IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on YYYY–MM–DD.
ISSN 2105–1232 
This article is available online with supplementary materials, software, datasets and online demo at
http://www.ipol.im/

PREPRINT July 21, 2013

# Block thresholding audio denoising algorithm

Marie de Masson d'Autume,
Christophe Varray
Eva Wesfreid

**Abstract**

This paper describes and analyzes the Guoshen Yu et al. non-diagonal block thresholding audio denoising algorithm. This algorithm attenuates the Short Time Fourier Transform coefficients of the noisy signal over a time-frequency block partition which is adapted to signal properties using the SURE theorem. Numerical experiments are performed over music extracts to compare denoised and underlying clean signals.

The ANSI C codes and data sound files used for experimentation are available on line. The Matlab code which uses these C functions is also on line.

## 1 Introduction

The Guoshen Yu et al. algorithm [9] focuses on audio signals corrupted with additive Gaussian white noise. This noise is specially hard to remove without damaging the underlying audio signal since it is located in all frequencies.
The described block thresholding algorithm attenuates the Short Time Fourier Transform (STFT) coefficients of the noisy signal by blocks using the same attenuation factor over each block. This non diagonal processing is different from the diagonal processing, which attenuates each coefficient independently without using potential dependencies between neighbor coefficients. This procedure introduces isolated time-frequency artefacts called *musical noise*. Y. Ephraim and D. Malah [4], [5] showed that *musical noise* is strongly attenuated with non-diagonal time-frequency estimators.
The Guoshen Yu et al. algorithm is based on the block thresholding estimators introduced by T. Cai [1] which choose a block time-frequency partition minimizing the Stein Unbiased Risk Estimate (SURE) [8]. The attenuated coefficients over this adapted block partition are smoothed with an ideal Wiener filter before reconstruction with the inverse STFT to obtain the denoised signal.
Many experiments show that in certain frequency bands the spectrum of the noisy signal masks the spectrum of the original one. The sound of the denoised signal is often damaged since a part of the underlying signal spectrum is removed after denoising.
This paper describes the STFT algorithm in Section 2. Section 3 details the Wiener filter. Section 4 introduces the different steps of block thresholding algorithm with the SURE estimation. Finally, experiments and results are shown in section 5 and the conclusion can be found in section 6.

# 2 Short Time Fourier Transform

We denote

$$\hat{f}(k) = \sum_{n=1}^{N} f(n) exp(\frac{-2i\pi kn}{N}) \tag{1}$$

the discrete Fourier transform of a signal $f = (f(n))_{1 \leq n \leq N}$ and

$$f(n) = \frac{1}{N} \sum_{k=1}^{N} \hat{f}(k) exp(\frac{2i\pi kn}{N}) \tag{2}$$

its inverse. The STFT decomposes a signal $f = (f(n))_{1 \leq n \leq N}$ into time-frequency atoms:

$$g_{j,k}(n) = w(n - jq) \exp\left(\frac{2i\pi kn}{W}\right) \tag{3}$$

where $(w(n))_{1 \leq n \leq W}$ is a real window function [6], $j$ and $k$ are respectively time and frequency indices, $1 \leq j \leq L$, $1 \leq k \leq W$, L is the number of window functions covering the signal support. The integer $q$ characterizes the superposition factor between the windows, generally half a window length. The STFT

$$f \longmapsto S_f(j,k) := \langle f \mid \overline{g_{j,k}} \rangle = \sum_{n=1}^{W} f(n).w(n - jq) \exp\left(\frac{-2i\pi kn}{W}\right) \tag{4}$$

for $1 \leq k \leq W$ and $1 \leq j \leq L$ computes the complex coefficients matrix

$$M_f = (S_f(j,k))_{j,k} \tag{5}$$

using Algorithm 1. The inverse STFT

$$f(n) = \frac{1}{W} \sum_{j=1}^{L} \sum_{k=1}^{W} S_f(j,k). \exp\left(\frac{2i\pi kn}{W}\right) \tag{6}$$

reconstructs the signal $f = (f(n))_{q \leq n \leq Lq}$ using Algorithm 2, provided that

$$\sum_{j=1}^{L} w(n - jq) = 1. \tag{7}$$

Since $f$ is a real signal the matrix of absolute values of its STFT coefficient is symmetric with respect to its middle row. Thus, a signal time-frequency behavior can be visualized with an image of half the matrix of absolute values of the coefficients matrix $M_f$.
Figure 1 shows the spectrogram (absolute value of the coefficients with logarithmic frequency scale) of a Mozart oboe music extract. Figure 2 shows the spectogram of the noised Mozart signal.

