Journal of Mathematical Imaging and Vision, 4, 5–18 (1994).

Decomposition of Morphological Structuring Elements

XINHUA ZHUANG
Department of Electrical and Computer Engineering, University of Missouri-Columbia, Columbia, MO 65211

Abstract. Mathematical morphology has become an important tool for machine vision since the influential work by Serra (1982). It is a branch of image analysis based on set-theoretic descriptions of images and their transformations. As is well known, the chain rule for basic morphological operations, i.e., dilation and erosion, lends itself well to pipelining. Specialized pipeline architecture hardware built in the past decade is capable of efficiently performing morphological operations. The nature of specialized hardware depends on the strategy for morphologically decomposing a structuring element. The two-pixel decomposition technique and the cellular decomposition technique are the two main techniques for morphological structuring-element decomposition. The former was used by the image flow computer and the latter by the cytocomputer.

This paper represents a continuation of the work reported by Zhuang and Haralick (1986). We first give the optimal two-pixel decomposition for the binary structuring element and subsequently attempt to solve the grayscale-structuring-element two-pixel decomposition problem. To our knowledge, no efficient algorithm for this problem has been found to date. The difficulty can be overcome by using an adequate representation for the grayscale structuring element. Representing a grayscale image as a specific 3D set, i.e., an umbra (Sternberg, 1986), makes it easier to shift all basic morphological theorems from the binary domain to the grayscale domain; however, a direct umbra representation is not appropriate for the grayscale-structuring-element decomposition. In this paper a morphologically realizable representation and a two-pixel decomposition for the grayscale structuring element are presented; moreover, the recursive algorithms, which are pipelineable for efficiently performing grayscale morphological operations, are developed on the basis of the proposed representation and decomposition. This research will certainly be beneficial to real-time image analysis in terms of computer architecture and software development.

Key words. structuring element, closed-form decomposition, recursive decomposition

1 Introduction

As is well known, mathematical morphology provides an important tool for machine vision [1]–[5]. Specialized hardware, such as the cytocomputer and the image flow computer [6]–[10], is capable of efficiently performing morphological operations. The nature of the specilized hardware depends on the strategy for the morphological decomposition of a structuring element [11]–[21]. The cytocomputer is a pipelineable machine that is suited to the 3×3 neighborhood morphological structuring-element decomposition. The image flow computer is a pipelineable machine that can process a morphological operation on any pair of pixels in each stage and thus is suited to the two-pixel morphological structuring-element decomposition. When a structuring element used in the morphological operations has a domain larger than the hardware can handle in one stage, the structuring element must be morphologically decomposed into smaller substructuring elements, which each can be handled by one stage in the pipeline and whose morphological composition gives the structuring element. The morphological-structuring-element decomposition problem is to determine a smallest sequence of those substructuring elements; this sequence is called the optimal decomposition.

This paper represents a continuation of the work reported in [13], in which the optimal two-pixel decomposition for a binary or grayscale

structuring element was reached by performing a tree search. The search, however, proved inefficient for a binary structuring element of large size or a grayscale structuring element in general. In this paper we first derive the optimal decomposition for the binary structuring element, which involves only three straightforward steps, and we subsequently attempt to solve the grayscale-structuring-element two-pixel decomposition problem. To our knowledge, no efficient algorithm for this problem has been found to date. The difficulty, from the author's point of view, is actually caused by using the umbra representation for the grayscale structuring element.

In the literature of mathematical morphology, a grayscale image is normally represented as a specific 3D set, i.e., an umbra. In terms of umbra homomorphism theorems [2], all basic morphological theorems can be easily shifted from the binary domain to the grayscale domain. However, it is much more difficult to directly decompose a 3D set than to directly decompose a 2D set, even if the 3D set is an umbra, by using either two-pixel decomposition or cellular decomposition. In this paper we present a morphologically realizable representation and a two-pixel decomposition for the grayscale structuring element and develop the recursive algorithms, which are pipelineable for efficiently performing grayscale morphological operations, on the basis of the proposed representation and decomposition.

The paper is organized as follows. In section 2 we explain the binary-structuring-element two-pixel decomposition technique. In section 3 we present the grayscale-structuring-element representation and two-pixel decomposition technique and develop the pipelineable recursive algorithms for grayscale morphological operations. Section 4 is a conclusion.

2 Binary-Structuring-Element Two-Pixel Decomposition

2.1 Basic Definitions

Morphology is the study of shape or form. Mathematical morphology, or image algebra, is the study of shape by using the tools of set theory. It is a branch of image analysis that uses set-theoretic descriptions of images and their transformations.

Dilation and erosion are the primary operations of mathematical morphology. In loose terms, these operations cause the swelling or shrinking of areas when the structuring element has a disklike shape. In the context of image analysis, the areas are usually portions of 2D images. Mathematical morphological operations apply to sets of any dimensions, such as Euclidean N-space R^N or its discrete or digitized equivalent, the set of N-tuples of integers Z^N. For the sake of simplicity we refer to either of these sets as E^N. We now present some definitions based on an image X and a structuring element B, where each is included in E^2. Dilation is defined as

$$X \oplus B = \{y \mid y = x + b, x \in X, b \in B\}.$$

Erosion is defined as

$$X \ominus B = \{y \mid \forall b \in B, y + b \in X\}.$$

If X_b denotes the translation of X by the pixel b, then it can be shown that

$$X \oplus B = \bigcup_{b \in B} X_b,$$
$$X \ominus B = \bigcap_{b \in B} X_{-b}.$$

This means that dilation can be accomplished by taking the union of all the translates of X, where the shifts in the translates come from B, and that erosion can be accomplished by taking the intersection of all the translates of X, where the shifts in the translates are the negated members of B. An especially interesting case for B is that in which B consists of two pixels, where one is the origin, namely,

$$B = \{0, b\}.$$

Then the dilation and erosion are computed simply by

$$X \oplus B = X \cup X_b,$$
$$X \ominus B = X \cap X_{-b}.$$

Dilation and erosion both exhibit the property of translation invariance:

$$X_b \oplus B = (X \oplus B)_b,$$
$$X \oplus B_b = (X \oplus B)_b,$$
$$X_b \ominus B = (X \ominus B)_b,$$
$$X \ominus B_b = (X \ominus B)_{-b}.$$

Dilation is commutative:

$$X \oplus B = B \oplus X.$$

There exist chain rule relations, or associative properties, for dilation and erosion:

$$(B_1 \oplus B_2) \oplus B_3 = B_1 \oplus (B_2 \oplus B_3),$$
$$(B_1 \ominus B_2) \ominus B_3 = B_1 \ominus (B_2 \oplus B_3).$$

This means that if

$$B = B_1 \oplus B_2 \oplus \cdots \oplus B_k,$$

then

$$X \oplus B = (\cdots[(X \oplus B_1) \oplus B_2] \oplus \cdots \oplus B_k),$$
$$X \ominus B = (\cdots[(X \ominus B_1) \ominus B_2] \ominus \cdots \ominus B_k).$$

Thus if a structuring element can be broken down to a chain of dilations of smaller substructuring elements, the desired operation may be performed as a string of suboperations. As stated before, this property lends itself well to pipelining.

