Chapter 9

Discrete Half-Plane Morphology for Restricted Domains

Tapas Kanungo and Robert M. Haralick

University of Washington
Seattle, Washington

I. INTRODUCTION

Morphological operations, when performed on objects represented as sets of discrete points, are of $O(n^2)$ complexity, where n is the size of each set [6]. But when objects of interest are convex, or can be decomposed into convex objects, a more appropriate representation of the object is in terms of its boundary [1,2].

In this paper we extend our earlier work [7] and give two boundary representations for a class of two-dimensional binary shapes and define all the morphological operations—dilation, erosion, opening, closing, n-fold dilation, and n-fold erosion—in terms of these boundary representations. Further, we prove that each of these algorithms is $O(1)$ and hence independent of the size of the object. In addition, we prove that the results of these algorithms are equivalent to those obtained using the regular set-theoretic definitions. We also suggest how the algorithms can be extended for more complicated objects.

Morphological algorithms using boundary representations have been attempted by Ghosh [3,4] for polytopes in continuous domain. Xu [9] gave algorithms for decomposition of a class of binary convex shapes using boundary representation. There he used the notion of dilation of boundaries but did not prove it equivalent to the set-theoretic definitions. Our work sets the basic foundation by providing all the basic morphological operations in terms of the boundary representations.

This paper is organized as follows. In Section II we set the stage by giving the basic definitions and notation to be used. We introduce the B-code representation in Section III. In Section IV we formally characterize restricted domains in terms of B-codes and half-planes and show how to interconvert the representations. Morphology on restricted domains using B-codes and discrete half-planes is addressed in Section V. Here we start from the set, perform set morphology, and show that the same result is obtained using morphology using the half-plane representations. In Section VI we discuss the algorithms, evaluate their computational complexity, and compare them with the complexity of existing set-theoretic algorithms. Here we also walk through some examples of dilation and erosion of restricted domains. The open problems and work in progress are discussed in Section VII, and in Section VIII we give our overall conclusion.

II. PRELIMINARIES

In this section we define all the necessary terms and give the notation used in this chapter.

Images are represented as mathematical functions over some finite domain. One of its common forms is $f : (x, y) \rightarrow z$ where $x, y, z \in \mathbf{R}$, and $\mathbf{R}$ is the set of real numbers. The ordered pair (x, y) represents the spatial coordinates of a point with respect to some reference frame and z is the luminance value at that point. One way to represent such a function in a computer is by discretizing the space variables and the values the function takes.

The space domain can be discretized by tessellating it. A tessellation can be obtained by dividing the space into a set of nonintersecting domains such that the union of all the domains is the $\mathbf{R}^2$ space. Each of these domains is then represented by a point inside it. The most common technique for tessellating the $\mathbf{R}^2$ space is to make all the domains squares of the same size and represent them by their centers. To make the treatment simple, we will use squares of size 1×1, centered on the points $(i, j) \in \mathbf{Z}^2$, where $\mathbf{Z}$ is the set of integers. An illustration is given Figure 1. Note that the origin (0,0) represents the unit square centered on it; that is, the real axis passes through its center. Thus, the tessellation domains can be uniquely represented by the ordered pair $(i, j) \in \mathbf{Z}^2$. The ordered pairs will be interchangeably called *lattice points*, *points*, or *pixels*.

The function values are defined on the lattice points and can also be discretized. We will not go into the details of discretization of the function values, although the process is similar to that of space discretization. In this chapter we are interested in the case where the function takes binary values of 0 or 1 at the lattice points. Such images are called *discrete binary images* or *binary images*. Since the set of all the lattice points having the value 1 completely characterizes an image, the term binary image will be used to imply the set of all lattice points where the function value is 1. The terms *structuring element* and *shape* will also refer to sets of lattice points with function value 1.

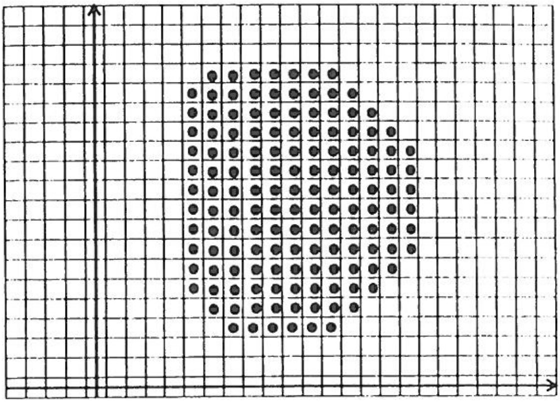

Figure 1. A binary image. Note that the lattice points are represented by the center of each pixel.

Definition 2.1. The **x**[ ] and **y**[ ] operators take a lattice point as an argument and return its x and y ordinates, respectively. Formally, let $p = (i, j) \in \mathbf{Z}^2$. Then $\mathbf{x} : \mathbf{Z}^2 \to \mathbf{Z}$ and $\mathbf{y} : \mathbf{Z}^2 \to \mathbf{Z}$ such that $x[p] = i$ and $y[p] = j$.

Definition 2.2. A point (x, y) is called a *foreground point* if its value is nonzero and a *background point* otherwise.

Definition 2.3. Two foreground points (i, j) and (k, l) are said to be *4-neighbors* if and only if (k, l) is an element of $\{(i + 1, j), (i - 1, j), (i, j), (i, j + 1), (i, j - 1)\}$.

Definition 2.4. Two foreground points (i, j) and (k, l) are said to be *8-neighbors* if and only if (k, l) is an element of $\{(i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1), (i - 1, j - 1), (i + 1, j + 1), (i + 1, j - 1), (i - 1, j + 1)\}$.

Definition 2.5. Two foreground points x_i and x_j are said to be *4-connected* if and only if there is a sequence of foreground points $\{x_1 = x_i, x_2, \ldots, x_{n-1}, x_n = x_j\}$ such that x_k and x_{k+1} are 4-neighbors for all k in $\{1, \ldots, n\}$.

Definition 2.6. Two foreground points x_i and x_j are said to be *8-connected* if and only if there is a sequence of foreground points $\{x_1 = x_i, x_2, \ldots, x_{n-1}, x_n = x_j\}$ such that x_k and x_{k+1} are 8-neighbors for all k in $\{1, \ldots, n\}$.

Definition 2.7. A set of foreground points F is a *4-connected component* if for all $x_i, x_j \in F$, x_i and x_j are 4-connected.

Definition 2.8. A set of foreground points F is an *8-connected component* if for all $x_i, x_j \in F$, x_i and x_j are 8-connected.

Definition 2.9. A foreground point of an 8-connected component is a *boundary* or *edge* point if one or more of its 4-neighbors is a background point.

Definition 2.10. A *discrete half-plane* is a set of lattice points $H \subset \mathbf{Z}^2$ defined as

$$H = \{(i, j) \mid a_0 i + b_0 j \leq c_0 \text{ such that } i, j, a_0, b_0, \text{ and } c_0 \in \mathbf{Z}\}$$

Definition 2.11. A 4- or 8-connected component F is *convex discretely* if and only if all the lattice points lying inside or on the convex hull of F belong to F. This definition directly implies that a discretely convex connected component has no holes.

Next, we restate the definitions of the basic morphological operations based on the tutorial by Haralick, et al. [6].

Dilation is the morphological transformation that combines two sets using vector addition of set elements. If A and B are sets in $\mathbf{Z}^2$, 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 2.12. The *dilation* of A by B is denoted by $A \oplus B$ and is defined by

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

Erosion is the morphological dual of dilation. If A and B are sets in $\mathbf{Z} \times \mathbf{Z}$, then the erosion of A by B is the set of all elements x for which $x + b \in A$ for every $b \in B$.

Definition 2.13. The *erosion* of A by B is denoted by $A \ominus B$ and is defined as

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

Definition 2.14. The *opening* of a set B by a structuring element K is denoted by $B \bigcirc K$ and is defined as

$$B \bigcirc K = (B \ominus K) \oplus K$$

Definition 2.15. The *closing* of a set B by a set K is denoted by $B \bullet K$ and

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

Definition 2.16. The *n-fold dilation* of a set B by a set A is denoted by $B \oplus (\oplus_n A)$ and is defined as

$$B \oplus (\underset{n}{\oplus} A) = B \overbrace{\oplus A \oplus A \oplus \cdots \oplus A}^{n \text{ times}}$$

Definition 2.17. The *n-fold erosion* of a set B by a set A is

$$B \ominus (\underset{n}{\oplus} A) = \overbrace{((\cdots((B \ominus A) \ominus A) \cdots) \ominus A}^{n \text{ times}}$$

III. BOUNDARY CODES

Line drawings have been commonly used to represent the boundaries of two-dimensional objects. In the case of discrete, binary images these line drawings of the object boundary can be represented in either of the following ways: (1) as a sequence of points, (2) by a chain code representation, or (3) as a sequence of line segments. A description of these methods can be found in [1,5].

The chain code representation as proposed by Freeman [2] does not incorporate the lengths of the edges in its notation. It nevertheless has a provision for a special token in the implementation that allows for the length of the edge to be stored. In this section, we discuss a notation for chain codes that requires explicit representation of the boundary edge lengths and directions. This boundary encoding scheme, referred to as B-code, uses a list data structure.

B-code is a representation scheme for connected components in terms of their boundary lattice points. Only a starting boundary point is represented explicitly, while the rest of the boundary points are represented in terms of successive displacements in one of possible eight directions. If the successive displacements happen to be in the same direction, they are encoded as the direction followed by the number of moves in that direction. The formal notation to represent a connected component A is given by

$$A = \langle (i_A, j_A) \mid (\mathbf{d}_l : n_l)(\mathbf{d}_{l+1} : n_{l+1}) \cdots (\mathbf{d}_m : n_m) \rangle \tag{9.1}$$

Here (i_A, j_A) is the starting boundary lattice point, and the ordered pairs following the vertical bar describe each successive displacement. The number of ordered pairs is equal to the number of changes in the direction of displacement. In the ordered pair $(\mathbf{d}_k : n_k)$, $\mathbf{d}_k \in \{ \mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_7\}$ represents the direction of the displacement and the nonnegative integer n_i following the colon represents the number of successive moves in that direction. The directions $\mathbf{d}_0, \ldots, \mathbf{d}_7$ are the same as the chain code directions $0, \ldots, 7$, which correspond to angles of $\{0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°\}$, with respect to the positive x-axis: $\mathbf{d}_0 = (1,0)$, $\mathbf{d}_1 = (1,1)$, $\mathbf{d}_2 = (0,1)$, $\mathbf{d}_3 = (-1,1)$, $\mathbf{d}_4 = (-1,0)$, $\mathbf{d}_5 = (-1,-1)$, $\mathbf{d}_6 = (0,-1)$, and $\mathbf{d}_7 = (1,-1)$.

