ABSTRACT

Decomposition of Geometric-Shaped Structuring Elements and Optimization on Euclidean Distance Transformation Using Morphology

by
Hong Wu

Mathematical morphology which is based on geometric shape, provides an approach to the processing and analysis of digital images. Several widely-used geometric-shaped structuring elements can be used to explore the shape characteristics of an object. In first chapter, we present a unified technique to simplify the decomposition of various types of big geometric-shaped structuring elements into dilations of smaller structuring components by the use of a mathematical transformation. Hence, the desired morphological erosion and dilation are equivalent to a simple inverse transformation over the result of operations on the transformed decomposable structuring elements. We also present a strategy to decompose a large cyclic cosine structuring element. The technique of decomposing a two-dimensional convex structuring element into one-dimensional ones is also developed.

A distance transformation converts a digital binary image which consists of object (foreground) and non-object (background) pixels, into a gray-level image in which all object pixels have a value corresponding to the minimum distance from the background. Computing the distance from a pixel to a set of background pixels is in principle a global operation, that is often prohibitively costly. The Euclidean distance measurement is very useful in object recognition and inspection because of the metric accuracy and rotation invariance. However, its global operation is difficult to decompose into small neighborhood operations because of the nonlinearity of Euclidean distance computation. In second chapterpresents three algorithms to the Euclidean distance transformation in digital images by the use of the grayscale morphological erosion with the squared Euclidean

distance structuring element. The optimal algorithm only requires four erosions by small structuring components and is independent of the object size. It can be implemented in parallel and is very efficient in computation because only the integer is used until the last step of a square-root operation.

Feature extraction and object recognition can be achieved by mathematical morphology. An object is analyzed by using a set of primitive shapes. Primitive shapes are designed in binary templete. To detect each matching pattern, mapping of a set of templetes on the unknown object need to be performed. The templetes along with morphological operations and weights corresponding to each matching pattern are designed priorily and stored in database. During recognition, we get several sub-result according to its templetes. These results combined with their weights will produce a final matching probability. In third chapter, a new feature selection criterion based on mathematical morphology is proposed.

Decomposition of Geometric-Shaped Structuring Elements and Optimization on Euclidean Distance Transformation Using Morphology

by
Hong Wu

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science
Department of Computer and Information Science
May, 1992

APPROVAL PAGE

Decomposition of Geometric-Shaped Structuring Elements and Optimization on Euclidean Distance Transformation Using Morphology

by
Hong Wu

5/11/92

Dr. Frank Y. Shih, Thesis Advisor
Assistant Professor of Computer and
Information Science Department, NJIT

ACKNOWLEDGEMENT

First and foremost, thanks to my advisor Dr. Frank Y. Shih, who has worked closely with me on every piece of work which has gone into my thesis.

I also acknowledge the financial support and precious advices on my study and research provided by Dr. Frank Y. Shih of the CIS department of New Jersey Institute of Technology.

Special thanks are also due to my groupmates Joy Moh, Wai-Tak Wong, Chen, and friends Ming Che Chen, Y. Chen, Ying Wu and my parents.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

DECOMPOSITION OF GEOMETRIC-SHAPED STRUCTURING ELEMENTS USING MORPHOLOGICAL TRANSFORMATIONS ON BINARY IMAGES

1.1 INTRODUCTION

Mathematical morphology which is based on geometric shape, provides an approach to the processing and analysis of digital images [2,3,7,8] The underlying strategy is to understand the characteristics of an object by probing its microstructure with various forms which are known as *structuring elements* The analysis is geometric in character and that approaches image processing from the vantage point of human perception Appropriately used, morphological operations also tend to simplify image data while preserving their essential shape characteristics and eliminating irrelevancies.

In gray-scale processing, two elementary morphological operations, *dilation* and *erosion*, are similar to the convolution operator, except addition/subtraction are substituted for multiplication and maximum/minimum for summation. Let $f : F \rightarrow F$ and $k : K \rightarrow E$ denote the gray-scale image and the gray-scale structuring element, respectively. The symbols, E, F, and K represent the Euclidean space, bounded domain for f, and bounded domain for k, respectively. *Gray-scale dilation* of f by k is given by

$$(f \oplus k)(x) = \max_{x-z \in F,\ z \in K} \{ f(x-z) + k(z) \}. \tag{1}$$

Gray-scale erosion of f by k is given by

$$(f \ominus k)(x) = \min_{x+z \in F,\ z \in K} \{ f(x+z) - k(z) \}. \tag{2}$$

The computation of gray-scale dilation and erosion can be implemented very efficiently by using the architecture of threshold decomposition of gray-scale morphology into binary morphology [4]. In practical applications, dilation and erosion pairs are used in sequence, either dilation of an image followed by erosion of the dilated result or vice versa. Either way, the result of iteratively applied dilations and erosions is an elimination of specific image details smaller than the structuring element without the global geometric distortion of unsuppressed features. The *opening* of f by k is defined as

$$A \circ B = (A \ominus B) \oplus B. \tag{3}$$

The *closing* of f by k is defined as

$$A \bullet B = (A \oplus B) \ominus B. \tag{4}$$

A variety of existing image processing architectures including CELLSCAN, BIP, PICAP, DIP, and Cytocomputer [1,6,7], are applicable to the morphological processing. However, each stage only processes 3×3 neighborhoods. Implementation difficulties arise when the applied algorithm requires morphological operations by large-sized structuring elements. Hence, the techniques of decomposing big structuring elements into combined structures of segmented small components are extremely important.

Another advantage of decomposing structuring elements is the reduction of calculation. For example, if we decompose a structuring element of size $(p+q-1) \times (p+q-1)$ into a dilation of two smaller structuring components of sizes $p \times p$ and $q \times q$, the number of calculations required in morphological operations will be reduced from $(p+q-1)^2$ to p^2+q^2. In order to satisfy $(p+q-1)^2 \geq p^2+q^2$, we rearrange it as $(p-1)(q-1) \geq ½$, that indicates $p, q \geq 2$. In order to incorporate into the hardware architecture and to achieve the optimal reduction rate, we choose to decompose any sized structuring element into smaller structuring components of size 3×3.

An optimalization algorithm for decomposing binary morphological structuring elements was developed by Zhuang and Haralick [8]. A strategy for decomposing certain types (*linear-sloped, convex,* and *concave*) of gray-scale morphological structuring elements into dilations or maximum selections of small components was explored by Shih and Mitchell [5]. In chapter one, we present a unified technique to simplify the decomposition of various types of big geometric-shaped structuring elements only into dilations of smaller structuring components.

Chapter one is organized as follows. In Section 2, the definitions concerning the types of morphological structuring elements are given. In Section 3, a few mathematical morphology properties related to the decomposition are introduced. We present the decomposition technique for one-dimensional (1D) geometric-shaped structuring elements in Section 4 and for two-dimensional (2D) ones in Section 5. In Section 6, a new mathematical transformation that is applied on a nondecreasing cosine structuring element to become decreasing and that creates a quite significant result is presented. In Section 7, a strategy dealing with the decomposition of 2D structuring elements into 1D elements is described. Finally, we make some conclusions.

1.2 DEFINITIONS OF TYPES OF STRUCTURING ELEMENTS

A few definitions related to the types of structuring elements are given in this section. For expressional simplicity, the definitions are described in one dimension. The extension to two dimensions is straightforward. Let the origin of a structuring element $k_{(2n+1)}(x)$ with the odd size $2n+1$ be located at the central point as shown in Fig. 1, such that the x-coordinate of the structuring element is: $x = -n, \cdots, -1, 0, 1, \cdots, n$. Let m_i denote the slope of the line segment i. *Definition 1.* A structuring element $k(x)$ is called *eccentrically decreasing* if

$$\begin{cases} k(i+1) \geq k(i) & \text{for } i = -n, \cdots, -1 \\ k(i-1) \geq k(i) & \text{for } i = 1, \cdots, n\ . \end{cases} \tag{5}$$

Otherwise, $k(x)$ is called *eccentrically nondecreasing*.

Note that we are only interested in the morphological operations by the eccentrically decreasing structuring elements because the morphological operations by a nondecreasing structuring element will induce nonsignificant results. Fig. 2 illustrates that the dilated result of a binary image f with two gray levels, zero indicating background and C indicating foreground, by a large cyclic cosine structuring element k (it is certainly nondecreasing) is a constant image with the gray level of the highest value $C + 1$ in the calculation, and the eroded result is a constant image with the gray level of the lowest value -1.

Definition 2. A structuring element $k(x)$ is called *linearly-sloped* if

$$\begin{cases} m_i = c_1 & \text{for } i = -(n-1), \cdots, -1 \\ m_i = c_2 & \text{for } i = 1, \cdots, n-1\ , \end{cases} \tag{6}$$

where c_1 and c_2 are constants. An example of a linearly-sloped structuring element with $c_1 = c_2$ is shown in Fig. 1. *Definition 3.* A structuring element $k(x)$ is called *convex*, as shown in Fig. 3, if

$$|m_1| \leq |m_2| \leq |m_3| \leq \cdots \tag{7}$$

Definition 4. A structuring element $k(x)$ is called *concave*, as shown in Fig. 4, if

$$|m_1| \geq |m_2| \geq |m_3| \geq \cdots \tag{8}$$

When working on the 2D structuring elements, we need the following definitions: *Definition 5.* A 2D gray-scale structuring element can be viewed as a 3D geometric surface with its heights corresponding to the element's gray values. Mathematically, either a parametric or a nonparametric form can be used to represent a surface. A non-parametric representation is either implicit or explicit. For a 3D geometric surface, an implicit, non-parametric form is given by

$$f(x,y,z) = 0. \tag{9}$$

In this form, for each x- and y-values the multiple z-values are allowed to exist. The representation of a 2D gray-scale structuring element can be expressed in an explicit, nonparametric form of

$$k(x,y) = \{z \mid z = g(x,y),\ (x,y) \in \text{geometric shape domain}\}. \tag{10}$$

In this form, for each x- and y-values only one z-value is obtained. Hence, the 2D gray-scale structuring element can be assigned a unique datum at each (x,y) location. Some often-used 2D geometric shapes are sphere, cone, ellipsoid, hyperboloid, Gaussian, exponential, damped sine, and damped cosine. *Definition 6.* A structuring element $k(x,y)$ is called *additively-separable* if

$$k(x,y) = p(x) + q(y), \tag{11}$$

where $p(x)$ and $q(y)$ are the functions of only x and y, respectively.

1.3 DECOMPOSITION PROPERTIES

The decomposition properties presented in this section can be suited for any binary (with two gray levels, ''0'' indicating background and ''C'' indicating foreground) or gray-scale image f, but only limited to the decreasing gray-scale structuring elements k. Note that the value C must be larger enough than the height of the structuring element.

Property 1. If a structuring element k can be decomposed into sequential dilations of several smaller structuring components, then a dilation (or erosion) of an image f by k is equal to successively applied dilations (or erosions) of the previous stage output by these structuring components. All convex structuring elements can be decomposed using this technique. As shown in Fig. 3, let k_1, $k_2, \cdots$, and k_n denote the segmented smaller linearly-sloped structuring components with the sizes s_1, $(s_2 - s_1)$, $\cdots$, and $(s_n - s_{n-1})$, respectively. That is

$$\text{if} \quad k = k_1 \oplus k_2 \oplus \cdots \oplus k_n,$$

$$\text{then} \quad f \oplus k = (\cdots((f \oplus k_1) \oplus k_2) \oplus \cdots) \oplus k_n$$

$$\text{and} \quad f \ominus k = (\cdots((f \ominus k_1) \ominus k_2) \ominus \cdots) \ominus k_n. \tag{12}$$

Property 2. If a structuring element k can be decomposed into a maximum selection over several structuring components, then a dilation (or erosion) of an image f by k is equal to finding a maximum (or minimum) of each applied dilation (or erosion) of the image by these structuring components. All concave structuring elements can be decomposed using this technique. As shown in Fig. 4, let k_1, $k_2, \cdots$, and k_n denote the segmented structuring components with the sizes s_1, s_2, $\cdots$, and s_n, respectively. That is

$$\text{if} \quad k = \max(k_1, k_2, \cdots, k_n),$$

$$\text{then} \quad f \oplus k = \max(f \oplus k_1,\ f \oplus k_2, \cdots, f \oplus k_n)$$

$$\text{and} \quad f \ominus k = \min(f \ominus k_1,\ f \ominus k_2, \cdots, f \ominus k_n). \tag{13}$$

The computational complexity increases much more for the concave decomposition because further decompositions are required to perform the dilations (or erosions) by the structuring components $k_2, k_3, \cdots, k_n$ which have larger sizes than those in the convex decomposition and also an additional maximum (or minimum) operation. *Property 3.* Let $f : F \rightarrow E$ and $k : K \rightarrow E$. Let $x \in (F \oplus K) \cap (F \ominus \hat{K})$ be given. The relationship between dilation and erosion is

$$f \ominus_g k = -((-f) \oplus_g \hat{k}), \quad \text{where} \quad \hat{k}(x) = k(-x), \ \text{reflection of} \quad k \tag{14}$$

This implies that we may calculate erosion using dilation, or vice versa. *Property 4.* For computational simplicity in decomposition, the gray-value at the origin (i.e. the highest value) of a decreasing structuring element can be suppressed to zero. In other words, we could subtract a constant $\overrightarrow{P_c}$ which is equal to the highest value, from the structuring element. The dilation and erosion then become

$$f \oplus (k - \overrightarrow{P_c}) = (f \oplus k) - \overrightarrow{P_c}\ , \tag{15}$$

$$f \ominus (k - \overrightarrow{P_c}) = (f \ominus k) + \overrightarrow{P_c}\ . \tag{16}$$

Property 5. All the linearly-sloped and 1D convex structuring elements can be decomposed into dilations of smaller structuring components. For 2D convex structuring elements, if each value $k(x,y)$ can be expressed as: $k(x,y) = p(x) + q(y)$, i.e., 2D additively-separable, then it can be decomposed into dilations of smaller structuring components. The other types of convex and all the concave structuring elements can be decomposed using a maximum selection over several segmented structuring components. Interested readers should refer to [5] for details.

