High-Speed Architectures for Morphological Image Processing

A.C.P. Loui, A.N. Venetsanopoulos and K.C. Smith

Department of Electrical Engineering
University of Toronto
Toronto, Ontario, CANADA M5S 1A4

ABSTRACT

This paper presents a dual architecture for the high-speed realization of basic morphological operations. Since morphological filtering can be described as a combination of erosion and dilation, two basic building blocks are required for the realization of any morphological filter. Architectures for the two basic units, namely the erosion unit and the dilation unit, are proposed and studied in terms of cycle time, hardware complexity, and cost. These basic units are similar in structure to the systolic array architecture used in the implementation of linear digital filters. Correspondingly, the proposed units are highly modular and are suitable for efficient VLSI implementation. These basic units allow the processing of either binary or gray-scale images. They are particularly suitable for applications in robotics, where speed, size and cost are of critical importance.

1. INTRODUCTION

Image processing and analysis based on mathematical morphology has been an active research area in recent years [1,2]. The strength of mathematical morphology lies in its natural coupling between the shape of the image under investigation and the structuring element. That is, by carefully selecting a suitable structuring element, morphological operations can be used for image filtering such as noise removal and image smoothing. Mathematical morphology can also be used as an image analysis tool which results in different shape descriptors. Examples of morphological shape descriptors are the pecstrum (pattern spectrum) [3,4], the skeleton transform [5], and the geometrical correlation function [6].

There exist a variety of machines that are capable of performing morphological or cellular-logic operations. In general, these machines can be divided into two main classes. The first class basically consists of two-dimensional (2-D) array processors which operate on an entire image (or subimage) in parallel. Examples of machines of this type of machines are MITE [7], CLIP [8], and PIXIE-5000 [9]. The main drawback of 2-D array processors is their cost. For example, a 512-by-512 array would require a quarter-of-a-million processing elements. In addition, due to the inherent serial nature of the input image data, full utilization of the processors may not be attained.

Machines of the second class can be called local-window processors; they operate by scanning in an image and performing operations on a small neighborhood window. Examples of machines of this class include the Cytocomputer [10], a structure based on convolution and table lookup [11] and the threshold-decomposition realization [12]. In particular, the Cytocomputer consists of a serial pipeline of neighborhood- processing stages, with a common clock, in which a single neighborhood transformation is performed in each stage of the pipeline.

Most of the machines listed above are rather general-purpose, in the sense that they usually require the control of a host computer. As a consequence, most of these machine may not be truly optimized for morphological image processing and analysis. Besides, the general-purpose nature of these machines does not provide a cost-effective way of implementing specific morphological filters and operators. Hence, as the applications of mathematical morphology become more and more

specialized, the need for dedicated architectures is inevitable. Furthermore, the real-time processing constraints usually imposed by machine vision applications, such as object identification and defect identification, highlight the critical importance of modular and flexible architectures and algorithms.

In this paper, architectures based on the idea of systolic arrays are proposed for high-speed morphological image processing. These dual architectures, which consist of two basic building units, do not depend on the image structure or size: rather, they relate to the structure or size of the structuring element. Hence, structuring elements for various morphological filters or operators can be configured using different combinations of the proposed basic building units. This approach is inherently modular, so that it provides a very flexible system for the implementation of any morphological filter or operator. It also provides a very cost-effective way of designing dedicated morphological filters for image processing as well as morphological operators for image analysis. Furthermore, it is capable of handling binary as well as gray-scale operations. In the next section, basic morphological operations, such as dilation, erosion, opening and closing, are reviewed. In Section 3, the implementation aspects of the morphological dilation and erosion transformations are examined. Specifically, dual architectures which implement the dilation and erosion units respectively, are presented and their operations are described. In addition, parameters such as cycle time, latency, hardware complexity and cost are examined. Both gray-scale and binary architectures are considered. In Section 4, some implementation examples based on the proposed dual architectures are given. These include implementations of the morphological skeleton transform and the pecstrum. Finally, some concluding remarks are given in Section 5.

2. BASIC MORPHOLOGICAL TRANSFORMATIONS

Mathematical morphology provides a very effective tool for extracting geometrical information from image signals. It is basically a set-theoretical methodology for signal analysis, whose primary function is to give a quantitative description of geometrical structures. In brief, there are two primary steps in image analysis that is based on mathematical morphology: a geometrical transformation and then some measurement.

Let X denotes the image signal under study. According to the above procedure, any morphological operation consists of first a transformation Λ (by a pre-selected structuring element B) from one domain to another, followed by a measure μ. This procedure is depicted in Figure 1. Examples of transformations Λ are opening, boundary extraction and skeletonization. The measurement $\mu[\Lambda(X)]$ could be a quantity representing weight, area or volume etc. Hence, quantitative information about size, shape, spatial distribution, connectivity, convexity, and orientation can be obtained by geometrically transforming the object representation using different structuring elements and subsequently making an appropriate measurement. Since morphological operations are global operations, this implies that a transformation can be carried out globally without a need to define the operations at local points.

