M. Tech (Computer Science) Dissertation Series

AUTOMATIC SELECTION OF STRUCTURING ELEMENT FOR OBJECT CLASSIFICATION THROUGH MORPHOLOGY

A dissertation submitted in partial fulfilment of the requirements for the M. Tech. (Computer Science) degree of the Indian Statistical Institute

by

RAHUL BHATTACHARYYA

under the supervision of

Dr. BHABATOSH CHANDA

INDIAN STATISTICAL INSTITUTE
203 Barrackpur Trunk Road
Calcutta 700 035, INDIA

Automatic Selection of Structuring Element for Object Classification through Morphology

Rahul Bhattacharyya

M. Tech II, Computer Science

June 29, 1992

Abstract

In this work an algorithm to extract unique feature for each image from given set of distinct images which is called structuring element in terms of mathematical morphology is proposed. A new algorithm for finding distance transform is also suggested. The structuring element extracted by the said algorithm can be used to classify images and for other shape analysis purposes.

Keywords: Binary image, Mathematical Morphology, Dilation, Erosion, Open, Close, Structuring element, Distance transforms.

1 Introduction

Mathematical morphology is becoming more and more popular tool for dealing with the shapes of objects in an image[1-3]. The main difference between a mathematical morphological technique and traditional image processing technique is that the former treats images as ensemble of sets rather than as a signal. The language of mathematical morphology is that of set theory and the operations are defined in terms of the interaction between an object and a structuring element. Automatic identification and classification of objects in an image is a major area for machine vision. To classify the objects correctly it is essential to extract unique shape characteristics of those objects from the images.

Morphological operations can simplify image data preserving their essential shape characteristics and eliminate irrelevancies. As the identification and decomposition of objects, object features, object surface defects, and assembly defects correlate directly with shape, it is only natural that mathematical morphology has an essential role to play in machine vision. During several years people have been working in this field and many interesting algorithms have been developed for restrictive classes of images. Another strong point in favour of morphology is that each step of morphological algorithms are mathematically well founded. That means, given input image and structuring element output of a morphological operations can be determined analytically. So selection of operations and their sequence to accomplish a task are less heuristic in case of morphological algorithms. However, one of the main disadvantages is that the procedure for selecting structuring element is completely adhoc in nature. As a result, the algorithm designer selects suitable structuring element depending on types of objects and application in hand applying his intuition and experience. Here, in this work, we have proposed an algorithm to extract a unique shape characteristics of an object with respect to a set of distinct classes of objects. This leads to automatic selection of structuring elements for a major class of application, namely object

classification.

This report is organized as follows. Section 2 presents some basic relevant definitions, while Section 3 describes about the morphological operations. Problem definition and solution is given in section 4. In section 5 detail description and implementation steps of proposed algorithm is given. Section 6 gives a brief description about the experimental results while a few conclusions are given in Section 7.

2 Definitions

In a continuous domain R^2 an image is defined as a nonnegative function f, where f(x,y) is the value of the function f at the point (x,y). A digital image is defined by a finite valued function over a discrete domain Z^2. Let us assume a digital image domain $I \subset Z^2$ is rectangular array of size M x N, i.e.,

$$I = \{ (r,c) \mid r = 0,1,.....M-1;\ c = 0,1,.....,N-1\}$$

obtained by sampling of step size h along two orthogonal directions. The origin of I is assumed to be at upper- left corner point, also called pixel, of the image. A frame F of the domain I may be defined as a subset of I containing pixels of the first and last rows, and pixels of the first and last columns. Here we consider only binary images. A pixel (r,c) in I is defined as a foreground pixel or object pixel if its value is 1, and as a background point if its value is 0. Therefore, a digital object A is a subset of points of I that lie within or on the boundary of an analog figure A. The value of all pixels of A is same; they have the value 1.

x_1	x_2	x_3
x_8	x_0	x_4
x_7	x_6	x_5

Fig.1

w_1	w_2	w_3
w_8	w_0	w_4
w_7	w_6	w_5

Fig.2

Consider the pixel x_0 and the pixels in its close vicinity as shown in Fig.1.

Definition 1: Any pixel x_k (k = 2, 4, 6, 8) called the *4-neighbour* of the pixel x_0. Set of all these pixels comprises *4-neighbourhood* of x_0.

Definition 2: Any pixel x_k (k = 1, 2, ..., 8) called the *8-neighbour* of the pixel x_0. Set of all these pixels comprises *8-neighbourhood* of x_0.

