NON-LINEAR PIPELINE ARCHITECTURES FOR MORPHOLOGICAL SIGNAL PROCESSING

*A. C. P. Loui, A. N. Venetsanopoulos, K. C. Smith and B. Benhabib**

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

*Department of Mechanical Engineering
University of Toronto
Toronto, CANADA M5S 1A4

ABSTRACT

A non-linear pipeline architecture is proposed for efficient and high-speed implementation of morphological filters. This non-linear processor is composed of dual architectures or units which implement respectively the two basic morphological transformations: dilation and erosion. Throughput and hardware complexity are analyzed; and comparisons are given for different sizes and gray-scales of structuring functions. A decomposition scheme is proposed to facilitate the implementation of morphological filters based on one-dimensional structuring functions constructed using the two standard units. The modularity of this approach provides flexibility in the design of morphological filters and is well-suited for VLSI implementation.

1. INTRODUCTION

Mathematical morphology has become a very powerful tool for signal processing and analysis [1,2]. The strength of mathematical morphology lies in the natural coupling it provides between the characteristics of the signal under investigation and the structuring element. That is, by carefully selecting a suitable structuring element, morphological operations can be used for various filtering tasks, including noise removal and smoothing. Recently, morphological techniques have been used as well for shape representation in the area of image analysis.

Morphological signal processing and analysis have found important applications in many areas. They are particularly useful in robotic-vision applications where speed, size and cost are of critical importance. The inherent non-recursive nature of the basic morphological transformations implies that very high throughput rates can be attained. This is very important as most practical machine-vision systems require real-time capability. Applications where there is a need for dynamic vision systems include automated assembly and materials-handling systems [3]. In these applications, the ability of the vision system to detect and estimate object motion is of paramount importance.

As the applications of mathematical morphology become more and more specialized, the need for dedicated architectures is inevitable. Already, there exist a variety of machines that are capable of performing morphological or cellular-logic operations. Some examples of these machines are MITE [4], Cytocomputer [5], a special structure based on convolution and table lookup [6] and the threshold decomposition realization [7].

In this paper, non-linear architectures based on systolic organizations are proposed for high-speed morphological signal processing. The proposed architecture, which can be constructed using two basic building units, does not depend on the structure or size of the signal: 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. In Section 2, the non-linear dual architectures are derived based on a direct-form description of gray-scale dilation and erosion. Section 3 provides an analysis and comparison of the proposed structures in terms of throughput rate and hardware complexity. A decomposition scheme which allows the efficient use of the proposed processors, is discussed in Section 4 together with an example.

2. THE NON-LINEAR PIPELINE PROCESSOR

The operations of gray-scale morphological dilation and erosion can be described in terms of the top surface of a set, and the related concept of the umbra of a surface, in Euclidean n space. The top surface of a set X is a function defined on the projection of X onto its first $(n-1)$ coordinates. The umbra of a function f is a set consisting of the surface f and everything below that surface. More specifically, gray-scale dilation is defined as the surface of the binary dilations of the umbra of a function f with the umbra of a structuring function g [2]. However, this definition does not provide us with an efficient way of computing gray-scale dilation. In this paper, an alternate definition is employed which leads to a more practical structure.

It can be shown that the above definition of gray-scale dilation is equivalent to taking the maximum of a set of sums. This is described by the following equation [2]:

$$(f \oplus g)(x) = \max\{f(x-z) + g(z)\} \quad \forall z \in G,\ x-z \in F \ . \tag{1}$$

where $\oplus$ denotes dilation, $\ominus$ denotes erosion, and F and G are the domain of f and g respectively. Note that when the structuring function is set to zero, i.e., $g(x) = 0, \ \forall x \in G$, (1) becomes

$$(f \oplus g)(x) = \max\{f(x-z)\} \quad \forall z \in G,\ x-z \in F \ . \tag{2}$$

This equation (2) is called the dilation of a function by a set [1]. The equation for binary dilation is different from (1) and is given by

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

where X is the binary input image, b is the component of the structuring element B, and $(X)_b$ is X shifted by b.

Equation (1) can be considered as the direct-form implementation of dilation. This direct-form representation implies that gray- scale morphological operations can be realized using a structure similar to that employed for linear digital filtering. However, in this case, the operations involved are shifting, addition and comparison, instead of the conventional operations of shifting, multiplication and addition used in linear digital filtering. The structure described by (1) is shown in Figure 1.

