The FatBear: a nonarithmetic pico filter

Harold Longbotham[1,2], Mike Johnson[1], Jack Harris[1], Redouan Rouzky[1]

[1]Nonlinear Signal Processing Lab
University of Texas at San Antonio
San Antonio, Texas, 78249
harold@ennex1.eng.utsa.edu

[2]Conceptual MindWorks, Inc.
San Antonio, Texas

ABSTRACT

This paper is but a first step toward a fast, robust, nonarithmetic filtering theory. The following will assume the sampled sequence is from a finite TOS (totally ordered set) S so that any subset (window) of sequence values may be ordered. In addition we will assume the existance of a distance function (metric) on S. This distance function may be as simple as counting but all results presented hold for any TOS with any distance function. In addition a median type operator will be defined that always has as output an element in S, unlike the usual median operator.

The restriction to these operators will gaurentee that all results are exact, i.e. there is no quantization error introduced by the filter after sampling, for example there is no roundoff error, and no overflow. This is a direct result of the fact that no arithmetic operations are used. In addition there are no assumptions on the sampling interval or amplitude quantization interval. Therefore the results hold for nonuniform sampling and quantization.

Frequency based filtering is not the goal of this paper, although the real valued filters on which this paper is based have been applied to the robust detection of an instantaneous change in frequency and could be extended to frequency based communication systems in the future. It is assumed the signal and noise are "pulse based," i.e. they are pulses (not necessarily rectangular) with the delineating factor as to whether they are information or noise the "width" of the pulse. In addition it will be shown that corruption due to the smoothing of edges in a PICO signal may also be eliminated.

Implementations will be discussed. It will be shown that implementation involves only a two layer NAND/NOR for ordering and range estimations, and the assumption of counting. Operation counts will be disscussed.

0-8194-1279-1/93/$6.00

2. SIGNALS, OPERATIONS, AND DEFINITIONS

The signal to be considered is $\{x_n\}_{1,L}$ a finite length sequence of samples from an arbitrary set with unspecified (not necessarily uniform) sampling. Filters for pulse width filtering, impulse rejection, and edge enhancement will be examined. It is usually assumed that the range of values is $\{0,1\}^B$ (i.e. the range to be register contents), and that these are sampled signals stored in memory in integer form, but the following results will also hold on sets that may have been quantized or sampled on irregular intervals. It will be shown that the minimal structure needed is a totally orded set with a distance (metric) and a suitably defined median operation.

It is assumed all inputs are sequences of finite length, bounded on both ends by a constant segment [x_1 and x_L repeated N

times], where 2N+1 is the assumed length of each window.

The operations allowed will be the ordering of a finite set of values, calculating the distance from one value to another, and taking the median (to be defined latter, but different than the usual median) of a set of values. It is noted that in the case of an even number of values, in statistics the median is usually defined to be the average of the two central ordered values. Therefore our median needs to be somewhat similar or reduce to this value when the value exist, i.e. is a value in S. Example- median of three and six does not exist in integer arithmetic, in statistics it would be 4 1/2 but by our definition, if S is the set of integers, the median will be 4, the next lowest integer.

Let S be the set of values from which the sequence values are selected. Then a TOS is defined as follows. A relation R and a set S are called a POS (partially ordered set), when R is

(i) reflexive, i.e. xRx for all x in S

(ii) transitive, i.e. xRy & yRz => xRz

(iii) antisymmetric, i.e. xRy & yRx => x=y

The set S is totally ordered if for every x,y in S, either xRy or yRx, henceforth we will assume R is represented by $\leq$.

In general, a metric it is a function d from SxS to the nonnegative reals which has the following properties:

(i) $d(x,y) \geq 0$,

(ii) $d(x,y)=0 \iff x=y$

(iii) $d(x,y)=d(y,x)$

(iv) $d(x,z) \leq d(x,y)+d(y,z)$.