The four basic morphological transformations are dilation, erosion, opening and closing. In the following, the definitions and some properties of these basic transformations are given. However, the emphasis will be on the dilation and erosion operations, since any morphological transformation can be realized by different combinations of these two most-basic operations.

2.1 Dilation and Erosion

Binary morphological transformations apply to sets of any dimensions, whether they constitute a Euclidean n-space, E^n, or its discrete or digitized equivalent, an integer n-space, Z^n. For simplicities sake, E^n will be used here to refer to either of the two spaces. Hence, if X and B are sets in E^n, then their elements are $x = (x_1, \cdots, x_n)$, and $b = (b_1, \cdots, b_n)$, respectively. Also note that the definitions of dilation and erosion used here are based on those in [2], which are slightly different from Serra's [1] definitions.

Let's begin with the definition of binary dilation. Binary dilation is the morphological transformation which combines two sets using vector addition of set elements.

Definition 1: Let X and B be subsets of E^n. The translation of X by b is denoted by $(X)_b$, and is defined by

$$(X)_b = \{z \in E^n \mid z = x + b \text{ for some } x \in X\} . \tag{1}$$

Definition 2: Let X and B be subsets of E^n. The *binary* dilation of X by B is denoted by $X \oplus B$, and is defined by

$$X \oplus B = \{z \in E^n \mid z = x + b \text{ for some } x \in X \text{ and } b \in B\} \tag{2}$$

According to the above definitions, we also may write

$$X \oplus B = \bigcup_{b \in B} (X)_b \tag{3}$$

Next, the definition of gray-scale dilation is given. Let's begin with the concepts of surface of a set and the umbra of a surface in Euclidean n space. For a set X, the top surface of X is a function defined on the projection of X onto its first $(n-1)$ coordinates. For each $(n-1)$-tuple x, the top surface of X at x is the highest value y, such that the n-tuple $(x,y) \in X$. For a digital space, a similar idea is used, but in this case, the supremum operation is translated into a maximum operation.

Definition 3: Let $X \subseteq X^n$ and $F = \{x \in E^{n-1} \mid \text{for some } y \in E, (x,y) \in X\}$. The top surface of X , $T[X] : F \to E$, is defined by

$$T[X](x) = \max \{y \mid (x,y) \in X\}. \tag{4}$$

Definition 4: Let $F \subseteq E^{n-1}$ and $f : F \to E$. The umbra of f, $U[f]$, $U[f] \subseteq F \times E$, is defined by

$$U[f] = \{(x,y) \in F \times E \mid y \leq f(x)\}. \tag{5}$$

This implies that the umbra of a function f, is a set consisting of the surface f and everying below the surface. Hence, a function can be considered as the top of its umbra. Having defined the operations of finding the top surface of a set and the umbra of a surface, we can now define gray-scale dilation.

Definition 5: Let $F, G \subseteq E^{n-1}$ and $f : F \to E$ and $g : G \to E$. The *gray-scale* dilation of f by g is denoted by $f \oplus g$, $f \oplus g : F \oplus G \to E$, and is defined by

$$f \oplus g = T[U[f] \oplus U[g]] \tag{6}$$

Hence, the gray-scale dilation of two functions is defined as the top surface of the dilation of their umbras.

Erosion is the morphological dual to dilation. For binary images, erosion is a morphological transform which combines two sets using the vector differences of set elements. Erosion is sometimes referred to as "shrinking" in the image processing literature.

Definition 6: Let $X \subseteq E^n$ and $B \subseteq E^n$. The *binary* erosion of X by B is denoted by $X \ominus B$ and is defined by

$$X \ominus B = \{z \in E^n \mid z + b \in B\} . \tag{7}$$

Similarly, erosion also can be defined as

$$X \ominus B = \bigcap_{b \in B} (X)_{-b} . \tag{8}$$

The definition for gray-scale erosion follows in a similar way to the definition of gray-scale dilation.

Definition 7: Let $F, G \subseteq E^{n-1}$ and $f : F \to E$ and $g : G \to E$. The *gray-scale* erosion of f by g is denoted by $f \ominus g$, $f \ominus g : F \ominus G \to E$, and is defined as

$$f \ominus g = T[U[f] \ominus U[g]] . \tag{9}$$

2.2 Opening and Closing

In many applications, erosion and dilation are usually used together. These types of iterative operations are known to have the characteristic of preserving global geometric structures of the unsuppressed features, i.e., only specific image details, which are smaller than the structuring element, are eliminated. Specifically, the opening of X by B, $X \circ B$, is defined as the eroding of X by B and then dilating the result by B, i.e.,

$$X \circ B = (X \ominus B) \oplus B \tag{10}$$

Its dual operation is closing, which is denoted $X \bullet B$, and is defined as the dilating of X by B and then eroding the result by B, i.e.,

$$X \bullet B = (X \oplus B) \ominus B \tag{11}$$

One important property of morphological opening and closing is that they are idempotent. This means that successive openings or closings by the same structuring element (or function in case of gray-scale) do not alter the result after the first application. This corresponds in concept to the ideal nonrealizable bandpass filters of conventional linear filtering [2]. That is, once an image is ideal-bandpass-filtered, no further bandpass filtering can change the result.

