Flexible Architectures for Morphological Image Processing and Analysis

Alexander C. P. Loui *Member, IEEE,* A. N. Venetsanopoulos, *Fellow, IEEE,* and K. C. Smith, *Fellow, IEEE*

Abstract—**This paper presents an architecture for the efficient and high-speed realization of morphological filters. Since morphological filtering can be described in terms of erosion and dilation, two basic building units performing these functions are required for the realization of any morphological filter. Dual architectures for erosion and dilation are proposed and their operations are described. Their structure, similar to the systolic array architecture as used in the implementation of linear digital filters, is highly modular and suitable for efficient very-large-scale integration (VLSI) implementation. A decomposition scheme is proposed to facilitate the implementation of two-dimensional morphological filters based on one-dimensional structuring elements constructed using the dual architectures. The proposed architectures, which also allow the processing of gray-scale images, are appropriate for applications where speed, size, and cost are of critical significance.**

I. Introduction

THE major requirement on useful image-analysis or machine-vision systems is that they be both time and cost effective. Accordingly, an important parameter in dynamic-vision design is the throughput rate R or the corresponding cycle time $T_c = 1/R$. One related issue, then, is the notion of real-time capability of the vision system. There are many definitions of "real-time" processing for vision. One point of view is that: "A real-time system should take at most as much time as a human system" [1]. Another definition of "real-time image processing" is that: "The processing of images be done at a speed, such that the data rate of the processed images is the same as that of the input images" [2]. In terms of the North American TV standard, this is equivalent to the capability of processing image data at the TV-camera rate of 30 frames/s or about 10 Mpixels/s. However, this is a very demanding constraint for a dynamic vision system that usually includes not only the task of image processing but also the high-level tasks of image analysis and image understanding.

There already exist a variety of machines that are capable of performing high-speed morphological or cellular-logic operations [3]–[10]. In general, these machines can be divided into two main classes: The first basically comprises two-dimensional (2-D) array processors that operate on an entire image or subimage in a set of parallel processes. Examples of this type of machine are CLIP [3], MPP [4], and PIXIE-5000 [5]. In general, all of these machines can also be considered as being the single instruction multiple data (SIMD) type. The main drawback of 2-D array processors is their cost. For example, a 512×512 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.

The second class of machines—local-window processors—scan an image and perform operations on a small neighborhood window. Examples of such machines include MITE [6], PIPE [7], Cytocomputer [8], a structure based on convolution and table-lookup [9], and the threshold-decomposition realization [10]. Note that, with this type of processor, an increase in image size requires a quadratic increase in processor speed in order to maintain a constant frame rate.

Most of the above local-window processors are general purpose in nature in that they are programmable. The only exception is the architecture based on threshold decomposition [10]. In particular, the Cytocomputer consists of a serial pipeline of programmable neighborhood processing stages, in which a single neighborhood transformation is performed in each stage of the pipeline [8]. The throughput of the Cytocomputer is about 1.6 million 8-b pixels/s and programming requires about 1 ms/stage [8]. Although these general-purpose cellular machines are flexible because of their programmability, they are relatively slow and they do 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 appears inevitable.

In this paper, a VLSI architecture based on the idea of systolic arrays [11], [12] is proposed for high-speed morphological image processing. This architecture, which consists of two basic building units, does not depend on the image structure or size. Rather, it is related 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. Furthermore, it is capable of handling binary as well as gray-scale operations.

Section II gives the mathematical background of the basic morphological transformations. Section III introduces a novel

Manuscript received October 29, 1990; revised October 1, 1991. A. C. P. Loui was with the Department of Electrical Engineering, University of Toronto, Toronto, Ontario M5S 1A4, Canada. He is now with Bell Communications Research, Red Bank, NJ 07701. This paper was recommended by Associate Editor Sarah Rajala.

A. N. Venetsanopoulos and K. C. Smith are with the Department of Electrical Engineering, University of Toronto, Toronto, Ontario M5S 1A4, Canada.

IEEE Log Number 9107417.

architecture, which we call the *nonlinear pipeline processor* (NPP), for high-speed morphological transformations. The two basic building units of the NPP—the dilation unit and the erosion unit—are presented and their operations are described. This is followed by an analysis in terms of hardware complexity, cycle time, and latency. In Section IV, we introduce a geometrical decomposition procedure. This procedure allows any 2-D structuring element to be decomposed into different geometrical combinations of simple 1-D structuring elements for more efficient implementation. Then a comparison with the architecture based on threshold decomposition [10] is given. Section V presents some exemplary implementations using the proposed architecture of morphological image-analysis schemes including the morphological skeleton transform, the pattern spectrum, and the geometrical correlation functions. Finally, concluding remarks are provided in Section VI.

II. Basic Morphological Transformations

Mathematical morphology provides a very effective tool for extracting structural information from image signals. In brief, there are two basic steps in an image analysis process based on mathematical morphology: a geometrical transformation and then a measurement. Let X denote the image signal under study. According to the procedure indicated above, any morphological operation consists of first a transformation Λ (by a preselected structuring element B) from one domain to another, followed by a measure μ [13] (see Fig. 1). Examples of transformations Λ are opening, boundary extraction, and skeletonization. The measurement $\mu[\Lambda(X)]$ is a number that can be a quantity representing weight, area, 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.

The four most basic morphological transformations are dilation, erosion, opening and closing. Binary morphological transformations apply to sets of any dimension, whether they constitute a Euclidean n-space E^n or its discrete or digitized equivalent, an integer n-space Z^n. For simplicity's sake, E^n is 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 by Haralick *et al.* [14], which are slightly different from those by Serra [13].

A. Dilation and Erosion

Let us begin with by definiting binary dilation: Binary dilation is the morphological transformation that combines two sets using the vector sums of set elements.

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

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

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

$$X \oplus B = \{z \in E^n \mid z = x + b \text{ for } x \in X \text{ and } b \in B\}. \quad (2)$$

According to the above definitions, we may also write

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

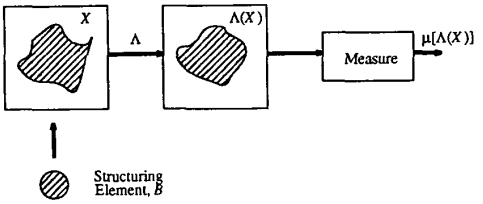


Fig. 1. The methodology of mathematical morphology.

Erosion is the morphological dual of dilation (or its complementary operation). 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 3: 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

$$X \ominus B = \{z \in E^n \mid z + b \in X \text{ for } b \in B\}. \quad (4)$$

Similarly, erosion can also be defined as

$$X \ominus B = \bigcap_{b \in B} (X)_{-b}. \quad (5)$$

Gray-scale dilation and erosion are defined in a rather different way than their binary counterparts. Let us begin by definiting gray-scale dilation by introducing 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 converted into a maximum operation.

