
José Luis Paredes-Quintero
Escuela de Ingeniería Eléctrica
Universidad de Los Andes

CIMENICS’2002
Abril, 2002
Figure 1: Cooling Functions
Overview

- Motivation
- Traditional Stack Filters (Stack Smoothers)
- Mirrored threshold decomposition
- Stack filters
- Optimization algorithms for stack filter design
- Contributions and Conclusions
Motivation

- In many important applications, it is not only feasible but also necessary to consider more realistic non-Gaussian models and avoid the loss of accuracy and/or resolution resulting from the oversimplified Gaussian assumptions.

- Nonlinear filters can effectively deal with many noise-corrupted situations that involve impulsive, multiplicative or signal dependent noise.

- The human visual system includes some nonlinear effects that need to be considered in order to develop effective image and video processing algorithms.
Motivation (cont)

- The need for developing robust filtering structures to address frequency-selective applications arises.

- To extend the well known class of stack smoothers to a richer and powerful class of stack structures capable of addressing a number of multimedia and communication applications.

- To develop a theoretical tool to analysis the recently introduced class of WM filters admitting negative values weights [Arce’98].
Overview

- Motivation
- Mirrored threshold decomposition and stack filters
- Optimization algorithms for stack filter design
- Output distribution of stack filters
- Recursive weighted median filters admitting negative values weights
- Optimization of recursive WM filters
- Applications of WM filters with negative weights
- Dissertation contributions and Conclusions
- Future work
Filtering operation with STACK Smoothers

The output of the filter is obtained by decomposing the input signal into a set of binary signals, carrying out the filtering operation separately on each binary signal and then summing up the results.

\[
S(\cdot) = \frac{1}{2} \sum f(\cdot)
\]
Threshold Decomposition

- is a theoretical tool used to analyze stack smoothers
- maps an integer-valued signal to a set of binary vectors

Consider $\mathbf{X} = [X_1, X_2, \cdots, X_N]^T$ where $X_i \in \{-M, \cdots, M\}$.

The threshold decomposition of $\mathbf{X}$ is the set of binary vectors $\mathbf{x}^{-M+1}, \cdots, \mathbf{x}^0, \cdots, \mathbf{x}^M$ where $\mathbf{x}^m = [x^m_1, x^m_2, \cdots, x^m_N]^T$ and

$$x^m_i = T^m(X_i) = \begin{cases} 1 & \text{if } X_i \geq m; \\ -1 & \text{if } X_i < m, \end{cases}$$
Binary Filters (Stack Smoothers)

- Performs a Boolean operation over the binary signals. Example

\[ f(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3 \]

where

\[ x_i + x_j = \max(x_i, x_j) \quad x_ix_j = \min(x_i, x_j) \]

- \( f(\cdot) \) satisfies the stacking property. i.e. if \( u \) and \( v \) stack \( u_i \leq v_i \), their respective outputs stack \( f(u) \leq f(v) \)

- Positive Boolean functions (PBF)
  - contain no complements of input variables
  - satisfy the stacking property
PBFs and the corresponding integer domain filters

If \( f(\cdot) \) is linearly separable

it can be expressed as

\[
f(x_1, x_2, \cdots, x_n) = \text{sgn}
\sum_{i=1}^{M} W_i x_i - T
\]

\[
f(x_1, x_2, \cdots, x_n) \Rightarrow \text{Weighted Order Statistic}
\]

In addition

if \( f(x_1, x_2, \cdots, x_n) \) is selfdual \( f(x) = 1 \Rightarrow f(x) = -1 \)

\[
f(x_1, x_2, \cdots, x_n) \Rightarrow \text{Weighted Median Smoother}
\]
Threshold Decomposition leads to stack smoothers

Binary domain filters

- PBF
  - Linearly separable
  - Selfdual

Integer domain filters with non-negative weights

- WOS smoothers
- WM smoothers

Stack smoothers, however, are very limited

1. Analogous to FIR filters with non-negative weights

2. Useless in applications requiring "band-pass" or "high-pass" characteristics
Mirrored Threshold Decomposition (M.T.D.)