3. ARCHITECTURES FOR MORPHOLOGICAL TRANSFORMATIONS

In this section, a pipeline architecture based on the idea of the systolic array, is proposed for performing morphological operations. Since the basic operations in mathematical morphology are dilation and erosion, it is then logical to construct only two basic building blocks, namely the dilation unit and the erosion unit. Thereafter, any other morphological operation can be implemented using the two basic building units and some logic gates. The advantage of this approach is the maximum utilization of processing elements attained due to the pipelined nature of the architecture. The architecture is also well-suited to VLSI implementation because of its modularity, as well as the fact that all data transfers are highly localized (except for a common-clock line).

In the design of special-purpose VLSI systems, hardware cost is usually less important than design cost since such systems are seldom produced in large quantities. Hence, it is important that the design cost of such a special- purpose system be relatively small for it to be competitive with a general- purpose system. As a consequence, modular structures that can be constructed using only a few simple building blocks are highly desirable. In addition, in VLSI design, routing cost normally dominates the power and area required for the implementation. Hence, modular structures that can be constructed using only a few standard cells with regular and simple communication and control are preferred. In other words, we are interested in architectures which require a minimum amount of global broadcasting and fan-in (collection of intermediate results), and which can be expanded easily without the need for changing the entire layout. The proposed architecture, which is composed of various combinations (or cascades) of two basic building blocks, is inherently modular; hence it is very suitable for efficient VLSI implementation.

3.1 Dilation Unit

In Section 2.1, the concept of the surface of a set, and the related concept of the umbra of a surface, are used to define gray-scale morphological operations. These definitions, however, do not provide us with efficient ways of computing gray-scale dilation and erosion. In this Section, alternative definitions of gray-scale morphological transformations are employed which will lead to more practical structures.

From Section 2, it is shown that gray-scale dilation (equation (6)) can be computed by taking the top surface of the dilation of the umbras of the image f and the structuring function g. However, according to the definition of top surface, this

basically is equivalent to performing a maximum operation. Specifically, if we apply the definitions of binary dilation (equation (2)) and umbra (equation (5)) in (6), after some algebraic manipulation, it can be shown that gray-scale dilation is equivalent to taking the maximum of a set of sums [2].

Proposition 1: Let $f:F\rightarrow E$ and $g:G\rightarrow E$. Then $f\oplus g:F\oplus G\rightarrow E$ can be computed by

$$(f\oplus g)(x)=\max\{f(x-z)+g(z)\}\quad \forall z\in G,\ x-z\in F\ . \tag{12}$$

Note that when the structuring function is defined as follows:

$$g(x)=0,\quad \forall x\in G\ , \tag{13}$$

then it is equivalent to a structuring set and $f\oplus g$ becomes:

$$(f\oplus g)(x)=\max\{f(x-z)\}\quad \forall z\in G,\ x-z\in F\ . \tag{14}$$

The operation implied by the equation above is called the dilation of a function by a set [1].

The proposition above leads naturally to what we call a direct-form implementation of gray-scale dilation. The direct-form representation implies that gray-scale dilation can be modeled using a similar structure as employed in linear digital filtering. The exception is that in this case the operations involved are shifting, addition and maximum, instead of the conventional convolution operations of shifting, multiplication and addition used in linear digital filtering.

The direct-form structure described by (12) is shown in Figure 2. However this structure is still not practical for VLSI implementation due to its inherent reliance on non-local data transfers. Hence, it was conjectured that by localizing the data transfers, a more practical structure could be created. One, which is based on the idea of systolic arrays as used in linear digital filtering, is proposed here. A diagram of such a structure, which we call a dilation unit, is shown in Figure 4 (a). A cascade of identical dilation units can be used to implement a dilation operation.

As seen from Figure 4 (a), the dilation unit is composed basically of four single-bit shift registers, one adder and one comparator. Due to the pipelined nature of these units, the cycle time, T_c, of this architecture is determined by the longest operation undertaken between two shift register elements: In this case, it is equal to one addition time, i.e., $T_c^d=T_a$, where T_a is the time for one addition. This implies that very high throughput rates can be attained after an initial latency period required to fill up the pipeline. The precision of the dilation unit is b bits. If b is equal to 1, the structure degenerates into a binary dilation unit. In this case, the addition becomes a logical AND operation, and the comparison (or maximum) becomes a logical OR. Hence, by replacing the adder by an AND gate and the comparator by an OR gate, a binary dilation unit results, as shown in Figure 5 (a). In this case, the cycle time reduces to one gate delay, i.e., $T_c^d=T_{pd}$ where T_{pd} is the propagation delay of a single logic gate.

The block diagram of an example system which implements a 1×3 structuring function, is depicted in Figure 6. The operations of this morphological transformation are described in the space-time diagram of Figure 7. In particular, the latency of this pipeline architecture, is $T_l^d=(N+3)T_c$, where N is the length of the structuring function.