Definition 3: Let p and q be two discrete pixels in Z^2 with coordinates (r_p, c_p) and (r_q, c_q), respectively, then the *distance* between p and q may be defined as

$$\text{(i)} \quad d_4(p,q) = |r_p - r_q| + |c_p - c_q|$$

$$\text{(ii)} \quad d_8(p,q) = \max\{ |r_p - r_q|, |c_p - c_q| \}$$

$$\text{(iii)} \quad d_e(p,q) = \sqrt{(r_p - r_q)^2 + (c_p - c_q)^2}$$

Definition 4: Two pixels p and q are said to be *connected* by a *4- (8-) connected path* if there exists a sequence of pixels $p_1 = p, p_2, ..., p_n = q$ such that for any i (i = 0, 1,....,n-1) value of p_i is same as that of p and q, and $d_4(p_i, p_{i+1}) = 1$ $[d_8(p_i, p_{i+1}) = 1]$.

Definition 5: A set of pixels in A is called *4- (8-) connected component* if every pair of pixels in A is connected by a *4- (8-) connected path.*

Unless explicitly specified, we assume objects as 8-connected component and the background as 4-connected.

Definition 6: A pixel p is called *surrounded by an 8-(4-)connected* object A if every 4-(8-)connected path from p to F contains at least one pixel of A. If the value of p be zero, then the set of all pixels connected to p constitutes a hole in A.

Definition 7: Let us define *a threshold function* t(v,θ), where v and θ are real numbers, which is defined as follows -

$$t(v,\theta) = \begin{cases} 1 & \text{if } v \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

For any image ((f(r,c))), t operated on ((f(r,c))) is defined as

$$t(f,\theta) = ((t(f(i,j),\theta)))$$

Definition 8: Let $w_0, w_1, \ldots\ldots, w_8$ represent the coefficients of 3x3 mask as shown in Fig.2 and let $x_0, x_1, \ldots., x_8$ be the corresponding pixel values when the mask is in an arbitrary position (r,c) in the image. The Coefficients and the pixel values can be expressed as column vectors; that is,

$$W = (w_0, w_1, w_2, \ldots, w_8)^t \text{ and}$$

$$X = (x_0, x_1, x_2, \ldots, x_8)^t$$

Then *convolution* at each point (r,c) may be expressed as dot product of these two vectors, i.e.,

$$X * W(r,c) = W.X = W^tX = x_0w_0 + x_1w_1 + \ldots + x_8w_8$$

Hence, convolving the image f with the mask W, is represented by translating W to every pixel of f and performing above operation, and is denoted by $f * W$.

3 Morphological operations

In morphological processing a small set (shape) is used as probe to extract shape characteristics of object (usually much bigger set). The operations are defined as interaction between a set (the object) and another set called structuring element. The primary operations are dilation and erosion. From these two, two other important operations, namely, opening and closing, are composed. Moreover, dilation and erosion are composed of even lower level operations; such as translation, set union and set intersection.

Definition 9: Let A be a subset of Z^2 and t be a point of Z^2. We denote *translation of A* by the point t by A_t and defined by

$$A_t = \{p \in Z^2 \mid p = a + t \text{ for some } a \in A\}$$

Definition 10: The *dilation* of a set $A \subset Z^2$ by another set $B \subset Z^2$ is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{p \in Z^2 \mid p = a + b \text{ for some } a \in A \text{ and } b \in B\}$$

The set B may be called structuring element. The dilation operation can also be represented as a union of translates:

$$A \oplus B = \bigcup_{a \in A} B_a = \bigcup_{b \in B} A_b$$

Dilation enlarges object by adding pixels (layer) around the boundary of object.

Definition 11: The *erosion* of a set $A \subset Z^2$ by another set $B \subset Z^2$ is denoted by $A \ominus B$ and is defined by

$$A \ominus B = \{p \in Z^2 \mid p + b \in A \text{ for every } b \in B\}$$

B is called a structuring element. Erosion can also be represented as an intersection of negative translates:

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

Erosion shrinks object by removing pixels retaining internal structure of object. Erosion and Dilation are dual operations as stated below.

Erosion Dilation Duality: $(A \ominus B)^c = A^c \oplus \check{B}$ where

$$\{\check{B} = \{x \mid \text{for some } b \in B,\ x = -b\}$$

Definition 12: The *opening* of a set $A \subset Z^2$ by another set $B \subset Z^2$ is denoted by $A \circ B$ and is defined by

$$A \circ B = (A \ominus B) \oplus B$$

The set B is called the structuring element. This open operation smoothes the contour, breaks narrow isthmuses, and eliminates small islands and sharp peaks or caps.

Definition 13:

The *closing* of a set $A \subset Z^2$ by another set $B \subset Z^2$ is denoted by $A \bullet B$ and is defined by

$$A \bullet B = (A \oplus B) \ominus B$$

The set B is called the structuring element. This close operation smoothes the contours, fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps on the contours. Opening and Closing are also dual operations.

