A Morphological Parallel Algorithm for Classifying Binary Image Contours

NEUCIMAR JERÔNIMO LEITE*

DCC/IMECC/UNICAMP
Caixa Postal 6065
13081-970 Campinas - SP
neucimar@dcc.unicamp.br

Abstract. In this paper we analyze some problems concerned with connectivity on binary images. More precisely, by means of Mathematical Morphology we describe an algorithm which detects and classifies contours of connected sets as inner or outer contours. The method discussed here is quite parallel and can be directly implemented in a SIMD massively parallel computer.

1 Introduction

Mathematical Morphology [Banon 94], [Coster 89], [Dougherty 93], [Haralick 87], [Serra 92], has been developed to meet the requirements in modern digital image processing, and can be applied to a large number of both practical and theoretical applications.

The interpretation of an image consists mainly of two steps: 1) the segmentation of the significant objects presented in the image; 2) the quantization of these objects by means of iconic or symbolic values for their classification. As opposed to linear methods [Rosenfeld 82] which treat an image as unity, Mathematical Morphology considers an image as grouping of objects and in this sense it constitutes a new and efficient basis for its analysis.

Informally, the Mathematical Morphology operations consist in comparing an unknown image with some shapes named structuring elements. We are supposed to control the characteristics of these structuring elements by means of boolean relations such as unions or intersections, where both image and structuring elements can be seen as sets. Hence, a binary image can be represented by the 1 valued points of the set, and a gray level image can be described by a two variable function f(x,y) associated to its umbra. The umbra represents the points in the (x, y, z) space, where z $\leq$ f(x,y) [Serra 92],[Haralick 87].

Due to the discrete nature of the different image sensors (scanners, satellite, radars etc), the images are generally represented in the $\mathcal{Z}^2$-space or in a subset of it (a sequence of images is naturally represented in a $\mathcal{Z}^3$-space).

This work is concerned with Mathematical Morphology and connectivity problems in binary images. A binary images can be obtained by thresholding a gray level image and be represented, in the computer, by a MxN array, where each pixel located at position (x,y) has value 1 or 0 depending on whether it is black or white. The interest in studying binary image is motivated by the large amount of applications in areas such as document analysis, optical character recognition, fingerprint analysis, traffic control, robotics, and so on.

In order to understand the binary images features one should investigate the problems concerned with geometrical and topological properties closely related to connectivity [Qian 92]. In this work we analyze some of such aspects, more precisely, by means of Mathematical Morphology we detect the contours of connected components and classify them as *inner* or *outer* contours. As we will see elsewhere, this classification can be used to extract topological informations such as the Euler-Poincaré or genus of a binary image, the number of connected components, detect the holes of multiply connected components, and so on.

If an image component has only outer contour, then obviously it is a simple connected component; if it has inner and outer contours then it is a multiply connected component [Qian 92]. Further, an image with more than one outer contour has more than one component, and as the inner contours are associated to the holes of a component, we can use them to detect all the holes of an image. Our algorithm provides such informations and can be directly executed in parallel architectures well-suited for image processing applications, such as CLIP, DAP and MPP [Fountain 87].

Some previous works concerned with binary connectivity have been presented in the literature. In [Rosenfeld 82] some algorithms describe how to de-

*The work reported in this paper was partially financed by a grant from the GEOTEC-PROTEM project of CNPq-Brazil.

termine and track borders in an image. In [Duff 86] global propagation has been introduced to count the number of connected components in a CLIP-like massively parallel computer. In [Rosenfeld 90] connectedness of a set **S** with its complement $\mathbf{S}^c$ and their boundaries have been investigated.

The problems we will deal with are close related to a recent work presented in [Qian 92]. The method in [Qian 92] is eminently sequential, where tracking border operations are executed in predefined raster scans, and the contours are detected and classified one by one according to the order in which they appear. This method needs the assumption that the distance between any two connected components is at least three pixels wide, and special configurations, which lead to some ambiguities in the classification process, must be taken into account.

Our method is quite parallel and can be easily implemented in a SIMD machine. It is based on Mathematical Morphology and no assumption about the distance between connected components and special configurations needs to be made.

This paper is organized as follows. Section 2 introduces some notions of Digital Topology and Mathematical Morphology. Section 3 describes the algorithm and illustrates the method. Conclusions are drawn in section 4.

