BIT-BASED WEIGHTED MEAN FILTER

Barun K. Kar,
Adv. Design Techology (SPS)
Motorola, 2200 E Elliot
Tempe, AZ-85284
email : kar@adtaz.sps.mot.com

Mitrajit Chatterjee and Dhiraj K. Pradhan
Dept. of Computer Science
Texas A & M Univ.
College Stn., TX-77843
mitrajit,pradhan@cs.tamu.edu

ABSTRACT

Linear-Nonlinear hybrid filters that have appeared in literature suffer from some severe disadvantages. They smear edges and are very hardware intensive. These shortcomings can be overcome by having a Bit-based Weighted Mean filtering scheme. This filter starts by calculating the median of a set of sample values. The sample values are then scaled. Those values which lie in the proximity of the median, are granted more weightage. The weighted sample values are then averaged to yield the filter output. Results show that these filters perform much better than their earlier counterparts with respect to edge preservation and minimizing the minimum absolute error criterion when applied to images corrupted by both impulsive and nonimpulsive noise. These filters are also much more hardware efficient than the L, ATM and M filters[5]-[6].

1 INTRODUCTION

Although median filters[1] is very efficient in removing impulse noise while preserving edges, they do not provide sufficient smoothing of nonimpulsive noise. On the other hand, linear filters [2] which incorporates the averaging operations though are very sensitive to impulsive aberrations, attenuates nonimpulsive noise effectively. This lead to the development of statistical operators which uses nonlinear filters in combination with linear filters called the Linear-Nonlinear Hybrid(LNH) filter. The nonlinear filter performs by editing some amount of data. The remaining data is then passed through a linear filter to smoothen out the nonimpulsive noise part. Among the better known operators are the L-filter, popularly known as the Order Statistic filter[5], the *alpha-trimmed mean filter* (ATM filter)[4] and the M-filters[6]. These filters are known to smear edges and have an intensive hardware requirement.

These drawbacks are taken care of by the Bit-based Weighted Mean Filter(BWMF). This filter calculates the median of a set of sample values. The sample values are then scaled, with those values in the proximity of the median(i.e. have more MSB's tallying with the median MSB's) given more weightage. The weighted sample values are then averaged to yield the filter output. Since the BWM filter performs a median operation (as opposed to the sorting in L filter), it is more hardware efficient than the L filter.

Various subclasses of the BWM filter can be conceived. Two of them, the Tag-based Trimmed Mean filter(TTMF) and the B filter has been presented in [7]. These filters are largely based on the statistical knowledge of the edge height and the variance of the noise corrupting the images. Results show that these filters perform much better with respect to removal of noise, preservation of edges and hardware requirements than the ATM and M filters.

The implementation schemes for these filters can be made very cost efficient by using a bit serial median filter algorithm[3]. The beauty of this algorithm lies in the fact that a fast parallel running-mean filter can be added to it with only a minimal increase in hardware.

In section 2, a bit serial median filtering has been discussed. This algorithm provides the basis for implementating the bit-serial LNH filters. The Bit-based Weighted Mean filtering algorithm and an implementation scheme has been presented in section 3. Two subclasses of the BWM filter has been proposed in section 4. Results shown in section 5, show that these filter perform much better in minimizing the absolute error criterions for images corrupted with both impulsive and nonimpulsive noise than the ATM and M filters.

2 Bit Serial Median Filter

Work done in this paper is based on the new realization that median filtering can be done in an efficient manner through a bit-serial algorithm[3]. The algorithm for finding out the i-th ranked number among $2N+1$ k-bit long numbers $\{x(1), x(2), \ldots, x(n)\}$ is given below:
$\text{median}(w(n) = \{x_{(k:1)}(n) : n = 0, \ldots, 2N+1\})$
$= \text{median}(w_{(k:q)}(n) = \{x_{(k:q)}(n) : n = 0, \ldots, 2N+1\}).2^q + \text{median}(\{x^\star_{(q:1)}(n) : n = 0, \ldots, 2N+1\})$

