COMPUTE
VISIO
and
IMAGE PROCESSIN

Edited by
Linda Shapiro
Department of Computer Science
University of Washington
Seattle, Washington

Azriel Rosenfeld
Center for Automation Research
University of Maryland
College Park, Maryland

ACADEMIC PRESS, INC.
Harcourt Brace Jovanovich, Publishers
Boston San Diego New York
London Sydney Tokyo Toronto

Foreword

Computer Vision, Graphics, and Image Processing has split into two
journals, *CVGIP: Image Understanding* and *CVGIP: Graphical Models
Image Processing*, starting in 1991. To allow the backlog of accepted CV
papers to be published as soon as possible, we created this volume.
papers in this publication are all reviewed papers that have gone through
regular CVGIP journal refereeing process. They cover a large variet
vision-related topics and appear as regular papers or notes as they w
have in the journal. We thank the authors of these articles for allowing u
publish them here.

Linda Shapiro and Azriel Rosen

PATTERN RECOGNITION OF BINARY IMAGE OBJECTS USING MORPHOLOGICAL SHAPE DECOMPOSITION

I. PITAS and N. D. SIDIROPOULOS

Department of Electrical Engineering,
University of Thessaloniki,
Thessaloniki 54006, Greece

A pattern recognition method based on mathematical morphology is presented. The characteristics of morphological shape decomposition are examined and an improved object-oriented decomposition technique is proposed. The performance of morphological decomposition under object translation, rotation, and scaling and its noise properties are presented. These properties are also depicted by experimental results.

I. Introduction

Shape description is a very important issue in pictorial pattern analysis and recognition. Therefore, many theories that attempt to explain different aspects of the problem exist [1, 2]. A number of them are concerned with the problem of decomposing an object X into a union of subsets $X_1, \ldots, X_n$ that are simpler and easier to describe. Though different decomposition methods may employ different decomposition strategies and seek different goals, each one must comply with the following set of desirable properties in order to maintain a high degree of generality and applicability [3].

1. It should conform with our intuitive notions of "simpler" components of a "complex" picture.
2. It should have a well-defined mathematical characterization.
3. Its characterization must be subject independent.
4. The complexity of the representation $X_1, \ldots, X_n$ must be comparable with the complexity of the original description of X.
5. It must be invariant under translation, scaling, and rotation.
6. It should allow arbitrary amounts of detail to be computed and also allow abstraction from detail.
7. It should be fast and unique.
8. It should be stable under noise.

Computer Vision and Image Processing

ISBN 0-12-033600-9

Note that not all properties are of equal importance (for example, the fifth constraint is important but not fundamental). An important problem in shape decomposition is the definition of the "simple" components. Usually they are convex polygons [1]. However, the results of the decomposition do not always correspond to the human intuitive shape representation, since it does not work only with convex polygon primitives. For example, it is not natural to try to decompose a circular disk or ring by using convex polygons. Furthermore, the results are not always unique, even if we restrict X_i to be maximally convex objects. In many cases, the derivation of the decomposition is rather tedious and the computational complexity high. These are the main disadvantages of the decomposition techniques. However, various decomposition methods have been proposed (Voronoi, Delaunay tesselations) [1]. Given the decomposition, a graphical object representation is constructed. Syntactic pattern recognition techniques are applied as the last stage of the analysis.

Our approach is to use the disk as the "simplest" object component and to analyze an image as a union of disks. Such an analysis is easily performed by morphological techniques [4] and it is shown to be unique and invariant under rotation, translation, and scaling. It has a relatively low computational complexity [5, 6]. Therefore our approach has several advantages over the known shape decomposition and skeletonization techniques. Object decomposition using mathematical morphology has also been proposed by other authors for structuring element decomposition [10] and for object recognition [17]. However, decomposition [10] can be used in an entirely different context, i.e., for the pipelining of morphological operations. Decomposition [17] uses line segments and triangles as primitives. Therefore it differs from our approach.

The outline of this paper is as follows. The morphological decomposition algorithm is described in Section II and its properties are reviewed in Section III. A discrete morphological decomposition algorithm is presented in Section IV. Section V is concerned with the description of an object based on its morphological decomposition. A pattern recognition algorithm, based on morphological decomposition, is presented in Section VI. In Section VII an improved object-oriented morphological decomposition algorithm is proposed. Experimental results are given in Section VIII.

II. Morphological Shape Decomposition

Morphological decomposition of an object into a union of disks is performed in the following way: The set of maximal inscribable disks in the object X defined over R^n that have the maximum (same) radius is found. This

set is the first cluster of the decomposition. The second set is obtained in the following way: The first set of the decomposition is subtracted from the object and the set of the maximal inscribable disks in the remainder of the object that have the maximum (same) radius is found afterward. This set is the second cluster of the decomposition. The first and second clusters are subtracted from the object and the procedure is repeated until the remainder of the object is an empty set. The whole procedure can be described in a mathematical way using mathematical morphology [4].

There exists no unique definition of the "simplicity" of the shape of an object. For our approach the "simple components" are structuring elements with radius r times the radius of the unit structuring element. If B is bounded and convex (which is valid for most practical cases), then

$$rB = B \oplus B \oplus \cdots \oplus B \qquad (r \text{ times}), \tag{1}$$

where B is a simple object, e.g., a disk. Note that in certain applications rotation invariance is of no importance and it is possible to decompose X using any other bounded and convex structuring element (e.g., a square).

From now on, a slightly modified notion of a simple set is used. A set X_i is called simple if it can be described using an expression of the form

$$X_i = L_i \oplus r_i B, \tag{2}$$

where the set L_i is an open set having zero thickness (in practice, a very thin line). We should now explain what the simple set X_i actually represents. This set is constructed if a structuring element having a radius r_i is the center of all points of the line L_i. In other words, the set that we have just described represents an object that looks like a curved ribbon. The line L_i is its "spine." The set X_i is constructed by moving the center of its "generator" $r_i B$ along its spine. If the generator is a disk then X_i is called a Blum ribbon [12] and looks like that shown in Fig. 1. The Blum ribbon is a connected topologically open set, whose radius of its maximal disk is constant. It has the following

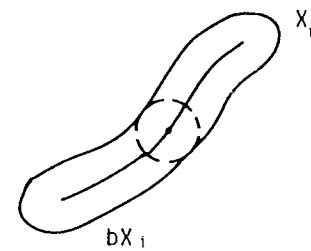


FIG. 1. A Blum ribbon defined over R^2 (bx_i denotes the border of X_i).

properties, which are direct consequences of its skeleton properties:

1. If X_i is simply connected and its border bX_i is smooth, any maximal disk contained in X_i is tangent to bX_i.
2. If X_i is a Blum ribbon, every maximal disk contained in X_i is one of the translates of r_iB.
3. If X_i is a Blum ribbon, the spine and set of generators of X_i are uniquely determined.

Their proofs can be found in [12].

