

# Realisation of soft morphological filters

A. Gasteratos  
I. Andreadis  
Ph. Tsalides

*Indexing terms: Mathematical morphology, Soft morphological filters, Algorithms*

**Abstract:** A new technique for the realisation of soft morphological filters based on the majority gate algorithm is presented. A pipelined systolic array architecture suitable to perform soft morphological filtering is also presented. The processing times of the proposed hardware structure do not depend on the data window size and the required silicon area is linearly related to the number of its inputs and the image resolution.

## 1 Introduction

Mathematical morphology is a powerful tool for solving image processing problems, such as image enhancement, noise removal, feature extraction, texture analysis etc. [7]. Morphological operations have been viewed as filters of which the properties have been studied [1, 8]. Another well-studied class of nonlinear filters are rank order filters [5]. Soft morphological filters are a combination of morphological and weighted rank order filters [3], which have been introduced to improve the behaviour of traditional morphological filters in noisy environments. The idea was to slightly relax the typical morphological definitions in such a way that a degree of robustness is achieved, while most of the desirable properties of typical morphological operations are maintained. Soft morphological filters are less sensitive to additive noise and to small variations in object shape than typical morphological filters. They can remove positive and negative impulse noise, preserving at the same time small details in images. Also, some soft morphological filters are not idempotent; idempotency is a desirable property in signal processing.

Soft morphological filters form a relatively new class of nonlinear filters, and therefore, only a few designing methods have been proposed [3]. Since they can be considered as a special case of weighted rank order filters, design methodologies which have been originally proposed for the realisation of rank order filters may be applied to soft morphological filters. Soft morphological filters can be implemented in VLSI using the threshold decomposition technique [9]. However, in

this approach hardware complexity increases exponentially with both the resolution of the numbers and the size of the data window. Therefore, implementation of filters capable of handling high resolution numbers is not practical. An efficient algorithm for median filtering has been presented in [4]. This algorithm has been modified to be applied into morphological filtering, through replacement of the majority gate by max/min computation circuits [2]. In this paper a novel realisation of soft morphological filters based on the modified majority gate algorithm is presented. The architecture of the proposed hardware structure is scalable; its hardware complexity is linearly related both to the number of its inputs and image resolution. Also, its processing times do not depend on the data window size.

## 2 Soft morphological operations

In soft morphological operations the max/min operations, used in standard morphology, are replaced by weighted order statistics. Also, the structuring element  $B$  is divided into two subsets: the core  $A$  and the soft boundary  $B \setminus A$ , where ' $\setminus$ ' stands for the set difference. In soft morphological dilation (erosion) of a function the pixels of the function are combined with the pixels of the structuring element as in traditional morphology; the results related to the soft boundary and to the core (repeated  $k$  times) are ordered in descending (ascending) sequence. The  $k$ th element of this sequence is the result of soft morphological dilation (erosion). Let  $\{k \diamond f(x)\}$  denote the  $k$  time repetition of  $f(x)$ ;  $\{k \diamond f(x)\} = \{f(x), f(x), \dots, f(x)\}$  ( $k$  times). The soft morphological dilation of a grey-scale image  $f$  by a soft grey-scale structuring element  $[a, \beta, k]$  is defined as [6]:

$$f \oplus [\beta, a, k](z) = k\text{th largest of} \\ (\{k \diamond (f(y) + a(z - y))\} \cup \{f(b) + \beta(z - b)\}) \quad (1)$$

where  $(z - y) \in A$  and  $(z - b) \in B \setminus A$ .

Also, the soft morphological erosion of  $f$  by  $[a, \beta, k]$  is defined as:

$$f \ominus [\beta, a, k](z) = k\text{th smallest of} \\ (\{k \diamond (f(y) - a(z + y))\} \cup \{f(b) - \beta(z + b)\}) \quad (2)$$

where  $(z + y) \in A$  and  $(z + b) \in B \setminus A$ .