Definition 4: Let $X \subseteq E^n$ 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 [14]

$$T[X](x) = \max\{y \mid (x, y) \in X\}. \quad (6)$$

Definition 5: 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 [14]

$$U[f] = \{(x, y) \in F \times E \mid y \le f(x)\}. \quad (7)$$

This implies that the umbra of a function f is a set consisting of the surface f and everything below the surface. Hence, a function can be considered as the top of its own 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 6: 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

$$f \oplus g = T[U[f] \oplus U[g]]. \quad (8)$$

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 of dilation. 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

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

B. Opening and Closing

In many applications, erosion and dilation are usually used in sequence. These types of iterative operations are known to have the characteristic of preserving global geometric structures of the unsuppressed features. That is, 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.,

Definition 8: The opening of an image X by a structuring element B is denoted by $X \circ B$ and is defined by

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

Geometrically, the opening smooths the contours of X, cutting the narrow isthmuses, and suppressing the small islands and the sharp capes of X. 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.,

Definition 9: The closing of an image X by a structuring element B is denoted by $X \bullet B$ and is defined by

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

By duality, closing blocks up narrow channels, small holes, and thin gulfs of X. One important property of morphological opening and closing is that they are idempotent, meaning that successive openings or closings by the same structuring element do not alter the result after the first application.

Gray-scale opening and closing are defined in an analogous way to opening and closing in the binary case. Specifically, the gray-scale opening of f by the structuring function g, $f \circ g$, is defined as the eroding of f by g and then the dilating of the result by g, i.e.,

$$f \circ g = (f \ominus g) \oplus g. \tag{12}$$

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

$$f \bullet g = (f \oplus g) \ominus g. \tag{13}$$

Note that the idempotent property of binary morphological opening and closing also holds for the gray-scale case.

III. Nonlinear Pipeline Architectures

In this section, a systolic-array-based architecture, called NPP, as defined earlier, is proposed for performing morphological operations. Since the basic operations in mathematical morphology are dilation and erosion, it is necessary only to construct two basic building blocks—the dilation unit and the erosion unit. Thereafter, any other morphological operation can be implemented using these two basic building blocks plus other appropriate logic gates. One advantage of this approach is that maximum utilization of processing elements is attained through reuse inherent in the pipelined nature of the architecture. Another advantage of the proposed architecture is that it is well suited for VLSI implementation.

A. Gray-Scale Architectures

In the previous section, 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 that will lead to more practical structures.

From Section II, it was shown that gray-scale dilation ((8)) 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, basically this is equivalent to performing a maximum operation. Specifically, if we apply the definitions of binary dilation (3) and umbra (7) in (8), after some algebraic manipulation, it can be shown that gray-scale dilation is equivalent to taking the maximum of a set of sums [14].

Proposition 10: Let f: $F \to E$ and g: $G \to E$. Then $f \oplus g$: $F \oplus G \to E$ can be computed by

$$(f \oplus g)(x) = \max\{f(x - z) + g(z)\}$$

$$\forall\, z \in G,\ x - z \in F. \tag{14}$$

Proposition 10 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 structure similar to that employed in linear digital filtering. The exception is that in this case the operations involved are shifting, addition, and maximum, instead of the operations of conventional convolution, i.e., shifting, multiplication, and addition as used in linear digital filtering.

The direct-form structure described by (14) is shown in Fig. 2(a). However this structure is still not practical for VLSI implementation due to its inherent reliance on nonlocal data transfers. Hence, in the process of creation of the architectures to be described here, it was conjectured that by localizing the data transfers, a more practical structure could be created. One result, 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 Fig. 3(a). A cascade of identical dilation units can be used to implement a dilation operation.

As seen from Fig. 3(a), the dilation unit is composed basically of four single-stage b-b 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

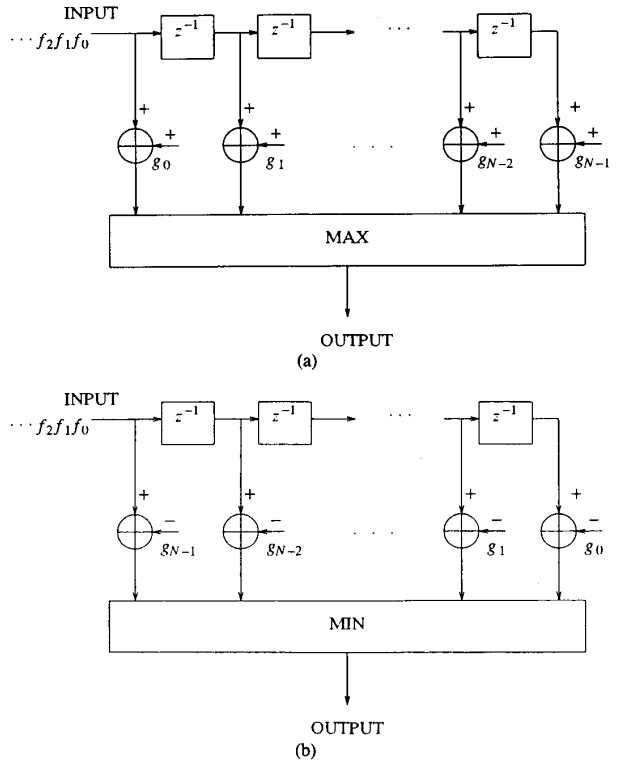


Fig. 2. (a) A direct-form implementation of gray-scale dilation. (b) A direct-form implementation of gray-scale erosion.

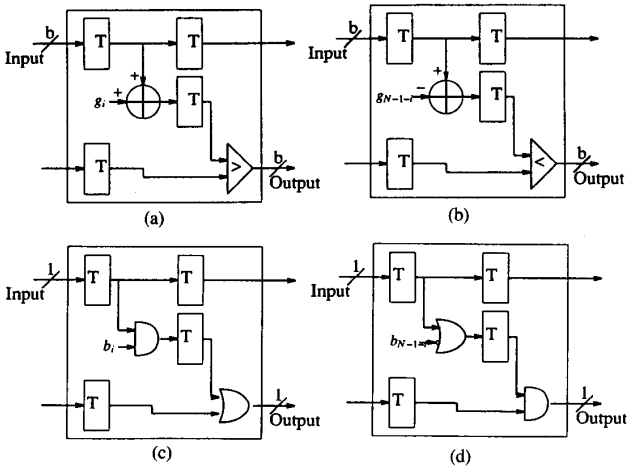


Fig. 3. (a) A gray-scale dilation unit. (b) A gray-scale erosion unit. (c) A binary dilation unit. (d) A binary erosion unit.

elements. In this case, it is equal to one addition time, or, more precisely, to one comparison time, since the comparator is implemented using an adder plus other logic. Hence, the cycle time, $T_c^d = T_a + T_{pd}$, where T_a is the time for one addition, and T_{pd} is the propagation delay of a logic gate. This implies that very high throughput rates can be attained after an initial latency period required to fill up the pipeline. Furthermore, it is easily shown that the latency of this pipeline architecture is given by $T_l^d = (N + 3)T_c$, where N is the length of the structuring function.