2 Preliminaries

We introduce now some basic concepts related to this work. For more details see, for example, [Rosenfeld 82], [Kong 89], [Serra 92]).

2.1 Digital Topology

Let $\mathbf{\Sigma}$ be a binary picture and **S** a subset of $\mathbf{\Sigma}$. A point $p = (x, y)$ of this digital picture has four horizontal and vertical neighbors, namely

$$(x-1, y),\ (x, y-1),\ (x, y+1),\ (x+1, y)$$

These points are the 4-neighbors of p and together with its four diagonal neighbors, namely

$$(x-1, y-1),\ (x-1, y+1),\ (x+1, y-1),\ (x+1, y+1),$$

constitute the 8-neighbors of p.

A 4-path (8-path) of length n from a point p to a point q in $\mathbf{\Sigma}$ is a sequence of points $p = p_0, p_1,\ldots,p_n = q$ such that p_i is a 4-neighbor (8-neighbor) of p_{i-1}, $1 \leq i \leq n$. A set **S** of $\mathbf{\Sigma}$ is said to be 4-connected (8-connected) if, for any two considered points of **S**, there is a 4-path (8-path) joining them.

Let $\mathbf{S}^c$ denote the complement of **S** relative to $\mathbf{\Sigma}$. The connected component of $\mathbf{S}^c$ which contains the border of $\mathbf{\Sigma}$ (namely, its top and bottom rows, and its left and right columns), is called the *background* of **S**. All other components of $\mathbf{S}^c$, if any, are called holes in **S**. If **S** is connected and has no holes, it is called *simply connected*; if it is connected but has holes, it is called *multiply connected.*

It is important to note that to avoid the connectivity paradox in dealing with connectedness, it turns out to be desirable to use opposite type of connectedness for **S** and $\mathbf{S}^c$. For example, if we use 4- for **S**, then we should use 8- for $\mathbf{S}^c$, and vice versa [Rosenfeld 82]. This will allow us to treat borders as closed curves.

If **S** and **T** are subsets of $\mathbf{\Sigma}$, then **S** surrounds **T** if any path from a point of **T** to the border of $\mathbf{\Sigma}$ must meet **S**. Evidently, the background of **T** surrounds itself, and **T** surrounds any hole in **T**.

If a component **S** has only *outer* contour (points of **S** closer to its background or to the holes of another component surrounding it), then obviously it is simply connected; if it has outer and *inner* contours (points of **S** closer to its hole), then **S** is multiply connected. Furthermore, if a binary image has more than one outer contour then it has more than one connected component. Fig. 3 shows such an image in which an outer contour is completely surrounded by an inner contour.

2.2 Mathematical Morphology

We introduce briefly, in this section, some basic notions of discrete and binary Mathematical Morphology.

Let $f(x)$ be our discrete and binary image $\mathbf{\Sigma}$, i.e, $\{f(x) \in 0,1\colon x \in \mathcal{Z}^2\}$, where $\mathcal{Z}$ denotes the set of integers. This image can be denoted by a set **X** given by $\mathbf{X} = \{x \in \mathcal{Z}^2 : f(x) = 1\}$. The complement $\mathbf{X}^c$ of a set **X** is given by $\mathbf{X}^c = \{x \in \mathcal{Z}^2 : f(x) = 0\}$. The symmetric $\check{\mathbf{X}}$ of a set **X** is denoted by $\check{\mathbf{X}} = \{-x\colon x \in \mathbf{X}\}$. The translation $\mathbf{X}_u$ of a set **X** by vector u is given by $\mathbf{X}_u = \{z\colon z = x+u,\ x \in \mathbf{X}\}$. The set difference $\mathbf{X} \backslash \mathbf{Y}$ of sets **X** and **Y** is given by $\mathbf{X} \backslash \mathbf{Y} = \mathbf{X} \cap \mathbf{Y}^c$.