From the above definitions it becomes clear that the computation of soft morphological dilation/erosion requires adders/subtractors. This should be followed by a circuit which computes the required order statistic of the results. A unit based on a median computation algorithm (to be described in the next section) has been

© IEE, 1998

*IEE Proceedings* online no. 19981743

Paper first received 12th May and in revised form 17th October 1997

The authors are with the Laboratory of Electronics, Section of Electronics and Information Systems Technology, Department of Electrical and Computer Engineering, Democritus University of Thrace, GR-67100 Xanthi, Greece

used. To make this unit capable of computing any order statistic, dummy inputs have been introduced.

### 3 Algorithm description

Suppose that there are  $W = 2N + 1$  numbers  $x_i$  of  $n$ -bit resolution, where  $0 < i \leq W$  and  $N$  is a positive integer number, of which the median value is required. The median  $m$  of numbers  $x_i$  is:

$$m = \sum_{j=1}^n o_j 2^{n-j} \quad (3)$$

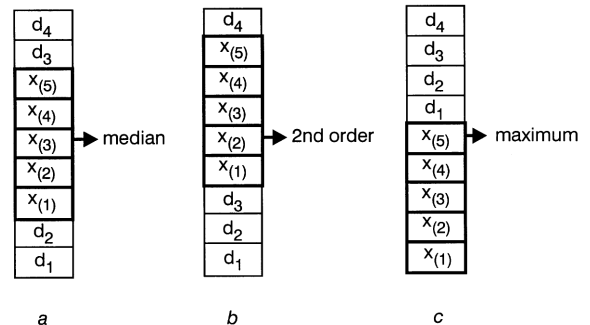
where  $o_j$  is the  $j$ th bit of the binary representation of  $m$ .

The following flags and intermediate signals are defined:

- $r_{i,j}$  is the rejecting flag signal, which indicates whether the number  $x_i$  remains within the subset of the candidate numbers to be the median, in the  $j$ th step of the algorithm. When  $r_{i,j}$  is '1'  $x_i$  remains within the subset, otherwise  $x_i$  is rejected. Once  $r_{i,j}$  is set to '0' it remains constant in the successive stages. Also, the remaining bits  $b_{i,j}$  are not taken into account.  $b_{i,j}$  is the  $j$ th bit of the binary representation of  $x_i$ .
- $t_{i,j}$  is the setting flag signal, which replaces the  $b_{i,j}$  bits of the rejected numbers in the majority selection process.  $t_{i,j}$  is set to the complementary value of the previous output bit  $o_{j-1}$ , when the number has been rejected. In this way, the number which has been rejected is pushed away from the median value. If the state of  $r_{i,j}$  has not been changed to '0' the setting flag is in a 'don't care' state.
- $i_{i,j}$  is an intermediate signal, which is either  $b_{i,j}$  (if  $x_i$  has not been rejected), or  $t_{i,j}$ . The output bit  $o_j$  is '1' if the majority of  $i_{i,j}$  is '1' otherwise it is '0'.

An efficient algorithm for median value computation follows [4]. The MSBs of the numbers in the list are processed first. The other bits are then processed sequentially until the LSBs are reached. Initially, the rejecting flag signals  $r_{i,1}$  are set to '1', since all the num-

bers are candidates to be the median. The setting flags  $t_{i,1}$  are in a 'don't care' state. If the majority of the MSBs  $b_{i,1}$  is found to be '1', then the MSB of the output is  $o_1 = '1'$ , otherwise  $o_1 = '0'$ . In the next stage the bits  $b_{i,2}$  of the numbers which have MSBs complementary to  $o_1$  are rejected and are not taken into account for the majority selection. In this case, signals  $t_{i,2}$  corresponding to the rejected numbers are considered. The majority selection procedure continues and the median  $m$  is obtained by collecting bits  $o_j$  according to eqn. 3.