Figure 2 illustrates the relation between B-codes and chain codes. Figure 2a is a binary image. Figure 2b shows the boundary pixels explicitly represented by the corresponding chain code:

(5, 1) : 000001111222244442222244444456666666666677

Figure 2c shows the pixels explicitly represented by the corresponding B-code: $\langle (5,1) \mid (\mathbf{d}_0 : 5)(\mathbf{d}_1 : 4)(\mathbf{d}_2 : 4)(\mathbf{d}_4 : 4)(\mathbf{d}_2 : 5)(\mathbf{d}_4 : 6)(\mathbf{d}_5 : 5)(\mathbf{d}_4 : 6)(\mathbf{d}_5 : 5)$ $(\mathbf{d}_6 : 10)(\mathbf{d}_7 : 2)\rangle$. It can be seen that the B-code representation can be thought of as a runlength encoding of the chain code. Also, any binary image, simply or multiply connected, that can be encoded using the chain codes can be encoded using the B-codes too.

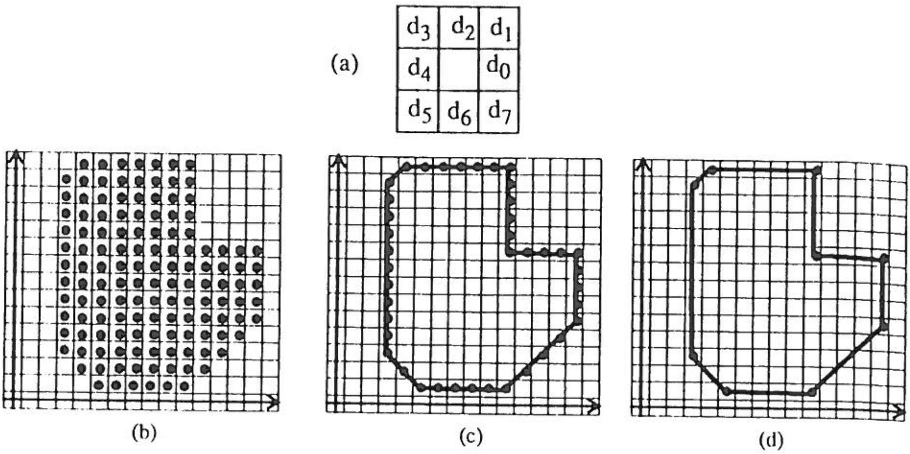

Figure 2. Example of B-coding of images. (a) Basic directions; (b) a binary image; (c) pixels explicitly represented by its chain code (5,1): 00000111122224444222224444445666666666677; (d) pixels explicitly represented by its B-code $\langle (5,1) \mid (\mathbf{d}_0 : 5)(\mathbf{d}_1 : 4)(\mathbf{d}_2 : 4)(\mathbf{d}_4 : 4)(\mathbf{d}_2 : 5)(\mathbf{d}_4 : 6)(\mathbf{d}_5 : 5)(\mathbf{d}_6 : 10)(\mathbf{d}_7 : 2) \rangle$.

IV. RESTRICTED DOMAINS

The class of objects we will decompose and work on will be discretely convex, four-connected sets all of whose boundaries are oriented at angles which are multiples of 45° and whose lengths are multiples of the pixel side lengths for 0° and 90° orientations and multiples of $\sqrt{2}$ times the pixel side length for 45° and 135° orientations. We will refer to the set of all objects belonging to this class as restricted domains.

Definition 4.1. A *restricted domain* is a discretely convex, four-connected shape whose convex hull has sides at angles that are multiples of 45° with respect to the positive x axis.

Some examples of restricted domains are given in Figure 3. In the following sections we will define the restricted domains in terms of their B-codes and present an equivalent representation in terms of half-planes.

A. B-Code Representation

1. Convention

Given a binary image of a restricted domain, A, we will represent it in the B-code form for further processing. The binary image of a restricted domain can be

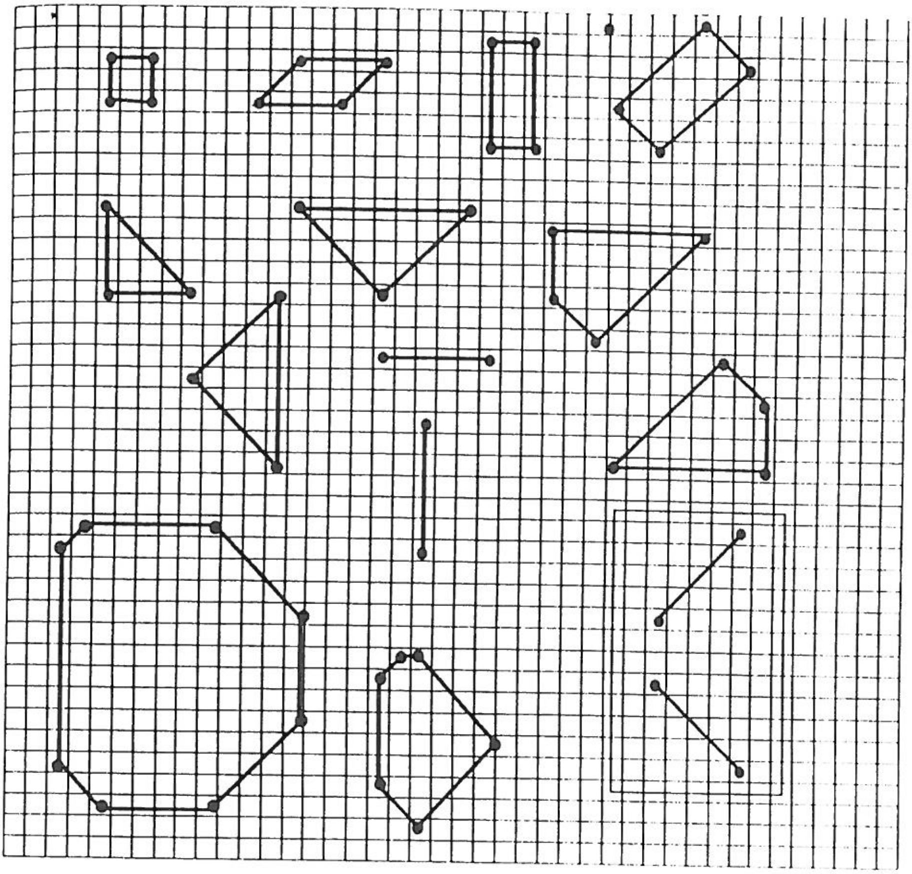

Figure 3. A few examples of restricted domains. Note that the diagonal lines in the box are not strictly restricted domains since they are not 4-connected.

represented in many ways using a B-code representation; i.e., the B-code representation is not unique. This is due to the fact that the only restriction on the starting point of a B-code representation is that it should be a vertex. Thus, there are as many B-code representations of a restricted domain as the number of vertices it has. To avoid ambiguity we will use the following convention:

The starting point will always be the lowest and leftmost vertex of the restricted domain. The rest of the vertices are encoded by traversing around the restricted domain along its boundary points in the counterclockwise direction, encoding the length of the edges that constitute A. The interior points of the set will be those on the left of the direction of motion.

The B-code obtained using this convention represents an equivalence class of B-codes—the class of all B-codes representing the considered restricted domain. Each B-code in the equivalence class is a rotated version of the other but representing the same set of lattice points nevertheless.

2. Properties of B-Coded Restricted Domains

In this section we present some useful properties of B-coded restricted domains that will be used in later proofs.

Property 4.1. Any restricted domain can be represented by a general B-code of the form $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)\ (\mathbf{d}_1 : n_1)\ (\mathbf{d}_2 : n_2)\ (\mathbf{d}_3 : n_3)\ (\mathbf{d}_4 : n_4)\ (\mathbf{d}_5 : n_5)\ (\mathbf{d}_6 : n_6)\ (\mathbf{d}_7 : n_7)\rangle$ by giving appropriate values to the n_i's. Thus in this representation there are always eight vertices, eight displacements, and the displacement angles are monotonically increasing from $\mathbf{d}_0$ to $\mathbf{d}_7$. If there is no displacement corresponding to one of the directions, the corresponding pair can be dropped from the B-code and the particular n_i is given a value zero. Note that in this case two vertices become coincident.

Given a closed contour, the net displacement on traversing its complete boundary is zero. Since the B-code of a restricted domain $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)\ (\mathbf{d}_1 : n_1) \cdots (\mathbf{d}_7 : n_7)\rangle$ represents a closed contour, it inherits the following two properties of a closed contour.

Property 4.2. The sum of displacements contributing to the positive x direction is equal to the sum of displacements contributing to the negative x direction:

$$n_0 + n_1 + n_7 = n_3 + n_4 + n_5 \tag{9.2}$$

Property 4.3. The sum of displacements contributing to the positive y direction is equal to the sum of displacements contributing to the negative y direction:

$$n_1 + n_2 + n_3 = n_5 + n_6 + n_7 \tag{9.3}$$

Property 4.4. Any B-code of the form $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)\ (\mathbf{d}_1 : n_1)\ (\mathbf{d}_2 : n_2)\ (\mathbf{d}_3 : n_3)\ (\mathbf{d}_4 : n_4)\ (\mathbf{d}_5 : n_5)\ (\mathbf{d}_6 : n_6)\ (\mathbf{d}_7 : n_7)\rangle$ whose n_i's satisfy the properties in Eqs. (9.2) and (9.3) is either a restricted domain or a line at 45° or 135°. The lines are special cases and are of the form $A = \langle (i, j) \mid (\mathbf{d}_1 : n_1)\ (\mathbf{d}_5 : n_5)\rangle$ and $A = \langle (i, j \mid \mathbf{d}_3 : n_3)\ (\mathbf{d}_7 : n_7)\rangle$. Details on this are given in the Appendix.

Given a B-code of a restricted domain $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)\ (\mathbf{d}_1 : n_1) \cdots (\mathbf{d}_7 : n_7)\rangle$, all the eight vertices of the polygon are uniquely defined and can be found in the following two ways.