If the lowercase letter f represents the grayscale image and the lowercase letter k represents the grayscale structuring element, the grayscale dilation and erosion can be computed by

$\forall x \in F \oplus K$,

$$(f \oplus k)(x) = \max_{u \in K}\{f(x-u) + k(u)\},$$

$\forall x \in F \ominus K$,

$$(f \ominus k)(x) = \min_{u \in K}\{f(x+u) - k(u)\},$$

where F, K, $F \oplus K$, and $F \ominus K$ denote the defining domain for f, k, $f \oplus k$, and $f \ominus k$, respectively. An especially interesting case for k is that in which k's domain K consists of two pixels, one of which is the origin, and in which k's values are zero, namely,

$$K = \{0, b\},$$
$$k(0) = k(b) = 0.$$

Then the grayscale dilation and erosion are simply computed by

$$(f \oplus k)(x) = \max\{f(x),\ f(x-b)\},$$
$$(f \ominus k)(x) = \min\{f(x),\ f(x+b)\}.$$

All of the relationships that hold for the binary case are preserved here for the grayscale case in a form in which intersection is replaced by min and union is replaced by max.

2.2 Digital-Line-Segment Two-Pixel Decomposition

Let L_{2^k} denote a special digital line segment that consists of 2^k pixels a_i, $i = 0, 1, \ldots, 2^k-1$, where a_0 is the origin and $a_i = ia_1$, $i = 0, 1, \ldots, 2^k-1$ (see figure 1). It is easy to verify that the optimal two-pixel decomposition of L_{2^k} is given by

$$L_{2^k} = \{0, a_1\} \oplus \{0, a_2\} \oplus \{0, a_4\} \oplus \cdots \oplus \{0, a_{2^{k-1}}\}.$$

Thus for the optimal two-pixel decomposition of L_{2^k}, which consists of 2^k pixels, we need only k two-pixel substructuring elements. In other words, the number of two-pixels in the optimal decomposition is equal to the base-two logarithm of the number of pixels in L_{2^k}.

Let L_m be a more general digital line segment consisting of m points a_i, $i = 0, \ldots, m-1$, where a_0 is the origin and $a_i = ia_1$, $i = 0, \ldots, m-1$. The optimal two-pixel decomposition for L_{2^k} for $m = 2^k$ has already been derived. If $2^k < m < 2^{k+1}$, it is easy to verify that the optimal two-pixel decomposition for L_m is given by

$$\begin{aligned} L_m &= L_{2^k} \oplus \{0, a_{m-2^k}\} \\ &= \{0, a_1\} \oplus \{0, a_2\} \oplus \{0, a_4\} \oplus \cdots \\ &\quad \oplus \{0, a_{2^{k-1}}\} \oplus \{0, a_{m-2^k}\}. \end{aligned} \tag{1}$$

As a result, the optimal two-pixel decomposition for a more general digital line segment L_m uses only k two-pixels if $m = 2^k$ and uses only $k+1$ two-pixels if $2^k < m < 2^{k+1}$. It is apparent that a general digital line segment L can always be expressed as a translated L_m; that is, there exists a pixel t such that L is equal to $(L_m)_t$. The optimal two-pixel decomposition for L is given by substituting $\{t, a_1 + t\}$ for the first term of the dilation in (1).

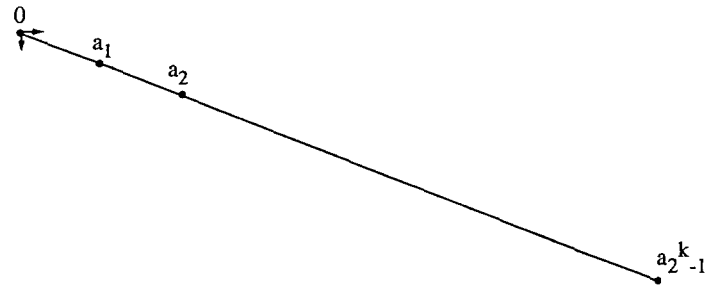

Fig. 1. Digital line segment L_{2^k}.

There is a close connection between two-pixel-decomposable and digital-line-segment-decomposable shapes. A two-pixel-decomposable shape is the shape that can be decomposed into a dilation of two-pixels, and a digital-line-segment-decomposable shape is the shape that can be decomposed into a dilation of digital line segments. Since it has already been proved that any digital line segment is two-pixel decomposable, a digital-line-segment-decomposable shape must be two-pixel decomposable. Conversely, a two-pixel-decomposable shape, if it is exteriorly digital convex (defined in subsection 2.3) must be digital-line-segment decomposable (proved in subsection 2.3).

2.3 *Two-Pixel Decomposition Procedure*

Now come the questions. What kinds of shapes can be decomposed into two-pixel or digital-line-segment substructuring elements? and How can they be decomposed?

DEFINITION 1. A shape B in E^2 is center symmetric if there exists a center point c in R^2 such that $x \in B$ implies $2c - x \in B$ for all $x \in B$ (see figure 2).

PROPOSITION 1. Assume that $B = B_1 \oplus B_2 \oplus \cdots \oplus B_k$ and that each B_i is center symmetric. Then B is center symmetric.

Proof. We need only to find the center point c for shape B. Let c_i denote the center point for B_i, $i = 1, \ldots, k$. Then $c = c_1 + \cdots + c_k$ is the center point of B. To prove this we notice

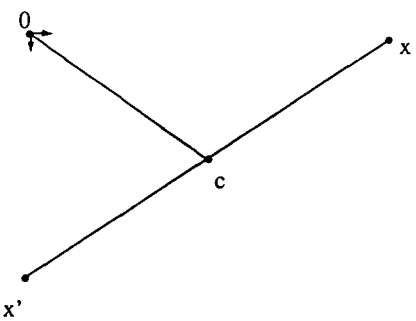

Fig. 2. Center point c; $x' = 2c - x$.

that any pixel $x \in B$ admits a representation as follows:

$$x = x_1 + \cdots + x_k,$$

with each $x_i \in B_i$, $i = 1, \ldots, k$, which follows from the assumption that $B = B_1 \oplus B_2 \oplus \cdots \oplus B_k$. The symmetric pixel x' of x with respect to c can be shown as follows:

$$\begin{aligned} x' &= 2c - x \\ &= (2c_1 - x_1) + \cdots + (2c_k - x_k) \\ &= x_1' + \cdots + x_k', \end{aligned}$$

where for each i, x_i' denotes the symmetric pixel of x_i with respect to c_i and thus belongs to B_i, which follows from the assumption that each B_i is center symmetric. Finally, that $x_i' \in B_i$ for each i implies that their sum x' belongs to B from the assumption that $B = B_1 \oplus B_2 \oplus \cdots \oplus B_k$, and this proves that c is the center point of B.

