

Design and VLSI Implementation of Real-Time Weighted Median Filters

Chun-Te Chen, Liang-Gee Chen, Tzi-Dar Chiueh, and Jue-Hsuan Hsiao

Department of Electrical Engineering

National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

In this paper, a novel design method and VLSI implementation of weighted median filters is presented. The weight parameters are incremented to minimize the mean square error of the input signal and desired signal when the sub-set of the image data is applied. The sub-optimal(optimal) weight set is achieved after 71 training cycles for 5*5 window size under the MSE criteria. A two-level adder tree architecture with pipelined latches is proposed to find the weighted median output with versatile definition of weight. It requires only K iterations to find output, where K means the resolution of the input signal samples. The task interleaved processing is also adopted to improve the throughput. The final chip layout with pipelined latches for 5*5 window size is also given in this paper. This high-speed VLSI implementation of weighted median filter will meet the real-time application requirements.

1 Introduction

The Weighted Median Filter(WMF) is first introduced by Justusson[1] and further discussed by Brownrigg[2] and Wendt *et al*[3]. This filter gives more weight to some samples within the sliding window to allow a degree control of the smoothing behavior. The sample in the sliding window is duplicated as weight define before sorting; then the center value of the sorted list is chosen as a median output. The total weight sum is equal to T_w , where T_w is an odd positive integer. These weighted median filters are of fundamental important since they are a large number of extensively used filter types, such as order statistic filters, Center Weighted Median Filters(CWMF) and standard Median Filters(MF). In the center weighted median filter proposed in [4], they give more weight only to the center value of the window, and thus it is easier to design and implement than general weighted median filters. However, it is too sensitive to the characteristic of the image with the impulsive noise. Recently, the adaptive stack(weighted) filter[5,6] techniques have

been proposed for enhancing image degraded by noise. These adaptive stack filters allows more parameters to be adjusted not only center parameter. By adjusting their weights depending the characteristic of the input image, these filters can outperform the median filter and the center weight median filter as showns [6]. The detail of the image is preserved and removed impulsive noise efficiently. The optimal weight set is determined by linear program and some advance algorithm as presented in [5,6]. It is more interesting that this estimated approach alleviates modeling of the signal and noise by taking a part of the input signal to train the weight filter. However, the problem with these methodology is that the number of parameter in the linear program grows exponentially in the window width of the filter. On the other hand, the existing approaches have a slow convergence speed since the adaptation does not utilize recent weight coefficients. Hence, we proposed an efficient method to determined optimal weight set with faster convergence speed.

Real-time implementation of a weighted median filter involves more computation than standard median filters since the number of samples to be sorted is increased. In the sorted network implementation of weighted median filters[7], each sample must be duplicated many times as weight define before applied to sorted network. The input size of network is increased. Therefore, it consists of a large number of compare-swap units made these unattractive. An alternative implementation of weighted median filters is represented in stack filters. The stack filters, introduced by Wendt et al[3] has used positive boolean functions with $O(2^r)$ area-time complexity to perform binary filtering on unit-weight signal. The chip was designed to perform any rank-order operation with maximum window width 9 on 6-bit input data. O.Y. Harja *et al*.[8] has been extended the PBF to realized the weighted median operation with some limitation. Chen[9] proposed bit-serial tree-search method to reduce the hardware complexity to only one PBF. This approach can only support smaller window size, since the hardware complexity will be increased for larger window size. Generally, to perform a median on 25 da-

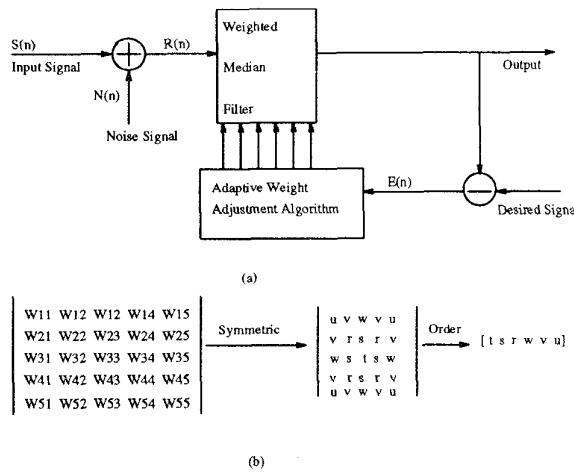


Figure 1: (a) The system block of weighted median filter definition. (b) The parameters of filter definition before applied to training algorithm.

