MEDIAN FILTERS THEORY AND APPLICATIONS

Milan STORK

e-mail: stork@kae.zcu.cz
University of West Bohemia, Faculty of Electrical Engineering, Department of Applied Electronics, 30614 Plzen, Czech Republic

Keywords: Median filter, recursive median filter, weighted median filter

ABSTRACT

The aim of this paper is to present of standard median and recursive median one-dimensional nonlinear filters. Median filtering is a popular method of noise removal, employed extensively in applications involving speech, signal and image processing. This non-linear technique has proven to be a good alternative to linear filtering as it can effectively suppress impulse noise while preserving edge information. The applications in industry and biomedical signal processing are presented in this article.

I. INTRODUCTION

The standard median (SM) filter is a simple nonlinear smoother that can suppress noise while retaining sharp sustained changes (edges) in signal values. It is particularly effective in reducing impulsive-type noise [1]. The output of SM filter at a point is the median value of the input data inside the window centered at the point. If is $\{x(k) \mid 1 \leq k \geq L\}$ and $\{y(k) \mid 1 \leq k \geq L\}$, respectively, the input and output of the one-dimensional (1-D) SM filter of window size $2N + 1$, then

$$y(k) = med\{x(k-N),..x(k - 1), x(k),x(k+1),..x(k+N)\} \quad (1)$$

Here to account for startup and end effect, $x(1)$ and $x(L)$, respectively, are repeated N times at the beginning and at the end of the input.

The recursive median (RM) filter is a modification of the SM filter defined in (1). Specifically, the output $y(k)$ of the RM filter of size $2N + 1$ is given by

$$y(k) = med\{y(k-N),... y(k - 1), x(k), x(k+1),..x(k+N)\} \quad (2)$$

At the beginning of the filtering, it is assumed that $y(1-N) = ... y(0) = x(1)$; the end effect is considered as a SM filtering. RM filtering can extract signal roots better than SM filtering, and is useful alternative to SM filtering in some applications. In general, RM filters are implemented by modifying an SM filtering algorithm [2] and, as a consequence, the implementation of RM filters is computationally and structurally more complex than that of SM filters.

In this paper, some practical examples of SM and RM filters are described.

II. SOME PROPERTIES OF MEDIAN FILTERS

In RM filter of window $2N+1$ is defined by replacing some of input samples in (1) by previously derived output samples in (2), so that the output of filter replaces the old input value before the filter window is moved to the next position. With the same amount of operation, the RM filter usually provides better smoothing capability than the SM filter, at the expense of increased distortion.

Frequency analysis and impulse response have no meaning in median and rank order filtering. The impulse response of a median filter is zero. As a result, new tools had to be developed to analyze and characterize the behavior of these nonlinear filters, deterministically and statistically. The basic descriptor of the deterministic properties of median filters is their root signal set, that is a set of signals which are invariant to further filtering [3]. The basic statistical descriptor of median filters is the set of output distributions which are used to study the noise attenuation properties of median filters.

The characterization of root signal is based on local signal structures, as defined in [4]. Those are summarized here for median filters of window width $2N+1$.

- A *constant neighborhood* is a region of at last N consecutive identically valued samples.
- An *edge* is a monotonically rising or falling set of samples surrounded on booth sides by constant neighbourhoods of different values.
- An *impulse* is set of at most N samples whose values ate different from the surrounding regions and whose surroundings regions are identically valued constant neighbourhoods.
- An *oscillation* is any signal structure which is not part of a constant neighbourhood, an edge or an impulse.

Example 1: Consider a length $N=2$ SM filter. Input vector $\boldsymbol{x} = [-1, 5, 8, 11, -2]$. After sorting the samples inside filter window are*:* $[11, 8, 5, -1, -2]$. The filter output $y = 5$ *(*Window is centered at the sample value 5).

For RM filter output computation is important point out that previous outputs $\{y(k-N),... y(k - 2)\}$ are not necessary to determine the present output $y(k)$, and are therefore redundant.