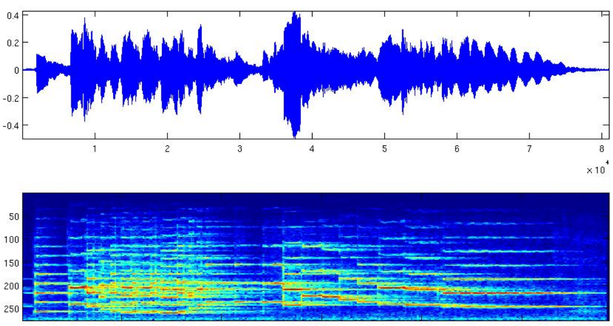


Figure 1: Spectrogram of an oboe music extract

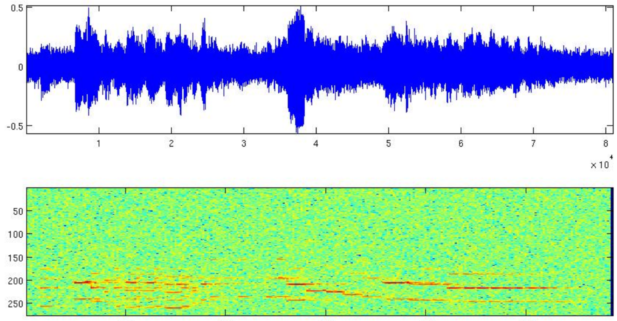


Figure 2: Spectrogram of a noisy oboe music extract

## 2.1 Hanning window function

The Guoshen Yu et al. algorithm computes the STFT using the Hanning window function (Figure 3). The Hanning window function is defined on $]-1,1[$ by the $C^1(\mathbb{R})$ function

$$h: \mathbb{R} \rightarrow [0,1]$$
$$x \mapsto \begin{cases} 0 & \text{if } |x| \geq 1 \\ \dfrac{1+\cos(\pi x)}{2} & \text{if } x \in [-1,1] \end{cases}$$

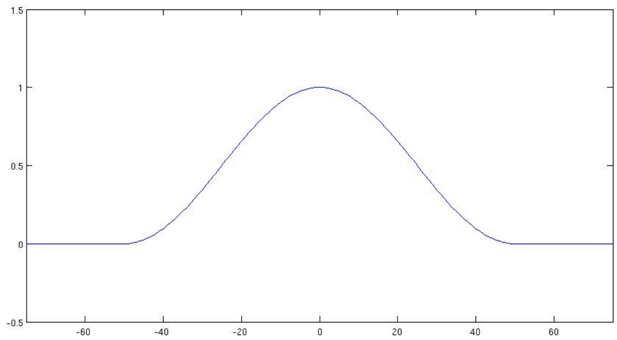


Figure 3: The Hanning window

Figure 4 shows the Hanning window functions with half length window support superposition.

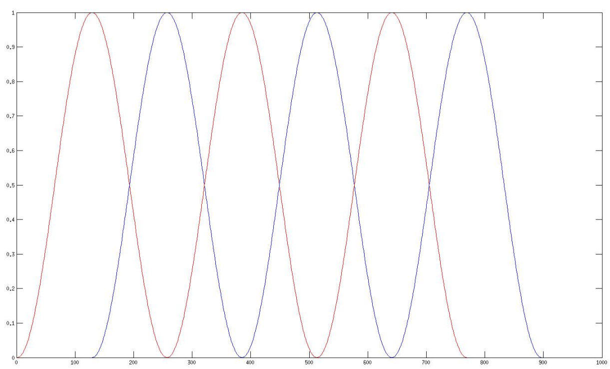

Figure 4: Hanning windows with half window support superposition

The Guoshen Yu et al. algorithm uses the STFT with discrete Hanning windows and odd support length, $W = 2q + 1$,

$$w(n) = \frac{1 + \cos(\frac{\pi n}{q})}{2} \quad \forall n \in [-q, q]. \tag{8}$$

The support of the signal $f = (f_n)_{1 \leq n \leq N} \in \mathbb{R}^N$ can be covered using $L$ windows, each one overlapping the next one on $q$ samples. Notice that $L \simeq \left\lfloor 2\frac{N}{W} \right\rfloor$.

- A first window function is considered as a signal defined on $\mathbb{N}$

$$w_1(n) = w(n - q) = \frac{1 + \cos\left(\frac{\pi}{q}(n - q)\right)}{2}$$

  with support $\{0, 1, ..., 2q\}$.

- This window function shifted by $q$ samples to the right is a signal defined on $\mathbb{N}$ with support $\{q, q + 1, ..., 3q\}$.

- The $j^{th}$ window shifted by $jq$ samples to the right

$$w_j(n) = w(n - jq) = \frac{1 + \cos\left(\frac{\pi}{q}(n - jq)\right)}{2} \tag{9}$$

  is a signal defined on $\mathbb{N}$ with support $\{(j-1)q, (j+1)q\}$.

**Proposition 1 (Reconstruction property)**

$$\sum_{j=1}^{L} w_j(n) = 1.$$

*for all $n \in [q, Lq]$ where $L$ is the number of window functions.*

In consequence if $f \in \mathbb{R}^N$,

$$\sum_{j=1}^{L} f(n) w_j(n) = f(n) \sum_{j=1}^{L} w_j(n) = f(n)$$

for $n \in [q + 1, Lq]$. This justifies the given inverse STFT in (6).