1.4 1D GEOMETRIC-SHAPED STRUCTURING ELEMENTS DECOMPOSITION

1.4.1 Semicircle, Semiellipse, Gaussian, Parabola, Semihyperbola, Cosine, and Sine

A 1D geometric-shaped structuring element $k(x)$ is actually the geometric curve $f(x,y)=0$, in a 2D Euclidean space. We assign each point in the structuring element by the negated distance from this point to the horizontal curve tangent to the top of the curve in order to ensure that $k(x)$ is decreasing An origin-centered semicircle can be represented implicitly by

$$f(x,y)=x^2+y^2-r^2=0, \tag{17}$$

where $|x|\le r$, $0\le y\le r$, and r denotes the radius. Eq (17) can be rewritten in the explicit form of

$$y=\sqrt{r^2-x^2} \tag{18}$$

Table 1. 1D Geometric-Shaped Structuring Elements

Name	Representation	Structuring Element
Semiellipse	$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$	$b\left[1-\frac{x^2}{a^2}\right]^{1/2}-b$
Gaussian	$y=\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$	$\frac{1}{\sqrt{2\pi}}e^{-x^2/2}-\frac{1}{\sqrt{2\pi}}$
Parabola	$x^2=-4ay$	$\frac{-x^2}{4a}$
Semihyperbola	$\frac{y^2}{b^2}-\frac{x^2}{a^2}=1$	$-b\left[1+\frac{x^2}{a^2}\right]^{1/2}+b$

From Definition 1 in Section 2, it is known that the semicircle is an eccentrically decreasing structuring element. For computational simplicity, a constant "$-r$" is added to suppress the y-value at the origin to zero. Hence,

$$k(x)=\sqrt{r^2-x^2}-r. \tag{19}$$

Assuming that $r = 3$, the structuring element $k(x) = \sqrt{9-x^2} - 3$, is numerically expressed as

$$k(x) = [\ -3\ \ \sqrt{5}-3\ \ \sqrt{8}-3\ \ 0\ \ \sqrt{8}-3\ \ \sqrt{5}-3\ \ -3\ ]\,.$$

This is a convex structuring element because the slopes $|m_i|$ is increasing eccentrically from the center. The similar method can also apply to other geometric-shaped structuring elements as listed in Table 1. Note that the cosine and sine structuring elements both contain a combination of convex and concave components.

1.4.2 Decomposition Strategy

The following propositions are based on the assumption that the input image f is binary with two gray values: "C" indicating foreground and "0" indicating background, where C is a large number. The reason for setting the object points to a sufficiently large number is that the minimum selection in erosion or the maximum selection in dilation will not be affected by the object's gray value. The erosion of f by a geometric-shaped structuring element results in the output image reflecting the minimum distance (i.e., the predefined values in the structuring element) from each object pixel to all the background pixels. *Proposition 1.* A 1D eccentrically decreasing, geometric-shaped structuring element $k(x)$ can be represented as $k(x) = g(x) + c$, where $k(x) \le 0$, $g(x)$ represents a geometric-shaped function, and c is a constant. It satisfies that for all $0 < x_1 < x_2$ and $x_2 < x_1 < 0$, there exists $g(x_1) > g(x_2)$. If a transformation Ψ satisfies that for all $0 < x_1 < x_2$ and $x_2 < x_1 < 0$, there exits $0 \le \Psi[|k(x_1)|] < \Psi[|k(x_2)|]$, then we can obtain a new structuring element k^*, in which $k^*(x) = -\Psi[|k(x)|]$, where $k^*(x) \le 0$. Let Ψ^{-1} is the inverse transformation of Ψ. The $k^*(x)$ can be used in morphological erosion on a binary image f, so that the desired result $(f \ominus k)(x)$ is obtained by

$$(f \ominus k)(x) = \Psi^{-1}[\,(f \ominus k^*)(x)\,], \tag{20}$$

for all nonzero values in $(f \ominus k^*)(x)$. Those zeros in $(f \ominus k^*)(x)$ still remain zeros in $(f \ominus k)(x)$. [*Proof*]: Since the binary image f is represented by two values "C" and "0", we obtain the following according to eq. (2)

$$(f \ominus k)(x) = \min\,\{\, C - k(z),\ 0 - k(z')\,\}, \tag{21}$$

where all z's satisfy that $x + z$ is located within the foreground and all z''s satisfy that $x + z'$ is located within background. Note that C has been selected to be large enough to ensure that "$C - k(z)$" is always greater than "$0 - k(z')$." Hence, we have

$$(f \ominus k)(x) = \min\,\{\, -k(z')\,\}. \tag{22}$$

Similarly, if the new transformed structuring element k^* is used for erosion, we obtain

$$(f \ominus k^*)(x) = \min\,\{\, -k^*(z_1')\,\}, \tag{23}$$

where all z_1''s satisfy that the pixel $x+z_1'$ is located within the background. Because

$$g(x_1) > g(x_2) \rightarrow k(x_1) > k(x_2) \tag{24}$$

and

$$\Psi\,[\,|k(x_1)|\,] < \Psi\,[\,|k(x_2)|\,] \rightarrow k^*(x_1) > k^*(x_2) \tag{25}$$

for all $0 < x_1 < x_2$ and $x_2 < x_1 < 0$, we have $k(x_1) > k(x_2)$ and $k^*(x_1) > k^*(x_2)$. This ensures that each location z' in eq. (22) is the same as the location z_1' in eq. (23) for the same input image f. Because $k^*(x) = -\Psi[\,|k(x)|\,]$, we have $k^*(z_1') = -\Psi[\,|k(z')|\,]$.

Then $k(z')$ can be obtained by $|k(z')| = \Psi^{-1}[-k^*(z_1')]$. Hence, $-k(z') = \Psi^{-1}[-k^*(z_1')]$. Because the transformation Ψ preserves the monotonically increasing property, we obtain

$$\min \{ -k(z') \} = \Psi^{-1}[\min \{-k^*(z_1')\}]. \tag{26}$$

Combining eqs. (22), (23), and (26), we can easily obtain the result in eq. (20).

Q.E.D.

Proposition 2. If $k(x)$, Ψ, and $k^*(x)$ all satisfy the conditions given in Proposition 1, then the $k^*(x)$ can be used in morphological dilation on a binary image f. The desired result $(f \oplus k)(x)$ can be obtained by

$$(f \oplus k)(x) = C - \Psi^{-1}[C - (f \oplus k^*)(x)], \tag{27}$$

for all nonzero values in $(f \oplus k^*)(x)$, where "C" represents the foreground value. Those zeros in $(f \oplus k^*)(x)$, hence, remain zeros in $f \oplus k(x)$. The proof which is similar to the proof in Proposition 1, is skipped. *Proposition 3.* The result of a binary image f eroded by a geometric-shaped structuring element k is an inverted geometric shape of k within the foreground domain. As shown in Fig. 5, those slopes eccentrically from the center in a convex structuring element k are $|m_1|$, $|m_2|$, and $|m_3|$; after erosion, its shape is inverted to become concave, hence the slopes eccentrically from the center become $|m_3|$, $|m_2|$, and $|m_1|$. [*Proof*]: From eq. (22), the erosion is the minimum of negated values of the structuring element. The weights in the geometric-shaped structuring element are designed as the negative of the heights counting from the point to the horizontal line which is tangent to the toppest location (i.e. the center) of the structuring element. Since k is decreasing and has values all negative except the center zero, the result of minimum selection will be the coordinate z' in $k(z')$ such that $x+z'$ is located in the background and z' is the closest to the center of the structuring element. Hence, if we trace the resulting erosion curve from the boundary toward the center of an object, it is equivalent to trace the geometric curve of the structuring element from its center outwards. In other words, the erosion curve is an inverted geometric shape of the structuring element.

Q.E.D.

Proposition 4. The result of a binary image f dilated by a geometric-shaped structuring element k is the geometric shape of k located in the the background initiating from the object boundary toward background. As shown in Fig. 5, those slopes eccentrically from the center in a convex structuring element k are $|m_1|$, $|m_2|$, and $|m_3|$; after dilation, its shape starting from the edge of the object falls down by following the same sequence of slopes $|m_1|$, $|m_2|$, and $|m_3|$ correspondingly. The proof which is similar to the proof in Proposition 3, is skipped.

Proposition 5. Morphological operations by a 1D eccentrically decreasing, geometric-shaped structuring element k, can be implemented by the operations by a linearly-sloped structuring element. [*Proof*]: Select $\Psi = |k^{-1}|$ in Proposition 1. Then $k^*(x) = -\Psi[|k(x)|] = -|k^{-1}|(|k(x)|) = -x$, that is a linearly-sloped structuring element. Hence, the morphological erosion and dilation can apply the eqs. (20) and (27).

Q.E.D.

Example 1: Let a cosine curve be $y = cos(x)$, where $|x| \leq \pi$. We quantize the cosine function in intervals of $\frac{\pi}{4}$ as the following: when x is

$$[\,-\pi \quad -\frac{3}{4}\pi \quad -\frac{\pi}{2} \quad -\frac{\pi}{4} \quad 0 \quad \frac{\pi}{4} \quad \frac{\pi}{2} \quad \frac{3}{4}\pi \quad \pi\,],$$

y is equal to

$$[\,-1 \;\; -0.7 \;\; 0 \;\; 0.7 \;\; 1 \;\; 0.7 \;\; 0 \;\; -0.7 \;\; -1\,].$$

It is a combination of convex and concave structuring components, such that when $|x| \le \frac{\pi}{2}$, it is convex, and when $\frac{\pi}{2} \le |x| \le \pi$, it is concave. The structuring element constructed by $k(x) = cos\,(x) - 1$, can be expressed as

$$k\colon\ [\,-2 \;\; -1.7 \;\; -1 \;\; -0.3 \;\; 0 \;\; -0.3 \;\; -1 \;\; -1.7 \;\; -2\,]\,.$$

The mathematical transformation defined as $\Psi[\,|k(x)|\,] = 10^{|k(x)|} - 1$, can satisfy the conditions in Proposition 1. By applying $k^* = -\,\Psi[\,|k(x)|\,]$, we can obtain

$$k^*\colon\ [\,-99 \;\; -49 \;\; -9 \;\; -1 \;\; 0 \;\; -1 \;\; -9 \;\; -49 \;\; -99\,]\,.$$

Now, it is observed that k^* is convex. Hence, k^* can be decomposed using dilations of smaller structuring components, such that $k^* = k_1 \oplus k_2 \oplus k_3 \oplus k_4$, where

$$k_1 = [-1 \;\; 0 \;\; -1], \quad k_2 = [-8 \;\; 0 \;\; -8], \quad k_3 = [-40 \;\; 0 \;\; -40], \text{ and } k_4 = [-50 \;\; 0 \;\; -50]\,.$$

Hence,

$$f \ominus k^* = (((f \ominus k_1) \ominus k_2) \ominus k_3) \ominus k_4. \tag{28}$$

The result of $f \ominus k$ can be obtained by using eq. (20), such that

$$f \ominus k = \Psi^{-1}[\,f \ominus k^*\,] = \log_{10}[\,(\,f \ominus k^*) + 1\,\,], \tag{29}$$

for all nonzero values in $f \ominus k^*$. *Example 2*: Let a parabolic structuring element be $k(x) = -x^2$, where a is equal to $\frac{1}{4}$. We quantize the parabolic function as the following:

when x is

$$[-4\ -3\ -2\ -1\ 0\ 1\ 2\ 3\ 4],$$

y is equal to

$$[-16\ -9\ -4\ -1\ 0\ -1\ -4\ -9\ -16].$$

It is clearly a convex structuring element. The mathematical transformation defined as $\Psi[|k(x)|] = |k(x)|^{1/2}$, can satisfy the conditions in Proposition 1. By applying $k^* = -\Psi[|k(x)|] = -x$, we can obtain

$$k^*: [-4\ -3\ -2\ -1\ 0\ -1\ -2\ -3\ -4].$$

Now, the structuring element k^* is linearly-sloped. Hence k^* can be decomposed using iterative dilations of a structuring component of size 3, such that $k^* = k_1 \oplus k_1 \oplus k_1 \oplus k_1$, where

$$k_1 = [-1\ 0\ -1].$$

Hence,

$$f \ominus k^* = (((f \ominus k_1) \ominus k_1) \ominus k_1) \ominus k_1. \tag{30}$$

The result of $f \ominus k$ can be obtained by using eq. (32), such that

$$f \ominus k = \Psi^{-1}[f \ominus k^*] = (f \ominus k^*)^2, \tag{31}$$

for all nonzero values in $f \ominus k^*$.

