N-time 시스톨릭 어레이 구조를 가지는 벡터 미디언 필터의 하드웨어 아키텍처
A New *N-time* Systolic Array Architecture for the Vector Median Filter

양영일*

Yeong-Yil Yang*

요 약

본 논문에서는 벡터 미디언 값을 계산하기 위한 시스톨릭 어레이 구조의 벡터 미디언 필터 구조를 제안하였다. 컬러영상처리에서 벡터 신호는 빨강, 녹색 파랑의 3개의 요소로 이루어져 있다. 벡터 미디어 필터는 빨강, 녹색 파랑 요소로 이루어진 벡터 신호들 중에서 벡터 신호를 크기 순서대로 나열하였을 때 가운데 값을 갖는 벡터 신호를 구하는 필터로, 컬러 영상처리에서 기본적으로 많이 사용되는 필터이다. 벡터 신호가 N 개가 있을 때, 지금 까지 제안된 구조에서는 $(3N+1)$ 클럭이 필요하나, 제안된 구조에서는 $(N+2)$ 클럭이 소요된다. 그리고 기존의 구조에서는 N 개의 입력 벡터 신호는 미디언 필터에 병렬로 입력되어야 하나 제안된 구조에서는 입력 신호는 직렬로 인가된다. FPGA를 사용하여 구현하였다.

Abstract

In this paper, we propose the systolic array architecture for the vector median filter. In the color image processing, the vec소or signal (i.e. the color) consists of three elements, *red, green* and *blue*. The vector median filter is very effective to utilize the correlation among *red, green* and *blue* elements. The computational complexity of the proposed architecture for computing the vector median of N vector signals is $(N+2)$ clock periods compared to the $(3N+1)$ clock periods in the previous method. In addition to, the input vector signals can be loaded in serial in the proposed architecture. In the previous method, N input vector signals should be loaded to the vector median filter in parallel at the first clock. The proposed architecture is implemented with FPGA.

Keywords : Median filter, Systolic array, Color image processing.

I. Introduction

In the image processing, the noise can be added to the original images through the transmission and the system itself. To reduce the noise especially the impulse noise, the median filter is widely used. Linear filters such as the lowpass filter are used to remove the Gaussian noise but these filters blur the edges and the sharp details. The median filter is suited for removing the impulse noise such as the salt and the pepper noise while preserving the edges.[1,2] The median filter is firstly proposed by Turkey[3] and Neuvo[4]extended the median filter to the vector median filter which processes a series of the vector signals each of which consists of the elements more than 2. In the color image processing, the vector signal (i.e. the color) consists of three elements, *red, green* and *blue*. The vector median filter is very effective to utilize the correlation among *red, green* and *blue* elements.

Chang *et. al*[5]proposed the systolic array architecture for computing the vector median. In that architecture, the $(3N+1)$ clock periods are required for computing the vector median from N vector signals and N input vector signals should be loaded to the vector median filter in parallel at the first clock. In this paper, we propose the new systolic array architecture computing the vector median. The first vector median for N vector signals is computed after the $(3N+2)$ clock periods. Following vector medians are computed at every $(N+2)$ clock periods. In the proposed architecture, N input vector signals are loaded in serial to the vector median filter.

In section 2, the definitions and vector median filter are described. The proposed architecture for the vector median filter is presented in section 3. Conclusion is given in section 4.

*경상대학교 전기전자공학부

논문 번호 : 2007-4-15　　접수 일자 : 2007. 9. 19
심사 완료 : 2007. 10. 24

II. Vector Median Filter

The median filtering is performed by sliding a window across the pixels of the image.