# 3 Wiener Filter

The ideal Wiener filter [6] [9] is an optimal time invariant linear denoising filter. It assumes that the underlying clean signal Fourier transform modulus is known. Let $f$ be an audio signal contaminated by a Gaussian white noise

$$y = f + \eta \in \mathbb{R}^N$$

with $\mathbb{E}[\eta] = 0$ and $\mathbb{E}[\eta^2] = \sigma^2$.

**Algorithm 1:** STFT using window functions with odd support length

**Input**:

- $f$: 1D signal of known length,
- $time_{win}$: window size in time (in ms),
- $f_{sampling}$: signal sampling frequency (in Hz).

Window length in number of samples: $size_{win} \leftarrow round(time_{win} * f_{sampling}/1000)$.
**if** $size_{win}$ *is even* **then**
    $size_{win} \leftarrow size_{win} + 1$
**end**
$halfsize_{win} \leftarrow (size_{win} - 1)/2$.
Make a Hanning window $w_{hanning}$ of size $size_{win}$ using (8).
Number of needed windows: $Nb_{win} \leftarrow floor(length(f) * 2/size_{win})$.
Initialize the coefficient matrix $STFT_{coef}$: $STFT_{coef} \leftarrow zeros(size_{win}, Nb_{win} - 2))$.
**foreach** $j$ *from* $1$ *to* $Nb_{win} - 2$ **do**
    Keep the restriction of $f$ on the $w_j$ (9) support: $f_j \leftarrow f$.
    Compute the windowed function $fw_j = f_j \; w_{hanning}$.
    Compute the Fourier transform of this windowed function $fw_j$.
    Store this $size_{win}$ vector in the $j^{th}$ column of the $STFT_{coef}$ matrix.
**end**
**Output**: $STFT_{coef}$.

**Algorithm 2:** Inverse STFT using window functions with odd support length

**Input**: :

- $STFT_{coef}$: Fourier coefficient matrix,
- $f_{sampling}$: signal sampling frequency in Hz,
- $length\_f$: reconstructed signal length.

Window length: $size_{win} \leftarrow$ number of lines of the $STFT_{coef}$ matrix.
$Nb \leftarrow$ number of columns of the $STFT_{coef}$ matrix.
**if** $size_{win}$ *is even* **then**
    $size_{win} \leftarrow size_{win} + 1$
**end**
$halfsize_{win} \leftarrow (size_{win} - 1)/2$.
Initialize $f_{rec}$ : $f_{rec} \leftarrow zeros(length\_f)$.
**foreach** $j$ *from 1 to* $Nb$ **do**
    Compute the inverse Fourier transform $(ift)$ of the $j^{th}$ column of $STFT_{coef}$: $f_{win_{rec}} \leftarrow ift(STFT_{coef}(:, j))$.
    Add the result to $f_{rec}$ in the right spot:
    $u = (j - 1)halfsize_{win}$
    $f_{rec}\,(u + 1 : u + size_{win}) \longleftarrow f_{rec}\,(u + 1 : u + size_{win}) + f_{win_{rec}}$.
**end**
**Output**: $f_{rec}$ reconstructed signal, array of length $length\_f$.

Let $(\hat{y}_k)_{1\le k\le N}$ be the discrete Fourier transform of $y = y(n)_{1\le n\le N}$ and $(\tilde{f}_n)_{1\le n\le N}$ the signal whose discrete Fourier transform is $(\hat{\tilde{f}}_k)_{1\le k\le N}$ with $\hat{\tilde{f}}_k = a_k\hat{y}_k$.
The Wiener filter finds the best attenuation factors $a_k$ by minimizing the mean square error

$$\begin{aligned}\text{MSE} &= \mathbb{E}\left[\sum_k |\hat{\tilde{f}}_k - \hat{f}_k|^2\right] \\ &= \mathbb{E}\left[\sum_k |a_k(\hat{f}_k + \hat{\eta}_k) - \hat{f}_k|^2\right] \\ &= \sum_k |(a_k-1)\hat{f}_k|^2 + |a_k|^2\sigma^2.\end{aligned}$$

The differentiation with respect to $a_k$ gives

$$\frac{\partial \text{MSE}}{\partial a_k} = 2(a_k-1)|\hat{f}_k|^2 + 2a_k\sigma^2,$$

which yields

$$a_k = \left(\frac{|\hat{f}_k|^2}{\sigma^2 + |\hat{f}_k|^2}\right).$$

The Wiener filter will be used by Guoshen Yu et al. to smooth the thresholded noisy signal coefficients before computing the denoised signal with the inverse STFT. Since the Fourier transform $\hat{f}_k$ of the signal is unknown, the empirical Wiener filter uses the equality

$$\mathbb{E}|\hat{y}_k|^2 = \sigma^2 + |\hat{f}_k|^2$$

to estimate $|\hat{f}_k|^2$ as $|\hat{y}_k|^2 - \sigma^2$. This gives the attenuation factor

$$a'_k = \left(\frac{|\hat{y}_k|^2 - \sigma^2}{|\hat{y}_k|^2}\right)_+.$$

The Wiener denoising step therefore reads

$$(f+\eta) \overset{STFT}{\longmapsto} S_{f+\eta}(j,k) = c_{j,k} \overset{Wiener}{\longmapsto} \tilde{c}_{j,k} = a'_{j,k}c_{j,k} \overset{STFT^{-1}}{\longmapsto} \tilde{f} \tag{10}$$

where

$$a'_{j,k} = \left(\frac{|c_{j,k}|^2 - \sigma^2}{|c_{j,k}|^2}\right)_+. \tag{11}$$

This empirical Wiener filter is a diagonal estimator which attenuates each coefficient separately, it causes signal distortion known under the name of musical noise.

