Gray-Scale Structuring Element Decomposition

Octavia I. Camps, *Member, IEEE*, Tapas Kanungo, *Student Member, IEEE*, and Robert M. Haralick, *Fellow, IEEE*

Abstract— **Efficient implementation of morphological operations requires the decomposition of structuring elements into the dilation of smaller structuring elements. Zhuang and Haralick presented a search algorithm to find optimal decompositions of structuring elements in binary morphology. In this paper, we use the concepts of Top of a set and Umbra of a surface to extend this algorithm to find an optimal decomposition of any arbitrary gray-scale structuring element.**

I. Introduction

WHEN the structuring element used in a morphological operation is larger than the largest element the hardware can handle in one stage, the structuring element must be decomposed into smaller structuring elements. Each of these elements has to be structured such that the hardware will be capable of handling it and such that the morphological composition is the given structuring element.

A tree-search algorithm that finds an optimal decomposition for binary structuring elements was proposed in [1]. All binary morphological operations are naturally extended to gray-scale imagery by using the Top and Umbra operations. In this paper, these ideas are used to extend the algorithm proposed in [1] for gray-scale structuring element decomposition.

In the next section, the related literature on morphological structuring element decomposition is discussed. In Section III, the basic definitions and notation used through the paper are given. In addition, many known morphological relations that are used later in proving some propositions are stated. In the following section, the formal statement of the gray-scale structuring element decomposition problem is given. In Section V, the morphological relations given in Section III are used to reduce the decomposition search space. The tree-search algorithm is described in Section VI, and in Section VII, an example to demonstrate the working of the algorithm is presented.

II. Related Work

There are many decomposition algorithms existing in the literature, e.g., [1]–[8]. All the algorithms impose some restriction on the shape of the structuring element. Zhuang and Haralick [1] presented an algorithm for n-point decomposition of binary structuring elements. The decomposition algorithm was a tree-search algorithm. The search space was reduced by using binary morphology constraints. Kanungo and Haralick [2], [3] and Xu [9] gave constant time algorithms for decomposition of convex binary structuring elements that have sides at orientations that are multiples of 45°. They expressed the decomposition as n-fold dilations of 13 primitive structuring elements, each of which fit into a 3×3 neighborhood. Richardson and Shafer [8] gave some bounds on the structuring element decomposition problems. All the algorithms mentioned above were binary decomposition algorithms and, with the exception of [1], are not generalizable to nontrivial gray-scale structuring elements. Jones and Svalbe [10] presented an algorithm for a basis decomposition of gray-scale morphological operations as opposed to a structural decomposition. In this approach, morphological filters are expressed as the supremum of erosions with the basis elements.

The gray-scale structuring element decomposition problem has been attempted by many researchers by performing a threshold decomposition on the structuring element [5], [6]. Here, the gray-scale image and structuring elements are decomposed into multiple binary images, and each binary image is processed separately in parallel. All the binary results are finally stacked to reconstruct the gray-scale result of the morphological processing. Ritter and Gader [7] proposed image algebra techniques for parallel image processing, and Gader [11] gave algorithms for decomposing gray-scale structuring elements with rectangular support into horizontal and vertical structuring elements. Zhuang [4] gave an algorithm for decomposing a gray-scale structuring element as a threshold decomposition but with the restriction that each binary component of the threshold decomposition is center symmetric, digitally convex, and two-point decomposable.

In this paper, we present a decomposition algorithm for an arbitrary gray-scale structuring element. Furthermore, we do not put any symmetry or convexity restriction on the structuring element as was done in other papers.

III. Preliminaries

In this section, we provide the basic notation and definitions used in this paper. We also provide some essential morphological relations that are used in propositions in the later sections.

A. Gray-Scale Morphology

For completeness, we begin by stating some definitions and propositions. An extended presentation of the definitions and the proof of the propositions in this subsection can be found in Haralick *et al.* [12].

Manuscript received December 15, 1993; revised March 7, 1995. The associate editor coordinating the review of this paper and approving it for publication was Prof. Rama Chellappa.

O. I. Camps is with the Department of Electrical Engineering and the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802 USA.

T. Kanungo and R. M. Haralick are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA.

Publisher Item Identifier S 1057-7149(96)00130-3.

First, we will provide the definitions of binary morphological operations: dilation and erosion. Next, we will then extend these definitions to gray-scale morphology.

Dilation is the morphological transformation that combines two sets using vector addition of set elements. If A and B are sets in E^N, the dilation of A by B is the set of all possible vector sums of pairs of elements: one coming from A and one coming from B.

Definition 1: The *dilation* of A by B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{c \in E^N \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\}.$$

Erosion is the morphological dual of dilation. If A and B are sets in E^N, then the erosion of A by B is the set of all elements of x for which $x + b \in A$ for every $b \in B$.

Definition 2: The *erosion* of A by B is denoted by $A \ominus B$ and is defined as

$$A \ominus B = \{x \in E^N \mid x + b \in A \text{ for every } b \in B\}.$$

The binary morphological operations of dilation, erosion, and opening will now be extended to gray-scale morphology by introducing the concept of the top surface of a set and the related concept of the umbra of a surface.

Definition 3: Let $A \subseteq E^N$ and $F = \{x \in E^N |$ for some $y \in E$, $(x, y) \in A\}$. The *top or top surface of* A, which is denoted by $T[A] : F \to E$, is defined by

$$T[A](x) = \max\{y | (x, y) \in A\}.$$

Definition 4: A set $A \subseteq E^N \times E$ is an *umbra* if and only if $(x, y) \in A$ implies that $(x, z) \in A$ for every $z \leq y$.

Definition 5: Let $F \subseteq E^N$ and $f : F \to E$. The *umbra of* f, which is denoted by $U[f]$, $U[f] \subseteq F \times E$, is defined by

$$U[f] = \{(x, y) \in F \times E | y \leq f(x)\}.$$

The top of the umbra of a function f is the function f itself.

Proposition 1: Let $F \subseteq E^N$ and $f : F \to E$. Then, $T[U[f]] = f$.

Having defined the operations of taking the Top of a set and the Umbra of a surface, gray-scale dilation and erosion can be defined:

Definition 6: Let F, $K \subseteq E^N$ and $f : F \to E$ and $k : K \to E$. The dilation of f by k is denoted by $f \oplus k$, $f \oplus k : F \oplus K \to E$ and is defined by

$$f \oplus k = T[U[f] \oplus U[k]].$$

Definition 7: Let F, $K \subseteq E^N$ and $f : F \to E$ and $k : K \to E$. The erosion of f by k is denoted by $f \ominus k$, $f \ominus k : F \ominus K \to E$ and is defined by

$$f \ominus k = T[U[f] \ominus U[k]].$$

Gray-scale dilation and erosion can be accomplished by using maximum and minimum operations.

Proposition 2: Let $f : F \to E$ and $k : K \to E$. Then, $f \oplus k : F \oplus K \to E$ can be computed by

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

Proposition 3: Let $f : F \to E$ and $k : K \to E$. Then, $f \ominus k : F \ominus K \to E$ can be computed by

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

Gray-scale opening is defined in an analogous way to opening in binary morphology.

Definition 8: Let $f : F \to E$ and $k : K \to E$. The opening of f by the structuring element k is denoted by $f \circ k$ and is defined by

$$f \circ k = (f \ominus k) \oplus k.$$

The following proposition expresses an opening operation in terms of the max and min operations. These relations are computational forms of the opening operation.