Two obvious metrics come to mind. For a finite set one could use the counting metric, i.e. d(x,y) = one more than the number of values between x and y, exclusive. This would only assume the ability to increment. A second metric one might use would be the usual absolute value of the difference (L_1 norm) if real values are allotted to the values in S.

The range of a set A, $R_A = MAX\{d(a,b), \text{ for } a,b \text{ in } A\}$. If x is any real number we will let $\lfloor x \rfloor = MAX\{ j \mid j \text{ is an integer and } j \leq x \}$. A constant region in $\{x_i\}$ is any L consecutive values with the same amplitude, if $L \geq N+1$. If x,y are elements of our totally ordered set S and $x \leq y$ then $MED_S(x,y) = MAX\{c \mid c \text{ is in } S \text{ and } d(x,c) \leq d(y,c)\}$. Therefore if $S = \{0,1,2,5,11\}$, the $MED_S\{0,2\} = 1$, and $MED_S\{0,5\} = 1$.

The above definition is fine for the MED of two values. But it must be extended to include the median of a set of more than 2 values. First consider a window length three, the median of 3 values would be the repeated value (if any) and the centrally ordered value if none are repeated. Therefore the definition for more than two values must contain the case of repeated values. There is the additional complication in that the median of a set with an even number of values may not necessarily be a value in the window. If there is an odd number of values in the set, simply order the values (where the ordering may include equality) and take the centrally ordered value. But if there is an even number of values, order the set of values, find the two centrally ordered values, and then take the MED of those two values as in the previous definition. Therefore we have the following definition for the median of a set of N values.

Def: Let A be a set of N values from a TOS, S, that may include repetitions, i.e.

$$A = \{x_i\}_{i=1,N} \text{ and } x_{(1)} \leq \ldots\ldots \leq x_{(N)}.$$

If N is odd, $MED(A) = x_{(N+1)/2}$.

If N is even $MED(A) = MED_S(x_{(N/2)}, x_{((N/2)+1)})$.

For example, assume S = {1, 2, 5/2, p, 10, 28}, then if (1) A = {2, 2, 10}, MED(A) = 2, (2) A = {2, 2, 10, 28}, MED(A) = MED(2, 10) = 5/2, if the distance is a simple counter, but if the distance function is the absolute difference the the output will be p.

Therefore, if we are assuming register/integer arithmetic with constant quantization intervals, if both values are even or odd the result is as usual, but if one is even and one is odd then there is one bit of resolution loss and this must be taken into account in all the proofs. The median will be defined so that in the above situation the median of an odd and even is the median of the original even value and the even value resulting from setting the least significant bit to 0 in the odd number. But remember we do not want the integer arithmetic to drive our work, we want the integer arithmetic to be an example that our work is valid on.

The FatBear filter, which is the main thrust of this paper, is now defined by the following output for each of the overlapping windows of length 2N+1,

$$y_i = FB\,[\{x_{i-n}, x_{i-n+1}, \ldots, x_{i+n}\}]$$

$$= MED\,\{\, x \mid x = MED\ A \text{ where A is in the class of subsets of } \{x_{i-n}, x_{i-n+1}, \ldots, x_{i+n}\} \text{ with N+1 values and with minimum range}\}.$$

The FatBear filter is an extension of the WMMR filter [1,2,3,4,5,6,7,8] and will be the only filter this paper is concerned with. The WMMR filters have similar properties when applied to real valued sequences, but use arithmetic operations.

3. RESULTS

The first five lemmas in this section describe how the Fat Bear filter eliminates impulses. Then techniques are described for using these results for low, high, and band pass filtering of pulse width signals. Theorems one and two then describe how the FatBear filter enhances certain corupted edges and put upper bounds on the number of filterings needed.

Lemma 1: Constant regions are fixed points (unchanged) upon filtering.

Proof: If a filter is centered on a point x in a constant region, then there is a unique set X of at least N+1 points in the window with range 0. The median of any subset of X with N+1 values is x and therefore the median of this resultant set of values is x.

