RECURSIVE MORPHOLOGY USING LINE STRUCTURING ELEMENTS

DESIKA CHARI NADADUR AND ROBERT M. HARALICK
University of Washington
Department of Electrical Engineering
Box 352500, Seattle, WA 98195-2500, U.S.A
E-mail: {desika, haralick}@george.ee.washington.edu

Abstract. Binary morphological dilation and erosion with long line structuring elements is computationally expensive when performed by the conventional methods of taking the unions and intersections of all translates of the input binary image with the structuring element. Thus, the overall computational complexity is a function of the product of the image size and that of the structuring element. This paper discusses one-pass constant time recursive algorithms for performing dilation and erosion of a binary image of a given size, with a line structuring element oriented in a given direction regardless of its length. The input binary image is scanned along a digital line generated in the specified orientation. Starting from every 1-pixel in the image directed distances of pixels are measured along the digital line and the pixel values are replaced with the computed values producing a grey scale image called the transform image. This is then thresholded with the desired length of the structuring element. When the resulting image is appropriately translated to account for the true origin of the structuring element, the result is the desired dilation/erosion. The timing of the recursive algorithm is evaluated with respect to the conventional morphological algorithm. It is shown to achieve a speedup of 5, on an average, over all orientations of the line structuring element of length 150 pixels when using a salt and pepper image of size 240 x 256 with the probability of a pixel being a 1-pixel set to 0.25.

Key words: Mathematical Morphology, Recursive Dilation, Recursive Erosion, Line Structuring Elements.

1. Introduction

Mathematical morphology, an algebraic system, provides a set of operators that when acting upon complex shapes are able to decompose them into their meaningful parts and separate them from the extraneous parts. Therefore, such an algebraic system with its operators and their combinations allows the underlying shapes to be identified and optimally reconstructed from their noisy, distorted forms.

The theory of mathematical morphology(Serra, 1982)(Haralick *et al.*, 1987) has been developed by many researchers. Mathematical morphology is described in the language of sets. Sets in mathematical morphology represent the shapes that appear in binary or static gray scale images. Sets in Euclidean 2-space denote the foreground regions in binary images. Sets in 3-space may denote time-varying images, static gray-scale images or binary solids. Further, sets in higher dimensional spaces may include additional information such as color or multiple perspective imagery. There are two basic operations, *dilation* and *erosion*. These are closely related to the Minkowski addition and subtraction in Euclidean space using translations, unions and intersections. Morphological algorithms can be developed that incorporate various compositions of dilation and erosion operations in order to extract shapes from the imagery. Morphological methods have excelled in image analysis techniques because of its sound mathematical basis and nonlinear nature.

Binary morphological dilation and erosion with long line structuring elements is computationally expensive when performed by the conventional methods of taking the unions and intersections of all translates of the input binary image with the structuring element. Thus, the overall computational complexity is a function of the product of the image size and that of the structuring element. In (Chen and Haralick, 1995) Chen and Haralick describe two-pass constant time algorithms for obtaining binary erosions and dilations recursively using arbitrary structuring elements. However, they do not handle the additional efficiencies which arise in the case of line structuring elements oriented in arbitrary directions.

This paper discusses one-pass constant time recursive algorithms for performing the dilation and erosion of a binary image of a given size, with line structuring elements oriented in arbitrary directions regardless of their length. The input binary image is scanned along a digital line generated in the specified orientation. Starting from every 1-pixel in the image directed distances of pixels are measured along the digital line and the pixel values are replaced with the computed values producing a grey scale image called the transform image. This is then thresholded with the desired length of the structuring element. When the resulting image is appropriately translated to account for the true origin of the structuring element, the result is the desired dilation/erosion. The timing of the recursive algorithm is evaluated with respect to the conventional morphological algorithm. It is shown to achieve a speedup of 5, on an average, over all orientations of a line structuring element of length 150 pixels when using a salt and pepper image of size 240×256 with the probability of a pixel being a 1-pixel set to 0.25.

The paper is organized as follows: Section 2 reviews some general definitions and properties of conventional and recursive mathematical morphological operations. Section 3 states some definitions related to lines in discrete space. Section 4 describes in detail the recursive dilation algorithm. Section 5 describes in detail the recursive erosion algorithm. Section 6 discusses the protocol we used for testing. Section 7 concludes the paper by discussing experimental results.

2. Definitions and Notation

This section provides some background in Mathematical Morphology using set theoretic notation (Serra, 1982)(Haralick *et al.*, 1987). Let $Z = \{z | 0 \leq z < \infty\}$ be the set of integers. Let A, B, C, and K be sets in Z^2 and let the O be the origin of Z^2, i.e., $O \in Z^2$.