- emerges as a powerful theoretical tool that provides the foundation for the definition of Stack Filters
- maps the integer-valued signal to two sets of binary vectors

Consider \( \mathbf{X} = [X_1, X_2, \ldots, X_N]^T \) where \( X_i \in \{-M, \ldots, M\} \).

Define its mirrored vector as

\[
S = [-X_1, -X_2, \ldots, -X_N]^T \\
= \begin{bmatrix}
S_1, & S_2, & \ldots, & S_N
\end{bmatrix}^T
\]

\[
X_i \overset{M.T.D.}{\leftrightarrow} ([x_i^m]; \{s_i^m\})
\]
\[ x_i^m = T^m(X_i) = \begin{cases} 
1 & \text{if } X_i \geq m \\
-1 & \text{if } X_i < m 
\end{cases} \]

\[ s_i^m = T^m(-X_i) = \begin{cases} 
1 & \text{if } (-X_i) \geq m \\
-1 & \text{if } (-X_i) < m 
\end{cases} \]

**Example:** \( X = [2, -1, 0, -2] \)

\[
\begin{align*}
\mathbf{x}^2 &= \begin{bmatrix} 1 & -1 & -1 & -1 \end{bmatrix}^T \\
\mathbf{x}^1 &= \begin{bmatrix} 1 & -1 & -1 & -1 \end{bmatrix}^T \\
\mathbf{x}^0 &= \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}^T \\
\mathbf{x}^{-1} &= \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}^T \\
\mathbf{s}^2 &= \begin{bmatrix} -1 & -1 & -1 & 1 \end{bmatrix}^T \\
\mathbf{s}^1 &= \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix}^T \\
\mathbf{s}^0 &= \begin{bmatrix} -1 & 1 & 1 & 1 \end{bmatrix}^T \\
\mathbf{s}^{-1} &= \begin{bmatrix} -1 & 1 & 1 & 1 \end{bmatrix}^T.
\end{align*}
\]

- \( x^m, s^m \) satisfy the stacking constraints
- \( X_i, S_i \) are both reversible from their corresponding sets of decomposed signals
Mirrored Threshold Decomposition leads to stack filters

\[ SF(X_1, \cdots, X_N) = \frac{1}{2} \sum_{m=-M+1}^{M} f(x_1^m, \cdots, x_N^m; s_1^m, \cdots, s_N^m). \]

If \( f(\cdot) \) is linearly separable and selfdual

\[ Y = SF(X_1, \cdots, X_N) = \frac{1}{2} \sum_{m=-M+1}^{M} sgn \left( W^T x^m + |H|^T s^m - T_0 \right) \]

Its corresponding integer domain representation

\[ Y = \text{MEDIAN} \left( W_1 \diamond X_1, |H_1| \diamond S_1, \cdots W_N \diamond X_N, |H_N| \diamond S_N \right) \]
\[ Y = \text{MEDIAN}(W_1 \diamond X_1, |H_1| \diamond -X_1 \cdots W_N \diamond X_N, |H_N| \diamond -X_N) \]

\[ = \text{MEDIAN} \left( W_1 \diamond X_1, H_1 \diamond X_1 \cdots W_N \diamond X_N, H_N \diamond X_N \right) \]

\[ = \text{MEDIAN} \left( \langle W_1, H_1 \rangle \diamond X_1 \cdots \langle W_N, H_N \rangle \diamond X_N \right) \]

with \( W_i \geq 0, \ H_i \leq 0 \).

**Mirrored Threshold Decomposition** leads to WM filter in the integer domain whose samples are weighted twice, once positively by \( W_i \), and once negatively by \( H_i \).
Two weights better than one?

Single weight per sample [Arce’98]

\[ \hat{\beta} = \text{MEDIAN} \left( |W_1| \diamond sgn(W_1) X_1, \ldots, |W_N| \diamond sgn(W_N) X_N \right) \]

passing the sign of \( W_i \) to the sample \( X_i \), the rank information of \( X_i \) is lost, when \( W_i \) is negative.

