Pergamon

Pattern Recognition, Vol. 30, No. 9, pp. 1571–1576, 1997

0031-3203/97 $17.00+.00

PII: S0031-3203(96)00170-7

REALIZATION OF RANK ORDER FILTERS BASED ON MAJORITY GATE

A. GASTERATOS, I. ANDREADIS* and Ph. TSALIDES

Laboratory of Electronics, Section of Electronics and Information Systems Technology, Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 671 00, Greece

(*Received* 3 *July* 1996; *received for publication* 23 *October* 1996)

Abstract—A new technique for the implementation of a single hardware structure capable of computing any rank order filter is presented in this paper. The proposed technique, which is based on the majority gate, achieves faster extraction of setting flag signals and, therefore, shorter processing times are attained. A pipelined systolic array, suitable for performing rank order filtering, is also presented. Applications of rank order filters include digital image processing, speech processing and coding and digital TV applications. © 1997 Pattern Recognition Society. Published by Elsevier Science Ltd.

Nonlinear filters Computer vision

1. INTRODUCTION

Rank order filters are a class of nonlinear filters. The input of such filters is a window of data with an odd number of elements. These elements are sorted in ascending order and the output of the rank order filter with rank r is the rth element (rth order statistic).[1] Special cases of rank order filters are median, minimum and maximum filters, where the outputs are the median, the minimum and the maximum values of the input data window, respectively. Rank order filters exhibit excellent robustness properties and provide solutions in many cases where linear filters are inappropriate. They can suppress high-frequency and impulse noise in an image, avoiding at the same time extensive blurring of the image, since they have good edge preservation properties. They have found numerous applications, such as in digital image analysis, in speech processing and coding, in digital TV applications, etc.[1,2]

Median and rank order filters are strongly related with morphological filters, another class of nonlinear filters.[3,4] It has been shown that erosions and dilations are special cases of rank order filters and that any rank order filter can be expressed either as a maximum of erosions or as a minimum of dilations.[3] Therefore, algorithms which originally have been devised for rank order and median filters can be used for realization of morphological operators.[5,6]

Several algorithms have been proposed for the realization of rank order filters, such as tree sorts, shell sorts and quick sorts.[1] Although these algorithms are suitable for software implementation, they result in inefficient hardware structures, since they handle the numbers in word-level. Rank order filters can be implemented in VLSI using the threshold decomposition technique.[7–9] However, in this case 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. Bit-sliced algorithms suitable for hardware implementation have been proposed.[10,11] In these implementations, the numbers are handled in bit-level in order to obtain local minima and maxima, from which the rth order statistic is obtained. However, several building blocks are required to implement the local minima and maxima functions, and, thus, the hardware complexity increases.

Different hardware structures of an efficient algorithm for rank order filters have been presented.[12–14] These are based on the selection of intermediate signals through a device which gives output "1" if the number of its inputs which are "1" is greater or equal to the rank of the filter, otherwise its output is "0". This device has been implemented using the following techniques:

1. Comparison after summation,[12]
2. Positive Boolean Functions (PBF)[13] and
3. CMOS programmable device.[14]

The device for the median computation (majority gate) is shown in Fig. 1. It is a nonlinear voltage divider, built by output-wired inverters and an inverting buffer. The latter approach is the most advantageous in terms of silicon area among the three. It replaces either an N 1-bit binary tree adder and a $[\log_2(N)+1]$-bit comparator (comparison after summation technique) or $N+1$ gates of $N-1$ inputs (PBF technique), with $2N+2$ transistors. However, once the device has been designed, the rank order of the filter is fixed (PBF and CMOS programmable device approaches). In the comparison after summation approach, it is possible to implement any rank order

* Author to whom correspondence should be addressed. E-mail: ioannis@orfeas.ee.duth.gr.

