International Journal of MC Square Scientific Research Vol.2, No.1,2010

Efficient Architecture for Median Filter for Image Enhancement

R.Balarani
HOD, ECE Dept,
SRM University, Ramapuram Campus,
Chennai

K.Thirunadana Sikamani
HOD, CSE Dept,
St.Peters University,
Chennai.

ABSTRACT

Digital image processing is an ever elaborating and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas. Applications such as these involve different processes like image enhancement and object detection. The function of the proposed circuit is to detect the existence of impulse noise in an image neighborhood and apply the operator of the median filter only when it is necessary. Experimental results with real images demonstrate the improved performance. The proposed digital hardware structure is capable to process gray-scale images and is fully pipelined, whereas parallel processing is used in order to minimize computational time. In the presented design, a pixel image neighborhood can be selected for the computation of the filter output. The proposed digital structure was designed, compiled and simulated using the Modelsim and Synthesized in Xilinx.

1 Introduction

Two applications of great importance in the area of image processing are noise filtering and image enhancement. These tasks are an essential part of any image processor whether the final image is utilized for visual interpretation or for automatic analysis. The aim of noise filtering is to eliminate noise and its effects on the original image, while corrupting the image as little as possible. To this end, nonlinear techniques (like the median and, in general, order statistics filters) have been found to provide more satisfactory results in comparison to linear methods. For this reason, a number of non-linear filters, which utilize correlation among vectors using various distance measures. However, these approaches are typically implemented uniformly across an image, and they also tend to modify pixels that are undisturbed by noise, at the expense of blurred and distorted features.

2 Adaptive median filter design

The output of a median filter at a point x of an image f depends on the values of the image points in the neighborhood of x. This neighborhood is determined by a window W that is located at point x of f including n points $x_1, x_2,\ldots, x_n$ of f. the median can be determined when the number of points included in W is odd i.e., when $n = 2k+1$. the n values $f(x_1), f(x_2),\ldots, f(x_n)$ of the n points $x_1, x_2, \ldots, x_n$ are placed in ascending order forming the set of ordered values $\{f_1, f_2,\ldots, f_n\}$, in which $f_1 \leq f_2 \leq \ldots, \leq f_n$. The median is defined as he (k+1)th value of the set $\{f_1, f_2,\ldots,f_n\}$, med = f_{k+1}.

Two major remarks about the presented adaptive algorithm should be made. First, the value of the parameter α is of great importance, since it controls the operation of the circuit and the result of the overall procedure for different noise cases. Second, an appropriate positive and negative threshold value must be utilized for the case of impulse noise, when $f_{threshold}(x) \geq 255$. For example, in the case of Fig. 2(a), if we consider the parameter α = 8, then $f_{threshold}(x) = f_{max}(x) * 8 = 336$ and the $f_{threshold}(x)$ is limited to the value 255, $f_{threshold}(x) = 255$. The central pixel value is $252 < f_{threshold}(x)$ and the central pixel is erroneously not considered to be impulse noise. An adjustable positive threshold value (for example 240) can be used as a limit of $f_{threshold}(x)$. In this way, $f_{threshold}(x) = 240$, whereas the central pixel value is $252 > f_{threshold}(x)$, and the central pixel is successfully detected as impulse noise. The meaning of this normalization procedure is that pixels occupying values between a range of the impulsive values (and not only pixels with values 0 and 255) should be considered as noisy pixels.

3 Hardware architecture

The proposed architecture is based on a sequence of pipeline stages in order to reduce computational time. Parallel processing has been employed to further accelerate the process. For the computation of the filter output, a 3x3 or 5x5 pixel image neighborhood can be selected.The structure of the adaptive filter comprises four basic functional units, the moving window unit, the median computation unit, the noise detection unit, and the output selection unit. The input data of the system are the gray-scale values of the pixels of the image neighborhood, the value of the parameter α, and the positive and negative threshold values. Additionally, two control signals required for the selection of the operation of the system (negative/positive noise suppression) and the neighborhood size (3x3 or 5x5) are also utilized.as shown fig 3.