Proposition 4: Let $F, K \subseteq E^N$ and $f : F \to E$ and $k : K \to E$. Then, $f \circ k : F \circ K \to E$ and can be computed by

$$(f \circ k)(x) = \max_{\substack{u \in K \\ x - u \in F \ominus K}} \left\{ \min_{v \in K} \{f(x - u + v) - k(v)\} + k(u) \right\}.$$

Proof: The proof follows easily from the definitions of erosion and dilation. Let $R \subseteq E^N$ and $r : R \to E$ such that $r = f \ominus k$. Thus, from the definition of erosion

$$R = F \ominus K$$
$$r(x) = \min_{v \in K} \{f(x + v) - k(v)\}.$$

Since $f \circ k = (f \ominus k) \oplus k$, and $f \ominus k = r$, we have

$$\begin{aligned}(f \circ k)(x) &= (r \oplus k)(x) \\ &= \max_{\substack{u \in K \\ x - u \in R}} \{r(x - u) + k(u)\}.\end{aligned}$$

Substituting for R and $r(x - u)$ we have,

$$(f \circ k)(x) = \max_{\substack{u \in K \\ x - u \in F \ominus K}} \left\{ \min_{v \in K} \{f(x - u + v) - k(v)\} + k(u) \right\}.$$

Thus, the proposition is proved.

B. Translation in Gray-Scale Morphology

To solve the decomposition problem, we need to define the translation of a gray-scale structuring element. In this section, we define the translation of a structuring element by a 1-point function and derive some useful properties.

Definition 9: Let $F \subseteq E^N$, $f : F \to E$, $G = \{\alpha\} \subseteq E^N$, and $g : G \to E$. The translation of f by g is denoted by $f_g : F_\alpha \to E$ and is defined by

$$f_g(x) = f(x - \alpha) + g(\alpha)$$

for every $x \in F_\alpha = \{x \in E^N | x = (y + \alpha)$ for some $y \in F\}$.

Consider $F \subseteq E^N$, $f : F \to E$, $G = \{\alpha\} \subseteq E^N$, $g : G \to E$, and $(\alpha, g(\alpha)) \in E^{N+1}$. The gray-scale translation has the following properties:

Proposition 5: $U[f_g] = (U[f])_{(\alpha,g(\alpha))}$.

Proof: Let $(x,y) \in E^{N+1}$. The point $(x,y) \in (U[f])_{(\alpha,g(\alpha))}$ if and only if there exists $(u,v) \in U[f]$ such that $(x,y) = (u,v) + (\alpha, g(\alpha))$. By definition of umbra $(u,v) \in U[f]$ if and only if $v \leq f(u)$. Then

$$y = v + g(\alpha) \leq f(u) + g(\alpha) = f(x-\alpha) + g(\alpha) = f_g(x).$$

Hence, $(x,y) \in U[f_g]$.

Proposition 6: $(T[U[f]])_g = T[(U[f])_{(\alpha,g(\alpha))}]$.

Proof: Let $x \in E^N$. Then, by Propositions 5 and 1

$$\begin{aligned} T[(U[f])_{(\alpha,g(\alpha))}](x) &= T[U[f_g]](x) \\ &= f_g(x) = (T[U[f]])_g(x). \end{aligned}$$

In the following proposition, we show that the translation of a function is equivalent to a dilation.

Proposition 7: $f_g = f \oplus g$.

Proof: By definition of dilation, $f \oplus g = T[U[f] \oplus U[g]]$. A point $(x,y) \in U[f] \oplus U[g]$ if and only if there exists $(x_1,y_1) \in U[f]$, and $(x_2,y_2) \in U[g]$ such that

$$(x,y) = (x_1,y_1) + (x_2,y_2).$$

However, $(x_1,y_1) \in U[f]$ implies $y_1 \leq f(x_1)$, and $(x_2,y_2) \in U[g]$ implies $x_2 = \alpha$ and $y_2 \leq g(x_2) = g(\alpha)$. Then

$$\begin{aligned} y &= y_1 + y_2 \\ &\leq f(x_1) + g(\alpha) \\ &= f(x-\alpha) + g(\alpha) \\ &= f_g(x). \end{aligned}$$

Hence, by definition of Top

$$\begin{aligned} (f \oplus g)(x) &= \max\{z | z \leq f(x-\alpha) + g(\alpha)\} \\ &= f(x-\alpha) + g(\alpha) \\ &= f_g(x). \end{aligned}$$

A useful relationship is that the dilation operation commutes with translation.

Proposition 8: Let $f : E^N \to E$, and $k : E^N \to E$. Then, $f_g \oplus k = (f \oplus k)_g$.

Proof: By definition of dilation

$$f_g \oplus k = T[U[f_g] \oplus U[k]].$$

By Proposition 6

$$\begin{aligned} f_g \oplus k &= T[(U[f] \oplus U[k])_{(\alpha,g(\alpha))}] \\ &= (T[U[f] \oplus U[k]])_g \\ &= (f \oplus k)_g. \end{aligned}$$

As in the case of dilations, the erosion operation also commutes with translation.

Proposition 9: Let $f : E^N \to E$, and $k : E^N \to E$. Then, $f_g \ominus k = (f \ominus k)_g$.

Proof: The proof is analogous to the one for Proposition 8.

If a known function f has been translated by an unknown function g, then the function g can be recovered by performing an erosion.

Proposition 10: If F is finite, then $f_g \ominus f = g$.

Proof: By Proposition 9,

$$f_g \ominus f = (f \ominus f)_g.$$

From Proposition 8 and the definition of erosion,

$$(f \ominus f)_g = (f \ominus f) \oplus g = g.$$

IV. Statement of the Problem

The gray-scale structuring element decomposition problem is formally stated as follows:

Given a space E, a structuring element s with $s : S \to E$, $S \subseteq E^N$, and an integer n, find the smallest integer M and the corresponding structuring elements $h_1, h_2, \ldots, h_M$, with $h_i : H_i \to E$, $H_i \subseteq E^N$, and $\#H_i \leq n$ for $i = 1, \ldots, M$ such that

$$s(x) = (h_1 \oplus h_2 \oplus \cdots \oplus h_M)(x).$$

In this paper, we will consider that the structuring element s is a discrete function defined in a discrete finite domain S.

V. Constraining the Search Space

The problem we solve in this section is to construct a decomposition of $s : S \to E$ into $h_1, h_2, \ldots, h_M$ with $h_i : H_i \to E$ for $i = 1, \ldots, M$ having the smallest M, if one exists, where each H_i has no more than n points. We also assume that the domain of s, which is S, is finite.

To determine such a decomposition of s, if one exists, requires a combinatorial search process for the domains H_i and for the values of the h_i functions at each point of their domains. Our algorithm limits the possible domains H_i and the possible values of the functions h_i and thereby greatly reduces the search space.

A. Translation Constraints

The gray-scale structuring element decomposition is equivalent to one with fewer degrees of freedom, where all the structuring elements used in the decomposition contain the origin.