Keeping in mind the properties discussed above, a recursive formulation of the morphological shape decomposition algorithm is now presented [5, 6]:

$$X_i = ((X - X_{i-1}) \ominus r_iB^s) \oplus r_iB = (X - X_{i-1})_{r_iB}. \tag{3}$$

B^s denotes the symmetric structuring element of B with respect to the origin, i.e., $B^s = \{-b, b \in B\}$. The object reconstruction follows:

$$X_i' = \bigcup_{0 \le j \le i} X_j \tag{4}$$

$$X_0' = \varnothing. \tag{5}$$

The above recursive formulation gives the decomposition $X_1, \ldots, X_n, \ldots$ of the object X as a sequence of subsets X_i, $i = 1, 2, \ldots$, where the subsets X_i are Blum ribbons, if the structuring element is a disk. The radius, r_i, of each subset X_i is the radius of the maximal inscribable objects, r_iB, in any of the connected components of $X - X_{i-1}'$ (if it has more than one connected component). The locus of the centers of the maximal inscribable objects r_iB is given by

$$L_i = (X - X_{i-1}') \ominus r_iB^s \tag{6}$$

and it consists of curved lines or a set of points or both. Consequently there exists a unique mapping between each subset (cluster) and its spine and the radius of its generator. As a result we can modify the formulation of the morphological shape decomposition algorithm and give it in the fully equivalent form

$$L_i = \left(X - \bigcup_{0 \le j \le i-1} (L_j \oplus r_jB)\right) \ominus r_iB^s \tag{7}$$

$$L_0 = \varnothing \tag{8}$$

$$L_i' = \bigcup_{0 \le j \le i} L_j. \tag{9}$$

The formulation of (3-5) is related to the formulation of (7-9) by the expressions

$$X_i = L_i \oplus r_iB \tag{10}$$

$$X_i' = \bigcup_{0 \le j \le i} L_j \oplus r_jB. \tag{11}$$

III. Properties of the Morphological Shape Decomposition

The morphological shape decomposition algorithm has a number of desirable characteristics and properties that make it suitable for shape description purposes. These properties are given below in the form of propositions. Their proof is simple and can be omitted.

PROPOSITION 1. The object decomposition, X_i', $i = 1, 2, \ldots$, given by (3-5) increases monotonically and is upper bounded by X. The sequence of radii r_i decreases monotonically.

PROPOSITION 2. The sets X_i given by (3-5) are simple according to our definition of a simple set.

PROPOSITION 3. The object decomposition $X_1, \ldots, X_i, \ldots$ is unique.

PROPOSITION 4. The object decomposition $X_1, \ldots, X_i, \ldots$ is translation and scale invariant.

PROPOSITION 5. The set decomposition $X_1, \ldots, X_i, \ldots$ is rotation invariant if the structuring element B is a disk in R^n.

PROPOSITION 6. The set decomposition $X_1, \ldots, X_i, \ldots$ is antiextensive.

PROPOSITION 7. The mapping $X \to X_i$, $i = 1, 2, \ldots$, is indempotent.

There exist points belonging to X which cannot be included in the morphological decomposition sets X_i; e.g., the tangent points of two sets X_i, X_j, $i = j$, are inaccessible by morphological decomposition. Thus at infinity a fractal set of inaccessible points is obtained. This fact poses some problems for the morphological significance of the decomposition, for the R^n case. Such a problem does not exist for the decomposition of discretized objects, because in that case all sets are both open and closed, using Euclidean topology [4]. Thus no points are inaccessible.

A very important fact that should be noted is that morphological shape decomposition is not a morphological operation. This comes from the fact that the operations described in formulation (3-5) or (7-9) involve X and therefore require global knowledge of the object. Therefore the "local knowledge" principle of morphological operations is violated. The complexity of the representation of $X_1, \ldots, X_i, \ldots$ by means of the disk loci and radii

(L_i, r_i) is less than the complexity of the original description of X as a set of points, at least for objects which do not contain sharp contours or fine details.

The results obtained from the morphological shape decomposition algorithm form a sequence of objects, whose size is decreasing monotonically. Therefore it is possible to allow abstraction from detail by truncating the output sequence at an arbitary chosen index k and neglecting the rest of the output $X_{k+1}, \ldots$.

The noise properties of the morphological decomposition cannot be studied easily. Usually, the noise properties of a shape description method are closely connected to its continuity properties [4, 14]. However, morphological decomposition has weak continuity properties [6]. This suggests that, when contaminated by noise, binary objects X may have a different morphological decomposition. Thus noise prefiltering is essential in many applications. Such noise filtering schemes may be the median filter and the close–opening and open–closing filters [14, 18, 19].

IV. Discrete Morphological Shape Decomposition Algorithm

Until now we have dealt with the problem of shape decomposition defined over R^n. However, in most practical cases we work with digitized shapes and images. Therefore we have to define the morphological shape decomposition for shapes over the Euclidean grid Z^n. Let a subset X of Z^n represent a discrete binary image. We further assume that X is nonempty and bounded. The problem of morphological decomposition is to decompose X into a union of simple subsets $X_1, \ldots, X_n$. The number n of these subsets is finite for a bounded set X. Both X and the unit structuring element B are defined over the discrete Euclidean grid. A collection of discrete structuring elements, defined in Z^2, is presented in Fig. 2. The discrete morphological shape decomposition algorithm utilizes CIRCLE as its unit structuring element but any other convex element, such as those shown in Fig. 2, may also be used for data compression/transmission purposes, where rotation invariance is of no importance.

We define a discrete simple set X_i as

$$X_i = L_i \oplus n_i B, \qquad n = 1, 2, 3, \ldots,$$

where L_i is a subset of Z^n that has zero thickness (it consists of curved lines or isolated points or both). On the basis of these notions we proceed by giving a formulation that describes the morphological shape decomposition algorithm over the discrete Euclidean grid:

$$X_i = ((X - X'_{i-1}) \ominus n_i B^s) \oplus n_i B. \tag{12}$$

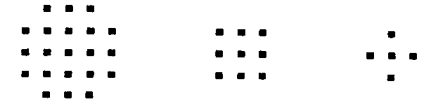


FIG. 2. A collection of discrete and convex structuring elements defined over Z^2.

The object reconstruction is as follows:

$$X'_i = \bigcup_{0 \le j \le i} X_j$$
$$X'_0 = \varnothing, \tag{13}$$

where n_i symbolizes the radius of the maximal inscribable disk with maximum radius, $n_i B$, in the set $X - X'_{i-1}$, and it is an integer. In the discrete version of the algorithm there exists a stopping condition

$$(X - X'_n) \ominus B^s = \varnothing. \tag{14}$$

In accordance with the alternative formulation that was introduced in the continuous case, its discrete equivalent based on the notion of the discrete simple set is now presented,

$$L_i = \left(X - \bigcup_{0 \le j \le i-1} (L_j \oplus n_j B)\right) \ominus n_i B^s \tag{15}$$

$$L_0 = \varnothing \tag{16}$$

$$L'_i = \bigcup_{0 \le j \le i} L_j, \tag{17}$$

where L_i represents the loci of the maximal inscribable disks with maximum radii $n_i B$ in the set $X - X'_{i-1}$.

The discrete morphological shape decomposition algorithm has the following properties:

1. X'_n is bounded by X. In certain cases $X - X'_n \neq \varnothing$. Therefore some fine details having size less than B may be missed.
2. The objects X_i are simple according to our definition of the simple set.
3. The object decomposition is unique, scale and translation invariant, antiextensive, and indempotent.

The rotation invariance is no longer valid. There exist two reasons for this. The first one is the very nature of the discrete grid Z^n, which does not support set rotation at arbitrary angles. The second problem is that there is no exact

equivalent of a disk in Z^n. However, there exist several discrete approximations to a disk. In the rectangular grid, the structuring element CIRCLE is a good approximation to a disk in Z^2. If a good approximation of the disk is used as the structuring element B then the algorithm exhibits a fairly good robustness to rotation.

The morphological decomposition algorithm is mainly composed of set erosions/dilations. Therefore its computational complexity is mainly determined by the efficiency of the method by which these morpholigical operators are implemented. In specialized morphological processors the time efficiency of the decomposition is very good indeed. For a given implementation it has been proven that the computational complexity of the algorithm approximates a linear function of the square root of the object X area [6].

