論文 94-19-10-2

EFFICIENT IMPLEMENTATION OF GRAYSCALE MORPHOLOGICAL OPERATORS

Sung-Jea Ko*, Kyung-Hoon Lee*

형태학 필터의 효과적 구현 방안에 관한 연구

正會員 高 聖 濟* 正會員 李 坰 勳*

Abstract

This paper presents efficient real time software implementation methods for the grayscale morphological composite function processing (FP) system. The proposed method is based on a matrix representation of the composite FP system using a basis matrix composed of structuring elements. We propose a procedure to derive the basis matrix for composite FP systems with any grayscale structuring element (GSE). It is shown that composite FP operations including morphological opening and closing are more efficiently accomplished by a local matrix operation with the basis matrix rather than cascade operations, eliminating delays and requiring less memory storage. In the second part of this paper, a VLSI implementation architecture for grayscale morphological operators is presented. The proposed implementation architecture employs a bit-serial approach which allows grayscale morphological operations to be decomposed into bit-level binary operation unit for the p-bit grayscale singnal. It is shown that this realization is simple and modular in structure and thus is suitable for VLSI implementation.

要 約

본 논문에서는 濃淡構造素(GSE, grayscale structuring element)를 갖는 형태학 필터의 실시간 처리를 위한 알고리즘을 제안하였다. 제안된 알고리즘에서는 GSE로부터 유도된 basis matrix와 입력 샘플들로 구성된 input matrix를 이용하여 각 형태학 연산들을 소역행렬연산(local matrix operation)으로 새롭게 정의하고 있는데, 이를 이용하여 opening이나 closing과 같은 복합 형태학 연산들을 실시간으로 처리할 수 있음을 보였다. 제안된 알고리즘은 복합 형태학 연산들을 erosion과 dilation의 직렬조합(cascade combination)으로 처리하던 기존의 방법에 비해 적은 메모리를 필요로 하면서도, 출력을 얻기까지의 遲延(delay)이 훨씬 적다는 장점을 갖는다. 또한 본 논문에서는 형태학 필터를 VLSI로 구현하기 위한 효율적 방안을 제안하였다. 제안된 방법에서는 p-bit으로 표현되는 신호에 대한 형태학 연산을 p개의 이진(binary) 형태학 연산자들의 조합으로 구현하였는데, 각 이진 연산자들

* 고려대학교 전자공학과 교수
Professor, Dept. of Electronic Engineering, Korea University
** 고려대학교 전자공학과 대학원
論文番號 : 93206
接受日字 : 1993年 11月 2日

은 MSB(most significant bit)부터 순차적으로 (bit-serial approach) 해당 레벨의 bit들을 처리하여 출력을 얻는 구조로 이루어져 있다. 본 논문에서는 형태학 필터의 VLSI 구현에 있어서 제안된 방법이 기존의 Threshold Decomposition 방법 등에 비해 보다 효율적이라는 것을 보였다.

I. INTRODUCTION

Mathematical morphology is a powerful tool for image analysis and computer vision. Morphological image analysis systems have been successfully used in many applications including object rocognition, image enhancement, texture analysis, and industrial inspection. The basic morphological operations are erosion and dilation. Based on these operations, several composite operations including closing and opening are defined. Many theoretical results concerning the operations of mathematical morphology can be found in [1]-[4].

The morphological operations can be classified into three types of operations: i) The most general morphological operation, the function-processing (FP) operation, uses the grayscale structuring element (GSE) and accepts as inputs multilevel signals and produces as outputs multilevel signals. ii) The second type of the morphological operation is the set-processing (SP) operation using the binary SE whose both inputs and outputs are binary signals. iii) The last one, the subclass of the FP operation, is the function and set processing (FSP) operation. This operation uses the binary SE and produces a binary signal whenever the input is binary signal. A variety of implementation algorithms developed for the order statistic and stack filters can be utilized for implementation of morphological FSP and SP operations. However, these algorithms can not be directly applied to FP operations using the GSE which do not obey the threshold decomposition. Moreover, FP opening / closing using the cascade representatin(erosion / dilation followed by dilation / erosion) requires memory storage for the first [illegible] are introduced.

In this paper, we first [illegible] efficient software implementation algorithm for the gray-scale morphological FP operations. The proposed method is based on a matrix representation of the composite FP system using a basis matrix which is an extension of the *basis function*[11][10]. We propose a procedure to derive the basis matrix for FP systems from any GSE. It is shown that opening and closing are accomplished by a local matrix operation with the basis matrix rather than cascade operations, eliminating delays and requiring less memory storage. In order to improve the computational efficiency of the proposed software implementation method, we utilize a recursive algorithm based on the observation of the basis matrix and input matrices. The analysis of the basis matrix shows that the basis matrix is skew symmetric and has many redundant entries. It is also shown that most entries in the successive input matrices at two adjacent time indices are identical. By eliminating these redundancies, a fast recursive formula for the proposed matrix operation can be obtained which can significantly reduce the required computation. To evaluate the computational efficiency of the proposed scheme, the required number of operations for each morphological operators including opening and closing is calculated. It is shown that, with the proposed scheme, both opening and closing can be obtained by 2N-2 additions and 2N-2 comparisons when the size of the GSE is equal to N.