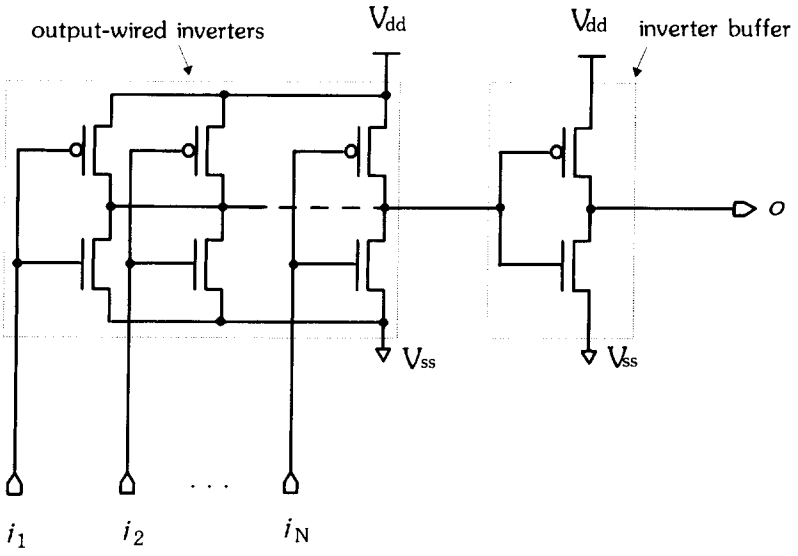

Fig. 1. Device programmable CMOS majority gate.

filter by controlling the second input of the comparator. However, this technique lacks in terms of both silicon area and operation speed.

In this paper the majority gate technique, based on the CMOS programmable device, has been modified and the proposed hardware structure is capable of computing any rank order filter. A pipelined systolic array suitable for performing rank order filtering is also proposed. Furthermore, efficient extraction of setting flag signals resulted in a faster hardware structure.

2. MEDIAN VALUE COMPUTATION ALGORITHM

In bit-sliced algorithms, bits of different significance are handled by dedicated Processing Elements (PEs) in different stages of the process. The process starts with the Most Significant Bits (MSBs). Flag signals derived from previous stages are used for further processing in the successive stages. The process is based on the majority selection of intermediate signals. The selection is achieved using a majority gate, which operates as follows: its output is "1" if over half of its inputs are "1", otherwise its output is "0".

2.1. Definitions and notations

Suppose that the total number of the data window elements x_i is $W = 2N + 1$, where $0 < i \leq W$ and N is a positive integer. The median of numbers x_i is

$$m = \text{med}(x_1, x_2, \ldots, x_W). \tag{1}$$

x_i numbers are represented in binary form of k-bit resolution. Suppose that $b_{i,j}$ is the jth bit of the binary representation of x_i, then

$$x_i = \sum_{j=1}^{k} b_{i,j} 2^{k-j}. \tag{2}$$

Also suppose that o_j is the jth bit of the binary representation of m, then

$$m = \sum_{j=1}^{k} o_j 2^{k-j}. \tag{3}$$

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 jth step of the algorithm. When $r_{i,j}$ is "1", number x_i remains within the subset, whereas when $r_{i,j}$ is "0", number x_i is rejected from the subset. Once $r_{i,j}$ is set to "0" it remains constant in the successive stages and the remaining bits b_i are not taken into account.
- $t_{i,j}$ is the setting flag signal, which replaces the $b_{i,j}$ bits of the rejected numbers in the majority selection process. The setting flag signal $t_{i,j}$ 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 the rejecting flag $r_{i,j}$

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 $t_{i,j}$ if x_i has been rejected. The output bit o_j is "1" if the majority of $i_{i,j}$ is "1", otherwise it is "0".