$$x^\star_{(q:1)}(n) = \begin{cases} x_{(q:1)}(n), & x_{(k:q)}(n) = \text{median}(w_{(k:q)}(n)) \\ 0, & x_{(k:q)}(n) < \text{median}(w_{(k:q)}(n)) \\ 2^{q-1}, & x_{(k:q)}(n) > \text{median}(w_{(k:q)}(n)) \end{cases}$$

```
PROCEDURE BWMF($\beta, x, n$);
begin
  sum = denom = 0   : Init. sum and denom.
  $tag_1 = tag_2 = \ldots tag_n = 1$
  $n_1 = n_2 = \ldots n_k = 0$   : Set $n_i$ to 0.
   for $\ell = k$ to 1 do
      begin   : Start Median Computation
      $\text{med}_\ell = \text{median}(x_\ell(1), x_\ell(2), \ldots, x_\ell(n))$;
       for $j = 1$ to $n$ do
        begin
        : Calculate $n_i$, modify nos and tag.
         if $x_\ell(j) \neq \text{med}_\ell$ and $\text{tag}_j = 1$ then
            sum = sum + $\beta_l$.x(j); : sum.
            $n_\ell := n_\ell + 1$;   : Calculating $n_\ell$.
            $tag_j := 0$;
            for $q = \ell - 1$ to 1 do $x_q(j) = x_\ell(j)$ ;
                              : Modifying the nos.
        end;
      end
    sum = sum + median.$\sum_{j=1}^{n} tag_j$;
       : The median being given the max wt.
   for $\ell = k$ to 1 do denom = denom + $\beta_\ell . n_\ell$;
   Output = sum / (denom + $\sum_{j=1}^{n} tag_j$);
return
```

Table 1: Filtering Algorithm

where $k > q$. The algorithm starts by filtering the MSB of the (2N+1)-numbers in the window to yield the MSB of the filter output. The MSB of the filter output is then compared with the MSB of all the numbers. Those numbers whose MSB is not equal to the filter output have their corresponding MSB propagated to the right thereby changing all the bits to the MSB, a constant value of 0 or 1. In this way we reduce the problem of finding rank-order filter output of k-bit numbers to finding out the filter output for the modified $(k-1)$-bit number. This process is repeated for successive bit positions to the right.

Definition 2.1 *The tag bit $t_{(k:q)}(m)$ associated with any number $x(m)$ in the window $w(n)$ is given as follows:*

$$t_{(k:q)}(m) = \begin{cases} 1, & x_{(k:q)}(m) = median(w_{(k:q)}(m)) \\ 0, & x_{(k:q)}(m) \neq median(w_{(k:q)}(m)) \end{cases}$$

The tag bit tells us whether the first $k-q$ most significant bits of $x(m)$ are equal to the first $k-q$ most significant bits of the median.

3 The Bit-Based Weighted Mean Filter

During the recent years linear-nonlinear hybrid filters have gained momentum because of their ability to remove both impulsive and nonimpulsive noise. A class of LNH filters is the L filter[5]. This generalized median filter encompasses a variety of LNH filters like the M filters, ATM filters etc. As has been discussed earlier, the filter operates by sorting the sample value in the filter window. The sorted value are then scaled accordingly and averaged to yield the output.

The L filters suffer from some severe drawbacks. Though an impulse would have less effect on the filter output, there is a significant smearing of edges. The edge smearing is caused due to the inability of the filter to recognize the range where the majority of the sample values lie, i.e., those values which should significantly affect the filter output. Let us consider an ideal edge of height H occurring at a finite time n_1. Note that this edge is nothing but a step function denoted by H.$u(n-n_1)$. Thus, a window of width = 7 centered at $n_1 - 1$ would have the sample values $W(n_1-1) = \{0\ 0\ 0\ 0\ H\ H\ H\}$ in the window. These sample values are then sorted, scaled by $\alpha = [\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_3\ \alpha_2\ \alpha_1]$ and averaged. The entire operation would be :
$y^L(n_1\text{-}1) = \alpha.[\text{sort}\{W(n_1 \text{ - } 1)\}]^T$
$=[\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_3\ \alpha_2\ \alpha_1].[sort\{W(n_1-1)\}]^T$
$= \alpha_1.0 + \alpha_2.0 + \alpha_3.0 + \alpha_4.0 + \alpha_3.H + \alpha_2.H + \alpha_1.H.$
Note that the value of H should have a negligible effect on $y(n_1-1)$, but it is being given the same preference(i.e., being scaled by the same set of coefficients). This would lead to edge smearing.