ta (5×5 window size), it need 5,200,300 items for logic OR operation and each item has 13 parameter for logic AND operation. The other limitation of PBF is that the weights set is fixed. This method requires huge connection network bandwidth for versatile definition of weight. Hence, we proposed an efficient VLSI implementation of the weighted median filters, whose weight are adjustable to support standard median filters, center weighted median filters, order statistic filters and weighted median filters. This paper is organized as follows. Section 2, a training algorithm and design method for weighted median filters is introduced. Section 3 describe the architecture of the weighted median filters. The final conclusion is given in section 4.

2 A Design Method for Weighted Median Filter

In this section, we proposed a training algorithm, which has a faster convergence rate to determine the weight in the filters. The weighted median filters is state as a optimal filtering problem as shown in Fig. 1(a). The process $R(n)$ at the input of the filters is assumed to be a corrupted version of some desired process $S(n)$. The corruption may be caused either by a noise process $N(n)$ or by some intentional operation, such as a modulation scheme. The goal is to adjust the weight of the filter such that the average MAE or MSE per time unit between the filter's output and the desired signal is minimized.

To reduce computation, the class of weighted median filters whose weights are symmetric about the window

center is discussed as in [2]. Hence, the number of the parameter to be adjusted is reduced almost 1/4 of the unsymmetric weighted median filters. The number of distinct filters is limited to only 53 types for 3×3 window size when the Brownring approach[2] is applied. The optimal weight set is found by exhausted search from this 53 distinct types. However, when this approach is extended for 5×5 window size, the number of the distinct filters is increase to over 5000. Therefore, we can not use this approach to determine the weights when the window size is larger than 9. However, the problem with those methodologies in [5,6] is two complex to implement. And the convergence rate is too slow in real-time applications. Hence, we proposed an efficient method, which is simple in VLSI implementation to determined optimal weight set with faster convergence speed.

The main idea of the proposed algorithm is described as follow paragraph. The MSE or MAE curve of the training image data depicted as a function of the central weight is convex and has a unique minimum value at a certain weight in the center weighted median filters. This result has been presented in [4]. Assume this property is also maintained in the generalized weighted median filters case. The cost function (MAS or MSE) became as a function with multiple parameters and its cost function is convex and has a unique minimum value. The initial value of the weighted median filter is set equal to 1 as that of standard median filter. The central value is adjusted at first; therefore, the performance of this filter will superior to the standard median filter. Then the weight set is stored in register for final determination. The parameter next closet to the center is adjusted after the central value is determined. When the next closed parameter is incremented, the central value must also be incremented to ensure to minimize the function with these parameters. The cost function is not always less than the previous result. We are forced to accept this result until the minimum of the cost function is reached. This minimum MAE or MSE must be compared with the stored results to determine which weight set is better. Then the parameter third closet the center is adjusted. And the same procedure goes on. The weights will be only increased at each traing cycle to improve performance since the previous weight set are stored for restoration. The performance of this filter will also superior to that of the center weighted median filter since the weight set with minimum MSE or MAE will be accept in train procedure.

The upper left quarter of the Lena image with noise was used to train to get sub-optimal weight set for 5×5 window size. The noise image is generated by corrupted Gaussian noise with zero mean and variance 30 to original Lena image. The probability of impulsives is 0.1. The training curve of the proposed design under MSE