3.2 Erosion Unit

The rationale underlying the erosion unit is similar to that of the dilation unit: The structure derives from the direct-form representation for gray-scale erosion. As done for dilation, this is created by applying the definitions of binary erosion and umbra to equation (9). After some algebraic manipulation, it can be shown that gray-scale erosion is equivalent to taking a minimum of a set of differences [2]. In fact, we can think of erosion as equivalent to correlation, where the summation operation is replaced by a minimum operation, and multiplications become subtractions.

The corresponding direct-form representation for gray-scale erosion is given by the following proposition [2]:

Proposition 2: Let $f:F \to E$ and $g:G \to E$. Then $f \ominus g:F \ominus G \to E$ can be computed by

$$(f \ominus g)(x) = \min\{f(x+z) - g(z)\} \quad \forall z \in G, x+z \in F \quad . \tag{15}$$

Similarly, if $g(x)$ is defined as in (13), (15) becomes

$$(f \ominus g)(x) = \min\{f(x+z)\} \quad \forall z \in G, x+z \in F \tag{16}$$

which is the erosion of a function by a set.

The block diagram of the direct-form implementation of erosion is depicted in Figure 3. Note that this is very similar to the one for dilation, wherein the maximum operations are replaced by minimum operations, and additions become subtractions. Note also that the order of the structuring function g_i is reversed. Similarly, based on the idea of a systolic array, a pipelineable unit, which we call an erosion unit, is derived. The internal structure of such a unit is shown in Figure 4 (b).

As seen from Figure 4 (b), the erosion unit is composed of four single-bit shift registers, one adder and one comparator. The cycle time in this case is the same as that for the dilation unit, i.e., $T_c^e = T_a$. The corresponding latency, $T_l^e = 2(N+1)T_c^e$. Note that latency for the erosion unit is always greater than that for the dilation unit. This follows, since erosion, being a shrinking operation, will require a longer delay for the appearance of the first output data (of the shrunken object).

Corresponding to the dilation situation, a binary erosion unit can be constructed using the configuration of Figure 4 (b) in which the subtractor is replaced by an EXCLUSIVE-NOR gate and the comparator (minimum) is replaced by an AND gate. The internal structure of the resulting binary erosion unit is shown in Figure 5 (b). In this case, hardware cost is drastically reduced, while the cycle time is also reduced to one gate propagation delay, i.e., $T_c^e = T_{pd}$.

4 SOME IMPLEMENTATION EXAMPLES

In this Section, some implementation examples which utilize the proposed dual architectures, are presented. Since in image processing, the structuring elements are usually 2-D, some means of decomposing a complex structuring element into a few simpler structuring elements will help reduce hardware complexity, as well as implementation cost. The following properties of mathematical morphology are particularly useful in creating efficient implementations.

Parallel Decomposition:

$$X \oplus (B_1 \cup \cdots \cup B_M) = (X \oplus B_1) \cup \cdots \cup (X \oplus B_M) \tag{17}$$

$$X \ominus (B_1 \cup \cdots \cup B_M) = (X \ominus B_1) \cap \cdots \cap (X \ominus B_M) \tag{18}$$

Serial Decomposition:

$$X \oplus (B_1 \oplus \cdots \oplus B_M) = (\cdots((X \oplus B_1) \oplus B_2) \cdots \oplus B_M) \tag{19}$$

$$X \ominus (B_1 \oplus \cdots \oplus B_M) = (\cdots((X \ominus B_1) \ominus B_2) \cdots \ominus B_M) \tag{20}$$

Note that the equations above also apply when X and B are gray-scale signals. Note, as well, that special case of the above decompositions is to restrict the structuring element to one-dimension (vertical or horizontal), in which case the proposed architectures are very efficient for 2-D processing.

Next, we will examine the implementation of two morphological operations which find application in computer vision. The one is the morphological skeleton transform. A skeleton of a binary object consists of the loci of the centres of the maximal inscribable disks in the object. The skeleton $SK(X)$ of a binary image X is defined as the union of the loci $S_m(X)$, $m = 1, \cdots, M$ of the maximal inscribable disks mB of radius m [3]. The formula for $SK(X)$ and $S_m(X)$ are given as follows:

$$SK(X) = \bigcup_{m=0}^{M} S_m(X) = \bigcup_{m=0}^{M} (X \ominus mB) \setminus [(X \ominus mB) \circ B] \tag{21}$$

where $M = \max \{ m : X \ominus mB \neq \emptyset \}$ and $\setminus$ denotes set difference.

A structure for implementing the algorithm above is depicted in Figure 8. Each of the dilation and erosion modules of Figure 8 can be implemented using various combinations of the respective dilation and erosion units presented in Section 3. Of course, the types of combination depend on the shape of the structuring element. A more efficient architecture can be achieved if the structuring element is decomposable (i.e., via (17)-(20)). Since $X \ominus mB$ is always larger than $(X \ominus mB) \circ B$ in (21), the set difference can be implemented using EXCLUSIVE-OR gates provided that proper synchronization is maintained.