A. Review of Mathematical Morphology

Definition 2.1 *The* Translation *of the set A to the point $t \in Z^2$ is defined as, $A_t = \{x | x = a + t$ for every $a \in A\}$.*

Definition 2.2 *The* Reflection *of the set K is denoted by $\check{K}$ and is defined by, $\check{K} = \{-x | x \in K\}$.*

Definition 2.3 Binary Dilation *of a set A by a structuring element K is denoted by $A \oplus K$ and is defined as,* $A \ominus K = \{x \in Z^2 | x = a + b,$ *for some* $a \in A$ *and* $b \in K\}$.

Geometrically, the dilation can be interpreted as the translation of A by all the points in K and then taking the union, i.e., $A \oplus K = \bigcup\{A_b | b \in K\}$.

Definition 2.4 Binary Erosion *of a set A by a structuring element K is denoted by $A \ominus K$ and is defined as,* $A \ominus K = \{x \in Z^2 | x + b \in A$ *for every* $b \in K\}$.

Geometrically, the erosion can be interpreted as the translation of A by all the points in $\check{K}$ and then taking the intersection, i.e., $A \ominus K = \bigcap\{A_b | b \in \check{K}\}$.

B. Review of Recursive Operators

Recursive operations on binary images are accomplished using a particular scan order of pixels in the image. This scan order can be specified using scanning functions.

Definition 2.5 *A* Scanning Fuction *S is defined as a one-to-one mapping from a finite set* $I = \{(x_1, x_2) \in Z^2 | 0 \le x_1 < n_1, 0 \le x_2 < n_2\}$ *to the set* $\{1, 2, \ldots, n_1 n_2\}$. *If $p \in I$, $q \in I$ and $S(p) < S(q)$, then the output value at p is computed before the output value at q.*

Definition 2.6 *A* Recursive Operator *on a binary image is an operator whose output depends not only on the input pixels in the domain of its kernel, but also on the values of the previously computed pixels, with respect to a given scanning function S.*

Definition 2.7 *A* Sequentially Computable recursive operator *is a recursive operator with respect to a given scanning function S in the finite set $I \in Z^2$ such that whenever an output value $q \in I$ is computed, it only depends on input pixel values and those output pixel values at $p \in I$ satisfying $S(p) < S(q)$.*

3. Digital Line Structuring Elements

This section provides definitions related to digital line structuring elements. The following definition states the definition of a continuous line passing through the origin and oriented at an angle θ.

Definition 3.1 *A line in the continuous domain, oriented at an angle θ measured counter-clockwise with respect to the X–axis in the X–Y coordinate system and passing through the origin $O \in R^2$ is denoted by L_θ and is defined as,* $L_\theta = \{(x, y) \in R^2 | x \cos\theta + y \sin\theta = 0\}$.

In the row-column (r-c) coordinate system, let the r-axis coincide with the X-axis and the c-axis coincide with the Y-axis of the X-Y coordinate system. Then in general a digital line closest to a continuous line of some angular orientation θ is defined as follows.

Definition 3.2 *A digital line $D_\theta \subseteq Z^2$ closest to a continuous line at some orientation θ is denoted by D_θ and is defined as,*

$$D_\theta = \left\{(r,c) \in Z^2 \mid \textit{for some } \theta,\ |r\cos\theta + c\sin\theta| \leq \frac{1}{2}\right\}$$

The following states a general definition of discretization operation $\mathcal{D}$ on a set $A \subseteq R^2$ which maps the set A into Z^2.

Definition 3.3 *Let $A = \{(x,y) \in R^2\}$ be a given set in R^2. The* Discretization Operation *$\mathcal{D} : R^2 \to Z^2$ is defined as,*

$$\mathcal{D}(A) = \{(r,c) \in Z^2 \mid (x,y) \in A,\ r = [x] \textit{ and } c = [y]\} \tag{1}$$

where $[\bullet]$ is the rounding off to the nearest integer.

In the following discussion, without loss of generality, we take all lines to be passing through the origin.

The following proposition establishes a way of computing a digital line $D_\theta \subseteq Z^2$ from a given continuous line $L_\theta \subseteq R^2$ oriented at an angle $\theta \in [0,\pi]$ using the definition 3.3.

Proposition 3.1 *Let $L_\theta \subseteq R^2$ be a line in R^2 (as in definition 3.1) oriented at angle $\theta \in [0,\pi]$. Then a* Digital Line *$D_\theta \subseteq Z^2$ is obtained by appying the discretization operation $\mathcal{D}$ from definition 3.3 to L_θ and is given by,*

$$D_\theta = \mathcal{D}(L_\theta) \tag{2}$$

The following describes a way of computing a digital line $D_{n,\theta}$ of length $n > 1$ from D_θ.

Proposition 3.2 *Let n be positive integer such that $n > 1$. Then a digital line $D_{n,\theta} \subseteq Z_n^2$ of length n pixels closest to the given continuous line $L_\theta \subseteq R^2$ oriented at an angle θ, where $Z_n = \{1,\ldots,n\}$ is given by,*

$$D_{n,\theta} = \{(r,c) \in Z_n^2 \mid (r,c) \in D_\theta\} \tag{3}$$

From now on D_θ will be used to denote a digital line closest to the continuous line $L_\theta \subseteq R^2$ oriented at an angle θ, obtained from the discretization $\mathcal{D}(L_\theta)$ of L_θ, and $D_{n,\theta}$ will be used to denote a digital line of length $n > 1$ obtained by restricting D_θ to Z_n^2.