The direct-form representation of erosion is similar to that of the dilation. Gray-scale erosion is basically defined as the surface of the binary erosion of the umbra of the function f with the umbra of the structuring function g. It is shown that this is equivalent to taking a minimum of a set of differences. In fact, erosion can be thought of as correlation where the summation operation is replaced by a minimum operation and multiplications become subtractions. The corresponding direct-form representation is given as follows [2]:

$$(f \ominus g)(x) = \min\{f(x+z) - g(z)\} \quad \forall z \in G,\ x+z \in F \ . \tag{4}$$

Similarly, if $g(x) = 0, \ \forall x \in G$,(4) becomes

CH2868-8/90/0000-1438$1.00 © 1990 IEEE

$$(f \ominus g)(x) = \max\{f(x+z)\} \quad \forall\, z \in G\,,\; x+z \in F \tag{5}$$

which is the erosion of a function by a set. The direct-form structure described by (4) is shown in Figure 2. The corresponding equation for binary erosion is given by

$$X \ominus B = \bigcap_{b \in B} (X)_{-b} \quad . \tag{6}$$

2.1 Gray-Scale Architectures

The direct-form structures of Figure 1 and 2 are not very practical for VLSI implementation due to their inherent reliance on non-local data transfers and processing. This inefficiency can be improved by distributing the computation processes and localizing the data transfers. This results in an structure similar to the systolic-array architecture as used in the implementation of many signal processing algorithms.

The resulting architecture for dilation is shown in Figure 3 (a) and is called a dilation unit. From Figure 3 (a), the dilation unit is basically composed of four latches, one adder and one comparator. Each unit has two input ports and two output ports for data transfers. A cascade of identical dilation units can be used to implement a dilation operation.

The corresponding architecture of the erosion unit is shown in Figure 3 (b). The internal structure of the erosion unit is similar to that of the dilation unit of Figure 3 (a). Except that in this case, the maximum operation becomes the minimum operation, and addition becomes subtraction. Also, the order of the structuring function g_i is reversed. In this case, the erosion unit is composed of four latches, one subtractor and one comparator. Again, a cascade of identical erosion units can be used to implement an erosion operation.

2.2 Binary Architectures

The precision of the gray-scale dilation unit is b bits. If b is equal to 1, the structure degenerates into a binary dilation unit. This applies also to the erosion unit. In the case of binary dilation the addition becomes a logical AND operation and the comparison (or maximum) operation becomes a logical OR. The resulting structure is depicted in Figure 4 (a). In the binary erosion unit, the corresponding elements are an EXCLUSIVE-NOR gate in place of the adder, with an AND gate for the comparator. The corresponding structure is shown in Figure 4 (b).

3. ANALYSIS AND COMPARISON

In this Section, a hardware complexity analysis and throughput analysis of the proposed architectures 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, as well as different algorithms for the same function, to be compared. The cost function is chosen to reflect as accurately as possible the functional (or logic) design independently of the technology used. This implies that the technology issue can be introduced separately in the design a practical system.

Let us begin with hardware complexity. Assume that the precision of both the input and output signals is b-bit. Since only addition and comparison operations are performed, the internal precision can also remain as b-bit. Also assume that full carry-lookahead adders are used, the typical number of logic gates of such adder being about $16b$ [8]. Hence for the implementation of a 4-bit carry-lookahead full adder, a total of approximately 64 logic gates are required. The basic magnitude comparators can be constructed using an adder and a complementer [9]. Since in the dilation (erosion) unit, the comparator is also required to provide an output of the maximum (minimum) of the two incoming signals, additional logic is needed to gate the appropriate value depending on the outcome of the comparison. The logic gate count for this comparator is estimated as $16b + 3b + 1 = 19b + 1$. The b-bit parallel-access latches are assumed to have a gate complexity of $10b$ logic gates [8].

