

遞迴型態學運算的聯集形式分解 及心脈式陣列設計

楊狄龍*

摘要

多摺型態學運算中大量的疊代計算造成及時實現的瓶頸，遞迴型態學運算可藉由少量的掃描次數解決此類疊代問題。然而，建構單元若具有相對於原點位置的元素不能被循序計算，不可以使用於遞迴型態學。因此，遞迴型態學運算中不能循序計算的建構單元必須被分解成若干能循序計算的較小建構單元。本文針對上述問題提出聯集形式的分解方法，任何包含原點的建構單元可利用此法被分解成僅含兩個元素的建構單元。此類建構單元具有最簡單的資料相依特性，方便於心脈式陣列設計中處理單元及時序的指定，以降低運算所需的時間及硬體成本。距離轉換相當於多摺型態學中的侵蝕運算，當疊代次數足夠大使得運算所定義的範圍不再變化為止。本文選擇距離轉換作為設計範例，相同範例本文所提出的設計比目前現有的設計速度快三倍，且成本僅須該設計三分之二。

目次

- | |
|---|
| I. INTRODUCTION |
| II. BRIEF REVIEW OF MATHEMATICAL MORPHOLOGY |
| III. THE UNION-FORM STRUCTURING ELEMENT DECOMPOSITION |
| IV. DISTANCE TRANSFORMS |
| V. MODULAR SYSTOLIC REALIZATION |
| VI. CONCLUSION |

關鍵字：型態學運算、建構單元分解、心脈式陣列。

楊狄龍* 高雄高工電子科教師，國立高雄海洋技術學院電訊系兼任助理教授／國立成功大學電機工程研究所通訊組博士畢業

A Union-Form Decomposition and Systolic Realization for Recursive Morphological Operations

Dyi-Long Yang and Chin-Hsing Chen

Telephone: 0920-379301,

E-mail: ydl12@ms38.hinet.net.

Abstract

A large number of iterations in the n -fold morphological operation results in a bottleneck in the real-time implementation. Recursive morphology can be used to solve the iterative processing problem by few scans. However, structuring elements with opposite points about the origin are not sequentially computable, they can't be applied in the recursive operation. Therefore, a non-sequentially computable structuring element must be decomposed into a set of sequentially computable structuring elements for the recursive morphology. This problem can be solved by the union-form decomposition method proposed in this paper. According the method, every structuring element containing the origin can be decomposed into a set of two-point structuring elements. This decomposition is significant because it has the simplest data dependence that offers advantages in processor assignment and the scheduling, thus reducing computation time and hardware cost. The distance transform, which can be implemented by the n -fold erosion when the value of n is large enough that the image of interest occurs idempotence, is taken as a design example for modular systolic arrays of the recursive morphology. For the same example, the speed of our design is 3 times and the cost is $2/3$ times that of the existing design.

Key words: Morphological operation, Structuring element decomposition, Systolic array.

I. INTRODUCTION

Morphological Operations are widely used in image processing applications for noise suppression, image enhancement, feature extraction, thinning, shape recognition, etc. Erosion and dilation are two elementary morphological operations since other morphological filters, such as openings and closings, are constructed from them.

Recursive morphology can be used to avoid the iterative processing in the distance transform [1]-[3]. Successive iterative morphological erosions on a binary image with distance structuring elements

produces the distance transform of the image. All distance structuring elements must satisfy the constraints of distance measures, i.e., positive definite, symmetrical and the triangle inequality. The number of iterations in a distance transform is proportional to the object size within the input image. In worst case, $M/2$ iterations are required for an $M \times M$ input image. It results in a bottleneck in the real-time implementation. To avoid iteration, the recursive morphological operation, which requires only few scans of the image, is employed. In this case, the distance transform becomes independent of the object size. For a recursive morphological operator, every pixel in the destination image is a function of both the central pixel (the normal pixel) that lies on the origin of the structuring element in the source image, and the pixels (the recursive pixels) that lie within the domain of the structuring element except for the central pixel in the destination image. Associated with a particular recursive morphological operator there is a particular scan order of the input pixels. This scan order governs the order where the operator is to be applied. The well known algorithm proposed by Rosenfeld, Pfalz [4] and G. Borgefors [5] is a two-pass operation using two complementary raster scans: the first pass goes from pixel to pixel in a top-to-bottom and left-to-right sequence (forward pass), and the second pass goes in the reverse direction (backward pass). In each pass, a subset of the structuring element of each pixel is examined, and a recursive operator is applied.

A systolic array is a network of processors which rhythmically computes and passes data through the system [6]. The data move through the processors in a rhythmic fashion and very simple operations are performed on them. The systolic architecture has the advantages of modularity, regularity, local interconnection, a high degree of pipelining and very amenable to VLSI implementation. Mapping an algorithm onto a systolic array [7] consists of two major tasks—mapping the algorithm to a dependence graph (DG) and mapping DG to a systolic array. A DG can be considered as the graphical representation of a single assignment algorithm in which every variable is assigned one value only during the execution of the algorithm. A complete DG specifies all the dependencies between all variables in the index space. An algorithm is sequentially computable if and only if the complete DG contains no loops or cycles [7]. There are two basic considerations for mapping a DG to a systolic array in order to obtain an efficient architecture in both hardware and speed. One is the processor assignment and the other is the scheduling. The processor assignment specifies the nodes of DG in a certain straight line to a processing element (PE). The schedule vector assigns the nodes of the same schedule hyperplane to be computed at the same clock cycle. In fact, every permissible schedule vector decides a scan order mentioned above of a recursive operator. The processor assignment and scheduling must satisfy that any edges of DG will have one or more than one delays and nodes on equitemporal hyperplanes are not projected to the same PE. There is no permissible linear schedule vector if the DG contains any loops or cycles.