Two weights per sample

\[ \hat{\beta} = \text{MEDIAN} \left( W_1 \diamond X_1, |H_1| \diamond sgn(H_1) X_1 \cdots \right) \]

takes into account the rank information of the original observed samples as well as their corresponding mirror samples
Figure 2: Samples are filtered by $[3, 2, 2, -3, 1]^T$ (solid line) and $[3, 2, 2, \langle 2, -3 \rangle, 1]^T$ (dashed line). The cost function shown is

$$G_1(\beta) = \sum_{i=1}^{N} (W_i |X_i - \beta| + |H_i| |X_i + \beta|)$$
Figure 3: Relationship among subclasses of stack filters and stack smoothers.
M. T. D. leads to STACK FILTERS

Binary domain filters

- PBF
  - Linearly separable
  - Self-dual

Integer domain filters
with negative and positive weights

- WOS Filters
- WM Filters

Methods of conversion

We also developed

- Methods of conversion PBFs ↔ WM filters
- Recursive stack filter structures and recursive WM filters
- Methods to synthesize RWM filters using non-recursive WM filters
<table>
<thead>
<tr>
<th>Stack Smoothers</th>
<th>Stack Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Defined in the binary domain of Threshold Decomposition</td>
<td>• Defined in the binary domain of Mirrored Threshold Decomposition</td>
</tr>
<tr>
<td>• Can synthesize lowpass filtering characteristics</td>
<td>• Can synthesize frequency-selective characteristics</td>
</tr>
<tr>
<td>• WM and WOS smoothers admitting positive weights</td>
<td>• WM and WOS filters admitting positive and negative weights</td>
</tr>
<tr>
<td>• Limited applications</td>
<td>• More powerful and versatile</td>
</tr>
</tbody>
</table>
Overview

- Motivation
- Mirrored threshold decomposition and stack filters
  - Optimization algorithms for stack filter design
- Output distribution of stack filters
- Recursive weighted median filters admitting negative values weights
- Optimization of recursive WM filters
- Applications of WM filters with negative weights
- Dissertation contributions and Conclusions
- Future work
Optimization of Stack Filters

GOAL: Find the optimal PBF that minimizes a performance cost function

The basic idea is to find for each possible binary vector the entry of the truth table that defines the Stack Filter

\[
\begin{array}{cccccc}
  x_1 & x_2 & x_3 & s_1 & s_2 & s_3 \\
  \hline
  -1 & -1 & -1 & -1 & -1 & -1 \\
  -1 & -1 & -1 & -1 & -1 & +1 \\
  \vdots & & & & & \\
  f(x_1, x_2, x_3; s_1, s_2, s_3) & ? & & & & \\
  \vdots & & & & & \\
  \end{array}
\]

How many decision variables have to be optimized?

22
Thresholded Signals Outputted by M.T.D.

**Theorem 1** For an observation window of size $N$, the number of possible binary vectors of length $2N$ that can be outputted by the mirrored threshold decomposition is $N^* = 2(3^N) - 2^N$.

- Not all the $2^{2N}$ possible binary vector of length $2N$ are generated by M.T.D.
- $N^*$ defines the number of entries of the truth table that describes the stack filter of window size $N$.
- Consequently, $N^*$ defines the number of decision variables to be optimized.
Stack Filter Optimization

Under the MAE criterion the goal is to find the $2N$-variable PBF, or equivalently the truth table, so as to minimize the cost function

$$J(f) = E|D(n) - SF(X(n))|$$

Using mirrored threshold decomposition,

$$J(f) = \frac{1}{2} E \sum_{m=-M+1}^{M} d^m(n) - f(x^m(n); s^m(n))$$

It was shown in the dissertation that $J(f)$ reduces to

$$J(f) = \frac{1}{2} E \sum_{m=-M+1}^{M} E|d^m(n) - f(x^m(n); s^m(n))|$$
Thus, the multilevel mean absolute error criterion reduces to the sum of the absolute errors on each threshold level.

The minimization of the cost function $J(f)$ reduces to solving the linear program

$$
\min_{z} \sum_{i=1}^{N^*} C_i z_i
$$

subject to the constraints

$$
z_i \leq z_j \quad \text{if} \quad \xi_i \leq \xi_j \quad \text{and} \quad H(\xi_i, \xi_j) = 1
$$

$$
-1 \leq z_i \leq +1
$$

where $\xi_i = [x_1, \ldots, x_N; s_1, \ldots, s_N]$, $z_i = f(\xi_i)$ and $H(\cdot, \cdot)$ is the Hamming distance.
Note that the linear program leads to a PBF that makes soft decision.