A VLSI implementation architecture of the stack

filter using the threshold decomposition was first proposed by Wendt *et. al.*[7], and Shih and Mitchell[8] modified this architecture for the implementation of the morphological operations. In the second part of this paper, we propose a VLSI implementation architecture for grayscale morphological FP operations based on the bit serial approach[9][12]. The bit serial architecture has been utilized for the implementation of the order statistic and stack filters. This technique, however, has never been applied to FP composite operators since these operators do not obey the superposition principle. The proposed bit-serial technique allows grayscale morphological operations to be decomposed into bit level binary operations by a bit modification algorithm. The hardware complexity of this realization grows linearly with the number of bits, as compared with the exponentially increasing complexity of the threshold decomposition method[13][14]. Furthermore, to increase data throughput rates, a bit level pipelined architecture is presented, which can significantly reduce the delay time.

The organization of this paper is as follows: In Section II, grayscale morphological operations are defined, and the local matrix operations for opening and closing are explained. The fast implementation algorithms for the composite morphological operators using the matrix representation are proposed in Section III, and the VLSI implementation algorithm and architecture are presented in Section IV. Finally, the concluding remarks are presented in Section V.

II. GRAYSCALE MORPHOLOGICAL FUNCTION PROCESSING

In this section, we first review grayscale morphological FP systems [1]-[4] and point out that composite FP morphological operation can be accomplished by a local operation of neighborhood input samples.

The basic morphological operations of a grayscale signal $\mathbf{f}$ by a GSE $\mathbf{k}$ with size N are defined as follows: Let the domain of $\mathbf{f}$ be denoted by F and the domain of $\mathbf{k}$ by K.

Dilation: The grayscale dilation of $\mathbf{f}$ by $\mathbf{k}$ is denoted by g_d, and is defined by

$$g_d(n) = (\boldsymbol{f} \oplus \boldsymbol{k})(n) = \max_z [\ f(n-z) + k(z)] \quad (1)$$

where the maximum is selected from the set of sums over all $z \in K$ and $n-z \in F$.

Erosion: The grayscale erosion of $\mathbf{f}$ by $\mathbf{k}$ is denoted by g_e, and is defined by

$$g_e(n) = (\boldsymbol{f} \ominus \boldsymbol{k})(n) = \min_z [\ f(n+z) - k(z)] \quad . (2)$$

where the minimum is taken from the set of differences over all $z \in K$ and $n+z \in F$.

Opening is dilation of a eroded signal, and closing is erosion of a dilated signal. These composite morphological operations are defined as follows:

Opening: The grayscale opening of $\mathbf{f}$ by $\mathbf{k}$ is denoted by g_o, and is defined by

$$g_o(n) = (\boldsymbol{f} \circ \boldsymbol{k})(n) = [\ (\boldsymbol{f} \ominus \boldsymbol{k}) \oplus \boldsymbol{k}]\ (n). \quad (3)$$

Closing: The grayscale closing of $\mathbf{f}$ by $\mathbf{k}$ is denoted by g_c, and is defined by

$$g_c(n) = (\boldsymbol{f} \bullet \boldsymbol{k})(n) = [\ (\boldsymbol{f} \oplus \boldsymbol{k}) \ominus \boldsymbol{k}]\ (n). \quad (4)$$

Next we develop a max-min/min-max representation for grayscale morphological opening/closing.

The opening and closing in (3) and (4) can be represented as follows:

Proposition 1: The FP opening and closing of $\mathbf{f}$ by $\mathbf{k}$ are equivalent to

$$g_o(n)=\max_{z}\left\{\min_{z'}\left[\ f(n-z+z')+b(z,z')\right]\right\},\quad(5a)$$

and

$$g_c(n)=\min_{z}\left\{\max_{z'}\left[\ f(n+z-z')-b(z,z')\right]\right\},\quad(5b)$$

where $b(z,z')=k(z)-k(z'),\ z\in K$ and $z'\in K$

Proof: From the definition of grayscale opening,

$$\begin{aligned}g_o(n)&=\left[\ (f\ominus k)\oplus k\right]\ (n)\\&=(g_e\oplus k)(n)\\&=\max_{z}\left[\ (g_e(n-z)+k(z)\right]\\&=\max_{z}\left\{\min_{z'}\left[\ f(n-z+z')-k(z')\right]\ +k(z)\right\},\end{aligned}$$

since $\min[\ a,b]\ +c=\min[\ a+c,b+c]$, g_0 can be represented as

$$g_o(n)=\max_{z}\left\{\min_{z'}\ +[\ f(n-z+z')\ +b(z,z')]\ \right\}$$

$g_c(n)$ can be derived similarly.