Proposition 11: Let $s : S \to E$, $S \subseteq E^N$, $s = h_1 \oplus h_2 \oplus \cdots \oplus h_M$, with $h_i : H_i \to E$, $H_i \subseteq E^N$, $\#H_i \leq m$ for $i = 1, \ldots, M$. Then, there exist $g : G \to E$, $G = \{\alpha\} \subseteq E^N$, and $j_1, j_2, \ldots, j_M$, $j_i : J_i \to E, J_i \subseteq E^N$, $i = 1, \ldots, M$ such that we have the following:

1) $\#J_i = \#H_i$ for $i = 1, \ldots, M$.
2) $j_i(0) = 0$ for $i = 1, \ldots, M$.
3) $s = g \oplus j_1 \oplus j_2 \oplus \cdots \oplus j_M$.
4) $g = s \ominus \{j_1 \oplus j_2 \oplus \cdots \oplus j_M\}$.

Proof: Let α_i be such that $h_i(\alpha_i) \geq h_i(x)$ for every $x \in H_i$. Define $j_i = (h_i)_{\check{g}_i}$, where $g_i : G_i \to E$, $G_i = \{\alpha_i\}$, $g_i(\alpha_i) = h_i(\alpha_i)$, and $\check{g}_i(x) = -g_i(-x)$. Then, $j_i(0) = h_i(0 + \alpha_i) - h_i(\alpha_i) = 0$. Note that with this definition h_i and j_i have the same number of points.

Next, we will show that $s = g \oplus j_1 \oplus j_2 \oplus \cdots \oplus j_M$. Since $j_i = (h_i)_{\check{g}_i}$, by the definition of translation, we have that

$h_i = (j_i)_{g_i}$. Therefore

$$s = h_1 \oplus h_2 \oplus \cdots \oplus h_M = (j_1)_{g_1} \oplus (j_2)_{g_2} \oplus \cdots \oplus (j_M)_{g_M}.$$

By Proposition 7

$$s = g \oplus (j_1 \oplus j_2 \oplus \cdots \oplus j_M)$$

where

$$g = (g_1 \oplus g_2 \oplus \cdots \oplus g_M).$$

Furthermore, by Proposition 10

$$g = s \ominus (j_1 \oplus j_2 \oplus \cdots \oplus j_M).$$

Proposition 11 in effect shows that the original problem can be reduced to one with fewer degrees of freedom. Since $j_i(0) = 0$, there is one less unknown point to determine each j_i compared with the number of unknown points in the corresponding h_i. Furthermore, the extra unknown g is determined without searching once all the j_i have been determined.

B. Domain and Functional Value Constraints

The search for the decomposition elements j can be reduced by establishing constraints on their domains and functional values. These constraints are based on the fact that if j is an element of the decomposition of s, it must be true that s is open under j, i.e., $s = s \circ j$. This property follows directly from the following propositions:

The opening of a set is *antiextensive*, i.e., the opening of a set A by a set B is always a subset of A.

Proposition 12: Let $A \subseteq E^N$ and $B \subseteq E^N$. Then, $A \circ B \subseteq A$.

Proof: Let $a \in A \circ B$. Then, there exists an $x \in A \ominus B$ and $b \in B$ such that $a = x + b$. However, $x \in A \ominus B$ implies that for every $y \in B$, $x + y \in A$. Since $b \in B$, $x + b \in A$, but $a = x + b$ so that $a \in A$.

If a set is equal to the dilation of two sets, then it is open under either one of these sets.

Proposition 13: Let $A = B \oplus C$. Then, $A \circ C = A$.

Proof: Let $a \in A$. Since $A = B \oplus C$, there exists a $b \in B$ and a $c \in C$ such that $a = b + c$, but for every $x \in B$ and $c \in C$, $x + c \in A$. Since $b \in B$, we must have, for every $y \in C$, $b + y \in A$. This implies that $b \in A \ominus C$ and $b \in B$ imply $a \in (A \ominus C) \oplus C$. Hence, $A \subseteq (A \circ C)$. By Proposition 12, $A \circ C \subseteq A$. Thus, $A \circ C = A$.

Finally, if a gray-scale structuring element is equal to the dilation of two structuring elements, then it is open under either of these structuring elements.

Proposition 14: Let $s = a \oplus j$. Then, $s = s \circ j$.

Proof: By definition of dilation, $s(x) = T[U[a] \oplus U[j]](x)$. Taking the umbra at both sides, $U[s] = U[a] \oplus U[j]$. By Proposition 13, $U[s] = U[s] \circ U[j]$. Then, $s(x) = T[U[s] \circ U[j]](x) = (s \circ j)(x)$.

1) Domain Constraints: In this section, it will be shown that the domains of the candidate structuring elements are made of points given by the difference of two points in the domain of the given structuring element s. That is, any point t in the domain of a candidate structuring element J must be equal to the difference of two points x_1 and x_2 in the domain of the structuring element S.

Proposition 15: Let $S = S \circ J$, $0 \in J$, and $t \in J$. Then, there exists x_1 and x_2 in S such that $t = x_1 - x_2$.

Proof: Consider $t \in J$. Since $S = (S \ominus J) \oplus J$, by definition of dilation, there exist $x_1 \in S$ and x_2 in $S \ominus J$ such that

$$x_1 = x_2 + t.$$

By definition of erosion, $x_2 + y \in S$ for all $y \in J$. In particular, $0 \in J$. Thus, $x_2 \in S$, and

$$t = x_1 - x_2$$

with x_1 and x_2 belonging to S.

2) Functional Value Constraints: The search for the decomposition elements j can be further reduced by using constraints on their functional values. Let $J \subset E^N$, $0 \in J$ be the n-point domain of the decomposition element j. It can be shown that for every t in J, there exists x_1 and x_2 in S such that $s(x_1 + t) - s(x_1) \geq j_i(t) \geq s(x_2) - s(x_2 - t)$. This relation can be used to reduce the search space by pruning out the candidates that do not satisfy this constraint. Next, we formally prove this relationship.

Proposition 16: Let $s : S \to E$ be a gray-scale function with domain $S \subseteq E^N$, and let $j : J \to E$ be an n-point structuring element such that $0 \in J \subseteq E^N$, and $j(0) = 0$. If s is open under j, $s = s \circ j$, then for each $x \in S$, at least one of the following is true:

1) $x \in S \ominus J$ and $j(t) \leq s(x + t) - s(x)$ for each $t \in J$; and/or
2) there exists $t \in J$ such that $x - t \in S \ominus J$ and $j(t) \geq s(x) - s(x - t)$.