Structuring elements with opposite points about their origin are not sequentially computable, since their corresponding DGs have loops or cycles. Therefore, they must be decomposed into smaller

sequentially computable structuring elements when a recursive operator is applied. Besides, a structuring element must be decomposed if its domain is too large to handle in one stage. A large number of methods about structuring element decomposition have been proposed in the literature [8]-[15]. In this paper, it will be shown that the structuring element can be decomposed into a sequence of subsets whose union is equal to the structure element itself for recursive morphological operations, in which the structure element and all subsets contain the origin. This is to say that the structuring element decomposition can be replaced as

$$B = B_1 \cup B_2 \cup \cdots \cup B_N \quad (1)$$

for recursive morphology, where B is the structuring element and B_i , $i = 1, 2, \dots, N$ are smaller sequentially computable structuring elements. This result indicates that a two-point structuring element decomposition involving only B_i , to which the origin belongs, can be obtained. The two-point decomposition is important because it has the simplest data dependence offering advantages in processor assignment and the scheduling. This can lead to the reduction of computation time and hardware cost. By the proposed method, the structure of every PE for the recursive dilation/erosion operation with any size or shape of structuring element can be implemented by a shift register, an adder/subtractor and a maximum/minimum operator. The above two-pass raster scans can be modified into a four-pass operation using two complementary horizontal line scans and two complementary vertical line scans. The first pass of the horizontal line scans goes from line to line in a left to right sequence, and the second pass goes from right to left. The first pass of the vertical line scans goes from line to line in a top to bottom sequence, and the second pass goes in the reverse direction.

In reference [1], the canonical two-pass algorithm is adopted to design a pipeline architecture for recursive morphological operations. In their design, for an $M \times M$ image with 3×3 square structuring element, M processing elements are required to form a systolic array for processing the whole image. The number of PE can be reduced to a half if a feedback path from the output of the $\lceil M/2 \rceil$ th PE is provided to the input of the first PE. Each PE is realized by six sets of an adder cascaded by a comparator. The total execution time of their systolic array for two scans of the whole image is $8M + 4(M-1)$ and the duration of a clock cycle is equal to the delays of one comparator plus one adder. This design is not a modular systolic array. In this paper, the proposed two-point decomposition is applied to design modular systolic arrays. For the same input image size and structuring element, the speed of our design is 3 times and its cost is $2/3$ times that of the systolic array designed in [1].

In the next section, we will review the definitions and some well-known properties of the basic morphological operations. In Section III, the union-form decomposition for the recursive morphology is developed, and then verified by the concept of vector space. In Section IV, we will review the well-known distance transform and its relationship to the n -fold erosion. In Section V, The distance

transform is taken as a design example to design the modular systolic array. The distance structuring element of interest is first decomposed into a set of two-point structuring element by the proposed union-form decomposition. Then, the resulting two-point set is applied to design the hardware architecture. Finally, in Section VI, conclusions are given.

II. BRIEF REVIEW OF MATHEMATICAL MORPHOLOGY

In this section, some essential properties of mathematical morphology related to this paper will be briefly reviewed, and the details can be found in [16], [17]

Definition 1: The *translation* of a set $A \subset Z^m$ by a point $x \in Z^m$ is denoted by A_x and is defined by

$$A_x = \{a+x \mid a \in A\}$$

Geometrically, A_x is A translated along vector x .

Definition 2: The *dilation* of a set $A \subset Z^m$ by a set $B \subset Z^m$ is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{a+b \mid a \in A \text{ and } b \in B\}$$

where A is the input image and B is the structuring element. $A \oplus B$ is the set of all possible vector sums of pair elements: one coming from A and the other coming from B . Thus, dilation operations are commutative and associative: $A \oplus B = B \oplus A$ and $A \oplus (B \oplus C) = A \oplus B \oplus C$.

Definition 3: The *erosion* of a set $A \subset Z^m$ by a set $B \subset Z^m$ is denoted by $A \ominus B$ and is defined by

$$A \ominus B = \{x \mid B_x \subseteq A\}$$

$A \ominus B$ consists of all points x for which the translation of B by x fits inside of A . Erosion is the morphological dual of dilation and has the iterative property: $A \ominus (B \oplus C) = A \ominus B \ominus C$. Rather than eroded by $B \oplus C$ whose size might be too large, A is iteratively eroded by B and then by C .

Morphological operations can be defined in both the binary domain and in the gray-scale domain. The processes of erosion and dilation in the gray-scale domain are the same as convolution and correlation if the multiplication and summation are replaced by subtraction (addition) and minimum (maximum) selection.

Definition 4: Let $F, K, \subset Z^m$, $f:F \rightarrow Z$ and $k:K \rightarrow Z$. The *gray-scale dilation* of a signal f by a structuring element k is denoted by $f \oplus k$, $f \oplus k : F \oplus K \rightarrow Z$ and is defined by

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

Definition 5: The *gray-scale erosion* of a signal f by a structuring element k is denoted by $f \ominus k$,

$f \ominus k : F \ominus K \rightarrow Z$ and is defined by

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

There is a close relationship between the gray-scale and the binary morphology. This relationship is formalized by the umbra transform. The importance of the umbra in mathematical morphology is that it provides a mechanism for expressing gray-scale operations in terms of binary operations.

Definition 6: The *umbra* of f consists of all points in the space lying beneath the graph of f , including the points on the graph itself. It is denoted by $U[f]$ and is defined by

$$U[f] = \{(x, y) | x \in F \text{ and } y \leq f(x)\}$$

where $U[f]$ is a subset space of $F \times Z$.