**Variance of a windowed noise signal**

The variance of $w(n)\eta(n)$ is $w(n)^2\sigma^2$ for each $n$ between 1 and $W$, thus their mean value is

$$\sigma^2\frac{1}{W}\sum_n w(n)^2 = \sigma^2\frac{1}{W}\sum_n \frac{1}{2}(1+\cos(\frac{2\pi n}{W}))^2 = 0.375\ \sigma^2 \tag{12}$$

# 4 Block thresholding

The thresholding signal denoising algorithm computes the STFT or other time-frequency transform of the noisy signal and processes the matrix of coefficients to attenuate the noise. The STFT is the transform mostly used to process audio signals.
Let us denote

$$y(n) = f(n) + \eta(n)$$

the audio noisy signal where $\eta$ is a zero-mean Gaussian process independent of $f$ and $1 \leq n \leq N$. The block thresholding processes the STFT coefficient matrix

$$C = \{c_{j,k} = S_{f+\eta}(j,k) \;\; 1 \leq j \leq L \;\; 1 \leq k \leq W \;\}$$

associated to the time-frequency matrix

$$X = \{(j,k) : \;\; 1 \leq j \leq L \;\; 1 \leq k \leq W \;\}$$

where $j$ and $k$ are time and frequency indices ($L$ is the number of windows used to compute the STFT transform). This time-frequency matrix is first segmented into rectangular macroblocks of size 8 x 16 (8 samples in time and 16 samples in frequency). Each macroblock is partitioned into rectangular blocks of the same size, $L_i$ x $W_i$ with time length $L_i = 8, 4, 2$ and wide frequencies $W_i = 16, 8, 4, 2, 1$. For each macroblock, these 15 partitions into blocks of sizes

$$M_b = \begin{pmatrix} 8\times 16 & 8\times 8 & 8\times 4 & 8\times 2 & 8\times 1 \\ 4\times 16 & 4\times 8 & 4\times 4 & 4\times 2 & 4\times 1 \\ 2\times 16 & 2\times 8 & 2\times 4 & 2\times 2 & 2\times 1 \end{pmatrix} \tag{13}$$

were compared using a Stein Unbiased Risk Estimator (SURE) to select the best. SURE is described in the next paragraph.

## 4.1 Attenuation factor

If $(B_i)_{i\in I}$ is a partition of the time-frequency domain $X$, the reconstruction formula (6) can be rewritten by a mere rearrangement of the summation

$$f(n) = \frac{1}{W} \sum_{i=1}^{I} \sum_{(j,k)\in B_i} S_f(j,k) \exp\left(\frac{2i\pi kn}{W}\right). \tag{14}$$

The coefficients of each block $B_i$ are attenuated, $\tilde{c}_{j,k} = a_i c_{j,k} \quad \forall (j,k) \in B_i$, by a common factor

$$a_i = \left(1 - \frac{\lambda \sigma^2 B_i^{\#}}{||Y_i||^2}\right)_+ \tag{15}$$

where

$$||Y_i||^2 = \sum_{(j,k)\in B_i} |c_{j,k}|^2 \tag{16}$$

and $B_i^{\#}$ the number of coefficients in $B_i$. The value of $\lambda$ is specified in the next paragraph. The restored signal with these attenuated coefficients is

$$\tilde{f}(n) = \frac{1}{W} \sum_{i=1}^{I} \sum_{(j,k)\in B_i} a_i c_{j,k} \exp\left(\frac{2i\pi kn}{W}\right). \tag{17}$$

**Computation of $\lambda$**

If the STFT coefficients of the noise

$$\epsilon_{j,k} =< \eta \mid g_{j,k} >$$

are computed with half overlapping Hanning window functions, the average noise energy follows a $\chi^2$ distribution with $B_i^{\#}$ degrees of freedom. The parameter $\lambda$ is computed from $B_i^{\#}$ by adjusting the residual noise probability

$$P(\bar{\epsilon}^2 > \lambda\sigma^2) < \delta$$

The "musical noise" becomes barely audible with $\delta \simeq 0.1$. Table I shows the corresponding $\lambda$ values for various block sizes $B_i^{\#}$:

| $B_i^{\#}$ | 4 | 8 | 16 | 32 | 64 | 128 |
|---|---|---|---|---|---|---|
| $\lambda$ | 4.7 | 3.5 | 2.5 | 2.0 | 1.8 | 1.5 |