Property 4.5. Let the vertex v_0 be the starting lattice point (i, j). The rest of the vertices are given recursively. Given the kth vertex v_k, the x and y coordinates of the $(k + 1)$th vertex v_{k+1} are given by the recursive equations

$$\mathbf{x}[v_{k+1}] = \mathbf{x}[v_k] + n_k\mathbf{x}[d_k] \tag{9.4}$$

$$\mathbf{y}[v_{k+1}] = \mathbf{y}[v_k] + n_k\mathbf{y}[d_k] \tag{9.5}$$

for $0 \leq k \leq 6$. Here $\mathbf{x}[v_0] = i$ and $\mathbf{y}[v_0] = j$—the x and y coordinates of the starting point of the B-code.

The coordinates of the vertices of A can also be computed relative to the starting point of the restricted domain.

Property 4.6. The coordinates of the kth vertex v_k can be computed in terms of the starting location (i,j), and the lengths n_l, $0 \leq l \leq k$. Let $\mathbf{V}_x$, $\mathbf{V}_y$, $\mathbf{V}$, and $\mathbf{N}$ be the matrices

$$\mathbf{V}_x = \begin{bmatrix} \mathbf{x}[v_0] \\ \mathbf{x}[v_1] \\ \vdots \\ \mathbf{x}[v_7] \end{bmatrix}, \quad \mathbf{V}_y = \begin{bmatrix} \mathbf{y}[v_0] \\ \mathbf{y}[v_1] \\ \vdots \\ \mathbf{y}[v_7] \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_x \\ \mathbf{V}_y \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} n_0 \\ n_1 \\ \vdots \\ n_7 \end{bmatrix} \tag{9.6}$$

Then,

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_x \\ \mathbf{V}_y \end{bmatrix} = \mathbf{P} \begin{bmatrix} i \\ j \\ \mathbf{N} \\ i \\ j \\ \mathbf{N} \end{bmatrix} \tag{9.7}$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix} \tag{9.8}$$

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & -1 & -1 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & -1 & -1 & -1 & 0 & 0 \end{bmatrix} \tag{9.9}$$

$$\mathbf{P}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & -1 & -1 & 0 \end{bmatrix} \tag{9.10}$$

B. Normalized Half-Plane Representation

Restricted domains can be represented in terms of the intersections of discrete half-planes. Let $A = \langle (i,j) \mid (\mathbf{d}_0 : n_0)(\mathbf{d}_1 : n_1)\cdots(\mathbf{d}_7 : n_7)\rangle$ be a restricted domain. Then the lattice points belonging to A can be defined in terms of intersections of eight discrete half-planes $\mathcal{H}_i$, $0 \leq i \leq 7$. Each of these half-planes $\mathcal{H}_i$ is a function of the basic directions of the displacement d_i and the vertices v_i of the restricted domain. Each discrete half-plane $\mathcal{H}_i$ is such that its boundary passes through the vertex v_i and its edge is along the direction d_i. The half-plane $\mathcal{H}_i$ represents all the points on the left and on the boundary while traversing in the direction d_i along the boundary. Therefore, a restricted domain $A = \langle (i,j) \mid (\mathbf{d}_0 : n_0)(\mathbf{d}_1 : n_1)\cdots(\mathbf{d}_7 : n_7)\rangle$ can be represented as

$$A = \mathcal{H}_0 \cap \mathcal{H}_1 \cap \cdots \cap \mathcal{H}_7 \tag{9.11}$$

where $\mathcal{H}_i$ is a discrete half-plane given by

$$\mathcal{H}_i = \left\{ p = (x,y) \in \mathbf{Z}^2 \text{ such that } \begin{vmatrix} x - \mathbf{x}[v_i] & y - \mathbf{y}[v_i] \\ \mathbf{x}[v_i] + \mathbf{x}[d_i] & \mathbf{y}[v_i] + \mathbf{y}[d_i] \end{vmatrix} \leq 0 \right\} \tag{9.12}$$

Figure 4 illustrates the half-plane concept. We can expand the above expression for the particular cases of $\mathcal{H}_i$, $0 \leq i \leq 7$. Substituting the expression for the vertex v_i of the restricted domain given in Eqs. (9.7) into the inequality (9.12), the inequalities for the half-planes $\mathcal{H}_0$ to $\mathcal{H}_7$ thus obtained are:

$$\begin{aligned} \mathcal{H}_0&: \quad (0)x + (-1)y \leq c_0 \\ \mathcal{H}_1&: \quad (1)x + (-1)y \leq c_1 \\ \mathcal{H}_2&: \quad (1)x + (0)y \leq c_2 \\ \mathcal{H}_3&: \quad (1)x + (1)y \leq c_3 \\ \mathcal{H}_4&: \quad (0)x + (1)y \leq c_4 \end{aligned} \tag{9.13}$$

$$\mathcal{H}_5: \quad (-1)x + (1)y \le c_5$$

$$\mathcal{H}_6: \quad (-1)x + (0)y \le c_6$$

$$\mathcal{H}_7: \quad (-1)x + (-1)y \le c_7$$

where x, y, $c_i \in \mathbf{Z}$ and the c_i are given by the equations

$$\mathbf{C} = \mathbf{L}\begin{bmatrix} i \\ j \\ \mathbf{N} \end{bmatrix} \tag{9.14}$$

where

$$\mathbf{C} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} \tag{9.15}$$

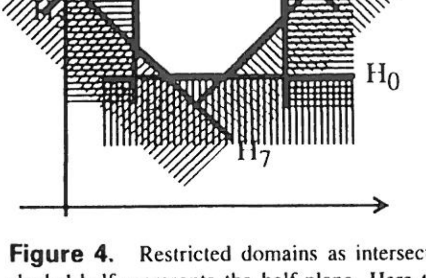

Figure 4. Restricted domains as intersections of half-planes. $\mathcal{H}_1 \ldots \mathcal{H}_7$. The unshaded half represents the half-plane. Here the intersection set is the unshaded central region.

$$\mathbf{L} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 & 1 & 1 & 1 & 0 & 0 \\ -1 & -1 & -1 & -2 & -1 & 0 & 1 & 2 & 1 & 0 \end{bmatrix} \tag{9.16}$$

To make the information more compact, we will use matrices to represent the system of linear inequalities in (9.13) as

$$\mathbf{M}p' \leq \mathbf{C} \tag{9.17}$$

where

$$\mathbf{M} = \begin{bmatrix} 0 & -1 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ -1 & 1 \\ -1 & 0 \\ -1 & -1 \end{bmatrix} \tag{9.18}$$

and $p = (x, y)$ is a lattice point. Note that the inequalities (9.17) are considered row-wise.

The physical interpretation of the system of inequalities (9.17) is as follows. Consider eight half-planes passing through the origin, each one corresponding to a direction d_i, $0 \leq i \leq 7$. The half-planes are translated from the origin up, down, left, and right such that they pass through the corresponding vertices v_i. The intersections of these half-planes gives us the lattice points belonging to the restricted domain.

Notice that since the d_i's are fixed, the slope of the discrete half-planes are also fixed and hence the half plane $\mathcal{H}_i$ is uniquely represented by the corresponding c_i's. But a set of c_i's representing a restricted domain need not be unique. For example, in Figure 5, we see that the half-planes corresponding to two different sets of c_i's represent the same intersection set. This is because the half-plane $\mathcal{H}_1$ is *redundant* and can be translated to infinitely many locations without having any effect on the intersection set. All the possible sets of c_i's representing a given

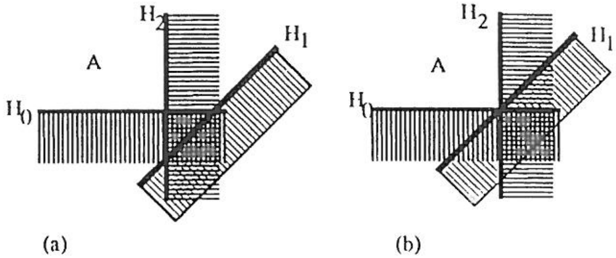

Figure 5. Unnormalized and normalized half-planes. The unshaded half represents the half-plane. The half-plane $\mathcal{H}_1$ in (a) is redundant and can be moved until it passes through the vertex of set A; this situation is shown in (b).

restricted domain form an equivalence class. This raises the question about a convention that we can follow such that an equivalence class of restricted domains can be represented through a unique c_i set. We notice that the c_i's that are obtained from the B-code representation using Eq. (9.14) always represent discrete half-planes passing through the vertices of the restricted domains. Those that are redundant—that is, those that correspond to a displacement of length zero along the d_i direction—also pass through a vertex even though they have potentially infinite possibilities. Thus, we will follow the convention that if a set of c_i's represents a restricted domain, it should be normalized such that all the half-planes pass through the vertices of the intersection set. Such a set of eight c_i's, represented using a vector $\mathbf{C}$ will be called the *normalized half-plane representation* of the restricted domain. The half-planes that are not redundant and form the sides of the polygon will be called *primary*.

Before we proceed further, we need to address the following issues:

1. Under what conditions the set of c_i's represents a nonempty set
2. Under what conditions the restricted domain represented by the set of c_i's is a normalized representation and, if it is not, how to normalize it

The c_i's represent a set of discrete half-planes. Hence, the set of points belonging to the intersection of these half-planes is not empty if and only if the set of points belonging to the intersection of any two of these half-planes is not empty. Figure 6a illustrates an example where the half-plane $\mathcal{H}_0$ is unnormalized. Since it should be moved such that it touches the intersection set, it is obvious that it should be moved to r, the intersection point of $\mathcal{H}_1$ and $\mathcal{H}_7$. The other possibilities could have been p, the intersection point of $\mathcal{H}_7$ and $\mathcal{H}_2$, or q, the intersection point of $\mathcal{H}_1$ and $\mathcal{H}_6$. Notice that p and q do not belong to the intersection set and they are below r. Figures 6b and c show examples where $\mathcal{H}_0$ has to be moved to p and q, respectively. Notice that in this case p is above q and r. And in Figure

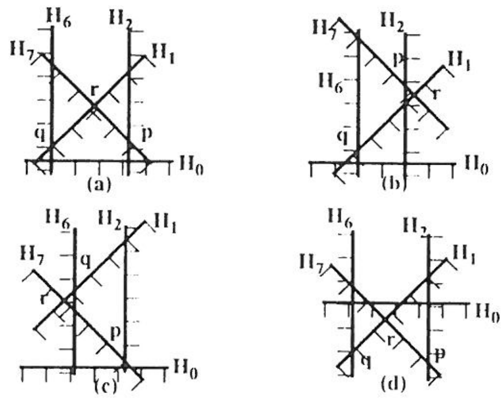

Figure 6. The normalization of $\mathcal{H}_0$—four cases. In (a), (b), and (c) the half-plane $\mathcal{H}_0$ is redunant and has to be normalized, that is, moved up so that it passes through r, p, and q, respectively. In (d) $\mathcal{H}_0$ is primary and cannot be moved.