The block diagram of an exemplary system, which implements a 1×3 structuring function for gray-scale dilation, is depicted in Fig. 4. The operations of the associated morphological transformation are described in the space-time diagram of Table I. It is assumed that both the input signal f and the structuring function g are represented in two's-complement code, and are bounded between -0.5 and 0.5. That is, $-0.5 \le f < 0.5$ and $-0.5 \le g < 0.5$. Note that unlike a similar linear structure proposed in [15], the proposed nonlinear structure will not produce the correct initial output by simply using zero initial conditions and setting the registers to their zero state. The main difficulty is to ensure that the initial output samples, such as $f_0 + g_0$ (see Table I), are propagated to the output register without alteration by the comparators. For example, if the initial value of the register R_{23}, X in Table I (row 4, column 8), is larger than $f_0 + g_0$, then the correct value $f_0 + g_0$ will not be propagated to the next stage and an erroneous output will result. Note that this problem is an initial transient and does not exist once the NPP is totally filled.

One solution is to ensure that the initial value of all the X's is less than that of the correct comparand. This can be done (for example) by setting the initial state of all the registers to -1 and restricting the signal range between -0.5 and 0.5. In this situation, the maximum value of X is -0.5, which is the minimum value of the signal range. The only problem with this solution is that overflow may occur if a component of the structuring function g_i is negative. This, however, can be circumvented by detecting the occurrence of overflow, and using the overflow information to set the output of the appropriate registers (e.g., R_{13}, R_{23}, and R_{33}) to their minimum value. The additional hardware required to handle this problem is fairly simple and involves only the use of a few logic gates. Note also that because of this nonlinear characteristic of the NPP, an extra bit is needed to accommodate the initial conditions of $+1$ and -1 for the dilation and erosion units respectively. In other words, for signal levels equal to 2^{b-1}, a wordlength of b-b is required for the hardware components.

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 for dilation, this is created by applying the definitions of binary erosion and umbra to (9). After some algebraic manipulation, it can be shown that gray-scale erosion is equivalent to taking a minimum of a set of differences. In fact, we can think of erosion as equivalent to correlation, where the summation operation is replaced by a minimum operation, and multiplication becomes subtraction.

The corresponding direct-form representation for gray-scale erosion is given by the following proposition [14]:

Proposition 11: Let f: $F \to E$ and g: $G \to E$. Then $f \ominus g$: $F \ominus G \to E$ can be computed by

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

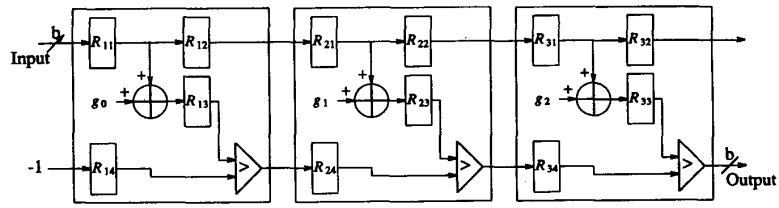


Fig. 4. An NPP implementation of a 1 × 3 structuring function for gray-scale dilation.

TABLE I
A SPACE-TIME DIAGRAM FOR GRAY-SCALE DILATION USING THE NPP OF FIG. 4

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	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
f_1	f_0	-1	X	-1	-1	-1	X	-1	-1	-1	X	-1	-1
f_2	f_1	f_0	$f_0 + g_0$	-1	-1	-1	X	X	-1	-1	X	X	X
f_3	f_2	f_1	$f_1 + g_0$	-1	f_0	-1	X	$f_0 + g_0$	-1	-1	X	X	X
f_4	f_3	f_2	$f_2 + g_0$	-1	f_1	f_0	$f_0 + g_1$	$f_1 + g_0$	-1	-1	X	$f_0 + g_0$	X
f_5	f_4	f_3	$f_3 + g_0$	-1	f_2	f_1	$f_1 + g_1$	$f_2 + g_0$	f_0	-1	X	MAX 1	$f_0 + g_0$
f_6	f_5	f_4	$f_4 + g_0$	-1	f_3	f_2	$f_2 + g_1$	$f_3 + g_0$	f_1	f_0	$f_0 + g_2$	MAX 2	MAX 3
f_7	f_6	f_5	$f_5 + g_0$	-1	f_4	f_3	$f_3 + g_1$	$f_4 + g_0$	f_2	f_1	$f_1 + g_2$	MAX 4	MAX 5
f_8	f_7	f_6	$f_6 + g_0$	-1	f_5	f_4	$f_4 + g_1$	$f_5 + g_0$	f_3	f_2	$f_2 + g_2$	MAX 6	MAX 7
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

where
MAX 1 = max $[f_0 + g_1, f_1 + g_0]$
MAX 2 = max $[f_2 + g_0, f_1 + g_1]$
MAX 3 = MAX 1
MAX 4 = max $[f_2 + g_1, f_3 + g_0]$
MAX 5 = max [MAX 2, $f_0 + g_2$]
MAX 6 = max $[f_3 + g_1, f_4 + g_0]$
MAX 7 = max [MAX 4, $f_1 + g_2$]
X = "don't care" condition.

The block diagram of the direct form implementation of erosion is depicted in Fig. 2(b). Note that this is very similar to the one for dilation wherein 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 systolic-array idea, a pipelinable unit, which we call an erosion unit, is derived. The internal structure of such a unit is shown in Fig. 3(b).

As Fig. 3(b) shows, the erosion unit is composed of four single-stage b-b shift registers, one adder, and one comparator. The cycle time in this case is the same as that for the dilation unit, i.e., $T_c^e = T_a + T_{pd}$. The corresponding latency is $T_f = 2(N + 1)T_c^e$. 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).

Similarly, the block diagram of an exemplary system that implements a 1 × 3 structuring function for gray-scale erosion, is shown in Fig. 5. The operations of this erosion transformation are described in the space-time diagram of Table II. In this case, the initial state of all the registers are set to +1. Again, this is done to ensure that the initial samples ($f_0 - g_2$ and MIN 3), which appear at the output port before the NPP is filled, are correct. Also, a b-b wordlength is required for the hardware components to achieve 2^{b-1} signal levels.

B. Binary Architectures

The precision of the dilation unit is b b. If b is equal to 1, the structure should degenerate into a binary dilation unit. However, this is not the case here because of the nonlinear character of morphological operations. Unlike linear-filter structures, where superposition holds at the bit level, gray-scale morphological operations do not obey this law. Instead, they satisfy a different superposition property called threshold decomposition [16], [17]. Specifically, $f^i(m)$, the threshold decomposition of a 2^b levels sequence of length L, is the set of binary sequences

$$f^i(m) = \begin{cases} 1, & \text{if } f(m) \geq i \\ 0, & \text{if } f(m) < i \end{cases} \tag{16}$$

where $1 \leq m \leq L$ and $1 \leq i \leq 2^b - 1$. This property allows gray-scale signals to be decomposed into multiple binary signals. Since the gray-scale signals in our case are not threshold decomposed, it is not surprising that the degenerated structure does not correspond to a binary structure. However, we show in Section IV that this approach is more efficient than the threshold-decomposition realization of [10] for gray-scale morphology, especially for cases when the wordlength b is large.