2.2. *Algorithm description*

The median value computation procedure follows.[12–14] The MSBs of the numbers within the data window are first processed. The other bits are then processed sequentially until the Less Significant Bits (LSBs) are reached. Initially, the rejecting flag signals $r_{i,1}$ are set to "1" since all the numbers are candidates to be the median value. The setting flags $t_{i,1}$ are in a "don't care" state. If the majority of the MSBs $b_{i,1}$ are found to be "1", then the MSB of the output is o_1="1", otherwise o_1="0". In the following stage the bits $b_{i,2}$ of the numbers which have MSBs complement to o_1 are rejected and are not taken into account for the majority selection. Instead of them the signals $t_{i,2}$, which correspond to the rejected numbers, are taken into account. The majority selection procedure continues in the next stages and the median value m is obtained by collecting bits o_j according to equation (3).

3. ALGORITHM FOR ANY RANK FILTER IMPLEMENTATION

The previously described algorithm can be easily implemented in hardware. However, it should be noticed that the rank order of this hardware structure is fixed after the design of the majority gate.[14] In this section, a new hardware implementation technique based on the majority gate is presented. This technique implements a single hardware structure capable of computing any rank order filter. Suppose that there are $W = 2N + 1$ numbers x_i , the rth order statistic of which is required. A hardware module similar to the one shown in Fig. 1 having $W' = 4N + 1$ inputs implementing the median computa-

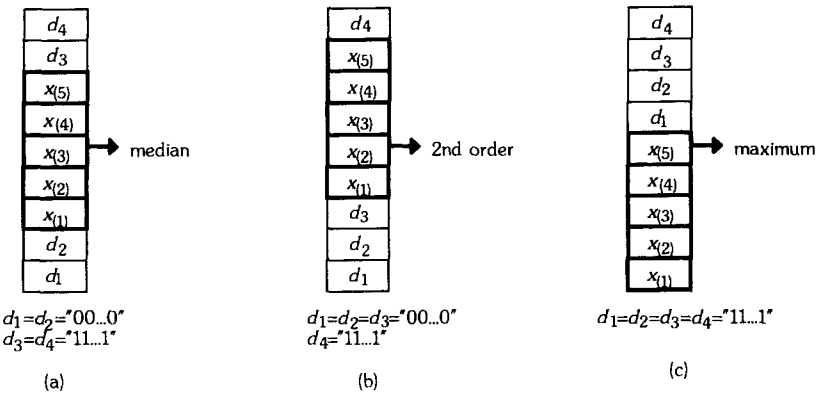

Fig. 2. Illustration of dummy input use.

Table 1. Computation of the second-order statistic of nine 4-bit numbers

j	1			2			3			4		
	b	r	t	b	r	t	b	r	t	b	r	t
x_1=5	0	1	X	1	1	X	0	0	1	1	0	1
x_2=6	0	1	X	1	1	X	1	0	1	0	0	1
x_3=3	0	1	X	0	1	X	1	1	X	1	1	X
x_4=13	1	1	X	1	0	1	0	0	1	1	0	1
x_5=4	0	1	X	1	1	X	0	0	1	0	0	1
x_6=2	0	1	X	0	1	X	1	1	X	0	1	X
x_7=10	1	1	X	0	0	1	1	0	1	0	0	1
x_8=12	1	1	X	1	0	1	0	0	1	0	0	1
x_9=8	1	1	X	0	0	1	0	0	1	0	0	1
d_1=0	0	1	X	0	1	X	0	1	X	0	0	0
d_2=0	0	1	X	0	1	X	0	1	X	0	0	0
d_3=0	0	1	X	0	1	X	0	1	X	0	0	0
d_4=0	0	1	X	0	1	X	0	1	X	0	0	0
d_5=0	0	1	X	0	1	X	0	1	X	0	0	0
d_6=0	0	1	X	0	1	X	0	1	X	0	0	0
d_7=0	0	1	X	0	1	X	0	1	X	0	0	0
d_8=15	1	1	X	1	0	1	1	0	1	1	0	1
o		0			0			1			1	