The complexity of the algorithm is also a function of the structuring element area. The bigger the unit structuring element, the less the computational complexity. For a complete discussion the reader is referred to [6]. A block-diagram form of the basic decomposition/reconstruction algorithm is shown in Fig. 3. The procedure erosion() repeatedly erodes the input set and stops one iteration before it is found empty. The procedure dilation() dilates its input set N times, where N is the input cluster radius. The object reconstruction is based on a scheme proposed in [3] and originally intended for skeleton reconstruction. It is shown in a block-diagram form in Fig. 4.

V. Object Representation Based on Morphological Shape Decomposition

The morphological shape decomposition gives the ability to represent an object as a union of simple objects, $X_1, \ldots, X_k$. Each of these objects is

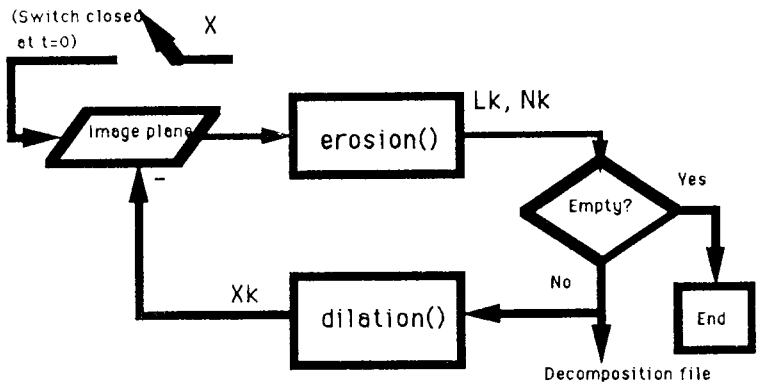


FIG. 3. A basic block-diagram form of the morphological shape decomposition algorithm.

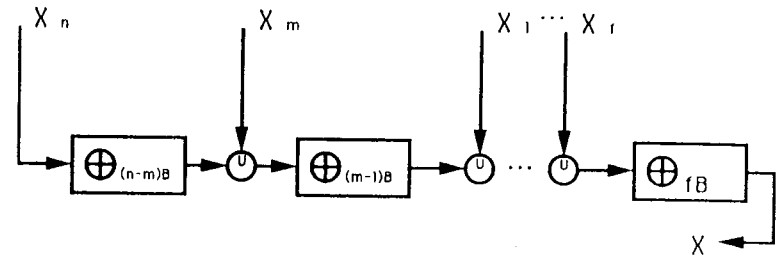


FIG. 4. A basic block-diagram form of a fast algorithm that reconstructs X using its decomposition data.

completely described by the locus L_i (its spine) and the radius r_i of the corresponding disks. In certain cases the locus L_i is composed of disconnected subsets $L_{i1}, \ldots, L_{i\lambda}$ that correspond to disconnected subsets $X_{i1}, \ldots, X_{i\lambda}$ of the simple object X_i. Therefore a representation for an object X has the form

$$X = \{X_1, \ldots, X_k\} \tag{18}$$

$$X_i = \{X_{i1}, \ldots, X_{i\lambda}\} \tag{19}$$

$$X_i = \{(L_{i1}, r_i), \ldots, (L_{i\lambda}, r_i)\} \tag{20}$$

$$L_i = \bigcup_{1 \le j \le \lambda} L_{ij}, \qquad (L_{ij} \oplus r_i B) \cap (L_{im} \oplus r_i B) = \varnothing, j \neq m. \tag{21}$$

An alternative approach that can be employed is as follows. The mass centers M_{ij}, of each simple set X_{ij} are computed and their distance from the center of mass, M_{11}, of the biggest single simple object X_{11} is calculated. The object X_{11} can be thought of as the "body" of the object X, whereas each X_{ij}, $i, j \neq 1$, can be thought of as being a "limb" that protrudes from the body of the object. Subsequently

$$M'_{ij} = M_{ij} - M_{11}. \tag{22}$$

For the spine of each simple object X_{ij} we compute the distance of each of its pixels P_{ij} from the local mass center X_{ij}:

$$P'_{ij} = P_{ij} - M_{ij}. \tag{23}$$

Therefore the simple object X_{ij} can be represented

$$X_{ij} = \{M'_{ij}, P'_{ij} \quad \text{for every} \quad P_{ij} \in L_{ij}\} \tag{24}$$

by having M_{11} as a base reference. The use of relations (18) (24) has the advantage that it can be seen as a star-like graph, whose center node is the

body X_{11} of the object X. Therefore, we have obtained a hierarchicial data structure that is suitable for many tasks commonly found in image analysis and processing. Such a graph representation can be used for object recognition by any known graph matching technique. The whole procedure gives substantial data compression because it reduces the magnitude of the required coordinates. An example of a star-like graph is given in Fig. 5.

VI. Description of a Pattern Recognition Algorithm Based on Morphological Shape Decomposition

In this section an elementary pattern recognition algorithm based on morphological shape decomposition is described. It is given in a natural language form in Fig. 6 and in a block form in Fig. 7. This scheme accepts an object decomposition as its input and attempts to decide upon its degree of similarity with a given reference object. It is assumed that the reference object has previously been decomposed and that its resulting decomposition is also available. Both decompositions are assumed to be stored on secondary storage devices.

The algorithm described in Figs. 6 and 7 implements a comparison scheme between a test object and a reference object. Repetitive calls of this procedure with various reference objects are used in order to identify a given object between a given set of possible reference objects.

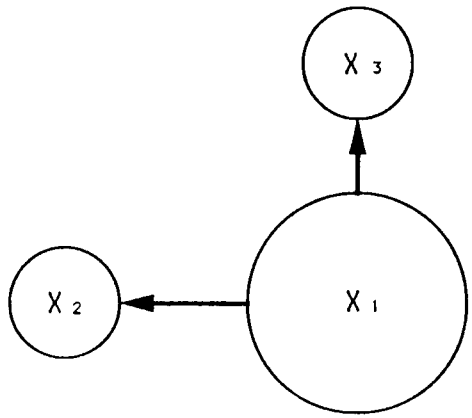


FIG. 5. An example of a star-like graph. X_1 is the "body" node whereas X_2 and X_3 are "limb" nodes.


```
MORPHOLOGICAL PATTERN RECOGNITION ALGORITHM

STEP 0 Initialization;
        current_rot_angle=0;
        r=(0,0);
        #pins=0;

STEP 1 Load Temp matrix with next cluster of reference
object. Dilate it N times ( N: reference cluster radii ). If the
reference clusters previously received were magnified then
magnify the current cluster too. Add the set contained in Temp
matrix to the one contained in Reference matrix.
STEP 2 Load Temp matrix with next cluster of test object.
Dilate it N times ( N: test cluster radii ). Translate the
resulting set by r. Rotate it according to current_rot_angle. If
the test clusters previously received were magnified then magnify
this one too. Add the resulting set to the one contained in Image
matrix.
STEP 3 Check which set has the largest area. If the two areas
are equal ( say within a 5% ) go to step 5.
STEP 4 Magnify the other one and go to step 3.
STEP 5 Find the mass center ( MC ) of each cluster and
translate the set contained in Image matrix by r where r =
MC(Image) - MC(Reference). Go to step 7.
STEP 6 Rotate the set contained in Image matrix by a given,
predefined step. Update current_rot_angle. If current_rot_angle =
360 degrees stop the execution and signal that the two objects
are different.
STEP 7 If the set contained in Image matrix is the same with
the one contained in Reference matrix ( say within a 85% ) then:
#pins = #pins + number of non-connected subsets in the
current reference cluster.
        else:
Go to step 6.
STEP 8 If all clusters perceived as being significant have
been received then stop the execution and signal that the two
objects are identical. Supply the calling procedure with
information about the test object orientation, scaling and
translation parameters. Else ( execute steps 1 and 2 and
afterwards: If #pins <=2 go to step 7 else: ( If the set
contained in Image is the same with the one contained in
Reference go to step 8 else stop the execution and signal that
the objects are different. ) )
```

FIG. 6. Basic pattern recognition algorithm.