Definition: If $L \leq N$, a burst of noise is a set of L consecutive values preceeded by and followed by constant regions each with constant regions having the same amplitude and the burst containing different values.

Lemma 2: A burst of noise is totally eliminated upon one pass of the filter.

Proof: Assume the window is centered on a value in the burst of noise. Then there are at least N+1 values in the window from the constant regions and therefore the central value is replaced by the value of the constant region.

Definition: $\{x_i\}$ is alternating if the value of x_i is A for i even and B for i odd, A„B.

Lemma 3: An alternating sequence is a fixed point (cycle of length 2) upon filtering with a window width 2N+1 with N even (odd).

Proof: Note that each window contains N+1 values equivalent to the central value if N is even and N+1 identical values different than the central value if N is odd.

Definition: A corrupted edge is two constant regions of different amplitudes seperated by L values, $L \leq N$, with at least one of the values distinct from the two constant regions.

Definition: A perfect edge may be of two types:

Type 1: Two adjoining constant regions with distinct amplitudes.

Type 2: Two constant regions with distinct amplitudes, seperated by a single point equidistance from each of the two distinct constant values and the median of the two constant values.

Lemma 4. Perfect edges are fixed points upon filtering.

Proof: Type 1 is obvious by Lemma 1. For a Type 2 we need only consider when the window is centered at the point between the two constant regions. But when centered at this point there are two distinct sets with the same minimum range. The median of each is the respective constant region and the median of the two constant values is the central point in the window.

Lemma 5: Due to the symmetry of the filter we may filter from the left to the right or from the right to the left with the same results.

Proof: Let X(n) be the windowed values of the sequence to be filtered. If we apply a filter W to the sequence X(n), from the left we can express the output as :

$$FB[\ x_{(n_0-N)},\ x_{(n_0-N+1)}, \ldots\ldots\ldots , x_{(n_0)}, \ldots\ldots\ldots, x_{(n_0+N-1)}, x_{(n_0+N)}\ ]$$

and the output of the sequence when we apply the filter from the right is :

$$FB[\ x_{(n_0+N)},\ x_{(n_0+N-1)}, \ldots\ldots\ldots , x_{(n_0)}, \ldots\ldots\ldots, x_{(n_0-N+1)}, x_{(n_0-N)}\ ]$$

where n_0 is the center of the window. Since the first step in the filtering process is the ordering of the windowed values and no spatial information is used in the filtering process, the two outputs are the same. Hence, the output is independent of the direction of motion of the window.

The above lemma describes how one would filter assuming the uncorrupted signal is PICO and the noise is a pulse width other than the PICO signal. The following three situations detail how one might construct low, high, and band pass filters: Low Pass - Assume the signal is PICO with the constant segments of minimum length M and the noise to be a pulse of width $K < M$. Then a FatBear filter of width 2N+1 with $K < N+1 \leq M$ would be used. The noise is totally eliminated if it is further than N samples from an edge. If it is closer to an edge, then the edge is corrupted and the following theorems apply. In all of this the following analogy applies and links the pulse domain to the frequency domain: A faster sampling rate may separate the frequency (pulse widths) delineating the noncorrupted signal and noise. High Pass - Assume the signal is PICO with maximum constant intervals of length M and noise in the form of pulses of length $K > M$. Then filtering with a window 2N+1 where $M < N+1 \leq K$ will remove the signal and leave the noise. Now the noise may be "subtracted" from the original signal. This method may introduce arithmetic errors, and modifications to avoid this are an active area of research. Band Pass - Assume the signal is PICO with constant regions of variable length M where for every M_U and M_L such that $M_L < M < M_U$. If the noise is of pulse width greater than M_U or less than M_L the modifications of the above will separate the signal from noise with errors steming from the edges and possibly subtraction.

Since this paper is to focus on errorless signal processing, only low pass filters will be of concern from this point on.

Definition: A^n means the value A repeated n times.

Definition: If x,y are in the totally ordered set X and $x \leq y$ but $x \neq y$, then we say $x<y$.