Table 2. The truth table for $r_{i,j+1}$, $i_{i,j}$ and $t_{i,j+1}$

$r_{i,j}$	$b_{i,j}$	$t_{i,j}$	o_j	$r_{i,j+1}$	$i_{i,j}$	$t_{i,j+1}$
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	1	1
0	1	1	1	0	1	1
1	0	0	0	1	0	X
1	0	0	1	0	0	0
1	0	1	0	1	0	X
1	0	1	1	0	0	0
1	1	0	0	0	1	1
1	1	0	1	1	1	X
1	1	1	0	0	1	1
1	1	1	1	1	1	X

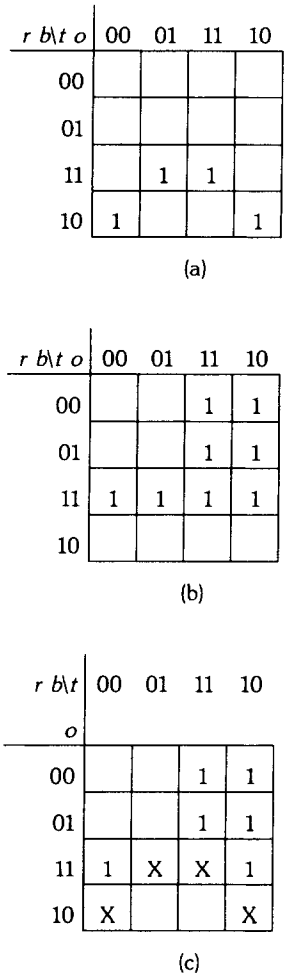

Fig. 3. The Karnaugh maps for $r_{i,j+1}$, $i_{i,j}$ and $t_{i,j+1}$.

tion (described in Section 2.2) is used. The $2N+1$ inputs are the numbers x_i, whereas the rest are dummy inputs d_l ($0 < l \leq 2N$). The binary values of the dummy inputs d_l can be either "00..0" or "11..1". This implies that if the

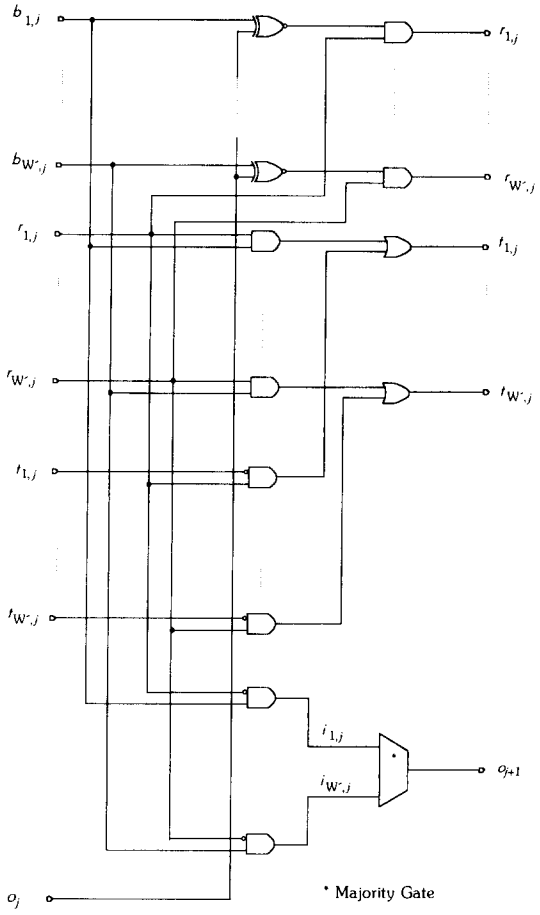

Fig. 4. The basic processing element (PE) used for median computation.