The L filters are very hardware intensive because of the sorting operation. Very few works has been reported in the past on its hardware realization.

These disadvantages in the L filter can be removed by a bit-based weighted mean filter. The filter operates by searching for the median of the sample values in the filter window. The sample values are then scaled. Those values which lie in the proximity to the median(i.e. have more MSB's tallying with the median MSB's) are given more weightage. The weighted sample values are then averaged to yield the filter output.

The filtering algorithm operating on (2N + 1) k-bit long numbers $\{x(1), x(2) \ldots x(n = 2N+1)\}$ is given in Table 1.

In the algorithm:

β_ℓ : denotes the weights or the scaling coefficients and is always less than zero.

n_ℓ : total number of sample values that starts to differ from the median at the ℓth bit stage.

n_0 : total number of sample values which tallies with the median.

The algorithm can be mathematically denoted as:

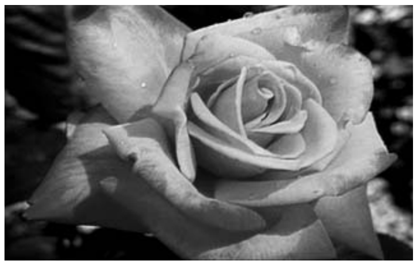
$$y = \frac{\sum_{l=1}^{k} \beta_l.\{x_l\} + median.n_0}{\sum_{l=1}^{k} \beta_l.n_l + n_0}$$

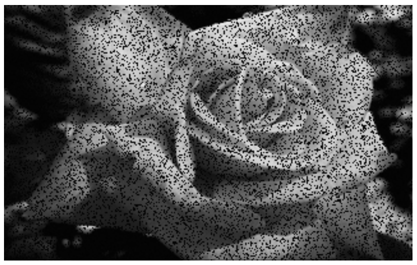
where, set $\{x_l\}$ is the set of sample values x(j) and x(j) = $median_{(k:l+1)}.2^l + x_{(l:1)}(n)$ where $x_{(l:1)}(n) \neq median_{(l:1)}(n)$ or $t_{(k:l+1)}(j) = 1$ and $t_{(l:1)}(j) = 0$.

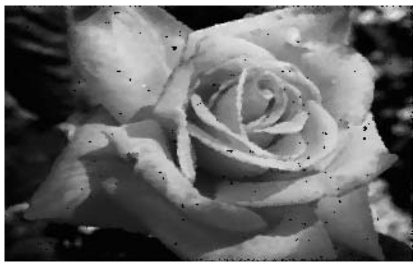
The algorithm calculates the median of a set of sample values in the filter window according to the algorithm

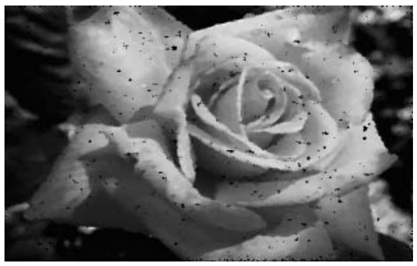
Nos.	10110	01011	00010	01110	01010	3rd R.B.	*The sum*
col 1 (MSB)	$\boxed{1}0110$	$\boxed{0}1011$	$\boxed{0}0010$	$\boxed{0}1110$	$\boxed{0}1010$	0	sum = β_5.(10110)
col 2	$1\boxed{1}111^*$	$0\boxed{1}011$	$0\boxed{0}010$	$0\boxed{1}110$	$0\boxed{1}010$	1	sum = β_4.(00010) + sum
col 3	$11\boxed{1}11$	$01\boxed{0}11$	$00\boxed{0}00^*$	$01\boxed{1}10$	$01\boxed{0}10$	0	sum = β_3.(01110) + sum
col 4	$111\boxed{1}1$	$010\boxed{1}1$	$000\boxed{0}0$	$111\boxed{1}1^*$	$010\boxed{1}0$	1	sum = sum
col 5	$1111\boxed{1}$	$0101\boxed{1}$	$0000\boxed{0}$	$1111\boxed{1}$	$0101\boxed{0}^*$	0	sum = sum + β_3.(01010)