COROLLARY 1. A two-pixel-decomposable shape must be center symmetric.

Fig. 3. The left center-symmetric shape is not two-pixel decomposable, and the right one is.

Proof. Since any two-pixel substructuring element is center symmetric, the conclusion follows from Proposition 1.

Remark. The inverse of Corollary 1, however, is not correct in general; see, for instance, figure 3.

PROPOSITION 2. Assume that B_1 and B_2 both are center symmetric and share the same center point c. Then the union $B_1 \cup B_2$, the intersection $B_1 \cap B_2$, and the difference $B_1 - B_2$ all are center symmetric.

Proof. We verify the last assertion. Assume that $x \in B_1 - B_2$; that is, that $x \in B_1$ and $x \notin B_2$. The former implies $2c - x \in B_1$ since B_1 is center symmetric with respect to c. The latter implies that $2c - x \notin B_2$. Otherwise, x would belong to B_2 too, since B_2 is also assumed center symmetric with respect to c. Thus $2c - x \in B_1 - B_2$, and this assures that $B_1 - B_2$ is center symmetric with respect to c.

DEFINITION 2. A digital shape $B \subset Z^2$ is called exteriorly digitally convex if the sampled boundary of its convex-hull extension $h(B)$ in R^2 (for a definition of the convex-hull extension see [1]) is included in B. That is, if $\partial h(B)$ denotes the boundary of $h(B)$ in R^2, then $\partial h(B) \cap Z^2$ is included in B; see figure 4.

DEFINITION 3. A digital shape $B \subset Z^2$ is called digitally convex if it is equal to the sampling of its convex-hull extension $h(B)$ in R^2; that is $B = h(B) \cap Z^2$.

Remark. A digitally convex shape is exteriorly digitally convex, but the converse is not true in general. The shape B in figure 4 is exteriorly digitally convex but is not digitally convex.

PROPOSITION 3. Assume that both A_1 and A_2 are an analog convex set in R^2. Then the dilation $A_1 \oplus A_2$ is also an analog convex set.

Proof. See reference [1].

Remark. For any two digitally convex sets B_1 and B_2, their dilation is not necessarily digitally convex; see figure 5.

PROPOSITION 4. An analog shape A in R^2 can be morphologically decomposed into a dilation of analog line segments $l_1, \ldots, l_k$, namely,

$$A = l_1 \oplus \cdots \oplus l_k, \tag{2}$$

if and only if A is a center-symmetric convex polygon.

Proof. That the analog line segments are center symmetric and convex and that their dilation is also follows from Propositions 1 and 3. It is also quite obvious that the dilation of analog line segments forms a polygon. Thus the "only if" part is proved. To prove that a center-symmetric convex polygon can be decomposed into a dilation of analog line segments, we need only to point out that, recursively, a center-symmetric convex polygon of $2n$ sides can be decomposed into a dilation of a center-symmetric convex polygon of $2(n-1)$ sides by one of its $2n$ sides, that the resulting center-symmetric convex polygon of $2(n-1)$ sides can be further decomposed into a dilation of a center-symmetric convex polygon of $2(n-2)$ sides by one of its remaining $2(n-1)$ sides, and so on. In this way the center-symmetric convex polygon of $2n$ sides is decomposed into a dilation of n analog line segments; if the order and translation are neglected, these n analog line segments are unique and actually compose half of the analog line segments on the boundary ∂A (see figure 6).

Remark. From Proposition 4 it is clear that an analog shape in R^2 is analog-line-segment decomposable if and only if it is a center-symmetric convex polygon. A similar conclusion does not hold for the digital case in general. This may be explained as follows. Suppose we are given a digital shape B that is finite, center symmetric, and digitally convex. Let A denote its

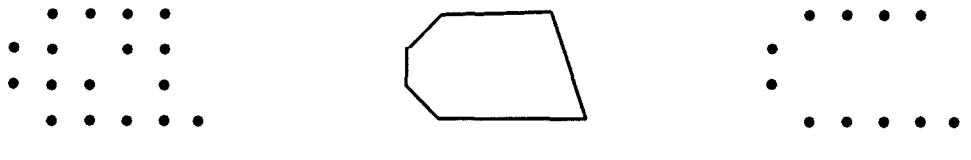

Fig. 4. $B \subset Z^2$ is exteriorly digitally convex but is not digitally convex; $h(B)$ denotes the convex-hull extension of B in R^2, and the sampled boundary of $h(B)$ is included in B. From the left to right are shown three sets, B, $h(B)$, and the sampled boundary of $h(B)$, i.e., $\partial h(B) \cap Z^2$.

Fig. 5. Illustration of how two digitally convex sets generate their dilation, which is not digitally convex.

convex-hull extension in R^2. Then A is an analog center-symmetric convex polygon and B is its sampling; i.e., $B = A \cap Z^2$. Let A be decomposed by half of its analog boundary line segments as in (2). Denote the sampling of l_i by L_i; i.e., $L_i = l_i \cap Z^2$, $i = 1, \ldots, k$. Then the analog-line-segment decomposition (2) of A implies only the following inclusion relationship in general:

$$L_1 \oplus \cdots \oplus L_k \subset B. \tag{3}$$

In fact, the dilation of digitally convex sets including digital line segments is always exteriorly digitally convex, but it may not be digitally convex. The holing phenomenon shown in figure 7 is actually caused by the digital sampling. Fortunately, almost all binary structuring elements built into specialized hardware to date not only are center symmetric and digitally convex but also admit a digital-line-segment decomposition.

PROPOSITION 5. A digital shape B is digital-line-segment-decomposable if and only if it is two-pixel decomposable and exteriorly digitally convex; moreover, those participating digital line segments compose half of its exterior digital boundary line segments.

Proof

(only if part) A digital-line-segment-decomposable shape must be two-pixel decomposable since each digital line segment is two-pixel decomposable, as was stated previously. Denote those participating digital line segments by L_i, $i = 1, \ldots, k$, and their respective convex-hull extensions in R^2 by l_i, $i = 1, \ldots, k$. Then it is easy to see that these l_i's are analog line segments and that their dilation denoted by A composes the convex-hull extension of B in R^2; i.e., $A = h(B)$. Since the l_i's compose half of the analog boundary line segments on ∂A, their samplings, the L_i's, compose half of the digital boundary line segments on $\partial A \cap Z^2$. The latter is the exterior digital boundary of B. In this argument we do not consider the order and translation of these digital line segments.