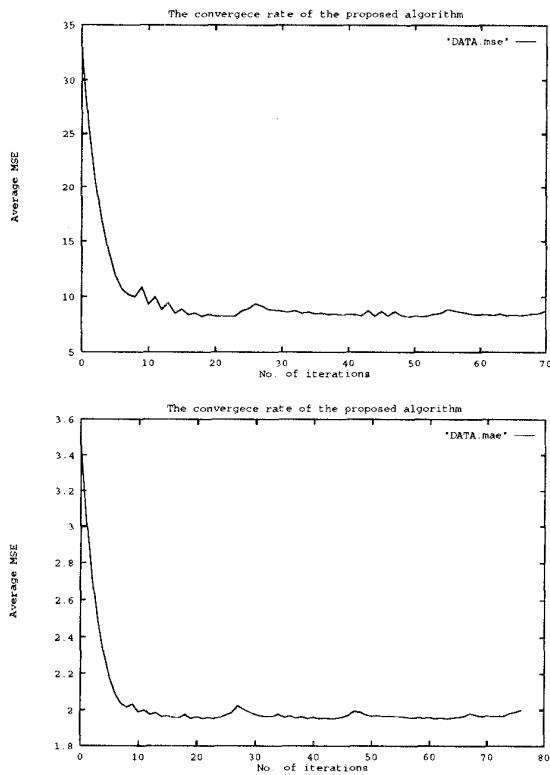


Figure 2: The sub-set of image data with noise is applied to train to get optimal weight set. The convergence curve of proposed algorithm under MSE (or MAE) criterion.

or MAE is shown in Fig. 2. The sub-optimal (optimal) weight set is achieved after 71 training cycles for MSE criterion or 75 training cycles of MAE criterion. The performance of these sub-optimal weight set for full size Lena image can outperform median filters and center weighted median filters as shown in Table 1.

3 Proposed Architecture for Weighted Median Filter

A detail architecture design for weighted medain filters is discussed in this section. At first, we introduce a novel pipelined method for finding median of the weighted medain filter. And then its architecture with task interleaving is adopted to improve throughput. The finer bit-level is also considered in this section.

Filter Type	MAE	MSE
MF (3*3)	2.738983	18.676353
MF (5*5)	3.703885	42.841957
CWMF	2.122288	12.500675
WMF(MSE)	2.016308	9.483128
WMF(MAE)	2.033897	9.922958

Table 1: MSE and MAE associated with Gaussian noise (with variance=30) and impulsive noise ($P=0.1$) for full size of image data. The weight set in weighted median filter is found by proposed training algorithm when the 1/4 size of image data is applied.

3.1 A novel pipelined weighted median finding method

The weighted median finding operation is to find a minimum boundary value (W_k) in the sliding window which divided the original sample in the sliding window into two sub-set, S_0 and S_1 . The samples which are equal to or larger than (W_k) are put into S_1 , otherwise they are put in S_0 . And the associated weight sum in S_1 must larger than $M=(T_w+1)/2$, where T_w means the total weight sum of the sliding window. The maximum number of the sliding window is first as a boundary value to evaluate the weight sum in S_1 . And then the second larger number is chosen and evaluated the partition, and so on. It will not stop until the weight sum of the sample in S_1 is equal to or larger than M . The output of the weighted median filter is the sample corresponding to the last boundary value. The computational complexity grows with K and N . Hence, an efficient method to find the boundary value and evaluate the weight sum is required for real-time implementation. A radix search method, not directly finding the maximum sample is proposed to find the boundary value as that of the Chen[9]'s bit-serial method. The most significant bit (MSB) of the boundary value is set equal to 1. The sample of the sliding window are divided into two partitions corresponding the this boundary value. Then we accumulated the weight sum associated in the S_1 sub-set. If it is greater than or equal to M value, the correct output of the weighted median filter will be "1", otherwise "0". The sample whose MSB bit with 0 are belong to S_0 partition and labeled them. On the other hand, if the weight sum is less than M , the sample whose MSB bit with 1 are belong to S_1 partition and labeled them. When the labels of the samples are determined, they will keep these labels until the final bit of the boundary bit is determined. The second MSB of boundary bit is also determined as the same manner. After K iterations, the final boundary value is determined for K -bit input data. The area-time complexity of the proposed realization is $O(K)$, as compare

$$W[.] = [1 \ 2 \ 3 \ 2 \ 1]$$

$$X[.] = [4 \ 7 \ 5 \ 11 \ 9] = [0100 \ 0111 \ 0101 \ 1011 \ 1001]$$

$T_w=9$, $M=5$,
 $k=4$,
 $WM[.] = [\dots \dots \dots] \rightarrow 0+0+0+2+1=3 < 5 \quad Y(4)=0$,
 $K=3$,
 $WM[.] = [\dots \ 1 \ 1] \rightarrow 1+2+3+2+1=9 >= 5 \quad Y(3)=1$,
 $K=2$,
 $WM[.] = [\dots \ 1 \ 1] \rightarrow 0+2+0+2+1=5 >= 5 \quad Y(2)=1$,
 $K=1$,
 $WM[.] = [0 \ 0 \ 1 \ 1] \rightarrow 0+2+0+2+1=5 >= 5 \quad Y(1)=1$,