This property is also a direct consequence of the morphological representation theory (Theorem 3 in [6]). the max-min / min-max representations using this property for grayscale morphological opening / closing, although more complex, are faster than the cascade representations(erosion / dilation followed by dilation / erosion). For example, consider a GSE $k=\{k(0),k(1),k(2)\}$.

The outputs of opening and closing, respectively, are given by

$$\begin{aligned}g_o(n)=\max\{&\min[\ f(n),f(n+1)+\\&+b(0,1),f(n+2)+b(0,2)]\ ,\\&\min[\ f(n-1)+b(1,0),f(n),f(n+1)\\&+b(1,2)]\ ,\\&\min[\ f(n-2)+b(2,0),f(n-1)+\\&+b(2,1),f(n)]\ \},\end{aligned}$$

and closing

$$\begin{aligned}g_c(n)=\min\{&\max[\ f(n),f(n-1)\\&-b(0,1),f(n-2)-b(0,2)]\ ,\\&\max[\ f(n+1)-b(1,0),f(n),f(n-1)\\&-b(1,2)]\ ,\\&\max[\ f(n+2)-b(2,0),f(n+1)\\&-b(2,1),f(n)]\ \},\end{aligned}$$

Next, we propose a fast implementation method for the local matrix operation of opening and closing using matrix notation.

Proposition 1 can be expressed in a matrix form using the input matrix, denoted by $F(n)$, and the basis matrix, denoted by B, as follows: The $N\times N$ input matrix $F(n)$ contains $2N-1$ input sample, $\{f(n-N+1),\cdots,f(n+N-1)\}$, and defined by

$$F(n)=\begin{pmatrix}f(n) & f(n+1) & \cdots & f(n+N-1)\\ f(n-1) & f(n) & \cdots & f(n+N-2)\\ \vdots & \vdots & \ddots & \vdots\\ f(n-N+1) & f(n-N+2) & \cdots & f(n)\end{pmatrix}.\quad(6)$$

The $N\times N$ basis matrix B, whose elements consist of $\{b(i,j)\}$, is defined by

$$B=\begin{pmatrix}b(0,0) & b(0,1) & \cdots & b(0,N-1)\\ b(1,0) & b(1,1) & \cdots & b(1,N-1)\\ \vdots & \vdots & \ddots & \vdots\\ b(N-1,0) & b(N-1,1) & \cdots & b(N-1,N-1)\end{pmatrix}.\quad(7)$$

It is interesting to observe several properties of this basis matrix. First, each row represents a basis function as defined by [5],[6]. Second, since $b(i,j)=k(i)-k(j)$, $b(i,i)=0$ and $b(i,j)=-b(j,i)$, the basis matrix can be written

as:

$$B=\begin{bmatrix} 0 & b(0,1) & \cdots & b(0,N-1) \\ -b(1,0) & 0 & \cdots & b(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ -b(0,N-1) & -b(1,N-1) & \cdots & 0 \end{bmatrix} \tag{8}$$

Note that the basis matrix is skew symmetric, that is, $B^T=-B$, where B^T is the transpose of B. This matrix also implies that the total number of distinct elements can be reduced from N^2 down to $(N^2-N)/2$.

The output of opening, $g_o(n)$, can be obtained by finding the maximum of the minima of each rows from the matrix $F(n)+B$. The output of closing, $g_c(n)$, is the minimum of the maxima of each columns of the matrix $F(n)+B$. This implementation of the opening and closing operations using the matrix operators requires less delays and memory storage than the straightforward two-stage cascade combinations of erosion and dilation.

III. FAST IMPLEMENTATION OF LOCAL MATRIX OPERATORS

In this section, a fast implementation method based on the proposed local operation is presented fo composite FP morphological operations.

As described in the previous section, the outputs of opening and closing, respectively, are determined by using the row-wise minima and the column-wise maxima of the matrix $F(n)+B$. Expressing the opening and closing operations in terms of the row-wise min and column-wise max operations gives

$$g_o(n)=\max[\ R_0(n), R_1(n), \cdots, R_{N-1}(n)]\ , \tag{9a}$$

$$g_c(n)=\min[\ C_0(n), C_1(n), \cdots, C_{N-1}(n)]\ , \tag{9b}$$

where $R_i(n)$ and $C_i(n)$, respectively, denote the minimum of the i^{th} row and the maximum of the i^{th} column of the matrix $F(n)+B$, *i.e.*,

$$R_i(n)=\min_j\ [\ f(n-i+j)+b(i,j)]\ , \tag{10a}$$

$$C_i(n)=\max_j\ [\ f(n-j+i)+b(j,i)]\ , \tag{10b}$$

where $j\in K$ and $j\in K$. It can be easily shown that the opening and closing operations in (9), respectively, require N^2-N additions and N^2-1 comparisons.