Since the effect of a burst of noise "close" to an edge is not necessarily eliminated by the above low, high, and band pass techniques, let us now consider the effect of such noise in more detail. The signal of interest is two constant regions of length 2N+1 or greater and a burst of noise of length N or less that occurs within N values of one constant region. Let us assume there are l values separating the burst of noise from the edge. The effect is the same as if the closest N-l values to the edge were perturbed by noise, since the first l values of the burst are perfectly restored by the filter on the first pass. Therefore, it is only necessary to consider the effect of a burst of noise imbedded in a constant region (as above) and the effect of the burst of noise at the edge between two constant regions (as below), assuming the signal is sampled fast enough to seperate burst withs of noise from those of the signal.

Theorem 1: Consider an edge of the type A^{N+1}, $\{x_i\}_{1,L}$,B^{N+1}, where $A<x_1<x_2 ... <x_L<B$, $L \leq N$. Convergence to a fixed point is gaurenteed in at most 3 passes.

Proof:

Consider L=1, then if it is a type 2 fixed point we are done. If not, then the value is not the median or is closer to one or the other constant region. In either case it is replaced upon one pass. Consider $1 < L \leq N$ and by Lemma 5 assume A<B. Note that the number of values in $\{x_i\}_{1,L}$ is at most $N=2\cdot|N/2|$ if N is even and $N=2|N/2|+1$ if N is odd, and that convergence will be slowest if all N values x_i are such that $MAX_i\{d(A,x_i)\} < (1/2)d(A,B)$ or if all N values x_i are such that $MAX_i\ d(B,x_i) < (1/2)$ d(A,B). WLOG we consider the case $d(A,x_i)<(1/2)d(A,B)$, for every i. Now we note that when the window is centered on x_i, $1 \leq i \leq |(N-1)/2|$, the output will be A since when the window is centered on the first $|N/2|$ values x_i, the minimum range set will have at least N+1-i values A in the set, therefore the output is A. On each pass another $|N/2|$ x_i will be replaced by A, therefore it will take at most 3 passes.

The inequalities allowed us to ignore ties, i.e. when we allow equalities there will be more sets with minimum range. Therefore the above theorem is weak and must be expanded to include equality. The case where more than one point in x_i (the corrupted edge) are equal distance from A and B and are also the MED(A,B), is left to another paper since this case has very low probability of occurance. If N is odd with N-1/2 values between A and MED (A,B) and N-1/2 values between B and MED(A,B) and one and only one point in $\{x_i\}$ is equal distance from A and B and is the MED(A,B) then that value is the central output upon first pass of the FatBear filter and fixed (does not vary) upon susequent passes. In most other cases the convergence is to a perfect edge. Next we consider the case of when $A<x_1, \leq x_2, ... <MED(A,B)<x_j, ... \leq x_L<B$ and note the convergence will be the slowest if all values x_i are strictly less than of greater than MED(A,B). Therefore the next theorem considers only this case.

Theorem 2: Assume a corrupted edge of the form $A^{N+1}<x_1 \leq x_2 \leq ... \leq x_L<MED(A,B)$, B^{N+1}, $1 \leq L \leq N$, then the corrupted edge converges to A^{N+L+1},B^{N+1}, in at most $|(N+1)/2|+2$ passes.

Proof: Worst case (slowest convergence), Assume L=N and $x_1=x_L$. When the window is centered at x_1, the following N sets have minimum range,

$(*)\ \{A^{N-i}, x_1^{i+1}\},\ 0 \le i \le N-1.$

Therefore the output is the calculated median of A and x_1.