Taking on continuous values, \( z_i \) can be expressed as a function of the probability that the PBF \( f(\cdot) \) outputs \(+1\) when the vector \( \xi_i \) is observed. That is,

\[
z_i = 2 \text{Prob}(+1|\xi_i) - 1
\]

- Complexity of the linear program is of polynomial type
- The number of variables in the linear program grows by a factor \( 2(3^N) - 2^N \)
- Computationally prohibitive when the size of the observation window is greater than 5
- Assume that the signal statistics are known
Adaptive Optimization Algorithm

- We allow $f(\cdot)$ to take continuous values in the interval $[-1, 1]$
- At each iteration, the adaptive algorithm tries to increase the probability that the PBF $f(\cdot)$ makes the correct decision when $\xi_i$ is observed
- Consequently, the cost function $J(\cdot)$ is minimized
- The updating operation has to take into account the stacking property, i.e., $z$, or equivalently $f(\cdot)$, has to satisfy the stacking constraint
Adaptive Optimization Algorithm

- **Step I**
  Increase the probability that the PBF $f(\cdot)$ makes the correct decision when $\xi_i$ is observed

  $$z(n') = z(n' - 1) + \mu(n')d(n')1_i$$

- **Step II**
  Check if the stacking constraints are satisfied after the update. If a stacking constraint is violated, pick the $z_j$ for which the stacking constraint is violated and perform a swap operation between $z_i$ and $z_j$

  $$z(n') = z(n') + \mu(n')d(n')(1_j - 1_i)$$
\( \mu_z(n') \) is the step-size of the algorithm. It is adjusted as the training progresses to ensure that at any iteration all the components of the decision vector \( z(n') \) are in the interval \([-1, 1]\).

\[
\mu(n') = \begin{cases} 
0 & \text{if } |z_i(n' - 1) + \mu_0 d(n')| > 1 \\
1/N' & \text{otherwise},
\end{cases}
\]

where \( N' \) is a large positive integer number.

At the end of the optimization algorithm, the optimal soft decision vector is mapped to a binary vector through the use of a soft-to-hard transformation.

\[
\hat{z} = \text{sgn}(z)
\]
Figure 4: (a) two-tone input signal, (b) desired signal, (c) optimal stack filter output, (d) FIR filter output, (e) optimal WM filter output (f) optimal stack smoother output
Figure 5: (a) two-tone input signal in $\alpha$-stable noise, (b) optimal stack filter output, (c) FIR filter output, (d) optimal WM filter output (e) optimal stack smoother output
Figure 6: (a) two-tone input signal, (b) desired signal, (c) optimal stack filter output, (d) FIR filter output, (e) optimal WM filter output (f) optimal stack smoother output
Overview

- Motivation
- Mirrored threshold decomposition and stack filters
- Optimization algorithms for stack filter design
- **Output distribution of stack filters**
- Recursive weighted median filters admitting negative values weights
- Optimization of recursive WM filters
- Applications of WM filters with negative weights
- Dissertation contributions and Conclusions
- Future work
Output distribution of stack filters

- We develop statistical analysis tools for the proposed class of stack filters.
- We derive the cumulative distribution formula of the output of any stack filter as a function of the input noise distribution.
- The sliding window operation of the stack filter is modeled by a deterministic finite automaton.
- The output distribution of the filter is obtained by interpreting the automaton as a Markov Chain whose transition probabilities depend on the probabilistic description of the binary input signal resulting from the mirrored threshold decomposition operation.
Deterministic finite automaton

Definition 1 A deterministic finite automaton (dfa) is a system \( M = (A, Q, V, \delta, \lambda) \) in which \( A = \{a_1, \ldots, a_m\} \) is the input alphabet, \( V = \{v_1, \ldots, v_k\} \) is the output alphabet, \( Q = \{q_1, \ldots, q_n\} \) is the set of states, and \( \delta : Q \times A \rightarrow Q \) and \( \lambda : Q \times A \rightarrow V \) are the transition and output functions, respectively.
## Modeling Stack Filters by a Deterministic Finite Automaton