Let the structuring element **B** be a finite set, i.e, its cardinality is finite. The first relation we can define between sets **X** and **B** is dilation

$$\mathbf{X} \oplus \mathbf{B} = \{u : \mathbf{X} \cap \mathbf{B}_u \neq \emptyset\} = \bigcup_{b \in B} \mathbf{X}_b \qquad (1)$$

The erosion of a set **X** by a structuring element **B** is denoted by

$$\mathbf{X} \ominus \mathbf{B} = \{u : \mathbf{B}_u \subseteq \mathbf{X}\} = \bigcap_{b \in \check{B}} \mathbf{X}_b \qquad (2)$$

Morphologically, the *interior contours*, $\mathbf{I}_c$, of a set $\mathbf{X}$ is a subset of $\mathbf{X}$ denoted by

$$\mathbf{I}_c = \mathbf{X} \backslash (\mathbf{X} \ominus \mathbf{B}), \quad (3)$$

and the *exterior contour*, $\mathbf{E}_c$, of a set $\mathbf{X}$ is a subset of $\mathbf{X}^c$ given by

$$\mathbf{E}_c = (\mathbf{X} \oplus \mathbf{B}) \backslash \mathbf{X}, \quad (4)$$

where for a square grid the structuring element $\mathbf{B}$ can be the elementary 4- or 8-connected set as in Fig. 1. Note that if $\mathbf{B}$ is the 4-connected (8-connected) set then it defines a 8-connected (4-connected) contour (see Fig. 3 for 8-connected contours).

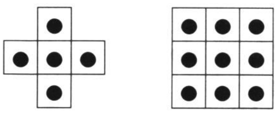


Figure 1: (a) 4-connected, and (b) 8-connected structuring elements.

2.2.1 Geodesic operations

Let x, y be two arbitrary points of $\mathbf{X}$. We define a geodesic arc of extremity xy as the lower bound of the lengths of the arcs in $\mathbf{X}$ ending at points x and y. Let us define the number $d_X(x,y)$ as this lower bound, if such arcs exist, and $+\infty$, if not. The function d_X is a distance function called geodesic distance [Lantuejoul 84], [Lantuejoul 80].

As stated in [Lantuejoul 84], all classical morphological transformations (dilation, erosion, skeletonizations etc) can be defined in the metric space $(\mathbf{X}, d_X)$. For example, if $\mathbf{Y} \subset \mathbf{X}$, points x of $\mathbf{X}$ such that $\mathbf{B}_X(x, \lambda)$ hits $\mathbf{Y}$ constitute the λ-dilated set from $\mathbf{Y}$ in $\mathbf{X}$ denoted by

$$D_X^\lambda(\mathbf{Y}) = \{x \in \mathbf{X} : \mathbf{B}_X(x, \lambda) \cap \mathbf{Y} \neq \emptyset\} \quad (5)$$

where $\mathbf{B}_X(x, \lambda) = \{x \in \mathbf{X}\colon d_X(x, y) \leq \lambda\}$

Practically, the digital geodesic dilation of size n, in the $\mathcal{Z}^2$-space, can be obtained from n iterations of a geodesic dilation of size 1, denoted by

$$D_X^1(\mathbf{Y}) = (\mathbf{Y} \oplus \mathbf{B}) \cap \mathbf{X}, \quad (6)$$

and hence

$$D_X^n(\mathbf{Y}) = \underbrace{D_X^1(D_X^1(...D_X^1(\mathbf{Y})))}_{n\ times} \quad (7)$$

The structuring element of size 1 can be the 4- or 8-connected set of Fig. 1.

In the same way we can define the λ-eroded set from $\mathbf{Y}$ in $\mathbf{X}$ as the points x of $\mathbf{X}$ such that $\mathbf{B}_X(x, \lambda)$ is totally included within $\mathbf{Y}$.

$$E_X^\lambda(\mathbf{Y}) = \{x \in \mathbf{X} : \mathbf{B}_x(x, \lambda) \subseteq \mathbf{Y}\} \quad (8)$$

As a simple example, let us consider the problem of eliminating the components which are partially included in an image to be analyzed [Coster 89] (Fig. 2). Henceforth we will suppose that black components have value 1 and white components have value 0.

If $\delta\mathbf{\Sigma}$ represents the subset of points belonging to the edge of the image, then the new set $\mathbf{X'}$ representing the components which are totally included in $\mathbf{X}$ can be defined by

$$\mathbf{X}' = \mathbf{X} \backslash D_X^\infty(\delta\mathbf{\Sigma}) \quad (9)$$

Informally, $D^\infty(\cdot)$ means execution of operation D until stabilization or idempotence.

3 Classifying contours in binary images

Since the geodesic operations are closely related to the notion of connectivity, we will consider them for detecting and classifying the contours of the simply or multiply connected components as inner or outer contours. These operations are based only on local properties of the pixels neighborhood, and can be directly executed in a SIMD massively parallel computer.