Definition 7: The *top surface* of a set $A \subset Z^m \times Z$ is denoted by $T[A]$ and is defined by

$$T[A] = \{(x, y) \in A | y \geq z \text{ for any } (x, z) \in A\}$$

for any signal f , the top surface of its umbra is the signal f itself, namely,

$$T[U[f]] = f \tag{2}$$

Regarding dilation and erosion, there are two fundamental umbra proposition relating binary and gray-scale morphology.

$$f \ominus k = T[U[f] \ominus U[k]] \tag{3}$$

$$f \oplus k = T[U[f] \oplus U[k]] \tag{4}$$

Gray-scale dilation (or erosion) can be performed by taking the umbra of the signal and the structuring element, performing a binary dilation (or erosion) using these umbra, and then taking the top surface of the result.

Definition 8: The *n-fold dilation* of a set B is denoted by nB and is defined by

$$nB = \underbrace{B \oplus B \oplus \dots \oplus B}_n$$

nB is extensive under increasing n when B contains the origin. If B can be decomposed into B_1 and B_2 , i.e., $B = B_1 \oplus B_2$, then

$$\begin{aligned} nB &= (B_1 \oplus B_2) \oplus (B_1 \oplus B_2) \oplus \dots \oplus (B_1 \oplus B_2) \\ &= \underbrace{(B_1 \oplus B_1 \oplus \dots \oplus B_1)}_{2n} \oplus \underbrace{(B_2 \oplus B_2 \oplus \dots \oplus B_2)}_n \\ &= nB_1 \oplus nB_2 \end{aligned} \tag{5}$$

According to the commutative and associative properties of the dilation operation, the *n-fold* dilation and erosion operations have the same properties as the dilation and erosion operations as follows

$$\begin{aligned} A \oplus nB &= (((A \oplus B) \oplus B) \oplus \cdots \oplus B) \\ &= ((A \oplus nB_1) \oplus nB_2) \end{aligned} \quad (6)$$

$$\begin{aligned} A \ominus nB &= (((A \ominus B) \ominus B) \cdots \ominus B) \\ &= ((A \ominus nB_1) \ominus nB_2) \end{aligned} \quad (7)$$

Rather than dilated or eroded by nB , in which B might be non-sequentially computable, A is dilated or eroded iteratively by $n B_1$ and then by $n B_2$, where both B_1 and B_2 are sequentially computable.

Definition 9: Let I denotes the domain of the image of interest. The n -fold dilation is said to be *idempotent* within the image of interest if, for any set B ,

$$I \cap nB = I \cap (n-1) B.$$

III. THE UNION-FORM STRUCTURING ELEMENT DECOMPOSITION

An n -fold morphological operation has an one-pass recursive form if its structuring element is sequentially computable. The one-pass recursive form is equivalent to the n -fold morphological operation when the value of n is large enough that the iteration of the morphological operation over the image occurs idempotence. For a binary operation with any n value, the n -fold morphological operation can be solved by a threshold operation after the computation of the recursive form has been completed [2]. Some sequentially computable structuring elements are shown in Fig. 1. If a structuring element is not a sequentially computable it can be decomposed into two or more smaller sequentially computable structuring elements. According to Eqs. (6) and (7), the n -fold morphological operation have an n -pass form and is based on a few passes of the operation over the image. This reduces the computation complexity of the n -fold morphological operation from $O(nM^2)$ to $O(M^2)$ for an $M \times M$ input image.

The structuring element decomposition problem can be solved by tree searching algorithms with some constraints including translation constraints, opening constraints and functional value constraints for a gray-scale structuring element decomposition [13]. Specially, it must satisfy the sequentially computable constraints for the n -fold morphological operation. It is observed that not any structuring elements can be decomposed under the above constraints. The problem can be solved by the union-form structuring element decomposition. In this section, the union-form structuring element decomposition is proposed and verified.

Proposition 1: Let $B \subset Z^m$. The n -fold dilation nB , for any set B containing the origin, decides a subset vector space of Z^m , where every member belonging to the subset vector space is a linear combination of the position vectors of the points within B except for the origin with non-negative integer coefficients whose summation is equal to or less than the value of n , that is

$$nB = \left\{ \sum_{i=1}^{L-1} c_i \alpha_i \mid \sum_{i=1}^{L-1} c_i \leq n \text{ and } \alpha_i \in B \right\}$$

where L is the number of points within B , c_i is a non-negative integer coefficient, α_i is the position vector of a point within B and α_0 is denoted as the origin of B .

Proof: The proof follows easily from *Definition 2*.

$$\begin{aligned} 2B &= B \oplus B \\ &= \left\{ \sum_{i=0}^{L-1} c_i \alpha_i \mid \sum_{i=0}^{L-1} c_i = 2 \text{ and } \alpha_i \in B \right\} \end{aligned} \quad (8)$$

Since c_i is a non-negative integer and $\sum_{i=0}^{L-1} c_i = 2$, we have $c_i \in \llbracket 0, 2 \rrbracket$ and

$$2B = \left\{ \sum_{i=1}^{L-1} c_i \alpha_i \mid \sum_{i=1}^{L-1} c_i \leq 2 \text{ and } \alpha_i \in B \right\} \quad (9)$$

Extending this result to nB , we have $c_i \in \llbracket 0, n \rrbracket$ and

$$nB = \left\{ \sum_{i=1}^{L-1} c_i \alpha_i \mid \sum_{i=1}^{L-1} c_i \leq n \text{ and } \alpha_i \in B \right\} \quad (10)$$

Proposition 2: Let $I \subset Z^m$ denote the domain of the image of interest. When n is large enough that the iteration of the morphological operation over the image occurs idempotence, the vector space of nB within I is a linear combination of the position vectors of the points within B .