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$x$</th>
<th>$s$</th>
</tr>
</thead>
</table>

### Set of States $Q$

- $q_1 = \xi_1$, $q_1 = \xi_2$ \ldots $q_{N^*} = \xi_{N^*}$

- $a_1 = (-1, -1)$
- $a_2 = (-1, +1)$
- $a_3 = (+1, -1)$
- $a_4 = (+1, +1)$

### Input Alphabet $A$

### Output Alphabet $V$

- $v_1 = -1$, $v_2 = +1$

### Output Function $\lambda : Q \times A \rightarrow V$

- Stack filter operation

### Transition Function $\delta : Q \times A \rightarrow Q$

- Filtering operation
Theorem 2  The output cumulative distribution function of the stack filter
defined by the positive Boolean function \( f : \{-1, +1\}^{2N} \rightarrow \{-1, +1\} \)
is equal to

\[
\Psi(t) = \sum_{v : f(v) = -1} F_{+,+}^{w(x \land s)}(t) \cdot F_{-,+}^{w(x \land \bar{s})}(t) \cdot F_{+,+}^{w(\bar{x} \land s)}(t) \cdot F_{-,+}^{w(x \land \bar{s})}(t)
\]

where

\[
v = \begin{bmatrix} x^T ; s^T \end{bmatrix}^T = [x_1, \ldots, x_N, s_1, \ldots, s_N]^T
\]

\( w(a) \) is the Hamming weight of the vector \( a \)

\[
F_{+,+}(t) = \text{Prob} \{ x = +1, s = +1 \} = \begin{cases} 
F(-t) - F(t) & \text{if } t \leq 0 \\
0 & \text{if } t > 0
\end{cases}
\]
Example

For the stack filter

\[ f(x_1, x_2, x_3; s_1, s_2, s_3) = x_1 s_2 + x_2 s_3 + x_3 + s_1 \]

Figure 7: a) Input \( \mathcal{N}(0, 1) \) and output probability density functions. (b) Input and output cumulative distribution functions.
Overview

- Motivation
- Mirrored threshold decomposition and stack filters
- Optimization algorithms for stack filter design
- Output distribution of stack filters
- Recursive weighted median filters admitting negative values weights
- Optimization of recursive WM filters
- Applications of WM filters with negative weights
- Dissertation contributions and Conclusions
- Future work
Definition 2 (Recursive WM filters) Given an $N$-input observation $X_k = [X_{k-L}, \ldots, X_k, \ldots, X_{k+L}]^T$, the recursive WM is obtained by replacing the leftmost $L$ samples of the input vector $X_k$ with the previous $L$ output samples $Y_{k-L}, Y_{k-L+1}, \ldots, Y_{k-1}$. That is

$$Y_k = \frac{1}{2} \sum_{m=-M+1}^{M} \text{sgn} \left( W_R^T r_k^m + |H_R|^T z_k^m \right)$$

where

$$X_k' = [Y_{k-L}, \ldots, Y_{k-1}, X_k, \ldots, X_{k+L}]^T$$

$$(\{r_k^m\}; \{s_k^m\}) \overset{M.T.D.}{\leftrightarrow} X_k'$$

$$W_R = [W_{R-L}, \ldots, W_{R_0}, \ldots, W_{R_L}]^T$$

$$|H_R| = [|H_{R-L}|, \ldots, |H_{R_0}|, \ldots, |H_{R_L}|]^T$$
\[ Y(n) = \text{MEDIAN} \left( |A_\ell| \odot \text{sgn}(A_\ell) \ Y(n - \ell) \big|_{\ell=1}^{N}, \right. \\
\left. |B_k| \odot \text{sgn}(B_k) \ X(n + k) \big|_{k=0}^{M} \right) \]
Properties of Recursive WM filters

- RWM filters offer a number of advantages over their non-recursive counterparts including:
  - A significant reduction in computational complexity
  - Increased robustness to noise
  - The ability to model “resonant” or vibratory behavior

- Unlike linear IIR filters, RWM filters are always stable under the bounded-input bounded-output criterion, regardless the values taken by the feedback filter weights