6d, $\mathcal{H}_0$ is a primary half-plane and cannot be moved. In this case $\mathcal{H}_0$ is above p, q, and r. Thus the algorithm for normalization of $\mathcal{H}_0$ then becomes: find the intersection points p, q, and r, and update c_0 such that $\mathcal{H}_0$ passes through the one belonging to the set. In the case that $\mathcal{H}_0$ is primary, nothing should be done to c_0. Conveniently, the c_0 found in this way also forms a bound for the half-plane $\mathcal{H}_3$, that is, the half plane $\mathcal{H}_3$ cannot be below this level. In case it is, the intersection of the half-planes results in an empty set. Using the same argument for all other half-planes, it can be shown that a set of eight c_i's represents a nonempty set if and only if

$$\mathbf{C} \geq \mathbf{C}_{\text{bound}} = \max[\mathbf{G}_1\mathbf{C}, \mathbf{G}_2\mathbf{C}, \mathbf{G}_3\mathbf{C}, -\lfloor \mathbf{G}_4\mathbf{C} \rfloor] \tag{9.19}$$

and a set of c_i's is normalized if

$$\mathbf{C} \geq \max[\mathbf{G}_1\mathbf{G}_2\mathbf{C}, \mathbf{G}_1\mathbf{G}_3\mathbf{C}, -\lfloor \mathbf{G}_1\mathbf{G}_4\mathbf{C} \rfloor] \tag{9.20}$$

where the 8×8 matrices $\mathbf{G}_1, \ldots, \mathbf{G}_4$ used in the algorithm are given below.

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9.21}$$

$$\mathbf{G}_2 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ -2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -2 & 0 & 0 & 0 \end{bmatrix} \tag{9.22}$$

$$\mathbf{G}_3 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ -2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{9.23}$$

$$\mathbf{G}_y = \begin{bmatrix} 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 1 & 0 & 0 & 0 & 0- & 0 & 1 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{9.24}$$

The lower ceilings come about because the 45° and 135° lines need not intersect at a lattice point. Notice that $\mathbf{G}_1^2 = \mathbf{I}$. Here the matrix multiplications find the intersection points. The max operation selects the one nearest to the set. Thus, both the issues mentioned above have been addressed.

Notice that when $c_1 = -c_5$ or $c_3 = -c_7$, we have diagonal lines at 45° or 135°, respectively. These are not strictly restricted domains since they are not 4-connected (but they are 8-connected). Thus since restricted domains are 4-connected, the following constraints should hold:

$$c_1 > -c_5 \qquad \text{and} \qquad c_3 > -c_7 \tag{9.25}$$

The algorithm *Normalize* given in Table 1 takes as input the **C** array of a restricted domain and returns the normalized **C** array if one exists, else it returns a NULL value. Since the algorithm has five multiplications of 8×8 matrices with 8×1 vectors, one lower ceiling of an 8×1 vector, one 8×1 vector comparison one row-wise max operation of four 8×1 vectors, and no loops, the algorithm is constant in time.

C. Conversion from Normalized Half-Plane to B-Code

Given the c_i's of a normalized restricted domain, we should be able to (1) find the vertices of a restricted domain in terms of the c_i's, (2) find the n_i's in terms of the c_i's, and (3) find the B-code representation of the restricted domain.

The vertices of the restricted domain can be computed by finding the intersections of the consecutive half-planes. They can be expressed in terms of the vector **C** as follows:

$$\mathbf{V} = \mathbf{D}\begin{bmatrix}\mathbf{C}\\ \mathbf{C}\end{bmatrix} \tag{9.26}$$

where

$$\mathbf{D} = \begin{bmatrix}\mathbf{D}_1 & 0\\ 0 & \mathbf{D}_2\end{bmatrix} \tag{9.27}$$

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1\\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \tag{9.28}$$

$$\mathbf{D}_2 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{9.29}$$

Table 1. The Algorithm for Normalizing Half-Planes

```
function Normalize(C) : ArrayObject

Input:
ArrayObject C;

begin
    C_bound := max[G_1C, G_2C, G_3C, −⌊G_4C⌋];
    if (C < C_bound)
    then
        return NULL;
    else
        C := max[C, G_1G_2C, G_1G_3C, −⌊G_1G_4C⌋];
        return C;
end Normalize;
```

Then n_i's can be computed by finding the distance between the two consecutive vertices v_{i+1} and v_i. Thus,

$$\mathbf{N} = \mathbf{QC} \tag{9.30}$$

where

$$\mathbf{Q} = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{9.31}$$

The B-code representation of the restricted domain is determined by v_0 and **N**.

V. BOUNDARY CODE MORPHOLOGY FOR RESTRICTED DOMAINS

In this section we will give constant time algorithms for dilation, erosion, opening, closing, n-fold dilation, and n-fold erosion of restricted domains using their half-plane and B-code representations. We will show that the results obtained using these algorithms are equivalent to those obtained using regular morphology. If the input restricted domains are in their B-code representations or if the output restricted domains are needed in their B-code representation, the results of the previous section can be used for the interconversion between representations.

A. Dilation of Restricted Domains

Let A and B be two restricted domains given by the B-codes

$$A = \langle (i_A, j_A) \mid \mathbf{d}_0 : n_0^A)(\mathbf{d}_1 : n_1^A)\cdots(\mathbf{d}_7 : n_7^A)\rangle \tag{9.32}$$

$$B = \langle (i_B, j_B) \mid (\mathbf{d}_0 : n_0^B)(\mathbf{d}_1 : n_1^B)\cdots(\mathbf{d}_7 : n_7^B)\rangle \tag{9.33}$$

and their normalized half-plane representations be

$$A = \{a \in \mathbf{Z}^2 \mid \mathbf{M}a' \leq \mathbf{C}^A\} \tag{9.34}$$

$$B = \{b \in \mathbf{Z}^2 \mid \mathbf{M}b' \leq \mathbf{C}^B\} \tag{9.35}$$

where $\mathbf{C}^A$ and $\mathbf{C}^B$ are given by

$$\mathbf{C}^A = \mathbf{L}\begin{bmatrix} i^A \\ j^A \\ \mathbf{N}^A \end{bmatrix} \tag{9.36}$$

$$\mathbf{C}^B = \mathbf{L}\begin{bmatrix} i^B \\ j^B \\ \mathbf{N}^B \end{bmatrix} \tag{9.37}$$

$\mathbf{N}^A$ and $\mathbf{N}^B$ are 8×1 column vectors with the respective edge lengths as their elements, and $\mathbf{M}$ and $\mathbf{L}$ matrices are defined in Eqs. (9.18), and (9.16), respectively.

Lemma 5.1. The set C given by

$$C = \{c \in \mathbf{Z}^2 \mid \mathbf{M}c' \leq \mathbf{C}^C\} \tag{9.38}$$

where $\mathbf{C}^C = \mathbf{C}^A + \mathbf{C}^B$, is a restricted domain, and the vector $\mathbf{C}^C$ is a normalized half-plane representation of C.

Proof. From the discussion in Section IV.B and Eq. (9.19), the sufficient condition for $\mathbf{C}^C$ to be a restricted domain is that $\mathbf{C}^C \geq \mathbf{C}^C_{\text{bound}}$. Since A and B are restricted domains and $\mathbf{C}^A$ and $\mathbf{C}^B$ are normalized half-plane representations,

$$\mathbf{C}^A \geq \mathbf{C}^A_{\text{bound}} = \max[\mathbf{G}_1\mathbf{C}^A, \mathbf{G}_2\mathbf{C}^A, \mathbf{G}_3\mathbf{C}^A, -\lfloor \mathbf{G}_4\mathbf{C}^A \rfloor] \tag{9.39}$$

$$\mathbf{C}^B \geq \mathbf{C}^B_{\text{bound}} = \max[\mathbf{G}_1\mathbf{C}^B, \mathbf{G}_2\mathbf{C}^B, \mathbf{G}_3\mathbf{C}^B, -\lfloor \mathbf{G}_4\mathbf{C}^B \rfloor] \tag{9.40}$$

Thus, adding the above equations we get

$$\begin{aligned} \mathbf{C}^C &= \mathbf{C}^A + \mathbf{C}^B \\ &\geq \max[\mathbf{G}_1\mathbf{C}^A, \mathbf{G}_2\mathbf{C}^A, \mathbf{G}_3\mathbf{C}^A, -\lfloor \mathbf{G}_4\mathbf{C}^A \rfloor] \\ &\quad + \max[\mathbf{G}_1\mathbf{C}^B, \mathbf{G}_2\mathbf{C}^B, \mathbf{G}_3\mathbf{C}^B, -\lfloor \mathbf{G}_4\mathbf{C}^B \rfloor] \end{aligned} \tag{9.41}$$

But we know that $\max[a,b] + \max[c,d] \geq \max[(a + c), (b + d)]$. Hence,

$$\mathbf{C}^C \geq \max[\mathbf{G}_1(\mathbf{C}^A + \mathbf{C}^B), \mathbf{G}_2(\mathbf{C}^A + \mathbf{C}^B), \mathbf{G}_3(\mathbf{C}^A + \mathbf{C}^B), -\lfloor \mathbf{G}_4(\mathbf{C}^A + \mathbf{C}^B) \rfloor] \tag{9.42}$$