The weighted median filter output= $Y(0111)=7$,

Figure 3: An example the weighted median filter

to $O(N)$ for direct implementation. An example of the weighted median filter is shown in Fig. 3. The weighted median filter output is achieved after $K(=4)$ iterations.

3.2 A Architecture of weighted median finding

In the boundary value finding, the weight sum operation is a higher computation task. In the traditional PBF implementation, it requires not only larger number of AND or OR gates but also harder to divide into many pipelined stages to improve throughput. Hence, a two-level tree architecture with pipelined latches to sum the corresponding weights is proposed. The smoothing of the data flow is also considered, otherwise the extra hardware will be induced. We used many row adder tree (RAT) modules and one column adder tree (CAT)module as shown in Fig. 4 to realize proposed algorithm for 2-D case. The output of the WM module are in parallelism multiplied with associated weights before applied to the tree adder. The outputs of RAT modules are down to CAT module to evaluate the final results. This final result is added by the 2's complement of median value, M . The carry-bit of this final adder is used to determine associated median output and update associated label matrix. In this architecture, all operations are executed from the LSB (least significant bit) to MSB. The weighted median output is come out from the MSB to LSB.

The input word data must be converted into bit-serial for weighted median finding. Here, we introduce a new method combined with task interleaving processing as shown in Fig. 6. The timing relationship of the input

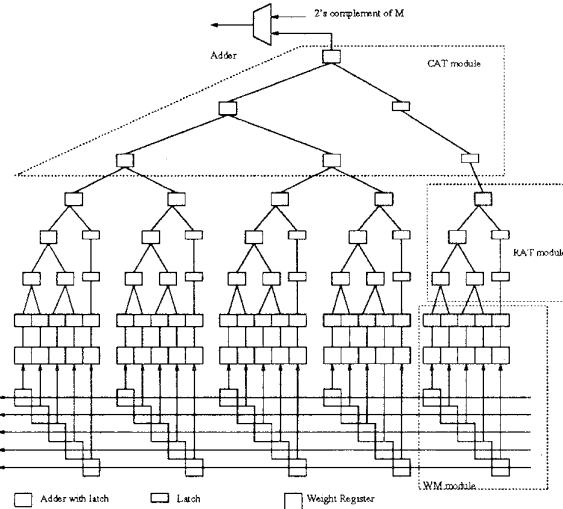


Figure 4: A two-level tree architecture of weighted median filter

data, interleaved task, and bit-serial output is also shown on the bottom of this figure. For 5×5 window size case, there are 5 number of bit stream are generated from this data arranged module to drive the WM module as shown in Fig. 5.

In the WM module as shown in Fig. 6, labels of every tasks are stored in a ring-register. It will shift one position per clock cycle corresponding to proper interleaved task. The label of sample in the ring-register will shift one position per cycle to evaluate other sliding window. The input data of WM module are arranged in bit parallel by its left data rearrange module and it was passed to its right WM module

For more flexibility, the weight registers in WM module are connected in cascade as a shift registers. The new weight set are loaded from a signal input port in sequence. It has $2 * (\log_2 N + 2)$ number of registers to store the label of interleaved task, where N means the window size of 2-D filter. The independent tasks are inserted interleavedly into the waiting stages. The process of K iteration to find a single output is defined as a single task. We can execute 8 tasks with the 5 line of inputs streams, for 2-D weighted median filter with 5×5 window size. And there are $320 (= 8 \text{ bits} * 8 \text{ tasks} * 5 \text{ streams})$ register to implement the word-parallel/bit-serial with task interleaving function.