(if part) To see this we first divide those participating two-pixels into groups. Each two-pixel determines a unique digital line direction. We put two-pixels in the same group when they share the same direction. We then dilate two-pixels in each group together to form a subshape. As can be easily seen, each subshape lies on a digital line. It actually is a single digital line segment without a hole or holes; otherwise, dilating those holed digital line segments will not form an exteriorly digitally convex shape.

From Proposition 5 we know how to efficiently work out the optimal two-pixel decomposition for a two-pixel-decomposable and exteriorly digitally convex structuring element. The following procedure involves only three straightforward steps.

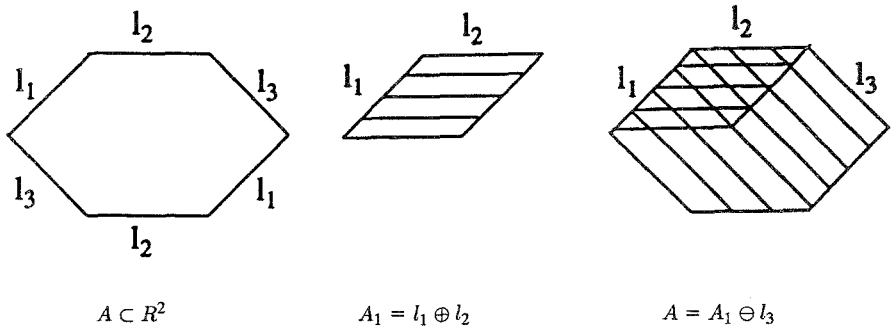

Fig. 6. Decomposition of an analog center-symmetric convex polygon of six sides by half of its analog boundary line segments.

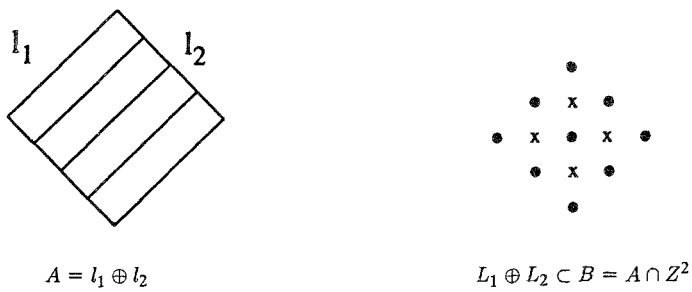

Fig. 7. The pixels in B marked by "x" are missing from $L_1 \oplus L_2$.

Step 1. Find the digital line segments that compose half of the exterior digital boundary line segments of the given structuring element.

Step 2. Optimally two-pixel-wise decompose each of those digital line segments.

Step 3. Perform the final decomposition of the given structuring element, which is given by the two-pixels used for decomposing the digital line segments in Step 2.

As figure 8 shows, the structuring element B consists of 43 pixels and its optimal decomposition uses six two-pixels. A direct dilation of X by B requires 43 image-translation operations plus 42 image-union operations, but the same dilation accomplished by using the six two-pixel decomposition requires only six image-translation operations plus five image-union operations, a tremendous saving of operations!

In general, for a two-pixel-decomposable structuring element B, whose optimal two-pixel decomposition uses m two-pixels, we need only m image translations plus $m - 1$ image-union (or image-intersection) operations in order to perform the binary dilation $X \oplus B$ (or erosion $X \ominus B$). Thus the complexity of performing binary dilation or erosion by using the optimal two-pixel decomposition of the structuring element has an upper bound of $2m$. If $B = L_1 \oplus \cdots \oplus L_k$, then $m = \sum_{i=1}^{k} \lceil \log_2 \#(L_i) \rceil$, where $\#(\cdot)$ denotes the number of pixels in $\cdot$ and $\lceil x \rceil \geq x$ represents the integer closest to x.

3 Grayscale-Structuring-Element Decomposition

3.1 Important Properties of Grayscale Dilation and Grayscale Erosion

Similar chain rule relations, or associative properties, hold for grayscale dilation and erosion:

$$(k_1 \oplus k_2) \oplus k_3 = k_1 \oplus (k_2 \oplus k_3),$$
$$(k_1 \ominus k_2) \ominus k_3 = k_1 \ominus (k_2 \oplus k_3).$$

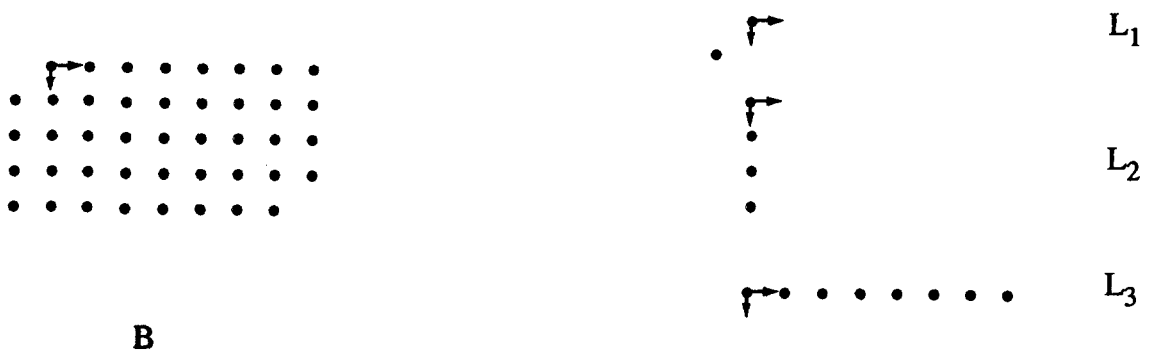

Fig. 8. Structing element $B = L_1 \oplus L_2 \oplus L_3$, where $L_1 = \{(0, 0), (1, -1)\}$, $L_2 = \{(0, 0), (1,0)\} \oplus \{(0, 0), (2, 0)\}$, and $L_3 = \{(0, 0), (0, 1)\} \oplus \{(0, 0), (0, 2)\} \oplus \{(0, 0), (0, 4)\}$.

This means that if

$$k = k_1 \oplus \cdots \oplus k_n, \tag{4}$$

then

$$f \oplus k = (\cdots[(f \oplus k_1) \oplus k_2] \oplus \cdots \oplus k_n), \tag{5}$$
$$f \ominus k = (\cdots[(f \ominus k_1) \ominus k_2] \ominus \cdots \ominus k_n). \tag{6}$$

Thus if a grayscale structuring element can be broken down to a chain of dilations of smaller substructuring elements, the desired operation may be performed as a string of suboperations. This property lends itself well to pipelining.

Another important property is that if

$$k = \max\{k_1, k_2\}, \tag{7}$$

then

$$f \oplus k = \max\{f \oplus k_1, f \oplus k_2\}, \tag{8}$$
$$f \ominus k = \min\{f \ominus k_1, f \ominus k_2\}, \tag{9}$$

where k_1, k_2, k are grayscale structuring elements, f represents a grayscale image, and max and min are pointwise operations.