Open Close Duality: $(A \circ B)^c = A^c \bullet \check{B}$

$$(A \bullet B)^c = A^c \circ \check{B}$$

4 Problem Definition and Solution

4.1 Formal definition of the problem

In most of the morphological image processing and image analysis algorithms, the structuring elements are selected in an adhoc basis depending on the problem and the class of images. In this work, on the other hand, we try to devise an algorithm that can select appropriate structuring element for a major class of problems, namely object classification. The problem may be formally presented as follows.

Given the m distinct objects A_k, k = 1,2,.....m; problem is to find the set of minimal structuring element $\{B_k\}$ such that

$$A_j \ominus B_k \begin{cases} \neq \phi & \text{for } j = k \\ = \phi & \text{otherwise} \end{cases}$$

By minimal structuring element we mean that structuring element which contains minimum number of 1-pixels. Minimality is important from computational point of view. Suppose $\#B$ denote number of pixels in structuring element B. Then A $\oplus$ B or A $\ominus$ B requires O(M.N. $\#B$) order of computation. Secondly, objects A_k, k = 1,2..... can take any size and shape. So by examining the complete object set we have to extract a set of structuring element $\{B_k\}$ in the learning phase. Each structuring element B_k is picked up from corresponding object A_k. Most simple, at the same time reasonable too, approach may be extracting a portion B_k of A_k lying within a window of size (2r+1)x(2r+1) such that

1. $B_k \not\subseteq A_j$ for $j \neq k$.

2. $r \leq m$ for all window of size (2m+1)x(2m+1) satisfying condition 1.

3. $\#B_k \leq \#B_{k_i}$ for all such portions of A_k satisfying condition 1 and 2.

4.2 A solution for the problem

Let us begin our discussion by defining distance transform of binary image as given below.

Definition 14: *Distance transform* maps a binary image matrix ((f(r,c))) to another matrix ((D(r,c))) where

$$D(r,c) = \min_{(k,l)}\{d_n((r,c),(k,l))\}$$

such that f(r,c) = 1 and f(k,l) = 0.

The value of n may be 4 or 8 depending on whether the object is 4- or 8-connected, respectively. So each element D(r,c) represents distance of 1-pixel (r,c) from background. There are different ways of finding distance transform of an image. Here we propose another way of computing this distance transform through convolution and a threshold function.

Suppose $R_r(p)$ denotes set of points(coordinates) of 1-pixels relative to p within a square of size (2r+1)x(2r+1) around 1-pixel p, and $\#R_r(p)$ be the number of elements in $R_r(p)$, i.e.

$$R_r(p) = \{ q - p \mid d_n(p,q) \leq r \text{ and value at } q = 1\}$$

and $\#R_r(p) = \text{card}[R_r(p)]$. Then $\#R_1(p)$ is the number of 1-pixels within a 3x3 window around p which are 4- or 8- connected of p depending on whether the object is 4- or 8-connected respectively. Now we define (black or 1-) surrounding of p, $s_r(p)$ respectively as

$$s_1(p) = \#R_1(p) \text{ and}$$

$$s_r(p) = \sum_{q \in R_1(p)} s_{r-1}(q) \text{ for } r > 1$$

Hence, $s_r(p)$ reflects number of 1-pixels around p at different distances from p, and thus gives an idea of distance of p from background. For example, if all the pixels within a square of size (2r+1)x(2r+1) around p be one then value of $s_r(p)$ becomes 9^r. The concept leads to the following algorithm. Let us denote the masks

N_8=

1	1	1
1	1	1
1	1	1

and N_4=

0	1	0
1	1	1
0	1	0

Therefore, for every element f(r,c) of f, the operation $f * N_n$ is defined as the sum of the all n-neighbour's of f(r,c) and f(r,c) and the result is placed in the location (r,c). The value of n may be 4 or 8 depending on object.

Algorithm 1:

Now, we compute the distance matrix D iteratively as follows-

$$\begin{aligned} S_0 &= f & D_0 &= t(S_0, \alpha^0) \\ S_1 &= S_0 * N_n & D_1 &= t(S_1, \alpha^1) + D_0 \\ S_2 &= S_1 * N_n & D_2 &= t(S_2, \alpha^2) + D_1 \\ &\cdot & &\cdot \\ &\cdot & &\cdot \\ &\cdot & &\cdot \\ S_k &= S_{k-1} * N_n & D_k &= t(S_k, \alpha^k) + D_{k-1} \end{aligned}$$

Then the distance transform $D = D_k$ for the smallest k such that $D_k = D_{k+1}$. The value of α is 5 if n is equal to 4 and 9 if n is equal to 8.

Above discussion and algorithm points out that surrounding has a monotonically non-decreasing relation with distance $d_n(p)$ of p from the background, that means for two pixels p and q with $d_n(p) < d_n(q)$ implies $s_{d_n(p)}(p) \leq s_{d_n(p)}(q)$ and $s_{d_n(q)}(p) < s_{d_n(q)}(q)$. In fact, surrounding of a pixel may be computed in a similar way as algorithm 1.