Layout of the chip is displayed in Fig. 7 for 5×5 window size. And the total maximum weight sum is limited to 255. It takes about 36 pins and 2,491 number of standard cells, which is less than that of PBF implementation (5200300). The active area of weighted median finding is about $3.91 * 3.91 mm^2$. The function of this chip is verified by Verilog Simulator. And the

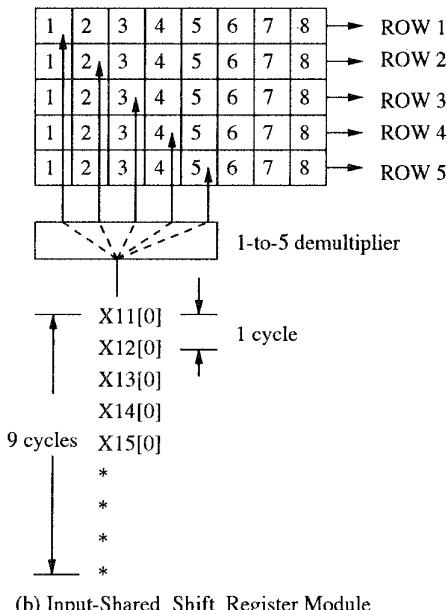
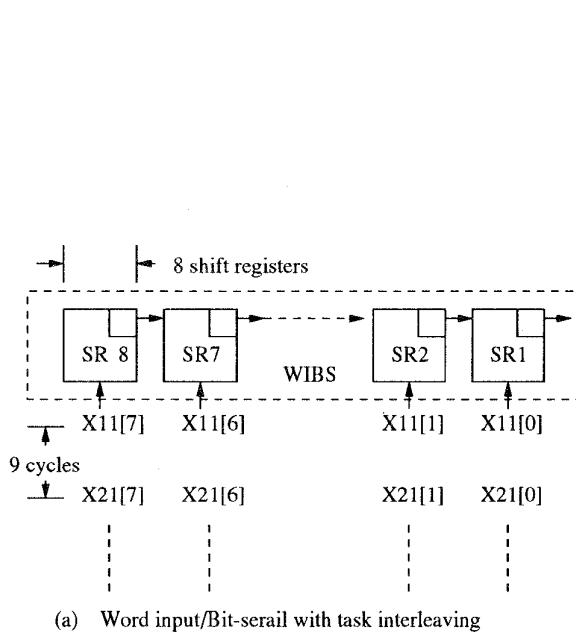


Figure 5: a) Architecture of Word input/Bit-serial converter with task interleaving. b) The architecture of Shift Register with input-shared.

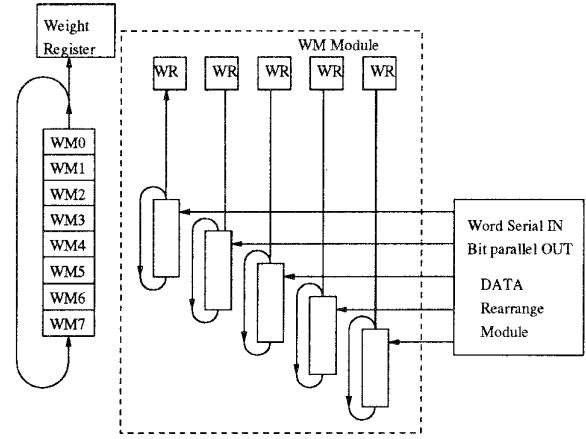


Figure 6: The architecture of WM module with a ring register for task interleaving process.