The following definition formalizes the angular separation between two given orientations. The definition of angular separation is necessary when we define the error between the orientation of a continuous line $L_\theta \subseteq R^2$ and that of a digital line $D_{n,\theta} \subseteq Z^2$.

Definition 3.4 Angular Distance *between or* Angular Separation *between two orientations $\theta_1, \theta_2 \in [0,\pi]$, denoted by $d_\theta(\theta_1,\theta_2)$, is defined as,*

$$d_\theta(\theta_1,\theta_2) = \min(|\theta_1 - \theta_2|, |\pi + \theta_1 - \theta_2|)$$

The following defines the orientation of a digital line $D_{n,\theta} \subseteq Z^2$ that is closest to and obtained by the discretization of the continuous line $L_\theta \subseteq R^2$, where L_θ goes through the origin as defined in the definition 3.1.

Definition 3.5 *The* Angular Orientation *of a digital line $D_{n,\theta}$ is denoted by $\Theta(D_{n,\theta})$ and is defined as,*

$$\Theta(D_{n,\theta}) = \arctan \left\{ \frac{\max_{(r,c)\in D_{n,\theta}} c}{\max_{(r,c)\in D_{n,\theta}} r} \right\}. \tag{4}$$

The following defines the orientation error between the continuous line $L_\theta \subseteq R^2$ and its length-n discretization $D_{n,\theta} \subseteq Z_n^2$ with the aid of the definition (3.4).

Definition 3.6 *Given a line $L_\theta \subseteq R^2$ oriented at angle θ and passing through the origin, and $D_{n,\theta} \subseteq Z_n^2$ the digital line closest to L_θ, the* Orientation Error *between of L_θ and $D_{n,\theta}$ is denoted by $\Delta\Theta(L_\theta, n)$ and is defined as, $\Delta\Theta(L_\theta, n) = d_\theta(\Theta(D_{n,\theta}), \theta)$.*

The following proposition states that as the length n of the digital line increases, the orientation error between the line $L_\theta \subseteq R^2$ and the digital line $D_{n,\theta} \subseteq Z^2$ decreases.

Proposition 3.3 *Given a line $L_\theta \subseteq R^2$, as the length n of the corresponding digital line (the size of the domain Z_n^2) increases, the orientation error $\Delta\Theta(L_\theta, n)$ decreases, that is, $\Delta\Theta(L_\theta, n) > \Delta\Theta(L_\theta, n+1)$ where $n > 1$.*

The following proposition states a way of determining the minimum length n of $D_{n,\theta} \subseteq Z^2$, given the line $L_\theta \subseteq R^2$ and the orientation error $\Delta\theta$, allowed between L_θ and $D_{n,\theta}$.

Proposition 3.4 *Given the orientation θ of the line $L_\theta \in R^2$ and the orientation error $\Delta\theta$ between L_θ and $D_{n,\theta} \in Z_n^2$, the value of n can be found as the smallest n satisfying the relation $\Delta\Theta(L_\theta, n) \leq \Delta\theta$.*

In practice, the smallest value of n is found as follows. A lookup table for $\Delta\Theta(L_\theta, n)$ is created in an off-line procedure using the equations (5). For each length n of the digital line, the angle θ of the line L_θ is varied over the range $[0, \pi]$ to obtain the minimum orientation errors for the variations in the orientation of the digital line of length n. Then the maximum of these minimum errors is calculated as the orientation error for a given θ and n. Algorithmically, this can be stated as follows with the following notation. n: Length of the digital line in pixels, i: Number of columns, and, j: Number of rows. Also when $j = 0$, $\arctan(\frac{i}{j}) = 90$ is used.
Construction of the lookup table:
The lookup table for determining the minimum length of the digital line is constructed using the following algorithm using the equations (5):

– For each length $n = 1, \ldots, N$ of the digital line,
  = For each θ in the range $[0, \pi]$,

* vary i the number of columns, and/or j the number of rows,
* Find the minimum orientation errors at each variation of i and/or j.

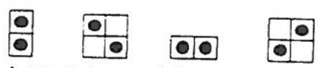
= Find the maximum of the minimum orientation errors.