Algorithm 2:

Suppose we define a matrix $S = ((s_k(i,j)))$ as follows

$$\begin{aligned} S_0 &= f \\ S_1 &= S_0 * N_n \\ S_2 &= S_1 * N_n \\ &\cdot \\ &\cdot \\ &\cdot \\ S_k &= S_{k-1} * N_n \ . \end{aligned}$$

Then the surrounding matrix $S = ((s_k(i,j)))$ is given by $S = S_k$ for smallest k such

that

$$\max \{s_k(i,j)\} < \alpha^k$$

Proposition 1: If $R_r(p) = R_r(q)$ then $s_r(p) = s_r(q)$.

Proof: $R_r(p) = R_r(q)$ implies that the set of points which are at a distance r or less from p is same as that of from q. That is the set of 1-pixels within the square of size (2r+1)x(2r+1) around p is same as that of around q within the square of same size. So the proof follows from this observation along with the definition of $s_r(.)$. Q. E. D.

Proposition 2: If $s_r(p) \neq s_r(q)$ then $R_r(p) \neq R_r(q)$.

Proof: Suppose not. That means that $s_r(p) \neq s_r(q)$ implies $R_r(p) = R_r(q)$. But from **Proposition 1** $s_r(p) = s_r(q)$. Hence the contradiction! Q. E. D.

So $s_r(p)$ gives an idea of concentration of 1-pixels around p within a square neighbourhood. Moreover, the surrounding $s_r(p)$ does not reveal how the neighbouring pixels are spatially distributed around p over the neighbourhood of size (2r+1)x(2r+1). That means, though $R_r(p) = R_r(q)$ implies that $s_r(p) = s_r(q)$ and $s_r(p) \neq s_r(q)$ implies that $R_r(p) \neq R_r(q)$, the reverse is not true. In other words, $s_r(p) = s_r(q)$ does not guarantee that $R_r(p) = R_r(q)$. One possible way to extract this information is assigning position variant weights to neighbouring pixels before computing the surrounding $s_r(p)$. As a result, positional information of pixels will be reflected. Let us call this quantity as weighted surrounding and denote it by $w_r(p)$. So $w_r(p)$ can recursively defined as

$$w_0(p) = (f * W_n)(p) \text{ and}$$

$$w_r(p) = \sum_{q \in R_1(p)} w_{r-1}(q) \text{ for } r > 0$$

where n can be 4 or 8 as stated before, and correspondingly the weighted masks or templates are

$W_8 =$

32	2	64
16	1	4
256	8	128

and $W_4 =$

0	2	0
16	1	4
0	8	0

Proposition 3: $R_r(p) = R_r(q)$ iff $w_r(p) = w_r(q)$.

Proof: Suppose $R_r(p) = R_r(q)$. That means the set of points which are at a distance r or less from p is same as that of from q. That is the set of 1-pixels within the square of size (2r+1)x(2r+1) around p is same as that of around q within the square of same size. So the proof follows from this observation along with the definition of $w_r(.)$.

On the other hand, suppose $w_r(p) = w_r(q)$. Since the W_n have got unique values and are variant over the direction, so convolving image f with $w_n(.)$ we get $w_0(.)$ which gives unique value for a particular distribution of 1-pixels within a 3x3 square. Now $w_r(.)$ is obtained by recursively convolving r times with a 3x3 mask N_n. So, $w_r(p) = w_r(q)$ implies the surrounding as well as spatial distribution of 1-pixels around p and q are same. Q. E. D.

Hence, $w_r(p)$ uniquely describes spatial distribution of 1-pixels.

5 Description of the Algorithm

For each object this algorithm finds a structuring element which is the minimal among the square structuring elements contained within a minimum odd square and which completely erodes the other objects. Here we use the template W_n only once in the first step of the algorithm and in the succeeding iterative stage we use $W_n = N_n$ and perform the convolution operation. Combination of these two operations, namely convolution with W_n once and that with N_n there after, say k times, results the value $M_k(i,j)$ at pixel (i,j) which reflects spatial distribution of 1-pixels around the candidate (i,j) over a neighbourhood of size (3+2k)x(3+2k). So the value gives an idea of structure of that part of the object. In other words