Proof: When n is large enough that the iteration of the morphological operation over the image satisfies idempotence, the condition $\sum_{i=1}^{L-1} c_i \leq n$ can be eliminated, that is

$$\lim_{n \rightarrow \infty} nB = \left\{ \sum_{i=1}^{L-1} c_i \alpha_i \mid \alpha_i \in B \right\} \quad (11)$$

Since α_0 denotes the origin of B , we have

$$\lim_{n \rightarrow \infty} nB = \left\{ \sum_{i=0}^{L-1} c_i \alpha_i \mid \alpha_i \in B \right\} \quad (12)$$

and the vector space within the image is

$$I \cap \lim_{n \rightarrow \infty} nB = I \cap \left\{ \sum_{i=0}^{L-1} c_i \alpha_i \mid \alpha_i \in B \right\} \quad (13)$$

Proposition 3: For the n -fold dilation when the image occurs idempotence, those points within

B are *redundant points* if its position vector is the linear combination of the position vector of the other points within B , since both the set of B and the set of B but redundant points decide the same vector space within the image I .

Definition 10: For the n -fold dilation when the image occurs idempotence, The points of B but redundant points can be defined as the bases which decide the vector space of nB within the image I .

Proposition 4: Let $B = B_1 \cup B_2$ and, $B' = B_1 \oplus B_2$ both B_1 and B_2 contain the origin α_0 . For the n -fold dilation when the image occurs idempotence, B and B' have the same bases.

Proof: From the definition of dilation

$$\begin{aligned}
 B' &= B_1 \oplus B_2 \\
 &= \left\{ \alpha_i + \alpha_j \mid \alpha_i \in B_1 \text{ and } \alpha_j \in B_2 \right\} \\
 &= \left\{ \alpha_i \mid \alpha_i \in B_1 \right\} \cup \left\{ \alpha_j \mid \alpha_j \in B_2 \right\} \\
 &\quad \cup \left\{ \alpha_i + \alpha_j \mid \alpha_i \in B_1, \alpha_j \in B_2 \text{ and } i, j \neq 0 \right\} \\
 &= B_1 \cup B_2 \cup \left\{ \alpha_i + \alpha_j \mid \alpha_i \in B_1, \alpha_j \in B_2 \text{ and } i, j \neq 0 \right\} \\
 &= B \cup \left\{ \alpha_i + \alpha_j \mid \alpha_i \in B_1, \alpha_j \in B_2 \text{ and } i, j \neq 0 \right\}
 \end{aligned} \tag{14}$$

This result indicates that $B \subset B'$ and the points belonging to $B' - B$ are redundant points for the n -fold dilation when the image occurs idempotence, since they are vector sums, linear combinations with non-negative integer coefficients, of pair elements coming from B . Therefore, B and B' have the same bases.

Proposition 5: Let I denote the domain of the image of interest, $B = B_1 \cup B_2$ and $B' = B_1 \oplus B_2$, both B_1 and B_2 contain the origin α_0 . For the n -fold dilation when the image occurs idempotence, B and B' decide the same vector space within the image I , i.e.,

$$I \cap \lim_{n \rightarrow \infty} nB = I \cap \lim_{n \rightarrow \infty} nB'$$

Proposition 6: For the n -fold dilation when the image occurs idempotence, the structuring element decomposition can be replaced by the union-form decomposition as the following

$$\lim_{n \rightarrow \infty} nB = \lim_{n \rightarrow \infty} nB' = \lim_{n \rightarrow \infty} n(B_1 \oplus B_2) = \lim_{n \rightarrow \infty} (nB_1 \oplus nB_2)$$

where

$$B = B_1 \cup B_2$$

The proposition can be extended as the following

$$\begin{aligned}
 \lim_{n \rightarrow \infty} nB &= \lim_{n \rightarrow \infty} nB' = \lim_{n \rightarrow \infty} n(B_1 \oplus B_2 \oplus \cdots \oplus B_N) \\
 &= \lim_{n \rightarrow \infty} (nB_1 \oplus nB_2 \oplus \cdots \oplus nB_N)
 \end{aligned} \tag{15}$$

where

$$B = B_1 \cup B_2 \cup \cdots \cup B_N$$

The importance of *Proposition 6* is that a structuring element, sequentially computable or non-sequentially

computable, containing the origin and the number of points within the structuring element is larger than two, can be decomposed into two or more smaller sequentially computable structuring elements containing the origin by the proposed union-form decomposition. Consequently, a two-point structuring element decomposition involves only B_i , including the origin, can be obtained.

According to the umbra transform reviewed in Section II, all the propositions developed in this section can be applied to gray-scale morphology. Similarly, any gray-scale structuring element containing the origin can be decomposed into a set of two-point structuring elements for a n -fold morphological operation when the image occurs idempotence. For example, the 3×3 square structuring element which is not sequentially computable can be decomposed into eight two-point structuring elements as shown in Fig. 2, in which every two-point structuring element consists of the origin whose value is equal to zero, and one of the other points within the original 3×3 square structuring element. Although the dilation of the eight two-point structuring elements is not equal to the original 3×3 square structuring element as shown in Fig. 3, it has the same bases as and contains the original 3×3 square structuring element. For convenience, the results of the 2-fold dilation of them are shown in Figs. 4 (a) and (b). Let the image of interest is the n -fold dilation of the original 3×3 square structuring element, it is observed that the union-form decomposition has the same result within the domain of the image of interest.

IV. DISTANCE TRANSFORMS

Consider a binary image consisting of object pixels and background pixels. Distance transforms produce a mapping from double-valued image, where background pixels are identified with value 0 and object pixels with value ∞ , to a gray-scale image where every pixel has a value proportional to its distance to the nearest background pixel in the original image. The distance transform has been widely used in shape analysis and skeletonization [18]-[21].