3.1 The algorithm

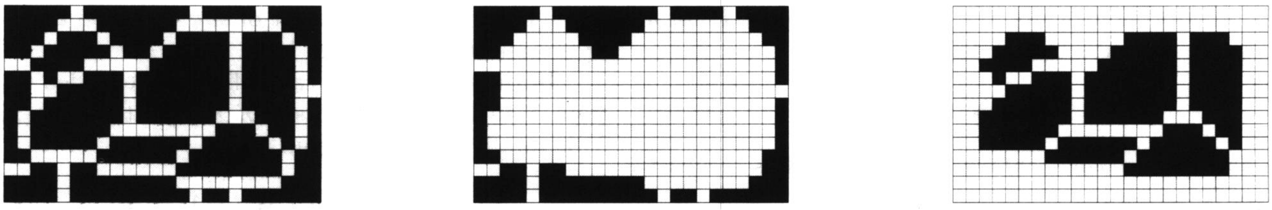
The method defines iteratively the subspaces of the image where the geodesic operations are defined. The contours of these subspaces are detected and classified by successive alternations of the outer and inner classes until a stopping criterion is met, as explained in this section.

Let, as before, $\delta\mathbf{X}$ be a marker set corresponding to the edge points of a binary image $\mathbf{\Sigma}$ (Fig. 3). We can detect the outermost contours $\mathbf{O}$ of the components $\mathbf{X}$, in $\mathbf{\Sigma}$, from the geodesic dilation of $\delta\mathbf{X}$ in $\mathbf{X}^c$. These outermost contours can be defined by (see Fig. 4.b).

$$\mathbf{Y} = D_{X^c}^\infty(\delta\mathbf{\Sigma}) \quad (10)$$

$$\mathbf{O} = (\mathbf{Y} \oplus \mathbf{B}) \backslash \mathbf{Y} \quad (11)$$

The set $\mathbf{Y}^c$ denotes the points which are interior to the current outer or inner detected contour, and defines the upper bound space for the computation of the next contours (Fig. 4.b).



(a) Image $\mathbf{\Sigma}$ containing partially included components (black pixels).

(b) Geodesic dilation of the set $\delta\mathbf{\Sigma}$ in $\mathbf{X}$.

(c) Set $\mathbf{X}$' of the components totally included in $\mathbf{\Sigma}$.

Figure 2: Eliminating partially included components of an image.

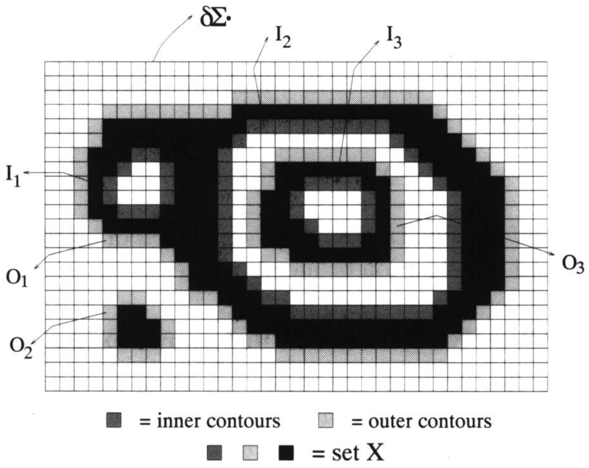


Figure 3: Set $\mathbf{X}$ and its inner and outer contours.

As in a SIMD programming model, where all pixels are transformed in parallel, sets such as $\mathbf{Y}$ and $\mathbf{O}$ can be seen as *memory planes*, as the same size as the original image, in which we store the points of the transformed set.

The set representing the holes of $\mathbf{X}$ can be obtained by the difference $\mathbf{X}^c \backslash \mathbf{Y}$ as illustrated in Fig. 4.c. The outer contours of this set correspond to the inner contours of the components considered previously. It means that to classify the contours of $\mathbf{X}$ we need only execute the above operations iteratively, and alternate the outer inner classes at each step (see Fig. 4). Here, the outer contours are stored in memory plane $\mathbf{O}$ and the inner contours in memory plane $\mathbf{I}$.

It is easy to see that the set $\mathbf{\Sigma}$ will be empty after the detection of the last contours of the image, i.e., when there will be no more connected component $\mathbf{X}$ in $\mathbf{\Sigma}$ (see Figs. 4.k and 4.l). This state will be used to denote the transitive closure of the algorithm.