Although the architecture of Fig. 3(a), (b) does not apply directly when $b = 1$, a binary structure can be obtained by simply replacing the adder by an AND gate and the compara-

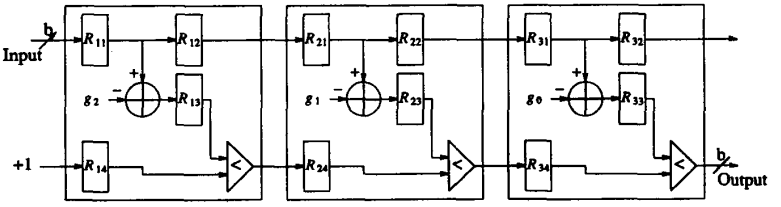


Fig. 5. An NPP implementation of a 1 × 3 structuring function for gray-scale erosion.

TABLE II
A SPACE-TIME DIAGRAM FOR GRAY-SCALE EROSION USING THE NPP OF FIG. 5

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	1	1	1	1	1	1	1	1	1	1	1	1	1
f_1	f_0	1	X	1	1	1	X	1	1	1	X	1	1
f_2	f_1	f_0	$f_0 - g_2$	1	1	1	X	X	1	1	X	X	X
f_3	f_2	f_1	$f_1 - g_2$	1	f_0	1	X	$f_0 - g_2$	1	1	X	X	X
f_4	f_3	f_2	$f_2 - g_2$	1	f_1	f_0	$f_0 - g_1$	$f_1 - g_2$	1	1	X	X	X
f_5	f_4	f_3	$f_3 - g_2$	1	f_2	f_1	$f_1 - g_1$	$f_2 - g_2$	f_0	1	X	X	X
f_6	f_5	f_4	$f_4 - g_2$	1	f_3	f_2	$f_2 - g_1$	$f_3 - g_2$	f_1	f_0	$f_0 - g_0$	MIN 1	X
f_7	f_6	f_5	$f_5 - g_2$	1	f_4	f_3	$f_3 - g_1$	$f_4 - g_2$	f_2	f_1	$f_1 - g_0$	MIN 2	MIN 3
f_8	f_7	f_6	$f_6 - g_2$	1	f_5	f_4	$f_4 - g_1$	$f_5 - g_2$	f_3	f_2	$f_2 - g_0$	MIN 4	MIN 5
f_9	f_8	f_7	$f_7 - g_2$	1	f_6	f_4	$f_5 - g_1$	$f_6 - g_2$	f_4	f_3	$f_3 - g_0$	MIN 6	MIN 7
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

where
MIN 1 = min $[f_1 - g_1, f_2 - g_2]$
MIN 2 = min $[f_2 - g_1, f_3 - g_2]$
MIN 3 = min [MIN 1, $f_0 - g_0$]
MIN 4 = min $[f_3 - g_1, f_4 - g_2]$
MIN 5 = min [MIN 2, $f_1 - g_0$]
MIN 6 = min $[f_4 - g_1, f_5 - g_2]$
MIN 7 = min [MIN 4, $f_2 - g_0$]
X = "don't care" condition.

tor by an OR gate. The resulting arrangement becomes a binary equivalent of the gray-scale dilation unit. We call this a binary-dilation unit and its internal structure is depicted in Fig. 3(c). In this case, the cycle time reduces to simply one gate delay, i.e., $T_c^d = T_{pd}$, where T_{pd} is the propagation delay of a single logic gate.

Corresponding to the dilation situation, a binary-erosion unit can be constructed based on the configuration of Fig. 3(b) in which the subtractor is replaced by an OR gate with a complement input for the corresponding component of the structuring element, and the comparator (minimum) is replaced by an AND gate. The internal structure of the resulting binary erosion unit is shown in Fig. 3(d). Special attention should be given here to whether the zeros of the structuring element imply a "don't care" condition or not. If zero does not imply a "don't care" condition, then the erosion units should be modified. The modification can be done by simply replacing the OR gate with an XNOR gate. In either case, hardware cost is drastically reduced, while the cycle time is also reduced to one gate propagation delay, i.e., $T_c^e = T_{pd}$.

C. Analysis

In this section, the hardware complexity and the throughput analysis of the proposed architecture are presented. Specifically, a cost function is developed for making quantitative comparisons between different digital implementations. This cost function provides an efficient tool allowing different implementations of the same algorithm, and different algorithms for the same function, to be compared. The cost function is chosen to reflect the functional design or logic design independently of the technology used as accurately as possible. This implies that technology issues can be introduced separately in designing a practical system. Certainly the only true comparison is one in which the details of the implementing technology are incorporated.

Let us begin with hardware complexity. Assume that the precisions of both the input and output signals are b b. Since only addition and comparison operations are performed, the internal precision can also remain as b b. Also assuming that full-carry-lookahead adders are used, the typical number of logic gates for such an adder is approximately $12b + 1$ [18]. Hence, for the implementation of a 4-b full-carry-lookahead adder, a total of approximately 49 logic gates is required. A basic magnitude comparator can be constructed using an adder and a complementer [19]. However, since only the carry-out bit of a full adder is required to determine the magnitude of the two comparands, the logic gates related to the sum output can be eliminated. Hence, a simpler comparator can be constructed using a modified full adder and a maximum selector. Specifically, the carry-out signal of the modified adder will be used to control the maximum selector.

The logic gate count for this comparator is estimated as $(12b + 1)(2/3) + b = 9b$ where the factor of $2/3$ accounts for the deletion of the summing gates, and b is the gate count for the maximum selector which can be constructed using a simple pass transistor. The b-b parallel-access latches are assumed to have a gate complexity of $10b$ logic gates [20].

According to the foregoing discussion, the complexity of the proposed dilation or erosion unit is given approximately by $C = 12b + 1 + 9b + 4(10b) = 61b + 1$. Using a 1×9 structuring element with $b = 8$ as an example, the logic-gate count is approximately equal to $(61(8) + 1)(9) = 4401$. Assuming that both the input signals and the structuring functions have the same number of gray-levels, the complexity functions C are shown in Fig. 6 in relation to the size of the 1-D structuring function for different gray levels. Note that the graphs of Fig. 6 apply to both the dilation and the erosion structures since they have basically the same hardware configuration. Also the hardware complexity is directly a linear function of both the precision b and the structuring function size N.

The cycle times of the proposed gray-scale and binary architectures for both dilation and erosion have been determined in the previous section. Specifically, for gray-scale dilation or erosion, $T_c \cong T_a + T_l$ where T_a is the addition time and T_l is the latch delay time. For the corresponding binary units, the cycle time reduces approximately to $T_c = 2T_{pd} + T_l$ where T_{pd} is a logic-gate propagation delay.