When centered at x_2 there are N-1 sets with minimum range, they are

$(**)\ \{A^{N-i}, x_1^{i+1}\},\ 1 \le i \le N-1.$

The median of each of these sets is A, x_1, or MED(A,x_1), depending upon which is the most numerous in the set, or in the case of an equal number of each the output is MED(A,x_1) and may be distinct from A and x_1. In (*) we note the output will include an equal number of A and x_1 (and possibly one value MED(A,x_1)), the median of which is MED(A,x_1). But in (**) there will be fewer sets with output A than with x_1 and therefore the median is x_1. Obviously this extends to the other x_i, therefore after one pass the edge has the form

$$A^{N+1}, MED(A,x_1), x_1^{L-1}, B^{N+1}.$$

Now we note that at the next pass the value MED(A,x_1) is replaced by A and x_2 is replaced by a value $\le$ MED(A,x_1). This will continue untill $|(N+1)/2|$ values x_i are set to zero, then it will take only one pass to set the rest of the x_i to zero. At that point one pass will reduce the resulting signal to the desired fixed point. Therefore it will take at most $|(N+1)/2| + 2$ passes to converge to a perfect edge.

Since the above theorem is rather complex an example is now presented illustrating the result for N even or odd. Let the window size be 7, and let the number of values comprising the corrupted edge of the signal be equal to the maximum number alllowed, N, or 3 in this case. Also, let the values comprising the corrupted edge all be equal.

Let the sample space be all of the values available in an 8-bit register (0,1,...,255). Now, if N=3, let the data samples be : A=0, B=255, x=100. So the data appears as: ...,0,0,0,0,100,100,100,255,255,255,255,...

For the first pass of the filter:

Data in: ...,0,0,0,0,100,100,100,255,255,255,255...

Data out: ...,0,0,0,0,50,75,100,255,255,255,255...

Now, for the second pass:

Data in: ...,0,0,0,0,50,75,100,255,255,255,255...

Data out: ...,0,0,0,0,0,25,62,255,255,255,255...

The results after a third pass will be:

Data in: ...,0,0,0,0,0,25,62,255,255,255,255...

Data out: ...,0,0,0,0,0,0,12,255,255,255,255...

Likewise, after a fourth pass the output is:

Data in: ...0,0,0,0,0,0,12,255,255,255,255...

Data out: ...0,0,0,0,0,0,0,255,255,255,255...

It took four passes through the filter to eliminate the corrupted edge when N=3. Therefore, as a worst case scenario, if there are N bits of data comprising a corrupted edge, it will take at most $|(N+1)/2| + 2$ passes to achieve the desired results.

If N=4, using data similar to that of the previous example, and using the same techniques as before the results after the first pass are:

Data in: ...,0,0,0,0,100,100,100,100,255,255,255,255...
Data out: ...,0,0,0,0,50,100,100,100,255,255,255,255...

After a second pass:

Data in: ...,0,0,0,0,50,100,100,100,255,255,255,255...
Data out: ...,0,0,0,0,0,50,100,100,255,255,255,255...

After a third pass:

Data in: ...,0,0,0,0,0,50,100,100,255,255,255,255...
Data out: ...,0,0,0,0,0,0,25,50,255,255,255,255...

After a fourth pass:

Data in: ...,0,0,0,0,0,0,25,50,255,255,255,255...
Data out: ...,0,0,0,0,0,0,0,255,255,255,255...

So again, it takes $|(N+1)/2| + 2$ passes to eliminate the corrupted edge.

4. Implementation

Operations required for a single filter output at point p:

1) Select window of width 2N+1 (centered on p), and order those values using standard (stack filter) techniques.
2) Break the main window into subwindows, of width N+1, and find the range R_i of each subwindow.
   (There will be a total of N+1 subwindows)
3) Select the window(s) with minimum range.
4) Output at p is the median of the selected window, if unique. If more than one, take the median of the set of medians of all of minimum range windows.

Stack filtering terminology:

L = Number of grey levels possible

$b_{nm} = 1, \; X_n >= m$

$\quad\;\; 0, \; X_n < m$

(This corresponds to the binary value at the mth row of column n in the matrix below.)