W' numbers are ordered in ascending sequence, d_l are placed to the extremes of this sequence. The key concept is that by having a method to compute the median value of $4N+1$ numbers and by being able to control $2N$ of these numbers, any rth order statistic of the rest $2N+1$ numbers can be determined. Figure 2(a)–(c) illustrate this concept. The dummy inputs are used for the computation of median, second-order statistic and maximum values, respectively. The bold window contains five numbers $x_{(1)}, x_{(2)}, \ldots, x_{(5)}$ in ascending order (the subscript in parentheses denotes the rank). The larger window contains nine numbers also in ascending order. By controlling the number of dummy inputs which are pushed to the top and to the bottom, any order statistic r of the numbers x_i can be obtained. More specifically, r exceeds by one the number of dummy inputs which are pushed to the top. Table 1 illustrates the proposed technique for the computation of the second-order statistic of nine 4-bit numbers (a "don't care" state is denoted by X). Since the second-order statistic is searched, seven of the dummy inputs have been set to 0000, whereas the remaining one has been set to 1111. In the first stage of the process the rejecting flags are "1", since all inputs are candidates and the setting flags are in a "don't care" state. The majority of the MSBs of the W' numbers is "0" and, therefore, the MSB of the output is "0". In the second

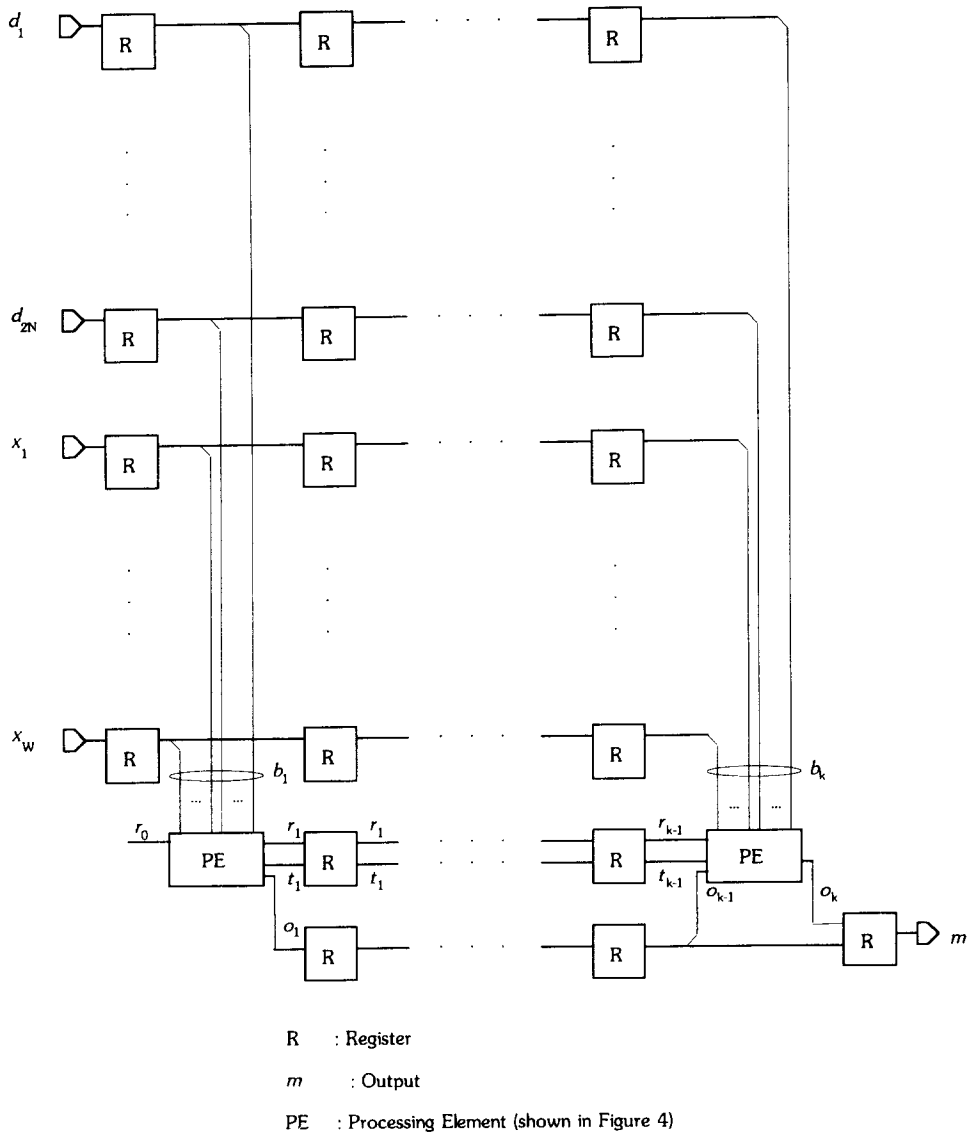

Fig. 5. Systolic array hardware structure implementing the majority gate technique for rank order filtering purposes.

stage the numbers of which the MSBs were "1" (i.e. x_4, x_7, x_8, x_9, and d_8) are rejected by setting their rejecting flag to "0". Then the corresponding setting flags are set to the complementary value of the MSB of the output (i.e. "1") and they are used for the majority selection. Also, in this stage, the majority of the bits $b_{i,2}$ of the numbers which are taken into account and the setting flags of the numbers which have been rejected is "0" and, thus, the output of this stage becomes "0". The process continues similarly in the two remaining stages and the output (x_3=3), which is both the median of the $W' = 4N + 1 = 17$ numbers and the second-order statistic of the $W = 2N + 1 = 9$ numbers, is obtained.

4. A SYSTOLIC ARRAY IMPLEMENTATION FOR RANK ORDER FILTERING