The proposition stated below indicates that the outputs of the opening and closing operators at time n can be recursively obtained from the previous operation results

$$\{R_i(n-1), 0\le i\le N-2\}, \{C_j(n-1), 1\le j\le N-1\}.$$

Proposition 2: The i^{th} row-wise minimum, $R_i(n)$, and the j^{th} column-wise maximum, $C_j(n)$, of $F(n)+B$ can be obtained using the following recursive formula:

$$R_i(n)=R_{i-1}(n-1)+b(i,i-1),\quad i=1,2,\cdots,N-1, \tag{11a}$$

$$C_j(n)=C_{j+1}(n-1)+b(j,j+1),\quad j=0,1\cdots,N-2. \tag{11b}$$

proof: For $i=1,2,\cdots,N-1$,

$$\begin{aligned} R_i(n)&=\min[\ f(n-i)+b(i,0), f(n-i+1)+b(i,1), \cdots, f(n+N-i-1)+b(i,N-1)] \\ &=\min[\ f(n-i)+b(i-1,0)+b(i,i-1), f(n-i+1)+b(i-1,1)+b(i,i-1), \cdots, f(n+N-i-1)+b(i,N-1)+b(i,i-1)] \\ &=\min[\ f(n-i)+b(i-1,0), f(n-i+1)\end{aligned}$$

$$+b(i,1),\cdots,f(n+N-i-1)+b(i,N-1)]$$
$$+b(i,i-1)$$
$$=R_{i-1}(n-1)+b(i,\,i-1) \tag{12}$$

In the similar way, $C_j(n)$ can be obtained using $C_{j+1}(n-1)$ recursively.

Using this proposition, the opening and closing operations can be redefined as

$$g_o(n)=\max[\ R_0(n),R_0(n-1)+b(1,0),$$
$$\cdots,R_{N-2}(n-1)+b(N-1,N-2)]\ , \tag{13a}$$
$$g_c(n)=\min[\ C_1(n-1)+b(0,1),\cdots,$$
$$\cdots,C_{N-1}(n-1)+b(N-2,N-1),$$
$$\cdot\ C_{N-1}(n)]\ . \tag{13b}$$

These equations represent a fast implementation method of the opening and closing operations. The calculation of $R_0(n)$ requires $N-1$ additions and $N-1$ comparisons. The calculation of $\{R_i(n-1)+b(i,i-1),0\le i\le N-2\}$ requires $N-1$ comparison. The total number of operations required to determine opening and closing at each point is equal to $2N-2$ additions and $2N-2$ comparisons. Therefore, the opening and closing operations using Proposition 2 is computationally more efficient than the operations without using Proposition $2(N^2-N)$ additions and N^2-1 comparisons).

To illustrate the computational efficiency of the proposed fast implementation method, two schematic diagrams for opening with the GSE of size three are presented in Fig. 1. The signal flow graph of the local matrix operator in (9a) is presented in Fig. 1(a), and the graph of the proposed recursive opening operator in (13a) is shown in Fig. 1(b). These two diagrams show that the proposed recursive structure has significantly less computations. For the software realization high-level language descriptions of the proposed fast opening and closing are given in Algorithm 1 and Algorithm 2 respectively.

Algorithm 1: The fast recursive algorithm for opening can be realized using the high-level language as follows:

```
begin (* main routine of opening*)

(* initialization*)
for i : = 0 to N-1 do (* N: the size of GSE *)
      R(i) = fmin(i, N-i-1);
out(0) = Rmax( );

(* recursive structure *)
for i: = 1 to L-2N+1 do (* L:the length of f *)
    for j: = 0 to N-2 do
        R(j) = R(j+1)+b(N-j-1, N-j-2);
    R(N-1) = fmin(i+N-1,0);
    out(i) = Rmax( );
end
function fmin(i, j) (* find the minimum of N
input samples at time i *)
begin
    min = f(i)+b(j, 0);
    for k : = 1 to N-1 do
      if (f(i+k)+b(j,k)<min) then min=f(i+k)+b(j,k);
return min;
end

function Rmax( )
(* find the maximum of R_i, 0≤i≤N-1*)
begin
 max = R(0);
 for i := 1 to N-1 do
    it (R(i)> max) then max = R(i)
 return max;
end
```

Algorithm 2: The fast recursive closing can be realized as follows:

```
begin (* main routine of closing *)
 (* initialization *)
 for i: =0 to N-1 do
     C(i)= fmax (i, i);
   out(0) = Cmin ( );

(* recursive structure *)
for i: =1 to L-2N+1 do (*L:the length of f *)
  for j: =0 to N-2 do
      C(j) = C(j+1) + b(j, j+1);
  C(N-1)= fmax (i+N-1, N-1);
   out(i)= Cmin( );
end

function fmax (i, j)
(* find the maximum of N input samples at time i *)
begin
    max = f(i) + b(N-1, j);
    for k: =1 to N-1 do
      if (f(i+k) + b(N-k-1, j) > max)
      then max = f(i+k) + b(N-k-1, j);
    return max;
end

function Cmin( )
(* find the minimum of C_i, 0≤i≤N-1 *)
begin
   min = C(0);
   for i: =1 to N-1 do
       if (C(i) < min) then min = C(i);
   return min;
end
```

Similarly, the fast implementation algorthm for closing can be easily obtained.