we get an idea of spatial arrangement of 1-pixels, and consequently the maximum structuring element that completely fits at that position of the given object. After each convolution operation we try to find the structuring element B. As we move from one iteration to the next the size of the structuring element increases to square of next odd number. Since we are interested to find out as small distinctive structuring element as possible, so we try to find it at the earliest iterative stage. Secondly, we have to make an ordering of structuring elements extracted in a particular iteration. At any iteration, there are two information for ordering these structuring elements : (i) number of pixels in the structuring elements, and (ii) the value $M_k(i,j)$ situated at the centre of window containing the structuring element. Now we have to decide relative priority of these two information as well as how can they be utilized. At any iterative stage, let B_n and B_N be the two structuring elements having weighted surrounding value n and N respectively such that $n < N$. Let number of pixels in B_n and B_N are $\#B_n$ and $\#B_N$, respectively. It is important to note that $\#B_n$ may be greater than, equal to or less than $\#B_N$; but $B_N \not\subseteq B_n$, i.e., B_N with larger value cannot be contained within B_n with smaller value. On the other hand $B_n \subset B_N$ or $B_n \not\subset B_N$, i.e., structuring element with smaller $\#B_n$ may or may not be contained in structuring element with larger $\#B_N$. The following algorithmic steps are based on these important observations. We also observe that the value we compute for each pixel could be at most $\alpha^k x(2^k - 1)$ ($k = 1, 2,....$) at the k^{th} iterative stage where α can be 9 or 5 depending on the 8- or 4-connectivity of object, respectively. Let there be m different objects $A_1, A_2, ..., A_m$ and the problem is to find the minimal structuring element for the i^{th} object A_i. So we move from k^{th} iterative stage to $(k+1)^{th}$ iterative stage of the algorithm for the image f_j containing A_j

(i) if in the k^{th} iterative stage maximum of the value M_{j_k} Computed for the object A_j, for all j and $j \neq i$, attains the maximum possible value $\alpha^k x(2^k - 1)$ ($k \geq 1$), or

(ii) if all the structures of size $(3+2k)x(3+2k)$ of the image A_i contained

in one or more structures of the other images A_j, $j \neq i$.

The search procedure for the required structuring element for the i^{th} image at the k^{th} iterative stage is as follows-

Arrange the distinct weighted surrounding values in the descending order of magnitude for all the images after performing convolution operation k times with N. Let us suppose that for the i^{th} image the values are stored in the array V_i, i=1,2,...,m. Check whether $V_j(1)$ for all j and $j \neq i$ equal to maximum value $\alpha^k x(2^k - 1)$. If it is found true then proceed for the next iteration, because there is no distinguishable structure in the object A_i with size (3+2k)x(3+2k) which can discriminate object A_i from the object B_j, $j \neq i$. If it is found that none of the $V_j(1)$ for all j and $j \neq i$ attains the maximum value then check for all $V_j(l)$'s with $V_j(l)$'s > $V_i(1)$ for all j and $j \neq i$ whether $V_i(1)$ is contained in any of $V_j(l)$'s for all j and $j \neq i$. If it is found to be false then $V_i(1)$ is one possible structuring element and then we consider $V_i(2)$ only if number of elements in this is less than $V_i(1)$ and search in a similar manner with $V_i(2)$, and so on. The algorithm eventually terminate because the images with which we have started are distinct.

Algorithm 3:

Input: m images f_i, i = 1,2,.....,m each containing a single object.

Output: Set of structuring element $\{B_i \mid i = 1, 2, ..., m\}$

For the i^{th} image perform the following steps -

Step 0. Set $\alpha = 5$ if the object is 4-connected or $\alpha = 9$ if the object is 8-connected.

Step 1. Set iteration = 0 and $F_i = 0$, i = 1,2,...,m.

Step 2. Perform operation $M_i = f_i * W_n$ for all i.

step 3. Size k = (3 + 2 * iteration)

step 4. Collect distinct elements of M_i in array V_i and order them in descending order for all i.

step 5. If $F_i = 0$ then check

if $V_j(1) = \alpha^k x(2^k - 1)$ for all j and $j \neq i$. then go to step 8

else set $p = 1$.

Step 6. For all V_i (p) do the following substeps -

(1) set flag = 0

(2) check for all q such that $V_j(q) > V_i(p)$ for all j and $j \neq i$, whether $V_i(p)$ is contained in any $V_j(q)$.

if false (a)

(i) then $B_i(r,s)$ is a possible structuring element of size kxk around location (r,s) of $A_i(p)$.

(ii) Check for the next p and if number of pixels in this structure around the location of $A_i(p)$ is less than number of pixels in $B_i(r,s)$ then go to step 6(2) otherwise set flag = 1.

if true (b)

(i) keep location of $A_i(p)$ in the array VV_i

(ii) check for next p and check whether the structure around $A_i(p)$ is contained in any of the structure around the location of VV_i array. if true go to 6.2.b(i) otherwise go to 6.

(3) if flag = 1 then declare that one structuring element has been found and set $F_i = 1$.

Step 7. If $F_i = 1$ for all i, then terminate.

Step 8. iteration = iteration + 1.