Distance transforms can be implemented by eroding the double-valued image by a large conical structuring element, its diameter must be greater than the largest distance existing in the image of interest. Such an erosion is basically a global operation which is prohibitively costly and unsuitable for VLSI implementation. Erosion and dilation with a large conical structuring element can be implemented as iterative operations with a distance structuring element which consider only the local neighborhood but still give a reasonable approximation to the Euclidean distance. When a binary-valued image is eroded by a local distance structuring element, the first iteration builds only a thin ring around the background. Every successive iteration adds the same wide ring around the one built by the previous erosion. The iterative erosion is continued until the rings fill all the object pixels. If eroding the image

further will not modify the results anymore; idempotence is then reached [22]. This gives a simple stopping criterion: the algorithm may be stopped when no pixels are modified at some iteration. The number of iteration is proportional to the greatest distance in the image of interest. In worst case, it is equal to $M/2$ for an $M \times M$ input image.

A distance structuring element is defined by the local distance between the origin and all the pixels in a small neighborhood around it. The values of the distance structuring element are selected to be negative of the associated distance measures because the erosion is the minimum selection after the subtraction is applied. The 3×3 and 5×5 distance structuring elements are shown in Fig. 5. The most common values of a, b and c for various distance transforms are listed in Table 1. Maximum errors that can occur between the distance transforms and the Euclidean distance are also listed [23], [24].

V. MODULAR SYSTOLIC REALIZATION

According to the proposed union-form decomposition, a set of two-point structuring elements can be obtained for the recursive morphology. In this section, this two-point set decomposition is applied to derive modular systolic arrays. For convenience, the distance transforms with 5×5 distance structuring element reviewed in the previous section are taken as a sign example.

A. Dependence Graphs

In the follow, we derive DG for the proposed two-point decomposition in the context of the distance transform. The resulting decomposition of the 5×5 distance structuring element consists of sixteen two-point structuring elements. They can be classified into four classes by their data dependence. The part of the horizontal line scan goes from left to right is shown in Fig. 6, the others of the rest line scans have the similar form and are neglected. According to the definition of erosion, the corresponding recursive algorithm of distance transforms can be represented as follows

For x from 1 to M

For y from 1 to M

$$f^1(x, y) = \min(f^0(x, y), f^1(x-1, y) + a) \quad (16)$$

$$f^2(x, y) = \min(f^1(x, y), f^2(x-1, y-1) + b) \quad (17)$$

$$f^3(x, y) = \min(f^2(x, y), f^3(x-2, y-1) + c) \quad (18)$$

$$f^4(x, y) = \min(f^3(x, y), f^4(x-2, y+1) + c) \quad (19)$$

where $f^{k-1}(x, y)$ and $f^k(x, y)$ denote the input and resulting images, respectively, when the k th two-point structuring element is executed, and $f^0(x, y)$ is the original input binary image. The execution order k of two-point structuring elements can be selected randomly since dilation operations are

commutative, but we assign an order for easy presentation in the above algorithm. This algorithm has been converted to a single assignment form by introducing the execution order k . The algorithm can be directly mapped onto the DG shown in Fig. 7 for the $M=4$ case, with each xy -plane drawn separately. What should have been included but not drawn are the dependence lines in the k -direction that run from points $(x, y, k-1)$ to (x, y, k) .

B. Mapping Dependence Graphs to Systolic Arrays

The DG in Fig. 7 is not totally regular since the dependence of each xy -plane is different, but it is directionally shift invariant and has the x -direction dependence component for each ij -plane, we can project it directly. In the following, we will use the algebraic approach [7] to determine the systolic schedule vector \vec{s} and the projection vector \vec{d} . A valid systolic schedule vector \vec{s} and projection vector \vec{d} should satisfy the following constraints

$$\vec{s}^T \vec{e}_i > 0; \text{ and } \vec{s}^T \vec{d} > 0 \quad (20)$$

where \vec{e}_i denotes a dependence vector in DG. There are five dependence vectors of the proposed DG, $\vec{e}_1, \dots, \vec{e}_5$, as displayed below.

$$\begin{bmatrix} \vec{e}_1 & \vec{e}_2 & \vec{e}_3 & \vec{e}_4 & \vec{e}_5 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 2 & 2 & 0 \\ 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

The minimum computation time solution for \vec{s} under the above constraints is

$$\vec{s}^T = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \quad (22)$$

The minimum computation time T of every 5×5 line scan for the distance structuring element is

$$T = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} M-1 \\ M-1 \\ 4 \end{bmatrix} + 1 = M + 4 \quad (23)$$

and it is $M+2$ for the 3×3 distance structuring element, i.e., The total computation time is $4 \times (M+4)$ for the 5×5 distance structuring element and is $4 \times (M+2)$ for the 3×3 distance structuring element. From observation, we see that the projection in x -direction is the simplest one. The pipelining period and the number of I/O pins of the resulting systolic are 1 and $2M$, respectively. With \vec{s} and \vec{d} defined, we can find, the projected edge \vec{e}'_i for the edge \vec{e}_i in the systolic array are:

$$\begin{bmatrix} \vec{e}'_1 & \vec{e}'_2 & \vec{e}'_3 & \vec{e}'_4 & \vec{e}'_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

and the delays on the edges given by $D(\vec{e}'_i) = \vec{s}^T \vec{e}'_i$ is

$$\begin{bmatrix} D(\vec{e}'_1) & D(\vec{e}'_2) & D(\vec{e}'_3) & D(\vec{e}'_4) & D(\vec{e}'_5) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2 & 2 & 1 \end{bmatrix} \quad (25)$$

With nodes determined by the projection in the x -direction and the edges and delays determined above, we are able to draw the systolic array shown in Fig. 8. The systolic array in Fig. 8 can be separated into three modules, which are a - b -, and c -modules, shown as Fig. 9. Only a -module is required to implement the city-block distance transform, and the combination of a -module and b -module as shown in Fig. 10 can implement the distance transforms with a 3×3 distance structuring element. Figure 11 shows the detail circuit of the systolic array resulted from Fig. 10. The clock cycle of the proposed systolic array is equal to the computation time of an adder plus a minimum selection. There are many methods to obtain high speed adders and comparators [25], [26]. For the same input image size and structuring element, the speed of the proposed systolic array is 3 times and its cost is 2/3 times that of the systolic array designed in [1].