1.5 2D GEOMETRIC-SHAPED STRUCTURING ELEMENTS DECOMPOSITION

1.5.1 Hemisphere, Hemiellipsoid, Gaussian and Elliptic Paraboloid

A 2D geometric-shaped structuring element $k(x,y)$ is actually the geometric surface: $f(x,y,z) = 0$ in a 3D Euclidean space. We assign each point in the structuring element by the negated distance from this point to the horizontal plane tangent to the top of the surface in order to ensure that $k(x,y)$ is decreasing. A hemisphere can be represented implicitly by

$$f(x,y,z) = x^2 + y^2 + z^2 - r^2 = 0 , \tag{32}$$

where $|x| \leq r$, $|y| \leq r$, $0 \leq z \leq r$, and r denotes the radius. Eq. (32) can be rewritten in the explicit form of

$$z = \sqrt{r^2 - x^2 - y^2} . \tag{33}$$

From Definition 1 in Section 2, it is known that the hemisphere is an eccentrically decreasing structuring element. For computational simplicity, a constant "$-r$" is added to suppress the z-value at the origin to zero. Hence,

$$k(x,y) = \sqrt{r^2 - x^2 - y^2} - r . \tag{34}$$

The similar method can also apply to other geometric-shaped structuring elements as listed in Table 2.

The presented 2D geometric-shaped structuring elements are all convex except the Gaussian which is combined convex and concave. Since all of them are not additively-separable, they can be decomposed only using the maximum selection of several segmented structuring components [5], such decomposition is not efficient in the calculation. In the following section, we present a strategy to decompose them into the dilations

Table 2. 2D Geometric-Shaped Structuring Elements

Name	Representation	Structuring Element
Hemiellipsoid	$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$	$c\left[1-\frac{x^2}{a^2}-\frac{y^2}{b^2}\right]^{1/2}-c$
Gaussian Surface	$z=\frac{1}{\sqrt{2\pi}}e^{-(x^2+y^2)/2}$	$\frac{1}{\sqrt{2\pi}}e^{-(x^2+y^2)/2}-\frac{1}{\sqrt{2\pi}}$
Elliptic Paraboloid	$\frac{x^2}{a^2}+\frac{y^2}{b^2}+cz=0$	$-\frac{x^2}{c\,a^2}-\frac{y^2}{c\,b^2}$
Hemihyperboloid	$\frac{z^2}{c^2}-\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$	$-c\left[\frac{x^2}{a^2}+\frac{y^2}{b^2}+1\right]^{1/2}+c$

of smaller structuring components

1.5.2 Decomposition Strategy

Proposition 6 A 2D eccentrically decreasing, geometric-shaped structuring element $k(x,y)$ can be represented by $k(x,y)=g(p(x)+q(y))+c$, where $g(\,)$ represents a geometric-shaped function, $p(x)$ and $q(y)$ are functions of only x and y, respectively, and c is a constant. If the function $g(\,)$ and a transformation Ψ satisfy that for all $0<x_1<x_2$ and $x_2<x_1<0$, there exists $g(p(x_1)+q(y))>g(p(x_2)+q(y))$ and $0\le\Psi[|k(x_1,y)|]<\Psi[|k(x_2,y)|]$, and for all $0<y_1<y_2$ and $y_2<y_1<0$, there exists $g(p(x)+q(y_1))>g(p(x)+q(y_2))$ and $0\le\Psi[|k(x,y_1)|]<\Psi[|k(x,y_2)|]$. Then we can get a new structuring element k^*, in which $k^*(x,y)=-\Psi[|k(x,y)|]$, where $k^*(x,y)\le 0$. The $k^*(x,y)$ can be used in morphological erosion and dilation on a binary image f, so that the desired result $(f\ominus k)(x,y)$ can be obtained by

$$(f\ominus k)(x,y)=\Psi^{-1}[\,(f\ominus k^*)(x,y)\,], \tag{35}$$

for all nonzero values in $(f \ominus k^*)(x,y)$, and $(f \oplus k)(x,y)$ can be obtained by

$$(f \oplus k)(x,y) = C - \Psi^{-1}[C - (f \oplus k^*)(x,y)]\,, \tag{36}$$

for all nonzero values in $(f \oplus k^*)(x,y)$. The proof which is similar to the proof in Proposition 1 in the 1D case, is skipped. *Proposition 7.* If a 2D eccentrically decreasing, geometric-shaped structuring element $k(x,y)$ which satisfies that $k(x,y) = g(\,p(x) + q(y)\,) + c$, where $g(\,)$ represents a geometric-shaped function, $p(x)$ and $q(y)$ are both convex and in the same sign, and c is a constant, then k can be decomposed using dilations of smaller structuring components through a mathematical transformation. [*Proof*]: Select a transformation $\Psi = |\,g^{-1}\,|$ in Proposition 6. Then we have

$$k^*(x,y) = -\,\Psi[\,|k(x,y)|\,] = -\,\Psi[\,|g(\,p(x) + q(y)\,) + c\,|\,] = -|p(x) + q(y) + c\,|. \tag{37}$$

Since $p(x)$ and $q(y)$ are convex and in the same sign, we can obtain

$$k^*(x,y) = -|p(x)| - |q(y)| \pm c\,. \tag{38}$$

It is clearly that $k^*(x,y)$ is a additively-separable convex structuring element, hence it can be decomposed using dilations of smaller structuring components according to Property 5 in Section 3.

Q.E.D.

Beside the above propositions, the Propositions 3 and 4 in Section 4B for the 1D case also can be applied to 2D. An example of the decomposition of a hemispheric-shaped structuring element is given below. *Example 3:* A 3D hemispheric surface is given by placing $r = 4$ in eq. (33), then we have

$$z = \sqrt{16 - x^2 - y^2}\,. \tag{39}$$

The hemispheric-shaped structuring element can be constructed as

$$k(x,y) = \sqrt{16 - x^2 - y^2} - 4 = -(4 - \sqrt{16 - x^2 - y^2}). \tag{40}$$

The mathematical transformation defined as

$\Psi[|k(x,y)|] = -((-(|k(x,y)| - 4))^2 - 16)$

can satisfy the conditions in Proposition 6. By applying $k^* = -\Psi[|k(x,y)|]$, we can obtain

$$k^*(x,y) = -x^2 - y^2. \tag{41}$$

The new structuring element k^* is additively-separable and convex and it can be numerically represented as

$$k^*: \begin{array}{ccccccc} & -13 & -10 & -9 & -10 & -13 & \\ -13 & -8 & -5 & -4 & -5 & -8 & -13 \\ -10 & -5 & -2 & -1 & -2 & -5 & -10 \\ -9 & -4 & -1 & 0 & -1 & -4 & -9 \\ -10 & -5 & -2 & -1 & -2 & -5 & -10 \\ -13 & -8 & -5 & -4 & -5 & -8 & -13 \\ & -13 & -10 & -9 & -10 & -13 & \end{array}$$

Hence, k^* can be decomposed as: $k^* = k_1 \oplus k_2 \oplus k_3$, where

$$k_1 = \begin{array}{ccc} -2 & -1 & -2 \\ -1 & 0 & -1 \\ -2 & -1 & -2 \end{array}, \quad k_2 = \begin{array}{ccc} -6 & -3 & -6 \\ -3 & 0 & -3 \\ -6 & -3 & -6 \end{array}, \quad k_3 = \begin{array}{ccc} & -5 & \\ -5 & 0 & -5 \\ & -5 & \end{array}.$$

The result of $f \ominus k$ can be obtained by using eq. (35), such that

$$f \ominus k = \Psi^{-1}[f \ominus k^*] = -\sqrt{-(f \ominus k^*) + 16} + 4. \tag{42}$$

1.6 DECOMPOSITION OF A LARGE TRANSFORMED CYCLIC COSINE STRUCTURING ELEMENT

In Fig. 2, we mention that a large cyclic cosine structuring element (i.e. nondecreasing) has no significant meaning, because its morphological operations will result in a constant image. However, in image processing and computer graphics, it is sometimes useful to form some geometric-shaped surfaces onto a xy-plane. By applying a mathematical transformation, the nondecreasing structuring element can be converted into a decreasing element. Therefore, the morphological erosion by the new transformed structuring element followed by an inverse transformation may yield such a geometric-shaped surface.

A 2D cyclic cosine structuring element $k(x,y)$ is given by

$$k(x,y) = \cos(\sqrt{x^2+y^2}) - 1 . \tag{43}$$

When the domain of $\sqrt{x^2+y^2}$ exceeds a cycle 2π, $k(x,y)$ will become nondecreasing. By applying a transformation $\Psi[\,|k(x,y)|\,] = (\cos^{-1}(-(|k(x,y)|-1)))^2$, a new structuring element is constructed, such that $k^*(x,y) = -\Psi[\,|k(x,y)|\,] = -x^2 - y^2$ which is decreasing and convex. The inverse transformed morphological erosion $\Psi^{-1}[f \ominus k^*]$, as illustrated in Fig. 6, is equal to

$$\Psi^{-1}[f \ominus k^*] = -\cos(\theta\sqrt{f \ominus k^*}) + 1 , \tag{44}$$

where θ is an angle which $x = \theta$ in the cosine function represents $x = 1$ in k^*. In the following decomposition, we describe the procedure by illustrating a 17×17 2D cyclic cosine structuring element. Many large-sized 1D or 2D cyclic geometric-shaped structuring elements can be obtained by applying the similar procedure.

Since the new structuring element $k^*(x,y) = -(x^2+y^2)$ which is symmetric to its center, it can be numerically expressed for simplicity by its upper-right quadrant using the unit $\theta = \frac{\pi}{4}$ in the cosine function as the following:

$$
\begin{array}{rrrrrrrrr}
-64 & -65 & -68 & -73 & -80 & -89 & -100 & -113 & -128 \\
-49 & -50 & -53 & -58 & -65 & -74 & -85 & -98 & -113 \\
-36 & -37 & -40 & -45 & -52 & -61 & -72 & -85 & -100 \\
-25 & -26 & -29 & -34 & -41 & -50 & -61 & -74 & -89 \\
-16 & -17 & -20 & -25 & -32 & -41 & -52 & -65 & -80 \\
-9 & -10 & -13 & -18 & -25 & -34 & -45 & -58 & -73 \\
-4 & -5 & -8 & -13 & -20 & -29 & -40 & -53 & -68 \\
-1 & -2 & -5 & -10 & -17 & -26 & -37 & -50 & -65 \\
0 & -1 & -4 & -9 & -16 & -25 & -36 & -49 & -64
\end{array}
$$

The erosion of a binary image f by the structuring element k^* is:

$$f \ominus k^* = (((((((f \ominus k_1) \ominus k_2) \ominus k_3) \ominus k_4) \ominus k_5) \ominus k_6) \ominus k_7) \ominus k_8 \,, \tag{45}$$

where

$$k_1 = \begin{bmatrix} -2 & -1 & -2 \\ -1 & 0 & -1 \\ -2 & -1 & -2 \end{bmatrix}, \quad k_2 = \begin{bmatrix} -6 & -3 & -6 \\ -3 & 0 & -3 \\ -6 & -3 & -6 \end{bmatrix}, \quad k_3 = \begin{bmatrix} -10 & -5 & -10 \\ -5 & 0 & -5 \\ -10 & -5 & -10 \end{bmatrix},$$

$$k_4 = \begin{bmatrix} -14 & -7 & -14 \\ -7 & 0 & -7 \\ -14 & -7 & -14 \end{bmatrix}, \quad k_5 = \begin{bmatrix} -18 & -9 & -18 \\ -9 & 0 & -9 \\ -18 & -9 & -18 \end{bmatrix}, \quad k_6 = \begin{bmatrix} -22 & -11 & -22 \\ -11 & 0 & -11 \\ -22 & -11 & -22 \end{bmatrix},$$

$$k_7 = \begin{bmatrix} -26 & -13 & -26 \\ -13 & 0 & -13 \\ -26 & -13 & -26 \end{bmatrix}, \quad \text{and} \quad k_8 = \begin{bmatrix} -30 & -15 & -30 \\ -15 & 0 & -15 \\ -30 & -15 & -30 \end{bmatrix}.$$

Hence,

$$\Psi^{-1} [\, f \ominus k^* \,] = -\cos\left[\frac{\pi}{4} \sqrt{f \ominus k^*} \right] + 1 \,, \tag{46}$$

as shown in Fig. 6.