If we consider a 1.5-μm CMOS technology, the typical dynamic D latch has a delay of around 4ns including setup ($\sim$ 1.4 ns) and hold times ($\sim$ 0 ns). The propagation delay of a logic gate is approximately 1.5 ns. Hence, if $b = 8$, the logic-gate depth of a carry-look-ahead adder is about 10 [18], this implies an internal throughput rate of $1/(10(1.5\text{ ns}) + 4\text{ ns}) \cong 50$ MHz. For the binary unit, an internal throughput rate of up to $1/(2(1.5\text{ ns}) + 4\text{ ns}) \cong 140$ MHz is possible. These high throughput rates can be attained after an initial latency period required to fill up the pipeline. It has also been shown that the latency of the NPP for dilation is $T_l^d = (N + 3)T_c$, where N is the length of the structuring function. The corresponding latency for the erosion module is $T_l^e = 2(N + 1)T_c$.

The cycle time and the latency for both the dilation and erosion structures, as a function of the size of the structuring function, are shown in Fig. 7. Subscripts c and l stand for cycle time and latency, while the superscripts gd, ge, bd, and be stand for gray-scale dilation, gray-scale erosion, binary dilation, and binary erosion, respectively. Note that the cycle time is independent of the size of the structuring function for both the gray-scale and binary structures. However, the latency is a linear function of the size of the structuring function.

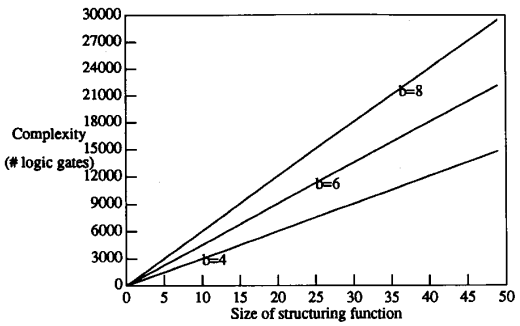


Fig. 6. Complexity of the dilation or erosion unit versus size of 1-D structuring function at a different gray level.

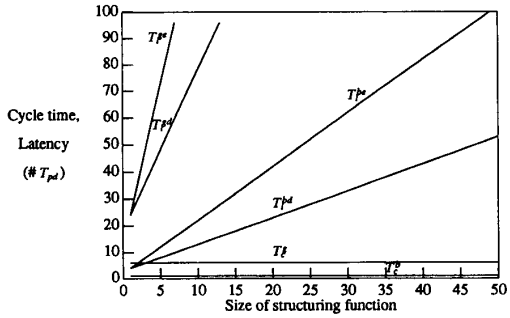


Fig. 7. Cycle time T_c and latency T_l of the dilation and erosion unit versus the size of a 1-D structuring function.

IV. Implementation of 2-D Structuring Elements

In morphological image processing and analysis, the structuring elements are usually two dimensional. In particular, in the implementation of morphological shape descriptors such as the skeleton transform, the pattern spectrum, and the geometrical correlation functions, the requirement is for repeated operations with a structuring element of increasing size or scale. Hence, a means of decomposing a complex structuring element into a few simpler structuring elements will help to reduce hardware complexity, as well as implementation cost. In this section, we describe the implementation of 2-D structuring elements based on the proposed NPP, as well as the issue of efficient decomposition of 2-D structuring elements. Then a comparison is made to the threshold-decomposition realization [10] in terms of hardware complexity for 2-D structuring elements.

A 2-D structuring element can be implemented by using a 2-D extension of the proposed NPP. This 2-D NPP basically comprises a parallel combinations of 1-D NPP's. For example, the structure of a NPP implementation of a 2-D structuring element for dilation is depicted in Fig. 8 where $B_{k,l}$ denotes either a gray-scale or binary dilation unit (see Figs. 3(a) or 3(c)). This structure assumes that all the delayed samples of the input $X_{i,j}$ are available at the input of the individual NPP. This can be realized by using $(K - 1)$ N-stage b-b shift registers where K is the number of parallel 1-D NPP's, and N is the number of samples in a scan line, or, equivalently, the horizontal dimension of the input image. The output of the 2-D NPP dilation is obtained by ORing the outputs from the individual 1-D NPP's. The total processing time required for these logic operations is $\log_2 K$. If K is large, multiple-input OR gates may be used. Alternatively, each of these output OR gates may be pipelined to reduce the effective cycle time.

The structure for the implementation of 2-D erosion is

similar to that of Fig. 8. One difference is that the output OR gates are replaced by AND gates in this case. The corresponding structure is shown in Fig. 9. It should be noted that the erosion units are arranged in an order that is reversed compared to the dilation structure of Fig. 8. This is necessary for the implementation of the 1-D NPP in which the erosion units are pipelined in reverse order. Thus, this is simply the 2-D extension of the 1-D configuration. Furthermore, the boundary of the input image can be handled effectively by simply surrounding the image with fixed-value pixels that may be discarded after processing.

The hardware complexity of the architecture based on the proposed systolic array is on the order of $K \times L$ where K and L are the maximum sizes of the structuring element in the vertical and horizontal directions. When either K or L is large, the hardware size, i.e., the number of erosion and dilation units, increases very rapidly and makes the implementation unattractive. Thus, in order to reduce the size of 2-D morphological filters of large dimension, an approach based on geometrical decomposition of structuring elements is proposed.

The idea of geometrical decomposition is very similar to that of the matrix decomposition used for 2-D linear digital filters [15], [21] and nonlinear digital filters [22]. In that case, the kernel matrix of a 2-D digital filter is decomposed into a finite and converging sum of other block matrices. It is shown that these matrices can always be decomposed into a product of 1-D polynomials, each one of which is a function of one variable only [21]. Thus the 2-D implementation problem is reduced to a set of 1-D problems.

As noted, in morphological image processing, the notion of structuring element is analogous to that of the kernel matrix of 2-D linear digital filtering. Therefore, if a complex structuring element can be decomposed into the Minkowski sum of several much simpler 1-D structuring elements in this case, the 2-D morphological image-processing problem can again be reduced to a 1-D problem. A large number of practical and useful structuring elements can be decomposed geometrically in the Minkowski sense. In this work, we propose the following decomposition structure that can be used to realized any structuring element in the digital space Z^2:

Proposition 12: Suppose that B is a structuring element in Z^2. It is possible to decompose B geometrically into the following form:

$$B = \bigoplus_{j=1}^{N_1} B_{1j} \cup \bigoplus_{j=1}^{N_2} B_{2j} \cup \cdots \cup \bigoplus_{j=1}^{N_m} B_{mj} = \bigcup_{i=1}^{m} \bigoplus_{j=1}^{N_i} B_{ij} \tag{17}$$

where $\cup$ denotes set union and each B_{ij} may be chosen as a 1-D substructuring element. In fact, if B is continuous and convex, it is infinitely decomposable, i.e.,

$$B = \bigcup_{i=1}^{1} \bigoplus_{j=1}^{k} \frac{1}{k} B = \bigoplus_{j=1}^{k} \frac{1}{k} B. \tag{18}$$

Proposition 12 can be thought of as a generalization of the idea of parallel and serial decompositions [13]:

Parallel Decomposition:

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

$$X \ominus (B_1 \cup \cdots \cup B_M) = (X \ominus B_1) \cap \cdots \cap (X \ominus B_M) \tag{19b}$$

Serial Decomposition:

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

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

Note that the equations above also apply when X and B are gray-scale signals. Hence, the decomposition in (19) corresponds to the case of $N_i = 0$, $1 \le i \le m$ in (17), whereas the decomposition in (20) corresponds to the case of $m = 1$ in (17). However, the advantage of using (17) is that it allows the decomposition of nonconvex structuring elements that contain certain substructures that can be decomposed serially via (20).

