- [8] H. Hadwiger, *Vorlesungen uber Inhalt, Oberflache und Isoperimetrie*. Berlin, Germany: Springer-Verlag, 1957.
- [9] J. Hilditch, "Linear skeletons from square cupboards," in *Machine Intelligence*, vol. 4, B. Meltzer and D. Michie, Eds. New York: American Elsevier, 1969, pp. 403-420.
- [10] T. S. Huang, "Coding of two-tone images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1406-1424, Nov. 1977.
- [11] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc IEEE*, vol. 40, pp. 1098-1101, Sept. 1952.
- [12] J. C. Kotelly, "A mathematical model of Blum's theory of pattern recognition," Air Force Cambridge Res. Labs, Bedford, MA, Rep. 63-164, Apr. 1963.
- [13] C. Lantuejoul, "La squelettisation et son application aux mesures topologiques des mosaïques polycristallines," these de Docteur-Ingenieur, School of Mines, Paris, France, 1978.
- [14] C. Lantuejoul, "Skeletonization in quantitative metallography," in *Issues of Digital Image Processing*, R. M. Haralick and J. C. Simon, Eds. Groningen, The Netherlands: Sijthoff and Noordhoff, 1980.
- [15] J. R. Mandeville, "Novel method for analysis of printed circuit images," *IBM J. Res. Devel.*, vol. 29, no. 1, pp. 73-86, Jan. 1985.
- [16] P. A. Maragos, "A unified theory of translation-invariant systems with applications to morphological analysis and coding of images," Ph.D. dissertation, School of Elec. Eng., Georgia Inst. Technol., (also Tech. Rep. DSPL-85-1), Atlanta, GA, July 1985.
- [17] P. A. Maragos and R. W. Schaffer, "A unification of linear, median, order-statistics and morphological filters under mathematical morphology," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa FL, Mar. 1985, pp. 34.8.1-34.8.4.
- [18] —, "Morphological skeleton representation and coding of binary images," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Diego CA, Apr. 1984, pp. 29.2.1-29.2.4.
- [19] G. Matheron, *Random Sets and Integral Geometry*. New York: Wiley, 1975.
- [20] U. Montanari, "A method for obtaining skeletons using a quasi-euclidean distance," *J. Assoc. Comput. Mach.*, vol. 15, pp. 600-624, Oct. 1968.
- [21] J. C. Mott-Smith, "Medial axis transformations," in *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970.
- [22] J. C. Mott-Smith and T. Baer, "Area and volume coding of pic-



- tures," in *Picture Bandwidth Compression*, T. S. Huang and O. J. Tretiak, Eds. New York: Gordon & Breach, 1972.
- [23] Y. Nakagawa and A. Rosenfeld, "A note on the use of local min and max operations in digital picture processing," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp. 632-635, Aug. 1978.
- [24] J. L. Pfaltz and A. Rosenfeld, "Computer representation of planar regions by their skeletons," *Commun. Assoc. Comput. Mach.*, vol. 10, pp. 119-122, 125, Feb. 1967.
- [25] O. Philbrick, "Shape recognition with the medial axis transformation," in *Pictorial Pattern Recognition (Proc. Symp. Automat. Photointerpret.)*, G. C. Cheng *et al.*, Eds. Washington, DC: Thompson, 1968.
- [26] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, vol. 2. New York: Academic, 1982.
- [27] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *J. Assoc. Comput. Mach.*, vol. 13, pp. 471-494, Oct. 1966.
- [28] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.
- [29] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Chicago, IL: Univ. Illinois Press, 1949.
- [30] T. P. Sosnowski and T. R. Hsing, "Toward the conveyance of deaf sign language over public telephone networks," in *Proc. 6th Annu. Conf. Rehab. Eng.*, San Diego, CA, 1983, pp. 162-164.
- [31] S. R. Sternberg, "Parallel architectures for image processing," in *Proc. 3rd IEEE Int. Comput. Software Appl. Conf.*, Chicago, IL, 1979, pp. 712-717.
- [32] S. R. Sternberg, "Biomedical image processing," *IEEE Trans. Comput.*, vol. C-32, pp. 22-34, Jan. 1983.



**Petros A. Maragos** (S'80-M'85) was born in Kalymnos, Greece, on November 4, 1957. He received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1980, and the M.S. and Ph.D. degrees both in electrical engineering from the Georgia Institute of Technology, Atlanta, in 1982 and 1985, respectively.

From 1980 to 1985 he was a Research Assistant at the Digital Signal Processing Laboratory of the Electrical Engineering School at Georgia Tech,

where he was engaged in research on image modeling, coding, segmentation, and shape analysis. Since August 1985 he has been an Assistant Professor of Electrical Engineering in the Division of Applied Sciences, Harvard University, Cambridge, MA. His current research interests include signal and image processing, computer vision, and pattern recognition.

Dr. Maragos received the M.S. thesis research award from Sigma Xi (Georgia Tech Chapter) in 1983.



**Ronald W. Schafer** (S'62-M'67-SM'74-F'77) received the B.S.E.E. and M.S.E.E. degrees from the University of Nebraska, Lincoln, in 1961 and 1962, respectively, and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1968.

From 1968 to 1974 he was a member of the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ, where he was engaged in research on speech analysis and synthesis, digital signal processing techniques, and digital waveform coding. Since 1974 he has been on the Faculty of the Georgia Institute of Technology, Atlanta, as John O. McCarty/Audichron Professor and Regents' Professor of Electrical Engineering. He is the co-author of the widely used textbooks, *Digital Signal Processing* and *Digital Processing of Speech Signals*.

Dr. Schafer has served as an Associate Editor of the TRANSACTIONS, member of several committees, Vice-President and President of the Society, and Chairman of the 1981 ICASSP. He is a Fellow of the Acoustical Society of America and a member of Sigma Xi, Eta Kappa Nu, and Phi Kappa Phi. He was awarded the Achievement Award and the Society Award of the IEEE ASSP Society in 1979 and 1983, respectively; the 1983 IEEE Region III Outstanding Engineer Award; and he shared the 1980 Emanuel R. Piore Award with L. R. Rabiner. In 1985 he received the Class of 1934 Distinguished Professor Award at Georgia Tech.