3.2 Grayscale-Structuring-Element Representation and Decomposition

As was pointed out previously, a direct umbra representation for a grayscale image is not appropriate for the grayscale-structuring-element decomposition. This seems to explain why no efficient algorithm has been found to date for the grayscale structuring-element decomposition. If we do not take the umbra representation of the grayscale structuring element for granted, we may recognize that a grayscale structuring element k could always be represented as

$$k = \max\{k_1, \ldots, k_n\}, \tag{10}$$

where each lowercase letter k_i represents a flat grayscale substructuring element. We denote its defining domain by an uppercase letter K_i. More precisely,

$$\forall x \in K_i, \qquad k_i(x) = v_i = \text{const.}, \tag{11}$$

and for each i, $i = 1, \ldots, n-1$, K_i is included in K_{i+1}; that is,

$$K_1 \subset K_2 \subset \cdots \subset K_n. \tag{12}$$

Figure 9 shows a pyramid and its representation of form (10)–(12).

The representation (10), however, is too general to admit a two-pixel decomposition in general. This paper focuses on only the grayscale structuring elements that are most often used

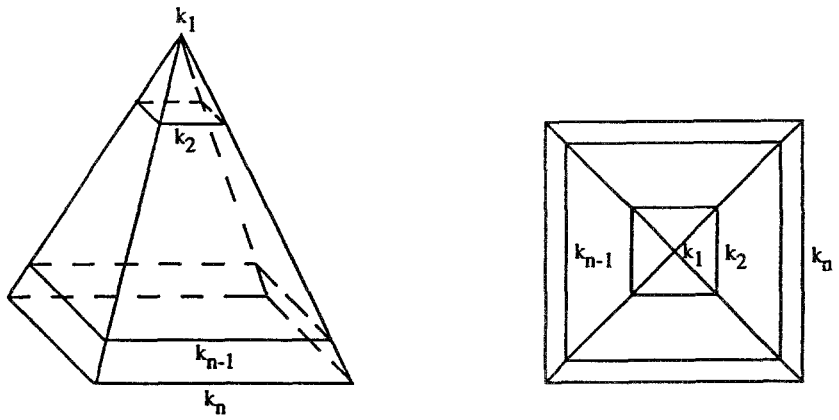

Fig. 9. A pyramid and its representation of form (10)–(12).

in morphological operations that justify a specialized hardware implementation in a general-purpose machine-vision system. For instance, those made into specialized hardware in the image flow computer fall into this category. Those needed possibly by only a specific application are not considered here.

From our knowledge about most frequently used grayscale structuring elements we may assume that each defining domain K_i is center symmetric, digitally convex, and two-pixel decomposable. Furthermore, if the translation is neglected, the two-pixel decomposition of K_{i+1} can be recursively formed from K_i in terms of an intermediate substructuring element S_i as follows:

$$\begin{aligned} K_1 &= \{0, a_1\} \oplus \{0, a_2\} \oplus \cdots \oplus \{0, a_{l_1}\}, \\ K_{i+1} &= K_i \oplus S_i, \\ S_i &= \{0, a_{l_{i+1}}\} \oplus \cdots \oplus \{0, a_{l_{i+1}}\}, \\ & \qquad i = 1, \ldots, n-1, \end{aligned} \tag{13}$$

where 0 represents the origin of Z^2 and $a_k \in Z^2$. It is obvious that the two-pixel decomposition of $K(= K_n)$ involves a total of l_n two-pixels.

Very often all S_i's are the same; that is,

$$S_1 = S_2 = \cdots = S_{n-1}, \tag{14}$$

and the values v_i's are monotonically decreasing or increasing. These properties, however, are not used in the following algorithmic development.

As is seen in subsection 3.3, the representation and decomposition (10)–(13) for the grayscale structuring element lend themselves well to recursively calculating grayscale dilation or erosion. The recursive algorithms developed thereof are also pipelineable.

3.3 Recursive Algorithms for Grayscale Dilation and Erosion

Due to (8)–(10), the dilation $f \oplus k$ and the erosion $f \ominus k$ take the following forms, respectively:

$$\begin{aligned} f \oplus k &= \max\{f \oplus k_1, \ldots, f \oplus k_n\}, \\ f \ominus k &= \min\{f \ominus k_1, \ldots, f \ominus k_n\}, \end{aligned} \tag{15}$$

where each dilation $f \oplus k_i$ or each erosion $f \ominus k_i$ is simply a dilation or an erosion by a flat grayscale substructuring element (equation (11)) defined on a two-pixel-decomposable domain (equations (12) and (13)). Due to (13), these dilations and erosions by a flat grayscale structuring element can be recursively performed. This is explained in detail for dilation in the following.

Let f_i denote the dilation $f \oplus k_i$, $i = 1, \ldots, n$, and let g_i, $i = 1, \ldots, n-1$, denote a grayscale function that is defined on the same domain as f_i and that differs from f_i by a constant $\lambda_i = v_{i+1} - v_i$; that is,

$$\forall x \in F \oplus K_i, \qquad g_i(x) = f_i(x) + \lambda_i, \quad i = 1, \ldots, n-1, \tag{16}$$

where F represents the defining domain of f, and $F \oplus K_i$ is the defining domain of f_i.

Let g_0 be a grayscale function that is defined on the same domain as the grayscale image f and that differs from f by a constant v_1; that is,

$$\forall x \in F, \qquad g_0(x) = f(x) + v_1. \tag{17}$$

Let s_i be defined on each intermediate substructuring element S_i (equation (13)) and take a zero value, $i = 1, \ldots, n-1$, and let s_0 be defined on K_1 and take a zero value also.

PROPOSITION 6. f_i, which represents the dilation $f \oplus k_i$, can be recursively formed as follows:

$$\begin{aligned} g_0 &= f + v_1, \\ f_i &= g_{i-1} \oplus s_{i-1}, \qquad i = 1, \ldots, n, \\ g_i &= f_i + \lambda_i, \qquad i = 1, \ldots, n-1, \end{aligned} \tag{18}$$

where the definition of $f_i + \lambda_i$ or $f + v_1$, i.e., a grayscale image plus a constant, is given by (16) or (17), respectively.

Proof. To prove (18) we first point out that the dilation of $k_i + \lambda_i$ by s_i gives k_{i+1} since $K_{i+1} = K_i \oplus S_i$, $v_{i+1} = v_i + \lambda_i$, and s_i is a zero-valued substructuring element defined on an intermediate substructuring element S_i; that is,

$$k_{i+1} = (k_i + \lambda_i) \oplus s_i. \tag{19}$$

Then we derive the following and finally validate (18):

$$\begin{aligned} f_{i+1} &= f \oplus k_{i+1} \\ &= (f \oplus ((k_i + \lambda_i) \oplus s_i)) \\ &= (f \oplus (k_i + \lambda_i)) \oplus s_i \\ &= ((f \oplus k_i) + \lambda_i) \oplus s_i \\ &= (f_i + \lambda_i) \oplus s_i \\ &= g_i \oplus s_i, \end{aligned}$$

where $(f \oplus k_i) + \lambda_i$ or $f_i + \lambda_i$ or g_i is defined by (16) and $k_i + \lambda_i$ is defined on K_i and takes the value $v_{i+1} = v_i + \lambda_i$.