Table 1

Threshold $\lambda$ calculated for various block sizes $B^{\#}$ with $\delta = 0.1\%$

Guoshen Yu et al. compute $\lambda$ for the 15 block sizes (13) using table 1. For $W_i = 1$, the values of $\lambda$ are choosen equal to those for $W_i = 2$. In consequence the $\lambda$ values corresponding to the 15 block sizes (13) are written in the following matrix

$$M_\lambda = \begin{pmatrix} 1.5 & 1.8 & 2 & 2.5 & 2.5 \\ 1.8 & 2 & 2.5 & 3.5 & 3.5 \\ 2 & 2.5 & 3.5 & 4.7 & 4.7 \end{pmatrix}. \tag{18}$$

Neither the risk of the method, defined by $R = E\{||f - \tilde{f}||^2\}$ nor its upper bound

$$R = E\{||f - \tilde{f}||^2\} \leq K \sum_{i=1}^{I} \sum_{(j,k)\in B_i} \mathbb{E}\{|S_f(j,k) - a_i c_{j,k}|^2\} \tag{19}$$

can be computed since $f$ and $S_f(j,k)$ are unknown. This upper bound is obtained using (14) and (17).

**SURE theorem**

To estimate this block thresholding risk, T. Cai [1] uses the *Stein Unbiased Risk Estimator (SURE)*: [8]

**Theorem 1** *Let $Y = (Y_1, Y_2, ..., Y_p)$ be a Gaussian vector with identity as covariance matrix and mean $F = (F_1, F_2, ..., F_p)$. Let $Y + h(Y)$ be an estimator of $F$ where $h = (h_1, h_2, ..., h_p)$ almost differentiable*(1) *(each $h_j : \mathbb{R}^p \to \mathbb{R}^1$ is almost differentiable) and*

$$\nabla.h = \sum_{j=1}^{p} \frac{\partial h_j}{\partial Y_j}.$$

*If $\mathbb{E}[\sum_{j=1}^{p} |\partial h_j(Y)/\partial Y_j|] < \infty$, then*

$$R = \mathbb{E}[||Y + h(Y) - F||^2] = p + \mathbb{E}[||h(Y)||^2 + 2\nabla.h(Y)]$$

*and*

$$\tilde{R} = p + ||h(Y)||_2^2 + 2\nabla.h(Y)$$

*is an unbiased estimator of the risk $R$ of $Y + h(Y)$. It is called the Stein Unbiased Risk Estimator (SURE).*

To apply this theorem to our problem, we set $y = f + \eta$, $Y = S_f + \epsilon$, $\mathbb{E}(\epsilon) = 0$ and call $Y$ (resp. $S_f$, $\epsilon$) the STFT coefficient matrix of $y$ (resp. $f$, $\eta$) so that

$$\mathbb{E}(Y) = S_f.$$

**Best macroblock partition**

For each macroblock partition $\{B_i \quad i \in I\}$

$$\mathbb{E}(Y_i) = F_i$$

where

$$Y_i = \{c_{j,k} = S_{f+\eta}(j,k) : \quad (j,k) \in B_i\}$$

and

$$F_i = \{S_f(j,k) : \quad (j,k) \in B_i\}.$$

Moreover, from (15) and (16), the attenuation factor can be written as follows

$$a_i = \left( \frac{\overline{Y_i^2} - \lambda\sigma^2}{\overline{Y_i^2}} \right)_+ \tag{20}$$

therefore

$$h(Y_i) = a_i Y_i - Y_i = -Y_i \left( \frac{\lambda\sigma^2}{\overline{Y_i^2}} 1_{\overline{Y_i^2} \geq \lambda\sigma^2} + 1_{\overline{Y_i^2} < \lambda\sigma^2} \right)$$

with $h : \mathbb{C}^{B_i^\#} \to \mathbb{C}^{B_i^\#}$ and

$$h(X) := \begin{cases} \dfrac{-X\lambda\sigma^2 B_i^\#}{||X||^2} & \text{if} \quad ||X||^2 > \lambda\sigma^2 B_i^\# \\ \qquad -X & \text{if} \quad ||X||^2 \leq \lambda\sigma^2 B_i^\# \end{cases} \tag{21}$$

which is continuous in $B_i$, differentiable in $A_i$ for

$$A_i = \{c_{j,k} = S_{f+\eta}(j,k) \in B_i : \quad \sum ||c_{j,k}||^2 \neq \lambda\sigma^2 B_i^\#\}$$

then Lipschitz over $A_i$. In consequence $h$ is almost differentiable.