**Fig. 1** Illustration of use of dummy inputs  
*a* Median  
*b* 2nd order  
*c* Maximum computation

To compute any order statistic of a sequence of numbers, additional dummy numbers are introduced. The concept is that by having a method to compute the median of a sequence of numbers and by being able to control the dummy numbers, any order statistic of the original sequence can be determined. Figs. 1*a*, *b* and *c* illustrate this concept. The dummy numbers are used in the computation of median, second-order statistic and maximum values, respectively. The bold window contains five binary numbers  $x_{(1)}, x_{(2)}, \dots, x_{(5)}$  in ascending order (the subscript in the parentheses denotes the rank). The larger window contains nine binary numbers also in ascending order. By controlling the number of the dummy binary numbers  $d_b$ , which are pushed to the top and to the bottom, any order statistic  $r$  of the

**Table 1: Computation of the 4th order statistic of nine 4-bit numbers**

| $j$        | 1         |           | 2         |           | 3         |           | 4         |           |           |           |           |           |           |           |           |           |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
|            | $b_{i,1}$ | $r_{i,1}$ | $t_{i,1}$ | $i_{i,1}$ | $b_{i,2}$ | $r_{i,2}$ | $t_{i,2}$ | $i_{i,2}$ | $b_{i,3}$ | $r_{i,3}$ | $t_{i,3}$ | $i_{i,3}$ | $b_{i,4}$ | $r_{i,4}$ | $t_{i,4}$ | $i_{i,4}$ |
| $x_1 = 6$  | 0         | 1         | X         | 0         | 1         | 1         | X         | 1         | 1         | 1         | X         | 1         | 0         | 0         | 1         | 1         |
| $x_2 = 3$  | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         | 1         | 0         | 0         | 0         | 1         | 0         | 0         | 0         |
| $x_3 = 4$  | 0         | 1         | X         | 0         | 1         | 1         | X         | 1         | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         |
| $x_4 = 5$  | 0         | 1         | X         | 0         | 1         | 1         | X         | 1         | 0         | 1         | X         | 0         | 1         | 1         | X         | 1         |
| $x_5 = 10$ | 1         | 1         | X         | 1         | 0         | 0         | 1         | 1         | 1         | 0         | 1         | 1         | 0         | 0         | 1         | 1         |
| $x_6 = 13$ | 1         | 1         | X         | 1         | 1         | 0         | 1         | 1         | 0         | 0         | 1         | 1         | 1         | 0         | 1         | 1         |
| $x_7 = 2$  | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         | 1         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $x_8 = 10$ | 1         | 1         | X         | 1         | 0         | 0         | 1         | 1         | 1         | 0         | 1         | 1         | 0         | 0         | 1         | 1         |
| $x_9 = 8$  | 1         | 1         | X         | 1         | 0         | 0         | 1         | 1         | 0         | 0         | 1         | 1         | 0         | 0         | 1         | 1         |
| $d_1 = 0$  | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $d_2 = 0$  | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $d_3 = 0$  | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $d_4 = 0$  | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $d_5 = 0$  | 0         | 1         | X         | 0         | 0         | 1         | X         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $d_6 = 15$ | 1         | 1         | X         | 1         | 1         | 0         | 1         | 1         | 1         | 0         | 1         | 1         | 1         | 0         | 1         | 1         |
| $d_7 = 15$ | 1         | 1         | X         | 1         | 1         | 0         | 1         | 1         | 1         | 0         | 1         | 1         | 1         | 0         | 1         | 1         |
| $d_8 = 15$ | 1         | 1         | X         | 1         | 1         | 0         | 1         | 1         | 1         | 0         | 1         | 1         | 1         | 0         | 1         | 1         |
| $o_j$      | 0         |           |           |           | 1         |           |           |           | 0         |           |           |           | 1         |           |           |           |

numbers  $x_i$  can be obtained. An illustrative example of this algorithm is depicted in Table 2. In this particular example the 4th-order statistic of nine 4-bit numbers is computed (a 'don't care' state is denoted by X).