According to the discussion above, the complexity of the proposed dilation or erosion unit is given by $C = 16b + 19b + 1 + 4(10b) = 75b + 1$. For example, the logic gate count for a 3×3 structuring element with a precision of $b = 8$ is approximately equal to $(75(8) + 1)(9) = 5409$. Assuming that both the input signal and the structuring function have the same gray-level, the complexity functions, C, are shown in Figure 5 as a function of the size of the structuring function for different gray-levels. Note that the graphs of Figure 5 apply to both the dilation and the erosion structures since they have basically the same hardware configuration. This is different from the implementation of [7], where the complexity, as measured by the number of logic gates for gray-scale erosion, is always about twice that for dilation. In case of 2-D structuring functions, it is assumed that the shift register units are used to provide the proper delay for the input of the structure. Note also that the hardware complexity is a linear function of both the precision b and of the structuring function size N.

The cycle time of the proposed architecture can be determined by the longest operation undertaken between two latches. In this case, it is approximately equal to one addition time, or more precisely one comparison time, since the comparator is implemented using an adder plus some other logic gates. Hence, the cycle time, $T_c = T_a + T_{pd}$, where T_a is the addition time and T_{pd} is the propagation delay of a logic gate. As can be seen from Figure 3, the cycle time is the same for the dilation unit as for the erosion unit. For the corresponding binary units, the cycle time reduces to one propagation delay, i.e., $T_c = T_{pd}$. It is also assumed that the addition time is approximately equal to 5 propagation delays, i.e., $T_a = 5T_{pd}$ [8]. Thus, the cycle time for the gray-scale unit is about 6 times longer than that of the binary unit.

If we consider a 1.2μm CMOS technology, the typical gate propagation delay is around 1ns. This implies a throughput rate of up to 160 MHz for the gary-scale unit and up to 1 GHz for the binary unit. These high throughput rates can be attained after an initial period of latency required to fill up the pipeline. For a dilation module which is constructed by a cascade of the standard unit shown in Figure 3 (a), the latency is $T_l^d = (N + 3)T_c$, where N is the length of the structuring function. The corresponding latency for a erosion module which is constructed by a cascade of the standard units shown in Figure 3 (b) is given by $T_l^e = 2(N + 1)T_c$. Note that the latency for the erosion module is always longer than that for the dilation module. This is true since erosion, being a shrinking operation, will require a longer delay for the appearance of the first element of output data.

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 Figure 6. 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.

4. DECOMPOSITION OF STRUCTURING ELEMENTS

In morphological image processing and analysis, the structuring elements used are usually two-dimensional. Some examples of such morphological filters include close-opening filters, open-closing filters, and order-statistic filters such as median filters. Examples of image analysis tools based on morphological operations include the pattern spectrum [10] and skeleton transform [11]. The implementation of these morphological shape descriptors requires repeated operations of structuring elements of increasing size or scale. Hence, a means of decomposing a large complex structuring element into a few simpler ones will help in reducing hardware complexity as well as implementation cost.

The hardware complexity of the direct implementation of 2-D structuring element is of the order of N^2, where the size of the structuring element is assumed to be equal to a maximum value, N×N. When N

is large, the hardware size increases very rapidly, making the implementation infeasible. In order to reduce the size of a 2-D morphological filter, an efficient decomposition of structuring elements is necessary.

In morphological image processing, the notion of a structuring element is analogous to the kernel matrix of a 2-D linear digital filter. Therefore, if a large structuring element can be decomposed, as in this case, into the Minkowski sum of several much simpler 1-D structuring elements, the 2-D morphological image-processing problem can be reduced to a 1-D problem.

Here, a decomposition structure is proposed which can be used to realize any structuring element in the digital space Z^2:

Proposition 1: Suppose B is a structuring element in Z^2. It is possible to decompose B geometrically into the following form:

$$B = \bigoplus_{j=1}^{N_1} B_{1j} \cup \bigoplus_{j=1}^{N_2} B_{2j} \cup \cdots \cup \bigoplus_{j=1}^{N_m} B_{mj} = \bigcup_{i=1}^{m} \bigoplus_{j=1}^{N_i} B_{ij} \tag{7}$$

where $\cup$ denotes set union and each B_{ij} is a 1-D sub-structuring element. When $m = 1$, equation (7) becomes the familiar dilation operation. The introduction of the union operations ensures that structuring elements, which are not geometrically connected, can also be decomposed. One such example is a pair of points separated by a distance h. This structuring element has found use for example, in the computation of morphological correlation [12].