Therefore, $\mathbf{C}^C$ represents a restricted domain. Similarly, we now show that $\mathbf{C}^C$ is a normalized half-plane representation. Since $\mathbf{C}^A$ and $\mathbf{C}^B$ are normalized representations, we have

$$\mathbf{C}^A \geq \max[\mathbf{G}_1\mathbf{G}_2\mathbf{C}^A, \mathbf{G}_1\mathbf{G}_3\mathbf{C}^A, -\lfloor \mathbf{G}_1\mathbf{G}_4\mathbf{C}^A \rfloor] \text{ and} \tag{9.43}$$

$$\mathbf{C}^B \geq \max[\mathbf{G}_1\mathbf{G}_2\mathbf{C}^B, \mathbf{G}_1\mathbf{G}_3\mathbf{C}^B, -\lfloor \mathbf{G}_1\mathbf{G}_4\mathbf{C}^B \rfloor] \tag{9.44}$$

As before, from the above equations we get

$$\mathbf{C}^A + \mathbf{C}^B \geq \max[\mathbf{G}_1\mathbf{G}_2(\mathbf{C}^A + \mathbf{C}^B), \mathbf{G}_1\mathbf{G}_3(\mathbf{C}^A + \mathbf{C}^B), -\lfloor \mathbf{G}_1\mathbf{G}_4(\mathbf{C}^A + \mathbf{C}^B) \rfloor] \tag{9.45}$$

Thus, $\mathbf{C}^C$ is normalized. Furthermore, since A and B are 4-connected restricted domains, we have from Eq. (9.25): $c_1^A > -c_5^A$, $c_3^A > -c_7^A$, $c_1^B > -c_5^B$, and $c_3^B > -c_7^B$. Manipulating, we get $c_1^A + c_1^B > c_5^A + c_5^B$ and $c_3^A + c_3^B > c_7^A + c_7^B$. Thus C is four-connected and $\mathbf{C}^C$ is a normalized half-plane representation of a restricted domain. Note that even if either A or B is a diagonal line, e.g., $c_1^A = -c_5^A$ or $c_1^B = -c_5^B$ for the case of 45° diagonal lines, the resultant shape is still a restricted domain (because the 4-connectivity constraints are still satisfied).

Lemma 5.2. The eight vertices $\mathbf{V}^C$ of C are the vector sums of the respective vertices $\mathbf{V}^A$ and $\mathbf{V}^B$ of A and B.

Proof. The vertices of C are given by

$$\mathbf{V}^C = \mathbf{D}\begin{bmatrix}\mathbf{C}^C \\ \mathbf{C}^C\end{bmatrix} \tag{9.46}$$

where $\mathbf{D}$ is the matrix given in Eq. (9.27). Since $\mathbf{C}^C = \mathbf{C}^A + \mathbf{C}^B$, we have

$$\mathbf{V}^C = \mathbf{D}\begin{bmatrix}\mathbf{C}^A + \mathbf{C}^B \\ \mathbf{C}^A + \mathbf{C}^B\end{bmatrix} = \mathbf{D}\begin{bmatrix}\mathbf{C}^A \\ \mathbf{C}^A\end{bmatrix} + \mathbf{D}\begin{bmatrix}\mathbf{C}^B \\ \mathbf{C}^B\end{bmatrix} \tag{9.47}$$

Hence,

$$\mathbf{V}^C = \mathbf{V}^A + \mathbf{V}^B \tag{9.48}$$

$$v_i^C = v_i^A + v_i^B, \qquad 0 \leq i \leq 7 \tag{9.49}$$

Thus the lemma is proved.

The dilation of two restricted domains can be performed by adding their respective $\mathbf{C}$ vectors.

Proposition 5.1. $A \oplus B$ is given by the restricted domain C whose normalized half plane representation is given by $\mathbf{C}^C = \mathbf{C}^A + \mathbf{C}^B$.

Proof. We will proceed by proving that (i) $A \oplus B \subseteq C$ and then (ii) $C \subseteq A \oplus B$.

(i) $A \oplus B \subseteq C$. Let $c \in A \oplus B$. Then, by definition of dilation, there exist $a \in A$ and $b \in B$ such that $c = a + b$. Since $a \in A$ and $b \in B$, from Eqs. (9.34) and (9.35) we have

$$\mathbf{M}a' \leq \mathbf{C}^A$$
$$\mathbf{M}b' \leq \mathbf{C}^B$$

Adding the above equations we get

$$\mathbf{M}a' + \mathbf{M}b' \leq \mathbf{C}^A + \mathbf{C}^B$$
$$\mathbf{M}(a + b)' \leq \mathbf{C}^A + \mathbf{C}^B$$

Hence $c \in C$. Therefore, $A \oplus B \subseteq C$.

(ii) $C \subseteq A \oplus B$. Let $c \in C$. We have to prove that there exist $a \in A$ and $b \in B$ such that $a + b = c$. Since $c \in C$, it satisfies the relation

$$\mathbf{M}c' \leq \mathbf{C}^A + \mathbf{C}^B \tag{9.50}$$

From Lemma 5.1 we know that C is a restricted domain and thus by definition it is discretely convex. Hence, c belongs to the convex hull of C, and it can be expressed as the convex combination of the vertices of C. Thus,

$$c = \sum_{0 \leq i \leq 7} \alpha_i v_i^C \tag{9.51}$$

where $\alpha_i \in \mathbf{R}$, $0 \leq \alpha_i \leq 1$, and $\sum \alpha_i = 1$. From Lemma 5.2 we get

$$c = \sum_{0 \leq i \leq 7} \alpha_i (v_i^A + v_i^B) \tag{9.52}$$

$$c = c_A + c_B \tag{9.53}$$

where $c_A = \sum_{0 \leq i \leq 7} \alpha_i v_i^A$ and $c_B = \sum_{0 \leq j \leq 7} \alpha_i v_i^B$. The first term, c_A, on the right-hand side of the above equation belongs to the convex hull of the restricted domain A, the second term, c_B, to the convex hull of the restricted domain B. Notice that Eq. (9.52) does not guarantee that c_A and c_B are lattice points; i.e., they need not belong to $\mathbf{Z}^2$. It just represents the fact the vector sum of two points c_A and c_B in $\mathbf{R}^2$ is the lattice point c in $\mathbf{Z}^2$. We will now show that we can always find a lattice point belonging to A, in the neighborhood of c_A, and another to B, in the neighborhood of c_B, such that their vector sum is the lattice point c. This is illustrated in Figure 7.

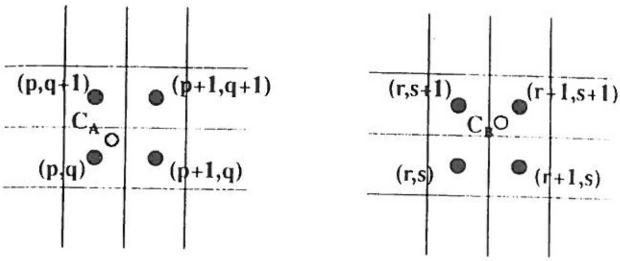

Figure 7. Relation between p, q, r, s, δ, and γ. We have to find one neighboring lattice point of c_A and another of c_B such that their vector sum is $c = (p + r + 1, q + s + 1)$.

Let $c = (l, m)$, $c_A = (p + \delta, q + \gamma)$, and $c_B = (r + 1 - \delta, s + 1 - \gamma)$ such that $l, m, p, q, r, s \in \mathbf{Z}$ and $0 \leq \delta, \gamma \leq 1$. Going back to Eq. (9.53) and replacing c, c^A, and c^B by their values, we get

$$\begin{aligned} c = (l,m) &= (p + \delta, q + \gamma) + (r + 1 - \delta, s + 1 - \gamma) \\ &= (p + r + 1, q + s + 1) \end{aligned} \tag{9.54}$$

It can be seen from the above equations that the point c_A lies between the four lattice points (p, q), $(p + 1, q)$, $(p, q + 1)$, and $(p + 1, q + 1)$. Similarly, the point c_B lies between the four lattice points (r, s), $(r + 1, s)$, $(r, s + 1)$, and $(r + 1, s + 1)$. We will prove that the vector sum of two of these eight points, one belonging to A and another belonging to B, is the lattice point $(l, m) = (p + r + 1, q + s + 1)$.

We can find out which of the four surrounding lattice points necessarily belong to the restricted domain A given c_A (and hence δ and γ). Depending on the values of δ and γ, the area between the lattice points surrounding c_A and c_B can be divided into several regions. The inclusion of a particular neighbor in the set A depends on where the point c_A falls. The attack has to be on a case-by-case basis.

Case (i)—c_A lies in the region defined by $\gamma > \delta$. We can see that in this case the neighboring lattice point $(p, q + 1)$ necessarily belongs to A. This is because if $(p, q + 1)$ did not belong to A, no convex combination of any subset of the other three neighboring lattice points could produce a c_A in the region defined by $\gamma > \delta$. From symmetry we can see that for $\gamma > \delta$, $(r + 1, s)$ necessarily belongs to the set B. Thus the desired lattice points are $(p, q + 1)$ and $(r + 1, s)$ since their vector sum is the lattice point $(p + 1, q + 1) = (l, m) = c$.

Similarly, for the cases (ii) $\gamma < \delta$, (iii) $\gamma > 1 - \delta$, and (iv) $\gamma < 1 - \delta$, we can find lattice points belonging to A and B such that their vector sum is the lattice point $(l, m) = c$.

The only region inside the square not yet considered is when $\delta = \gamma = 0.5$, which is the center of the square. When $\delta = \gamma = 0.5$, we fall into neither of the above categories and hence we have to treat this case separately. We notice that the lattice point $(p+0.5, q+0.5)$ can result from the convex combination if (i) all four neighboring lattice points belong to the set, (ii) only three of the neighboring lattice points belong to the set, and (iii) any two diagonally opposite lattice points belong to the set. We can eliminate the third case since it implies that A is a diagonal line and hence is not four connected and thereby contradicting our assumptions. Thus for the case when $\delta = \gamma = 0.5$, either three or four of the lattice points neighboring c_A necessarily belong to A. The same is true for the set B. Note that the lattice points belonging to A do not in any way constrain the ones belonging to B. It is easy to verify that given any three lattice points surrounding C_A and three lattice points surrounding c_B, we can always find two lattice points, one neighboring c_A and one neighboring c_B, such that their sum is $c = (l, m) = (p + r + 1, q + s + 1)$. In fact there are many such pairs.

Thus we have proved that $C \subset A \oplus B$. Hence $C = A \oplus B$.

We will now prove an important lemma that says that a dilation of A by B is just the addition of the respective side lengths and the starting points.