4 Moving window unit

The pixel values of the input image, denoted as “IMAGE_INPUT [7..0],” are imported into this unit in serial. The value of the parameter is denoted as “MOD_VALUE[7..0]” and the positive and negative threshold values as POS/NEG THRESHOLD respectively. The parameter α is a real number, 5 and 3 bits are used for the representation of the integral and the fractional part, respectively. The “NEG/POS” control signal is used to determine the noise type. When “NEG/POS” is equal to “0” (“1”) the circuit operation is negative (positive) noise suppression.

For the moving window operation, a 3x3 (5x5) pixel serpentine type memory is used, consisting of 9 (25) registers, illustrated in Fig. 6. In this way, when the window is moved into the next image neighborhood, only 3 or 5 pixel values stored in the memory are altered. The outputs of this unit are rows of pixel values (3 or 5, respectively), which are the inputs to the median computation unit.

5 Median computation unit

In this stage, the median value of the image neighborhood is computed in order to substitute the central pixel value, if necessary. In this unit, the min/max value of the neighborhood is also computed, used in the noise detection process. For the computation of both the median and the min/max value a 24-input sorter is utilized, the central pixel value is not included. In this way, the complexity of the design is reduced since no additional min/max modules are utilized. The modules of the sorter used only in the case of the 5x5 pixel neighborhood are enabled by the “en5x5” control signal. A CS block is a max/min module; its first output is the maximum of the inputs and its second output the minimum. The implementation of a CS block includes a comparator and two multiplexers and is depicted in Fig. 4.

6 Noise detection unit

The task of the noise detection unit is to compute the threshold value for the detection of a noise pixel, $f_{threshold}(x)$, and to limit this value to the positive (negative) threshold. Initially, the min/max value of the neighborhood is selected, and for that reason, the values "OUT_0[7..0]," "OUT_7[7..0]" and "OUT_23[7..0]" (min and max values, respectively) are imported into a multiplexer. The selection is based on the values of the "NEG/POS" control signals. In the next step, the output of the multiplexer is multiplied by the parameter α (8 bits) using a multiplier module, the resultant 16-bit value is denoted. An additional 2-to-1 multiplexer is utilized to select the positive or negative threshold to which the THRESHOLD_VALUE should be normalized, controlled by the "NEG/POS" control signal. A comparator is used to compare the THRESHOLD_VALUE to the positive or negative threshold and a multiplexer to select the corresponding output threshold value, denoted as THRESHOLD

7 Output selection unit

The final stage of the design is the output selection unit. In this unit, the appropriate output value for the performed operation is selected. For the selection of the output value the corresponding threshold value for the image neighborhood, "THRESHOLD," is used. The value "THRESHOLD " is compared to the central pixel value, denoted in the circuit as "CENTRAL_PIXEL."

Conclusion

This paper presents a new design of an adaptive median filter, which is capable of performing impulse noise suppression for 8-bit grayscale images using a 3x3 pixel neighborhood. The proposed circuit detects the existence of noise in the image neighborhood and applies the corresponding median filter only when it is necessary. The noise detection procedure is controllable, and, thus, pixel values other than the two extreme ones can be considered as impulse noise, provided that they are significantly different from the central pixel value. In this way, the blurring of the image is avoided.

REFERENCES

[1] W. K. Pratt, Digital Image Processing. New York: Wiley, 1991.

[2] G. R. Arce, N. C. Gallagher, and T. Nodes, "Median filters: Theory and applications," in Advances in Computer Vision and Image Processing, T. Huang, Ed. Greenwich, CT: JAI, 1986.

[3] T. A. Nodes and N. C. Gallagher Jr., "The output distribution of median type filters," IEEE Trans. Commun., vol. COM-32, pp. 532–541, May 1984.

[4] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," Pattern Recognit. Lett., vol. 15, pp. 341–347, Apr. 1994.

[5] E. Abreau, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new effici-cient approach for the removal of impulse noise from highly corrupted images," IEEE Trans. Image Processing, vol. 5, pp. 1012–1025, June 1996.

[6] K. S. Ali, "Digital circuit design using FPGAs," in Proc. 19th ICCIE, 1996, pp. 127–129.

[7] K. E. Batcher, "Sorting networks and their applications," in Proc. AFIPS-SJCC, 1968, pp. 307–314.

[8] I. Pitas and A.N. Venetsanopoulos Nonlinear Digital Filters: Principles and Applications. Boston, MA: Kluwer, 1990.