The second example is the implementation of pecstrum. The pecstrum (or pattern spectrum) of a binary image is given by the following equation [3,4]:

$$P(m) = Mes\,[X \circ mB\,] - Mes\,[X \circ (m+1)B] \quad m = 0,1,\cdots,M \tag{22}$$

where $M = \min \{m : X \circ mB = \emptyset \}$, and *Mes* denotes the measure, which is assumed to be the area in this case. Since $X \circ mB$ is always larger than $X \circ (m+1)B$, the subtraction can be replaced by set difference if proper synchronization is maintained. In that case, (22) becomes

$$P(m) = Mes\,[X \circ mB \setminus X \circ (m+1)B] \quad m = 0,1,\cdots,M \tag{23}$$

The implementation described by (23) is shown in Figure 9. The set difference can be implemented by a simple EXCLUSIVE-OR gate as in the skeleton transform. Each of the erosion and dilation modules in Figure 9 can be implemented using different combinations of the proposed binary erosion and dilation units. If the structuring element, B, is a line, the structure of the module becomes a simple cascade of the appropriate units. As seen from Figure 9, the value of the pecstrum, $P(m)$, is given by the output of the corresponding counter. The delay elements used around the EXCLUSIVE-OR gate are for synchronization purpose. The number N_m depends on the shape of the structuring element. For example, if a line structuring element of length N is used, $N_m = (M-m)[T_l^d + T_l^e] = (M-m)[3N+5]$. In general, the discrete pecstrum calculation requires that only a small number of openings be performed before one of them reduces the measure to zero.

5 SUMMARY

A modular high-speed architecture has been proposed here for efficient VLSI implementation of morphological operations. The two standard building components, namely the dilation and erosion units, can be used to realize any required morphological transformation. The advantage of this approach is that both gray-scale and binary signals can be handled in identical structures. In comparison to the approach based on threshold decomposition [12], this is much simpler, requires less hardware and is, thereby, more cost effective. Furthermore, due to their modularity, corresponding architecture can be utilized directly to design any morphological-related function. This fact was illustrated by the two implementation examples.

REFERENCES

[1] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[2] R.M. Haralick, S.R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. PAMI*, Vol. 9, No. 4, July 1987, pp. 532-550.

[3] J.F. Bronskill and A.N. Venetsanopoulos, "Multidimensional shape description and recognition using mathematical morphology," *Journal of Intelligent and Robotic Systems*, Vol. 1, 1988, pp. 117-143.

[4] P. Maragos, "Pattern spectrum and multiscale shape representation," *IEEE Trans. PAMI*, Vol. 11, No. 7, July 1989, pp. 701-716.

[5] P. Maragos and R.W. Schafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. ASSP*, Vol. 34, No. 5, Oct 1986.

[6] A.C.P. Loui, A.N. Venetsanopoulos and K.C. Smith, "Two-dimensional shape representation using morphological correlation functions," *Proc. ICASSP*, Albuquerque, New Mexico, April 3-6, 1990, accepted for publication.

[7] M.J. Kimmel, R.S. Jaffe, J.R. Manderville and M.A. Lavin, "MITE: Morphic image transform engine, an architecture for reconfigurable pipelines of neighborhood processors," in *Proc. IEEE Comp. Soc. Workshop Computer Architecture for Pattern Analysis and Image Database Management*, Miami Beach, FL, Nov. 18-20, 1985, pp. 493-500.

[8] M. Duff, "Parallel processors for digital image processing," in *Advances in Digital Image Processing*, P. Stucki, Ed., New York, Plenum, pp. 265-279, 1979.

[9] S. Wilson, "The Pixie-5000 - A systolic array processor," in *Proc. IEEE Comp. Soc. Workshop Computer Architecture for Pattern Analysis and Image Database Management*, Miami Beach, FL, Nov. 18-20, 1985, pp. 477-483.

[10] R.M. Lougheed, D.L. McCubbrey and S.R. Sternberg, "Cytocomputer: Architectures for parallel image processing," in *Proc. of the Workshop on Picture Data Description and Management*, Asilomar, California, Aug. 27-28, 1980, pp. 281-286.

[11] F.A. Gerritsen and P.W. Verbeek, "Implementation of cellular logic operators using 3×3 convolution and table lookup hardware," *Computer Vision, Graphics, Image Processing*, vol. 27, 1984, pp. 115-123.

[12] F.Y.C. Shih and O.R. Mitchell, "Threshold decomposition of gray scale morphology into binary morphology," *IEEE Trans. PAMI*, Vol. 11, No. 1, Jan. 1989, pp. 31-42.