Proof: Consider $x \in S$. Since s is open under j, we have

$$s(x) = (s \circ j)(x)$$

and from the definition of opening and Proposition 2, we have

$$s(x) = \max_{\substack{u \in J \\ x-u \in S \ominus J}} \{(s \ominus j)(x - u) + j(u)\}. \tag{1}$$

From Proposition 3

$$(s \ominus j)(x - u) = \min_{v \in J} \{s(x - u + v) - j(v)\}$$

and (1) can be rewritten as

$$s(x) = \max_{\substack{u \in J \\ x-u \in S \ominus J}} \left\{ \min_{v \in J} \{s(x - u + v) - j(v)\} + j(u) \right\}. \tag{2}$$

Let $u_o \in J$ be the point where the maximum in the right-hand side of (1) is achieved. That is, let u_o be a point in J such that $x - u_o \in S \ominus J$ and

$$\begin{aligned} s(x) &= (s \ominus j)(x - u_o) + j(u_o) \\ &= \max_{\substack{u \in J \\ x-u \in S \ominus J}} \{(s \ominus j)(x - u) + j(u)\}. \end{aligned} \tag{3}$$

Using (2) and (3), we get the following two constraints:

$$s(x) \geq \min_{v \in J} \{s(x-u+v) - j(v)\} + j(u) \quad (4)$$
$$\text{for all } u \in J,\ x-u \in S \ominus J$$
$$s(x) = \min_{v \in J} \{s(x-u_o+v) - j(v)\} + j(u_o). \quad (5)$$

It can be seen that (4) does not provide any useful constraint since $s(x) - j(u) = s(x-u+v) - j(v)$ for $v = u$, and the relation $a \geq \min\{b_1, b_2, \ldots, a, \ldots, b_n\}$ is a tautology, that is, it is true for all values of a and $b_1, \ldots, b_n$. However, on rearranging the terms in (5), we do get a useful constraint:

$$s(x) - j(u_o) \leq s(x-u_o+v) - j(v) \text{ , for all } v \in J. \quad (6)$$

We will consider two cases $u_o = 0$ and $u_o \neq 0$.

Case I ($u_o = 0$, $x \in S \ominus J$): Consider (6) for $u_o = 0$. Using the fact that $j(0) = 0$, we have

$$j(v) \leq s(x+v) - s(x) \text{ , for all } v \in J.$$

Thus, we have proved the first condition. Note that if $x \notin S \ominus J$, this condition is automatically not valid since the term corresponding to $u = 0$ is not a term in the maximization function in (1). Further, note that since $S = S \circ J$ and $0 \in J$, there must exist at least one $x \in S$ such that $x \in S \ominus J$.

Case II ($u_o \neq 0$, $x - u_o \in S \ominus J$): Consider (6) for $v = 0$. Since $j(0) = 0$, we have

$$s(x) - s(x-u_o) \leq j(u_o).$$

Thus, we have proved the second condition. Note that if $x - u_o \notin S \ominus J$, this condition is automatically not valid.

Finally, since $S = (S \ominus J) \oplus J$, for every $t \in J$, there exist $x \in S$ such that $x - t \in S \ominus J$. Thus, the above constraints can be applied for every $t \in J$.

In the Appendix, we provide an example to illustrate how the constraints are used to detect not decomposable structuring elements.

C. A Morphological Opening Constraint

Assume that we know that s is not open under the structuring element a. Then, the result in this section shows that s will not be open under any structuring element k such that $k = a \oplus b$, where b is any other structuring element. This result is useful in look-ahead pruning of the search space.

Proposition 17: If $S = S \circ (A \oplus B)$, then $S = S \circ A$.

For a proof, see Haralick *et al.* [12].

Proposition 18: If $s = s \circ (a \oplus b)$, then $s = s \circ a$.

Proof: By hypothesis, $s(x) = T[U[s] \circ U[a \oplus b]](x)$. Then, by Proposition 17, $U[s] = U[s] \circ (U[a] \oplus U[b]) = U[s] \circ U[a]$. Therefore, $s(x) = T[U[s] \circ U[a](x)$, and $s(x) = (s \circ a)(x)$.

VI. A Tree-Search Algorithm

The algorithm to accomplish the decomposition of a gray-scale structuring element s consists of a breadth-first tree search with forward checking. This algorithm is a generalization of the algorithm proposed in [1].

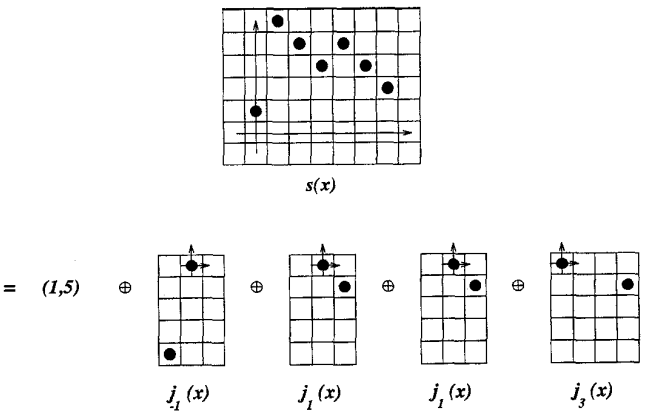


Fig. 1. Decomposition of a gray-scale structuring element. The structuring element s is decomposed as $s = g \oplus j_{-1} \oplus j_1 \oplus j_1 \oplus j_3$, where $g(1) = 5$ is a one point structuring element representing a translation, and j_i are two-point gray-scale structuring elements, each containing the origin (0,0).

A node i in the tree corresponds to a candidate structuring element $j_i(x)$. Each node i also has associated with it the following entities:

1) a list of all its possible descendents L_i
2) the partial decomposition so far $k_i = j_1 \oplus j_2 \cdots \oplus j_i$ corresponding to its dilation with all its predecessor nodes
3) the undecomposed part or residue $t_i = s \ominus k_i$.

The root of the tree is initialized such that $L_o = j_1, j_2, \ldots, j_M$ is the set of all possible structuring elements, $k_o = \{0\}$, and $t_o = s$.

The following considerations are used in reducing the tree search:

1) Since dilation is commutative, for a node associated with j_q, there is only need to consider the possible descendents $j_{q+1}, j_{q+2}, \ldots, j_J$.
2) Forward checking can be used to control the growth of the tree. We will show that if at some level l, $j_1, j_2, \ldots, j_l$ have been determined, then the only j's that need to be considered for any node in the subtree below are those that satisfy
$$(t_l \ominus j) \oplus j \oplus k_l = s$$
where $t_l = s \ominus k_l$, and $k_l = j_1 \oplus j_2 \oplus \cdots \oplus j_l$. A proof of this is given in Appendix B.
3) Proposition 18 shows that if s is not open under $k_l \oplus j$, then it is not open under $k_m \oplus j$, where k_m is the partial decomposition at a node in the subtree below l. This condition can be used as a forward check to prune the search tree.
4) If the hardware has a constraint on neighborhood size, then all the candidate j_i's that do not satisfy this constraint at a particular node can be eliminated. An example of this type of constraint is when the domain J_i of the function j_i has to fit inside a 3×3 neighborhood.

The algorithm proceeds as follows. Before opening a node in the tree, a forward check is made through the possible descendents of the node. The checking eliminates those structuring elements in L_i that do not satisfy the forward check

constraints. Once the forward checking is finished, the nodes corresponding to the elements in L_i that survived the test are opened. Any decomposition found at the lower level of the tree is an optimal decomposition. A decomposition is found when the number of elements in the domain of the residue t is one, and it corresponds to the 1-point function g.

VII. Example

In the following, we give two examples illustrating the use of the algorithm to decompose gray-scale structuring elements into sets of two and three points.

A. Two-Point Decomposition

Let $s : S \rightarrow E$ be the gray-scale structuring element to be decomposed, where

$$S = \{0, 1, 2, 3, 4, 5, 6\}$$

and

$$\begin{array}{ccccccc} s(0) & s(1) & s(2) & s(3) & s(4) & s(5) & s(6) \\ 1 & 5 & 4 & 3 & 4 & 3 & 2 \end{array} \tag{7}$$

as shown in Fig. 1.