Property 1: In RM filtering, the output $y(k)$ is represented by

$$y(k) = med\{y(k-1), .., y(k-1), x(k), x(k+1),.., x(k+N)\} \quad (3)$$

Note: $y(k-1)$ is used N times in equation (3)!

Proof: The output $y(k)$ is a locally monotonic sequence of length $N+1$, that is, $\{y(k-N),..., y(k)\}$ is either nondecreasing or nonincreasing for any k [5]. Consider the nondecreasing case, $y(k-N) \leq ... \leq y(k-1)$. Since the output is locally monotonic of length $N+1$, $y(k-1) \leq y(k)$. This implies that if the number of samples in $\{x(k),...,x(k+N)\}$

that are greater than or equal to $y(k-1)$ is q, then $q \geq 1$ and clearly $y(k) = y(k-1)$ if $1 \leq q \leq N$, and $y(k) = min\{x(k),...,x(k+N)\}$ if $q = N+1$ because, otherwise, the local monotonicity is violated. Thus $y(k)$ is represented as

$y(k) = min\{x(k),... , x(k+N)\}$, if $x(k+i) \geq y(k-1)$
for all i, $0 \leq i \leq N$, $y(k) = y(k-1)$, otherwise, and
$y(k) = med\{y(k-1), .., y(k-1), x(k), x(k+1),..,x(k+N)\}$

This can be prove similarly for nonincreasing case.

Obviously, the implementation of RM filtering will become easier by using this property. The following property is a direct sequel to *Property 1* and is particularly useful in implementing 1-D RM filters on a general purpose computer.

Property 2: The output y(k) of the RM filter is given by

$$y(k) = med\{ x_{min} , x_{max}, y(k-1)\} \quad (4)$$

where x_{min} and x_{max}, respectively, are the minimum and the maximum of $\{x(k),...,x(k+N)\}$. By incorporating this property fast 1-D RM filters may be obtained.

The output of the RM filter, can be represented as a Boolean expression. For example, under the assumption that the input to this filter is restricted to binary values,

$$y(k)=y(k-1)\wedge[x(k)\oplus x(k+1)\oplus..\oplus x(k+N)]\oplus [x(k)\wedge x(k+1)..\wedge x(k+N)] \quad (5)$$

where represent the OR operation by $\oplus$, and the AND operation by $\wedge$. Fig. 1 illustrates the logic network realizing the Boolean function in (5). The output of SM filter with multilevel input sequence can be obtained from the output of binary SM filters. This has been done through either the threshold decomposition [6], or bit-serial approach [7].

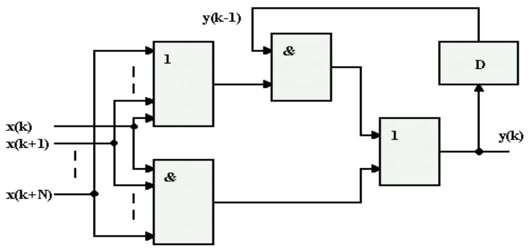

Figure 1: The logic network for binary RM filtering (window size $2N + 1$).

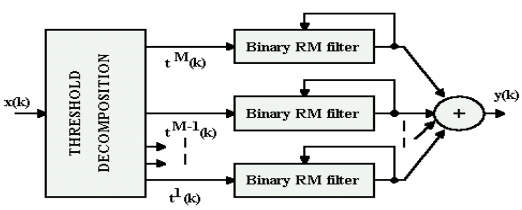

Figure 2: The implementation of an RM filter based on the threshold decomposition where $t^j(k) = 1$, if $x(k) \geq j$, $t^j(k) = 0$, if $x(k) < j$.

The SM filter for binary signals may by implemented by using counter followed by comparator [8], which is more complicated than the proposed RM filtering algorithm. In [6], in order to avoid the difficulty in implementing SM filters with digital logic, an analog circuit structure was developed. The analog circuit for SM filtering, however, may require development of a customized chip, which is usually expensive, while proposed digital structure admits inexpensive implementation using logic array chips.