Lemma 5.3. If $C = A \oplus B$, then $(i_C, j_C) = (i_A, j_A) + (i_B, j_B)$ and $\mathbf{N}^C = \mathbf{N}^A + \mathbf{N}^B$.

Proof. Since $\mathbf{C}^C = \mathbf{C}^A + \mathbf{C}^B$, we have $\mathbf{N}^C = \mathbf{Q}\mathbf{C}^C = \mathbf{Q}(\mathbf{C}^A + \mathbf{C}^B) = \mathbf{N}^A + \mathbf{N}^B$. The rest of the lemma follows from the fact that $\mathbf{V}^C = \mathbf{V}^A + \mathbf{V}^B$.

B. Erosion of Restricted Domains

Let A and B be two restricted domains with normalized half-plane representations,

$$A = \{a \in \mathbf{Z}^2 \mid \mathbf{M}a' \le \mathbf{C}^A\} \tag{9.55}$$

$$B = \{b \in \mathbf{Z}^2 \mid \mathbf{M}b' \le \mathbf{C}^B\} \tag{9.56}$$

where $\mathbf{C}^A$ and $\mathbf{C}^B$ are given by

$$\mathbf{C}^A = \mathbf{L}\begin{bmatrix} i^A \\ j^A \\ \mathbf{N}^A \end{bmatrix} \tag{9.57}$$

$$\mathbf{C}^B = \mathbf{L}\begin{bmatrix} i^B \\ j^B \\ \mathbf{N}^B \end{bmatrix} \tag{9.58}$$

$\mathbf{N}^A$ and $\mathbf{N}^B$ are 8×1 column vectors with the respective edge lengths as their elements, $\mathbf{M}$ and $\mathbf{L}$ are the matrices defined in Eqs. (9.18) and (9.16), respectively.

The erosion of A by B can be performed by subtracting the $\mathbf{C}$ matrix of B from that of A. The resulting $\mathbf{C}$ matrix need not be a normalized half-plane representation—it has to be normalized using the algorithm given in the previous section. Furthermore, the erosion of a restricted domain with another need not produce a restricted domain. Consider, for example, the erosion of a rectangle by a rhombus where the sides of the rectangle and the rhombus are oriented at 45° and 135° and the side lengths of the rhombus equal the smaller of the two sides of the rectangle. It can easily be seen that the result of the erosion is a line oriented along the longer side of the rectangle, i.e., a line at 45° or 135°. Since lines at 45° and 135° are not 4-connected (but are 8-connected), they are not restricted domains. These special cases have to be considered separately.

Proposition 5.2. $A \ominus B$ is given by C, whose half-plane representation is given by

$$C = \{c \in \mathbf{Z}^2 \mid \mathbf{M}c' \leq \mathbf{C}^C\}$$

where $\mathbf{C}^C = \mathbf{C}^A - \mathbf{C}^B$. C can be either a restricted domain or a diagonal line.

Proof. We will proceed by proving that (i) $A \ominus B \subseteq C$ and then (ii) $C \subseteq A \ominus B$.

(i) $A \ominus B \subseteq C$: Let $c \in A \ominus B$. By definition of erosion, $c + b \in A$ for all $b \in B$. Therefore,

$$\mathbf{M}(c + b)' \leq \mathbf{C}^A \qquad \text{for all } b \in B \tag{9.59}$$

We will prove it by contradiction. Suppose that $c \notin C$. Thus,

$$\mathbf{M}c' \nleq \mathbf{C}^C \tag{9.60}$$

That is, there exists at least one inequality in the system of inequalities (9.60) that does not hold. Without loss of generality, let i, $0 \leq i \leq 7$, be the number of the inequality that is not satisfied:

$$\mathbf{e}_i\mathbf{M}c' > \mathbf{e}_i\mathbf{C}^C \tag{9.61}$$

where $\mathbf{e}_i$ is a 1×8 row vector with 1 in the ith column and zeros elsewhere.

Since the vector $\mathbf{C}^B$ is a normalized half-plane representation of the restricted domain B, the ith vertex of B, v_i^B, lies on $\mathcal{H}_i^B$. Thus,

$$\mathbf{e}_i\mathbf{M}(v_i^B)' = \mathbf{e}_i\mathbf{C}^B \tag{9.62}$$

The system of inequalities (9.59) holds for every $b \in B$, and in particular it holds for the vertex v_i^B:

$$\mathbf{M}(c + v_i^B)' = \mathbf{M}c' + \mathbf{M}(v_i^B)' \leq \mathbf{C}^A \tag{9.63}$$

Substituting Eq. (9.62) in the ith row of the system of inequalities (9.63), we get

$$\mathbf{e}_i\mathbf{M}c' + \mathbf{e}_i\mathbf{C}^B \leq \mathbf{e}_i\mathbf{C}^A \tag{9.64}$$

Since $\mathbf{C}^C = \mathbf{C}^A - \mathbf{C}^B$, we have

$$\mathbf{e}_i \mathbf{M}c' \leq \mathbf{e}_i \mathbf{C}^C \tag{9.65}$$

contradicting Eq. (9.61). Thus, $c \in C$, and $A \ominus B \subseteq C$.

(ii) $C \subseteq A \ominus B$: Let $c \in C$. We need to prove that for all $b \in B$, $c + b \in A$. Since $c \in C$, c satisfies the set of inequalities:

$$\mathbf{M}c' \leq \mathbf{C}^A - \mathbf{C}^B$$

Adding $\mathbf{C}^B$ to both sides, we have

$$\mathbf{M}c' + \mathbf{C}^B \leq \mathbf{C}^A$$

But, $\mathbf{M}b' \leq \mathbf{C}^B$ for all $b \in B$. Thus,

$$\mathbf{M}c' + \mathbf{M}b' \leq \mathbf{M}c' + \mathbf{C}^B \leq \mathbf{C}^A$$

Taking $\mathbf{M}$ as a common factor,

$$\mathbf{M}(c + b)' \leq \mathbf{C}^A \qquad \text{for all } b \in B$$

Therefore, $c + b \in A$ for all $b \in B$, and $C \subseteq A \ominus B$. C is a diagonal line when either $c_1^C = -c_5^C$ or $c_3^C = -c_7^C$.

C. Opening

Morphological opening of a binary set A by another binary set B is denoted by $A \bigcirc B$ and is defined as

$$A \bigcirc B = (A \ominus B) \oplus B \tag{9.66}$$

Since dilations and erosions of restricted domains have been defined, the above definition of opening is also valid for restricted domains. The definition is also valid for the following more general cases when either A or B or both are not restricted domains: (i) $A \ominus B$ is a restricted domain and B is a line at 45°; (ii) $A \ominus B$ is a restricted domain and B is a line at 135°; (iii) $A \ominus B$ and B are lines at 135°; (iv) $A \ominus B$ and B are lines at 45°. Note that lines at 45° and 135° are not restricted domains since they are not 4-connected. The algorithm cannot be used if $A \ominus B$ and B are lines at 45° and 135°, respectively. This constraint is due to the fact that the dilation of a 45° line with a 135° line results in a rhombus-like shape with one-pixel holes; i.e., the shape is not filled. Thus the set theory dilation results in a shape that not filled but the half-plane and B-code dilation algorithms produce a shape that is filled.

D. Closing

Morphological closing of a binary set A by another binary set B is denoted by $A \bullet B$ and is defined as

$$A \bullet B = (A \oplus B) \ominus B \tag{9.67}$$

Since dilations and erosions of restricted domains have been defined, the above definition of opening is also valid for restricted domains. The definition is also valid for the following more general cases when either A or B or both are not restricted domains: (i) A is a line at 45° and B is a restricted domain; (ii) B is a line at 45° and A is a restricted domain; (iii) A and B are lines at 45°; (iv) A and B are lines at 135°. Note that lines at 45° and 135° are not restricted domains since they are not 4-connected. The algorithm cannot be used if A and B are lines at 45° and 135°, respectively. This constraint is due to the fact that the set theory dilation of a 45° line with a 135° line results in a rhombus-like shape with one-pixel holes; i.e., the shape is not filled. Thus the set theory dilation results in a shape that not filled but the half-plane and B-code dilation algorithms produce a shape that is filled.

VI. ALGORITHMS AND THEIR COMPLEXITY

In this section we give the algorithms for computing the dilation and erosion of restricted domains represented by half-planes. The algorithms for opening and closing can be easily obtained by applying the dilation and erosion algorithms in the appropriate order. The algorithms for n-fold dilation and n-fold erosion need one multiplication step, which we explain at the end of this section.

The following data structures are used in the algorithms:

ArrayObject is a data structure containing an array and its dimensions. In the algorithms the vectors associated with B-codes, half-planes, etc. are stored using this data structure type.

RDObject is a data structure used to represent restricted domains. It contains the three matrices **N**, **V**, and **C** associated with the restricted domain.

The procedure *DilateRDObject* takes as input two RDObject and outputs RDObject which is the dilation of the two input RDObject. The algorithm is given in Table 2.

The procedure *ErodeRDObject* takes as input two RDObject and outputs RDObject which is the erosion of the two input RDObject. The algorithm is given in Table 3. This procedure calls the normalization function which is given in Table 1. The function *Normalize* takes as input an ArrayObject containing the **C** array of a restricted domain and returns the normalized **C** array if one exists; else it returns a NULL value.

The n-fold dilation of a restricted domain B by a restricted domain A is $B \oplus (\oplus_n A)$; the dilation of B by the n-fold dilation of A. In the B-code domain it amounts to multiplying the side lengths of A by n and adding it to the starting point of B. If A and B have the side lengths given by the vectors $\mathbf{N}^A$ and $\mathbf{N}^B$, and

Table 2. The Algorithm for Dilation of Restricted Domains

```
procedure DilateRDObject (A, B, C)

Input:
RDObject A, B;
Output:
RDObject C;

begin
    N^C := N^A + N^B;
    (i_C, j_C) := (i_A, j_A) + (i_B, j_B);

end DilateRDObjects;
```

Table 3. The Algorithm for Erosion of Restricted Domains

```
procedure ErodeRDObject (A, B, C)

Input:
RDObject A, B;
Output:
RDObject C;

begin
    C^C := C^A - C^B;
    C^C := Normalize(C^C);
    N^C := QC^C;
    V^C := D [C^C; C^C];
end ErodeRDObjects;
```

starting points (i_A, j_A) and (i_B, j_B), then $(\oplus_n A)$ has side lengths given by the vector $n\mathbf{N}^A$ and starting point $n(i_A, j_A)$. It follows that $B \oplus (\oplus_n A)$ has side lengths given by the vector $\mathbf{N} = \mathbf{N}^B + n\mathbf{N}^A$ and the starting point $(i_B, j_B) + n(i_A, j_A)$. Dilation can also be performed by going into the discrete half-plane representation and adding the $\mathbf{C}$ vectors associated with A and B. Thus the $\mathbf{C}$ vector associated with $B \oplus (\oplus_n A)$ is $\mathbf{C} = \mathbf{C}^B + n\mathbf{C}^A$.