An interesting aspect of the algorithm is the use of the variable denoted with #pins. It is used in order to determine whether the angle of relative rotation between our test object and our reference has been identified. If this has been done, no further rotation is necessary and all subsequent clusters are expected to arrive in given relative positions. The criterion used is that if three simple nonconnected subsets have been identified, the test object is rotated current_rot_angle degrees with respect to our reference, because three points are sufficient for the determination of the position of a solid object. The

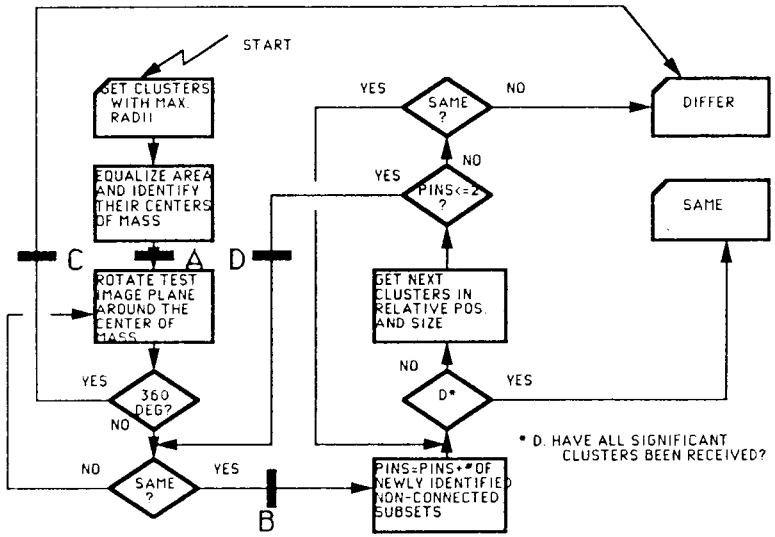


FIG. 7. A block diagram form of the basic pattern recognition algorithm.

variable # pins counts the number of simple nonconnected subsets that have already been identified.

The object recognition algorithm that has been described is a simple and adequate approach to the problem of recognizing an object which may be rotated by ϑ degrees with respect to the reference object. However, it has a disadvantage: ϑ may take values only from a given set of possible angles depending on the current_rot_angle increment that is used (e.g., $\vartheta = -3\pi/4$, $-\pi/2$, $\pi/4$, $\pi/2$, $3\pi/4$ for an increment of $\pi/4$). Furthermore, the object rotation angle must be one of the above with great accuracy, otherwise the recognition scheme fails. This disadvantage can be compensated (up to a certain extent) by defining a small angle increment. This solution may prove costly in terms of time complexity because the smaller the step the more set rotation necessary. Thus a heavy load of floating-point operations is involved. The reason for this disadvantage lies in the way we search for the correct angle. In the elementary recognition scheme described above a sequential search of the rotation angle is used, by incrementing it stepwise and by comparing the rotated test object with the reference object. Thus several rotations must be performed before the correct rotation angle is found. If we could devise a method to make a reasonable initial assumption about the relative rotation angle and therefore restrict ourselves to a limited

"pivot" area, it could be possible to bypass all unnecessary set rotations and make the algorithm much more time efficient.

An interesting approach to this problem is the following. An expression can be computed that gives the inertia exhibited by the object as a function of its second-order moments and with respect to a certain axis which runs by the objects center of mass and defines an angle ϑ with the x axis. This expression can be minimized with respect to ϑ and the object orientation (axis of least inertia) can be found. This is the direction in which the object is elongated. If the object is too "round-faced" for a reasonable assumption about its elongation to be made, we may obtain unpredictable results. If this is done for the test object and for the reference object the test object can be rotated using the angle difference and their orientation axes can be effectively identified. Once the centers of mass and the orientation axes have been identified, only two object positions are possible: the one that resulted from the previous rotation and the one that results if the object is rotated by 180°. Subsequently if the test object is not found to be the same with the reference object in one of these two positions, it can be discarded and all further rotations can be avoided.

An extensive coverage of the problem of computing expressions for first and second-order moments of objects defined in the Euclidean two-dimensional continuous grid can be found in the literature [15]. A suitable measure of the orientation of an object, the full procedure for devising an expression for object inertia $E(\vartheta)$ as a function of ϑ, and a formula for computing a ϑ^* such that $E(\vartheta^*) = \min$ can be found in [15].

The second-order moments on a continuous grid are defined as

$$a = \iint_{I'} (x')^2 b(x, y)\, dx'\, dy' \tag{25}$$

$$b = 2 \iint_{I'} (x'y') b(x, y)\, dx'\, dy' \tag{26}$$

$$c = \iint_{I'} (y')^2 b(x, y)\, dx'\, dy' \tag{27}$$

$$x' = x - \bar{x} \tag{28}$$

$$y' = y - \bar{y}, \tag{29}$$

where x, y denote grid coordinates and $\bar{x}, \bar{y}$ the mass center coordinates. The expression that gives the inertia integral E as a function of ϑ is

$$E = [(a + c) - (a - c)\cos 2\vartheta - b \sin 2\vartheta]\, 2. \tag{30}$$

By differentiating with respect to ϑ and setting the result equal to zero we obtain

$$\tan 2\vartheta = b/(a - c) \tag{31}$$

except for the case when $b = 0$ and $a = c$. Therefore we have

$$\sin 2\vartheta = \pm b/\sqrt{b^2 + (a - c)^2} \tag{32}$$

$$\cos 2\vartheta = \pm(a - c)/\sqrt{b^2 + (a - c)^2}. \tag{33}$$

Of the two solutions the one with the positive sign in the expressions for $\sin 2\vartheta$ and $\cos 2\vartheta$ is the one that leads us to the desirable minimum for the E, whereas the one with the negative sign gives us E maximum. This can be shown by examining the sign of the second derivative of E with respect to ϑ. If both $b = 0$ and $a = c$ are found, the object in consideration is too symmetric to allow for an orientation axis to be defined. This can happen if the object is a fairly good approximation of a disk or ring.

Equations (25) (27) have the following form on a discrete grid:

$$a = \sum_x \sum_y (x - \mathbf{x})^2 b(x, y) \tag{34}$$

$$b = \sum_x \sum_y (x - \mathbf{x})(y - \mathbf{y}) b(x, y) \tag{35}$$

$$c = \sum_x \sum_y (y - \mathbf{y})^2 b(x, y). \tag{36}$$

In practice a, b, c are computed and the computation of $\sin 2\vartheta$, $\cos 2\theta$ follows using Eqs. (32) and (33), both with positive sign. Thus

$$\varphi = \pi - \arctan(\sin 2\vartheta/\cos 2\vartheta)/2. \tag{37}$$

The two coordinate systems and the angles ϑ and φ for a typical object are presented in Fig. 8. The test object is rotated by φ_d degrees, where $\varphi_d = \varphi_t - \varphi_r$. φ_t is the orientation angle of the test object and φ_r is the orientation angle of the reference object. If the test object is recognized at this angle, the goal has been obtained. If not, a rotation by 180° is necessary. If this step fails too, it is concluded that our test object is different from our reference object. In practice we may allow a limited pivoting around the center positions in order to compensate for inaccuracies in the computation of orientation due to the restrictions posed by the nature of the discrete rectangular grid and the image formation process. An iterative method similar to the Newton-Raphson algorithm may be used.