• = Boolean AND

† = Boolean OR

~ = Boolean NOT

								or						
L	0	0	0	0	...	0		b_{1L}	b_{2L}	b_{3L}	b_{4L}	...	b_{NL}
...	...	...	...	...	...	...							
4	0	1	0	0	...	0		b_{14}	b_{24}	b_{34}	b_{44}	...	b_{N4}
3	0	1	0	1	...	0		b_{13}	b_{23}	b_{33}	b_{43}	...	b_{N3}

2	0	1	0	1	...	1	b_{12}	b_{22}	b_{32}	b_{42}	...	b_{N2}
1	1	1	0	1	...	1	b_{11}	b_{21}	b_{31}	b_{41}	...	b_{N1}

$X_1\ X_2\ X_3\ X_4 \ldots X_N \qquad X_1\quad X_2\quad X_3\quad X_4\quad \ldots\ X_N$

Value: 1 4 0 3 ... 2

1) Order the values in the main window:

The use of a stack filter for ordering of data values is well documented.

2) Range of a sequence of N values:

There are two (logically equivalent) approaches:

1) Range = $\Sigma\ (\sim b_{1m} \cdot b_{2m} \cdot b_{3m} \ldots b_{Nm}) \dagger (b_{1m} \cdot \sim b_{2m} \cdot b_{3m} \ldots b_{Nm}) \dagger \ldots$

Algebraic sum over L of all terms with 1 < # negated components < N

To find number of terms, use binomial distribution: $(^N{}_R) = N! / (N-R)!\ R!$

Total terms per row: $(^N{}_1) + (^N{}_2) + \ldots + (^N{}_{N-1})$ Number of rows = L

ex. N=5, (5 1)+(5 2)+(5 3)+(5 4) = 20 product terms (of N And's) per row

2) Range = $\Sigma \sim (b_{1m} \cdot b_{2m} \cdot b_{3m} \ldots b_{Nm} \dagger \sim b_{1m} \cdot \sim b_{2m} \cdot \sim b_{3m} \ldots \sim b_{Nm})$

Algebraic sum over L of all terms which are not all 0's and not all 1's

TOTAL No. of terms: 2 per row, each of N And's. No. of Rows = L

We can see that the second approach is less complex by a full order of magnitude in the example for N=5. In general, for all N>2, the second approach is significantly improved. This expression for product terms applies to the range over any number of values; the only thing which changes is the number of AND 's in each product term, which is equal to the number of values being ranged. The expression for generation of a range over the generic values $(X_{N+i} - X_i)$ is as follows:

$$\Sigma \sim (b_{im} \cdot b_{(i+1)m} \cdot b_{(i+2)m} \ldots b_{(N+i)m}) \dagger \sim b_{im} \cdot \sim b_{(i+1)m} \cdot \sim b_{(i+2)m} \ldots \sim b_{(N+i)m})$$

With a total window size of 2N+1, the requirements to find all the subwindow ranges is as follows, assuming a parallel implementation of all row operations:

TOTAL = (N+1) steps, with L parallel, 2 product-term computations per step

3) Find the minimum range(s):

If we set up the (N+1) resulting range values into a stack configuration like that used for the original data (this makes a parallel implementation over the rows much simpler), a boolean operation for finding the minimum is just the algebraic sum of all the row vectors which are all 1's.

$$MIN = \Sigma (b_{1m} \cdot b_{2m} \cdot b_{3m} \cdot \ldots b_{(N+1)m})$$

In addition, we can see which of the values are minimum(there may be several groups with that value) by examining the row vector at level (MIN+1) of the stack, since only minimum values will be take on binary value 0 at that level.

4) Find the median of the set(s) with minimum value range(s):

The output of this step is the final output in the case where there is only a single minimum range. We have already looked at a boolean mechanism for ordering, so all we need to do is pick out the median value:

Situation A: Odd number of values in the window (N is even). For this case, the median is simply the central value in the window., i.e., MED $(X_i - X_{N+i}) = X_{N/2+i}$, after ordering.