#### 4 Systolic array implementation for soft morphological filtering

##### 4.1 A systolic array for a $(3 \times 3)$ structuring element

A pipelined systolic array capable of computing soft grey scale dilation/erosion on a  $(3 \times 3)$  image window using a  $(3 \times 3)$  structuring element, both of 8-bit resolution, is presented in this Section. The central pixel of the structuring element is its core, whereas the rest eight pixels are the soft boundary. The systolic array is shown in Fig. 2. The inputs to this array are the 9 pixels of the image window, the 9 pixels of the soft morphological structuring element and a control signal MODE. Latches (L1) hold the image window, latches (L\*1) hold the structuring element and latch (L\*\*1) holds number  $k$ . Signal MODE is used to select the operation. When this is '1' soft dilation is performed, whereas when it is '0' soft erosion operation is performed. Image data is collected through multiplexers MUX1, which are controlled by the signal MODE. The pixels of structuring element remain either unchanged for the operation of dilation or they are complemented (by means of XNOR gates) for the operation of erosion. In the next stage of the pipeline, data is fed to nine adders. In the case of soft erosion the 2's complements of the pixels of the structuring element are added to the image pixels. This is equivalent to the subtraction operation.

When  $k > \text{Card}(B \setminus A)$ , the soft morphological operations are reduced and only the core of the structuring element is considered [9]. Therefore to preserve the nature of soft morphological operations, the constraint  $k \leq \min\{\text{Card}(B)/2, \text{Card}(B \setminus A)\}$  is used. In this case  $k$  is in the range  $1 \leq k \leq 4$ . Table 2 shows the number of the elements, contained in the list, as well as the number of the dummy elements. For soft dilation all

the dummy inputs are pushed to the top, whereas for soft erosion they are pushed to the bottom. Thus, the appropriate result is obtained from the order statistic unit. A control unit controls an array of multiplexers MUX2 (its input is number  $k$ ). This is a decoder and its truth table is provided in Table 3. It provides the input to the order statistic unit, either a dummy number or a copy of the addition/subtraction result of the core. The order statistic unit, consists of identical processing elements (PEs) separated by latches (L\*\*4 to L\*\*11). The resolution of the latches which hold the addition/subtraction results or the dummy numbers (L3 to L11), decreases by one bit in each successive stage, since there is no need to carry the  $b_{i,j}$  coefficients which have been already processed. On the other hand, the resolution of the latches which hold the result (L4\* to L\*11), increases by one bit in each successive stage.

Table 2: Use of dummy numbers in the computation of weight order statistics

| $k$ | Sequence of numbers | Dummy numbers |
|-----|---------------------|---------------|
| 1   | 9                   | 8             |
| 2   | 10                  | 7             |
| 3   | 11                  | 6             |
| 4   | 12                  | 5             |

Table 3: Truth table of the control unit

| Input | Outputs |    |    |    |    |    |    |    |
|-------|---------|----|----|----|----|----|----|----|
| $k$   | i1      | i2 | i3 | i4 | i5 | i6 | i7 | i8 |
| 0001  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0010  | 1       | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0011  | 1       | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0100  | 1       | 1  | 1  | 0  | 0  | 0  | 0  | 0  |

For the realisation of the PEs, the definitions of Section 3 are used. From these definitions and through logic operations (Karnaugh maps) the following