The modified part of our original object recognition algorithm is presented in Fig. 9. If we substitute this block in place of the block contained between

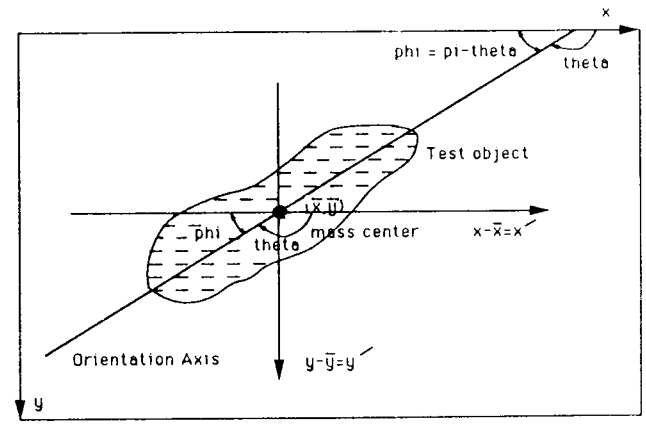


FIG. 8. Definition of the notion of object orientation.

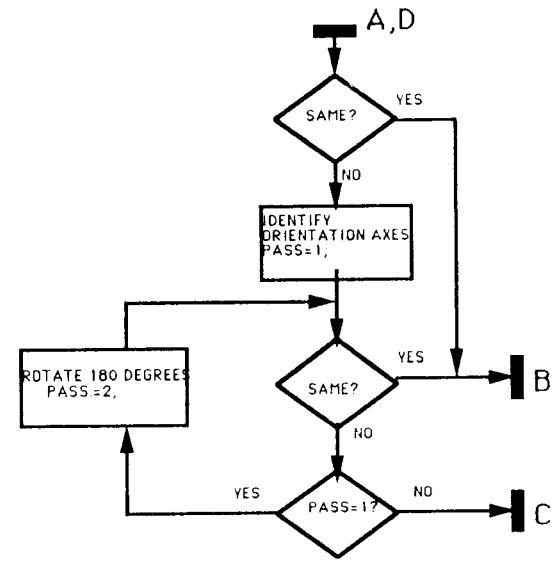


FIG. 9. Modified search procedure for the improved, two-step, random angle pattern recognition algorithm. This block is substituted in place of the block contained between the points A, B, C, D, in Fig. 7 to yield the improved algorithm.

points A, B, C, D in the body of the algorithm we get the improved object recognition scheme described above. This approach can recognize an object regardless of its orientation angle, which in fact may take any possible value $[0 \cdots 2\pi)$. Furthermore the mean time complexity of the algorithm is independent of the actual orientation angle.

VII. Internal and External Morphological Shape Decomposition

In the following, another improved version of the morphological shape decomposition algorithm is proposed. The basic idea is the following: Until now, set X was decomposed to a union of simple sets $X_1, \ldots, X_n$. A more general approach would be to decompose X using either the maximal inscribable disk or the minimal disk $M(X)$ that circumvents X. We call the first approach internal decomposition and the second approach external decomposition. In external decomposition the set $M(X) - X$ is decomposed externally at the second step. This procedure can be repeated recursively. The advantage of the general approach over the initial decomposition algorithm is clearly shown in the following example. Suppose that a disk having a small circular hole must be decomposed. Using the internal decomposition scheme it is described with a large number of smaller disks. Alternatively, it can be described as one full disk minus a smaller disk that describes the hole (external decomposition). This description leads to great data compression. Furthermore, it is much more natural and compatible with our intuitive understanding of the object. The above object is certainly not perceived as a union of maximal inscribable disks. In order to outline the basic notion behind this discussion the reconstruction of the object FACE, shown in Fig. 10a, is presented using the internal and the external decomposition methods. FACE is decomposed externally into a minimal disk that is centered in its mass center and fully covers it minus a number of smaller disks that describe the eye holes. Its reconstruction using this approach is given in Fig. 10b. Its reconstruction using the internal decomposition algorithm is given in Fig. 10c. The data compression ratio of each method can be computed. A brute-force transmission of FACE would require 512 bytes (64×64 pixels). Its internal decomposition requires approximately 530 bytes and thus leads to a 3.5% data expansion! On the other hand the alternative external decomposition needs only 43 bytes and gives a compression ratio of 1:12.

It should be noted that internal decomposition preserves the holes of the object with size less than that of the structuring element. These details are missed when external decomposition is used. On the other hand, the latter approach preserves peninsulas that protrude from the body of the object and are smaller than the structuring element.

A critical question regarding the choice of the best decomposition method for an arbitrary object arises. In general, a decision as to whether to decompose a specific object subset internally (that is, using maximal inscribable disks) or externally (using a minimal disk that circumvents the specific subset) has to be made during each step of the decomposition algorithm. The following procedure can be applied. Suppose that a subset X_i of X must be decomposed. Its center of mass is computed and, using successive dilations of this point with CIRCLE element, the minimal resulting disk $M(X_i)$ that fully overlaps X_i is computed. Note that this disk is a fair approximation to the disk that circumvents the object X_i only if X_i exhibits a fairly normal centroidal profile and feature holes that are comparatively small and evenly distributed. An initial prediction about the size of this minimal disk can be made by measuring the area $A(X_i)$ of the subset X_i and starting the iterations with a disk that has a radius of $r = \sqrt{A(X_i)/\pi}$. After the minimal disk $M(X_i)$ is found, the set difference $M(X_i) - X_i$ is computed and its area $A(M(X_i) - X_i)$ is compared with the area of X_i. The external approach is well suited if the area $A(M(X_i) - X_i)$ is small. If this area is smaller than $A(X_i)$, X_i is decomposed externally; otherwise it is decomposed internally. This scheme works in many cases but it may fail in some others, such as in the case of a ring-type object. This procedure can be repeated iteratively for each decomposition step. However, in many cases this procedure can be used only at the first step, i.e., for $X_i = X$. Internal decomposition can be used for the set $M(X_i) - X$ and its subsets Y_i, $i = 1, \ldots, m$. In this case the object reconstruction has the form

$$X = M(X) - Y_1 - Y_2 - \cdots - Y_m. \tag{38}$$

VIII. Examples and Conclusions

In order to evaluate the performance of the algorithms described in the previous sections we have designed and built an experimental Morphological Digital Image Processing System (called MDIPS). We do attempt to give a qualitative picture of the behavior of the various algorithms that have been proposed. All images used in the examples have a 64×64-pixel resolution. The algorithm has some preprocessing steps. If the input image is contaminated with salt-and-pepper noise a closing filter with structuring element RHOMBUS, a close-opening filter, an open-closing filter, or a median filter is used before we try to decompose the object [14, 18, 19]. After each iteration of the decomposition algorithm a thinning [4] operation is performed on the