Therefore each entry in the table shows the orientation error for a given θ and n.

$$\Delta\Theta(L_\theta, n) = \begin{cases} \max\limits_{j=1,\ldots,n} \min\limits_{i=0,\ldots,j} |\theta - \arctan(\frac{i}{j})| & 0 \le \theta \le \frac{\pi}{4} \\ & n = 1,\ldots,N \\ \max\limits_{i=1,\ldots,n} \min\limits_{j=0,\ldots,i} |\theta - \arctan(\frac{i}{j})| & \frac{\pi}{4} < \theta \le \frac{\pi}{2} \\ & n = 1,\ldots,N \\ \max\limits_{i=1,\ldots,n} \min\limits_{j=0,\ldots,i} |\theta - (180 - \arctan(\frac{i}{j}))| & \frac{\pi}{2} < \theta \le \frac{3\pi}{4} \\ & n = 1,\ldots,N \\ \max\limits_{j=1,\ldots,n} \min\limits_{i=0,\ldots,j} |\theta - (180 - \arctan(\frac{i}{j}))| & \frac{3\pi}{4} < \theta < \pi \\ & n = 1,\ldots,N \end{cases} \quad (5)$$

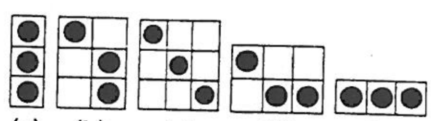
The user specified orientation error value is compared with the error values in the lookup table and the value of n corresponding to the closest error value is taken as the smallest length required to represent the digital line with the specified orientation error.

Figures 1, 2 and 3 show a few examples of the digital lines for $n = 2, 3, 5$ and for varying values of $\theta \in [0, \frac{\pi}{2}]$.



(a) 0° (b) 45° (c) 90° (d) 135°

Fig. 1. $D_{2,\theta}$ specified with $\Delta\Theta(L_\theta, n) = 0$.



(a) (b) (c) (d) (e)

Fig. 2. A few of the possible configurations for $D_{3,\theta}$ when (a)–(c) the number of columns is varied with the number of rows held constant at 3, and (d)–(e) the number of rows is varied with the number of columns held constant at 3.

4. The Recursive Dilation Transform

The dilation transform of a binary image is based on the successive morphological dilations of the image. Given a binary image $I \subseteq Z^2$ and the set $A \subseteq I$ of foreground or one pixels in I, the dilation transform with respect to the digital line structuring element $D_{n,\theta}$ is a grey scale image where the grey level of each pixel $x \in Z^2$ is the generalized distance of x to the set A, i.e., the generalized distance of x to the foreground or one pixels according to a given scanning function S. That is they depict the smallest positive integers n, such that $x \in A \oplus D_{n,\theta}$. If no such n exists,

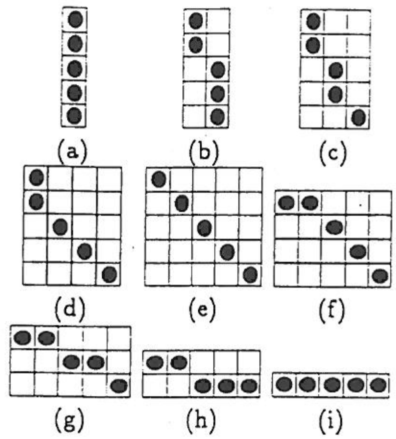


Fig. 3. A few of the possible configurations for $D_{5,\theta}$ when (a)–(e) the number of columns is varied with the number of rows held constant at 5, and (f)–(i) the number of rows is varied with the number of columns held constant at 5.

where $x \notin A \oplus D_{n,\theta}$ for all n, then the dilation transform at $x \in Z^2$ is designated as zero. The support for dilation is the entire Z^2, because the dilation is extensive.

Definition 4.1 *Let $I \subseteq Z^2$ be the binary image and $A \subseteq I$ be the set of foreground or one pixels in I. Let $D_{n,\theta} \subseteq Z^2$ be the digital line structuring element containing the origin $O \in D_{n,\theta}$ at its top end. The dilation transform of the set A is denoted by*

$$F_d(A, D_{n,\theta})(x) = \begin{cases} \min\{n | x \in A \oplus D_{n,\theta}\} & \text{if } \forall n,\ x \in A \oplus D_{n,\theta} \\ 0 & \text{if } \forall n,\ x \notin A \oplus D_{n,\theta} \end{cases} \quad (6)$$

The following proposition states that the dilation of the set A by a line structuring element of length n with its origin at its top end, can be obtained by a simple thresholding step.

Proposition 4.1 *Let l be a positive integer. If $A \subseteq Z^2$ is set and $D_{n,\theta}$ is a digital line structuring element containing the origin $O \in D_{n,\theta}$ at its top end, and $B_l = \{x \in Z^2 | 0 < F_d(A, D_{n,\theta}) \leq l\}$, then $A \oplus D_{l,\theta} = B_l$.*

The following proposition establishes that since the thresholding step results in a dilation with a line structuring element with the origin at its top end, a translation is necessary to account for the true origin of the actual line structuring element.