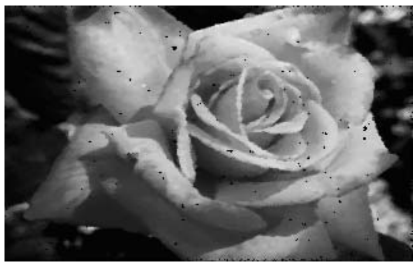
Table 2: The BWMF operating on n=5 and k=5.

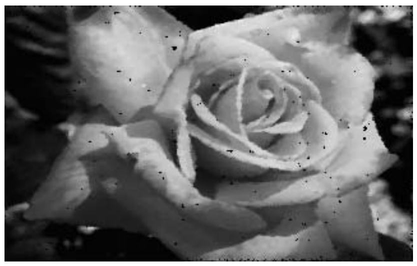

(a) Original Image.

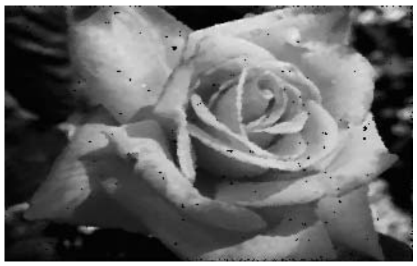

(b) Noisy Image.


(c) M Filter.


(d) Alpha Filter.


(e) B Filter.


(f) *TTMF*.


(g) BWM$_2$ Filter.

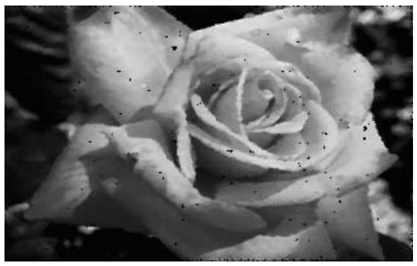

(h) Median Filter.

Figure 1: Effect of Filters on the image *Rose*

discussed earlier. If, at any bit stage(starting from the MSB) any of the sample value starts to differ from the median (basically we are looking for the transition of the tag bits from 1 to 0) the value is scaled with a particular weight associated with that bit stage. The median is scaled with the largest weight, i.e., '1'. These value are then averaged out to yield the filter output.

Note that this filter yields a variety of linear and non-linear filters by properly chosing the weights. The filters range from the mean filter ($\beta_l = 1|1 \leq l \leq k$) to the median filter ($\beta_l = 0|1 \leq l \leq k$). The weights depend largely on the distribution of the noise corrupting the signal. Calculation of the weights depending on the noise distribution will be discussed in a later publication.

An example of how this algorithm works is given in Table 2.

Output of the filter = (sum + 01011)/(β_5+β_4+β_3+2) and if $\beta_i = 2^{-i}$ then the output is equal to 01011.

The beauty of this algorithm lie in the fact that :

- Edge smearing is less than the L filter. This is due to the fact that the values which are more in concensus with the median (i.e. tally more with the median) are given more weightage. Let us consider the effect of an ideal edge at the filter output.

$$y(n_1 - 1) = \frac{\sum_{l=1}^{k} \beta_l . \{x_l\} + median.n_0}{\sum_{l=1}^{k} \beta_l . n_l + n_0}$$

$$= \frac{\beta_0.[0+0+0+0+0] + \beta_{\lceil log_2 H \rceil}.[H+H+H]}{\beta_{denom} = 5.\beta_0 + 3.\beta_{\lceil log_2 H \rceil}}$$

.