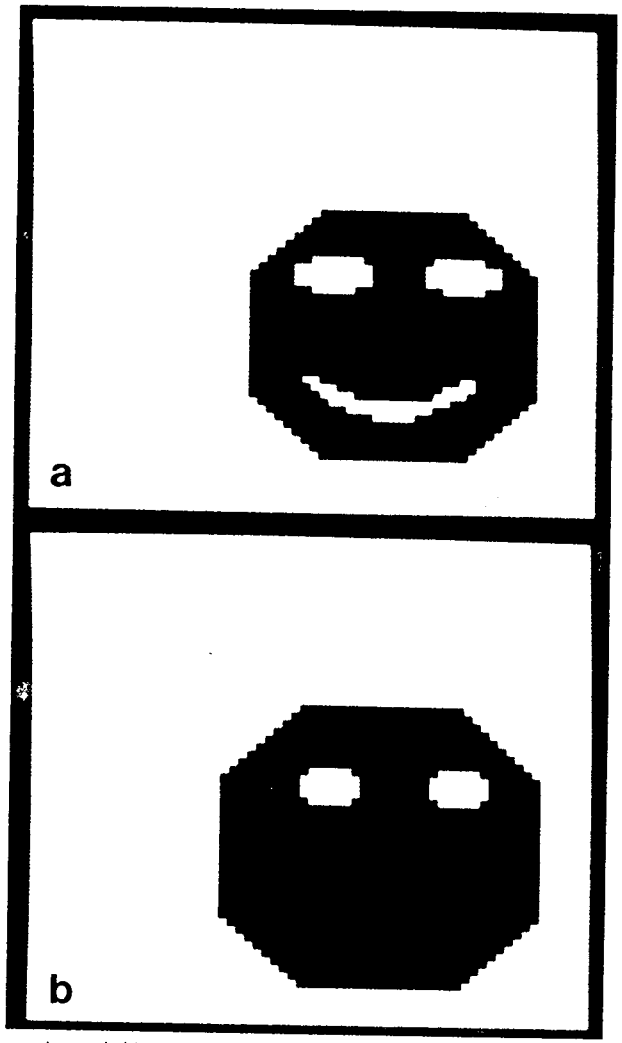


FIG. 10. (a) A sample binary object; (b) its reconstruction using the external decomposition approach; and (c) its reconstruction using the internal decomposition approach.

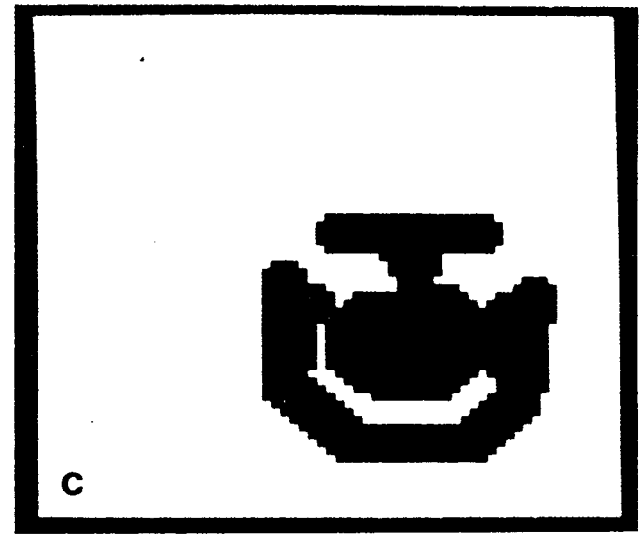


FIG. 10. — *Continued*

resulting set in order to give clusters that comply with the definition of a Blum Ribbon in the discrete rectangular grid [12]. The rotation invariance properties are shown in Fig. 11. The object of Fig. 11d is a rotated version of Fig. 11a. However, both objects have similar decompositions and they can be recognized by our system. The improved two-step random angle pattern recognition algorithm invariably recognizes the object no matter what its rotation angle. A small problem arises with orientation differences in the order of 5° to 2°. Due to the nature of the discrete rectangular grid, random angle set rotation is not supported and this leads to a subsequent rounding of the pixel positions. As a result a limited pivoting around the predicted object position is necessary in order to recognize the object. This problem may sometimes be bypassed by giving a loose identification criterion. A better but costly approach is to use an iterative method similar to the well-known Newton–Raphson algorithm in order to compute the correct angle.

We note that the decomposition algorithm produces results that are translation invariant. Object recognition in this case is an easy task. Object scaling is a more difficult situation. In order for its reconstruction to remain invariant the scaling factor must be such that the subsequent magnification reduction of the object results in the addition/subtraction of a strip that has a width which is a multiple of the unit structuring element radii. The smaller the unit structuring element the better the scaling-invariance characteristics.

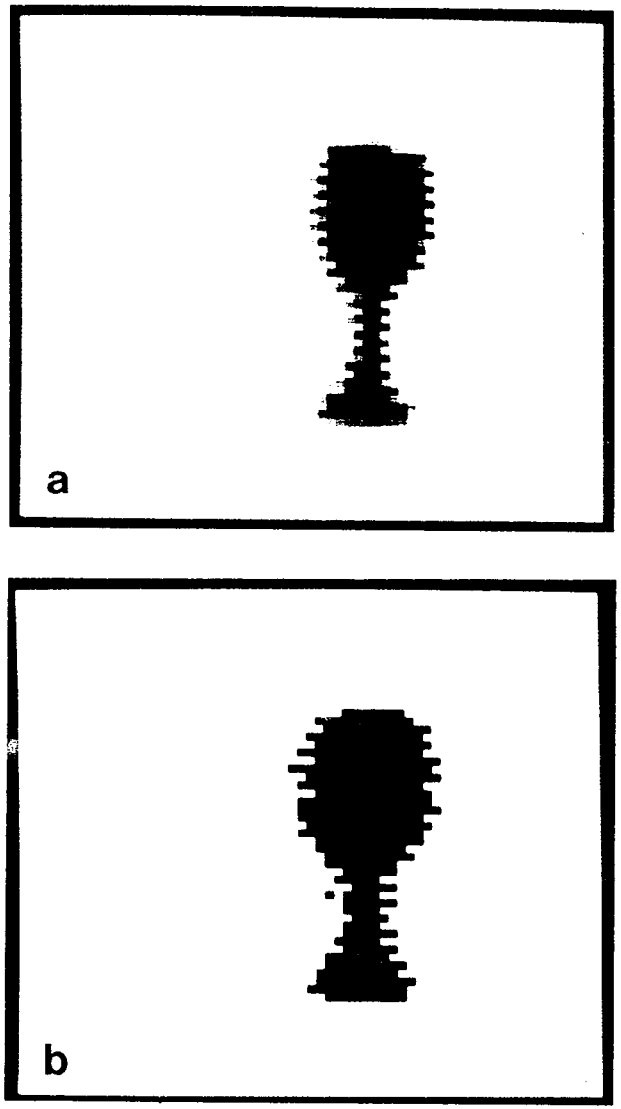


FIG. 11. (a) A sample gray-level picture containing a foreground object; (b) a thresholded version of (a); (c) its reconstruction using the internal decomposition method; (d) a rotated version of (a); (e) its reconstruction using the internal decomposition method; (f) a magnified and translated version of (a); and (g) its reconstruction using the internal decomposition method.