IV. A VLSI IMPLEMENTATION ALGORITHM FOR GRAYSCALE MORPHOLOGY

Suppose that both the input **f** and the GSE **k** with size N are p bit(2^p level) nonnegative signals and that the output of the FP system is also a p-bit non negative signal. To simplify notation, we denote by $W=\{A_0, A_1, \cdots, A_{L-1}\}$ the set of sums or differences for and FP operation.

The output g of a grayscale morphological operator with the GSE of size N can be expressed by

$$g = \Phi\{A_0, A_1, \cdots, A_{L-1}\}, \tag{14}$$

where Φ is a min/max operator representing either a single operation (min or max), or a combined operation of min and max (min of maxima or max of minima), and $L=N$ for erosion and dilation, and $L=N^2$ for opening and closing. For example, the outputs of the grayscale opening and closing with the GSE of size N, repectively, are given by

$$\begin{aligned} g_o = \max\{ & \min[A_0, \cdots, A_{N-1}], \\ & \min[A_N, \cdots, A_{2N-1}], \\ & \min[A_{L-N}, \cdots, A_{L-1}]\}, \\ g_c = \min\{ & \max[A_0, \cdots, A_{N-1}], \\ & \max[A_N, \cdots, A_{2N-1}], \\ & \max[A_{L-N}, \cdots, A_{L-1}]\}, \end{aligned}$$

From (1) (5), the A_i in the set W for each $i, 1 \le i \le L-1$, is given by

$$A_i=\begin{cases} f(n-1)+k(i), & \text{for dilation,}\\ f(n+1)-k(i), & \text{for erosion,}\\ f(n+i-(N+1)\lfloor i/N\rfloor,\\ \quad +b(\lfloor i/n\rfloor, & \\ \quad i-(N+1)\lfloor i/N\rfloor), & \text{for opening,}\\ f(n-i+(N+1)\lfloor i/N\rfloor,\\ \quad -b(\lfloor i/n\rfloor, i & \\ \quad -(N+1)\lfloor i/N\rfloor), & \text{for closing,}\end{cases} \tag{15}$$

where $\lfloor i/N\rfloor$ represents the integer part of i/N. In order to ensure that the output is limited to the range $[\,0,\ 2p-1]$, it is assumed that the set of data in the window is clipped in such a way that $A_i=0$ if $A_i\le 0$ and $A_i=2^{p-1}$ if $A_i\ge 2_p-1$.

The output of FP operators can be obtained by either the threshold decomposition method [7],[13],[14] or the bit-serial method [9]-[12] in conjunction with the circuit that generates the A_i inputs. In the threshold decomposition method, a p-bit grayscale input signal is decomposed into 2^p-1 binary signals, $\{t_i^1, t_i^2, \cdots, t_i^{2p-1}\}$, and each binary signal are processed in parallel, and finally outputs of each binary operation are summed to reconstruct a grayscale output. Since the min/max operator has the threshold decomposition property, it can be implemented as shown in Fig. 2. Obviously, the Boolean AND/OR circuits for binary operations can be applied to this realization. method consider, an example for erosion with the GSE of size three. Suppose that the input to the min/max operator is given by $\{A_0, A_1, A_2\}=\{1,3,2\}$. Thresholding this input at level 1, 2, and 3 generates three binary signals $\{t_0^3, t_1^3, t_2^3\}=\{0,1,0\}, \{t_0^2, t_1^2, t_2^2\}=\{0,1,1\}$, $\{t_0^1, t_1^1, t_2^1\}=\{1,1,1\}$. Each binary signals are processed in parallel through the binary erosion operators which can be implemented simply by using a three-input Boolean AND operator. Finally, adding all the outputs of each binary operators gives the output of erosion. Next we shall show that FP operators can be implemented by the bit-serial method incorporating the binary morphological operators.