RM filters have the threshold decomposition property [9], based on which they can be implemented, as shown in Fig. 2. Here the input is assumed to be an M-level signal.

The implementation based on the threshold decomposition property is parallel and modular, but its hardware complexity grows exponentially with the number of bits in the inputs.

Like in linear filtering, it is common to use cascades of median based filters for, e.g. analysis and filtering purposes. Theoretical analysis of median filter cascades is difficult. Fortunately, cascades of SM filters can be often represented by single weighted median (WM) filter.

Definition 1: For the discrete-time continuous-valued input vector $\boldsymbol{x} = [x_1, x_2,..., x_R]$, the output y of the WM filter of span N associated with the integer weights

$$\boldsymbol{w} = [w_1, w_2,..., w_R] \quad (6)$$

is given by

$$y = MED[w_1 \lozenge x_1, w_2 \lozenge x_2 ,..., w_R \lozenge x_R] \quad (7)$$

where $R=2N+1$ and $MED[.]$ denotes the median operation and $\lozenge$ denotes duplication, i.e.:

$$p \lozenge x = \overbrace{x , x ... x}^{p-times} \quad (8)$$

(x is used p times)

Example 2: Consider a length 5 WM filter with integer weights $[1, 2, 3, 2, 1]$. Now apply the filter to the following sequence so that the window is centered at the sample value 9.

$x = [-2, 4, 9, 12, -3]$

After sorting and duplication, the samples inside filter window are $[12, 12, 9, 9, 9, 4, 4, -2, -3]$. The WM filter output is $y=9$, whereas, the 5-point SM would have produced the result $y=4$.

By cascade connection of filters F and G we mean that the original input signal X is filtered by filter F which produces an intermediate signal Y. This in turn is filtered by filter G which produces the output signal Z. Cascaded filters F and G can be represented as a single filter H which produces the output directly from the input X. The maximum window length of filter H is $R_H = R_F + R_G - 1$, where R_F and R_G are window lengths of filters F and G, respectively.

Some cascade connection examples of SM filters are shown in Table 1.

$[1,1,1]^2 = [1,2,3,2,1]$

$[1,1,1]^3 = [1,2,5,7,5,2,1]$

$[1,1,1]^4 = [1,2,5,12,17,12,5,2,1]$

$[1,1,1]^5 = [1,2,5,12,29,70,99,70,29,12,5,2,1]$

$[1,1,1,1,1]\,[1,1,1] = [1,2,3,3,3,2,1]$

$[1,1,1,1,1]\,[1,1,1]^2 = [1,3,7,10,11,10,7,3,1]$

$[1,1,1]^2\,[1,1,3,1,1]^1 = [1,1,4,9,13,9,4,1,1]$

Table 1. Some cascade SM filters

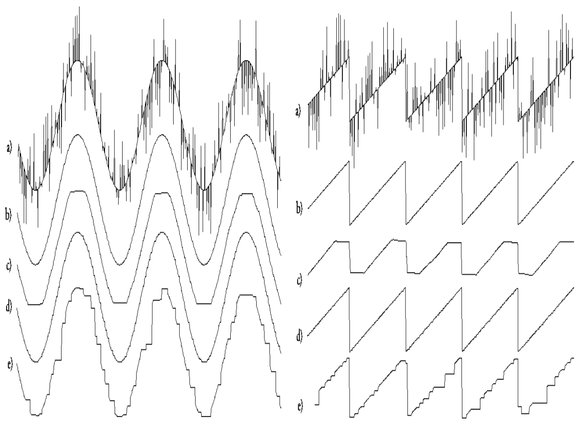

Figure 3: Example of filtering sinusoidal (left) and sawtooth (right) signal by means of SM and RM filters. a) input signal, b) SM filter *N=2*, c) SM filter *N=50*, d) RM filter *N=2*, e) RM filter *N=8*.