In making the recursion (18) genuinely efficient, the calculation in (18) of the dilations $g_{i-1} \oplus s_{i-1}$ for $i = 1, \ldots, n$ is a major cost factor; here g_{i-1} is grayscale function defined on $F \oplus K_{i-1}$ and s_{i-1} is a zero-valued substructuring element defined on a two-pixel-decomposable intermediate substructuring element S_{i-1}. In the following we prove that these dilations can be recursively calculated. In fact, the following proposition holds.

PROPOSITION 7. The dilation $h \oplus r$ of an arbitrary grayscale function h by a zero-valued structuring element r defined on a two-pixel-decomposable domain R,

$$R = \{0, b_1\} \oplus \{0, b_2\} \oplus \cdots \oplus \{0, b_m\}, \tag{20}$$

can be recursively performed as follows:

$$\begin{aligned} h_1(x) &= \max\{h(x), h(x - b_1)\}, \\ h_i(x) &= \max\{h_{i-1}(x), h_{i-1}(x - b_i)\}, \\ & \qquad i = 2, \ldots, m, \end{aligned} \tag{21}$$

where h_m gives the desired dilation $h \oplus r$.

Proof. To validate (21) we notice that r can be chained as a dilation $r = r_1 \oplus \cdots \oplus r_m$, where r_k is defined on $\{0, b_k\}$ and takes a zero value. Thus the dilation $h \oplus r$ can be chained as $h \oplus r = (\cdots (h \oplus r_1) \oplus \cdots \oplus r_m)$ by (4) and (5), and each dilation in the chain turns out to be a special grayscale dilation discused in subseciton 2.1. Therefore each $g_{i-1} \oplus s_{i-1}$ having form $h \oplus r$ can be recursively calculated.

It is clear that recursive formulas (18) and (21) for sequentially calculating each dilation $f \oplus k_i$ are pipelineable and that so is the dilation $f \oplus k$, which is equal to the maximum of the n dilations in the sequence due to (15).

A similar procedure for the erosion $f \ominus k$ can be derived. By simply replacing + by $-$, $\oplus$ by $\ominus$, and max by min in (18) and (21), respectively, we obtain

$$\begin{aligned} g_0 &= f - v_1, \\ f_i &= g_{i-1} \ominus s_{i-1}, \qquad i = 1, \ldots, n, \\ g_i &= f_i - \lambda_i, \qquad i = 1, \ldots, n-1, \end{aligned} \tag{22}$$

and

$$\begin{aligned} h_1(x) &= \min\{h(x), h(x + b_1)\}, \\ h_i(x) &= \min\{h_{i-1}(x), h_{i-1}(x + b_i)\}, \\ & \qquad i = 2, \ldots, m, \end{aligned} \tag{23}$$

where f_i gives $f \ominus k_i$ and h_m calculates the desired erosion $h \ominus r$. In terms of (23), each $g_{i-1} \ominus s_{i-1}$ having form $h \ominus r$ can be recursively calculated. The combination of recursive formulas (22) and (23) for sequentially calculating each $f \ominus k_i$ are pipelineable, and so is the erosion $f \ominus k$, which is equal to the minimum of the erosions in the sequence due to (15).

It is easily verified that when f is a binary image and k is a binary structuring element, performing $f \oplus k$ or $f \ominus k$ in the binary domain by using the two-pixel decomposition has essentially the same efficiency as performing the same operations in the grayscale domain by using the recursive formulas.

Example 1. Let

$$\begin{aligned} K_1 &= \{(0, 0), (1, -1)\}, \\ K_2 &= K_1 \oplus S_1, \\ S_1 &= \{(0, 0), (1, 0)\} \oplus \{(0, 0), (2, 0)\}, \\ K_3 &= K_2 \oplus S_2, \\ S_2 &= \{(0, 0), (0, 1)\} \oplus \{(0, 0), (0, 2)\} \\ &\quad \oplus \{(0, 0), (0, 4)\}. \end{aligned}$$

Let k_i be defined on K_i and take a value v_i, $i = 1, 2, 3$, and let the gray scale structuring element k be defined as the maximum function of k_1, k_2, k_3 namely,

$$k = \max\{k_1, k_2, k_3\}.$$

It may be seen that the defining domain of k, i.e., $K = K_3$, consists of 43 pixels. A direct dilation of f by k takes the following form:

$$(f \oplus k)(x) = \max_{b \in K}\{f(x - b) + k(b)\}.$$

It requires 43 grayscale-image translations plus 43 grayscale-image additions and 42 maximum operations. Denote 43 pixels of K by $b_1, \ldots, b_{43}$. Let

$$\begin{aligned} c_i(x) &= f(x - b_i), \qquad i = 1, \ldots, 43, \\ d_i(x) &= c_i(x) + k(b_i), \qquad i = 1, \ldots, 43, \\ e_1(x) &= d_1(x), \\ e_i(x) &= \max\{e_{i-1}(x), d_i(x)\}, \\ &\qquad i = 2, \ldots, 43. \end{aligned}$$

Then

$$f \oplus k = e_{43}.$$

The same dilation can be accomplished by using the proposed representation and decomposition and the proposed recursive formulas, as follows:

$$\begin{aligned} g_0 &= f + v_1, \\ f_1(x) &= (g_0 \oplus s_0)(x) : (s_0 \text{ is defined on } K_1, \\ &\qquad \text{takes a zero value}) \\ &= \max\{g_0(x), g_0(x - (1, -1))\}, \\ g_1 &= f_1 + \lambda_1, \\ f_2(x) &= (g_1 \oplus s_1)(x) : (s_1 \text{ is defined on } S_1, \\ &\qquad \text{takes a zero value}) \\ h_1(x) &= \max\{g_1(x), g_1(x-(1, 0))\}, \\ h_2(x) &= \max\{h_1(x), h_1(x-(2, 0))\}, \\ f_2 &= h_2, \\ g_2 &= f_2 + \lambda_2, \\ f_3(x) &= (g_2 \oplus s_2)(x) : (s_2 \text{ is defined on } S_2, \\ &\qquad \text{takes a zero value}) \\ h_1(x) &= \max\{g_2(x), g_2(x-(0, 1))\}, \\ h_2(x) &= \max\{h_1(x), h_1(x-(0, 2))\}, \\ h_3(x) &= \max\{h_2(x), h_2(x-(0, 4))\}, \\ f_3 &= h_3, \\ f \oplus k &= \max\{f \oplus k_1, f \oplus k_2, f \oplus k_3\} \\ &= \max\{f_1, f_2, f_3\} \\ &= \max\{\max\{f_1, f_2\}, f_3\}. \end{aligned}$$

As was clearly shown, we need only six grayscale-image translations plus three grayscale-image additions and eight maximum operations, a tremendous saving of operations!