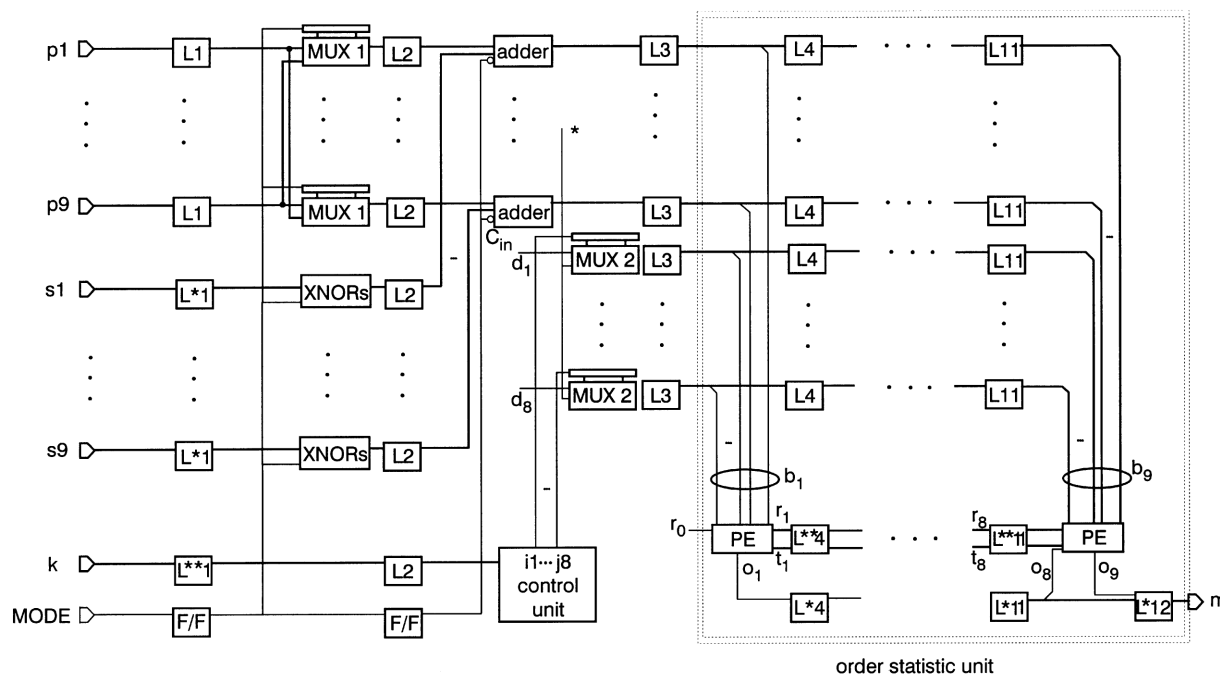


Fig. 2 Systolic array hardware structure implementing the majority gate technique for soft morphological filtering  
 $p_i$  = image data;  $s_i$  = structuring element data;  $d_i$  = dummy numbers; \* from the adder corresponding to the core pixel

equations for  $r_{i,j+1}$ ,  $i_{i,j}$  and  $t_{i,j+1}$ , are obtained:

$$r_{i,j+1} = r_{i,j} \cdot (o_j \cdot b_{i,j} + \overline{o_j} \cdot \overline{b_{i,j}}) = r_{i,j} \cdot (o_j \otimes b_{i,j}) \quad (4)$$

where  $\cdot$ ,  $+$ ,  $\overline{\phantom{x}}$  and  $\otimes$  stand for logic AND, OR, NOT and XNOR, respectively.

$$i_{i,j} = r_{i,j} \cdot b_{i,j} + \overline{r_{i,j}} \cdot t_{i,j} \quad (5)$$

and

$$t_{i,j+1} = (\overline{r_{i,j}} \cdot t_{i,j}) + (r_{i,j} \cdot b_{i,j}) \quad (6)$$

The circuit diagram of this PE, based on the above equations, is shown in Fig. 3. In this Figure  $W = 4N + 1$ ; the  $2N + 1$  inputs are the numbers  $x_p$  whereas the rest are the dummy inputs.