III. SERIAL BIT REALIZATION OF RM FILTER

Suppose that the input level $M = 2$ (P-bit signal). To simplify notation, denote $\{A_0, A_1 ... A_{N+1})\}$ the set of data in the window $\{y(k-1), x(k), ..., x(k+N)\}$ where $A_0 = y(k-1)$ and $A_i = x(k+i-1),\ 1 \le i \le N+1$. Let the code words (radix-2 binary representation) of A_i and $y(k)$ respectively, be $(a_i^1\ a_i^2 ...\ a_i^P)$ and $(b^1\ b^2 ...\ b^P)$ where a_i^1 and b^1 are the most significant bits. In the bit-serial realization, the output at each bit is obtained sequentially, starting with most significant bit. At each bit-level, with the exception of the most significant bit, the binary input values at the level are modified before filtering depending on the outputs of more significant bits. To implement the RM filter using the bit-serial approach and the binary RM filter, the input data at each bit should be modified as in the following property.

Property 3: The output of the binary RM filter at the *j*th bit is given by

$$b^j = \hat{a}_0^j \wedge [\hat{a}_1^j \oplus \hat{a}_2^j \oplus ... \oplus \hat{a}_{N+1}^j] \oplus [\hat{a}_1^j \wedge \hat{a}_2^j ... \wedge \hat{a}_{N+1}^j] \quad (9)$$

where $\hat{a}_i^1 = a_i^1,\ 0 \le i \le N+1$, and for each $j,\ 2 \le j \le P$,

$$\hat{a}_i^j = a_i^j,$$

if $a_i^m = b^m$, for all $m,\ 1 \le m \le j-1$, or (10)

$\hat{a}_i^j = a_i^r$, if $a_i^m = b^m$, for all $m,\ 1 \le m \le r-1$, and for some $r,\ 1 \le r \le j-1,\ a_i^r \ne b^r$.

This property was proved in [8]. Equation (7) can be rewritten as follows:

$$\hat{a}_i^j = \hat{a}_i^{j-1} \wedge U_i^j + a_i^j \wedge \bar{U}_i^j \quad (11)$$

where $U_i^j = [\hat{a}_1^{j-1} \otimes b^{j-1}] \oplus U_i^{j-1}$, and $U_i^1 = 0$. Exclusive-OR operation is represent by $\otimes$, and the complement operation by ($\bar{U}$), the OR operation by $\oplus$, and the AND operation by $\wedge$. This Boolean expression is useful for VLSI or microcomputer realization.

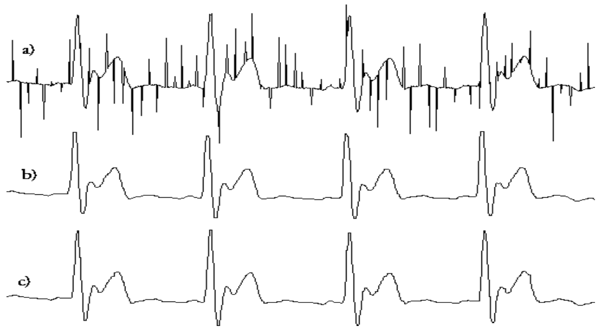

Figure 4: Example of ECG (electrocardiograph) signal filtered by means of SM and RM filters. a) input signal, b) SM filter $N = 4$, c) RM filter $N=2$. Sampling frequency = 960 Hz.

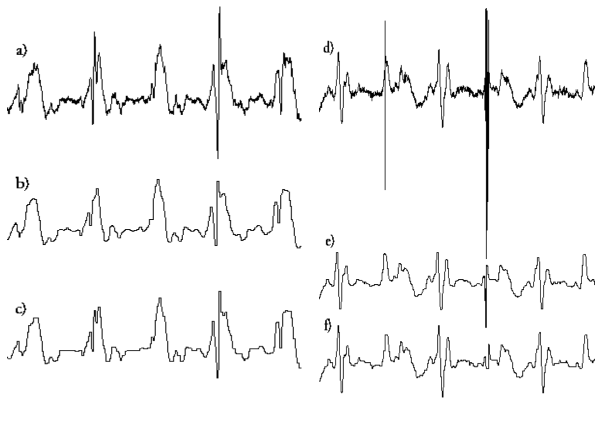

Figure 5: The signal from accelerometer filtered by SM and RM filters. a) Input signal, accelerometer fixed on arm, b) SM filter N=15, c) RM filter N=6 d) Input signal, accelerometer fixed on leg, e) SM filter N=10, f) RM filter N=8, Sampling frequency = 960 Hz., Physical activity - fast walk.