A pipelined systolic array suitable for implementing rank order filters is presented in this section. Its architecture is scalable and its hardware complexity expands linearly both with the size of the data window and the resolution of the numbers. The proposed architecture operates faster than other existing ones,[14] since intermediate signals are derived faster. Using the definitions of Section 2.1, the truth table for $r_{i,j+1}$, $i_{i,j}$ and $t_{i,j+1}$ is constructed as shown in Table 2. From this table the Karnaugh maps, shown in Fig. 3, are derived. From Fig. 3(a) it is obvious that

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

where $\cdot$, $+$, $\bar{}$ and $\otimes$ stand for logical AND, OR, NOT and XNOR, respectively. Also, from Fig. 3(b),

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

Finally, from Fig. 3(c) $t_{i,j+1}$ can be written as

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

or

$$t_{i,j+1} = (\overline{r_{i,j}} \cdot t_{i,j}) + (r_{i,j} \cdot b_{i,j}). \quad (7)$$

Realization of equation (7) leads to a faster hardware structure, since propagation delay of $\overline{o_j}$ signal is omitted.

The realization of the proposed Processing Element (PE) is based on equations (4), (5) and (7). The circuit diagram of this PE is shown in Fig. 4. Due to its simplicity (there are only three stages of gates including the inverters), it can attain very short processing times, independent of the data window size. Also, it becomes clear that the hardware complexity of the PE grows linearly with the number of its inputs.

A pipelined systolic array capable of computing rank order values is shown in Fig. 5. The inputs to this array are $W' = 4N + 1$ numbers (from which $W = 2N + 1$ are the data window and the $2N$ are the dummy inputs). The systolic array consists of PEs separated by registers (R). The resolution of the registers, which hold the data window numbers, is reduced by one bit in each successive stage, since there is no need to carry the $b_{i,j}$ coefficients which have already been processed. On the other hand, the resolution of the registers, which hold the result, is increased by one bit in each successive stage.

5. CONCLUSIONS

A new technique for realization of rank order filters based on the modification of the majority gate has been presented in this paper. A new PE has been designed for more efficient extraction of setting flag signals and, therefore, shorter processing times, independent of the data window size, have been achieved. Also, the hardware complexity of the PE grows linearly with the number of its inputs. A pipelined systolic array architecture suitable for performing rank order filtering has also been proposed. Typical applications of such an array include digital image processing, speech processing and coding, as well as digital TV applications, where rank order filters are employed.