The n-fold erosion of a restricted domain B by a restricted domain A is $B \ominus (\oplus_n A)$, the erosion of B by the n-fold dilation of A. Let $\mathbf{C}^A$ and $\mathbf{C}^B$ be the vectors associated with A and B. Then $n\mathbf{C}^A$ is the vector associated with $(\oplus_n A)$. Thus in the half-plane domain the n-fold erosion of B by A amounts to $\mathbf{C}^B - n\mathbf{C}^A$.

The algorithm for the dilation of restricted domains given in Table 2 consists of 10 additions only. Hence it is a constant-time algorithm. Note that the time complexity is independent of the size of the structuring element. In conventional morphology this is not the case—the time complexity is $O(n^2)$, where n is the number of elements in each set.

The algorithm for an n-fold dilation of restricted domains consists of eight multiplications. Hence it is also a constant-time algorithm.

The erosion algorithm given in Table 3 consists of eight subtractions followed by the process of normalization. The normalization algorithm in Table 1 was shown to be constant in time. Thus, the erosion algorithm is a constant-time algorithm.

The n-fold erosion is represented in terms of n-fold dilation, and since the n-fold dilation algorithm is constant in time, the n-fold erosion algorithm is constant in time.

The algorithm for opening consists of two stages—an erosion stage followed by a dilation stage. Since erosion and dilation algorithms are constant in time, the algorithm for opening is also constant in time. Similarly, the algorithm for closing consists of two stages—a dilation stage followed by a erosion stage. Since erosion and dilation both are constant in time, the algorithm for closing is also constant in time.

A. Walkthrough

In this section we apply the algorithms on some typical restricted domains. Figure 8 shows an example of a dilation. Here $C = A \oplus B$ where

$$A = \langle (0,2) \mid (\mathbf{d}_1 : 3)(\mathbf{d}_3 : 2)(\mathbf{d}_4 : 1)(\mathbf{d}_6 : 5) \rangle$$
$$B = \langle (1,1) \mid (\mathbf{d}_1 : 1)(\mathbf{d}_3 : 1)(\mathbf{d}_5 : 1)(\mathbf{d}_7 : 1) \rangle$$

Thus,

$$(i_A, j_A) = (0,2) \quad \text{and} \quad \mathbf{N}^A = [0\ 3\ 0\ 2\ 1\ 0\ 5\ 0]'$$
$$(i_B, j_B) = (1,1) \quad \text{and} \quad \mathbf{N}^B = [0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]'$$
$$\mathbf{N}^C = \mathbf{N}^A + \mathbf{N}^B = [0\ 4\ 0\ 3\ 1\ 1\ 5\ 1]'$$
$$(i_C, j_C) = (i_A, j_A) + (i_B, j_B) = (1,3)$$

Therefore,

$$C = \langle (1,3) \mid (\mathbf{d}_1 : 4)(\mathbf{d}_3 : 3)(\mathbf{d}_4 : 1)(\mathbf{d}_5 : 1)(\mathbf{d}_6 : 5)(\mathbf{d}_7 : 1) \rangle$$

Thus we started from the B-code representations of A and B. Then we added the respective side lengths and starting locations to get the B-code of C.

Figure 9 shows an erosion when the shape is not open under the structuring element. Here $C = A \ominus B$ where

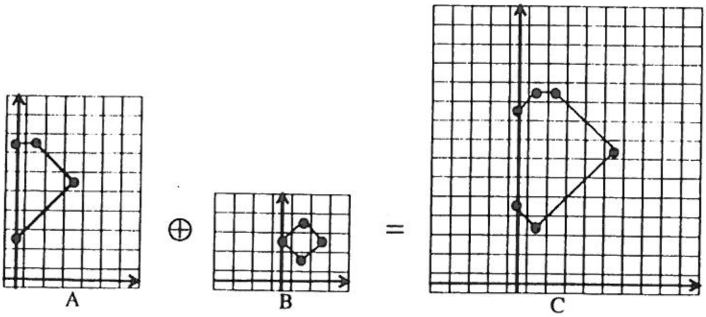

Figure 8. An example of dilation. C is obtained by dilating A by B. Here $A = \langle (0,2) \mid (\mathbf{d}_1 : 3)(\mathbf{d}_3 : 2)(\mathbf{d}_4 : 1)(\mathbf{d}_6 : 5)\rangle$, $B = \langle (1,1) \mid (\mathbf{d}_1 : 1)(\mathbf{d}_3 : 1)(\mathbf{d}_5 : 1)(\mathbf{d}_7 : 1)\rangle$, and $C = \langle (1,3) \mid (\mathbf{d}_1 : 4)(\mathbf{d}_3 : 3)(\mathbf{d}_4 : 1)(\mathbf{d}_5 : 1)(\mathbf{d}_6 : 5)(\mathbf{d}_7 : 1)\rangle$.

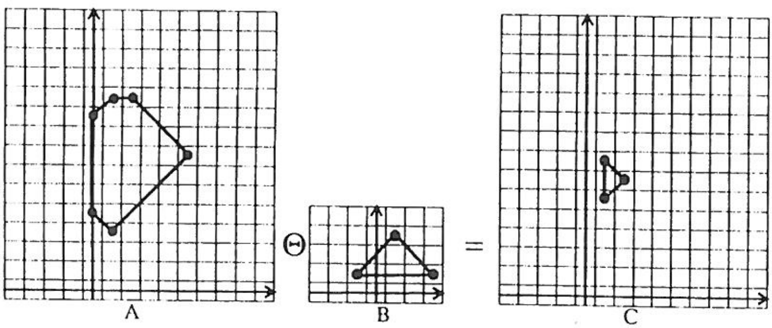

Figure 9. An example of erosion. C is obtained by eroding A by B. Here $A = \langle (1,3) \mid (\mathbf{d}_1 : 4)(\mathbf{d}_3 : 3)(\mathbf{d}_4 : 1)(\mathbf{d}_5 : 1)(\mathbf{d}_6 : 5)(\mathbf{d}_7 : 1)\rangle$, $B = \langle (-1,1) \mid (\mathbf{d}_0 : 4)(\mathbf{d}_3 : 2)(\mathbf{d}_5 : 2)\rangle$, and $C = \langle (1,5) \mid (\mathbf{d}_1 : 1)(\mathbf{d}_3 : 1)(\mathbf{d}_6 : 2)\rangle$.

$$A = \langle (1,3) \mid (\mathbf{d}_1 : 4)(\mathbf{d}_3 : 3)(\mathbf{d}_4 : 1)(\mathbf{d}_5 : 1)(\mathbf{d}_6 : {}_5)(\mathbf{d}_7 : 1)\rangle$$
$$B = \langle (-1,1) \mid (\mathbf{d}_0 : 4)(\mathbf{d}_3 : 2)(\mathbf{d}_5 : 2)\rangle$$

Thus,

$$(i_A, j_A) = (1,3) \quad \text{and} \quad \mathbf{N}^A = [0\ 4\ 0\ 3\ 1\ 1\ 5\ 1]'$$
$$(i_B, j_B) = (-1,1) \quad \text{and} \quad \mathbf{N}^B = [4\ 0\ 0\ 2\ 0\ 2\ 0\ 0]'$$
$$\mathbf{C}^A = \mathbf{L}[i_A\ j_A\ N^A]' = [-3\ \ -2\ 5\ 12\ 10\ 9\ 0\ \ -4]'$$
$$\mathbf{C}^B = \mathbf{L}[i_B\ j_B\ N^B]' = [-1\ 2\ 3\ 4\ 3\ 2\ 1\ 0]'$$

$$\mathbf{C}^C = \mathbf{C}^A - \mathbf{C}^B = [-2\ -4\ 2\ 8\ 7\ 7\ -1\ -4]'$$
$$\mathbf{C}^C = \text{Normalize}(\mathbf{C}^C) = [-5\ -4\ 2\ 8\ 7\ 6\ -1\ -6]'$$
$$\mathbf{N}^C = \mathbf{Q}\mathbf{C}^C = [0\ 1\ 0\ 1\ 0\ 0\ 2\ 0]'$$
$$\mathbf{V}^C = \mathbf{D}[\mathbf{C}^C\ \mathbf{C}^C]' = [\ 1\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 5\ 5\ 6\ 6\ 7\ 7\ 7\ 5]'$$
$$(i_C, j_C) = (\mathbf{V}^C[0], \mathbf{V}^C[8]) = (1,5)$$

Thus,

$$C = \langle (1,5) \mid (\mathbf{d}_1 : 1)(\mathbf{d}_3 : 1)(\mathbf{d}_6 : 2)\rangle$$

Thus we proceeded differently from the way we did for dilation. First we constructed the normalized half-plane representations of A and B and performed the erosion in this representation. Next we normalized the result. We then converted the result back into the B-code form.

VII. WORK IN PROGRESS

Many extensions of the work presented here are being tried out. Here we list a few of them.

Basic set theory operations with B-codes are an immediate task. Questions that come up are: is the B-coded shape A a subset of B? A superset? Or, does A intersect B or is the intersection empty? Is there an algorithm for finding these sets?

The algorithms presented in this chapter can be generalized for the case of any discrete, convex figure. In that case the polygon edges can be at any angle. These angles can be defined in terms of the basic angles that can be formed by a vector starting from the origin and ending on any pixel (m,n) such that m and n are coprime. The problem of holes in the dilation of shapes needs to be addressed.

Further, the algorithms have been extended to the case of continuous convex polyhedra. But B-code data structure cannot be used for representing the polyhedra. In fact, the polyhedra are represented as the intersection of n-dimensional continuous polyhedra.