Applying the SURE theorem to

- $p = B_i^\#$
- $h(Y_i) = (a_i - 1)Y_i$
- $a_i$ given by (20)

the formula

$$\tilde{R}_i = \sigma^2 \left( B_i^\# + \mathbb{E}\left\{ \left|\left| h\left(\frac{Y_i}{\sigma_i}\right) \right|\right|^2 + 2\nabla.h\left(\frac{Y_i}{\sigma_i}\right) \right\} \right) \tag{22}$$

gives an estimator of the $i^{th}$-block risk:

$$\begin{aligned} R_i &= \sum_{(i,j)\in B_i} \mathbb{E}[|F_{i,j} - a_i Y_i|^2] \\ &= \sum_{(i,j)\in B_i} \mathbb{E}[|F_{i,j} - Y_i - h(Y_i)|^2]. \end{aligned}$$

Therefore [9]

$$\tilde{R}_i = \sigma^2 \left( B_i^\# + \frac{\lambda^2 B_i^\# - 2\lambda(B_i^\# - 2)}{\frac{\overline{Y_i^2}}{\sigma^2}} \mathbb{1}_{\overline{Y_i^2} \geqslant \lambda\sigma^2} + B_i^\# \left( \frac{\overline{Y_i^2}}{\sigma^2} - 2 \right) \mathbb{1}_{\overline{Y_i^2} < \lambda\sigma^2} \right). \tag{23}$$

The risk of a macroblock partition is the sum of the estimated block risks, $\sum_i \hat{R}_i$. Each macroblock is treated independently, the risk values of the 15 different macroblock partitions are computed and the one with minimal risk is kept. This is done using Algorithms 3, 4, 5, 6 and 7.

**Algorithm 3:** Block Thresholding

**Input**: $f$: the noisy signal, $time_{win}$: window length in time (in ms), $f_{sampling}$: signal sampling frequency (in Hz), $\sigma_{noise}$: the noise variance.

Window size in number of samples: $size_{win} \leftarrow round(time_{win} * f_{sampling}/1000)$.
**if** $size_{win}$ *is even* **then**
    $size_{win} \leftarrow size_{win} + 1$
**end**
$halfsize_{win} \leftarrow (size_{win} - 1)/2$
Number of needed windows: $Nb_{win} \leftarrow floor(length(f) * 2/size_{win})$.
Number of macroblock columns: $nb\_Macroblk\_columns$.
Number of macroblock lines: $nb\_Macroblk\_lines$.
Half number of macroblock lines: $half\_nb\_Macroblk\_lines$.
Compute matrix $STFT_{coef}$ of size $size_{win} \times (Nb_{win} - 2)$ using Algorithm 1.
Initialize $STFT_{coef_{th}}$ to a null matrix of size $size_{win} \times Nb_{win}$
$\sigma_{hanning} \leftarrow size_{win}\ \sigma_{noise}\ \sqrt{0.375}$ using (1) and (12).

For negative frequencies (the $halfsize_{win} + 1$ first lines of $STFT_{coef}$):
**foreach** $j$ *from* 1 *to* $nb\_Macroblk\_columns$ **do**
    For the first line of $STFT_{coef}$ (which values are real) compute the attenuation factors using equation (15) and update the corresponding $STFT_{coef_{th}}$ coefficients.
    **foreach** $i$ *from* 2 *to* $half\_nb\_Macroblk\_lines$ **do**
        Compute the macroblock $i, j$ of $STFT_{coef_{th}}$ calling Algorithm 4 with input macroblock $i, j$ of $STFT_{coef}, \sigma_{hanning}$.
    **end**
    For the last few frequencies outside the set of macroblocks, compute the attenuation factors using equation (15) and update the corresponding $STFT_{coef_{th}}$ coefficients.
**end**

For positive frequencies, conjugate the computed last $halfsize_{win}$ lines of $STFT_{coef_{th}}$.
Apply Wiener Filter: $STFT_{coef_{id}} \leftarrow Wiener(STFT_{coef_{th}})$.
Invert the result: $f_{rec} \leftarrow inverseSTFT(STFT_{coef_{id}}, time_{win}, f_{sampling}, length(f))$ using Algorithm 2.

**Output**: $f_{rec}$.

**Algorithm 4:** Best partition on a macroblock using a SURE matrix

**Input**: macroblock of $STFT_{coef}$, $\sigma_{noise}$.

Initialize $SURE$ to a null matrix of size $3 \times 5$ (number of possible macroblock partitions).
**foreach** *Possible macroblock partition* **do**
    $\lambda \leftarrow M_\lambda$ (18) value corresponding to the current partition.
    Compute the risk of the current macroblock partition using Algorithm 5 with input: $STFT_{coef}$, partition, $\lambda$, $\sigma$.
    Store the result in the $SURE$ matrix coefficient corresponding to the current partition.
**end**
Find the Minimum of $SURE$ and the matched partition.
Compute the $STFT_{coef_{th}}$ corresponding macroblock by calling Algorithm 6 with input: macroblock of $STFT_{coef}$, best partition, corresponding $\lambda$ value, $\sigma$.