VI. CONCLUSION

There are two major contributions for the recursive morphology in this paper. One is the derivation of the union-form structuring element decomposition, and the other is the design of a modular systolic array using the proposed decomposition. This result solves the problem that not any structuring element can be decomposed under the opening constraint, the functional value constraint and the sequentially computable constraint. According the proposed union-form decomposition, a set of two-point structuring element containing the origin can be obtained. By applying this two-point set, several modules of linear systolic arrays have been designed. Recursive dilation and erosion operations with any shape and size can be implemented by the combination of these modules.

REFERENCE

- [1] F. Y. Shih, C. T. King and C. C. Pu, "Pipeline architecture for recursive morphological operations," *IEEE Trans. Image Processing*, vol. 4, no. 1, pp. 11-17, Jan. 1995.
- [2] 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.
- [3] C. H. Chen and D. L. Yang, "Fast algorithm and its systolic realization for distance transformation," *IEE Proc. Comput. And Digital Techniques*, vol. 143, no. 3, pp. 168-173, May, 1996.
- [4] A. Rosenfeld and J. L. Pfaltz, "Distance functions in digital pictures," *Patt. Recogn.*, vol. 1, pp. 33-61, 1968.
- [5] G. Borgefors, "Distance transformations in arbitrary dimensions," *Comput. Vision, Graphics, Image Proc.*, vol. 27, pp. 321-345, 1984.

- [6] H. T. Kung, "Why systolic architecture?," *IEEE computer*, vol. 15, no. 1, pp. 37-46, Jan. 1980.
- [7] S. Y. Kung, VLSI array processors. *Prentice-Hall International Editions*, 1988.
- [8] X. Zhuang and R. M. Haralick, "Morphological structuring element decomposition," *Comput. Vision, Graphics, Image Processing*, vol. 35, pp. 370-382, 1986.
- [9] T. Kanungo and R. M. Haralick, "Vector space interpretation for a morphological shape decomposition problem," *J. Math, Image Vision*, vol. 2, no. 1, pp. 51-82, Oct. 1982.
- [10] F. K. Shih and O. R. Mitchell, "Threshold decomposition of gray-scale morphology into binary morphology," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 31-42, 1989.
- [11] X. Zhuang, "Grayscale structuring function decomposition," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, Champaign, IL, June 15-18, 1992.
- [12] C. H. Richardson and R. W. Schafer, "A lower bound for structuring element decompositions," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 4, pp. 365-369, April 1991.
- [13] O. I. Camps, T. Kanungo and R. M. Haralick, "Gray-scale structuring element decomposition," *IEEE Trans. Image Processing*, vol. 5, no. 1, pp. 111-120, Jan. 1996H.
- [14] H. Park and R. T. Chen, "Optimal decomposition of convex morphological structuring elements for 4-connected parallel array processors," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 16, no. 3, pp. 304-313, March 1994.
- [15] —, "Decomposition of arbitrarily shaped morphological structuring elements," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 17, no. 1, pp. 2-15, Jan. 1995.
- [16] R. M. Haralick, S. R. Sternberg and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 4, pp. 532-550, July 1987.
- [17] E. R. Dougherty, An introduction to morphological image processing. Washington: SPIE, 1992.
- [18] P. E. Danielsson, "A new shape factor," *Comput. Graphics Image Proc.*, pp. 292-299, 1978.
- [19] C. Lantuejoul, "Skeletonization in quantitative metallography," *Issues In Digital Image processing*, R. M. Haralick and J. C. Simon Eds. Maryland: Sijthoff and Noordhoff, 1980.
- [20] P. A. Maragos and R. M. Scafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 34, pp. 1228-1244, Oct. 1986.
- [21] C. W. Niblack, P. B. Gibbons and D. W. Capson, "Generating skeletons and centerlines from the distance transform," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 5, pp. 420-437, Sep. 1992.
- [22] A. Bleau, J. D. Guise and A. R. LeBlanc, "A new set of algorithms for mathematical morphology I. Idempotence geodesic transforms," *CVGIP: Image Understanding*, vol. 56, no. 2, pp. 178-209, Sep. 1992.
- [23] G. Borgefors, "Distance transformations in digital images," *Comput. Vision, Graphics, Image*

Proc., vol. 34, pp. 344-371, 1986.

[24] —, “Note: another comment on ‘A note on ‘Distance transformations in digital images’”, “CVGIP: Image Understanding, vol. 54, no. 2, pp. 301-306, Sep. 1991.

[25] M. Shoji, CMOS digital circuit technology. Prentice-Hall, 1987.

[26] N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI design. Addison-Wesley, 1993.

A	b	c	Name	Max. error %
1	-	-	city-block	58.6
1	1	-	chessboard	41.4
2	3	-	2-3 chamfer DT	13.4
3	4	-	3-4 chamfer DT	8.1
5	7	-	5-7 chamfer DT	8.3
4	6	9	4-6-9 chamfer DT	8.9
5	7	11	5-7-11 chamfer DT	2.0

Table 1 Maximum errors that can occur between the distance transforms and the Euclidean distance.