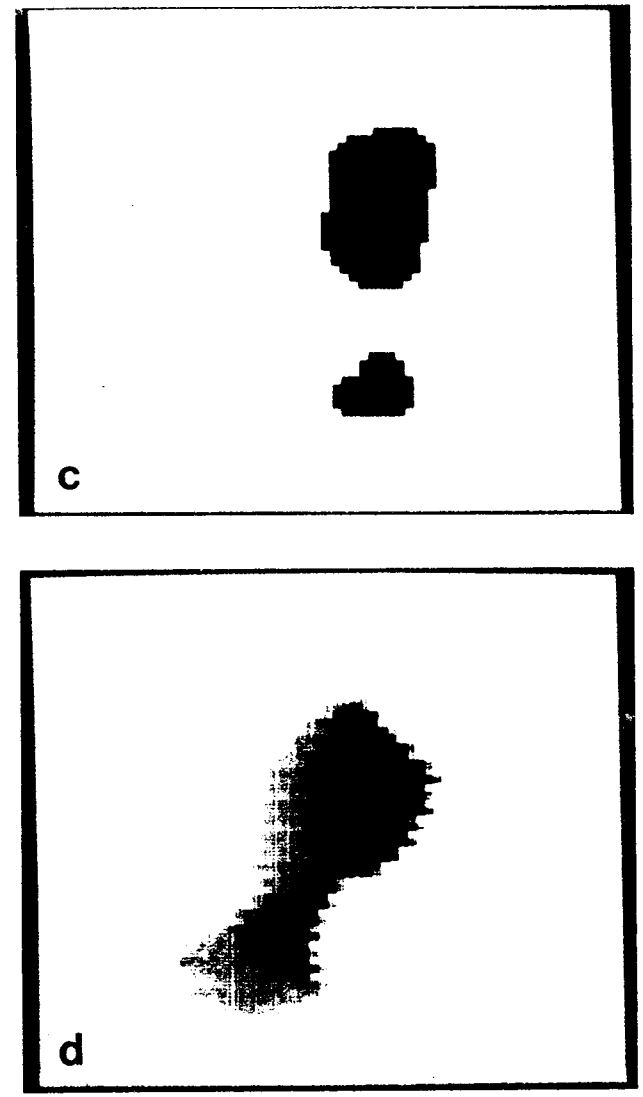
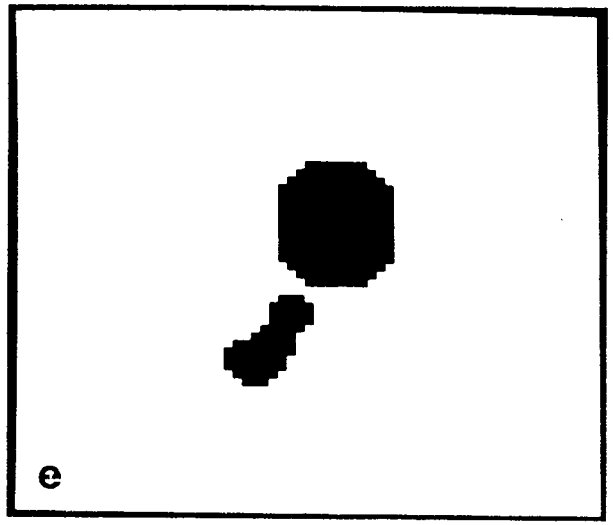


FIG. 11. —*Continued*



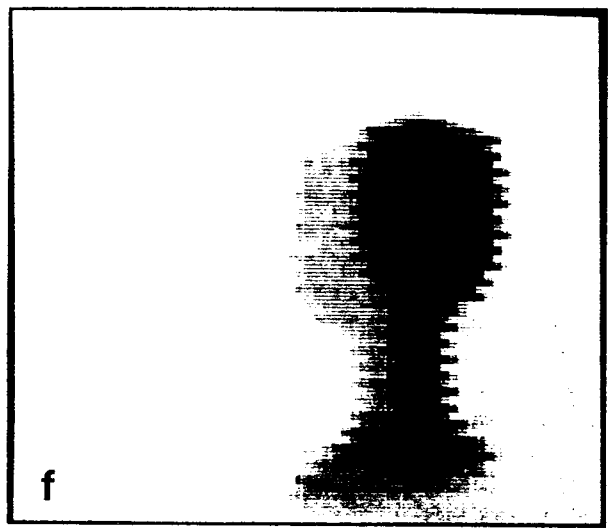


FIG. 11. — *Continued*

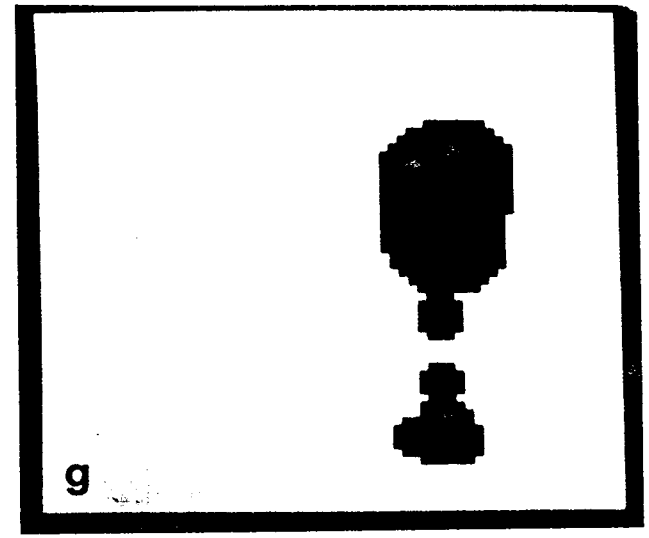


FIG. 11. — *Continued*

In the continuous case an absolute scale invariance is expected. An examp of the scaling invariance properties is also shown in Fig. 11. The object of Fi 11f is a scaled version of Fig. 11a. However, its decomposition in Fig. 11g not affected greatly by scaling.

The noise characteristics of the morphological decomposition/reconstru tion scheme are now discussed. A uniform space distribution salt-and-pepp noise model is well suited for this purpose, because it is the kind of noise th is expected to be found in most applications. This noise may be introduc due to errors in the channel decoding process. We note that the decompo tion algorithm by itself acts as a kind of nonlinear filter because during t first iteration an erosion operation is carried out and "salt" noise eliminated (up to a certain extent). The system uses an open-closing or clos opening filter or a 3 × 3 median filter for salt-and-pepper noise removal [1 18, 19]. A measure of the salt-and-pepper noise probability of occurrence the percentage of image pixels that are expected to be contaminated. If t probability is below or equals 1% then the system works well and obj decomposition/reconstruction remains practically invariant. When the curance probability is in the range from 1 to 9% the results vary from ve good to poor but are average to fair when compared to the originals. If t noise probability is greater than 10% the success of the morphologic recognition drops dramatically. An example of the noise performance of t proposed scheme is shown in Fig. 12. The presence of 9% salt-and-pepp

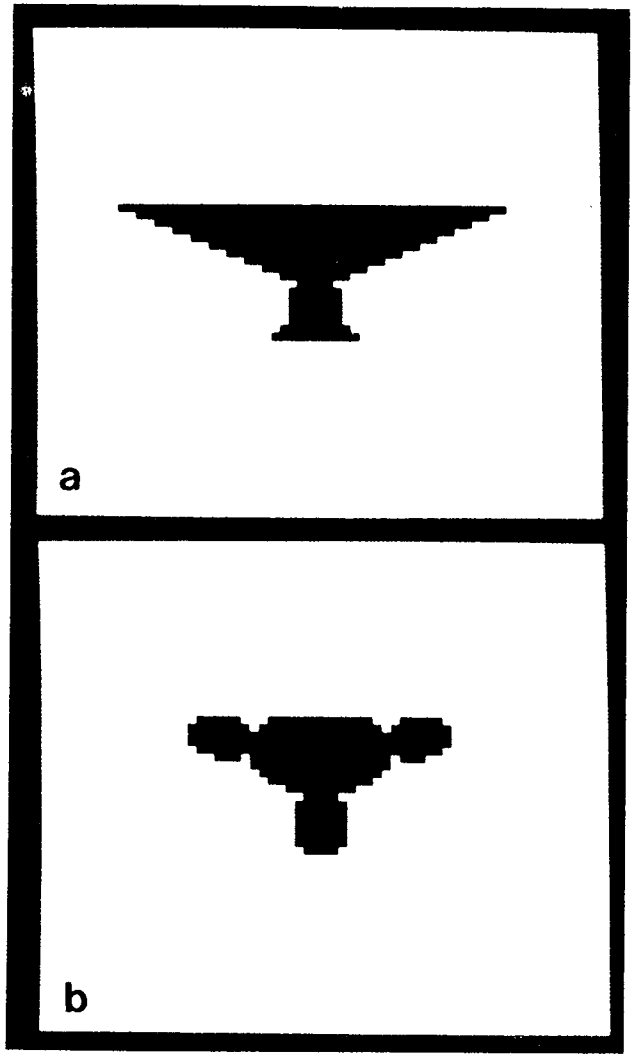


FIG. 12. (a) A sample binary object; (b) its reconstruction using the internal decomposition method; (c) noisy version of (a) contaminated with salt-and-pepper noise having an occurance probability of 9"„; and (d) its reconstruction using the internal decomposition algorithm.