Step 9. Perform $M_i = M_i * N_n$ and go to Step 3.

5.1 Noise immunity

In the present work, we have to deal with binary images and the problem is to find the structuring element which can erode all objects except the candidate one. As we have discussed earlier, the structuring elements are extracted from the objects themselves and are basically some portion of them. Now, these binary objects are obtained either by segmenting graylevel images or by suitable lighting arrangement and hardware. The binary images that are being used to extract structuring element may be contaminated by noise. Noise may be additive (that converts background pixels to object pixels) or may be subtractive (that converts object pixels to background). If the portion of training object from which structuring element be extracted contains additive noise then the structuring element may erode test objects of some class completely, since binary erosion tries to find exact match. Situation would be worse if the test objects are corrupted by subtractive noise. To solve this problem, i.e., to make algorithm noise insensitive we adapt the following approach. Our target is to extract structuring element which would work even in noisy situation. Let us assume that the noise is random and area of largest blob formed by either additive noise or subtractive noise alone be N_a. Then opening the training images by a disc structuring element of size greater than N_a removes additive noise, and closing the test images by the disc structuring element removes subtractive noise. To be on a safe side, in lieu of opening and closing, erosion and dilation operation, respectively can be used. Present algorithm has to extract structuring element B_i from A_i such that $A_j \ominus B_i = \phi$ for all j and $j \neq i$. So to extract most distinguishing structuring element B_i we may open A_i and close A_j for all j and $j \neq i$ as discussed earlier. Again open and close may be replaced by erode and dilate respectively.

6 Experimental Results

The algorithms described in previous section has been developed in C language and implemented on Vax providing Ultrix operating system, and tested on several images. Only a few of them are presented here.

The images shown (a) in Figs. 3 to 10 were given input to the algorithm 3 all at a time. Each figure contains a image of a single representative object from 8 different classes (that means m=8). Here we have assumed 8-connectivity for the objects. To assure that the extracted structuring element B_i be immune to noise, image corresponding to object A_i is opened and rest of the images are closed as suggested in section 5.1. In each of these 8 images Fig. (a) is the original image, Figs. (b) and (c) are the open and closed version of the original images obtained by opening and closing, respectively image with a 3x3 structuring element. Fig. (d) is the required structuring element extracted by the algorithm, and is a part of Fig. (b) but not of others.

Apart from that we have also studied pairwise cases. Only two sets of results for pairwise cases are presented here, and are shown in Fig. 11 to 20. In the pairwise cases we have adopted both the ways for removing noise from the images, namely dilate-erode and open-close. Fig. 11(a) and (b) represents the original images, Fig. 11(c) and (d) represent the structuring elements when object is assumed to be 4-connected. Fig. 11(e) and (f) represent the structuring elements when object is assumed to be 8-connected.

For extracting noise immune structuring element corresponding to Fig. 11(a), Figs. 12(b) and 13(a) which are eroded and dilated, respectively, considered. The results are shown in Figs. 12(c) and 12(d) when object is assumed to be 4-connected and 8-connected, respectively. Similarway, dilated and eroded image as shown in Figs. 12(a) and 13(b) are given to the algorithm as input to extract structuring element corresponding to Fig. 11(b). The results are shown in Figs. 13(c) and 13(d).

For extracting noise immune structuring element corresponding to Fig. 11(a), Figs. 14(a) and 15(b) which are opened and closed, respectively, are considered. The results are shown in Figs. 14(c) and 14(d) when object is assumed to be 4-connected and 8-connected, respectively. Similarway, closed and opened images as shown in Figs. 14(b) and 15(a) are given to the algorithm as input to extract structuring element corresponding to Fig. 11(b). The results are shown in Figs. 15(c) and 15(d).

Figs. 16 to 20 also represent the pairwise experimental results with two images given in Fig. 16(a) and (b) and the explanation is exactly same as the explanation given for Figs. 11 to 15.

7 Conclusion

In this report an algorithm for automatic extraction of structuring element is proposed. Given a set of m different objects, the algorithm can extract a set of m structuring elements for, say, classification purpose. The structuring elements are extracted from the given set of objects and are basically small portion of the corresponding objects. Small portions are selected based on the spatial distribution of object pixels over a odd size window. Under this constraint the selected structuring elements are minimal set which is assured by the search procedure described in Algorithm 3. Extracted structuring element is made immune to noise by opening(eroding) candidate object corresponding to which structuring element is required and closing(dilating) rest of the objects. Experiments are carried out by assuming both 8- and 4-connectivity for objects. It is found that the size of extracted structuring element is larger in case of 4-connecctivity than that if 8-connectivity is assumed for objects. Time required for extracting complete set of structuring elements depends on number of objects given (i.e. value of m) as well as on the similarity among objects.