The decomposition structure suggested by equation (7) can also be extended to a gray-scale structuring function. In that case, the unions are replaced by maximum operations, and binary dilation becomes gray-scale dilation. Hence, after the structuring element (function) has been decomposed, each of the sub-structuring elements (functions), B_{ij} (g_{ij}), can be implemented using the proposed architectures.

As an example, let us look at the implementation of the geometrical correlation function (GCF) proposed in [12] for shape representation and analysis. The definition of the unnormalized GCF is given by the following equation:

$$K_\phi(h) \triangleq Mes[X \ominus B_h^\phi], \quad h = 0,1,2,\cdots \tag{8}$$

where X is the binary input image, B_h^ϕ is a structuring element which is composed of two single points separated by a distance h at an angle ϕ relative to 0°, and $Mes[X]$ is a measure which is defined as the digital area of the image X. The GCF can be restricted to a particular direction, for example, horizontal. In this situation, (8) becomes

$$K_0(h) = Mes\,[X \ominus s(h)], \quad h = 0,1,2,\cdots \tag{9}$$

and the structuring element is given by

$$s(h) = (1 * \cdots * 1) \tag{10}$$

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

$$A_0 = 1, \quad B_0 = 1 \tag{11a}$$

$$A_1 = 1\,*, \quad B_1 = *\,1 \tag{11b}$$

Then $s(h)$ can be written as

$$s(0) = A_0 \text{ or } B_0, \tag{12a}$$

$$s(h) = \{A_{h-1} \oplus A_1\} \cup \{B_{h-1} \oplus B_1\} \quad h \geq 1 \tag{12b}$$

where $A_h = A_{h-1} \oplus A_1$ and $B_h = B_{h-1} \oplus B_1$. Hence, substituting (12) into (9) yields

$$K_0(0) = Mes\,[X \ominus A_0] = Mes\,[X] \tag{13a}$$

$$K_0(h) = Mes\,\{[X \ominus A_{h-1}] \ominus A_1\} \cap \{[X \ominus B_{h-1}] \ominus B_1\} \quad h \geq 1 \tag{13b}$$

The implementation of (13) can now be carried out quite easily, and is shown in Figure 7. Each of the modules A_1^i is identical due to the decomposition of (12). The same applies to the module B_1^i. Each of these modules can in fact be implemented using a cascade of two binary erosion units as shown in Figure 4 (b).

5. SUMMARY

In summary, this paper presents high-speed and modular architectures for the efficient VLSI implementation of gray-scale and binary morphological filters or operators. This is accomplished using two basic building blocks. 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 signal processing, as well as morphological operators for signal analysis. The advantage of the systolic approach suggested is that maximum utilization of processing elements is attained through the pipelined nature of the architecture. In addition, this paper also introduces the idea of geometrical decomposition, a process by which the implementation of large complex structuring elements can be made practical using 1-D structuring elements. This idea is illustrated by a sample implementation of a morphological correlation function, provided.

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] R.U. Ayres et al, *Robotics and flexible manufacturing technologies: Assessment, impacts and forecast*, Noyes Publications, New Jersey, 1985.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] J.S. Ward, P. Barton, J.B.G. Roberts and B.J. Stanier, "Figures of merit for VLSI implementations of digital signal processing algorithms," *IEE Proceedings*, Vol. 131, Part F, No. 1, Feb 1984, pp. 64-70.

[9] N. Weste and K. Eshraghian, *Principles of CMOS VLSI design: A system perspective*, Addison-Wesley, 1985.

[10] 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.