**Output**: macroblock of $STFT_{coef_{th}}$.

**Algorithm 5:** Risk value of the macroblock partition

**Input**: macroblock of $STFT_{coef}$, macroblock partition, $\lambda$, $\sigma$.

```
Initialize the variable Risk to 0.
foreach Mini-block do
    Compute its risk value calling Algorithm 7.
    Add this value to Risk.
end
```

**Output**: Risk.

**Algorithm 6:** Attenuation factor

**Input**: (macroblock of $STFT_{coef}$, macroblock partition, $\lambda$, $\sigma$).

```
Initialize the corresponding macroblock of STFT_{coef_th} to zero.
foreach Block in the partition of the STFT_{coef} macroblock do
    Compute the attenuation factor, using the equation (20).
    Multiply all the STFT coefficients of the current block by this attenuation factor to
    compute the corresponding macroblock of STFT_{coef_th}.
end
```

**Output**: macroblock of $STFT_{coef_{th}}$.

**Algorithm 7:** Risk value on a block of coefficients

**Input**: (block $B$ of $STFT_{coef}$, $\lambda$, $\sigma$).

```
Compute the block risk using equation (23).
```

**Output**: Risk.

# 5 Experiments and Results

The described block thresholding procedure

$$(f+\eta) \overset{STFT}{\longmapsto} c_{j,k} = S_{f+\eta}(j,k) \overset{Thresholding}{\longmapsto} \tilde{c}_{j,k} = a_i c_{j,k} \overset{Wiener}{\longmapsto} \tilde{\tilde{c}}_{j,k} \overset{STFT^{-1}}{\longmapsto} \tilde{f}$$

performs a strong noise reduction without preserving the whole time-frequency spectrum of the underlying signal $f$. In consequence, sound damages are often audible. This happens when the attenuation factor over a block $B_i$ is zero ($a_i = 0$) and the STFT coefficients of the signal $f$ are different from zero over $B_i$ ($S_f(j,k) \neq 0$ for $(j,k) \in B_i$).

This loss of spectrum coefficients is clearly visible in figure 5 which shows $S_{f+\eta}$, $S_f$ et $S_{\tilde{f}}$ for a trumpet audio signal. This figure shows that for high frequencies the spectrum of the noisy signal masks the one of the underlying clean signal and illustrates a high frequency distortion of the denoising algorithm, which is also audible.

Figure 6 et 7 compare $S_{f+\eta}$, $S_f$ et $S_{\tilde{f}}$, for a fixed time value $j = j_0$. They show that for high frequencies, the noisy signal masks the original spectrum and the denoised signal is set to zero.