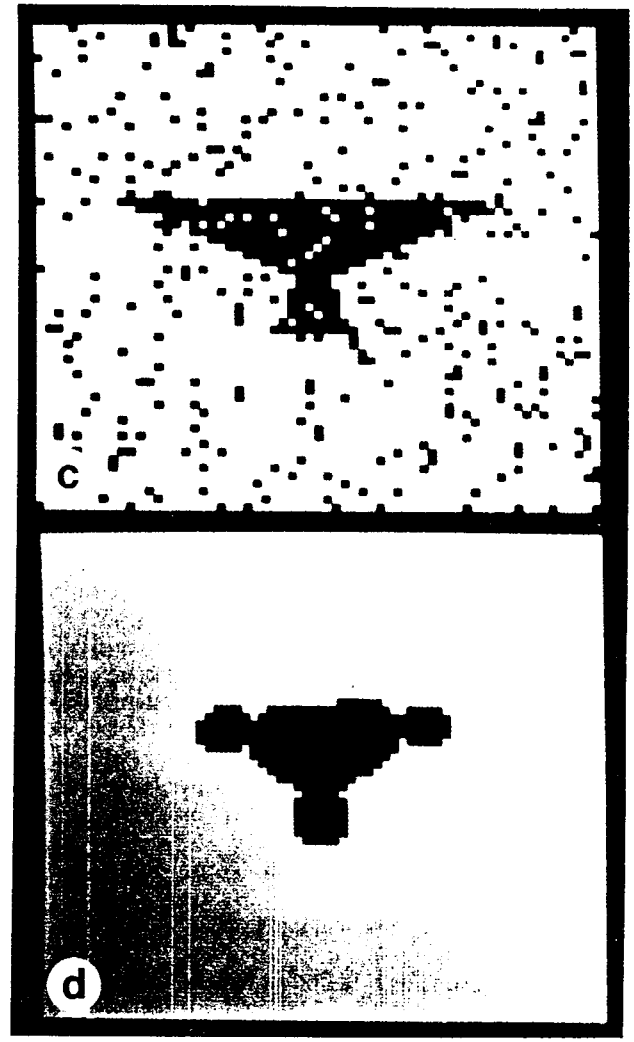


FIG. 12. *Continued*

noise does not significantly deteriorate the results of the decomposition. This is partially due to the preprocessing steps of MDIPS.

The morphological decomposition and recognition is relatively fast. The decomposition, reconstruction, and recognition of the object in Fig. 12a (64 × 64 pixels) require 16, 5, and 15 s CPU and I/O time on an EPSON PC AX computer using a C complier, a hard disk, and a 80287 mathematical coprocessor. It should be kept in mind that these figures may be dramatically reduced when the system is running on a dedicated machine and morphological operators are implemented in parallel.

From a data compression point of view MDIPS averages a compression ratio of 1 : 4 if the last cluster (unit radii) is retained. If it is not retained, a compression ratio of 1 : 15 is obtained. In general this last cluster describes the fine details of the object and is not needed in the case where the goal is a coarse description. The reader should also refer to the earlier discussion concerning the benefits of using a signed decomposition algorithm.

The algorithms that have been proposed can be easily modified for use in the Euclidean three-dimensional grid. In this case the structuring element can be a sphere, a cube, or any other bounded and convex solid. In the case of the external decomposition scheme the notion of a minimal disk can be replaced by that of a minimal sphere. A decision of whether to decompose a solid object internally or externally can be based on volume measurements. In a broad perspective these techniques are related to a more general approach of three-dimensional shape description, known as Constructive Solid Geometry [16]. In fact these techniques can be thought of as special cases of CSG descriptions if we pose some restrictions on the structuring elements and on the set operators used. Internal morphological shape decomposition uses one structuring element and one set operator (set union). External morphological shape decomposition uses one structuring element and two set operators (set union and set difference). Recent research efforts have tried to couple CSG and mathematical morphology [20, 21]. In general, the use of several structuring elements and morphological and set operators can give better, object-oriented, economical descriptions.

REFERENCES

1. M. D. Levine, *Vision in Man and Machine*, McGraw Hill, New York, 1985.
2. T. Pavlides, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD, 1982.
3. P. Maragos and R. W. Schafer, Morphological skeleton representation and coding of binary images, *IEEE Trans. Acoust. Speech, Signal Proc.* **ASSP-34**, 5, October 1986, 1228–1244.
4. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, San Diego, CA, 1982.

5. I. Pitas and A. N. Venetsanopoulos, Shape decomposition by mathematical morphology, in *Proceedings, First International Conference on Computer Vision, 1987*, pp. 621–625.
6. I. Pitas and A. N. Venetsanopoulos, Morphological shape decomposition, *IEEE Trans. Pattern Anal. Mach. Intelligence* **PAMI-12**, 1, January 1990, 38–46.
7. T. Pavlides, *Structural Pattern Recognition*, Spring-Verlag, New York Berlin, 1977.
8. T. Pavlides, A review of algorithms for shape analysis, *Comput. Graphics Image Process.* 7, 2, April 1978, 243–258.
9. T. Pavlides, Structural pattern recognition: Primitives and juxtaposition relations, in *Frontiers of Pattern Recognition*, (S. Watanabe, Ed.), Academic Press, San Diego, CA, 1972.
10. X. Zhuang and R. M. Haralick, Morphological structuring element decomposition, *Comput. Vision Graphics Image Process.* **35**, 1986, 370–382.
11. M. D. Levine, P. B. Noble, and Y. M. Youssef, Understanding blood cell motion, *Comput. Graphics Image Process.* **21**, 1, January 1983, 185–209.
12. A. Rosenfeld, Axial representation of shape, *Comput. Vision Graphics Image Process.* **33**, 1986, 156–173.
13. R. M. Haralick, S. R. Sternberg and X. Zhuang, Image analysis using mathematical morphology, *IEEE Trans. Pattern Anal. Mach. Intelligence* **PAMI-9**, 4, July 1987, 532–551.
14. I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic, Dordrecht, 1990.
15. B. Klaus and P. Horn, *Robot Vision*, MIT Press McGraw Hill, Cambridge New York, 1986.
16. D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ, 1982.
17. Y. Zhao and R. M. Haralick, Binary shape recognition based on automatic morphological shape decomposition, in *Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing, Glasgow, 1989*, pp. 1691–1694.
18. P. Maragos and R. W. Schafer, Morphological filters. II. Their relations to median, order-statistic, and stack filters, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-35**, 8, August 1987, 1170–1184.
19. R. L. Stevenson and G. R. Arce, Morphological filters: Statistics and further syntactic properties, *IEEE Tran. Circuits Syst.* **CAS-34**, November 1987, 1292–1305.
20. P. Ghosh, A mathematical model for shape description using Minkowski operators, *Comput. Vision Graphics Image Process.*, **44**, 1988, 239–269.
21. I. Pitas and A. N. Venetsanopoulos, Morphological shape representation, *Pattern Recognition*, under review.