1.7 DECOMPOSITION OF 2D STRUCTURING ELEMENTS INTO 1D ELEMENTS

According to Proposition 6 in Section 5B, we observe that many 2D geometric-shaped structuring elements in the form of $k(x,y) = g\,(\,p(x) + q(y)\,) + c$, can be transformed into $k^*(x,y)$ which is additively-separable and convex. This kind of structuring element $k^*(x,y)$ can be further decomposed into a dilation of two 1D structuring components, one in x-direction and the other in y-direction, and both are convex. Furthermore, these 1D convex structuring elements can be again decomposed into a series of dilations of size 3 structuring subcomponents according to Property 5 in Section 3. *Example 4*: Let $k^*(x,y)$ be

$$k^* = \begin{bmatrix} -8 & -5 & -4 & -5 & -8 \\ -5 & -2 & -1 & -2 & -5 \\ -4 & -1 & 0 & -1 & -4 \\ -5 & -2 & -1 & -2 & -5 \\ -8 & -5 & -4 & -5 & -8 \end{bmatrix} .$$

The $k^*(x,y)$ can be decomposed as $k^*(x,y) = k_1(x) \oplus k_2(y)$, where

$$k_1(x) = [\,-4 \;\; -1 \;\; 0 \;\; -1 \;\; -4\,]\,, \quad k_2(y) = \begin{bmatrix} -4 \\ -1 \\ 0 \\ -1 \\ -4 \end{bmatrix} .$$

The $k_1(x)$ and $k_2(y)$ can be further decomposed into

$$k_1(x) = [\,-1 \;\; 0 \;\; -1\,] \oplus [\,-3 \;\; 0 \;\; -3\,]\,,$$

$$k_2(y) = \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix} \oplus \begin{bmatrix} -3 \\ 0 \\ -3 \end{bmatrix} .$$

1.8 CONCLUSIONS

In this chapter, we have presented a unified decomposition technique by the use of a mathematical transformation. The method can apply to all kinds of 1D gray-scale structuring elements to decompose them into dilations of smaller structuring components. For 2D gray-scale structuring elements, if they are constructed by a geometric function of two additively-separable functions of x and y individually, then the decomposition using dilations of smaller structuring components can be achieved. Besides, a significant new result of applying the transformation to a large, cyclic, nondecreasing structuring element converted into another kind of the decreasing element and then performing erosion to construct the structuring-shaped surface is developed. Finally, the strategy of decomposing 2D additively-separable convex structuring elements into 1D elements is also developed. The decomposition into 1D structuring elements will have the advantage of less calculations required than the decomposition into 2D smaller structuring elements. These techniques allow a user to implement his morphological algorithms and architectures with more freedom in choosing any kind or any size of structuring elements and in the achievement of real-time feasibility.

CHAPTER 2

OPTIMIZATION ON EUCLIDEAN DISTANCE TRANSFORMATION USING GRAYSCALE MORPHOLOGY

2.1 INTRODUCTION

A distance transformation converts a digital binary image which consists of object (foreground) and non-object (background) pixels, into a gray-level image in which all object pixels have a value corresponding to the minimum distance from the background. Computing the distance from a pixel to a set of background pixels is in principle a global operation. It is often prohibitively costly except that the digital image is very small. Among many types of distance measures in digital image processing, city-block and chessboard distances are easy to compute and they can be recursively accumulated by considering only a small neighborhood at a time. However, both distance measures are very sensitive to the orientation of the object.

The Euclidean distance measurement is very useful in object recognition and inspection because of the metric accuracy and rotation invariance. However, the global operation is difficult to decompose into small neighborhood operations because of the nonlinearity of Euclidean distance computation. Hence, the algorithms concerning the approximation of the Euclidean distance transformation are extensively discussed [9,10,11,15,16,23].

Borgefors [2] optimized the local distances used in 3×3, 5×5, and 7×7 neighborhoods by minimizing the maximum of the absolute value of the difference between her proposed distance transformation and the exact Euclidean distance transformation. An accurate Euclidean distance transformation by the morphological decomposition was developed by Shih and Mitchell [20]. They utilized the mathematical morphology properties and decomposed the Euclidean distance transformation into successive erosions by small structuring components. Their computational complexity is the order of the object size. In this chapter we improve the decomposition strategy by analyzing the square of the Euclidean distance structuring element. It requires only four erosions without iterative operations and is independent of the object width.

Shih, King and Pu [21] defined a new concept called *back-propagation morphology*, which is different from the traditionally defined morphology called *forward morphology*. Unlike the forward morphological operations, the back-propagation morphological operations intend to feed back the output at each pixel during image scanning to overwrite its input and to continue in the same way until all the pixels are scanned. They developed a two-scan algorithm using the back-propagation morphology to derive the root generation of a signal without recursively applying such forward morphology. The operation is independent of the object size and significantly saves much of the computational time as compared with the algorithms proportional to the number of iterations in forward morphology.

The grayscale morphology can be defined using the concepts of surface of a set and the umbra of a surface [12,13,6]. Suppose a set A in Euclidean N-space (E^N is given. Let F and K be $\{x \in E^{N-1} \mid \text{for } some\ y \in E,\ (x,y) \in A\}$ *and be the domains of the grayscale image f and the grayscale structuring element k, respectively. In distance transformation of using graysclae morphology, the image f is limited to positive integers and the structuring element k to negative integers. The grayscale dilation* of f by k, denoted by $f \oplus_g k$, is defined as:

$$(f \oplus_g k)(x,y) = \sup \{ f(x-m, y-n) + k(m,n) \}, \qquad (1)$$

for all $(m,n) \in K$ *and* $(x-m, y-n) \in F$. *The grayscale erosion* of f by k, denoted by $f \ominus_g k$, is defined as:

$$(f \ominus_g k)(x,y) = \inf \{ f(x+m, y+n) - k(m,n) \}, \qquad (2)$$

for all $(m,n) \in K$ *and* $(x+m,y+n) \in F$. *Note that the subscript "g" indicating the grayscale will be dropped out, since we are only concerned with the grayscale morphological operations in this chapter.*

Let $N(O)$ be the set of (4- or 8-) neighbors that precede the central pixel O and itself in a scanning sequence of the picture within the window of a structuring element. Then the algorithm of *back-propagation dilation* of f by k, denoted by $f \oplus k$, is as follows:

$$(f \oplus k)(x,y) = \sup \{ [f \oplus k(x-m,y-n)] + k(m,n) \}, \tag{3}$$

for all $(m,n) \in K$ and $(x-m,y-n) \in (N \cap F)$. Then the algorithm of *back-propagation erosion* of f by k, denoted by $f \ominus k$, *is as follows:*

$$(f \ominus k)(x,y) = \inf \{ [f \ominus k(x+m,y+n)] - k(m,n) \}, \tag{4}$$

for all $(m,n) \in K$ *and* $(x+m,y+n) \in (N \cap F)$. *Readers should note that the back-propagation morphological operations are identical to sequential operations proposed by Rosenfeld and Pfaltz [17] if they are applied to distance transformation.*

Since the back-propagation dilation (erosion) adopts the dilated (eroded) results of the preceding scanned neighbors to be involved in its computation, its output inherently depends on the image scanning sequence. In general, an image scanning can be classified into one-dimensional (1D) and two-dimensional (2D) scans. In moving from point to point, we always move from a point to one of its eight neighbors. The moving direction is measured counterclockwise from the positive x-axis. Thus, the 1D scan can be separated into eight directions: 1) "L" denotes left-to-right or 0°; 2) "R" denotes right-to-left or 180°; 3) "T" denotes top-to-bottom or 270°; 4) "B" denotes bottom-to-top or 90°; 5) "45°"; 6)"135°"; 7)"225°"; 8)"315°." The 2D scan can have four scanning sequences starting from a corner pixel: 1) "LT" denotes left-to-right and top-to-bottom

(Note: this is a usual television raster scan.); 2) ``*RB*'' denotes right-to-left and bottom-to-top; 3) ``*LB*'' denotes left-to-right and bottom-to-top; 4) ``*RT*'' denotes right-to-left and top-to-bottom.

Assume a 3×3 structuring element k is denoted by

$$k = \begin{bmatrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & A_9 \end{bmatrix}. \tag{5}$$

Because of the back-propagation effect with respect to the scanning direction, all nine elements $A_1, \cdots, A_9$ actually are not used at a time in computation. The redefined structuring element for the back-propagation morphology must satisfy the following criterion: wherever a pixel is being dealt with, all its neighbors defined within the structuring element k must have already been visited before by the specified scanning sequence. Thus, the k in one dimension is redefined as:

$$k_L = [\, A_4 \;\; A_5 \;\; \mathrm{x} \,], \quad k_R = [\, \mathrm{x} \;\; A_5 \; A_6 \,], \quad k_T = \begin{bmatrix} A_2 \\ A_5 \\ \mathrm{x} \end{bmatrix}, \quad k_B = \begin{bmatrix} \mathrm{x} \\ A_5 \\ A_8 \end{bmatrix},$$

$$k_{45^\circ} = \begin{bmatrix} & & \mathrm{x} \\ & A_5 & \\ A_7 & & \end{bmatrix}, \quad k_{135^\circ} = \begin{bmatrix} \mathrm{x} & & \\ & A_5 & \\ & & A_9 \end{bmatrix}, \quad k_{225^\circ} = \begin{bmatrix} & & A_3 \\ & A_5 & \\ \mathrm{x} & & \end{bmatrix}, \quad k_{315^\circ} = \begin{bmatrix} A_1 & & \\ & A_5 & \\ & & \mathrm{x} \end{bmatrix}; \tag{6}$$

the k in two dimensions is redefined as:

$$k_{LT} = \begin{bmatrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \end{bmatrix}, \quad k_{RB} = \begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & A_5 & A_6 \\ A_7 & A_8 & A_9 \end{bmatrix}, \quad k_{LB} = \begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} \\ A_4 & A_5 & \mathrm{x} \\ A_7 & A_8 & A_9 \end{bmatrix}, \quad k_{RT} = \begin{bmatrix} A_1 & A_2 & A_3 \\ \mathrm{x} & A_5 & A_6 \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \end{bmatrix}, \tag{7}$$

where ``x'' means *don't care* or it can be defined as ``$-\infty$'' according to the mathematical morphology properties [12,19,2].

A complete omnidirectional scanning in the image space can be achieved by using eight-scan in one dimension or two-scan in two dimensions. The eight-scan method adopts eight back-propagation morphological filters sequentially in a non-specific order of eight scanning directions, L, R, T, B, 45°, 135°, 225°, 315° working on the corresponding structuring elements k_L, k_R, k_T, k_B, $k_{45°}$, $k_{135°}$, $k_{225°}$, $k_{315°}$, respectively. The two-scan method adopts two back-propagation morphological filters, the first one being any of the four 2D scans LT, RB, LB, RT working on its corresponding structuring element, and the second one being the opposite scanning sequence of the first scan and its scan-related structuring element. For example, the opposite scanning of LT is RB.

In this chapter, we present three algorithms to the Euclidean distance transformation in digital images by the use of the grayscale morphological erosion with the squared Euclidean distance structuring element. The theoretical verification is provided in Section 2. Section 3 describes a 1D decomposition algorithm for the squared Euclidean distance structuring element. Section 4 gives another 2D decomposition algorithm using iterative erosions. In Section 5, we present an optimal double two-scan algorithm. The example and computational complexity of each algorithm are also provided in each section.

2.2 EUCLIDEAN DISTANCE TRANSFORMATION USING THE SQUARE OF THE EUCLIDEAN DISTANCE STRUCTURING ELEMENT

There are six types of distance measures discussed by Borgefors in [10]: City-block, Chessboard, Euclidean, Octagonal, Chamfer 3-4, and Chamfer 5-7-11. The morphological approach to the Euclidean distance transformation presented here is a general method which uses a pre-defined distance structuring element to obtain the expected type of the distance measure.

By defining the central point O of the structuring element as the origin in the

(x, y)-coordinates, we can represent the value of each element by the negative of its distance from O (so that O is represented as 0). We select the element to be the negative of the related distance measures because the grayscale erosion is a minimum selection after subtraction. This gives a positive distance measure.