Let the p-bit code words (radix-2 binary representation) of A_i and g, respectively, be $(a_i^1, a_i^2, \cdots, a_i^p)$, and $(b^1, b^2, \cdots, b^P)$ where a_i^1 and b^1 are the most significant bits (MSB). In the bit-serial realization, the output of each bit is obtained sequentially, starting with the MSB. At each bit-level, with the exception of the MSB, the binary input values at the level are modified before being applied to the binary morphological operator. This modification depends upon the outputs of the more significant bits as follows:

Proposition 3: In an FP operation, the output of the binary morphological operator at the j^{th} bit level is given by

$$b^j=\begin{cases} \hat{a}_o+\hat{a}_1+\cdots+\hat{a}_{N-1}, & \text{for dilation,}\\ \hat{a}_o\,\hat{a}_1\cdots\hat{a}_{N-1}, & \text{for erosion,}\\ (\hat{a}_o\cdots\hat{a}_{N-1})+ & \\ \quad(\hat{a}_N\cdots\hat{a}_{2N-1})+ & \\ \quad\cdots+(\hat{a}_{N^2-N}\cdots\hat{a}_{N^2-1}), & \text{for opening,}\\ (\hat{a}_o+\cdots+\hat{a}_{N-1}) & \\ \cdot(\hat{a}_N+\cdots+\hat{a}_{2N-1}) & \\ \quad\cdots(\hat{a}_{N^2-N}+\cdots+\hat{a}_{N^2-1}), & \text{for closing,}\end{cases} \tag{16a}$$

where addition and multiplication represent the Boolean OR and AND operations, respectively, and $\hat{a}_i^1=a_i^1$ for all $i, 0\le i\le L-1$, and for each j, $2\le j\le p$

$$\hat{a}_i^j = \begin{cases} \hat{a}_i^j, & \text{if } a_i^m = b^m \text{ for } m = 1,2,\cdots,j-1 \\ a_i^r, & \text{if } a_i^m = b^m \text{ for m} = 1,2,\cdots,\text{r}-1 \\ & \quad \text{and for some } r,\ r \le j-1,\ a_i^r \neq b^r \end{cases} \tag{16b}$$

This property can be proved by a slight modification of the proof of Theorem 3 in [11]. In essence, this algorithm replaces A_i with a certain value which is greater (smaller) than A_i, once it becomes evident that A_i is greater (smaller) than the output g. Fig. 3. shows the logic network which produces the outputs of the binary morphological operators when $N=2$.

The following example illustrates the bit serial implementation of an FP opening using Proposition 3.

Example: Consider an FP opening operation of **f** by **k** given by $k(0)=1$ and $k(1)=2$. Suppose that the input samples are given by $f(n-1)=10$, $f(n)=6$, and $f(n+1)=4$. Using (15), the set of inputs to the min/max operator for FP opening is obtained: $W=\{A_0, A_1, A_2, A_3\}=\{6,3,11,6\}$ This set of data can be represented by the 4 bit binary codes:

A_0=6=(0110)

A_1=3=(0011)

A_2=11=(1011)

A_3=6=(0110)

The process of opening using Proposition 3 is summarized in Table 1. Note that the correct output value (0110)=6 is obtained, and A_1=(0011)=3, which is smaller than the output g=(0110)=6, is replaced with (0000) while A_2=11, which is greater than the output, is replaced with (1111)=15.

In Fig. 4, the output b^j in (16a) is calculated using the bit serial realization in conjunction with the binary morphological operators. In this figure, the binary morphological operators are implemented by using the logic network presented in Fig. 3. And the bit modification logics can be implemented by realizing the modification formula of (16b) in terms of a Boolean equation as follows:

$$a_i^j = \hat{a}_i^{j-1} M_i^j + a_i^j \overline{M}_i^j \tag{17}$$

Where $M_i^j = [\hat{a}_i^{j-1} \oplus b^{j-1}] + M_i^{j-1}$, for j≥2 and M_i^1 0. Here we represent the exclusive OR operation by ⊕ and the complement operation by (¯). Fig. 5 shows the logic network realizing the Boolean function in (17).

The main criteria for the evaluation of any VLSI implementation algorithm are delay time and the hardware complexity. The implementation based on the threshold decomposition property requires a logic network decomposing a p bit input signal to 2^p-1 binary signals, and 2^p-1 binary morphological operators, and a logic network regenerating the grayscale output signal from each outputs of the binary operators. Thus, the hardware complexity of the implementation based on the threshold decomposition property grows exponentially with the number of bits of the input signal. On the other hand, the implementation based on the bit serial approach requires one binary morphological operator, and L bit modification logic networks, and thus the hardware complexity grows linearly with the number of bits. Thus, the hardware of the bit serial architecture is simpler in structrue than that of the threshold decomposition based architecture. In the sense of the delay time, the threshold decomposition based implementation has the delay of $\tau_d+\tau_o+\tau_r$ where τ_d denotes the decomposition delay, τ_o the binary operation delay, and τ_r the

regeneration delay. The delay of bit-serial approach is $p\tau_o+(p-1)\tau_m$ where τ_m denotes the bit modification delay. Thus, the threshold decomposition based implementation is faster than the bit-serial approach. However, parallel use of the logic unit in Fig. 5 can speed up the computation of the bit-serial approach. The bit-level pipelined architecture for the bit-serial implementation is shown in Fig. 6. The delay experienced by data propagating through this architecture becomes $\tau_o+\tau_m$. Therefore, the bit-level pipelined architecture for the implementation of the FP operations can provide greatly increased data throughput rate.