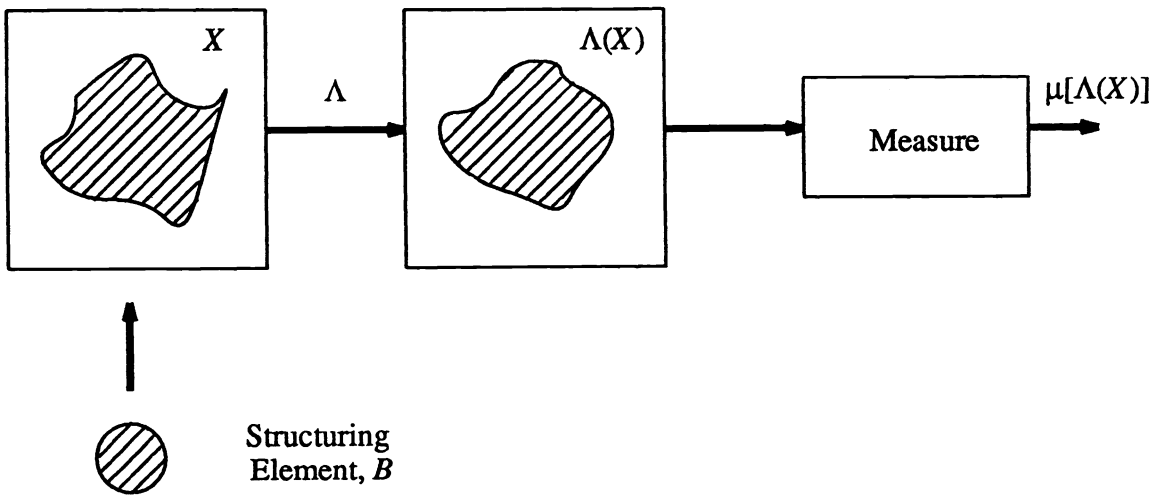


Figure 1 The methodology of mathematical morphology

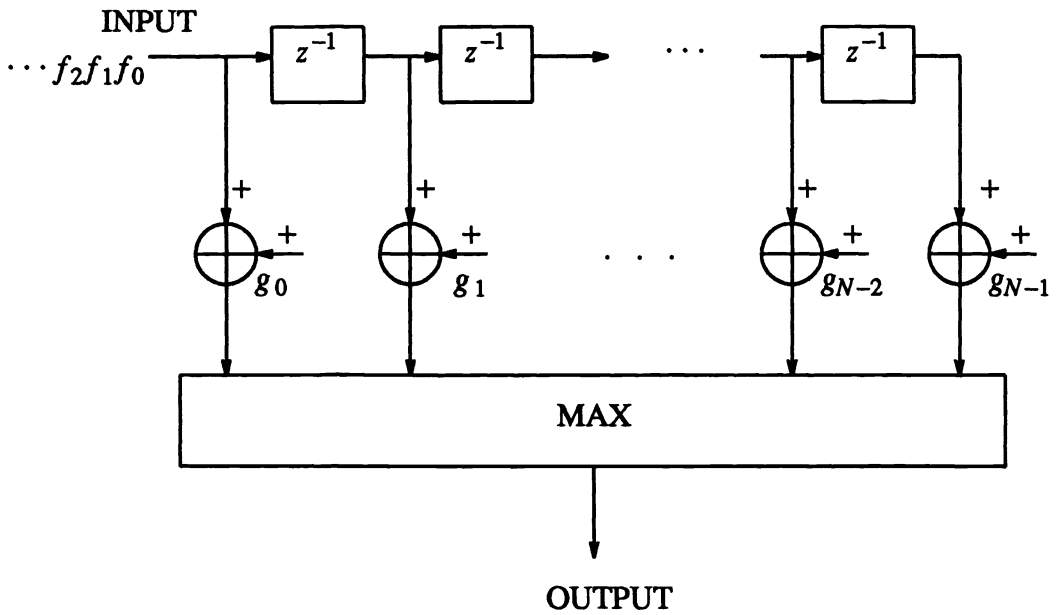


Figure 2 A direct-form implementation of gray-scale dilation

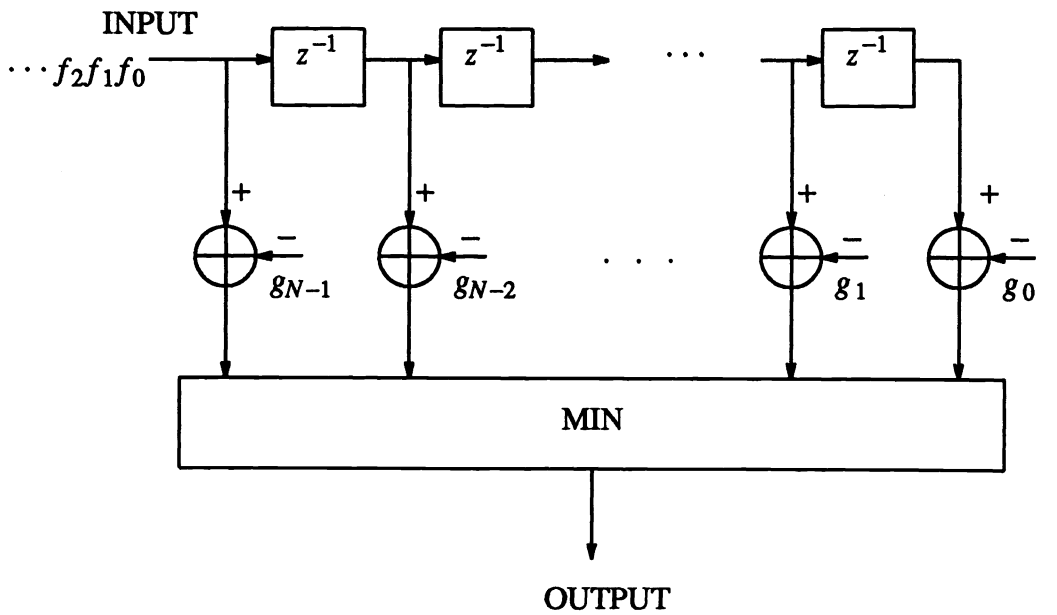


Figure 3 A direct form implementation of gray-scale erosion

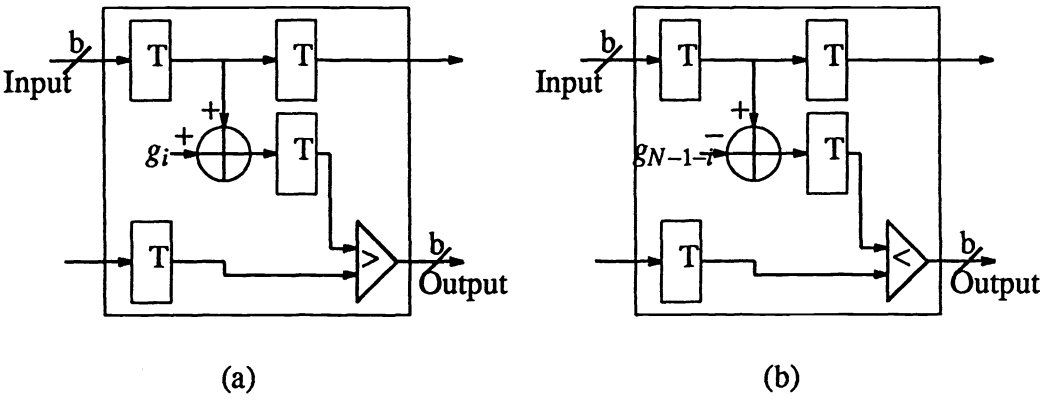