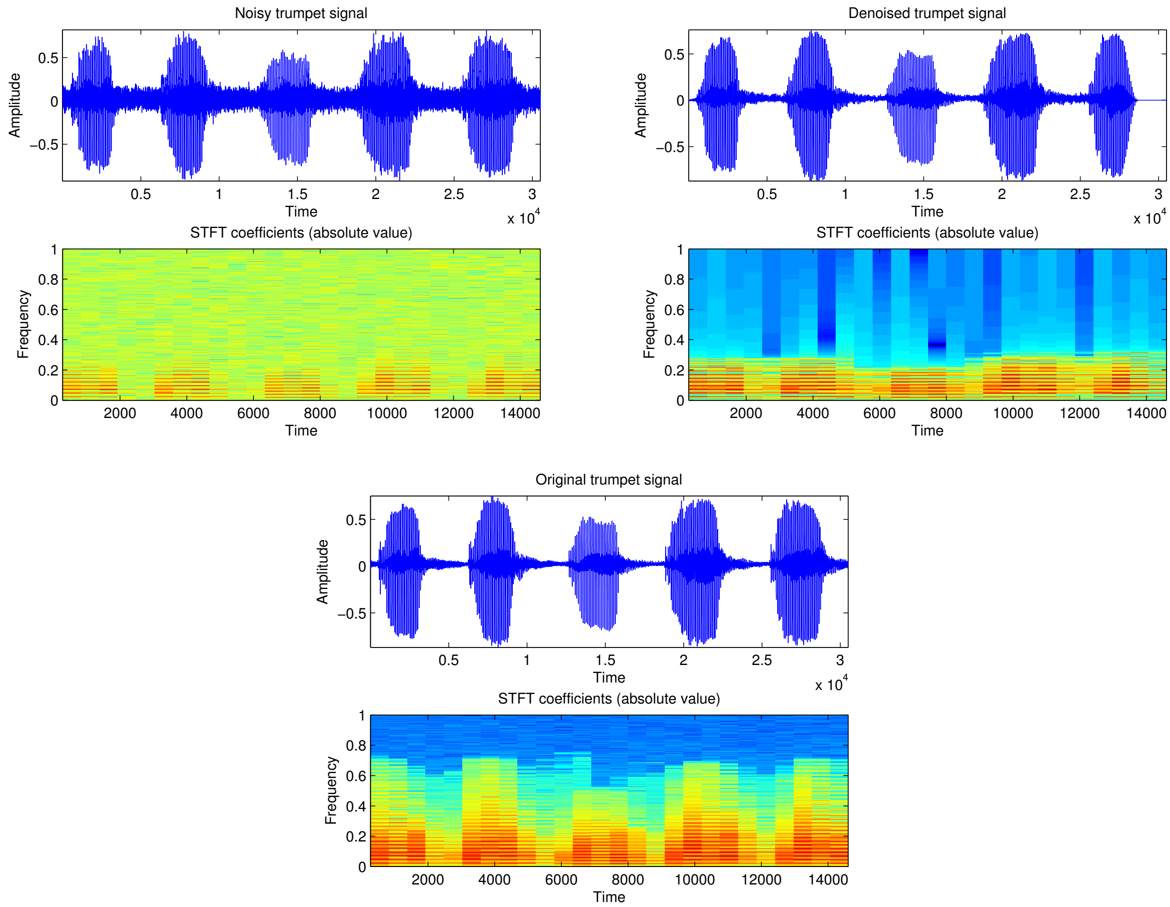


Figure 5: Noisy, denoised and original trumpet signal with spectrograms

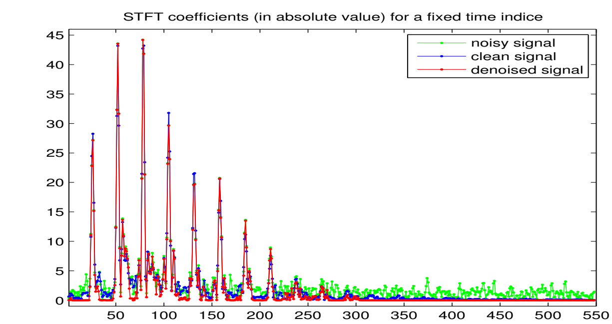


Figure 6: Noisy, denoised and clean audio trumpet signal

Fig. 7 superposes $f + \eta$, $\tilde{f}$ and $f$ for comparison.

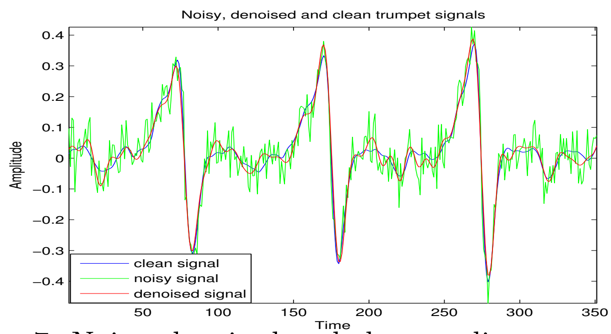


Figure 7: Noisy, denoised and clean audio trumpet signal

# 6 Conclusion

This paper describes and reviews performances of the Guoshen Yu et al. block audio denoising algorithm. This algorithm based on a non-diagonal time-frequency processing attenuates "musical noise" artifacts. It denoises audio signals corrupted with additive Gaussian white noise.
A loss of the original signal spectrum after denoising is clearly visible. In consequence, in spite of the improved audition, some damage is often audible. Inevitably, the sound of the denoised signal is often damaged compared with the sound of the orginal signal since a part of the underlying signal spectrum is removed by this procedure.

The ANSI C implementation (available on line) improves highly the algorithm running time compared to the Matlab implementation.

# References


[1] Y. Cai, H. Zhou *A data-driving block thresholding approach to wavelet estimation*, Annals of Statistics, vol. 37, pp. 569-595, 2009. DOI: 10.1214/07-AOS538.

[2] O. Cappé, *Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor*, IEEE Transactions on Speech and Audio Processing, vol. 2, pp. 345-349, 1994. DOI: 10.1109/89.279283.

[3] D. Donoho and I. Johnstone, *Ideal spatial adaptation via wavelet shrinkage*, Biometrika, vol. 81, pp. 425-455, 1994. doi: 10.1093/biomet/81.3.425

[4] Y. Ephraim and D. Malah, *Speech enhancement using a minimum mean square error short-time spectral amplitude estimator*, IEEE Transactions on Acoustics Speech and Signal Processing, vol. 32, no. 6, pp. 1109-1121, Dec. 1984. DOI: 10.1109/TASSP.1984.1164453.

[5] Y. Ephraim and D. Malah, *Speech enhancement using a minimum mean square error log-spectral amplitude estimator*, IEEE Transactions on Acoustics Speech and Signal Processing, vol. ASSP-33, no. 2, pp. 443-445, Apr. 1985. DOI: 10.1109/TASSP.1985.1164550.

[6] Stéphane Mallat, *a Wavelet tour of signal processing - The Sparse Way*, $3^{rd}$ edition, December 2009. ISBN:0123743702 9780123743701.

[7] R. J. McAulay and M. L. Malpass, *Speech enhancement using soft decisions noise suppression filter*, IEEE Transactions on Acoustics Speech and Signal Processing, ASSP-28, pp. 137-145, 1980.

[8] Charles M. Stein, *Estimation of the Mean of a Multivariate normal distribution*, Annals of Statistics, Vol. 9, No.6, 1135-1151, 1981. DOI:10.1214/aos/1176345632.

[9] Guoshen Yu, Stéphane Mallat and Emmanuel Bacry, *Audio Denoising by Time-Frequency Block Thresholding*, IEEE Transactions on Acoustics Speech and Signal Processing, Vol. 56, No. 5, May 2008. DOI: 10.1109/TSP.2007.912893.