There are 12 candidate domains for the structuring elements j_i that satisfy the domain constraint derived in Proposition 15:

$$\begin{array}{ll} J_1 = \{0, 1\} & J_{-1} = \{0, -1\} \\ J_2 = \{0, 2\} & J_{-2} = \{0, -2\} \\ J_3 = \{0, 3\} & J_{-3} = \{0, -3\} \\ J_4 = \{0, 4\} & J_{-4} = \{0, -4\} \\ J_5 = \{0, 5\} & J_{-5} = \{0, -5\} \\ J_6 = \{0, 6\} & J_{-6} = \{0, -6\} \end{array} \tag{8}$$

The erosions of S with the domains J_i, $i = -6, \ldots, 6$ are given by

$$\begin{array}{ll} S \ominus J_1 = \{0, 1, 2, 3, 4, 5\} & S \ominus J_{-1} = \{1, 2, 3, 4, 5, 6\} \\ S \ominus J_2 = \{0, 1, 2, 3, 4\} & S \ominus J_{-2} = \{2, 3, 4, 5, 6\} \\ S \ominus J_3 = \{0, 1, 2, 3\} & S \ominus J_{-3} = \{3, 4, 5, 6\} \\ S \ominus J_4 = \{0, 1, 2\} & S \ominus J_{-4} = \{4, 5, 6\} \\ S \ominus J_5 = \{0, 1\} & S \ominus J_{-5} = \{5, 6\} \\ S \ominus J_6 = \{0\} & S \ominus J_{-6} = \{6\}. \end{array} \tag{9}$$

The possible functional values of the structuring elements are determined by using the constraints derived in Proposition 16. Each structuring element $j_i(t)$ has to satisfy either one or both of the following conditions:

1) $x \in S \ominus J_i$ and $j_i(t) \leq s(x + t) - s(x)$ or
2) $(x - t) \in S \ominus J_i$ and $j_i(t) \geq s(x) - s(x - t)$,

and the additional constraints

1) $j_i(0) = 0$ and
2) $j_i(x) \leq 0$ for all $x \in J_i$.

Table I lists the values of $s(x + t) - s(x)$ for $x \in S \ominus J_i$ and $s(x) - s(x - t)$ for $(x - t) \in S \ominus J_i$ for J_i, $i = -6, \ldots, 6$. Entries of the form "—" correspond to elements x that do not satisfy the necessary condition $x \in S \ominus J_i$ or $x - t \in S \ominus J_i$.

From these values, the following constraints are found:

1) Since the columns for $J_4, J_5, J_6, J_{-4}, J_{-5}$, and J_{-6} have rows where neither of the differences defined above are defined, s is not opened under any structuring element having one of these domains.
2) The functional constraints for the rest of the domains are found by applying
   a) $x \in S \ominus J_i$ and $j_i(t) \leq s(x + t) - s(x)$ **or**
   b) $(x - t) \in S \ominus J_i$ and $j_i(t) \geq s(x) - s(x - t)$

   where $s(x + t) - s(x)$ and $s(x) - s(x - t)$ are given in the respective columns and rejecting all the functional values that are not compatible with at least one of the two conditions. For example, looking at the column corresponding to J_1, we have the following constraints:
   a) $x = 0$: $j_1(1) \leq 4$; thus, reject $j_1(1) > 4$;
   b) $x = 1$: $j_1(1) \leq -1$ or $j_1(1) \geq 4$; thus, reject $-1 < j_1(1) < 4$;
   c) $x = 2$: $j_1(1) \leq -1$ or $j_1(1) \geq -1$; thus, $j_1(1)$ is not restricted by this condition;
   d) $x = 3$: $j_1(1) \leq 1$ or $j_1(1) \geq -1$; thus, $j_1(1)$ is not restricted from this condition;
   e) $x = 4$: $j_1(1) \leq -1$ or $j_1(1) \geq 1$; thus, reject $-1 < j_1(1) < 1$;
   f) $x = 5$: $j_1(1) \leq -1$ or $j_1(1) \geq -1$; thus, $j_1(1)$ is not restricted by this condition; and
   g) $x = 6$: $j_1(1) \geq -1$; thus, reject $j_1(1) < -1$.

   The summary of all the constraints for the remaining structuring elements are illustrated in Fig. 2(a)–(f), where shaded regions correspond to the values that $j_i(t)$ cannot take if $s(x)$ is open under each of them. From these diagrams, we see that $s(x)$ is not open under any structuring element with domain J_2, J_{-2} and J_{-3}. Furthermore, $s(x)$ is open under the following structuring elements:

$$\begin{array}{ll} j_1(0) = 0 & j_1(1) = -1 \\ j_3(0) = 0 & j_3(3) = -1 \\ j_{-1}(0) = 0 & j_{-1}(-1) = -4. \end{array} \tag{10}$$

Fig. 3 shows the tree created by the algorithm. The optimal decomposition of s $s = g \oplus j_{-1} \oplus j_1 \oplus j_1 \oplus j_3$, $g(1) = 5$ is shown in Fig. 1.

B. Three-Point Decomposition

Let $s : S \rightarrow E$ be the gray-scale structuring element to be decomposed using three-point gray-scale elements, where

$$S = \{0, 1, 2, 3, 4\}$$

and

$$\begin{array}{ccccc} s(0) & s(1) & s(2) & s(3) & s(4) \\ 7 & 9 & 13 & 11 & 10 \end{array} \tag{11}$$

as shown in Fig. 4.

There are 28 possible differences to form the domains of the candidate structuring elements. The possible functional values of the structuring elements are determined by using the constraints derived in Proposition 15, resulting in a total of 378

TABLE I

VALUES OF $s(x+t)-s(x)$ FOR $x \in S \ominus J_i$ AND $s(x)-s(x-t)$ FOR $(x-t) \in S \ominus J_i$ FOR J_i, $I=-6,\ldots,6$. ENTRIES OF THE FORM "—" CORRESPOND TO ELEMENTS x THAT DO NOT SATISFY THE NECESSARY CONDITION $x \in S \ominus J_i$ OR $x-t \in S \ominus J_i$

	$J_1=\{0,1\}$		$J_2=\{0,2\}$		$J_3=\{0,3\}$	
x	$s(x+1)-s(x)$	$s(x)-s(x-1)$	$s(x+2)-s(x)$	$s(x)-s(x-2)$	$s(x+3)-s(x)$	$s(x)-s(x-3)$
$x \in S$	$x \in S \ominus J_1$	$(x-1) \in S \ominus J_1$	$x \in S \ominus J_2$	$(x-2) \in S \ominus J_2$	$x \in S \ominus J_3$	$(x-3) \in S \ominus J_3$
0	4	—	3	—	2	—
1	-1	4	-2	—	-1	—
2	-1	-1	0	3	-1	—
3	1	-1	0	-2	-1	2
4	-1	1	-2	0	—	-1
5	-1	-1	—	0	—	-1
6	—	-1	—	-2	—	-1