Situation B: Even number of values in the window. (N is odd)

For this case, we define the median of the window to be the median of the two centrally ordered values in the group. Because of our restriction to TOS, the output is the largest member of the TOS which is closer to the smaller of the two values than it is to the larger one. (If A and B are the two central values, MED(A,B) will be the output.

For the case where the TOS consists of binary 8-bit numbers interpreted to be in integer form, we can implement the distance between two values with a binary right shift (no carry), and use that as a counter for incrementing the lower value to the output. Also, if no shift functions are available, individual incrementing and range operations can be done until the range to the lower value exceeds the range to the higher, followed by a single decrement. Either of these is an exact implementation of the given definition for median.

4b) Find the median of the outputs from the previous step (necessary only when there are multiple minimum ranges)

If several of the windows are found to have minimum range, then we must once again take the median, this time of the set of medians found in the previous step. To determine the final output, use the same formulas described in the last step for Situation A (odd number of values in the group) and Situation B (even number of values in the group), as is consistent with our definition of the median operation for members of the totally ordered set.

TOTAL required operations:

Recall that the window (of length 2N +1) has already been ordered. Requirements for the ordering of these values have not been included here. A step is defined as a serial operation, requiring the previous step to end first. A computation is defined as a single two-level gate structure, with binary output.

Step 2: TOTAL (N+1) steps, with L parallel computations of 2 product terms each

Step 3: Incremental counter required. Max number of steps is L.
Each increment is a single product term of (N+1) ANDs.
TOTAL is L steps (max) , with 1 product term each

Step 4: Situation A requires but one step. Situation B requires a counter, with an absolute maximum of L/2 iterations of 4 steps (increment, find distance to A, find distance to B, and compare)
TOTAL 2L steps (max), with only L steps requiring a boolean operation, of at most L product terms

TOTAL REQUIREMENTS (after initial ordering of window):
3L+N+1 maximum steps - serial operations
L maximimum computations operating in parallel during a given step
Boolean expressions of up to 2L total product terms
Product terms of up to (N+1) individual AND terms.

5. References

[1]Harold Longbotham, David Eberly,"Statistical Properties, Fixed Points, and Decomposition With the WMMR Filters," invited paper for special issue of the Journal on the Mathematics of Imaging and Vision," published, Dec., 1992.

[2]Harold Longbotham, David Eberly,"The WMMR Filters: A Class of Robust Edge Enhancers," published, IEEE SP, April, 1993.

[3]Harold Longbotham, Walter Richardson,Dmitry Gokhman,"Design of Optimal Linear Operators for the Haar Basis," presented at and published in the proceedings of the Wavelets and Applications Conference, Toulouse, France, June 1992

[4]Daniel Shelton, Harold Longbotham,"Robust Time Domain Frequency Analysis," presented at and published in the proceedings of the SPIE/IS&T Conference on Image Processing (Nonlinear Signal Processing III), San Jose, California, 1992

[5]Harold Longbotham,"A Class of Robust Nonlinear Filters for Signal Decomposition Utilizing the Haar Basis," presented at and published in the proceedings of the IEEE ASSP Conference, San Francisco, California, March, 1992

[6]Harold Longbotham, David Eberly,"Fixed Points and Convergence Properties of Some Ordering Based Filters," presented at and published in the proceedings of the SPIE/IS&T Conference on Image Processing (Nonlinear Signal Processing III), San Jose, California, 1992

[7]Amy Glatt, Harold Longbotham, Thomas Arnow, Daniel Shelton, Peter Ravdin, "An Application of WMMR filters to Detection and Sizing of Tumors in Mammograms," presented at and published in the proceedings of the SPIE Conference on Medical Imaging, San Jose, California, Feb. 1992.

[8]Harold Longbotham and Norman Barsalou,"A Class of GOS Filters, Similar to the Median, that provide edge enhancement," presented at and published in the proceedings of the SPIE/SPSE Symposium on Electronic Imaging: Science and Technology, Nonlinear Image Processing Section, San Jose, Calif., February, 1991.