Morphologically, the algorithm can be described as follows.

step = 0 ; /* initialize variables */
$\mathbf{O}, \mathbf{I} = \emptyset$

while not ZERO($\mathbf{\Sigma}$) do
  step = step + 1
  $\mathbf{Y} = D^{\infty}_{X^c}(\delta\mathbf{\Sigma})$

/* consider inner or outer class */
  if((step MOD 2) $\neq$ 0) then

/* step odd: detect interior outer contours */
    $\mathbf{O} = [(\mathbf{Y} \oplus \mathbf{B}) \backslash \mathbf{Y}] \cup \mathbf{O}$
  else

/* step even: detect interior inner contours */
    $\mathbf{I} = (\mathbf{Y} \backslash (\mathbf{Y} \ominus \mathbf{B})] \cup \mathbf{I}$

  end_if

/* define the next computation space */
$\mathbf{X} = \mathbf{X}^c \backslash \mathbf{Y}$
end_while

Fig. 4 illustrates the algorithm for the set $\mathbf{X}$ in Fig. 3. The detected contours are 8-connected and they belong to the 1 valued components of the image (interior contours). Exterior and 4-connected contours can be defined in a similar way.

The function ZERO($\mathbf{\Sigma}$) verifies if all the pixels of $\mathbf{\Sigma}$ are equal to 0, namely,

ZERO($\mathbf{\Sigma}$) := if $\mathbf{\Sigma} \equiv \emptyset$ then 1 else 0 end_if

This function can be executed within time $O(\log N)$, for a NxN image [Klette 80]. All the above operations are easily implemented in a parallel architecture such as CLIP, MPP and DAP. These processor arrays have some hardware flexibilities which allow the execution of local operations in a simple and fast way [Fountain 87].

We can speedup the execution of the algorithm by detecting the contours, in time t, from the contours detected in time $t-1$, i.e., not always from the geodesic dilation of $\delta\mathbf{\Sigma}$. This consideration leads to the following procedure.

```
step = 0
O', I', δZ = ∅                    /* initialize variables */
Y = D∞_{X^c}(δΣ)

while not ZERO(Y) do
   step = step + 1
   Y = D∞_{X^c}(δΣ)

/* consider inner or outer class */
   if((step MOD 2) ≠ 0) then

/* define the current interior outer contour */
         O' = [(Y⊕B)\Y]\I'

/* define the corresponding exterior outer contour */
         δZ = Y\(Y⊖B)\δZ

         Y = D∞_X(δO')

/* actualize the outer contour set */
         O = O' ∪ O

   else

/* define the current interior inner contour */
         I' = [(Y\(Y⊖B)]\O'

/* define the corresponding exterior inner contour */
         δZ = [(Y⊕B)\Y]\δZ

/* which is the new set to be geodesically dilated */
         Y = D∞_{X^c}(δZ)

/* actualize the inner contour set */
         I' = I' ∪ I

   end_if
end_while
```

The sets $\mathbf{O'}$ and $\mathbf{I'}$ are temporary memory planes containing the current outer and inner contours, respectively. The set $\delta\mathbf{Z}$ denotes the new marker used in the geodesic dilation for the computation of the next contour. This set corresponds to a hole of $\mathbf{\Sigma}$ connecting an inner to an outer contour. Again, all the contours defined here are interior contours, i.e., they belong to the 1 valued sets of the image $\mathbf{\Sigma}$.

Since the memory plane $\mathbf{Y}$ will be empty when the set difference $\mathbf{O'}$ or $\mathbf{I'}$ will also be empty (the geodesic dilation of an empty set is an empty set), which means that the last outer or inner contour was found, we can use the state of this memory plane to denote the end of the algorithm.