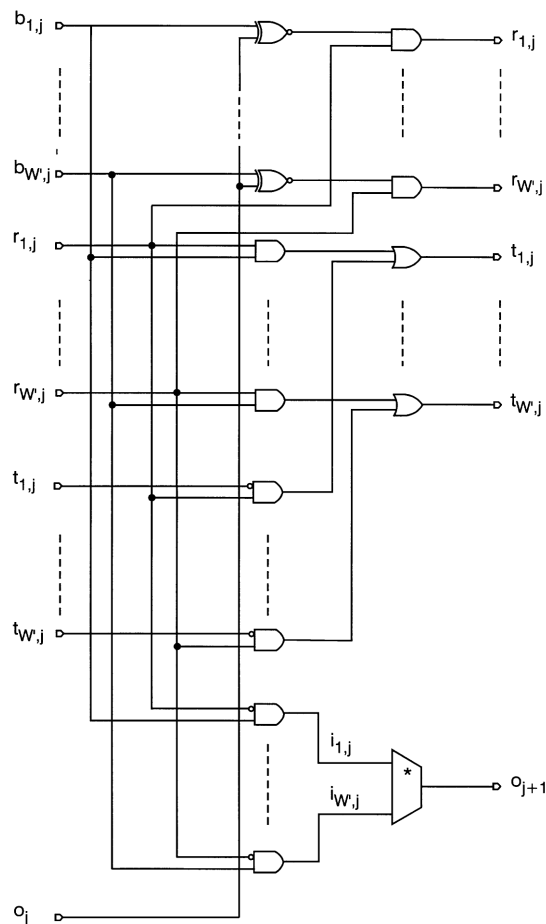


Fig. 3 The basic processing element (PE)  
\* majority gate

The threshold decomposition approach to soft morphological filtering in [9] for a 6-bit image and structuring element requires a decoder, which transforms the  $(3 \times 3)$  pixel image window into  $2^6$   $(3 \times 3)$  binary image windows,  $2^6$  neighbourhood generators (these form the neighbourhood according to  $k$ ),  $2^6$  parallel counters of 12 input bits ( $\text{Card}(B/A) + k\text{Card}(A) = 8 + 4 = 12$ ),  $2^6$  comparators which compare the output of the counter with  $k$  and an adder (this adds  $2^6$  1-bit numbers) to produce the grey-scale result. Moreover, for an 8-bit image and structuring element the required components are: a decoder, which produces  $2^8$  binary image windows,  $2^8$  neighbourhood generators,  $2^8$  parallel counters of 12 input bits,  $2^8$  comparators and an adder with  $2^8$  1-bit inputs.

Thus, in this approach, there is an exponential relationship between image resolution and hardware complexity. On the contrary, in the proposed technique if

the resolution of the image is increased by two bits, then only two extra stages in the order statistic unit (Fig. 2) are required. Therefore, hardware complexity in this case is linearly related to image resolution. Similarly, it can be shown that in the threshold decomposition approach hardware complexity is exponentially related to the number of the inputs. From Figs. 2 and 3 it can be deduced that the hardware complexity of the proposed structure is linearly related to the number of its inputs. More specifically, for each additional input a MUX1, eight XOR gates and an adder are required. The PE (Fig. 3) is altered by adding four AND gates, one XNOR gate, one OR gate and two inverters. Also, the majority gate requires two extra transistors. Furthermore, owing to the simplicity of the PE (there are only three stages of gates including the inverters), the processing element can attain very short processing times, independent of the data window size. Thus, it is obvious that the technique described in this paper is superior both in terms of silicon area and speed.

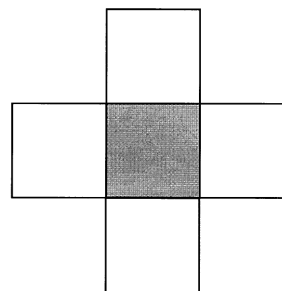


Fig. 4 Cross-shaped structuring element

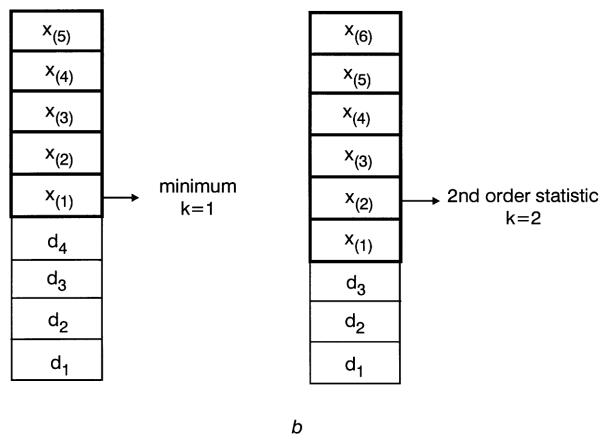
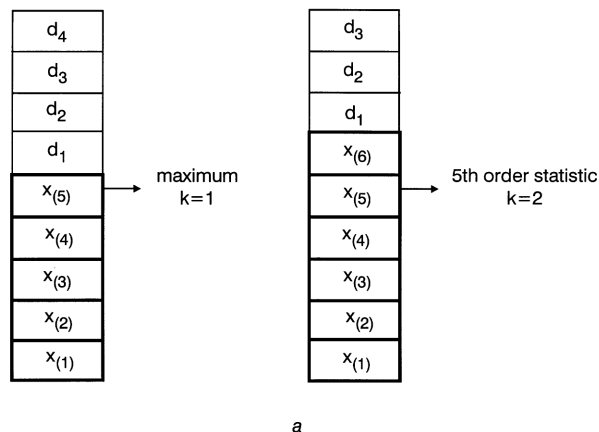