IV. APPLICATIONS EXAMPLES

In this part some examples of SM and RM are shown. The 1-D SM and RM filters were implemented on personal computer and tested for different types of data. Fig. 3 illustrates example of sinusoidal (left) and saw-tooth (right), contaminated with impulse noise filtered by SM and RM filters with different window size.

From results, presented in Fig. 3 is clearly visible the different results for SM and RM filters for different window size. RM filter is more sensitive for size than SM

filter.
Example of ECG signal filtering is shown in Fig. 4 [10]. One of the possibilities of the physical activity measuring can be the application of an accelerometer fixed on the human body. For medical purposes and "in field" activity testing, the device must be wireless. Therefore wireless data transmission was used. The received digital signal from accelerometer was also filtered by SM and RM filters.
Fig. 5 illustrates the signal from accelerometer fixed on arm and leg, during the fast walk [11]. This device was developed for medical purposes and physical activity measuring during sport activity or stress test.

V. CONCLUSION

The nonlinear filters were considered in this paper, they are SM and RM filters. Presented filters pointed out for their usefulness in impulsive environment.
In real world there are many signals which are corrupted by impulsive noise or have impulsive character. Signal and noise in this class are more likely to occur spikes or accidentally bursts of outlying observations than one would expect from normally distributed signals. As a result, their density functions decay in the tails less rapidly than Gaussian density function. Underwater acoustic signal, low-frequency atmospheric noise, radar, mobile communications, computer network traffic and many types of man-made noise have all been found to belong to this class. Nonlinear filters offers a flexible, robust approach to the problem of estimating signals in the presence of impulsive noise. Examples presented in this paper, show that in area of biomedical signal processing these filters could be useful too.

ACKNOWLEDGMENT

This paper is based upon work sponsored by the Ministry of Education of the Czech Republic under research and development project LN00B084.

REFERENCES

1. Huang, Ed., T. S.: Two-dimensional digital signal processing II: Transforms and median filters. New York: Springer-Verlag, 1981.
2. Richards, D. S.: VLSI median filters, IEEE Trans. Acoustic, Speech, Signal Processing, vol. 38, pp. 145-153, January, 1990
3. Yin L. - Yang R. - Gabbouj M. - Neuvo Y.: Weighted Median Filters: a Tutorial, IEEE Trans. On Circuits and Systems - II: Analog and Digital Signal Processing, vol. 43, No. 3, pp. 157-192, March 1996.
4. Gallagher, Jr., Wise, G., W.: A theoretical analysis of the properties of median filters, IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-29, pp. 1136-1141, December 1981.
5. Gallagher, N., C., Nodes, T., A.: Median filters: Some modifications and their properties, IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-30, pp. 739-746, December 1982.
6. Harber, R., G., Bass, C., S., Neudeck, G., L.: VLSI implementation of a fast rank-order algorithm, Proceedings ICASSP '85, Tampa, Florida, Marc 1985.
7. Chen, K.: Bit-serial realization of a class of nonlinear filters based on positive Boolean function, IEEE Trans. Circuits Systems, vol. 36, pp. 785-794, June 1989.
8. Wendt, P., D., Coyle, J., E., Gallagher, N., C.: Stack filters, , IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-34, pp. 898-911, August 1986.
9. Fitch, J., P., Coyle, J., E., Gallagher, N., C.: Median filtering by threshold decomposition, IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-32, pp. 1183-1188, December 1984.
10. Štork, M.: Telemetry System for ECG and Spiroergometric Data Transmission and Evaluation, SCS 2001, Internal Symposium on Signal Circuits and Systems, Proceedings, Iasi, Romania, ISBN 973-8050-99-5, pp. 137-139, July 2001.
11. Štork M., Holejšovská P.: Physical (Fitness) Activity Measuring Based on Accelerometer, International Conference Applied Electronics, ISBN 80-7082-758-0, University of West Bohemia, September Pilsen, 2001, pp. 224-227, 2001.