Proposition 4.2 *Let $D_{l,\theta} \subseteq Z^2$ be the digital line structuring element of length l pixels obtained from the same $L_\theta \subseteq R^2$ as $D_{n,\theta}$. Let $z \in D_{l,\theta}$ denote the origin of the structuring element $D_{l,\theta}$, then, $B_d = \{x + z | x \in B_l, z \in D_{l,\theta}\}$ is the desired recursive dilation result.*

4.1. The Dilation Transform Algorithm

In (Rosenfeld and Pfaltz, 1968), Rosenfeld and Pfaltz described a two-pass recursive algorithm to compute the city-block and chess-board distance transform of a binary image. They calculate the global distances in the image by propagating the local distances, i.e., the distances between the neighbouring pixels. As a generalization, Bertrand (Bertrand and Wang, 1988),(Wang and Bertrand, 1992) and Haralick (Haralick and Shapiro, 1992) described a two-pass recursive algorithm to compute the generalized distance transform.

The recursive dilation transform algorithm descibed here computes the dilation transform in a single-pass over the image as follows: The minimum length digital line structuring element computed as in section 3 is used to scan the the image in the left-to-right and top-to-bottom sequence if $0^o \leq \theta \leq 90^o$ and in the right-to-left and top-to-bottom sequence if $90 < \theta \leq 180^o$ assuming its origin to be at the top end. For each one pixel encountered, the transform values are propagated along the digital line in the top-to-bottom fashion till the border of the image is reached. Then, the digital line is moved to the next one pixel in the image and the transform is computed. This procedure is followed for all the one pixels in the image.

The following defines the nature of the scanning function over the image used while scanning the pixels in the image, by the digital line structuring element, while obtaining the dilation transform.

Definition 4.2 *Let $I \subseteq Z^2$ denote the binary image. The Scanning function S over the image I is denoted by $S(I)$ and is established as,*

$$S(I) = \begin{cases} \textit{left-to-right and top-to-bottom scan} & \textit{if } 0 \leq \theta \leq \frac{\pi}{2} \\ \textit{right-to-left and top-to-bottom scan} & \textit{if } \frac{\pi}{2} < \theta \leq \pi \end{cases} \tag{7}$$

The following definition establishes the nature of the scanning function over the digital line structuring element. This scanning function is used to propagate transform values to the pixels in the image along the structuring element starting from a given pixel in the image in obtaining the dilation transform.

Definition 4.3 *Let $D_{n,\theta} \subseteq Z^2$ denote a digital line structuring element. The Scanning function S over $D_{n,\theta}$ is denoted by $S(D_{n,\theta})$ and is established as,*

$$S(D_{n,\theta}) = \textit{top-to-bottom along the digital line for } 0 \leq \theta \leq \pi \tag{8}$$

The following proposition establishes the dilation transform property for line structuring elements. It indicates that the dilation transform at any pixel is one plus the nonzero dilation transform value computed at its immediately preceding pixel in the image according to the scanning function over the digital line structuring element.

Proposition 4.3 *If $A \subseteq Z^2$ is a set, $D_{n,\theta} \subseteq Z^2$ is digital line structuring element containing the origin at its top end. If $x \in Z^2$ and $C = \{x - 1 | F_d(A, D_{n,\theta})(x-1) > 0\}$ is the singleton set containing the nonzero immediate predecessor of x according to*

$S(D_{n,\theta})$, *then*

$$F_d(A, D_{n,\theta})(x) = \begin{cases} 1 & \textit{if } x \in A \\ \min\{F_d(A, D_{n,\theta})(x), \\ \quad \{F_d(A, D_{n,\theta})(x-1) | x-1 \in C\} + 1\} & \textit{if } x \notin A,\ C \neq \phi \\ 0 & \textit{if } x \notin A,\ C = \phi \end{cases} \quad (9)$$

The above proposition 4.3 leads to the following recursive algorithm for the computation of the dilation transform.

4.1.1. *Algorithm Description*

Let $A \subseteq Z^2$ be a set (of foreground pixels in the input image), and let $D_{n,\theta} \subseteq Z^2$ be the line structuring element with the origin $O \in D_{n,\theta}$ at its top end. Let the scanning functions be chosen as in propositions (4.2) and (4.3).

Algorithm: Recursive Dilation Transform

1. Perform the following filtering on each pixel x in the input image (Proposition 4.3):
   – if $x \in A$, then $F_d(A, D_{n,\theta})(x) = 1$.
   – if $x \notin A$, then,

$$F_d(A, D_{n,\theta})(x) = \min\{F_d(A, D_{n,\theta})(x), \{F_d(A, D_{n,\theta})(x-1) | x-1 \in C\} + 1\}.$$

The following section discusses the process of obtaining the dilation of the input image A by the structuring element $D_{n,\theta}$ from the dilation transform image of A.

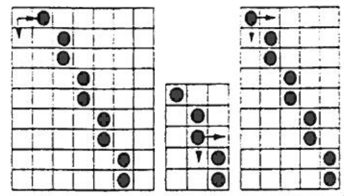
4.2. The Recursive Dilation

This section describes the process of obtaining the dilation from the dilation transform. According to the proposition (4.1) the dilation transform is thresholded by the length of the actual structuring element. The binary image resulting from thresholding process represents a dilation with a line structuring element with its origin at its top end. According to the proposition (4.2) the binary image is translated to account for the true location of the structuring element origin. The following sub-section provides an illustration for the *recursive dilation algorithm*.