	$J_4=\{0,4\}$		$J_5=\{0,5\}$		$J_6=\{0,6\}$	
x	$s(x+4)-s(x)$	$s(x)-s(x-4)$	$s(x+5)-s(x)$	$s(x)-s(x-5)$	$s(x+6)-s(x)$	$s(x)-s(x-6)$
$x \in S$	$x \in S \ominus J_4$	$(x-4) \in S \ominus J_4$	$x \in S \ominus J_5$	$(x-5) \in S \ominus J_5$	$x \in S \ominus J_6$	$(x-6) \in S \ominus J_6$
0	3	—	2	—	1	—
1	-2	—	-3	—	—	—
2	-2	—	—	—	—	—
3	—	—	—	—	—	—
4	—	3	—	—	—	—
5	—	-2	—	2	—	—
6	—	-2	—	-3	—	1

	$J_{-1}=\{0,-1\}$		$J_{-2}=\{0,-2\}$		$J_{-3}=\{0,-3\}$	
x	$s(x-1)-s(x)$	$s(x)-s(x+1)$	$s(x-2)-s(x)$	$s(x)-s(x+2)$	$s(x-3)-s(x)$	$s(x)-s(x+3)$
$x \in S$	$x \in S \ominus J_{-1}$	$(x+1) \in S \ominus J_{-1}$	$x \in S \ominus J_{-2}$	$(x+2) \in S \ominus J_{-2}$	$x \in S \ominus J_{-3}$	$(x+3) \in S \ominus J_{-3}$
0	—	-4	—	-3	—	-2
1	-4	1	—	2	—	1
2	1	1	-3	0	—	1
3	1	-1	2	0	-2	1
4	-1	1	0	2	1	—
5	1	1	0	—	1	—
6	1	—	2	—	1	—

	$J_{-4}=\{0,-4\}$		$J_{-5}=\{0,-5\}$		$J_{-6}=\{0,-6\}$	
x	$s(x-4)-s(x)$	$s(x)-s(x+4)$	$s(x-5)-s(x)$	$s(x)-s(x+5)$	$s(x-6)-s(x)$	$s(x)-s(x+6)$
$x \in S$	$x \in S \ominus J_{-4}$	$(x+4) \in S \ominus J_{-4}$	$x \in S \ominus J_{-5}$	$(x+5) \in S \ominus J_{-5}$	$x \in S \ominus J_{-6}$	$(x+6) \in S \ominus J_{-6}$
0	—	-3	—	-2	—	-1
1	—	2	—	3	—	—
2	—	2	—	—	—	—
3	—	—	—	—	—	—
4	-3	—	—	—	—	—
5	2	—	-2	—	—	—
6	2	—	3	—	-1	—

possible elements. However, when these candidate elements are examined using forward checking, only the two elements j_1 and j_2 given in (12) survive the test:

$$\begin{array}{lll} j_1(-2)=-6 & j_1(-1)=-4 & j_1(0)=0 \\ j_2(0)=0 & j_2(1)=-2 & j_2(2)=-3. \end{array} \tag{12}$$

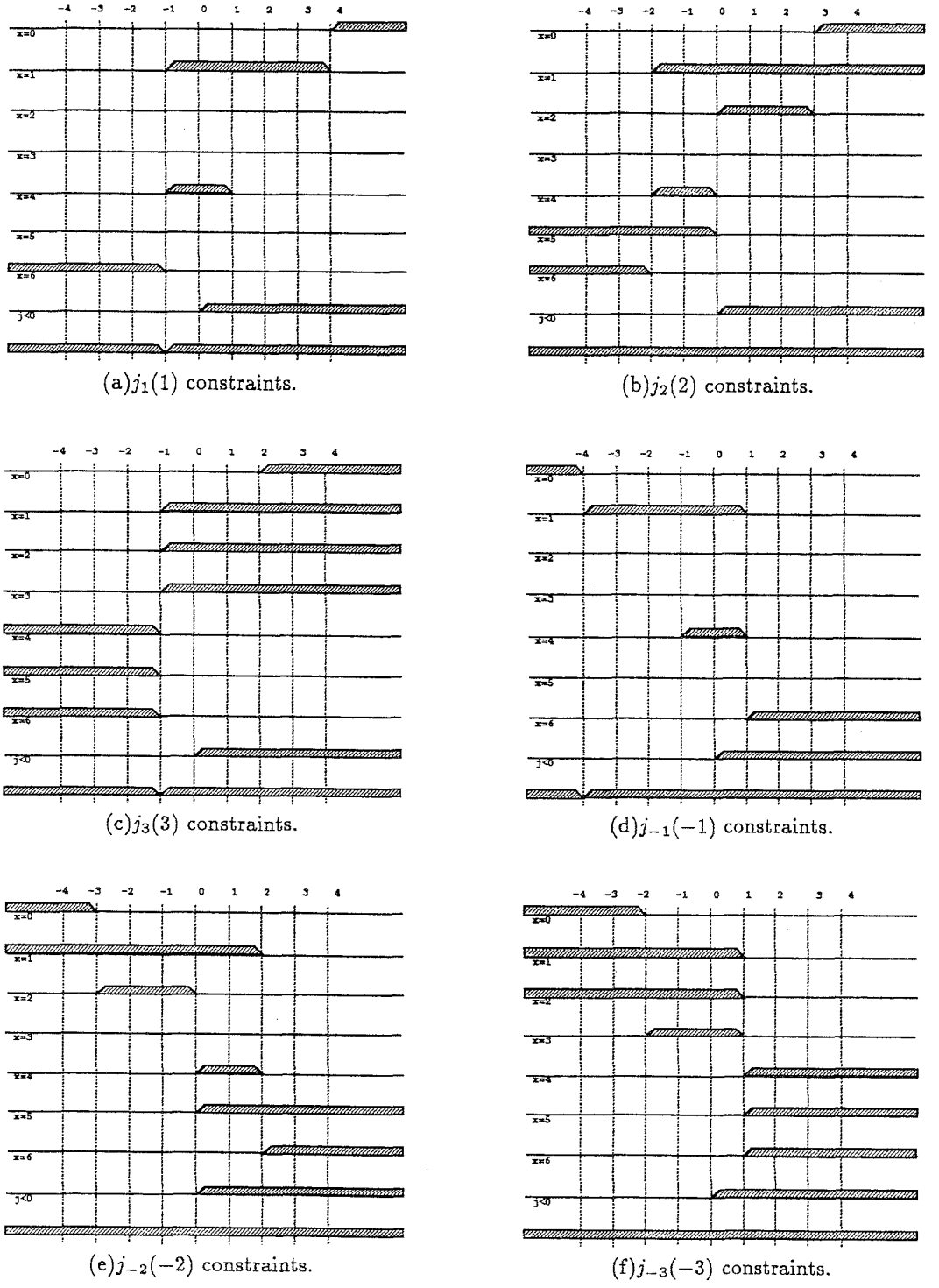
Fig. 5 shows the tree created by the algorithm. The optimal decomposition of s, $s=g \oplus j_1 \oplus j_2$, $g(2)=13$, is shown in Fig. 4.

VIII. CONCLUSION

The gray-scale structuring element decomposition problem can be solved in a similar way to the binary structuring element decomposition problem. In this paper, we showed that the decomposition problem can be solved by simply searching among a finite set of values. The essence of the algorithm is the same as the essence of the binary problem: 1) The domain of the structuring elements participating in the decomposition must have members that are the differences between members of the domain of the given structuring element and 2) that it is necessary for the undecomposed part of the structuring element to be morphologically open with respect to any structuring element participating in its further decomposition. When the decomposition is constrained to two-point decomposition, the search space can be further reduced by utilizing the morphological properties of a two-point decomposition.