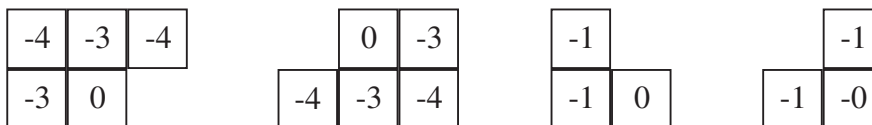


Fig. 1 Several sequentially computable structuring elements.

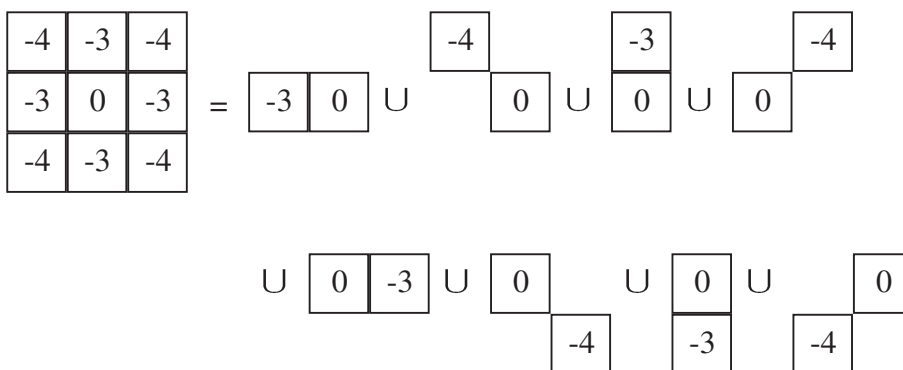


Fig. 2 An example of the proposed union-form decomposition of the structuring element.

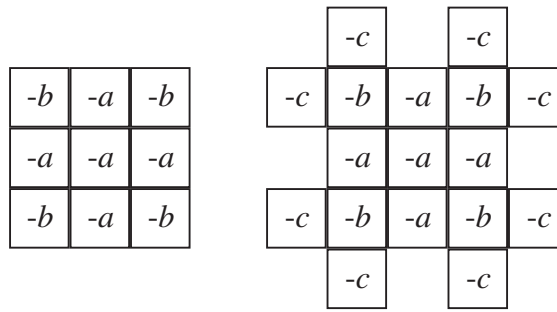


Fig. 5 Structuring elements for distance transformation.

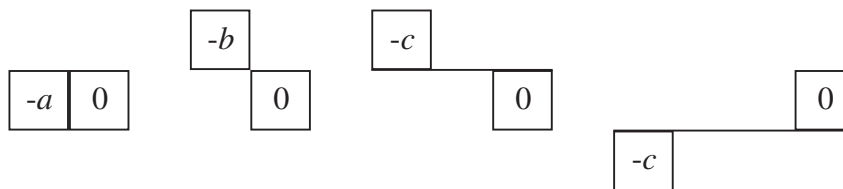


Fig. 6 Partial decomposed two-point structuring elements of the horizontal line scan goes from left to right of the 5×5 distance-structuring element.

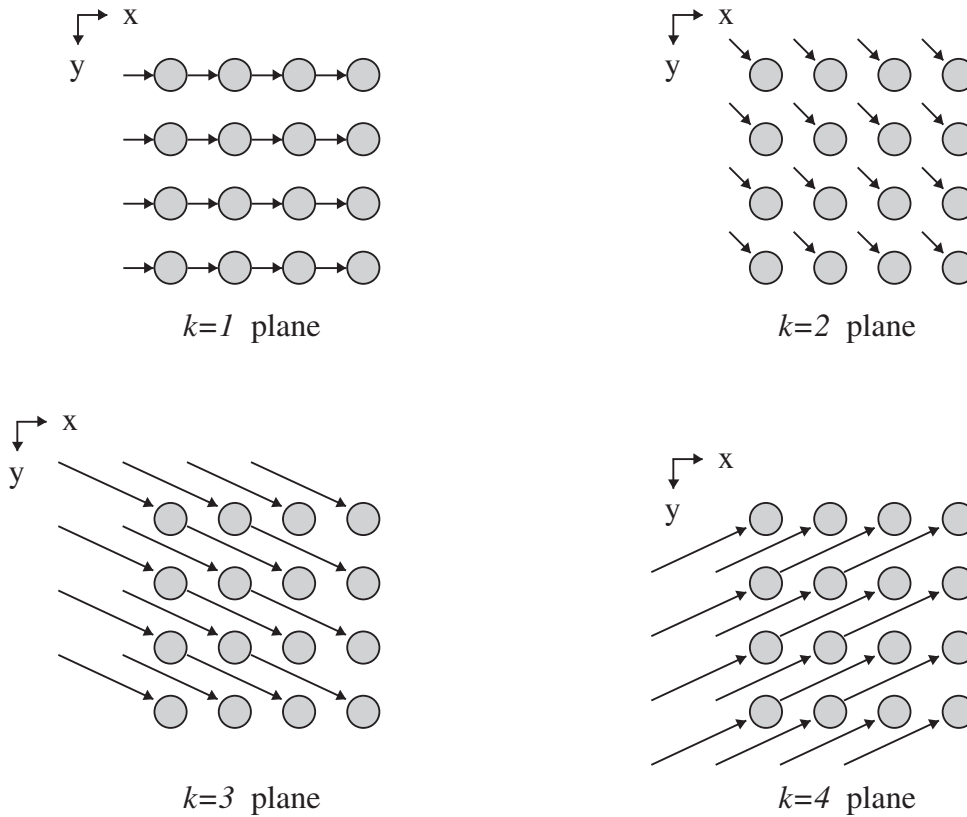


Fig. 7 Dependence arcs in ij -plane in DG for the case $M=4$.