4.3. An Illustration for Recursive Dilation Algorithm

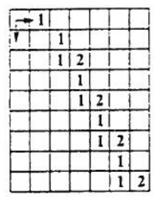
This section provides an illustration to make clear the workings of the *recursive dilation algorithm*.

We use the image A depicted in figure 4 as the input image to the algorithm. The digital line $D_{9,30^\circ}$ used in obtaining the dilation transform and the actual line structuring element $D_{5,30^\circ}$ are also shown in figure 4. The threshold value is therefore, $l = 5$. Let the origin of $D_{5,30^\circ}$ be located at $z = (2,1)$. The Scanning functions are chosen as described in the propositions (4.2) and (4.3). After thresholding the dilation transform image shown in figure 5 by l and translating the result of thresholding by z, we obtain the dilation of the input binary image with the structuring element of length 5 with the origin at $(2,1)$. The result of recursive dilation algorithm is shown in figure 6. In the next section we describe the recursive erosion transform.



A $D_{l,\theta}$ $D_{m,\theta}$

Fig. 4. Input Binary Image A, and the actual Line Structuring Element $D_{l,\theta}$ and the line structuring element $D_{m,\theta}$ used in obtaining the dilation transform with $\theta = 30°$, $l = 5$ and $m = 9$.



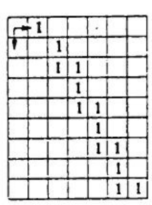
$F_d(A, D_{9,30°})$

Fig. 5. Resulting dilation transform of image A shown in figure 4 using the structuring element $D_{9,30°}$ also shown in figure 4.

5. The Recursive Erosion Transform

The recursive erosion transform is based on the successive morphological erosions of the binary image. It is a generalization of the distance transform commonly known in the literature (Rosenfeld and Pfaltz, 1968).

Given a binary image $I \subseteq Z^2$, and the set $A \subseteq I$ of all the one pixels or the foreground pixels, the erosion transform of A with respect to the digital line structuring element $D_{n,\theta} \subseteq Z^2$ is a grey scale image where the grey level of each pixel $x \in A$ is the generalized distance of x to the image background, i.e., the largest positive integer n such that $x \in A \ominus D_{n,\theta}$. The generalized distance at a pixel x indicates the maximum number of consecutive erosions of A by $D_{n,\theta}$ such that x is still contained



$A \oplus D_{5,30°}$

Fig. 6. Final output of the recursive dilation algorithm. This is the dilation of the image A by a digital line structuring element of length 5 and oriented at 30° from the row axis.

in the eroded image foreground. The support for erosion is the foreground set A. The following definition 5.1 defines the erosion transform of A by the structuring element $D_{n,\theta}$.

Definition 5.1 *Let $I \subseteq Z^2$ be a binary image and $A \subseteq I$ be the set of foreground or one pixels in I. Let $D_{n,\theta} \subset Z^2$ be the digital line structuring element containing the origin $O \in D_{n,\theta}$ at its top end. The dilation transform of the set A with respect to $D_{n,\theta}$ is denoted by $F_e(A, D_{n,\theta})$ and is defined as,*

$$F_e(A, D_{n,\theta})(x) = \begin{cases} \max\{n|x \in A \ominus D_{n,\theta}\} & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \tag{10}$$

The following proposition establishes that the erosion of the set A by a line structuring element of length l containing the origin O at its top end, can be accomplished by a simple thresholding step.

Proposition 5.1 *Let l be a positive integer. If $A \subseteq Z^2$ is a set and $D_{n,\theta}$ is a line structuring element containing the origin $O \in D_{n,\theta}$ at its top end, and $B_l = \{x \in A | F_e(A, D_{n,\theta})(x) \geq l\}$, then $A \ominus D_{l,\theta} = B_l$.*

According to the proposition 5.1 the erosion obtained is with respect to the line structuring element with the origin at its top end (at the bottom end of the reflected structuring element). Therefore, a translation by the negative points of the structuring element is necessary to account for the true origin of the structuring element. The following proposition establishes the process of translation.

Proposition 5.2 *Let $D_{l,\theta} \subseteq Z^2$ be the digital line structuring element and $z \in D_{l,\theta}$ denote the origin of the line structuring element $D_{l,\theta}$, then,*

$$B_e = \{x - z | x \in B_l, z \in D_{l,\theta}\}$$

is the desired erosion result.

The following section describes the erosion transform algorithm.

5.1. The Erosion Transform Algorithm

The recursive erosion transform algorithm descibed here computes the erosion transform in a single-pass over the image as follows: The minimum length digital line structuring element computed as in section 3 is used to scan the the image in the right-to-left and bottom-to-top sequence if $0^o \leq \theta \leq 90^o$ and in the left-to-right and bottom-to-top sequence if $90 < \theta \leq 180^o$ assuming its origin to be at the bottom of the reflected digital line structuring element. For each one pixel encountered, the transform values are propagated along the digital line in the bottom-to-top fashion till the border of the image is reached. Then, the digital line is moved to the next one pixel in the image and the transform is computed. This procedure is followed for all the one pixels in the image.