APPENDIX A
USING THE CONSTRAINTS: TWO-POINT NOT DECOMPOSABLE EXAMPLE

In this Appendix, we give an example where we use the two-point decomposition constraints developed in Section V-B-2.



(a)$j_1(1)$ constraints. (b)$j_2(2)$ constraints.

(c)$j_3(3)$ constraints. (d)$j_{-1}(-1)$ constraints.

(e)$j_{-2}(-2)$ constraints. (f)$j_{-3}(-3)$ constraints.

Fig. 2. Functional values constraints. The shaded areas correspond to values that grayscale structuring elements j_i in the decomposition can **not** take. In each table, along the rows we choose a point x in the domain, S, of the original structuring, s, and along the columns we give the values j_i, the structuring element in the decomposition can **not** take up. For example, in (a) we see that if we choose, $1 = x \in S$, $j_1(1) \notin \{-1, 0, 1, 2, 3, 4\}$.

In the example problem, the gray-scale structuring element s is not open under the structuring element j for any gray-value assignment for elements in the domain of j. Thus, j cannot be a structuring element in the decomposition of s. We then modify s so that it is open under j for a set of gray-value assignments for the elements in the domain of j and then show that the modified j can be the element in the decomposition of s. In Section V-B-2, we proved that if $s = s \circ j$, then for all $x \in S$,

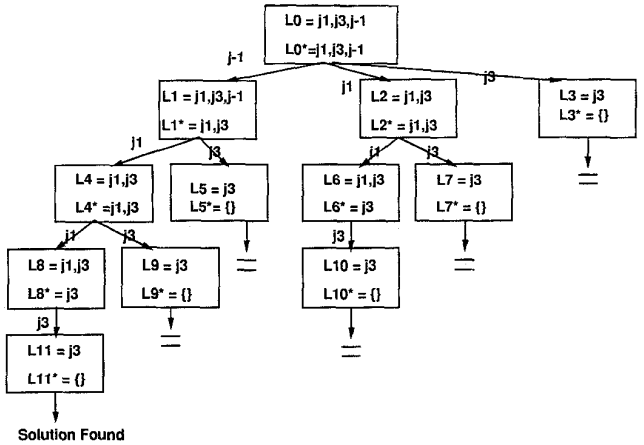


Fig. 3. Tree search for a two-point decomposition of the gray-scale structuring element $s(x)$ shown in Fig. 1. At the root of the tree, there are only three candidate structuring elements: j_{-1}, j_1, and j_3. The optimal solution $g \oplus j_{-1} \oplus j_1 \oplus j_1 \oplus j_3$ is found when a breath-first search produces a residue with a single point (g).

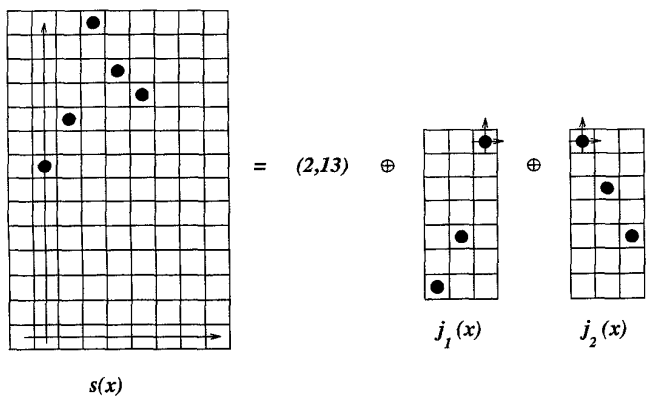


Fig. 4. Three-point decomposition of a gray-scale structuring element. The structuring element s is decomposed as $s = g \oplus j_1 \oplus j_2$, where $g(2) = 13$ is a one-point structuring element representing a translation, and j_i are three-point gray-scale structuring elements each containing the origin (0,0).

at least one of the following is true:

1) $x \in S \ominus J$ and $j(t) \leq s(x+t) - s(x)$; or
2) $x - t \in S \ominus J$ and $j(t) \geq s(x) - s(x-t)$.

Note: If $x \notin S \ominus J$ for a particular x, condition 1 is automatically not true, and similarly, if $x-t \notin S \ominus J$, condition 2 is not true. Furthermore, if for a particular $x \in S$ we find that both the conditions are not met, it must be the case that $s \neq s \circ j$. Let us now consider an example. Let $S = \{-t, 0, t\}$ and $s(-t) = 5$, $s(0) = 3$, and $s(t) = 2$. Let the domain structuring element j be $J = \{0, t\}$ and $j(0) = 0$, $j(t) \leq 0$. We want to find out the range of possible values that $j(t)$ can have when s is open under j, i.e., $s = s \circ j$. Next, we will construct the constraints for $j(t)$. Since $S \ominus J = \{-t, 0\}$ and $x \in S$ can take up three possible values, we have the following three sets of constraints.

Case $x = -t$: Since $x - t \notin S \ominus J = \{-t, 0\}$ for $x = -t$, the second condition is not met. The first condition gives us

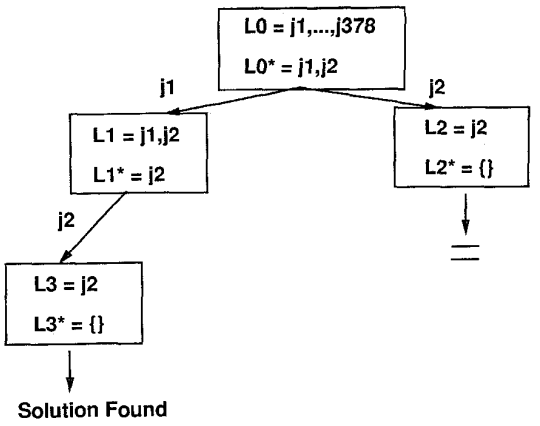


Fig. 5. Tree search for a three-point decomposition of the gray-scale structuring element $s(x)$ shown in Fig. 4. At the root of the tree out of 378 candidate structuring elements, only two j_1 and j_2 survive the forward checking test. The optimal solution $g \oplus j_1 \oplus j_2$ is found when a breath-first search produces a residue with a single point (g).

the constraint $j(t) \leq s(0) - s(-t) = 3 - 5 = -2$. Thus

$$j(t) \leq -2. \tag{13}$$

Case $x = 0$: Since $x, x - t \in S \ominus J = \{-t, 0\}$ for $x = 0$, both the conditions are met. The first condition gives us the constraint $j(t) \leq s(t) - s(0) = 2 - 3 = -1$. The second condition gives us the constraint $j(t) \geq s(0) - s(-t) = 3 - 5 = -2$. Thus, we have

$$j(t) \leq -1 \qquad \text{or} \qquad j(t) \geq -2.$$

Case $x = t$: Since $x \notin S \ominus J = \{-t, 0\}$ for $x = t$, the first condition is not met. The second condition gives us the constraint $j(t) \geq s(t) - s(0) = 2 - 3 = -1$. Thus

$$j(t) \geq -1. \tag{14}$$

We see that constraint (13) contradicts constraint (14). Thus, we conclude that there is no j with domain $J = \{0, t\}$ such that $s = s \circ j$. Now, in the previous example, if we make $s(t) = 1$ and let all other values remain the same, we will get the following constraints:

$$j(t) \leq -2 \tag{15}$$

$$j(t) \leq -2 \text{ or } j(t) \leq -2 \tag{16}$$

$$j(t) \geq -2. \tag{17}$$

The only solution to the constraints is $j(t) = -2$. Thus, $s = s \circ j$ if $j(t) = -2$.