The Euclidean distance structuring element, denoted by $k(x,y) = -\sqrt{x^2+y^2}$, can be expressed as follows:

$$\begin{bmatrix} -\sqrt{18} & -\sqrt{13} & -\sqrt{10} & -3 & -\sqrt{10} & -\sqrt{13} & -\sqrt{18} \\ -\sqrt{13} & -\sqrt{8} & -\sqrt{5} & -2 & -\sqrt{5} & -\sqrt{8} & -\sqrt{13} \\ -\sqrt{10} & -\sqrt{5} & -\sqrt{2} & -1 & -\sqrt{2} & -\sqrt{5} & -\sqrt{10} \\ -3 & -2 & -1 & 0 & -1 & -2 & -3 \\ -\sqrt{10} & -\sqrt{5} & -\sqrt{2} & -1 & -\sqrt{2} & -\sqrt{5} & -\sqrt{10} \\ -\sqrt{13} & -\sqrt{8} & -\sqrt{5} & -2 & -\sqrt{5} & -\sqrt{8} & -\sqrt{13} \\ -\sqrt{18} & -\sqrt{13} & -\sqrt{10} & -3 & -\sqrt{10} & -\sqrt{13} & -\sqrt{18} \end{bmatrix}$$

The structuring element k can be decomposed using the concave decomposition strategy [5]. In other words, the structuring element can be decomposed into the maximum selection of several structuring components and thus needs more operations than the convex decomposition in which the structuring element can be decomposed into successive dilations of these segmented linearly-sloped structuring components. The square of the Euclidean distance structuring element, denoted by $k^2(x,y) = -(x^2+y^2)$, is expressed as follows [14,5]:

In k^2 each absolute value is a square of the absolute value in k. The Euclidean distance transformation of a binary image f is to erode the binary image f by the distance structuring element k. Since the binary image f is represented by two values "$+\infty$" and

"0," we have from eq. (2)

$$\begin{bmatrix} -32 & -25 & -20 & -17 & -16 & -17 & -20 & -25 & -32 \\ -25 & -18 & -13 & -10 & -9 & -10 & -13 & -18 & -25 \\ -20 & -13 & -8 & -5 & -4 & -5 & -8 & -13 & -20 \\ -17 & -10 & -5 & -2 & -1 & -2 & -5 & -10 & -17 \\ -16 & -9 & -4 & -1 & 0 & -1 & -4 & -9 & -16 \\ -17 & -10 & -5 & -2 & -1 & -2 & -5 & -10 & -17 \\ -20 & -13 & -8 & -5 & -4 & -5 & -8 & -13 & -20 \\ -25 & -18 & -13 & -10 & -9 & -10 & -13 & -18 & -25 \\ -32 & -25 & -20 & -17 & -16 & -17 & -20 & -25 & -32 \end{bmatrix}$$

$$(f \ominus k)(x) = \min(\inf\{\infty - k(z) \mid z \in K \text{ and } x+z \in F\}, \tag{8}$$

$$\inf\{-k(z') \mid z' \in K \text{ and } x+z' \in F\}).$$

That is all z's satisfy that $x + z$ is located within the object and all z''s satisfy that $x+z'$ is located within the background. Because the differences of all $k(z)$ and $k(z')$ are finite, "$\infty - k(z)$" *is greater than* "$0 - k(z')$" *all the time. Hence, we have*

$$(f \ominus k)(x) = \inf\{-k(z') \mid z' \in K \text{ and } x+z' \in F\}. \tag{9}$$

Because the values in the structuring element are the negative of the distance measure from the center of the structuring element, the binary image f eroded by the distance structuring element k is exactly equal to the minimum distance from the object point to the background.

If k^2 is used as the distance structuring element, we have

$$(f \ominus k^2)(x) = \min(\inf\{\infty - k^2(z_1) \mid z_1 \in K \text{ and } x+z_1 \in F\},$$

$$\inf\{-k^2(z_1') \mid z_1' \in K \text{ and } x+z_1' \in F\}). \tag{10}$$

Because the differences of all $k^2(z)$ and $k^2(z')$ are finite, "$0 - k^2(z')$" is always smaller than "$\infty - k^2(z)$." Hence, we obtain

$$(f \ominus k^2)(x) = \inf\{-k^2(z_1') \mid z_1' \in K \text{ and } x + z_1' \in F\}\ . \tag{11}$$

Hence, each z' in eq. (9) is the same as the z_1' in eq. (11) for the same image f. That is $z' = z_1'$. Because $-k(z') \geq 0$ and $-k^2(z_1') \geq 0$, we have $-k^2(z_1') = [\,-k(z')\,]^2$ for all the object pixels. Because all the background pixels in the image f are zeros, they all satisfy that $-k^2(z_1') = [\,-k(z')\,]^2$. Hence, we can obtain

$$\inf\{-k(z') \mid z \in K \text{ and } x + z \in F\} = \sqrt{\inf\{-k^2(z_1') \mid z_1' \in K \text{ and } x + z_1' \in F\}}\ , \tag{12}$$

which is always true for all the pixels in the image f. Therefore, we conclude that

$$f \ominus k = \sqrt{f \ominus k^2}\ . \tag{13}$$

The derived result in eq. (13) shows that when the Euclidean distance transformation is performed, the squared structuring element k^2 can be used as the structuring element in the morphological erosion. A square-root of the result obtained by k^2 is exactly the same as the desired distance transformation by k. A structuring element $k(x,y)$ is called *additively-separable* if $k(x,y) = k_1(x) + k_2(y)$. The structuring element k^2 has the advantage that it is additively-separable and convex [5], so that it can be easily decomposed as dilations of small structuring components.

2.3 1D DECOMPOSITION ALGORITHM FOR THE SQUARED EUCLIDEAN DISTANCE STRUCTURING ELEMENT

The Euclidean distance structuring element can not be decomposed into the morphological dilations of smaller structuring components, since it poses a non-additively separable problem in the off-axis and off-diagonal directions. However, if we square each pixel's value of the structuring element, i.e., k^2, then it becomes convex and additively separable. Because of the simple decomposition property for a convex and additively separable structuring element, applying k^2 to implement the Euclidean distance transformation is much faster and more efficient. After the image is eroded by the structuring elements k^2, the exact Euclidean distance values can be obtained by taking a square-root on the eroded output.

There are several kinds of decomposition strategy which can be applied to k^2. One is to decompose k^2 into a dilation of two 1D structuring components, and the others will be presented in Sections 4 and 5. Assuming that the object has the size $2N \times 2M$, the distance structuring element must have a larger size than the object. Suppose the values outside the defined structuring element are considered as "$-\infty$." Thus, the dilation of a 1D horizontal structuring element $k_1(x)$ with a 1D vertical one $k_2(y)$ will construct a 2D structuring element $k(x,y)$. The Euclidean distance transformation (EDT) can be obtained by

$$EDT = \sqrt{(f \ominus k^2)(x, y)}\ , \quad \text{where} \quad k^2(x,y) = k_1(x) \oplus k_2(y)\,. \tag{14}$$

The $k_1(x)$ and $k_2(y)$ can be further decomposed as

$$k_1(x) = k_{11}(x) \oplus k_{12}(x) \oplus \cdots \oplus k_{1N}(x)\,, \tag{15}$$

$$k_2(y) = k_{21}(y) \oplus k_{22}(y) \oplus \cdots \oplus k_{2M}(y). \tag{16}$$

Furthermore, one of the two iterative erosions in $k_1(x)$ and $k_2(y)$ can be performed by using the two-scan back-propagation erosion. Since $k_1(x) = -x^2$ and $k_2(y) = -y^2$ are convex, we can take a square-root of them to obtain the linearly-sloped structuring elements [19]. That is $k_1^*(x) = -x$ and $k_2^*(y) = -y$. The iterative operations of linearly-sloped structuring elements can be achieved by using the two-scan algorithm mentioned in Section 1. That is, the $k_1^*(x)$ and $k_2^*(y)$ can be separated into $\{k_{11}^*(x), k_{12}^*(x)\}$ and $\{k_{21}^*(y), k_{22}^*(y)\}$, respectively, according to eq. (6). Hence,

$$EDT = \begin{cases} \sqrt{(f \ominus k_{11}^* \ominus k_{12}^*)^2 \ominus k_2}\ , & \text{if } N \geq M \\ \sqrt{(f \ominus k_{21}^* \ominus k_{22}^*)^2 \ominus k_1}\ , & \text{if } N < M\ . \end{cases} \tag{17}$$

Example 1. For expressional simplicity, the squared Euclidean distance structuring element $k^2(x,y)$ of size 7×5 is given as

$$k^2(x,y) = \begin{bmatrix} -13 & -8 & -5 & -4 & -5 & -8 & -13 \\ -10 & -5 & -2 & -1 & -2 & -5 & -10 \\ -9 & -4 & -1 & 0 & -1 & -4 & -9 \\ -10 & -5 & -2 & -1 & -2 & -5 & -10 \\ -13 & -8 & -5 & -4 & -5 & -8 & -13 \end{bmatrix} .$$

The $k^2(x,y)$ can be decomposed as $k^2(x,y) = k_1(x) \oplus k_2(y)$, where

$$k_1(x) = [\,-9\ \ -4\ \ -1\ \ 0\ \ -1\ \ -4\ \ -9\,]\,, \quad k_2(y) = \begin{bmatrix} -4 \\ -1 \\ 0 \\ -1 \\ -4 \end{bmatrix} .$$

Since k^2 have the size 7×5, we obtain $N = 3$ *and* $M = 2$. The $k_1(x)$ and $k_2(y)$ can be further decomposed into

$$k_1(x) = k_{11}(x) \oplus k_{12}(x) \oplus k_{13}(x) = [\,-1\ \ 0\ \ -1\,] \oplus [\,-3\ \ 0\ \ -3\,] \oplus [\,-5\ \ 0\ \ -5\,]\,,$$

$$k_2(y) = k_{21}(y) \oplus k_{22}(y) = \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix} \oplus \begin{bmatrix} -3 \\ 0 \\ -3 \end{bmatrix} .$$

Because N > M, the eq.(17) is applied. We have

$$EDT = \sqrt{(f \ominus k_{11}^*(x) \ominus k_{12}^*(x))^2 \ominus k_2(y)}\ ,$$

where

$$k_{11}^{*}(x) = [\,-1 \;\; 0 \;\; \mathrm{x}\,]\,, \quad k_{12}^{*}(x) = [\,\mathrm{x} \;\; 0 \;\; -1\,]\,,$$

$$k_2(y) = \begin{bmatrix} -4 \\ -1 \\ 0 \\ -1 \\ -4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix} \oplus \begin{bmatrix} -3 \\ 0 \\ -3 \end{bmatrix} .$$

Most of the Euclidean distance transformation algorithms need iterations of calculation over all pixels' values. The number of iterations is proportional to the largest length or width of the object. In other words, the number of iterations is equal to the largest chessboard distance in the distance transformation. Here we denote it as C.

2.3.1 1D Algorithm

In general, the distance transformation of a 2D binary image is a 2D global operation. It can be separated into successive 1D grayscale morphological operations. Assuming $N \geq M$, the 1D Euclidean distance transformation algorithm is described as follows.

(1) $d = f \,\tilde{\ominus}\, [\,-1 \;\; 0 \;\; \mathrm{x}\,] \,\tilde{\ominus}\, [\,\mathrm{x} \;\; 0 \;\; -1\,]$

(2) For $i = 1$, *length*, 1 do

For $j = 1$, *width*, 1 do

if ($d_{ij} > 1$) then $\quad d_{ij} = {d_{ij}}^2$

(3) $e = d \ominus \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix} \ominus \begin{bmatrix} -3 \\ 0 \\ -3 \end{bmatrix} \ominus \cdots \ominus \begin{bmatrix} -(2C-1) \\ 0 \\ -(2C-1) \end{bmatrix} = d \ominus * \begin{bmatrix} -(2l-1) \\ 0 \\ -(2l-1) \end{bmatrix}, \quad l = 1\,, 2, \cdots, C$

where " $\ominus *$" denotes the iterative erosions and l indicates the iteration number.

(4) For $i = 1$, *length*, 1 do

For $j = 1, width, 1$ do

if ($e_{ij} > 1$) then $e_{ij} = (e_{ij})^{1/2}$.

The final output e is the result of Euclidean distance transformation. Note that in the third step if the largest chessboard distance C is unknown, the iterations will stop until two successive outputs do not change any more.