Finally, the proposed NPP realization is compared to the threshold-decomposition realization in terms of hardware complexity. A criterion based on the number of gates required to implement a $n \times n$ structuring element is used for comparison. The results are tabulated in Tables III and IV for dilation and erosion transformations, respectively. Note that the NPP realization is different from the implementation of [10] where the complexity, as measured by the number of logic gates for gray-scale erosion, is always about twice the number for dilation. It is found that the hardware complexity of the threshold-decomposition approach increases in an exponential fashion for large values of the signal precision b. On the other hand, as we have seen, the hardware complexity of the NPP approach increases only linearly. Nevertheless, for small values of b, the hardware complexity of the threshold-decomposition realization is slightly less than that of the NPP realization. However, it should be noted that the structure based on threshold decomposition is not regular due to the extensive routing required to access data in different internal storage registers (see Fig. 6 of [10]). In addition, if the size of the structuring element changes, the routing has to be changed completely, and additional logic gates, which must be added to the layout, might require extensive rewiring. This, then, does not allow modular expansion of the architecture once a particular size of structuring element is implemented. In the case of the NPP realization, structuring-element-size changes can be handled very easily by simply appending additional units to the original structure, and no rerouting is required.

V. Some Implementation Examples

A. Morphological-Skeleton Transform

Let us begin with the morphological-skeleton transform. A skeleton of a binary object is defined to consist of the loci of

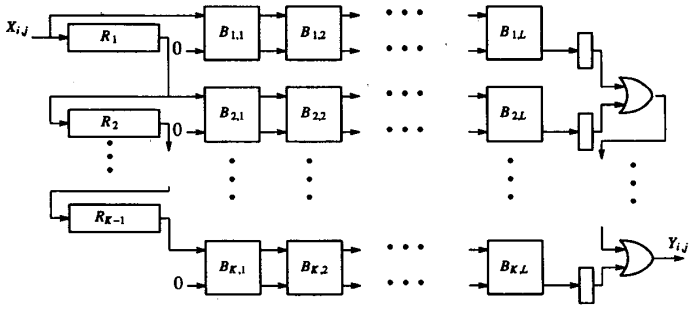


Fig. 8. An NPP implementation of a 2-D structuring element for dilation.

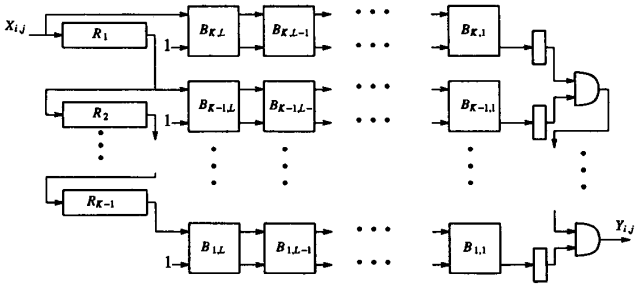


Fig. 9. An NPP implementation of a 2-D structuring function for erosion.

TABLE III
A COMPARISON OF HARDWARE COMPLEXITY FOR GRAY-SCALE DILATION

Approx. No. of Gates	Threshold Decomposition Realization [10]			NPP Realization		
Structuring Element Size	Bits of Gray Level			Bits of Gray Level		
	4	6	8	4	6	8
3 × 3	1214	20.2K	326.7K	2205	3303	4401
4 × 4	2054	34.3K	555.1K	3920	5872	7824
5 × 5	3134	52.5K	848.9K	6125	9175	12.2K

TABLE IV
A COMPARISON OF HARDWARE COMPLEXITY FOR GRAY-SCALE EROSION

Approx. No. of Gates	Threshold Decomposition Realization [10]			NPP Realization		
Structuring Element Size	Bits of Gray Level			Bits of Gray Level		
	4	6	8	4	6	8
3 × 3	2294	38.4K	620.4K	2205	3303	4401
4 × 4	3974	66.6K	1077.4K	3920	5872	7824
5 × 5	6134	102.9K	1664.9K	6125	9175	12.2K

the centres of the maximally 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 maximally inscribable disks mB of radius m [23]. The formula for $SK(X)$ and $S_m(X)$ are given as follows:

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

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

A structure for implementing the algorithm above is depicted in Fig. 10. Each of the dilation and erosion modules of Fig. 10 can be implemented using various combinations of the respective dilation and erosion units presented in Section III. 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 (using (17)). Since $X \ominus mB$ is always larger than $(X \ominus mB) \circ B$ in (21), the set difference can be implemented using XOR gates, provided that proper synchronization is maintained.

B. Pattern Spectrum

The second example is the implementation of pattern spectrum (or pecstrum). The pattern spectrum of a binary image is given by the following equation [24], [25]:

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

where $M = \min\{m: X \circ mB = \varnothing\}$, 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

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

The implementation described by (23) is shown in Fig. 11. The set difference can be implemented by a simple XOR gate as in the skeleton transform. Each of the erosion and dilation modules in Fig. 11 can be implemented using different combinations of the proposed binary erosion and dilation units. If the structuring element B is a "line" (i.e., 1-D),

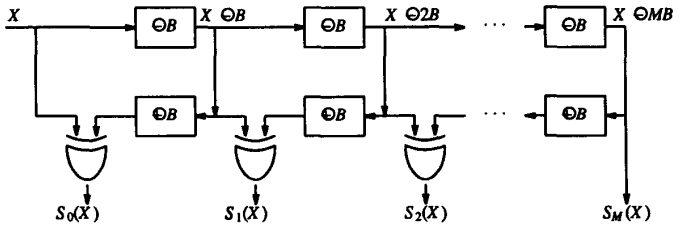


Fig. 10. An implementation of the morphological-skeleton transform.