REFERENCES

1. I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters—Principles and Applications*. Kluwer Academic Publishers, Boston (1990).
2. R. J. Schalkoff, *Digital Image Processing and Computer Vision*. Wiley, New York (1989).
3. P. Maragos and R. W. Schafer, Morphological filters—Part II: Their relations to median, order-statistic, and stack filters, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-35**, 1170–1184 (1987).
4. M. Chefchaouni and D. Schonefeld, Morphological representation of order-statistics filters, *IEEE Trans. Image Process.* **4**, 835–845 (1995).
5. I. Andreadis, A. Gasteratos and Ph. Tsalides, An ASIC for fast grey-scale dilation, *Microprocess. Microsystems* **20**, 89–95 (1996).
6. A. Gasteratos, I. Andreadis and Ph. Tsalides, Improvement of the majority gate algorithm for grey scale dilation/erosion, *Electronics Lett.* **32**, 806–807 (1996).
7. J. P. Fitch, E. J. Coyle and N. C. Gallagher, Jr., Median filtering by threshold decomposition, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-32**, 1183–1188 (1984).
8. J. P. Fitch, E. J. Coyle and N. C. Gallagher, Jr., Threshold decomposition of multidimensional ranked order operations, *IEEE Trans. Circuits Systems* **32**, 445–450 (1985).
9. C. J. Zarowski, R. D. McLeod and H. C. Card, Primitive cellular automata, threshold decomposition, and ranked order operations, *IEEE Trans. Computers* **38**, 148–149 (1989).
10. J. P. Fitch, Software and VLSI algorithm for generalized ranked order filtering, *IEEE Trans. Circuits Systems* **CAS-34**, 553–559 (1987).
11. L. E. Lucke and K. K. Parhi, Parallel processing architectures for rank order and stack filters, *IEEE Trans. Signal Process.* **42**, 1178–1189 (1994).
12. B. Arambepola, VLSI architecture for high-speed rank and median filtering, *Electronics Lett.* **24**, 1179–1180 (1988).
13. K. Chen, Bit-serial realizations of a class of nonlinear filters based on positive Boolean functions, *IEEE Trans. Circuits Systems* **36**, 785–794 (1989).
14. C. L. Lee and C. W. Jen, Bit-sliced median filter design based on majority gate, *IEE Proc. G* **139**, 63–71 (1992).

About the Author—ANTONIOS GASTERATOS received the Diploma from the Department of Electrical and Computer Engineering, Democritus University of Thrace (DUTH), Greece, in 1994. He is a Ph.D. candidate in the Department of Electrical and Computer Engineering, DUTH. His research interests are mainly in mathematical morphology, parallel architectures and VLSI image processing. He is a member of the Technical Chamber of Greece (TEE).

About the Author—IOANNIS ANDREADIS received the Diploma from the Department of Electrical Engineering, DUTH, Greece, in 1983, and the M.Sc. and Ph.D. degrees from the University of Manchester Institute of Science and Technology (UMIST), U.K., in 1985 and 1989, respectively. His research interests are mainly in machine vision and VLSI based computing architectures for machine vision. He joined the Department of Electrical and Computer Engineering, DUTH, in 1992. He is a member of the Editorial Board of the Pattern Recognition Journal, the Technical Chamber of Greece (TEE) and the IEEE.

About the Author—PHILIPPOS TSALIDES was born in Mirina Limnou, Greece, on 14 October 1953. He received the Diploma from the University of Padova, Italy, in 1979 and the Ph.D. degree from DUTH, Greece, in 1985. He is the Professor of Applied Electronics in the Department of Electrical and Computer Engineering, DUTH. His current research interests include VLSI architectures, VLSI systems, BIST techniques, smart imaging systems, applications of cellular automata in image processing, as well as computational systems. He has published a number of papers and a textbook on VLSI systems (*Basic Principles of Design and Fabrication*). He is a fellow member of the IEE and a member of the Technical Chamber of Greece (TEE).