Example 2. Let a binary image f be of size 12×11:

$$f = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&255&255&255&255&255&255&255&255&0&0 \\ 0&0&255&255&255&255&255&255&255&255&0&0 \\ 0&0&255&255&255&255&255&255&255&255&0&0 \\ 0&0&255&255&255&255&255&255&0&0&0&0 \\ 0&0&255&255&255&255&255&255&0&0&0&0 \\ 0&0&255&255&255&255&0&0&0&0&0&0 \\ 0&0&255&255&255&255&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}, \qquad (18)$$

where "255" represents the object pixel and "0" the background pixel. The result of exact Euclidean distance transformation should be:

$$EDT = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&1&1&1&1&0&0 \\ 0&0&1&2&2&2&2&2&2&1&0&0 \\ 0&0&1&2&3&3&\sqrt{5}&\sqrt{2}&1&1&0&0 \\ 0&0&1&2&\sqrt{8}&\sqrt{5}&2&1&0&0&0&0 \\ 0&0&1&2&\sqrt{5}&\sqrt{2}&1&1&0&0&0&0 \\ 0&0&1&2&2&1&0&0&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}. \qquad (19)$$

In the first step by applying the back-propagation morphological erosion to f with the structuring element [-1 0 x] in the left-to-right scanning, we obtain

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&2&3&4&5&6&7&8&0&0 \\ 0&0&1&2&3&4&5&6&7&8&0&0 \\ 0&0&1&2&3&4&5&6&7&8&0&0 \\ 0&0&1&2&3&4&5&6&0&0&0&0 \\ 0&0&1&2&3&4&5&6&0&0&0&0 \\ 0&0&1&2&3&4&0&0&0&0&0&0 \\ 0&0&1&2&3&4&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}.$$

Next by applying the back-propagation morphological erosion with the structuring element [x 0 -1] in the right-to-left scanning, we obtain

$$d = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&2&3&4&4&3&2&1&0&0 \\ 0&0&1&2&3&4&4&3&2&1&0&0 \\ 0&0&1&2&3&4&4&3&2&1&0&0 \\ 0&0&1&2&3&3&2&1&0&0&0&0 \\ 0&0&1&2&3&3&2&1&0&0&0&0 \\ 0&0&1&2&2&1&0&0&0&0&0&0 \\ 0&0&1&2&2&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}.$$

In the second step by squaring each object pixel, we obtain

$$d^2 = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&4&9&16&16&9&4&1&0&0 \\ 0&0&1&4&9&16&16&9&4&1&0&0 \\ 0&0&1&4&9&16&16&9&4&1&0&0 \\ 0&0&1&4&9&9&4&1&0&0&0&0 \\ 0&0&1&4&9&9&4&1&0&0&0&0 \\ 0&0&1&4&4&1&0&0&0&0&0&0 \\ 0&0&1&4&4&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}.$$

In the third step, we perform iterative erosions by 1D structuring components of size

3 until no further pixel value changes. After applying the forward morphological erosion by

$$\begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix},$$

we have

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 4 & 9 & 16 & 16 & 9 & 4 & 1 & 0 & 0 \\
0 & 0 & 1 & 4 & 9 & 10 & 5 & 2 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 4 & 9 & 9 & 4 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

After applying the erosion by

$$\begin{bmatrix} -3 \\ 0 \\ -3 \end{bmatrix},$$

we obtain

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & 1 & 0 & 0 \\
0 & 0 & 1 & 4 & 9 & 10 & 5 & 2 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 4 & 8 & 5 & 4 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

After applying the erosion by

$$\begin{bmatrix} -5 \\ 0 \\ -5 \end{bmatrix},$$

we obtain

$$e = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&1&1&1&1&0 \\ 0&0&1&4&4&4&4&4&4&1&0 \\ 0&0&1&4&9&9&5&2&1&1&0 \\ 0&0&1&4&8&5&4&1&0&0&0 \\ 0&0&1&4&5&2&1&1&0&0&0 \\ 0&0&1&4&4&1&0&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}. \tag{20}$$

After applying the erosion by

$$\begin{bmatrix} -7 \\ 0 \\ -7 \end{bmatrix},$$

the output remains unchanged, hence the iterations stop. At last, by taking a square-root for each object pixel, we accomplish the Euclidean distance transformation which is the same as the expected result in eq. (19).

2.3.2 Computational Complexity

The 1D decomposition algorithm for the Euclidean distance transformation is mainly composed of morphological erosions and the mathematical computation such as square and square-root. Morphological erosions involve subtraction and minimum selection. Therefore, the computational complexity depends on the number of the subtraction, minimum selection, square, and square-root operations. Assume P is the total number of pixels in the object, and C is the largest chessboard distance. The computational complexity is computed as:

$$\begin{aligned}\text{Complexity} &= P\ [\ 2\ (\ 2\ subtraction + 1\ minimum\ ) + 1\ square \\ &\quad + C\ (\ 3\ subtraction + 1\ minimum\ ) + 1\ square{-}root\ ] \\ &= (\ 4P + 3CP\ )\ subtraction + (\ 2P + CP\ )\ minimum + P\ square + P\ square{-}root \\ &= O\ (\ CP\ ),\end{aligned}$$

which is the order of CP.

2.4 2D DECOMPOSITION ALGORITHM USING ITERATIVE EROSIONS

Since the structuring element k^2 is convex and additively separable, it can be decomposed into dilations of small structuring components. Hence, the Euclidean distance transformation is the iterative erosions by a set of small structuring components and then a square-root operation.

2.4.1 2D Iterative Erosions Algorithm

(1) $d = f \ominus * \begin{bmatrix} -(4l-2) & -(2l-1) & -(4l-2) \\ -(2l-1) & 0 & -(2l-1) \\ -(4l-2) & -(2l-1) & -(4l-2) \end{bmatrix}, \quad l = 1, 2, \cdots, C$

(2) For $i = 1,\ length,\ 1$ do

For $j = 1,\ width,\ 1$ do

if ($d_{ij} > 1$) then $\qquad d_{ij} = (\ d_{ij}\ )^{½}$.

The final output d_{ij} is the result of the Euclidean distance transformation.

Example 3. Let the binary image f be the same as given in eq. (18). In the first step by applying the erosion by

$$\begin{bmatrix} -2 & -1 & -2 \\ -1 & 0 & -1 \\ -2 & -1 & -2 \end{bmatrix},$$

we have the following:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 255 & 255 & 255 & 255 & 255 & 255 & 1 & 0 & 0 \\ 0 & 0 & 1 & 255 & 255 & 255 & 255 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 255 & 255 & 255 & 255 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 255 & 255 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 255 & 255 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

After applying the erosion by

$$\begin{bmatrix} -6 & -3 & -6 \\ -3 & 0 & -3 \\ -6 & -3 & -6 \end{bmatrix},$$

we have

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 255 & 255 & 5 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 8 & 5 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

After applying the erosion by

$$\begin{bmatrix} -10 & -5 & -10 \\ -5 & 0 & -5 \\ -10 & -5 & -10 \end{bmatrix},$$

we obtain the result in eq. (20) in which each object pixel is the square of the Euclidean Distance.

2.4.2 Computational Complexity

$$\begin{aligned}\text{Complexity} &= P\ [\ C\ (\ 9\ subtraction + 1\ minimum\ ) + 1\ square-root]\\ &= 9CP\ subtraction + CP\ minimum + P\ square-root\\ &= O\ (\ CP\ )\end{aligned}$$

Note that the above two algorithms have the same computational complexity as a parallel algorithm by Yamada [19] and a linear algorithm by Vincent [17].

2.5 OPTIMAL DOUBLE TWO-SCAN ALGORITHM

The computational complexity of the algorithms presented so far is the order of CP. In other words, those algorithms consist of iterative operations, such that they depend on the number of iterations which is the object width. In this section, we present an optimal double two-scan algorithm which employs only four back-propagation morphological erosions to implement the Euclidean distance transformation in order to avoid the time-consuming iterations. In the algorithm, k_1 and $g_1(l)$ are the raster-scan structuring elements, and they are used in the image scanning left-to-right and top-to-bottom. The k_2 and $g_2(l)$ are the inversed raster-scan structuring elements, and they are used in the image scanning right-to-left and bottom-to-top.

The structuring elements used in the first two steps are to achieve the chessboard distance transformation. The k_1 and k_2 are given as follows:

$$k_1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 0 & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \end{bmatrix}, \quad k_2 = \begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & 0 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \tag{21}$$

The structuring elements $g_1(l)$ and $g_2(l)$ used in the third and fourth steps are functions of the parameter l. That is

$$g_1(l) = \begin{bmatrix} -(4l-2) & -(2l-1) & -(4l-2) \\ -(2l-1) & 0 & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} \end{bmatrix}, \tag{22}$$

$$g_2(l) = \begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & 0 & -(2l-1) \\ -(4l-2) & -(2l-1) & -(4l-2) \end{bmatrix}. \tag{23}$$

2.5.1 Double Two-scan Algorithm:

(1) $D = f \ominus k_1$.

(2) $E = f \ominus k_2$.

(3) For $i = 1$, *length*, 1 do

  For $j = 1$, *width*, 1 do

    $l = D_{ij.}$

    $R_{ij} = [\, f \ominus g_1(l)\,]_{i,j}$.

(4) For $i =$ *length* , 1 , -1 do

  For $j =$ *width* , 1 , -1 do

    $l = E_{ij}$

    $Q_{ij} = [\, R \ominus g_2(l)\,]_{i,j}$.

(5) For $i = 1$, *length*, 1 do

  For $j = 1$, *width*, 1 do

$$\text{if} (Q_{ij} > 1) \quad Q_{ij} = \sqrt{Q_{ij}}.$$

In the third step, the image D is used as the index value l to obtain $g_1(l)$ in eq. (22). In the fourth step, the image E is used as the index value l to obtain $g_2(l)$ in eq. (23). The flow chart of the double two-scan algorithm is shown in Fig. 1.

Example 4. Let the binary image f be the same as given in eq. (18). After the first step, we obtain

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&1&1&1&1&0&0\\
0&0&1&2&2&2&2&2&2&1&0&0\\
0&0&1&2&3&3&3&3&2&1&0&0\\
0&0&1&2&3&4&4&3&0&0&0&0\\
0&0&1&2&3&4&4&1&0&0&0&0\\
0&0&1&2&3&4&0&0&0&0&0&0\\
0&0&1&2&3&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}.$$

After the second step, the result is

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&2&3&3&3&3&2&1&0&0\\
0&0&1&2&3&3&2&2&2&1&0&0\\
0&0&1&2&3&3&2&1&1&1&0&0\\
0&0&1&2&2&2&2&1&0&0&0&0\\
0&0&1&2&2&1&1&1&0&0&0&0\\
0&0&1&2&2&1&0&0&0&0&0&0\\
0&0&1&1&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}.$$

After the third step, we obtain the result R

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&1&1&1&1&0&0 \\ 0&0&1&4&4&4&4&4&4&2&0&0 \\ 0&0&1&4&9&9&9&9&7&2&0&0 \\ 0&0&1&4&9&16&16&14&0&0&0&0 \\ 0&0&1&4&9&16&23&2&0&0&0&0 \\ 0&0&1&4&9&16&0&0&0&0&0&0 \\ 0&0&1&4&9&2&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} .$$

After the fourth step, the result is

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&1&1&1&1&0 \\ 0&0&1&4&4&4&4&4&4&1&0 \\ 0&0&1&4&9&9&5&2&1&1&0 \\ 0&0&1&4&8&5&4&1&0&0&0 \\ 0&0&1&4&5&2&1&1&0&0&0 \\ 0&0&1&4&4&1&0&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} .$$

At last, by taking the square-root for all object pixels, the result is the Euclidean distance transformation.

2.5.2 Computational Complexity

The parametric structuring elements $g_1(l)$ and $g_2(l)$ can be obtained easily using the table look-up. The computational complexity of this algorithm is computed as follows:

$$\begin{aligned} \text{Complexity} &= P\ [\ 4\ (\ 5\ \textit{subtraction} + 1\ \textit{minimum}\ ) \\ &\quad + 2\ \textit{look-up} + 1\ \textit{square-root}\ ] \\ &= 20P\ \textit{subtraction} + 2P\ \textit{look-up} + 4P\ \textit{minimum} \end{aligned}$$

$$+ P \ square - root$$

$$= O(P).$$

Hence this algorithm can be implemented very fast by using the systolic array two-scan architecture [15], especially when the object size grows or the largest chessboard distance is large.

2.6 CONCLUSIONS

Three new techniques for the Euclidean distance transformation are proposed. They are based on the morphological erosion. The forward morphology and back-propagation morphology techniques have been used in these algorithms. Using these algorithms, we can obtain the exact result of the Euclidean distance measure instead of the approximated result proposed by others previously. The computational complexity of three algorithms has been analyzed in this chapter. The double two-scan algorithm has the optimal computational complexity which is independent of the object size. Using the parallel processing, we can further speed up the Euclidean distance transformation when the double two-scan algorithm is employed.