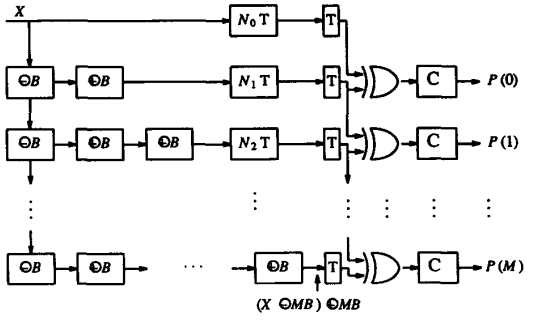


Fig. 11. An implementation of the pattern spectrum. T = delay element, C = counter.

the structure of the module becomes a simple cascade of the appropriate units. As Fig. 11 shows, the value of the pecstrum, $P(m)$ is given by the output of the corresponding counter. The delay elements used around the XOR gate are for synchronization purposes. The number of unit delays N_m (see Fig. 11) 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.

C. Geometrical Correlation Functions

The third example is the implementation of the *geometrical correlation functions* (GCF's) [26], [27]. The family of GCF's has been shown to possess very interesting characteristics that are very useful for shape representation and recognition. Specifically, the GCF, $K_\phi(h)$ of a binary image signal X is given by the following equation:

$$K_\phi(h) \triangleq \frac{\text{Mes}\left[X \ominus B_h^\phi\right]}{\text{Mes}\left[Y\right]} \qquad 0 \le \phi < \pi \qquad (24)$$

where B_h^ϕ is a structuring element that is composed of two single points separated by a distance h at an angle ϕ relative to 0°, Mes $[X]$ is a measure that is defined as the digital area of the image X, and Y is a predefined standard binary shape (e.g., a square of size 200 × 200 pixels). The structuring element B_h^ϕ, in fact, is an example of nonconvex structuring elements. The GCF can be restricted to a particular direction, for instance, horizontal. In this situation, the unnormalized GCF becomes

$$K_0(h) = \text{Mes}\left[X \ominus S(h)\right] \qquad h = 0, 1, 2, \cdots \qquad (25)$$

and the structuring element is given by

$$s(h) = (1^* \cdots {}^*1) \qquad (26)$$

where h is the distance separating the two "1" elements and * denotes a "don't care" condition. One decomposition based on (17), that allows repeated applications of structuring elements of increasing size, as well as ease of implementation, is described as follows: Let

$$A_0 = 1, \qquad B_0 = 1 \qquad (27a)$$

$$A_1 = 1^* \qquad B_1 = {}^*1. \qquad (27b)$$

Then $s(h)$ can be written as

$$s(0) = A_0 \text{ or } B_0 \qquad (28a)$$

$$s(h) = \{A_{h-1} \oplus A_1\} \cup \{B_{h-1} \oplus B_1\}\ h \ge 1 \qquad (28b)$$

where $A_h = A_{h-1} \oplus A_1$ and $B_h = B_{h-1} \oplus B_1$. Hence, substituting (28) into (25) yields

$$K_0(0) = \text{Mes}\left[X \ominus A_0\right] = \text{Mes}\left[X\right] \qquad (29a)$$

$$K_0(h) = \text{Mes}\left\{\left[X \ominus A_{h-1}\right] \ominus A_1\right\} \cap \left\{\left[X \ominus B_{h-1}\right] \ominus B_1\right\} h \ge 1. \qquad (29b)$$

Equation (29) can now be implemented using the structure depicted in Fig. 12. All of the modules A_1^i of Fig. 12 are identical due to the decomposition of (28). The same applies to all modules B_1^i. Each of these two module types can be implemented using an NPP that consists of a cascade of two binary erosion units of Fig. 3(d). Consider the processing of a 512 × 512 image for a 200-point GCF with $M = 20$: The total processing time, $T_p = 200/M \times 512 \times 512 \times T_{pd} = 10 \times 512 \times 512 \times 1$ ns = 2.62 ms, which is well below the 1/30 s (or 33.3 ms) constraint for video-rate processing. This, of course, also assumes that an input frame buffer is available so that the output samples from the final output port (the right-most AND gate) can be circulated back to the input for further processing.

The operation of the morphological correlator of Fig. 12 is described by example, as follows. Consider an input X = 1011100 where we want to calculate the first two points of the GCF starting at $h = 1$. According to (26), the two structuring elements are given by $s(1) = 11$ and $s(2) = 1^*1$, respectively. From (28b), we have

$$X \ominus s(1) = X \ominus (A_1 \cup B_1) = (X \ominus A_1) \cap (X \ominus B_1). \qquad (30)$$

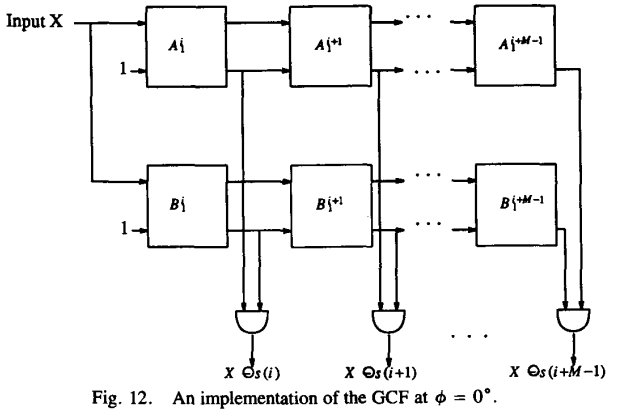


Fig. 12. An implementation of the GCF at $\phi = 0°$.

Substituting $X = 1011100$ into (30) yields

$$X \ominus s(1) = (X \ominus A_1) \cap (X \ominus B_1)$$
$$= 1011 \cap 0111 = 0011 \quad (31)$$

which is the expected result. Now let us proceed to calculate $X \ominus s(2)$, which is given by

$$X \ominus s(2) = ((X \ominus A_1) \ominus A_1) \cap ((X \ominus B_1) \ominus B_1). \quad (32)$$

Equation (32) implies that $X \ominus s(2)$ can be obtained by eroding the previous results, $X \ominus A_1$ and $X \ominus B_1$, one more time. The output of these operations is given by

$$X \ominus s(2) = 101 \cap 111 = 101. \quad (33)$$

VI. Summary

In summary, this paper has presented high-speed and modular nonlinear pipeline architectures for the efficient VLSI implementation of gray-scale and binary morphological filters or operators. Two standard building units—the dilation unit and the erosion unit—can be used to realize any required morphological transformation. This approach provides a very cost-effective way of designing dedicated morphological filters for image processing, as well as morphological operators for image analysis. The advantage of the systolic approach suggested is that maximum utilization of processing elements is attained through pipelining.

In comparison to the approach based on threshold decomposition [10], this design scheme is much simpler, requires less hardware, and is, thereby, more cost effective. Furthermore, due to unit modularity, corresponding architectures can be utilized directly to design any morphological-related function. This fact was illustrated by the flexibility exhibited in implementing three different morphological shape-representation schemes, including the morphological skeleton transform, the pattern spectrum, and the geometrical correlation functions. In addition, this paper also introduced the idea of geometrical decomposition, a process by which the implementation of large convex or nonconvex 2-D structuring elements can be made practical using 1-D structuring elements. This procedure is utilized for example in the efficient implementation of the geometrical correlation functions.