V. CONCLUSION

Efficient real time software inplementation methods for the composite FP morphological operators were presented using the recursive structure based on the redundancy of the basis matrix and input matrices. If was shown that the composite morphological operators including opening and closing can be accomplished by a local matrix operation. It was also shown that, with the proposed fast algorithm, both opening and closing can be determined by 2N-2 additions and 2N-2 comparisons when the size of the GSE B equal to N. This fast implementation method can be directly extended to the implementation of composite operations such as the close-opening and open-closing operations. In the second part of this paper, a VLSI implementation architecture for grayscale morphological operators was presented. The proposed implementation architecture employed a bit-serial approach which decompose grayscale morphological operations into bit-level binary operation unit for the p-bit grayscale signal. It was shown that the bit-level pipelined architecture for the proposed scheme can greatly increase the data throughput rate.

REFERENCES

[1] G. Matheron, *Random Sets and Image Geometry*, New York:Wiley, 1975.

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

[3] C. R. Giardina and E. R. Dougherty, *Morphological Methods in Image and Signal Processing*, New Jersey: Prentice Hall, 1988.

[4] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology", *IEEE Trans. Patt. Analy. and Mach. Intell.*, vol. PAMI-9, pp. 523-550, Jul. 1987.

[5] P. Maragos and R. W. Schafer, "Morphological filters-Part I: Their set-theoretic analysis and relations to linear shift-invariant filters", *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. ASSP-35, pp. 1153-1169, Aug. 1987.

[6] P. Maragos, "A representation theory for morphological image and signal processing", *IEEE Trans Patt. Analy. and Mach. Intell.* vol. 11, no. 6, pp. 586-599, Jun. 1989

[7] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, Jr., "Stack filters", *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. ASSP-34, pp. 898-911, Aug. 1986.

[8] F.Y. Shih, and O. R. Mitchell, "Threshold decomposition of grayscale morphology into binary morphology", *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. PAMI-11, pp. 31-42, Jan. 1989.

[9] E. Ataman, V. K. Aatre, and K. M. Wong, "A fast method for real-time median filtering", *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. ASSP-28, pp. 415-421, Aug. 1980.

[10] R. T. Hoctor, and S. Kassam, "An algorithm and a pipelined architecture for orderstatistic determination and L-filterings," *IEEE Trans. Ciruits and Systems*, vol. CAS-36, pp. 344-351, Mar. 1989.

[11] K. Chen, "Bit-serial realization of a class of nonlinear filters based of positive boolean functions", *IEEE Trans. Ciruits and Systems*, vol. CAS-36, pp. 785-793, June 1989.

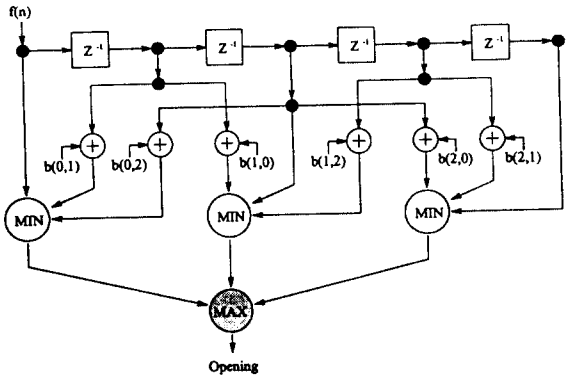
[12] S. J. Ko, Y. H. Lee, and A. T. Fam, "Efficient implementation of one-dimensional recursive median filters", *IEEE Trans. Circuits and Systems*, vol. CAS-37, pp. 1447-1450, Nov. 1990.

[13] J. P. Fitch, E. J. Colye, and N. C. Gallagher, Jr., "Median filtering by threshold decomposition", *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. ASSP-32, pp. 1183-1188, Dec. 1984.

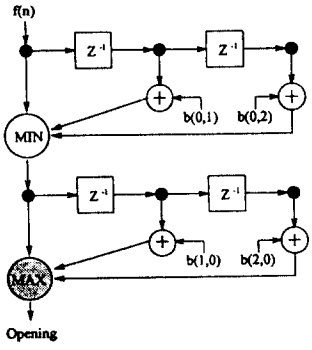
[14] R. G. Harber, S. C. Bass, and G. L. Neudeck, "VLSI implementation of a fast rank order algorithm", *Proc. ICASSP'85*, Tampa, FL., pp. 1396-1399, Mar. 1985.

Table 1. Original input bits, modified input bits, and output bits for the exadmple in Section III.

1	0010	0010	0
2	1001	1011	1
3	1111	1011	1
4	0110	0010	0



(a)



(b)

Fig. 1. The schematic diagram of opening, (a) signal flow graph of local matrix operator, (b) signal flow graph of fast recursive operator.