Fig. 5 Arrangement of the dummy numbers in soft morphological dilation and erosion using the structuring element of Fig. 4  
a Soft morphological dilation  
b Soft morphological erosion

## 4.2 Order statistic unit hardware requirements for other structuring elements

In this subsection the hardware requirements for the order statistic unit of other structuring elements are presented. The rest of the circuitry consists of a number of adders/subtractors equal to the number of pixels of the structuring element. Fig. 4 illustrates a cross-shaped structuring element. The shaded area represents the core, whereas the rest represents the soft boundary. In this case:  $\text{Card}(B) = 5$ ,  $\text{Card}(A) = 1$ ,  $\text{Card}(B \setminus A) = 4$  and  $k \leq \min\{2.5, 4\}$ , i.e.  $1 \leq k \leq 2$ . Fig. 5a illustrates the position of the dummy numbers in soft dilation for  $k = 1$  and  $k = 2$ . Fig. 5b illustrates the position of the dummy numbers in soft erosion.

The next example illustrates a more complex situation. Figs. 6 and 7 show a structuring element defined as the one of Figs. 4 and 5. In this case:  $\text{Card}(B) = 12$ ,  $\text{Card}(A) = 8$ ,  $\text{Card}(B \setminus A) = 4$  and  $k \leq \min\{6, 4\}$ , i.e.  $1 \leq k \leq 4$ . When  $k = 4$  the maximum number of the elements of the sequence to be ordered is  $\text{Card}(B \setminus A) + k\text{Card}(A) = 36$ . The 33rd (4th)-order statistic of this sequence is searched. Thus, the total number of the