References

[1] A. K. Jain, Ed., *Real-Time Object Measurement and Classification*. New York: Springer-Verlag, 1988.

[2] A. N. Venetsanopoulos and V. Cappellini, "Real-time image processing," *Multidimensional Systems: Techniques and Applications*, S. G. Tzafestas, Ed. New York: Marcel Dekker, 1986, pp. 345-399.

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

[4] K. E. Batcher, "Design of a massively parallel processor," *IEEE Trans. Comput.*, vol. C-29, pp. 836-840, Sept. 1980.

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

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

[7] E. W. Kent and S. L. Tanimoto, "Hierarchical cell logic and the PIPE processor: structural and functional correspondence," in *Proc. IEEE Comput. Soc. Workshop Comp. Architecture for Pattern Analysis and Image Database Management*, Miami Beach, FL, Nov. 18-20, 1985, pp. 311-319.

[8] 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, CA, Aug. 27-28, 1980, pp. 281-286.

[9] 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, pp. 115-123, 1984.

[10] F. Y. C. Shih and O. R. Mitchell, "Threshold decomposition of gray scale morphology into binary morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 1, pp. 31-42, Jan. 1989.

[11] H. T. Kung, "Why systolic architectures?" *Computer*, vol. 15, pp. 37-46, Jan. 1982.

[12] S. Y. Kung, *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

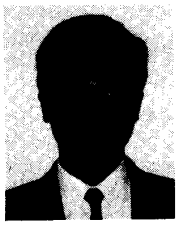
[13] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic Press, 1982.

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

[15] A. C. P. Loui, A. N. Venetsanpooulos, and C. L. Nikias, "Modular architectures for two-dimensional digital signal processing," *IEEE Trans. Circuits Syst.*, vol. 35, no. 1, pp. 43-56, Jan. 1988.

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

[17] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, "Stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 898-911, Aug. 1986.
[18] K. Hwang, *Computer Arithmetic—Principles, Architecture, and Design*. New York: Wiley, 1979.
[19] N. Weste and K. Eshraghian, *Principles of CMOS VLSI design: A system perspective*. Reading, MA: Addison-Wesley, 1985.
[20] J. S. Ward, P. Barton, J. B. G. Roberts, and B. J. Stanier, "Figures of merit for VLSI implementations of digital signal processing algorithms," *IEEE Proc.*, vol. 131, part F, no. 1, pp. 64-70, Feb. 1984.
[21] A. N. Venetsanopoulos and B. G. Mertzios, "A decomposition theorem and its implications to the design and realization of two-dimensional filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1562-1575, Dec. 1985.
[22] A. C. P. Loui, A. N. Venetsanopoulos, and C. L. Nikias, "Modular implementation of quadratic digital filters," in *Proc. 29th Midwest Symp. on Circuits and Systems*, pp. 937-940, Lincoln, NE, Aug. 1986.
[23] P. Maragos and R. W. Schafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 5, pp. 1228-1244, Oct. 1986.
[24] J. F. Bronskill and A. N. Venetsanopoulos, "Multidimensional shape description and recognition using mathematical morphology," *J. Intelligent and Robotic Syst.*, vol. 1, pp. 117-143, 1988.
[25] P. Maragos, "Pattern spectrum and multiscale shape representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 701-716, July 1989.
[26] A. C. P. Loui, A. N. Venetsanopoulos, and K. C. Smith, "Two-dimensional shape representation using morphological correlation functions," in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing*, pp. 2165-2168, Albuquerque, NM, April 3-6, 1990.
[27] A. C. P. Loui, "A morphological approach to moving-object recognition with applications to machine vision," Ph.D. dissertation, Department of Electrical Engineering, University of Toronto, Toronto, Canada, Sept. 1990.



Alexander C. P. Loui (S'82-M'90) received the B.A.Sc. (Honor), M.A.Sc. and Ph.D. degrees in electrical engineering, all from the University of Toronto, Toronto, Canada in 1983, 1986, and 1990, respectively

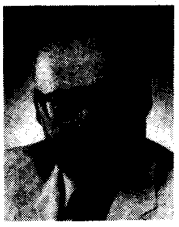
From 1986 to 1990, he was Research Assistant of the Signal Processing Laboratory at the Department of Electrical Engineering, University of Toronto. In 1990 he joined Bell Communications Research, New Jersey, as a Member of Technical Staff. His research interests currently include video signal processing, high-speed architectures, image processing, and computer vision.

He received the Natural Sciences and Engineering Research Council of Canada Postgraduate Scholarship from 1983 to 1987 and the University of Toronto Doctoral Fellowship from 1987 to 1989.

Anastasios N. Venetsanopoulos (S'66-M'69-SM'79-F'88) received the B.S. degree from the National Technical University of Athens (NTU), Greece, in 1965, and the M.S., M.Phil., and Ph.D. degrees in electrical engineering, all from Yale University, in 1966, 1968, and 1969, respectively.

He joined the University of Toronto, Canada, in September 1968, where he is now Professor in the Department of Electrical Engineering. He also served as Chairman of the Communications Group (1974 to 1978 and 1981 to 1986), and as Associate Chairman of the Department of Electrical Engineering (1978 to 1979). He was on research leave at the Swiss Federal Institute of Technology, the University of Florence, the Federal University of Rio de Janeiro, the National Technical University of Athens, and the Imperial College of Science and Technology, and was Adjunct Professor at Concordia University. He served as Lecturer of numerous short courses to industry and continuing education programs: he is a contributor to eleven books and has published over 300 papers in digital signal and image processing, and digital communications; he also served as consultant to several organizations, and as Editor of the *Canadian Electrical Engineering Journal* (1981 to 1983).

Dr. Venetsanopoulos was President of the Canadian Society for Electrical Engineering and Vice-President of the Engineering Institute of Canada (1983-1984). He was a Fulbright Scholar, A. F. Schmitt Scholar, and recipient of the J. Vakis Award. He is a member of the New York Academy of Sciences, Sigma Xi, the International Society for Optical Engineering, and the Technical Chamber of Greece; he is a Registered Professional Engineer in Ontario and Greece, and a Fellow of the Engineering Institute of Canada.



Kenneth Carless Smith (F'78) received the Ph.D. in solid-state physics from the University of Toronto in 1960.

He is presently a Professor of Electrical Engineering, Mechanical Engineering, Computer Science, and Library and Information Science at the University of Toronto. In addition, he has extensive industrial experience in the design and application of computers and electronic circuits. His research interests currently include analog VLSI, multiple-valued logic, flexible manufacturing, machine vision, instrumentation, array architectures, human factors, and reliability. He is widely published in these and other areas, with well over 150 journal, proceedings, books, and book contributions.

Dr. Smith was elected Honorary Professor at the Shanghai Institute for Railway Technology in 1989.