The following definition establishes the nature of the scanning function over the image used while scanning the pixels in the image by the digital line structuring element while obtaining the erosion transform.

Definition 5.2 *Let $I \subseteq Z^2$ denote a binary image. The scanning function S over the image I is denoted by $S(I)$ and is established as,*

$$S(I) = \begin{cases} \textit{right-to-left and bottom-to-top scan} & \textit{if } 0 \leq \theta \leq \frac{\pi}{2} \\ \textit{left-to-right and bottom-to-top scan} & \textit{if } \frac{\pi}{2} < \theta \leq \pi \end{cases} \quad (11)$$

The following definition establishes the nature of the scanning function over the digital line structuring element. This scanning function is used to propagate transform values to the pixels in the image along the structuring element starting from a given pixel in the image in obtaining the erosion transform.

Definition 5.3 *Let $D_{n,\theta}$ denote a digital line structuring element. The scanning function S over $D_{n,\theta}$ is denoted by $S(D_{n,\theta})$, and is established as,*

$$S(D_{n,\theta}) = \textit{bottom-to-top along the digital line for } 0 \leq \theta \leq \pi \quad (12)$$

The following proposition establishes the erosion transform property for line structuring elements. It indicates that the erosion transform at any pixel is one plus the erosion transform value computed at its immediately preceding pixel in the image according to the scanning function chosen over the digital line structuring element.

Proposition 5.3 *Let $A \subseteq Z^2$ is a set, $D_{n,\theta} \subseteq Z^2$ is a digital line structuring element containing the origin $O \in D_{n,\theta}$ at its top end. If $x \in Z^2$ and $C = \{x - 1 | F_e(A, D_{n,\theta}) \geq 0\}$ is the singleton set containing the immediate predecessor of x according to the scanning function $S(D_{n,\theta})$ over the digital line structuring element, then,*

$$F_e(A, D_{n,\theta})(x) = \begin{cases} 0; & \textit{if } x \notin A, \\ \max\{F_e(A, D_{n,\theta})(x), \\ \quad \{F_e(A, D_{n,\theta})(x-1) | x - 1 \in C\} + 1\}; & \textit{if } x \in A,\ C \neq \phi, \\ \max\{1, F_e(A, D_{n,\theta})(x)\}; & \textit{if } x \in A,\ C = \phi. \end{cases} \quad (13)$$

5.1.1. *Algorithm Description*

Let $A \subseteq Z^2$ be a set (of foreground pixels in the input image), and let $D_{n,\theta} \subseteq Z^2$ be a digital line structuring element with the origin $O \in D_{n,\theta}$ at its top end. Let the scanning functions be chosen as described in propositions (5.2) and (5.3).

Algorithm: Recursive Erosion Transform

1. Perform the following filtering on each pixel x in the input image (Proposition 5.3):
   - If $x \notin A$, then $F_e(A, D_{n,\theta})(x) = 0$;
   - If $x \in A$, then

$$F_e(A, D_{n,\theta})(x) = \max\{F_e(A, D_{n,\theta})(x), \{F_e(A, D_{n,\theta})(x-1) | x - 1 \in C\} + 1\}.$$

The following section describes the process of obtaining the erosion of image A by the structuring element $D_{n,\theta}$ from the erosion transform of the image A.

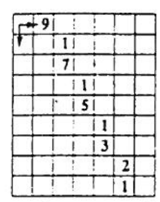
5.2. The Recursive Erosion

This section describes the process of obtaining the erosion from the erosion transform. According to the proposition (5.1) the erosion transform is thresholded by the length of the true structuring element. The resulting binary image represents an erosion of input image with a line structuring element with its origin at its top end. According to the proposition (5.2) the result of thresholding is translated to account for the true location of the structuring element origin. The following sub-section provides an illustration for the *resursive erosion algorithm.*

5.3. An Illustration for Recursive Erosion Algorithm

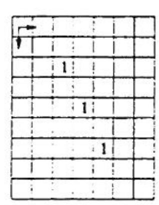
This section provides an illustration to clarify the workings of the *recursive erosion algorithm.*

We use the image A depicted in figure 4 as input to the algorithm. The digital line $D_{9,30^\circ}$ used in obtaining the erosion transform and the actual line structuring element $D_{5,30^\circ}$ is also shown in figure 4. The *threshold* value is $l = 5$. Let the origin of $D_{5,30^\circ}$ be located at $z = (2, 1)$. Therefore, the translation required is z. Scanning function is chosen as in the propositions (5.2) and (5.3) to obtain the erosion transform. After thresholding the erosion transform image shown in figure 7 by l and translating the result of thresholding by z, we obtain the erosion of the input binary image with the structuring element of length 5 with the origin at $(2, 1)$. The result of erosion is shown in figure 8.