Figure 4 (a) A gray-scale dilation unit (b) A gray-scale erosion unit

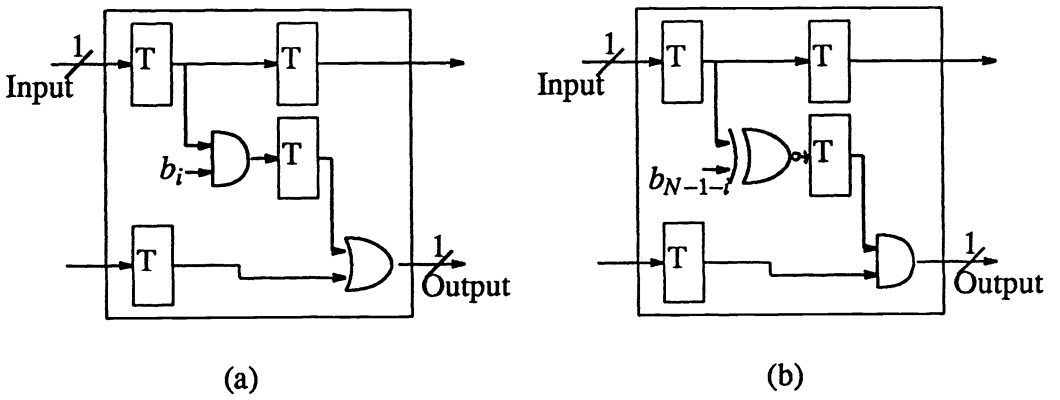


Figure 5 (a) A binary dilation unit (b) A binary erosion unit

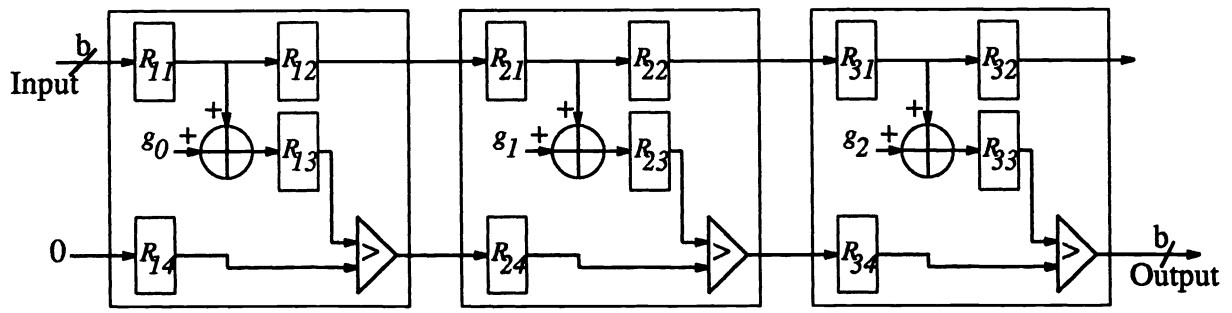


Figure 6 An implementation of a 1x3 structuring function using a pipeline of dilation units