Now,

$$\frac{\beta_{\lceil log_2 H \rceil}}{\beta_{denom}} << \frac{\alpha_3 + \alpha_2 + \alpha_1}{3}$$

since $\alpha_3, \alpha_2, \alpha_1$ are weights for values nearer to the median in the L filter. Thus, $y(n_1 - 1)$ would be lower than $y^L(n_1 - 1)$ thereby lessening the smearing.

- The sorting operation of the L filter has been replaced by the median in the BWM filter thereby decreasing the hardware requirements significantly. The hardware implementation for the median computation has been discussed in great details in [3].

4 Results

For experimentation, the proposed filters and other related filters were applied on various noise distributions on 2-D grey level images. The noise consisted of two components - Impulse Noise (Impulse height depends inversely with the parameter R_{imp}) and Non-Impulse Noise (Variance = σ). Different distributions of noise (both impulse and non-impulse) can be best minimized based on the MAE criterion by a particular window size and a particular filter. Filters considered here are the median filter, M filter, α filter, B-filter, TTM-filter and the BWM$_2$-filter (BWM filters with $\beta_i = (1/2^i)$). The results presented in Table 3, show that the proposed filters outperform the B-filters and the TTM filters in cases where impulse noise and non-impulse noise is high. Figure 1 illustrates the original, noisy and the filtered outputs of the image *Rose* where $\sigma = 10$ and $R_{imp} = 5$.

Filter Type	Var. σ	R_{imp}	Win--dow	MAE	Best Flt
Median	10	10	(5,3)	334248	
M (m=7)	10	10	(5,3)	327551	
Alpha (α=5)	10	10	(5,3)	322582	
B (b=2)	10	10	(5,3)	320632	*
TTM (α=4)	10	10	(5,3)	341982	
BWM$_2$	10	10	(5,3)	332377	
Median	20	5	(3,3)	708203	
M (m=1)	20	5	(3,3)	698471	
Alpha (α=7)	20	5	(3,3)	909415	
B (b=2)	20	5	(3,3)	659301	
TTM (α=6)	20	5	(3,3)	609897	*
BWM$_2$	20	5	(3,3)	708203	
Median	10	5	(3,3)	421400	
M (m=1)	10	5	(3,3)	406892	
Alpha (α=3)	10	5	(3,3)	422521	
B (b=2)	10	5	(3,3)	385050	*
TTM (α=1)	10	5	(3,3)	391400	
BWM$_2$	10	5	(3,3)	391400	

Table 3: Performance on the Image *Rose*

References

[1] T.A.Nodes and N.C.Gallagher, "Median filter: Some modifications and their properties," *IEEE TASSP*, vol. ASSP-30, pp. 739-746,1982.

[2] N.S.Jayant, "Average and Median-based Smoothing techniques for improving Digital speech quality in the presence of transmission error," *IEEE TASSP*, pp. 1043-1045, Sep. 1986.

[3] B.K.Kar and D.K.Pradhan,"A New Algorithm for Order Statistic," *IEEE Trans. ASSP*, vol.41, pp. 2688-2694, Aug 1993.

[4] J.B.Bednar and T.L.Watt,"Alpha-Trimmed Means and Their Relationship to Median Filters", *IEEE TASSP*, vol.ASSP-32, pp.145-153, Feb 1984.

[5] A.C.Bovik, T.S.Huang, and D.C.Munson, Jr.,"A generalization of median filtering using linear combinations of their combinations of order statistics ," *IEEE TASSP* vol.ASSP-31, pp. 1342-1350, Dec 1983.

[6] Y.H.Lee and S.A.Kassam,"Generalized median filtering and related nonlinear filtering techniques," *IEEE TASSP*, vol.ASSP-33, pp. 672-683, June 1985.

[7] B.K.Kar *et al*,"Bit-serial Generalized Median Filtering Technique For Image Enhancement and their Implementation techniques," ISCAS-1994.