The above procedure detects and classifies in parallel the contours of a binary image. Based on this classification we can easily extract some informations of binary sets. For instance, all the multiply connected components of an image, $\mathbf{X}_h$, if any, are defined by

$$\mathbf{X}_h = D_X^{\infty}(\mathbf{I}) \tag{12}$$

where $\mathbf{I}$ is the set of the inner contours, as defined before. Given a known inner contour $\mathbf{I'}$, the corresponding outer contour $\mathbf{O'}$ of this set is simply

$$\mathbf{O}' = D_X^{\infty}(\mathbf{I}')\backslash\mathbf{O}^c \tag{13}$$

where $\mathbf{O}$ is the set of the outer contours. In the same way, the set $\mathbf{I}_O$ of inner contours which surround outer contours, if any, is

$$\mathbf{I}_O = D_{X^c}^{\infty}(\mathbf{O}\oplus\mathbf{B})\backslash\mathbf{I}^c \tag{14}$$

The Euler number or Euler-Poincaré characteristic [Rosenfeld 82], E, can be defined by

$$E = N(\mathbf{O}) - N(\mathbf{I}) \tag{15}$$

where N(*) denotes the number of the components in the set (memory plane) *. An example of computing N(*) in a parallel machine can be found in [Levialdi 72].

4 Conclusion

Mathematical Morphology provides simple and efficient tools for solving problems in image processing and analysis. We have considered some geodesic operations, closely related to the notion of connectivity, to detect and classify contours of connected components as inner or outer contours. Since the morphological operations presented in this work are eminently parallel, and based on local properties of the pixels neighborhood, the method discussed here can be directly implemented in a SIMD massively parallel architecture, such as the processor arrays, well-suited for low level image processing.

References

[Banon 94] G. J. F. Banon & J. Barrera, *Bases da Morfologia Matemática para a Análise de Imagens Binárias*, in IX Escola de Computação, 24 a 31 de Julho, Recife, PE, Brazil, 1994.

[Coster 89] M. Coster & J. L. Chermant, *Précis d'Analise d'Images*, Presses du CNRS, Paris, 1989.

[Dougherty 93] E. R. Dougherty, *Mathematical Morphology in Image Processing*, Marcel Dekker, New York, 1993.

[Duff 86] M. J. B. Duff & T. J. Fountain, *Cellular Logic Image Processing*, Academic Press, London, 1986.

[Fountain 87] T. J. Fountain, *Processor Arrays* - Architecture and Applications, Academic Press, London, 1987.

[Haralick 87] R. M. Haralick et alli, *"Image Analysis Using Mathematical Morphology"*, IEEE Trans. on Pattern Analys. and Mach. Intell., vol. 9, n. 4, July 1987, pp. 532-550.

[Klette 80] R. Klette, *"Parallel Operations on Binary Images"*, Computer Graphics and Image Processing 14, 1980, pp. 145-158.

[Kong 89] T. Y. Kong & A. Rosenfeld, *"Digital Topology: Introduction and Survey"*, Computer Vision, Graphics and Image Processing, v. 48, 1989, pp. 357-393.

[Lantuejoul 80] C. Lantuejoul & S. Beucher, *"On the Use of Geodesic Metric in Image Analysis"*, Journal of Microscopy, v. 121, Pt 1, Jan. 1980, pp. 39-49.

[Lantuejoul 84] C. Lantuejoul & F. Maisonneuve, *"Geodesic Methods in Quantitative Image Analysis"*, Pattern Recognition, v. 17, n. 2, 1984, pp. 177-187.

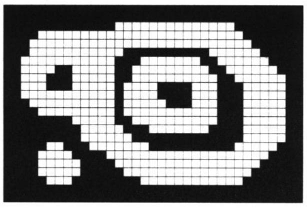
[Levialdi 72] S. Levialdi, *"On Shrinking Binary Picture Patterns"*, Commun. of the ACM, v. 15, n. 1, pp. 7-10, 1972.

[Qian 92] K. Qian & P. Bhattacharya, *"Determining Holes and Connectivity in Binary Images"*, Comput. & Graphics, vol. 16, n. 3, 1992, pp. 283-288.

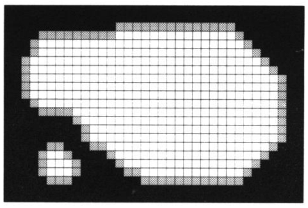
[Rosenfeld 82] A. Rosenfeld & A. C. Kak, *Digital Picture Processing*, vol. I et II, Academic Press, Orlando, Florida, 1982.

[Rosenfeld 90] A. Rosenfeld & T. Y. Kong, *"Connectedness of a Set, its Complement and their Common Boundary"*, in Technical Report CS-TR-2435, Univ. of Maryland, CAR, 1990.

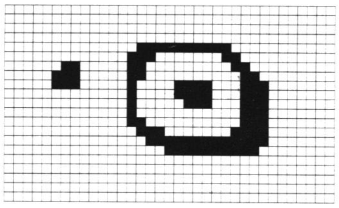
[Serra 92] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, second printing, London, 1992.