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

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

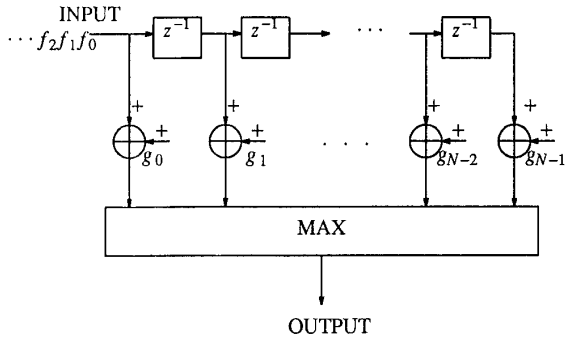

Figure 1 A direct-form implementation of gray-scale dilation

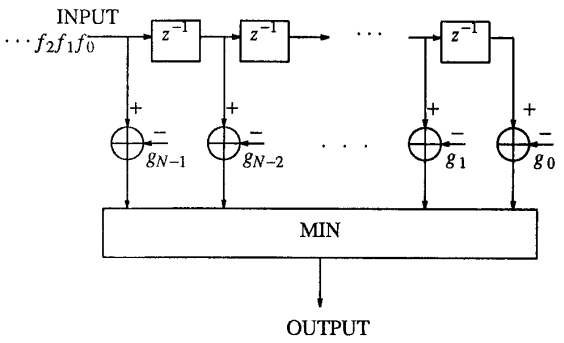

Figure 2 A direct-form implementation of gray-scale erosion

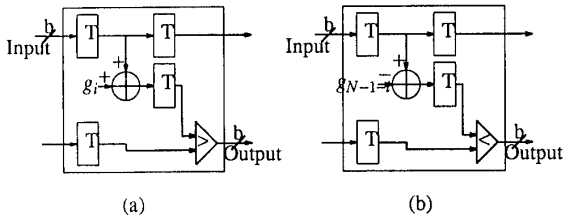

Figure 3 (a) A gray-scale dilation unit (b) A gray-scale erosion unit

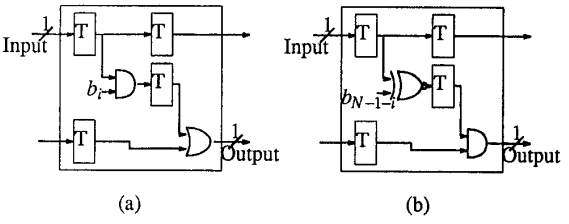

Figure 4 (a) A binary dilation unit (b) A binary erosion unit

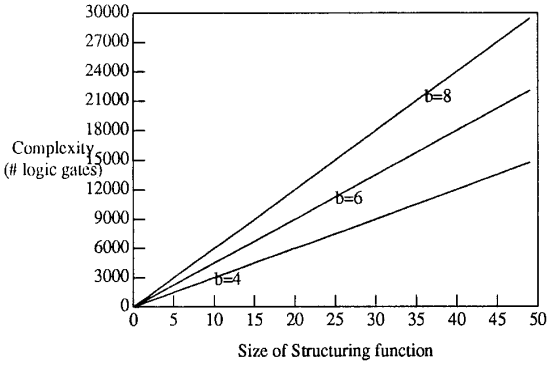

Figure 5 Complexity of the dilation or erosion unit vs size of structuring function at different gray-level

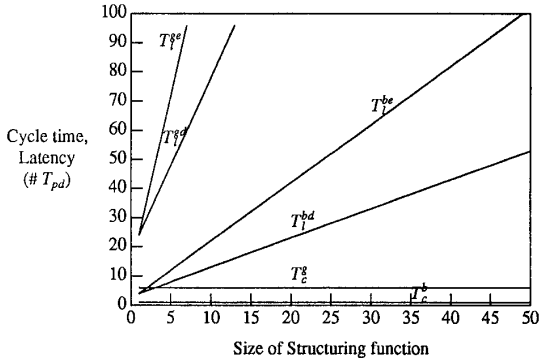

Figure 6 Cycle time T_c and latency T_l of the dilation and erosion unit vs size of structuring function

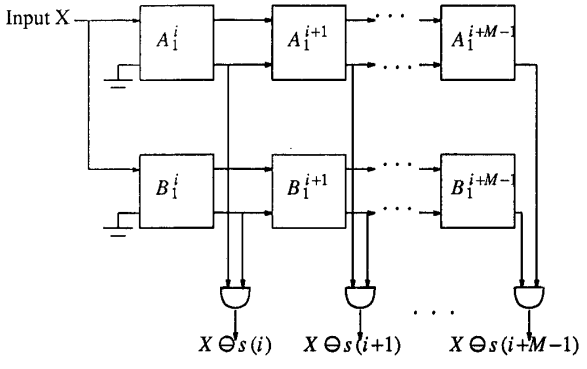

Figure 7 An architecture for the computation of the geometrical correlation function