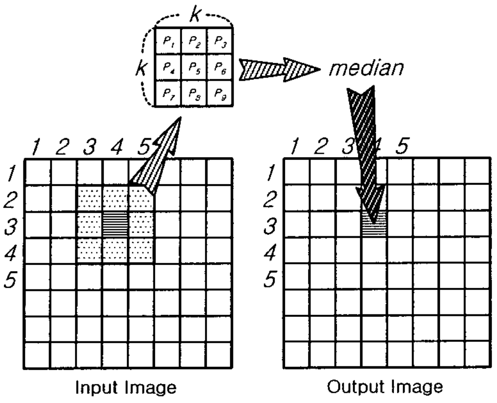

그림 1. 3×3윈도를 가지는 미디언 필터링.
Fig. 1. The median filtering with the 3×3 window.

Fig. 1. shows the vector median filtering with the 3×3 window whose center pixel is located at the third row and the fourth column. The number of pixels in the $k\times k$ window is $k^2=N$ (9 pixels in the 3×3 window) and the pixel P_i within the window has the value x_i. The pixel values in the window, $x_1,x_2,\cdots,x_N$, are sorted in ascending order and the medium value ((N+1)/2th element in the sorted list) is selected as the median x_{med}. The median value can be expressed as in Eq. (1).

$$x_{med}=Med(x_1,x_2,\cdots,x_N)=Med(x_i) \quad (1)$$

The computed median x_{med} is placed to the pixel in the output image corresponding to the center pixel of the window. (The pixel at the third row and the fourth column in Fig. 1.) The above process is performed repeatedly by sliding the window over one position until all the pixels in the input image are covered.

In the color image processing, the value of pixels consists of three elements, *red*, *green* and *blue*. The pixel is represented as the vector signal $x_i=(x_r(i),x_g(i),x_b(i))$. To compute the median of the vector signals, two methods are possible. One method is to compute the vector median by computing the median for each element separately as expressed in Eq. (2).

$$x_{v,med}=(Med(x_r(i)),Med(x_g(i)),Med(x_b(i))) \quad (2)$$

Because each element of the computed vector median is obtained independently, there are no correlations among the elements of the computed vector median $x_{v,med}$. To overcome this problem, following method is used to compute the vector median. The median x_{med} of a series of the scalar signal can be computed using Eq. (3).

$$\sum_{i=1}^{N}\|x_{med}-x_i\| \le \sum_{i=1}^{N}\|y-x_i\|,\ y\in\{x_1,x_2,\cdots,x_N\} \quad (3)$$

, where $\|x_i-x_j\|$ is the distance between the input signals, $x_i$and x_j. Eq. (3) can be extended to the vector median filter. The distance between two vector signals, $x_i=(x_r(i),x_g(i),x_b(i))$ and $x_j=(x_r(j),x_g(j),x_b(j))$ is denoted as $\|x_i-x_j\|$ and can be calculated by Eq. (4).

$$\|x_i-x_j\|=(x_r(i)-x_r(j))^2+(x_g(i)-x_g(j))^2+(x_b(i)-x_b(j))^2 \quad (4)$$

The distance D_i of the vector signal x_i from the other vector signals can be computed by Eq. (5). The vector median $x_{v,med}$ is defined as the vector signal having the minimum distance from the other vector signals as expressed in Eq. (6). The vector signal having the minimum D_i is the vector median $x_{v,med}$.

$$D_i=\sum_{j=1}^{N}\|x_i-x_j\|,\ j=1,\cdots,N \quad (5)$$

$$\sum_{i=1}^{N}\|x_{v,med}-x_i\| \le \sum_{i=1}^{N}\|y-x_i\|,y\in\{x_1,x_2,\cdots,x_N\} \quad (6)$$

III. Proposed Architecture for the Vector Median Filter

We describe the proposed architecture to compute the vector median from the given N vector signals. The proposed architecture operates in the systolic manner and finds the vector median at every (N+2) clock periods.