Input	R_{11}	R_{12}	R_{13}	R_{14}	R_{21}	R_{22}	R_{23}	R_{24}	R_{31}	R_{32}	R_{33}	R_{34}	Output
f_0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_1	f_0	0	0	0	0	0	0	0	0	0	0	0	0
f_2	f_1	f_0	f_0+g_0	0	0	0	0	0	0	0	0	0	0
f_3	f_2	f_1	f_1+g_0	0	f_0	0	0	0	0	0	0	0	0
f_4	f_3	f_2	f_2+g_0	0	f_1	f_0	f_0+g_1	f_1+g_0	0	0	0	f_0+g_0	0
f_5	f_4	f_3	f_3+g_0	0	f_2	f_1	f_1+g_1	f_2+g_0	f_0	0	0	MAX1	f_0+g_0
f_6	f_5	f_4	f_4+g_0	0	f_3	f_2	f_2+g_1	f_3+g_0	f_1	f_0	f_0+g_2	MAX2	MAX3
f_7	f_6	f_5	f_5+g_0	0	f_4	f_3	f_3+g_1	f_4+g_0	f_2	f_1	f_1+g_2	MAX4	MAX5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

where

$$\text{MAX1} = \max [f_0+g_1 , f_1+g_0]$$

$$\text{MAX2} = \max [f_2+g_0 , f_1+g_1]$$

$$\text{MAX3} = \text{MAX1}$$

$$\text{MAX4} = \max [f_2+g_1 , f_3+g_0]$$

$$\text{MAX5} = \max [\text{MAX2} , f_0+g_2]$$

Figure 7 Space-time diagram for gray-scale dilation using the structure of Figure 6.

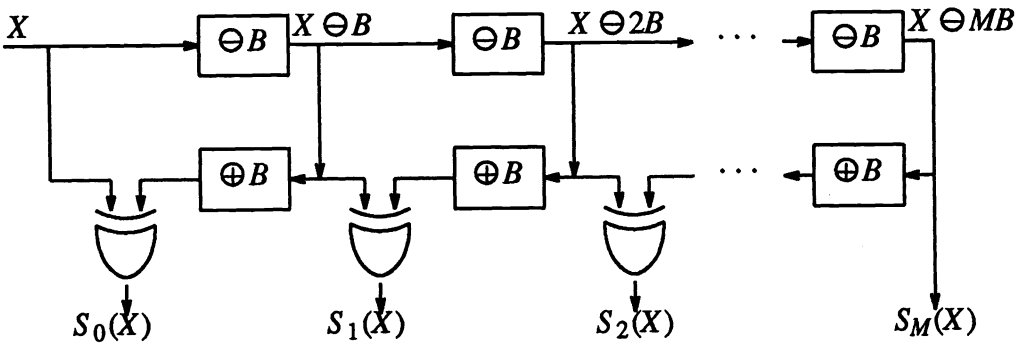
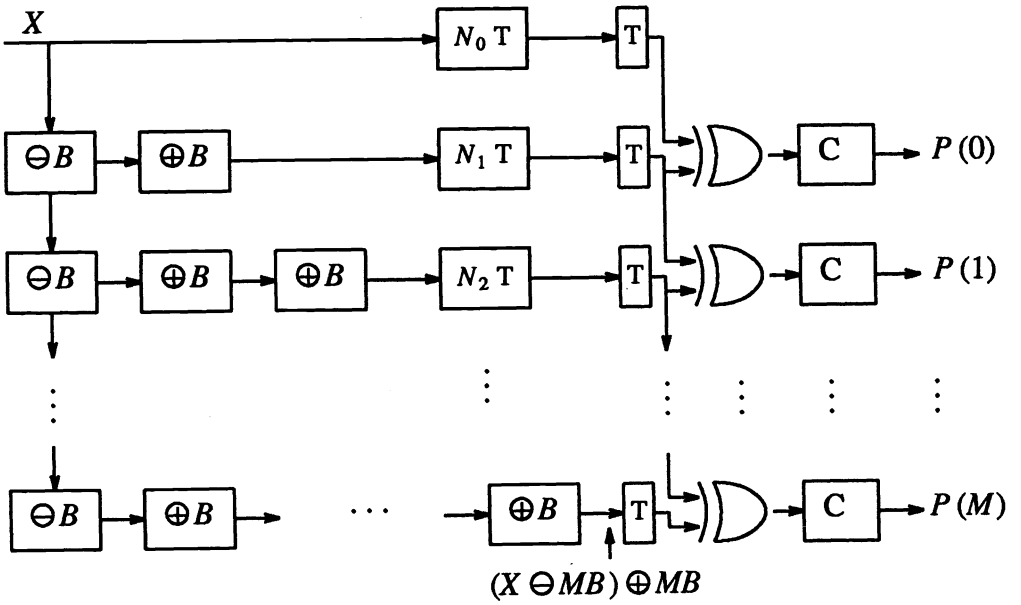


Figure 8 Implementation of morphological skeleton transforms



T - delay element
C - counter

Figure 9 An implementation of pecstrum