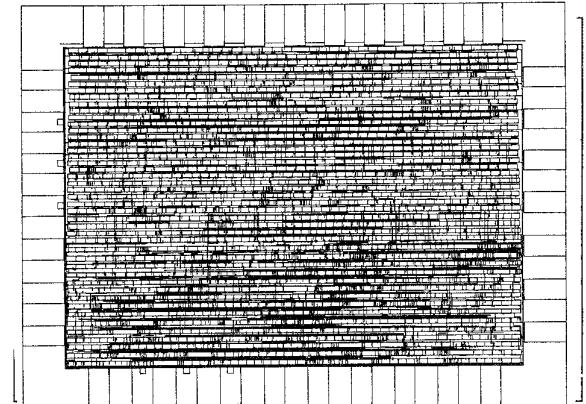


Figure 7: The final layout of the weighted median filter.

critical path or pipelined latency(8-bit adder) is about 16ns for 0.8 micron CCL standard cell library. The proposed chip can support weighted median filters, standard median filters and order statistic filters since the weight set are down loaded from the external of chip. Compared with Positive Boolean Function in [9], the proposed two-level tree architecture has less number of logic gates even with pipelined latches. This architecture is very regular to pipeline in bit-level operation. That is because the weight sum operation and 2's complement M adding may all execute from LSB to MSB. Hence, we can insert 16 tasks(more than 8) with some modification. The pipelined cycle is reduced to only one bit carry-save-adder(CSA). This bit-level pipelined architecture, which is extended the current word-level version can support

higher speed applications.

4 Conclusions

A design method to train the weighted median filter and its VLSI implementation are presented in this paper. The sub-optimal weight set is achieved rapidly with proposed traing algorithm. In the experiment results, the curve of MSE is reduced very fast in very short training cycle and become smoothing. Hence, the training cycle can be reduced to get acceptable result. The detail of the image is preverved efficiently, which is overcome the smoothing output of the center weighted median filter. The arithmetic method for weighted median filter is also proposed. It requires K iterations of the weight sum operation to complete the full operation of median finding. The adder tree architecture is adopted to reduce each cycle time. By doing task interleaving, the adjacent tasks can be mutual-exclusively interleaved in order to fill the pipeline 100%. The throughput is increased as the number of tasks inserted. The proposed design is regular to pipeline in bit-level for high-speed applications. The final chip layout for generalization weighted median filter with 5*5 window size is also given in this paper. For non-integer weight set, they can either normalized or shift to be integer without loss their generality. Obviously, this chip can support standard median filters and other adaptive stack filters[5,6]. In the future, the bit-level pipelined architecture of the medain finding operation will be also realized in VLSI implementation. And its real-word applications will be developed based on the proposed architecture and the training algorithm.

References

- [1] B.I.Justusson, " Median filtering: Statistic properties," in Topic in Applied Physical, Two-Dimensional Digital Signal Processing II, T. S. Huang, Ed. Berlin: Springer, 1981.
- [2] D. R. K. Brownrigg, " The weighted median filter," *Commun. Assoc. Comput. Mach.*, vol. 27, pp. 807-818, Aug. 1984.
- [3] P.D.Wendt, E.J.Coyle, and N. C. Gallagher, "Stack Filters," *IEEE Trans. on Acoust.,Speech, and Signal Processing*, vol. 34, pp. 898-911, 1986.
- [4] S. J. Ko and Y. H. Lee, "Center Weighted Median Filters and Their Application to Image Enhancement," *IEEE Trans. on Circuits and Systems*, vol. 38 pp. 984-993,1991
- [5] J. H. Lin, T. M. Sellke, and E. J. Coyle, "Adaptive Stack Filtering Under the Mean Absolute Error Criterion," *IEEE Trans. on Acoust.,Speech, and Signal Processing*, vol. 38, No. 6, pp. 938-954, 1990
- [6] L. Yin J. T. Astola, and Y. A.Neuvo, "Adaptive Stack Filtering with Application to Image Processing,", *IEEE Trans. on Signal Processing*, vol. 41, pp. 162-184, 1993.
- [7] M. Karaman, L. Onural, and A. Atalar, "Design and Implementation of a General-Purpose Median Filter Unit in CMOS VLSI,", *IEEE Journal of Solid-State Circuits*, vol. 25, No. 2 pp. 505-512, 1990.
- [8] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the Properties of Median and Weighted Median Filters using Threshold Logic and Stack Filter Representation,", *IEEE Trans. on Signal Processing*, vol. 39, pp. 395-410, 1991.
- [9] K. Chen, "Bit-serial Realizations of a class on Non-linear filter based on Positive Boolean Function," *IEEE Trans. on Circuits and Systems*, vol. 36, pp. 785-794,1989.