Reference

1. J. Seerra, "Image Analysis and Mathematical Morphology", Academic Press, London 1982.

2. Charles R. Giardina and Edward R. Daugherty, "Morphological Methods in Image and Signal Processing", Prentice Hall, 1987.

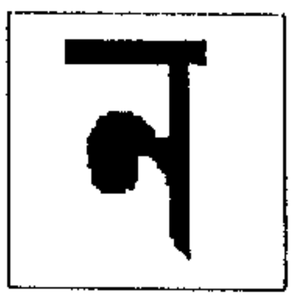
3. R. M. Haralick and L. G. Shapiro, "Computer and Robot vision [Ch. 5]", Addision-Wesley Publishing Company, Reading Mass, 1992.

original

(a)

open

(b)



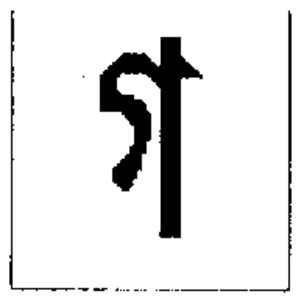
close

(c)

se

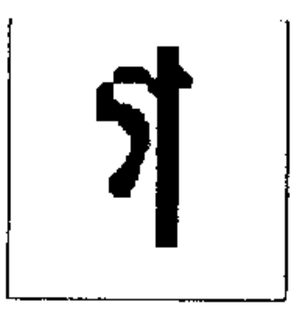
(d)

Fig.3 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.



original

(a)



open

(b)

close

(c)

se

(d)

Fig.4 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.

original

(a)

open

(b)

close

(c)

se

(d)

Fig.5 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.

original

(a)

open

(b)

close

(c)

se

(d)

Fig.6 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.

original

(a)

open

(b)

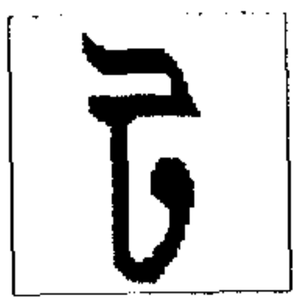
close

(c)

se

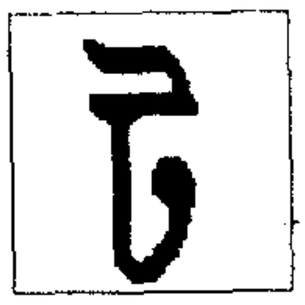
(d)

Fig.7 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.



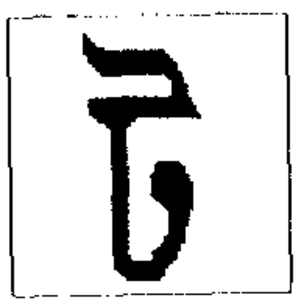
original

(a)



open

(b)



close

(c)

se

(d)

Fig.8 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.

original

(a)

open

(b)

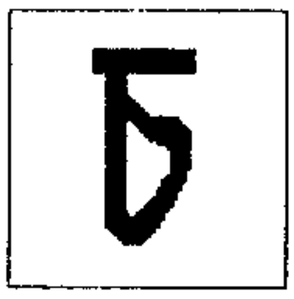
close

(c)

s

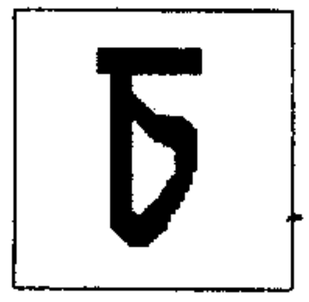
(d)

Fig.9 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.



original

(a)



open

(b)

close

(c)

se

(d)

Fig.10 Results showing noise immune structuring element extracted by proposed algorithm.
(a) Original image, (b) Opened image, (c) Closed image and (d) Extracted structuring element assuming 8-connectivity.

original

(a)

original

(b)

s

(c)

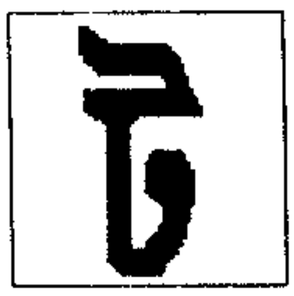
s

(d)

(e)

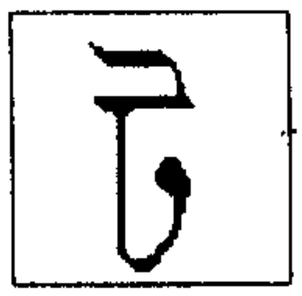
(f)

Fig.11 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Original image of the first object, (b) Original image of the second object, (c) Extracted structuring element from (a) assuming 4-connectivity, (d) Extracted structuring element from (b) assuming 4-connectivity, (e)Extracted structuring element from (a) assuming 8-connectivity, and (f) Extracted structuring element from (b) assuming 8-connectivity.