- RWM filters are more robust that linear IIR filters
Overview

- Motivation
- Mirrored threshold decomposition and stack filters
- Optimization algorithms for stack filter design
- Output distribution of stack filters
- Recursive weighted median filters admitting negative values weights
  - Optimization of recursive WM filters
- Applications of WM filters with negative weights
- Dissertation contributions and Conclusions
- Future work
Optimization Algorithms for the design of RWMF

• In general, the coefficients of the recursive WM filter have to be designed in some optimal fashion

• The main objective of the optimization is to find the best filter coefficients such that a performance cost criterion is minimized

• Under the MAE criterion the goal is to determine the weights $\{A_\ell\}_{\ell=1}^M$ and $\{B_k\}_{k=0}^M$ so as to minimize the cost function

$$J(A_1 \cdots A_N, B_0 \cdots B_M) = J(A, B) = E\{|D(n) - Y(n)|\}$$

where $\{X(n)\}$ is a process statistically related to a desired process $\{D(n)\}$
Adaptive Optimization Algorithm

- To form an iterative optimization algorithm, the steepest descent algorithm is used in which the filter coefficients are updated according to

$$A_{\ell}(n + 1) = A_{\ell}(n) + 2\mu[-\frac{\partial}{\partial A_{\ell}} J(A_1, \cdots, A_N, B_0, \cdots, B_M)]$$

$$B_k(n + 1) = B_k(n) + 2\mu[-\frac{\partial}{\partial B_k} J(A_1, \cdots, A_N, B_0, \cdots, B_M)]$$

- Due to the feedback operation inherent in the recursive WM filter, however, the computation of gradient of $J$ becomes intractable.
\[ Y(n) = \text{MEDIAN} \left( |A_\ell| \odot \text{sgn}(A_\ell) Y(n - \ell) \right)_{\ell=1}^{N}, \]
\[ |B_k| \odot \text{sgn}(B_k) X(n + k) \bigg|_{k=0}^{M} \]
To overcome this problem, we use the fact that ideally the filter’s output is close to the desired response. Hence, the previous outputs \( \{Y(n - \ell)\}_{\ell=1}^N \) are replaced with the previous desired outputs \( \{D(n - \ell)\}_{\ell=1}^N \), namely,

\[
\hat{Y}(n) = \text{MEDIAN} \left( |A_\ell| \diamond \text{sgn}(A_\ell) D(n - \ell)\right)_{\ell=1}^N,
\]

\[
|B_k| \diamond \text{sgn}(B_k) X(n + k)\right)_{k=0}^M
\]

This approximation leads to an output \( \hat{Y}(n) \) that does not depend on delayed output samples and, therefore, the filter no longer introduces feedback, reducing the RWM filter to a two-input, single output non-recursive system.
Recursive-Decoupling Optimization Algorithm

\[ X(n+M) \rightarrow Z^{-1} \rightarrow B_0 \rightarrow X(n+1) \rightarrow Z^{-1} \rightarrow B_1 \rightarrow \ldots \rightarrow Z^{-1} \rightarrow B_M \rightarrow X(n+M) \]

\[ D(n-N) \rightarrow A_1 \rightarrow D(n-1) \rightarrow A_2 \rightarrow \ldots \rightarrow A_N \rightarrow D(n-N) \]

\[ Y(n) = \sum_{i=0}^{M} X(n+i) B_i + \sum_{i=1}^{N} D(n+i) A_i \]

Adaptive algorithm
According to the approximate filtering structure, the cost function to be minimized is

\[ \hat{J}(A_1, \cdots, A_N, B_0, \cdots B_M) = E\{|D(n) - \hat{Y}(n)|\} \]

where \( \hat{Y}(n) \) is the non-recursive filter output

\[ \hat{Y}(n) = \text{MEDIAN} \left( |A_\ell| \diamond \text{sgn}(A_\ell) D(n - \ell)|_{\ell=1}^{N}, \right. \]
\[ |B_k| \diamond \text{sgn}(B_k) X(n + k)|_{k=0}^{M} \]\n
Since \( D(n) \) and \( X(n) \) are not functions of the filter coefficients, the derivative of \( \hat{J}(A_1, \cdots, A_N, B_0, \cdots B_M) \) with respect to the filter weights is non-recursive and its computation is straightforward.
Recursive-Decoupling Optimization Algorithm

\[
A_\ell(n + 1) = A_\ell(n) + \mu(D(n) - \hat{Y}(n)) \, \text{sgn}(A_\ell(n)) \, s_{D_\ell}^{\hat{Y}(n)}
\]

\[
B_k(n + 1) = B_k(n) + \mu(D(n) - \hat{Y}(n)) \, \text{sgn}(B_k(n)) \, s_{X_k}^{\hat{Y}(n)}
\]
Applications of Recursive WM filters