$F_e(A, D_{9,30^\circ})$

Fig. 7. Erosion Transform of image A shown in figure 4 obtained using the reflection of the structuring element $D_{9,30^\circ}$ also shown in figure 4.



$A \ominus D_{5,30^\circ}$

Fig. 8. Resultant image after the translation operation; this is the final erosion using symmetrical line structuring element of length 5 oriented at 30° from the row axis.

6. Testing Protocol

The following steps are carried out in the timing evaluation of the recursive dilation and erosion algorithms for lines.

1. Input binary images are generated as follows: Salt and Pepper images with maximum value in the image = 1; minimum value in the image = 0 are generated with varying fraction of 1-pixels. It can be observed that the execution time is a function of number of one pixels or the foreground pixels in the image.
2. Then the recursive dilation and erosion algorithms are run on these images with structuring elements in different orientations with differing lengths for each orientation. Hence, the total number of tests = number of images × number of orientations × number of lengths.
3. Conventional dilation and erosion algorithms performed by the unions and intersections of the input binary image with the structuring element, are run on the same set of input binary images using the same set of actual digital line structuring structuring elements (for example, in the discussions in the previous sections the actual digital line structuring element is $D_{5,30^\circ}$).
4. Execution times were noted for both the recursive and the conventional morphological algorithms and are plotted against the length of the structuring elements on the X-axis.
5. It was expected that the curves obtained for the recursive algorithms to be flat depicting a constant execution time and those for the conventional morphology to be linearly increasing with the lengths of the structuring elements.
6. The *average* and the *worst-case* execution times are also calculated and plotted.

7. Experimental Results

Several experiments were carried out according to the protocol described in section 6 to compare the time taken by the algorithm discussed in this paper and that taken by the conventional method of obtaining dilation and erosion.

The results were plotted with lengths of the structuring elements as independent variable on the X-axis and the execution times as dependent variables on the Y-axis.

Results for dilation and erosion algorithms are given in figure. 9. Plots for 0^o and 90^o are shown along with those for average and worst-case execution times. In obtaining these plots, *salt and pepper* images of size 240 × 256 pixels with the probability of a pixel being a 1-pixel set to 0.25, generated synthetically, are used. The codes for conventional as well as recursive dilation and erosion were run on SUN Sparc-2 machines with the programs compiled with *Optimize* flag on. From the graphs it becomes evident that the recursive algorithm works at constant time ignoring (1) the effect of image size increase with the size of the structuring element, due to the required buffering for border pixels and (2) the inaccuracies in measuring the CPU time consumed in running the algorithm. These curves show only the times taken to run the algorithm and thus do not include the I/O time, the time taken to generate the structuring element, and buffering for border pixels. It can be seen that on an *average*, over all orientations of a line structuring element of length 150, the recursive algorithm shows a speedup of approximately 5 over the conventional algorithm using a salt and pepper image of size 240 × 256 with the probability of a pixel being a 1-pixel set to 0.25. It can be seen from the graphs that on average a speedup of 5 is obtained with recursive algorithm over the conventional algorithm for digital line structuring elements of length 150 pixels over all orientations.

8. Conclusions and Future Work

In this paper, recursive algorithms for binary dilation and erosion using digital line structuring elements are discussed. It is shown to take constant time irrespective of the length of the structuring element for its various orientations, for a given size of the binary image. We showed that our algorithm achieved a speedup of about 5 for salt-pepper images of size 240×256 with the probability of pixel being a one pixel set to 0.25, over the conventional morphological operations based on unions and intersections when a digital line structuring element of length 150 pixels is used.

In the future, we would like to extend these recursive algorithms to arbitrarily shaped structuring elements, since they can be obtained by the union of translations of parallel digital line structuring elements.

9. Acknowledgements

The first author greatfully acknowledges the valuable discussions with V.Ramesh, A.Bedekar, B.Modayur and C-K.Lee, and the source code for conventional morphology from T.Kanungo.

References

R. M. Haralick and L. G. Shapiro, *Computer And Robot Vision*, Addison-Wesley Publishers, 1992.

R.M.Haralick S.R.Sternberg X. Zhuang, "Image Analysis Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 9, no. 4, pp. 532–550, 1987.

J.Serra, *Image Analysis and Mathematical Morpphology*, London: Academic, 1982.

S. Chen and R.M. Haralick, "Recursive erosion, dilation, opening, and closing transforms", *IEEE trans. Image Processing*, vol. 4, no. 3, pp. 335–345, March 1995.

A. Rosenfeld and J.L. Pfaltz, "Distance functions in digital pictures", *Pattern Recognition*, vol. 1, pp. 33–61, 1968.

G. Bertrand and X. Wang, "An algorithm for a generalized distance transformation based on minskowski operations", *in Proc. 9th ICPR (Rome)*, pp. 1163–1167, November 1988.

X.Wang and G.Bertrand, "Some sequential algorithms for a generalized distance transformation based on minskowski operations", *IEEE trans. PAMI.*, vol. 14, no. 11, pp. 1114–1121, 1992.