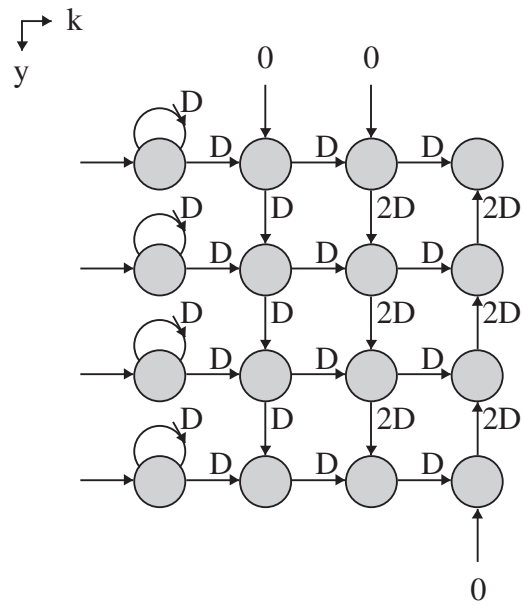


Fig. 8 The resulting systolic array of distance transforms with the 5×5 distance structuring element.

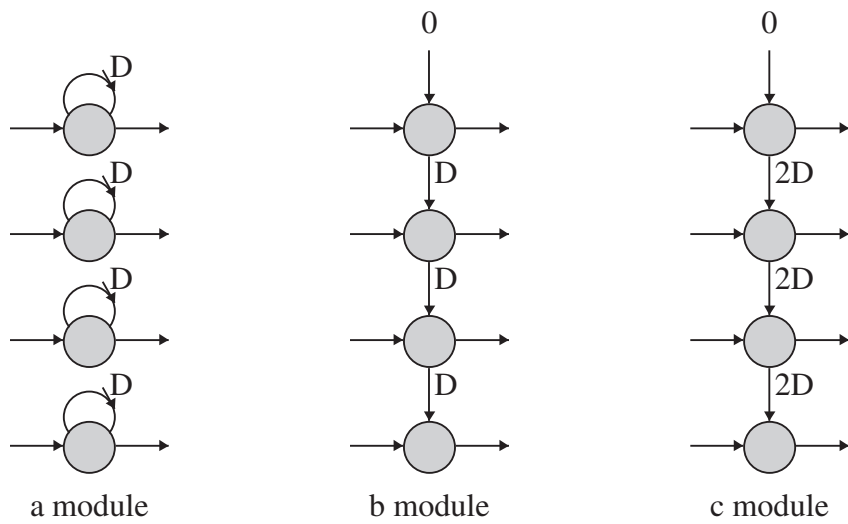


Fig. 9 Three systolic modules separated from Fig. 8.

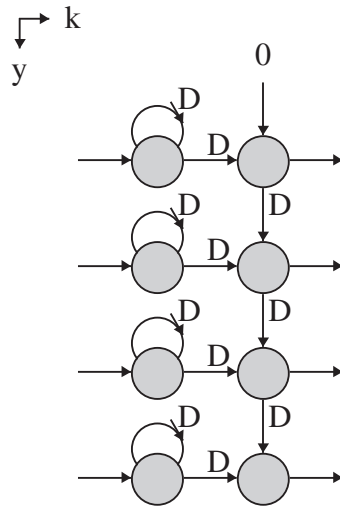


Fig. 10 The resulting systolic array of distance transforms with the 3×3 distance structuring element.

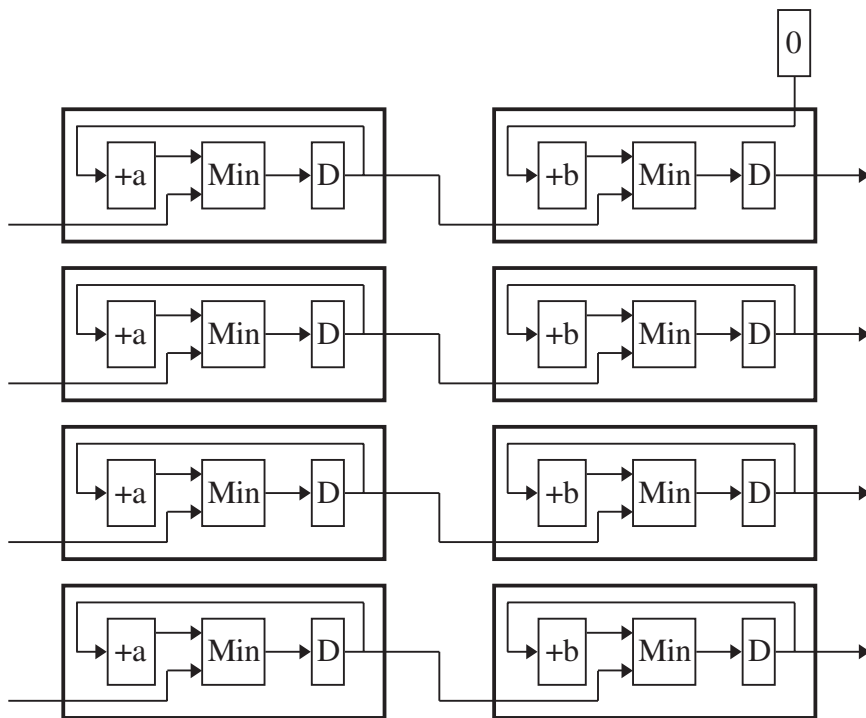


Fig. 11 The detail circuit of the systolic array of distance transforms with the 3×3 distance structuring element.