CHAPTER 3

Shape Features Extraction Using Mathematical Morphology And Its Application to Character Recognition

3.1 INTRODUCTION

Character Recognition has been a widly researched problem. There are some solutions to this problem which have been tested and used In particular, mathematical morphology can be a good approach Mathematical morphology can be used very efficiently in extracting the object's features by designing a set of structuring elements corresponding to the primitive shapes of the object. The structuring elements design gives the randomness and flexiblity to feature represention People use intuitive shape instead of mathematical denotion

The designing of a set of suitable structuring elements is a key issue and a long-term research problem The set of primitive shapes in an object have different degrees of importance to represent the object and to distinguish from the other objects. The pixels in a primitive shape have also the different degrees of importance. Traditionally in template matching, a designed mask corresponding to all pixels in the primitive shape is coorelated with the image to extract the locations of the matching pattern. In this chapter, the structuring element we use is composed of only a few important pixels instead of all the pixels in the primitive shape. Hence, the structuring element is simpler and the morphological operation is faster. Besides, the set of structuring elements are independent to each other and can be operated in parallel. The result can be combined by a weighted sum of intermediate outputs that can be implemented using the neural networks approach. Thus, the application to character recognition can be performed fast

The morphological operations used are erosion and hit-and-miss [25]. If the matching of the foreground and background pixels is intended, then the hit-and-miss operation is used. If the matching of only foreground pixels is intended, then the erosion is adopted. After either operation, if an object contains the primitive shape embeded in a structuring element, then some points will be left in the object.

Since the domain of objects dealt with in optical character recognition is restricted, all the structuring elements can be priorly designed and stored in a database with their indexes. Each character is linked to several indexes of structuring elements. Each index is accompanied with the information such as the choice of the erosion or hit-and-miss operation and the structuring element's weight along with a flag which is ''1'' or ''-1'' indicateing this templete should exist in object or not.

This chapter describes a technique of feature extraction and recognition by mathematical morphology. Such a technique is quick and simple. Ths second part introduces recofnition procedure, third part presents the recognition criterion.

3.2 Morphological Features Selection

There are two types of features: global and local features. The global feature is to extract the shape information in the whole character image. The local feature is to extract the shape information limited in a portion of the character image, including geometric and topological properties, e.g. stroke direction, length, and position. Based on the intra-character and inter-character relationship, the importance (or unimportance) can be put on different weights summarized as excitatory (positive) or inhibitory (negative) weights.

The advantages of our technique are that no need of thinning and distortion allowance.

3.2.1 Intersection points

Characters consist of strokes or curves which may intersect at certain locations.

Considering a noisy character image, the occurrence of the center of such intersection points has the largest probability to exist. That is the use of those intersecting centers is reliable for the pattern matching purpose. If the binary character image is performed by a distance transformation, the center of intersection points of lines is equivalent to the point with the distance values which is the local maxima. Thus these centers can be used to design templetes.

3.2.2 Shape information

Shape imformation play an important role in character recognition. There are many different shapes, like straight line, curve, hole, point, etc. For curve, which is most complicated to represent, has many kinds of location, radius, and orientation. Designing morphological structuring elements to represent these, however, is not very complicated. This is one of the advantage of morphology.

The number of primitive shapes is limited in a particular domain. Say, we want to recognize the capital and non-capital characters, we can use only about 20 primitive shapes to detect features. Most of them are strokes, arcs and combination of strokes or arcs or both of them. Strokes include long vertical line, short vertical line, horizontal line, slanted line, and combination of straights, such as ''|-'' ,etc. Arcs are designed as half circle, (left or right half circle), shape to detect the upper curve, shape to detect lower part of characters and also some special shapes. Structuring elements are designed in binary templetes, for each above primitive shape, we use suitable pixels in templete to represent it.

After we have choose the simple shapes, problems left is how to represent it in structuring elements. Since we use normalized characters as objects, the window size we use here is 16*16. The criterion here is these pixels for one feature in the structuring element along with the operation can not match any other feature. That means, the pixels can and only can represent its own feature. In the meanwhile, the pixels should be minimal in order to reduce the computational complexity. For example, we have two kinds of vertical lines: long vertical line and short vertical line. For long vertical line, we use five pixels distributed equally in the vertical direction in the templete to match it, the operation we use is morphological erosion. For short vertical line, we use three pixels distributed equally in half of the window length, the operation we use is morphological hit-and-miss.

Thus, structuring elements are designed to represent corresponding simple shapes in the objects domain. Complex shapes are formed by these simple shapes. Feature recognition is to map the fixed list of simple shapes to the unknown object. Then, using the previous formular to combine the results and get the result.

3.2.3 Distinguishable points

Some characters can be matched as the same for all other structuring elements except their exclusive curves or exclusive points. For examples, in non-capital characters, ''i'' and ''l'' , the points in the upper part

of the center column are distingusable points. In capital characters, like ''I' and ''J'', ''P'' and ''R'', etc. In these cases , these distinguishable points are one of the most important recognition condotions for them. To recognize these characters, we design the structuring element containing the distinguishable parts represented by several pixels. For instance, to seperate ''P'' from ''R'', we disign a templete having feature of the lower left part of ''R'', which is composed by an arc and a slanted line. When we do matching with ''P'', the corresponding operation is erosion, with the flag ''-1''; while matching with ''R'' we perform erosion and the flag is ''1''.

3.3 Recognition Procedure

Each character can be recognized by designing a corresponding set of structuring elements. Different characters should have different sets of structuring elements. Some structuring elements may be applied to multiple characters. The designing criteria will be presented in Section ?. According to the degree of importance of the primitive shape, the structuring element is assigned a predefined weight, say p_i. The p_i are determined by statistics and experiments which can be positive within a given range.

Suppose there are totally n structuring elements K_i $(i = 1, \cdots, n)$ used in a character. For each structuring element, the input image is performed by a morphological erosion or hit-and-miss operation. If there are some pixels left in the output image, the value of flag is given; otherwise "0" is given. Let the value be denoted by v_i for the i-th structuring element. Thus, the matching result of the character is computed by

$$\sum_{i=1}^{n} v_i p_i$$

Perform morphological erosion to unknown object using all sets of structuring elements in parallel processing, and selects the biggest probability value and its object, we can quickly do the recognization.

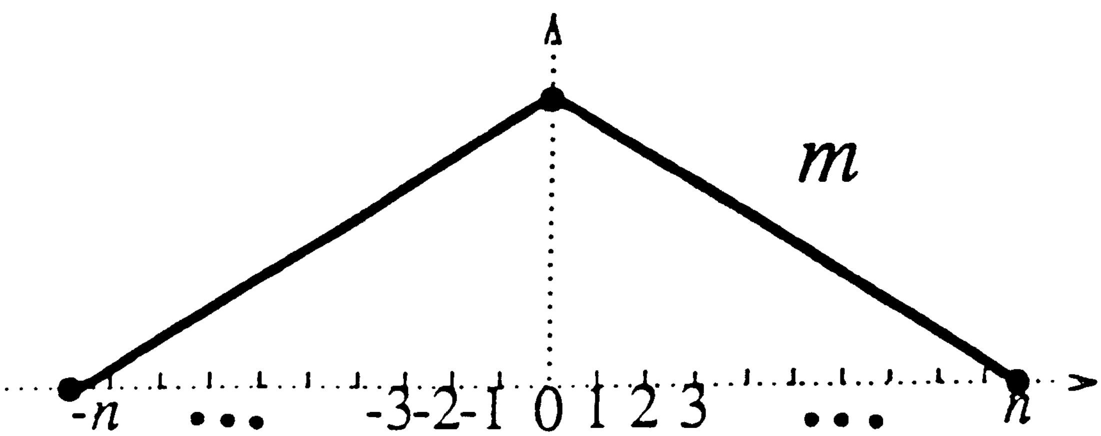


Fig. 1. A linearly-sloped structuring element.

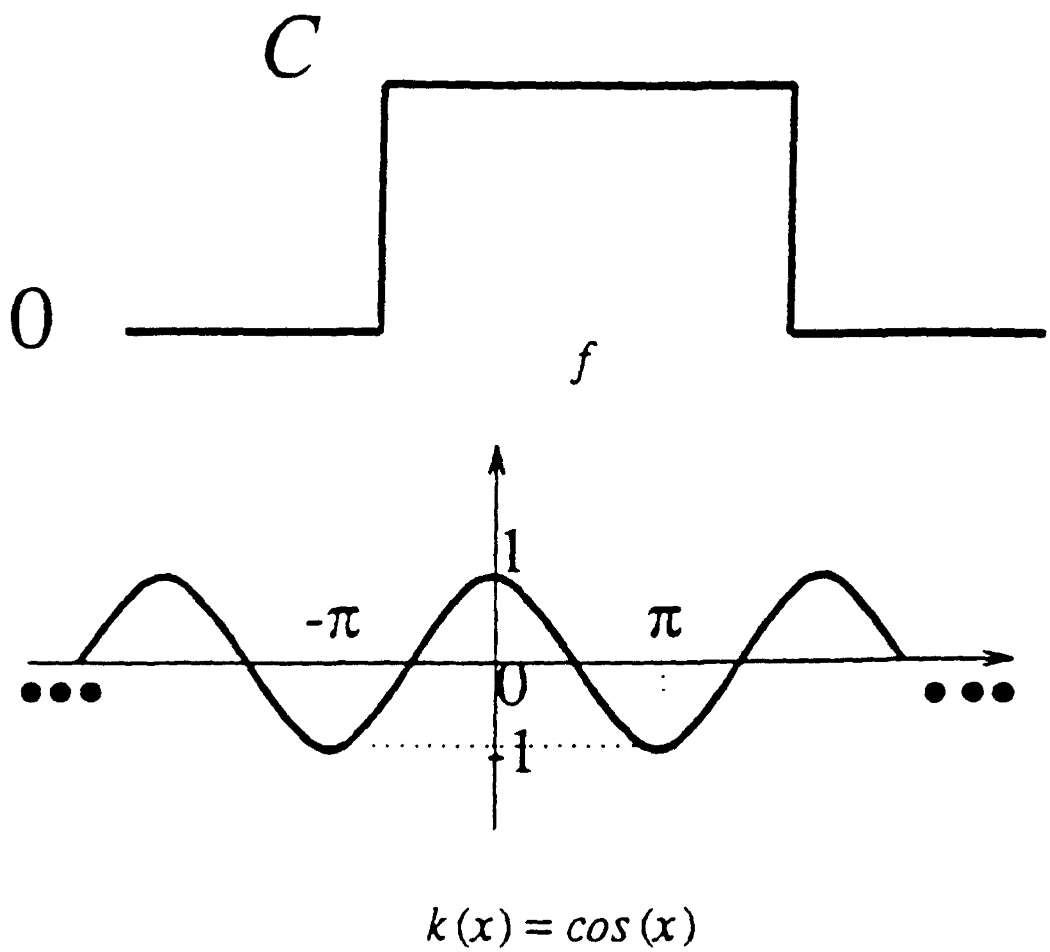


Fig. 2. A binary image f with two gray levels, 0 indicating background and C indicating foreground, and a large cyclic cosine structuring element k.

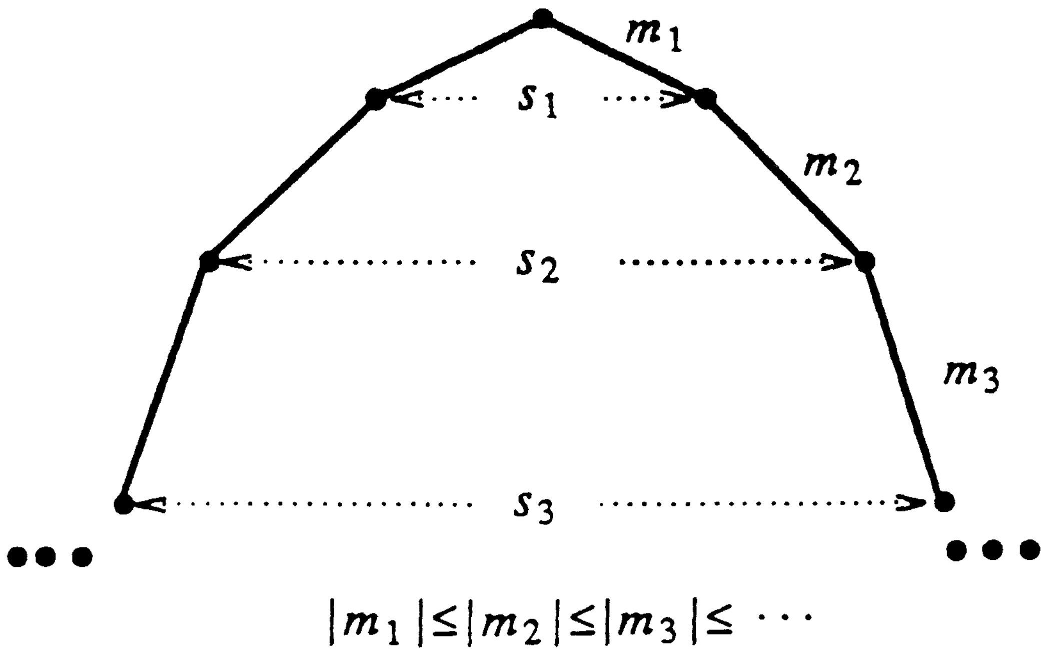


Fig. 3. A convex structuring element.