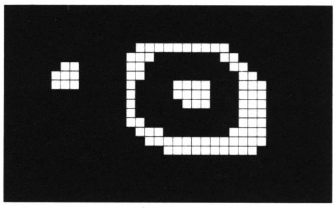
(a) Complement $\mathbf{X}^c$ of the set $\mathbf{X}$ in Fig. 3.



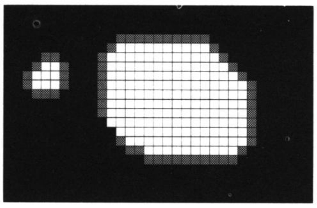
(b) Set $\mathbf{Y}=\mathrm{D}^{\infty}_{X^c}(\delta\mathbf{\Sigma})$ and definition of $\mathbf{O}_1$, $\mathbf{O}_2=(\mathbf{Y}\oplus\mathbf{B})\backslash\mathbf{Y}$.



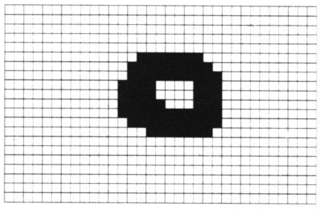
(c) New set $\mathbf{X}=\mathbf{X}^c\backslash\mathbf{Y}$.



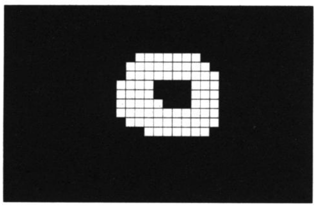
(d) Complement $\mathbf{X}^c$ of $\mathbf{X}$.



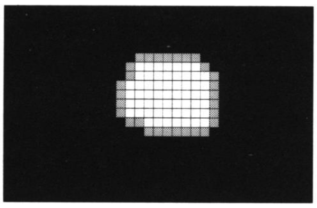
(e) New set $\mathbf{Y}=\mathrm{D}^{\infty}_{X^c}(\delta\mathbf{\Sigma})$ and definition of $\mathbf{I}_1$, $\mathbf{I}_2=\mathbf{Y}\backslash(\mathbf{Y}\ominus\mathbf{B})$.



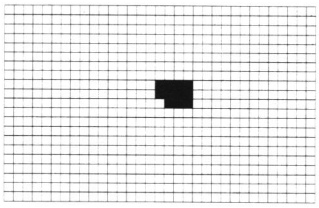
(f) New set $\mathbf{X}=\mathbf{X}^c\backslash\mathbf{Y}$.



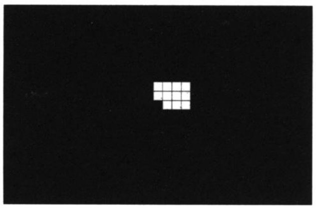
(g) Complement $\mathbf{X}^c$ of $\mathbf{X}$.



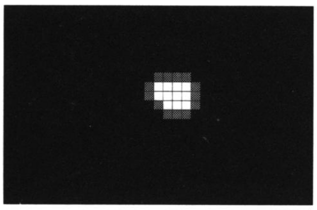
(h) New set $\mathbf{Y}=\mathrm{D}^{\infty}_{X^c}(\delta\mathbf{\Sigma})$ and definition of $\mathbf{O}_3=(\mathbf{Y}\oplus\mathbf{B})\backslash\mathbf{Y}$.



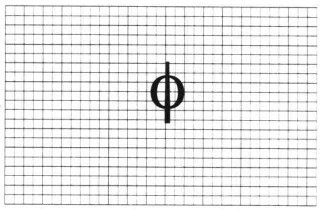
(i) New set $\mathbf{X}=\mathbf{X}^c\backslash\mathbf{Y}$.



(j) Complement $\mathbf{X}^c$ of $\mathbf{X}$.



(k) New set $\mathbf{Y}=\mathrm{D}^{\infty}_{X^c}(\delta\mathbf{\Sigma})$ and definition of $\mathbf{I}_3=\mathbf{Y}\backslash(\mathbf{Y}\ominus\mathbf{B})$.



(l) The set $\mathbf{X}=\mathbf{X}^c\backslash\mathbf{Y}=\emptyset$ denotes the end of the algorithm.

Figure 4: Classifying inner and outer contours.