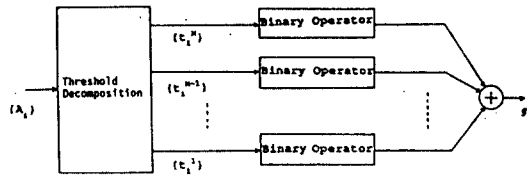


Fig. 2. Implementation of the FP system based on the threshold decomposition where

$$t_i^j = \begin{cases} 1, \text{ if } A_i \geq j, \\ 0, \text{ if } A_i < j. \end{cases}$$

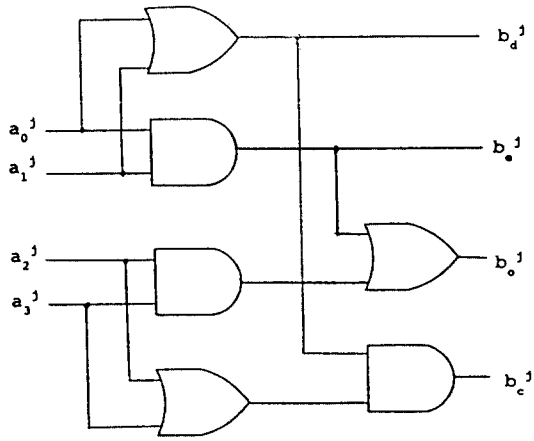


Fig. 3. Logic network producing the outputs of binary morphological operators, b_d^j(dilation), b_e^j(erosion), b_o^j(opening), and b_c^j(closing) where N=2.

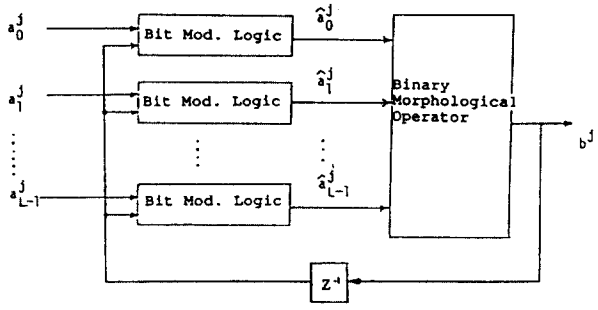


Fig. 4. Bit serial realization of the FP system with bit modification logics.

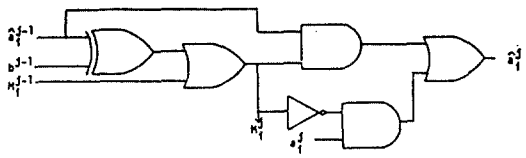


Fig. 5. Logic network for the bit modification operation.

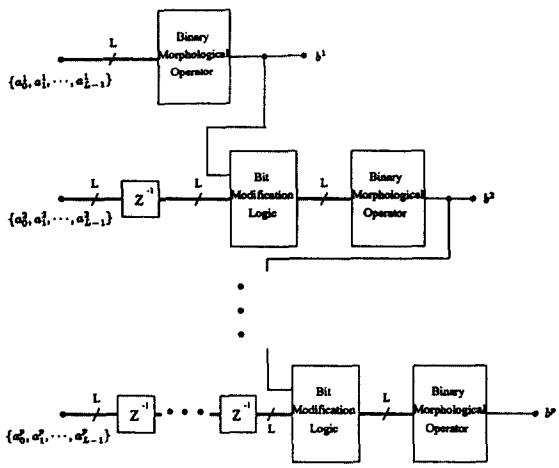


Fig. 6. The bit-level pipelined architecture for the bit-serial implementation of the FP morphological operator.

高 聖 濟 (Sung Jea Ko) 정회원

1980년 : 고려대학교 전자공학과 (공학학사)

1986년 : 미국 State University of New York at Buffalo, 전기 및 컴퓨터 공학(공학석사)

1988년 : 미국 State University of New York at Buffalo, 전기 및 컴퓨터 공학(공학박사)

1981년 8월 ~ 1983년 12월 : 대우 통신 컴퓨터 사업부 연구원

1988년 8월 ~ 1992년 5월 : 미국 The University of Michigan-Dearborn, 전기 및 컴퓨터 공학과 조교수

1992년 3월 ~ 현재 : 고려대학교 전자공학과 부교수

※주관심분야 : 디지탈 신호 및 영상처리, 패턴 인식 및 신경망 회로, 전자 제어 및 Car Electronics.

李境勳(Kyung Hoon Lee) 정회원

1992년 2월 : 고려대학교 전자공학과(공학사)

1994년 9월 : 고려대학교 전자공학과(공학석사)

1994년 9월 : 고려대학교 전자공학과 박사과정입학예정

※주관심분야 : 디지탈 신호 및 영상처리, 패턴 인식 및 신경망 회로.