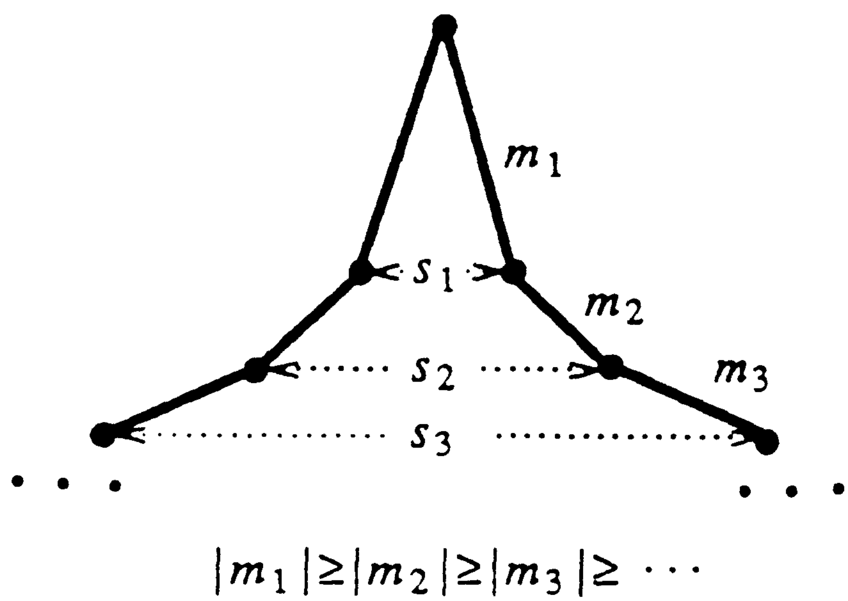


Fig. 4. A concave structuring element.

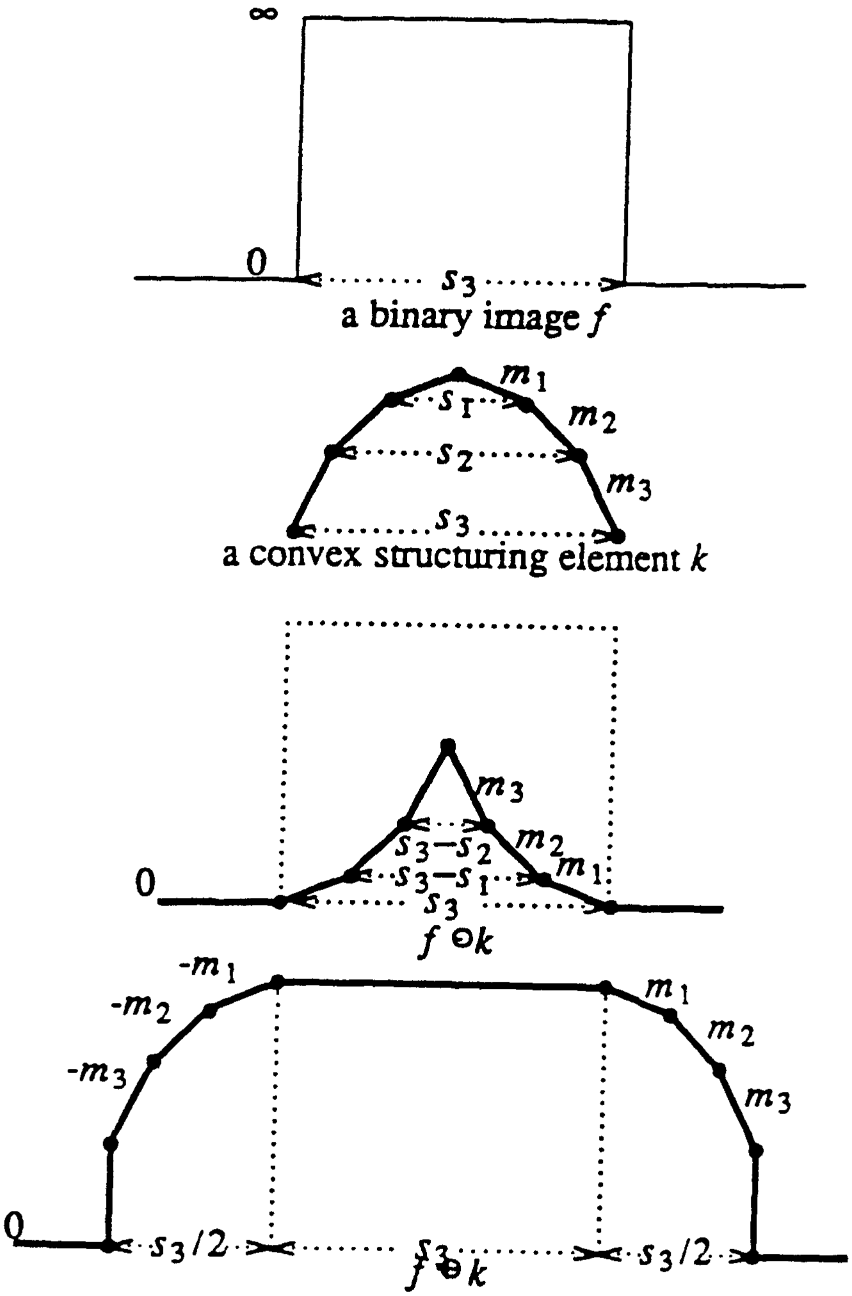
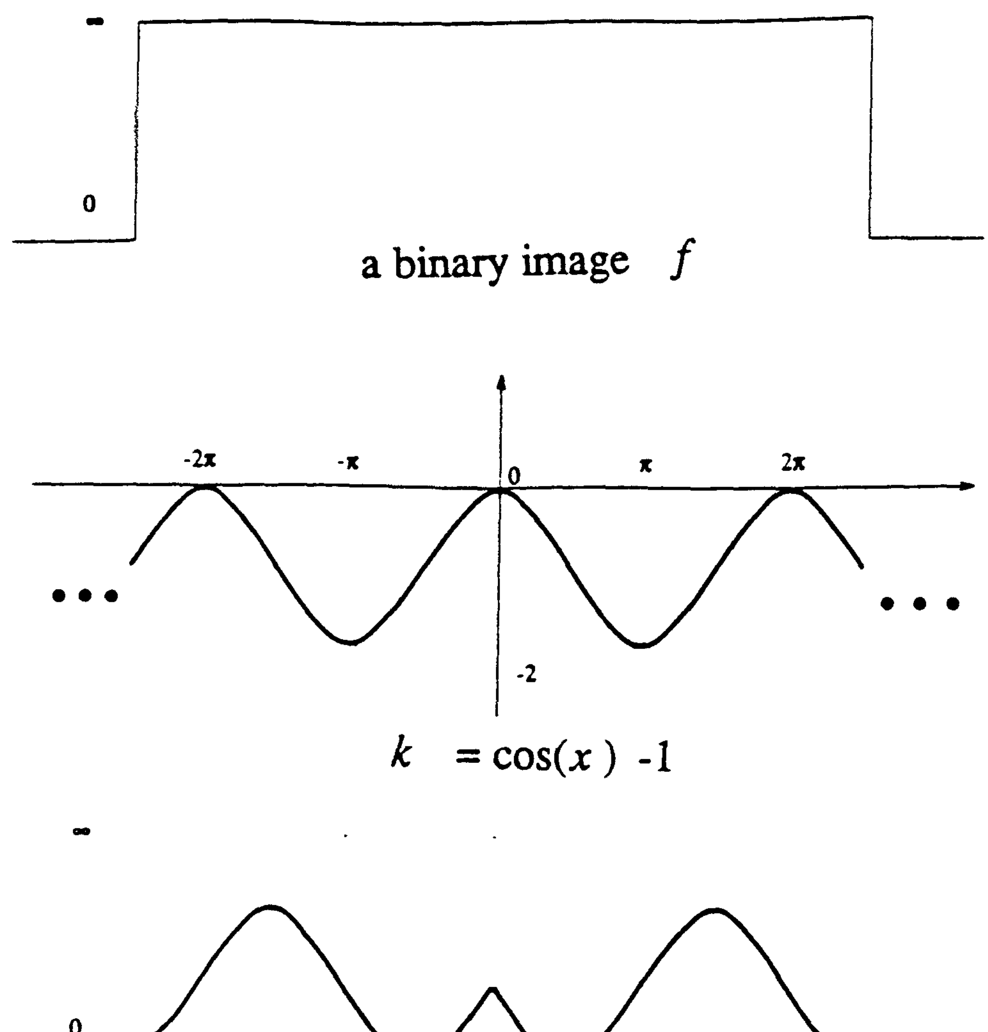


Fig. 5. An example of a binary image f eroded and dilated by a convex structuring element k



$\Psi^{-1}[f \ominus k^*]$

Fig. 6. The inverse transformation of the eroded result of f by k^*, such that k^* is the new transformed cyclic structuring element of k.

REFERENCES

1. Nawrath, R. and Serra, J.. "Quantitative Image Analysis: Theory and Instrumentation." *Microsc. Acta* **82**, (1979): 101-111.

2. Serr, a J., *Image Analysis and Mathematical Morphology.* Acad. Press, New York, 1982.

3. Serra, J., *Image Analysis and Mathematical Morphology. Vol. 2, Theoretical Advances,* Acad. Press, New York, 1988.

4. Shih, F. Y. and Mitchell, O. R., "Threshold Decomposition of Grayscale Morphology Into Binary Morphology." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, Jan. (1989):**1**,31-42.

5. Shih, F. Y. and Mitchell, O. R., "Decomposition of Gray Scale Morphological Struc- turing Elements." *Pattern Recognition,* vol. 24, March (1991):**3**, 195-203.

6. Sternberg, S. R., "Biomedical Image Processing." *IEEE Computer*, vol. 16, Jan. (1983) :22-34.

7. Sternberg, S. R., "Gray-Scale Morphology." *Computer Vision, Graphics, and Image Processing,* Vol. 35, Sep. (1986):333-355.

8. Zhuang, X. and Haralick, R. M., "Morphological Structuring Element Decomposition." *Computer Vision, Graphics, and Image Processing,* Vol. 35, (1986):370 -382.

9. Borgefors, G., "Distance Transformations In Arbitrary Dimensions." *Computer Vision, Graphics, and Image Processing,* vol. 27, (1984):321-345.

10. Borgefors, G., "Distance Transformations in Digital Images." *Computer Vision, Graphics, and Image Processing,* vol. 34, (1986):344-371.

11. Danielsson, P. E., "Euclidean distance mapping." *Computer Graphics Image Processing,* vol. 14, (1980): 227-248.

12. Giardina, C. R. and Dougherty, E. R., *Morphological Methods in Image and Signal Processing*, Prentice Hall, 1988.

13. Haralick, R.M., Sternberg, S. R., and Zhuang, X., "Image Analysis Using Mathematical Morphology." *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 9, no. 4, July (1987): 532-550.

14. Huang, C. and Mitchell, O. R., "Shape Analysis Using Morphology." *Proc. Mid-*

con, (1990): 436-440.

15. Montanari, U., ''A Method For Obtaining Skeletons Using a Quasi-Euclidean Distance In N Dimensions.'' *J. Assoc. Comput.*, vol. 15, (1968): 600-624.

16. Rosenfeld, A. and Kak, A. C., *Digital Picture Processing*. Academic Press, 1982.

17. Rosenfeld, A. and Pfaltz, J. L., ''Sequential Operations In Digital Picture Processing. '' *J. Assoc. Comp. Mach.*, vol. 13, no. 4, (1966): 471-494.

18. Rosenfeld, A. and Pfaltz, J. L., ''Distance Functions On Digital Pictures.'' *Pattern Recognition*, vol. 1, (1968): 33-61.

19. Serra, J., *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

20. Shih, F. Y. and Mitchell, O. R., ''A Mathematical Morphology Approach To Euclidean Distance Transformation.'' *IEEE Trans. Image Processing*, vol. 1, no. 2, April 1992.

21. Shih, F. Y., King, C. T., and Pu, C. C., ''A Two-Scan Algorithm And Architecture To A Root For Morphological Filters.'' *Proc. IEEE Inter. Phoenix Conf. Computers and Communications*, Arizona, March 1990.

22. Vincent, L., ''Exact Euclidean Distance Function by Chain Propagations.'' *Proc. IEEE Computer Vision and Pattern Recognition*, Hawaii, (1991): 520-525.

23. Vossepoel, A. M., ''A Note on Distance Transformations in Digital Images.''*Computer Vision, Graphics, and Image Processing*, vol. 43, (1988): 88-97.

24. Yamada, H., ''Complete Euclidean Distance Transformation by Parallel Operations.'' *Proc. 7th Inter. Conf. Pattern Recognition*, Montreal, (1984):69-71.

25. Baird, H. S., ''Feature Identification for Hybrid Structural/Statistical Pattern Classification.'' *Proc. IEEE Conf. Computer Vision Pattern Recognition*, (1986): 150-155.