dilate

(a)



erode

(b)

se

(c)

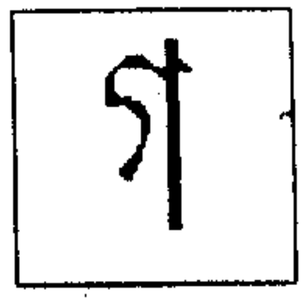
se

(d)

Fig.12 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Dilated image of Fig.11(a), (b)Eroded image of Fig.11(a), (c) Extracted structuring element from (b) assuming 4-connectivity, (d)Extracted structuring element from (b) assuming 8-connectivity.

dilate

(a)



erode

(b)

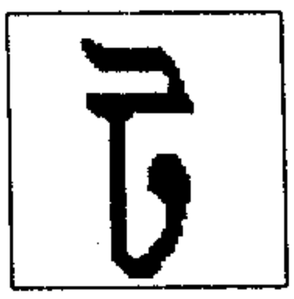
s

(c)

s

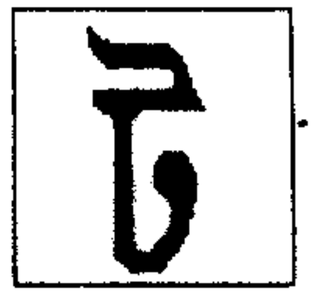
(d)

Fig.13 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Dilated image of Fig.11(b), (b)Eroded image of Fig.11(b), (c) Extracted structuring element from (b) assuming 4-connectivity, (d)Extracted structuring element from (b) assuming 8-connectivity.



open

(a)



close

(b)

s

(c)

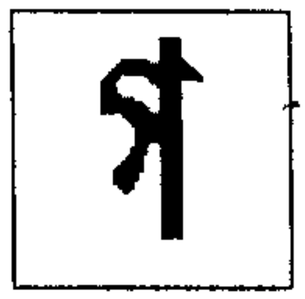
s

(d)

Fig.14 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Opened image of Fig.11(a), (b) Cloed image of Fig.11(a), (c) Extracted structuring element from (a) assuming 4-connectivity, (d)Extracted structuring element from (a) assuming 8-connectivity.

open

(a)



close

(b)

S

(c)

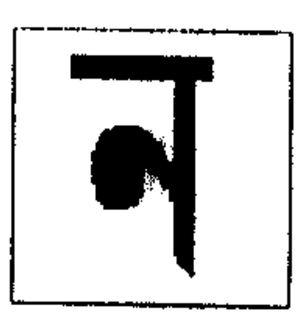
S

(d)

Fig.15 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Opened image of Fig.11(b), (b) Closed image of Fig.11(b), (c) Extracted structuring element from (a) assuming 4-connectivity, (d)Extracted structuring element from (a) assuming 8-connectivity.

original

(a)



original

(b)

se

(c)

s

(d)

s

(e)

s

(f)

Fig.16 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Original image of the first object, (b) Original image of the second object, (c) Extracted structuring element from (a) assuming 4-connectivity, (d) Extracted structuring element from (b) assuming 4-connectivity, (e)Extracted structuring element from (a) assuming 8-connectivity, and (f) Extracted structuring element from (b) assuming 8-connectivity.

dilate

(a)

erode

(b)

se

(c)

se

(d)

Fig.17 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Dilated image of Fig.16(a), (b)Eroded image of Fig.16(a), (c) Extracted structuring element from (b) assuming 4-connectivity, (d)Extracted structuring element from (b) assuming 8-connectivity.

dilate

(a)

erode

(b)

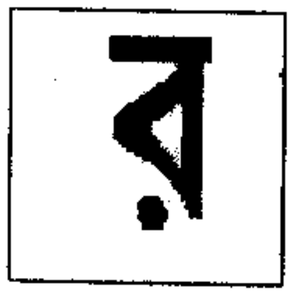
se

(c)

s

(d)

Fig.18 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Dilated image of Fig.16(b), (b)Eroded image of Fig.16(b), (c) Extracted structuring element from (b) assuming 4-connectivity, (d)Extracted structuring element from (b) assuming 8-connectivity.



open

(a)

close

(b)

se

(c)

s

(d)

Fig.19 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Opened image of Fig.16(a), (b) Cloed image of Fig.16(a), (c) Extracted structuring element from (a) assuming 4-connectivity, (d)Extracted structuring element from (a) assuming 8-connectivity.

open

(a)

close

(b)

s

(c)

s

(d)

Fig.20 Results showing noise immune structuring element extracted by proposed algorithm for pairwise cases. (a) Opened image of Fig.16(b), (b) Closed image of Fig.16(b), (c) Extracted structuring element from (a) assuming 4-connectivity, (d)Extracted structuring element from (a) assuming 8-connectivity.