Appendix B
Proof for the Look-Ahead Step

Proposition 19: If at some level l of the decomposition tree $j_1, j_2, \ldots, j_l$ have been determined, then the only $j \in L_l$ that need to be considered for any node in the subtree below are those that satisfy

$$(t_l \ominus j) \oplus j \oplus k_l = s$$

where $t_l = s \ominus k_l$, and $k_l = j_1 \oplus j_2 \oplus \cdots \oplus j_l$.

Proof: Assume that $s = g \oplus j_1 \oplus \cdots \oplus j_M$ with $M > l$ and that we need to find j_{l+1}. The structuring element s can be rewritten as

$$s = (g \oplus j_{l+2} \oplus \cdots \oplus j_M) \oplus k_l \oplus j_{l+1}.$$

Then, by Proposition 14, s must be open under k_l dilated by the function j, which is the candidate to be selected as j_{l+1}:

$$s = (s \ominus (k_l \oplus j)) \oplus (k_l \oplus j) = (t_l \ominus j) \oplus j \oplus k_l.$$

It is also true that this condition holds for the level l considered and for any node in the subtree below: Let $m > l$. Then, for j to be considered as a child of a node at level m, it must satisfy $s = s \circ (k_m \oplus j)$, but $k_m \oplus j = (k_l \oplus j) \oplus (j_{l+1} \oplus j_{l+2} \oplus \cdots \oplus j_M)$.

ACKNOWLEDGMENT

The authors would like to thank Dr. D. C. Benson for his many helpful comments on the implementation of the decomposition algorithm.

REFERENCES

[1] X. Zhuang and R. M. Haralick, "Morphological structuring element decomposition," *Comput. Vision, Graphics, Image Processing*, vol. 35, pp. 370–382, 1986.

[2] T. Kanungo and R. M. Haralick, "Morphological decomposition of restricted domains: A vector space solution," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, Champaign, IL, June 15-18, 1992, pp. 627–629.

[3] ——, "Vector space interpretation for a morphological shape decomposition problem," *J. Math. Imaging Vision*, vol. 2, no. 1, pp. 51–82, Oct. 1992.

[4] X. Zhuang, "Grayscale structuring function decomposition," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn*, Champaign, IL, June 15-18, 1992.

[5] F. Y. Shih and O. R. Mitchell, "Threshold decomposition of gray-scale morphology into binary morphology," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 31–42, 1989.

[6] J. P. Fitch, E. J. Coyle, and N. C. Gallager, "Threshold decomposition of multidimensional ranked order operations," *IEEE Trans. Circuits Systems*, vol. 32, pp. 445–450, 1985.

[7] G. X. Ritter and P. D. Gader, "Image algebra techniques for parallel image processing," *J. Parallel Distributed Comput.*, vol. 4, no. 5, pp. 7–44, 1987.

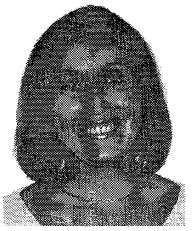
[8] C. H. Richardson and R. W. Schafer, "Lower bound for structuring element decomposition," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 4, pp. 365–369, 1991.

[9] J. Xu, "The decomposition of convex polygonal morphological structuring elements into neighborhood subsets," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 2, pp. 153–162, 1991.

[10] R. Jones and I. Svalbe, "Algorithms for the decomposition of gray-scale morphological operations," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 16, no. 6, pp. 581–588, June 1994.

[11] P. D. Gader, "Separable decompositions and approximations of grayscale morphological templates," *CVGIP: Image Understanding*, vol. 53, no. 3, pp. 288–296, 1991.

[12] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 4, pp. 532–550, July 1987.



Octavia I. Camps (S'89–M'91) received the B.S. degree in computer science and the B.S. degree in electrical engineering from the Universidad de la Republica, Uruguay, in 1981 and 1984, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seatle, USA, in 1987 and 1992, respectively.

From 1986 to 1991, she was a Research Assistant in the Intelligent Systems Laboratory at the University of Washington. In 1992, she joined the faculty at The Pennsylvania State University, University Park, USA, where she is currently an Assistant Professor in the Department of Electrical Engineering and the Department of Computer Science and Engineering and a co-director of the Center for Intelligent Information Processing (CIIP). Her current research interests include object recognition, reverse engineering systems, image processing, and pattern recognition.

Dr. Camps was awarded the SWE Outstanding Female Engineering Student Award in 1988, a GTE Fellowship Award in 1990, and an NSF Research Initiation Award in 1993 for her work on robust 3-D object recognition. She is a member of the IEEE Computer, Robotics and Automation, and Signal Processing Societies, the ASEE, and Tau Beta Pi.

Tapas Kanungo (S'94) was born on May 5, 1964 in Varanasi, India. He received the Bachelors degree in electronics and communication engineering from Regional Engineering College, Tiruchirapalli, India, in 1986. He is currently working toward the Ph.D. degree in electrical engineering at the University of Washington, Seattle, USA, where he received the M.S. degree in electrical engineering in 1990.

He worked at the AT&T Bell Laboratories, Murray Hill, NJ, USA, during the summer of 1994 and at the IBM Almaden Research Center, San Jose, CA, USA, during the summer of 1993. From 1988 onwards, he has been a Research Assistant at the Intelligent Systems Laboratory at the University of Washington. From 1986 to 1988, he was with the Computer Science Group at the Tata Institute of Fundamental Research, Bombay, India. His research interests include reconstruction, document understanding, performance evaluation, image databases, morphology, shape decomposition, and human vision.

In 1992, Mr. Kanungo received the second prize at the Annual Industrial Affiliate's Poster Competition, and in 1990, he was a recipient of the Watamull Scholarship.

Robert M. Haralick (S'62–S'67–M'69–SM'76–F'84) received the B.A. degree in mathematics in 1964, the B.S. degree in electrical engineering in 1966, and the M.S. degree in electrical engineering in 1967, and the Ph.D. degree in 1969, all from the University of Kansas, Lawrence, USA.

He is the Boeing Clairmont Egtvedt Professor in Electrical Engineering at the University of Washington, Seattle, USA. His recent work is in shape analysis and extraction using the techniques of mathematical morphology, robust pose estimation, techniques for making geometric inferences from perspective projection information, propagation of random perturbations through image analysis algorithms, and document analysis. He joined the faculty of the Electrical Engineering Department at the University of Kansas, where he last served as Professor from 1975 to 1978. In 1979, he joined the Electrical Engineering Department at Virginia Polytechnic Institute and State University, Blacksburg, USA, where he was a Professor and Director of the Spatial Data Analysis Laboratory. From 1984 to 1986, he served as Vice President of Research at Machine Vision International, Ann Arbor, MI, USA.

Prof. Haralick is a Fellow of IEEE for his contributions in computer vision and image processing. He is a Fellow of the IAPR for his contributions in image processing, computer vision, and mathematical morphology. He serves on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and *Real Time Imaging*. He is an associate editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and an associate editor for *Journal of Electronic Imaging*.