In general, for a gray scale structuring element having representation and decomposition (10)–(13), we need only l_n grayscale-image translations plus n grayscale-image additions (or subtractions) and $l_n + (n - 1)$ grayscale-image maximum (or minimum) operations to perform the grayscale dilation $f \oplus k$ (or erosion $f \ominus k$), where l_n maximum (or minimum) operations are used to form $f_i = f \oplus k_i$ (or $f_i = f \ominus k_i$), $i = 1, \ldots, n$, and the remaining $(n - 1)$ maximum (or minimum) operations to form either

$$\begin{aligned} f \oplus k = \max\{f_n, \max\{f_{n-1}, \ldots, \\ \max\{f_2, f_1\} \cdots\}\} \end{aligned}$$

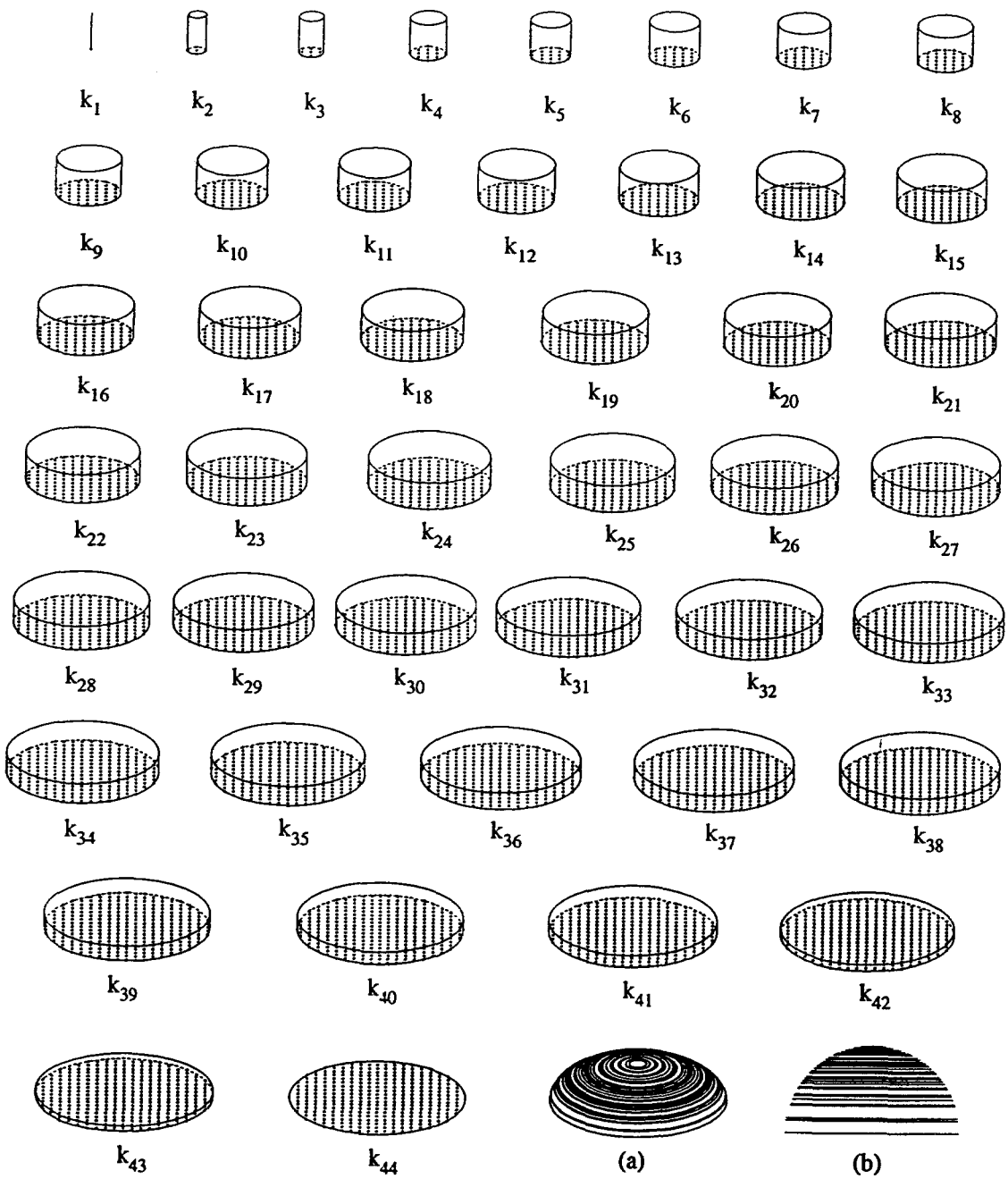

Fig. 10. 3D sphere of radius 10 ((a) shows the top view and (b) shows the front view) formed by using the proposed representation and decomposition.

or

$$f \ominus k = \min\{f_n, \min\{f_{n-1}, \ldots, \min\{f_2, f_1\}\cdots\}\}.$$

l_n represents the number of two-pixels involved in the two-pixel decomposition of $K(= K_n)$, and it is actually equal to the sum of numbers of two-pixels used for decomposing K_1 and each S_i for $i = 1, \ldots, n-1$. Therefore $l_n \geq n$ and the complexity of proposed recursive dilation and erosion algorithms has an upper bound of $4l_n$. As stated in section 2, the complexity of performing binary dilation or erosion by using the optimal two-pixel decomposition of the structuring element has an upper bound of $2m$, where m represents the number of two-pixels involved in the structuring-element two-pixel decomposition. Since combining the optimal two-pixel decompositions for K_1 and each S_i for $i = 1, \ldots, n-1$ forms a possibly suboptimal two-pixel decomposition for K, the complexity of grayscale dilation or erosion performed by using the proposed recursive algorithms is just twice as great as the complexity of binary dilation $F \oplus K$ or erosion $F \ominus K$ when the same suboptimal two-pixel decomposition is used for the structuring element K.

Example 2. It has been difficult to directly and accurately decompose a 3D sphere by using either two-pixel decomposition or cellular decomposition. Figure 10 shows the 3D sphere of radius 10 formed by using the proposed grayscale-structuring-element representation and decomposition.