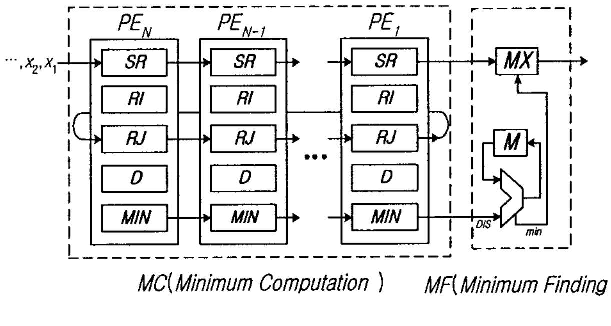

그림 2. 제안된 벡터 미디언 필터의 블록 다이어그램.
Fig. 2. The block diagram of the proposed vector median filter.

As shown in Fig. 2., the proposed architecture consists of two blocks, the *MC*(*Minimum Computation*) block which computes the distance D_i for the vector signal x_i and the *MF*(*Minimum Finding*) block which finds the vector signal having the minimum D_i value.

The *MC* block consists of N *processing elements*. To computer the vector median, the vector signals

$x_1, x_2, \cdots, x_N$ are loaded to the register *SR* of PE_N in serial. The data of the register *SR* of the PE_i are shifted to the register *SR* of the PE_{i-1} at every clock. After N clock periods, the register *SR* of PE_N stores the vector signal x_N and the register *SR* of PE_{N-1} stores the vector signal x_{N-1} and the register *SR* of PE_1 stores the vector signal x_1. The detail block diagram of the processing element is shown in Fig. 3.

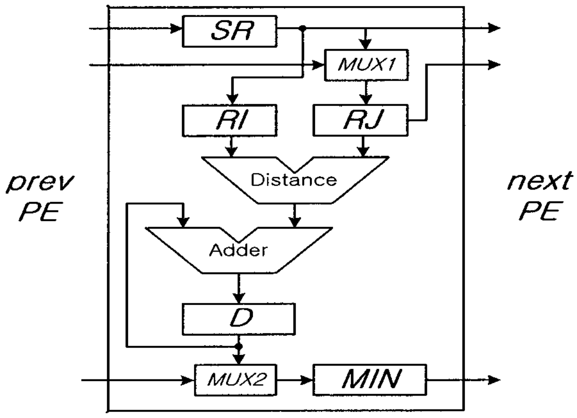

그림 3. 프로세싱 엘리먼트의 상세 블록 다이어그램.
Fig. 3. The detail block diagram of the Processing Element.

If all the vector signals are loaded to the registers *SRs*, the data of the register *SR* are transferred to the register *RI* and the register *RJ*. At this time, *MUX*1 selects the output of the register *SR*. The register *D* is initially set to zero. The ALU named *distance* computes the distance between two vector signals stored in register *RI* and the register *RJ* by using Eq. (4). The ALU named *adder* adds the output of the ALU named *distance* and the data of the register *D* which contains the cumulative distance. The added result is loaded to the register *D*. Because *MUX*1 of PE_i selects the output of the register *RJ* of the previous processing element PE_{i+1} during N clock periods, the data of the register *RJ* of PE_i are transferred to the register *RJ* of PE_{i-1}. The data of the register *RJ* of PE_1 are transferred to the register *RJ* of PE_N. During N clock periods, the register *RJ* of PE_i stores all the input vector signals. Therefore, the register *D* of PE_i stores the distance D_i for the vector signal x_i, the computation result of Eq. (5). If the distance D_i of Eq. (5) is computed, *MUX*2 selects the output of the register *D* and the data of the register *D* are transferred to the register *MIN*. If the data $D_i$is loaded to the register *MIN*, *MUX*2 selects the data coming from the previous *PE*. The data of the register *MIN* storing the D_i value are shifted to the next *PE* at every clock. The data of the register *SR* of PE_1 become the input to the *MF* block.