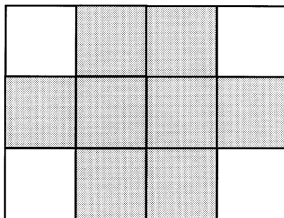


Fig. 6 Structuring element

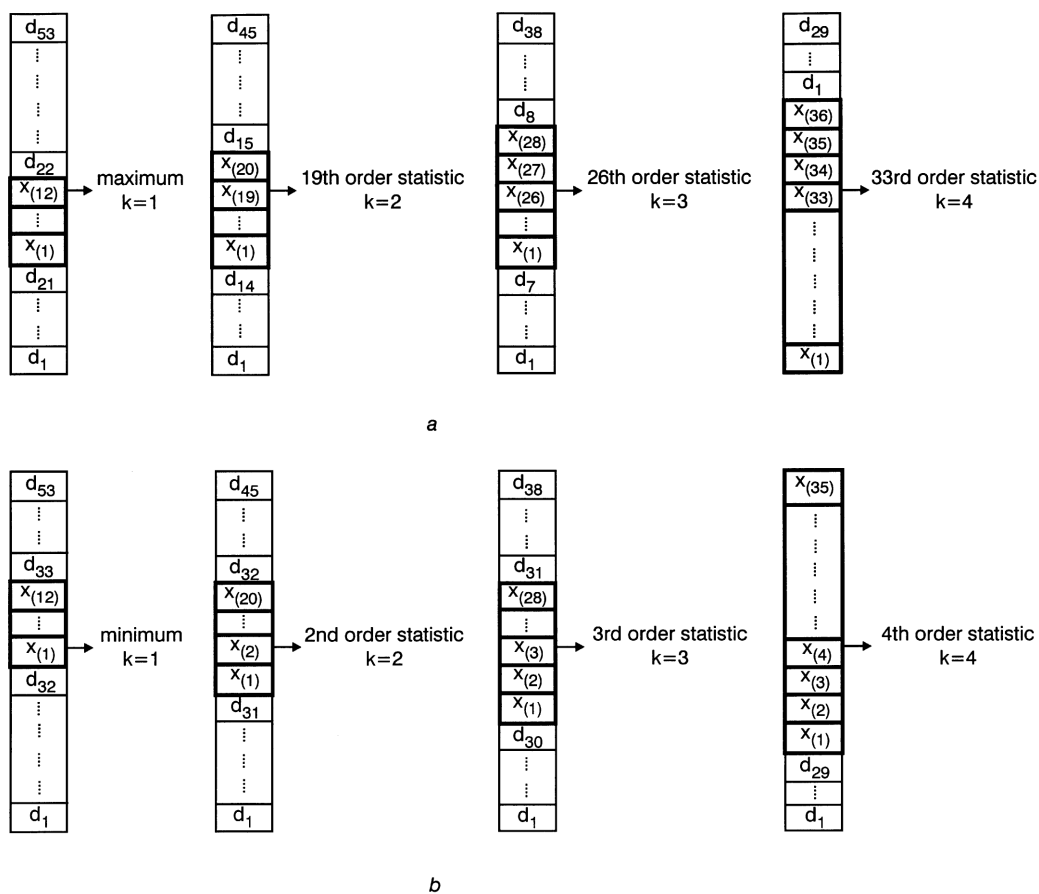


Fig. 7 Arrangement of the dummy numbers in soft morphological dilation and erosion using the structuring element of Fig. 6  
a Soft morphological dilation  
b Soft morphological erosion

inputs to the order statistic unit is 65. The dummy numbers which are pushed to the top (bottom), in the operation of soft dilation (erosion), are 29. When  $k = 3$ , the numbers of the sequence are 28 and the 26th (3rd)-order statistic is searched. Now the dummy numbers which are pushed to the top (bottom) are 30 and to the bottom (top) are 7. In the same way, when  $k = 2$  the numbers of the sequence are 20 and the 19th (2nd)-order statistic is searched and the dummy numbers which are pushed to the top (bottom) are 31 and to the bottom (top) are 14. Finally, when  $k = 1$  the numbers of the sequence are 12 and the 12th (1st)-order statistic is searched. In this case the dummy numbers which are pushed to the top (bottom) are 32 and to the bottom (top) are 21. For any structuring element an order statistic unit can be constructed following the above procedure.

## 5 Conclusions

Soft morphological filters form a new class of nonlinear filters and only a few designing methods exist. The reported threshold decomposition technique becomes prohibitively expensive when typical images (8-bit resolution) are processed. In this paper a new systolic array architecture (based on the majority gate) suitable to perform soft morphological filtering has been presented. The processing times of the proposed hardware structure are independent of the data window size and its hardware complexity is linearly related both to the number of its inputs and the resolution of the image.

## 6 References

- 1 DOUGHERTY, E.R. (Ed.): 'Digital image processing methods' (Marcel Dekker Inc., 1994)
- 2 GASTERATOS, A., ANDREADIS, I., and TSALIDES, PH.: 'Improvement of the majority gate algorithm for grey scale dilation/erosion', *Electron. Lett.*, 1996, 32, (9), pp. 806-807
- 3 KUOSMANEN, P., and ASTOLA, J.: 'Soft morphological filtering', *J. Math. Imaging Vis. (Netherlands)*, 1995, 5, (3), pp. 231-262
- 4 LEE, C.L., and JEN, C.W.: 'Bit-sliced median filter design based on majority gate', *IEE Proc.-G*, 1992, 139, (1), pp. 63-71
- 5 PITAS, I., and VENETSANOPOULOS, A.N.: 'Nonlinear digital filters - principles and applications' (Kluwer Academic Publishers 1990)
- 6 PU, C.C., and SHIH, F.Y.: 'Threshold decomposition of grey-scale soft morphology into binary soft morphology', *CVGIP, Graph. Models Image Process.*, 1995, 57, (6), pp. 522-526
- 7 SERRA, J.: 'Image analysis and mathematical morphology - Vol. I' (Academic Press, 1982)
- 8 SERRA, J. (Ed.): 'Image analysis and mathematical morphology - Vol II, theoretical advantages' (Academic Press, 1988)
- 9 SHIH, F.Y., and PU, C.C.: 'Analysis of the properties of soft morphological filtering using threshold decomposition', *IEEE Trans. Signal Proc.*, 1995, 43, (2), pp. 539-544