4 Conclusion

Both binary and grayscale morphology are becoming increasingly important for machine vision. We have shown how to optimally decompose a binary two-pixel-decomposable structuring element and have provided recursive formulas for grayscale dilation and erosion. The grayscale structuring element was represented as the maximum function of a number of flat grayscale substructuring elements, each of which was defined on a two-pixel-decomposable domain. However, we did not directly decompose the corresponding umbra. As was seen, the decomposition process and morphological operations took place simultaneously; moreover, the proposed recursive formulas for both dilation and erosion were pipelineable. In the near future we will show those pipeline architecture designs.

In a recent paper by Xu [19] an elegant optimal cellular decomposition procedure was proposed for special convex polygonal structuring elements that actually were octagons or degenerate octagons with boundary line segments oriented only at a multiples of 45°. It is easily verified that the procedure will not work for a general 2D disk. Thus the problem of a cellular decomposition procedure for more general convex polygonal structuring elements remains unsolved.

Other recent advances are included in the papers by Richardson and Schafer [21] and by Gader [20]; in the former a lower bound for structuring-element decompositions is derived, and in the latter the linear-algebra approach is used to handle the decomposition problem.

Structuring-element decomposition is a special case of general shape decomposition. The latter is useful in shape recognition. It is challenging to develop an efficient procedure for morphologically decomposing a general convex polygon into meaningful subshapes.

Acknowledgment

The author thanks the reviewers for comments and suggestions.

References

1. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press: London, 1982.
2. S.R. Sternberg, "Morphology for grey tone functions," *Comput. Vis., Graph., Image Process*, vol 35, pp. 333–355, 1986.
3. R.M. Haralick, S.R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-9, pp. 523–550, 1987.
4. P. Maragos and R.W. Schafer, "Morphological filters—part I: Their set-theoretic analysis and relations to linear

shift invariant filters," *IEEE Trans. Acoust. Speech., Signal Process*, vol. ASSP-35, pp. 1153–1169, 1987.
5. P. Maragos and R.W. Schafer, "Morphological filters—part II : Their relations to median, order-statistics, and stack filters," *IEEE Trans. Acoust. Speech, Signal Process.* vol. ASSP-35, pp. 1170–1184, 1987.
6. R.M. Loughheed, D.L. McCubbrey, and S.R. Sternberg, "Cytocomputers: Architectures for parallel image processing," in *Proc. Workshop on Picture Data Description and Management*, Pacific Grove, CA, 1980, pp. 282–286.
7. S. Sternberg, "Cellular computers and biomedical image processing," in *Proc. Biomedical Images and Computers,* J. Sklansky and J.C. Bisonte, eds., Lecture Notes in Medical Informatics, vol. 17, Springer-Verlag: Berlin, 1980, pp. 274–319.
8. S.R. Sternberg, "Languages and architectures for parallel image processing," in *Proc. Conf. on Pattern Recognition in Practice*, L.N. Kanal and E.S. Gelsema eds., North-Holland: Amsterdam, 1980.
9. S.R. Sternberg, "Pipeline architectures for image processing," in *Multicomputers and Image Processing—Algorithms and Programs*, L. Uhr, ed., Academic Press: New York, 1982, pp. 291–305.
10. R.M. Lougheed, "Architectures and algorithms for digital image processing II," *Proc. Soc. Photo-Opt. Instrum. Eng.*, vol. 534, pp. 22–33, 1985.
11. J.L. Potter, "Image processing on the massively parallel processor," *Computer*, vol. 16, pp. 62–67, 1983.
12. J.P. Fitch, E.J. Coyle, and N.C. Gallagher, "Threshold decomposition of multidimensional ranked order operations," *IEEE Trans. Circuits and Systems*, vol. CAS-32, pp. 445-450, 1985.
13. X. Zhuang and R.M. Haralick, "Morphological structuring element decomposition," *Comput. Vis. Graph., Image Process.*, vol. 35, pp. 370–382, 1986.
14. G.X. Ritter and P.D. Gader, "Image algebra techniques for parallel image processing," *J. Parallel Distr. Comput.*, vol. 4 pp. 7–44, 1987.
15. L. Abbott, R.M. Haralick, and X. Zhuang, "Pipeline architecture for morphological image analysis," *Intl. J. Mach. Vis. Appl.*, vol. 1, pp. 23–40, 1988.
16. J.N. Wilson, G.R. Fischer, and G.X. Ritter, "Implementation and use of an image processing algebra for programming massively parallel machines," *Proc. Frontiers in Massively Parallel Computing*, October 1988.
17. F.Y. Shih and O.R. Mitchell, "Threshold decomposition of gray-scale morphology into binary morphology," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-11 pp. 31–42, 1989.
18. T. Kanungo, R.M. Haralick, and X. Zhuang, "B-code dilation and structuring element decomposition for restricted convex shapes," *Proc. Soc. Photo-Opt. Instrum. Eng.*, vol. 1350, pp. 419–430, 1990.
19. J. Xu, "Decomposition of convex polygonal morphological structuring elements into neighborhood subsets," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, pp. 153–162, 1991.
20. P. Gader, "P.D. separable decomposition and approximations of grayscale morphological templates," *Comput. Vis., Graph., Image Process.*, vol. 53, pp. 288–296, 1991.
21. C.H. Richardson and R.W. Schafer, " A lower bound for structuring element decomposition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, pp. 365–369, 1991.

Xinhua Zhuang graduated from Beijing University in 1963 after completing undergraduate and graduate programs in mathematics. Until 1983 he served as a senior research engineer at the Research Institute for Computer Technology, Hangzhou, China. He was visiting scientist in electrical engineering at the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, from 1983 to 1984, and was a visiting scientist in electrical and computer engineering at the University of Michigan, Ann Arbor, Michigan, under a grant from Machine Vision International, Ann Arbor, Michigan, from 1984 to 1985. He was selected as a consultant to the NATO Advisory Group for Aerospace Research and Development in 1985. He was a visiting research professor of the Coordinated Science Laboratory and a visiting professor of electrical and computer engineering at the University of Illinois at Urbana-Champaign from 1985 to 1986. He was a full professor of computer science and engineering and of optical instrument engineering and was the director of the Center for Intelligent Systems Research at the Zhejiang University, Hangzhou, China, from 1987 to 1989. He was a visiting professor of electrical engineering at the University of Washington at Seattle in 1989. Currently, he is an associate professor of electrical and computer engineering at the University of Missouri-Columbia. He has been a consultant for the NASA Goddard Space Flight Center; the Speech Technology Laboratory of Panasonic, Inc.; Siemans Medical Systems, Inc.; Neo Path, Inc. and the Norwegian Defense Research Establishment of NATO.

His professional interests lie in artificial intelligence, pattern recognition, computer and robot vision, signal and image processing, computer architecture, neural network computing and applied mathematics. He has published over 100 journal or conference papers. He is a contributor to six books, all published in the United States. He is a senior member of IEEE and serves as an associate editor of *IEEE Transactions on Image Processing*.