APPENDIX A
MEDIAN COST FUNCTION

Median filter output is the value θ which minimizes the cost function *F(θ)*

$$F(\theta) = \sum_{k=-N}^{N} | x(k) - \theta | \tag{A1}$$

and after differentiating:

$$\frac{d}{d\theta}F(\theta) = \frac{d}{d\theta}\sum_{k=-N}^{N}| x(k) - \theta | = \tag{A2}$$

$$= \sum_{k=-N}^{N} sign | x(k) - \theta |$$

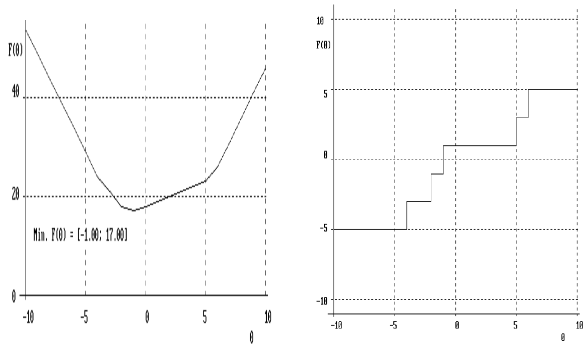

Fig. A1 (left) and A2 (right). A1 - median filter function, A2 - derivative of median filter function

Median filter example is computed for: x(-2) = 6, x(-1) = -2, x(0) = -4, x(1) = 5, x(2) = -1, after sorting [-4, -2, -1, 5, 6], median = -1. Graph of median cost function is shown in Fig. A1. Graph of derivative median function *F(θ)* is illustrates in Fig. A2.

APPENDIX B
ONE-DIMENSIONAL MEDIAN FILTER ALGORITHM

Given a sequence of n samples within a window:
$W = \{x_i\}$, where: $-N \leq i \geq N$. Each sample x_i is represented in *w* binary bits, as:

$$x_i = b_i^w \, b_i^{(w-1)} \, \ldots \, b_i^2 \, b_i^1$$

where b_i^k is the *k*-th bit of x_i with a weight of $2^{(k-1)}$.
Similarly, denoting the median *M*, to have the binary representation:

$$M = m^w \, m^{(w-1)} \, \ldots \, m^2 \, m^1$$

then the algorithm can be stated as folows:

1. Start with $j=w$
2. $m^j = majority\ (b^j_{-N}\ ,\ b^j_{(-N+1)}\ ,\ b^j_{(-N+2)} \ldots b^j_N)$
3. Do, in parallel, for all *i*, where: $-N \leq i \geq N$
   - If $b^j_i = m^j$, then bits of x_i do not change
   - If $b^j_i \neq m^j$ and $b^j_i = 1$, then all bits of x_i are set to ones
   - If $b^j_i \neq m^j$ and $b^j_i = 0$, then all bits of x_i are set to zeros
4. Decrement *j*. If $j \neq 0$, go to step 2
5. End

Example: Median{12, 2, 5}

```
12     2      5 Dec. numbers

H      L      L      => maj8(H,L,L)=L
H      L      H
L      H      L
L      L      H

in 1. row bit MSB = H and maj = L
=> all bits in 1. coll. are set to H
↓
H      L      L
H      L      H      => maj4(H,L,H)=H
H      H      L
H      L      H

in 2. row bit = L and maj = H
=> all bits in 2. coll. are set to L
       ↓
H      L      L
H      L      H
H      L      L      => maj2(H,L,L)=L
H      L      H

in 3. row bit = H and maj = L
=> all bits in 1. coll. are set to H
↓
H      L      L
H      L      H
H      L      L
H      L      H      => maj1(H,L,H)=H

maj8maj4maj2maj1 = LHLH => median = 5
```