- **Image Denoising**
  In this first example, the performance of the RWM filters is compared with that of the non-recursive counterpart in image denoising

- **Design of a Band-Pass RWM filter**
  The performance of the resultant filters is compared with that of a linear IIR filter and of a non-recursive WM filter designed for the same application
Figure 8: (a) Original, (b) noisy, (c) non-recursive center WMF, (d) optimal non-recursive WMF, (e) recursive center WMF, (f) optimal RWMF
Figure 9: (a) Original, (b) $\alpha$-stable noise, (c) non-recursive center WMF, (d) optimal non-recursive WMF, (e) recursive center WMF, (f) optimal RWMF
Table 1: Results for impulsive noise removal

<table>
<thead>
<tr>
<th></th>
<th>salt and pepper</th>
<th>$\alpha$-stable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>MAE</td>
</tr>
<tr>
<td>Noisy image</td>
<td>2545.20</td>
<td>12.98</td>
</tr>
<tr>
<td>Recursive center WM filter</td>
<td>189.44</td>
<td>1.69</td>
</tr>
<tr>
<td>Non-recursive center WM filter</td>
<td>243.83</td>
<td>1.92</td>
</tr>
<tr>
<td>Optimal non-recursive WM filter</td>
<td>156.30</td>
<td>1.66</td>
</tr>
<tr>
<td>Optimal RWM filter</td>
<td>88.13</td>
<td>1.57</td>
</tr>
</tbody>
</table>
Figure 10: (a) Input test signal, (b) desired signal, (c) linear FIR filter, (d) non-recursive WM filter, (e) linear IIR filter, (f) recursive WM filter outputs
Figure 11: (a) Chirp test signal in stable noise, (b) linear FIR filter, (c) non-recursive WM filter, (d) linear IIR filter, (e) recursive WM filter outputs
Figure 12: Frequency response to (left) a noiseless sinusoidal signal and (right) a noisy sinusoidal signal; (—) RWM, (— • — • —) non-recursive WM filter, (---) linear FIR filter, and (----) linear IIR filter.
Overview

- Motivation
- Mirrored threshold decomposition and stack filters
- Optimization algorithms for stack filter design
- Output distribution of stack filters
- Recursive weighted median filters admitting negative values weights
- Optimization of recursive WM filters
- Applications of WM filters with negative weights
- Dissertation contributions and Conclusions
- Future work
Dissertation contributions

- Presents a theoretical tool (mirrored threshold decomposition) to analyze WM filters admitting real-valued weights
- Introduces a more general and richer class of stack filters based on mirrored threshold decomposition and shows that these filters have been empowered with bandpass and highpass filtering characteristics
- Develops adaptive optimization algorithms for the design of stack filters in the binary domain of mirrored threshold decomposition
- Develops statistical analysis tools for the class of stack filters by modeling the stack filtering operation using a deterministic finite automata
Dissertation contributions

• Introduces the class of recursive WM filters with real-valued weights and demonstrates that these filters have superior performance than non-recursive WM filters and than linear IIR filters

• Proposes and develops adaptive optimization algorithms for the design of recursive WM filters

• Presents a image sharpening structure based on WM filters that is robust and insensitive to noise and artifacts introduced by compression algorithms
Future Work ...

- Extend the applications of the proposed class of stack filters
- Look at the class of Weighted Order Statistic admitting real-valued weights
- Develop blind optimization algorithms for the design of the proposed filters
- Study the convergence of the proposed adaptive algorithms
- Establish a collaborative partnership between University of Los Andes, Venezuela and U.D. Signal Processing group