The problem of decomposing nonconvex shapes, two- or three-dimensional, is difficult. One way to attack this problem is to represent the nonconvex shape as a union of restricted domains and then decompose each of the restricted domains of the union. Another approach is to represent a shape A as a union of disjoint sets A^1 and K^1 where K^1 is the largest restricted domain that is a subset of A and $A = A^1 - K^1$. This process can be repeated and the shape A can be represented as $A = K^1 \cup K^2 \cup \cdots \cup K^n$, where each restricted domain K^i can then be decomposed further.

Morphological dilation on nonconvex shapes will have to be carried out by first representing the shape as a union of restricted domains. Morphological erosion of nonconvex shapes can be done by representing the shapes as intersections of restricted domains and complements of restricted domains. How to decompose a nonconvex shape as a union of restricted domains and intersection of restricted domains is a problem. An algorithm for doing this has to be developed. Furthermore, in higher dimensions the representation scheme of nonconvex shapes has to be in terms of half-planes.

VIII. CONCLUSION

We defined restricted domains—a restricted class of two-dimensional shapes. Two boundary schemes for representing a restricted domains, the B-code and the discrete half-planes representation, were introduced. Morphological dilation, erosion, n-fold dilation, n-fold erosion, openings, and closings of restricted domains with structuring elements that are also restricted domains were expressed in terms of B-codes and half-planes. Algorithms for performing these operations were provided and were proved to be constant-time algorithms.

Suggestions have been made as to how the algorithms can be generalized to any arbitrary two- and three-dimensional convex shapes. Further work needs to be done in the direction of set operations on restricted domains. Finally, it is important to solve the difficult problem of decomposing nonconvex shapes into restricted domains so that the algorithms presented in this chapter can be used on general images.

REFERENCES

1. Ballard, D. H., and Brown, C. M., *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
2. Freeman, H., Computer processing of line-drawing images, *Comput. Surv.*, *6*(1), 57–97 (1974).
3. Ghosh, P. K., A mathematical model for shape description using Minkowski operations, *Comput. Vision Graphics Image Process.*, *44*, 239–269 (1988).
4. Ghosh, P. K., A solution of polygon containment, spatial planning, and other related problems using Minkowski operations, *Comput. Vision Graphics Image Process.*, *49*, 1–35 (1990).
5. Haralick, R. M., and Shapiro, L. G., *Machine Vision*, Addison-Wesley, Reading, Massachusetts, 1992.
6. Haralick, R. M., Sternberg, S. R., and Zhuang, X., Image analysis using mathematical morphology, *IEEE Trans. Pattern Anal. Machine Intell.*, *PAMI-9*(4), 532–550 (1987).
7. Kanungo, T., Haralick, R. M., and Zhuang, X., B-code dilation and decomposition of restricted convex shapes, in *Proceedings of SPIE*, vol. 1350: *Image Algebra and Morphological Image Processing*, San Diego, 1990.
8. Preparata, F. P., and Shamos, M. I., *Computational Geometry*, Springer-Verlag, New York, 1985.
9. Xu, J., The optimal implementation of morphological operations on neighbourhood connected array processors, in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 166–171, 1989.

APPENDIX: B-CODE DETAILS

For a B-code to be the representation of a restricted domain, the sequence of displacements in the directions d_i should be ordered monotonically increasing on the directions between d_0 and d_7. The maximum number of nonzero displacements along the boundary can be eight and the minimum zero, which corresponds to a singular point—the starting point. Thus the B-codes of restricted domains are of the forms $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)(\mathbf{d}_1 : n_1) \cdots (\mathbf{d}_7 : n_7)\rangle$. Note that two exceptions, corresponding to diagonal lines at 45° and 135° are of similar form: $A = \langle (i, j)\ \mathbf{d}_1 : n_1)(\mathbf{d}_5 : n_5)\rangle$ and $A = \langle (i, j) \mid (\mathbf{d}_3 : n_3)(\mathbf{d}_7 : n_7)\rangle$. These are not restricted domains since they are not 4-connected. To prove this property, we need the following two lemmas:

Lemma A.1. Let $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)(\mathbf{d}_1 : n_1) \cdots (\mathbf{d}_7 : n_7)$ be a B-code representing a shape with more than one point. Then there exists i, $0 \le i \le 3$, such that $n_i \ne 0$.

Proof. By contradiction: Let $n_0 = n_1 = n_2 = n_3 = 0$. Since $n_i \ge 0$ for $0 \le i \le 7$, from Eq. (9.3):

$$n_5 = n_6 = n_7 = 0$$

and from Eq. (9.2):

$$n_4 = 0$$

Hence, $n_i = 0$ for $0 \le i \le 7$, and the B-code A reduces to just one point—the starting point (i, j). This contradicts the hypothesis. Therefore, at least one of n_i with $0 \le i \le 3$ must be nonzero.

Lemma A.2. Let $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)(\mathbf{d}_1 : n_1) \cdots (\mathbf{d}_7 : n_7)\rangle$ be a B-code representing a shape with more than one point. Then, if $n_i \ne 0$, $0 \le i \le 7$, there exists j, $\mathrm{mod}(i + 1, 8) \le j \le \mathrm{mod}(i + 4, 8)$, such that $n_j \ne 0$.

Proof. The lemma can be proved using Eqs. (9.2), (9.3) and the fact that some $n_i \ge 0$ for $0 \le i \le 7$.

Case (1): $i = 0$. We have to prove that if $n_0 \ne 0$, then there exists j, $1 \le j \le 4$ such that $n_j \ne 0$. From Eq. (9.2)

$$n_3 + n_4 + n_5 > 0$$

and therefore at least one of n_3, n_4, or n_5 is nonzero.

1. $n_3 \ne 0$. Then $j = 3$.
2. $n_4 \ne 0$. Then $j = 4$.
3. $n_3 = n_4 = 0$, and $n_5 \ne 0$. From Eq. (9.3) we have

$$n_1 + n_2 > 0$$

and therefore at least one of n_1 or n_2 is nonzero.

(a) $n_1 \neq 0$. Then $j = 1$.
(b) $n_2 \neq 0$. Then $j = 2$.

Case (2): $i = 1$. We have to prove that if $n_1 \neq 0$, then there exists j, $2 \leq j \leq 5$ such that $n_j \neq 0$. From Eq. (9.2) we have

$$n_3 + n_4 + n_5 > 0$$

and therefore at least one of n_3, n_4, or n_5 is nonzero.

1. $n_3 \neq 0$. Then $j = 3$.
2. $n_4 \neq 0$. Then $j = 4$.
3. $n_5 \neq 0$. Then $j = 5$.

Similarly we can prove that the lemma holds for the cases when $i = 2, 3, 4, 5, 6, 7$.

Theorem A.1. A B-code of the form $A = \langle (i, j) \mid (\mathbf{d}_0 : n_0)(\mathbf{d}_1 : n_1)\cdots(\mathbf{d}_7 : n_7)\rangle$ is a restricted domain or a line at 45° or 135°.

Proof. We have to show that the B-code represents a discretely convex shape. This we will do by finding the interior angles at each of the vertices and showing that the are concave (less than 180°). In the case when it is exactly 180° it becomes a line but is still discretely convex. Only the lines at 45° and 135° are 8-connected. All the rest of the lines and restricted domains are 4-connected. Details of this method can be found in the book by Preparata and Shamos [8].

Since A is a B-code, either none, two, or more than two of the lengths are nonzero. Hence, we will consider the following three cases:

Case (1): All lengths are zero. Since $n_i = 0$, for all i, $0 \leq i \leq 7$, A reduces to the point (i, j). Thus, A is a restricted domain.

Case (2): Two lengths are nonzero. Let i and j, $0 \leq i < j \leq 7$, such that $n_i \neq 0$ and $n_j \neq 0$. From Lemma A.2 we see that that $j = i + 4$, and A represents a line. Thus A is discretely convex.

Case (3): Three or more lengths are nonzero. We will proceed by first ordering the vertices of A in an increasing order of counterclockwise angular displacement around the starting point. Then we take three consecutive vertices at a time and check if the angle between them is concave. If all such consecutive triples have angles less than 180°, the binary shape represented by the B-code is discretely convex. Let $v_0, v_1, \ldots, v_7, v_8 = v_0$ be the eight vertices of A. If n_i is zero, vertices v_i and v_{i+1} will coincide. Only the distinct vertices of the shape will contribute to the edges of the convex hull. Let $v_{k+1}, v_{l+1}, v_{m+1}$, with $0 \leq k + 1 < l + 1 < m + 1 \leq 7$ be three distinct consecutive vertices of A. A detailed discussion of the technique can be found in [8].

The vertices $v_{k+1}, v_{l+1}, v_{m+1}$ form a concave angle if

$$\begin{vmatrix} \cos\theta_l & \sin\theta_l & 0 \\ \cos\theta_m & \sin\theta_m & 0 \\ 0 & 0 & 1 \end{vmatrix} = \sin(\theta_m - \theta_l) \geq 0 \tag{9.68}$$

where θ_i is the angle formed by the direction d_i with the positive x-axis. Since $\theta_i = i\pi/4$, Eq. (9.68) reduces to

$$\sin(\theta_m - \theta_l) = \sin(m - l)\frac{\pi}{4} \tag{9.69}$$

Consider the following two cases:

1. $0 \leq k + 1 < l + 1 \leq 4$. Since the vertices $v_{k+1}, v_{l+1}, v_{m+1}$ are distinct and consecutive we have

   $n_k \neq 0$

   $n_l \neq 0$

   $n_m \neq 0$

   Since $l \leq 3$, from Lemma A.2, there exists i, $l + 1 \leq i \leq l + 4$, such that $n_i \neq 0$. Since v_{m+1} is consecutive to v_{l+1}, we have

   $$l + 1 \leq m \leq i \leq l + 4 \tag{9.70}$$

   Therefore,

   $$1 \leq m - l \leq 4 \tag{9.71}$$

   and $\sin(\theta_m - \theta_l) \geq 0$.
2. $l \geq 4$. Since v_{l+1} and v_{m+1} are consecutive, we have

   $$l + 1 \leq m \leq 8 \tag{9.72}$$

   Since $l \geq 4$, we have $m - l \leq 8 - 4 = 4$,

   $$1 \leq m - l \leq 4 \tag{9.73}$$

   and $\sin(\theta_m - \theta_l) \geq 0$.

Hence, A is a discretely convex shape. Since the edges of A are in the directions d_i, they are oriented at angles that are multiples of 45° with respect to the positive x-axis. Thus, A is a restricted domain or a line at 45° or 135°.