The *MF* block finds the vector signal having the minimum D_i value. The computed distances in the *MC* block, $D_1, D_2, D_3, \cdots$ are applied to the *MF* block in serial. The register *M* in the *MF* block keeps the minimum D_i value. The register *M* is initially set to the maximum value. If the data *DIS* coming from the *MC* block is smaller than the data of the register *M*, the data *DIS* is loaded to the register *M* and sets the signal *min*. If the signal *min* is set, the register *MX* loads the data of the register *SR* of PE_1. After N clock periods, the register *MX* stores the median.

The required time for computing the first vector median for N vector signals is as follows. N clock periods are required to load the vector signals to the register *SRs*. One clock period is necessary to load the data of the register *SR* to the register *RI* and the register *RJ*. To computer the distance D_i for each signal x_i, N clock periods are required. To computer the D_is in the *MC* block, N clock period is needed. The first vector median for N vector signals can be computed after $(3N+2)$ clock periods. Shifting of the input vector signals through the register *SRs* and the computation of the $D_i s$ and shifting of the data through the register *MINs* are performed separately to the different sets of N vector signals. While the computed D_is for the pth set of N vector signals are shifted through the register *MINs*, $D_i s$ for the $(p+1)$th set of N vector signals is computed and the $(p+2)$th set of N vector signals are shifted through the register *SRs*. After the first vector median is computed, following vector medians are computed at every $(N+2)$ clock periods.

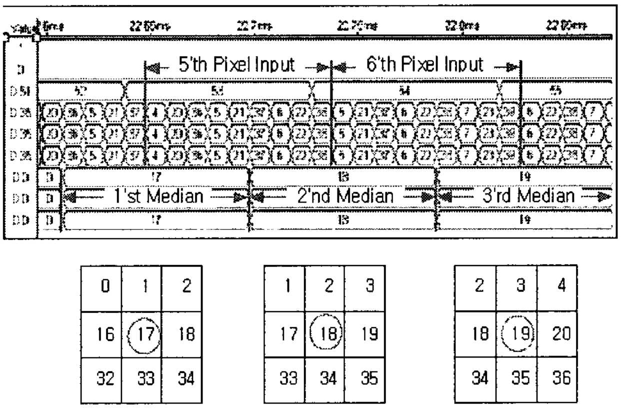

그림 4. 제안된 구조의 시뮬레이션 결과.
Fig. 4. Simulation result of the proposed architecture.

Fig. 4 shows the simulation result of the proposed architecture.

The proposed architecture is designed with MAX+II design tool and implemented on FPGA chip, *EPF10K200SRC240-3*. The number of gates in proposed vector median filter is about 65,000.

IV. Conclusion

In this paper, we propose the systolic array architecture for the vector median filter. The computational complexity of the proposed architecture for computing the vector median of N vector signals is $(N+2)$ clock periods compared to the $(3N+1)$ clock periods in the previous method. In addition to, the input vector signals can be loaded in serial in the proposed architecture. The proposed architecture consists of two blocks, the *MC* block and the *MF* block. The proposed architecture is implemented with FPGA.

References

[1] Gonzalez and Wood, *Digital Image Processing*, Addison Wesley Publishing Company, 1993.

[2] R. Crane, *A simplified Approach to Image Processing*, Prentice Hall PTR, 1997.

[3] J. W. Tukey, Exploratory Data Analysis. Reading, MA: Addison-Wesley, 1977.

[4] J. Astro, P. Haavisto, Y. Neubo, "Vector Median Filters," in *Proc. of IEEE*, Vol.78, No. 4, pp. 678-689, 1990.

[5] Long-Wen Chang, "A New Systolic Array Architecture for Vector Median Filters," in *Proc. of Visual Communication and Image Processing 99*, pp. 905-912, 1999.

양 영 일(Yeong-Yil Yang)

1983년 2월 경북대 전자공학과(공학사)

1985년 2월 한국과학기술원 전기 및 전자공학과(공학석사)

1989년 8월 한국과학기술원 전기 및 전자공학과(공학박사)

1990년 3월 ~ 현재 경상대학교 전기전자공학부 교수

※주관심분야 : 실시간 영상처리, VLSI Architecture