Image Denoising and the SURE-LET Methodology

Thierry Blu¹ and Florian Luisier²

¹Department of Electronic Engineering
The Chinese University of Hong Kong

²Biomedical Imaging Group
École Polytechnique Fédérale de Lausanne
Statistics and Information Sciences Laboratory
Harvard University, Cambridge, MA

September 2010

Outline

1 Image Denoising Methods
   - An Abundant Literature
   - Statistical Approaches
   - Regularization Approaches

2 The SURE-LET Methodology
   - Stein’s Unbiased Risk Estimate (SURE)
   - Linear Expansion of Thresholds (LET)
   - The SURE-LET Optimization
   - Computational Issues

3 SURE-LET Algorithmics
   - Transform domain denoising
   - Orthogonal Representations/Transformations
   - Redundant Representations/Transformations
   - Noise Variance Estimation

4 Algorithm Comparisons
   - Grayscale Image Denoising
   - Color Image Denoising
   - Video Denoising

5 Extension to Poisson-Gaussian Denoising
   - Poisson-Gaussian MSE Estimate
   - Interscale Haar-Wavelet Algorithm
   - Redundant Algorithm
   - Some Comparisons
   - Fluorescence Microscopy Results

Noise in Images: Noise Sources

Noise: a random, undesirable, and often unavoidable perturbation.

Two main sources:
- Random nature of photon emission and detection;
- Imperfection of the electronic devices (photosensors, A/D converter,...).

Tremendous impact on image visualization and analysis (segmentation, tracking, recognition,...).
**Prior-Based Statistical Approaches**

In the prior-based statistical approaches the signal to restore is considered as the realization of a random variable.

Various possible objectives to optimize:
- Maximum a posteriori (MAP)
- Minimum mean-squared error (MMSE)

All these methods assume that the following are explicitly given:
- The statistical relation (likelihood) between the measurements and the signal to restore:
  \[ P(y|x) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{|y-x|^2}{2\sigma^2}\right) \]
- The probability density function (pdf) of the original signal \( P(x) \).

Highly sensitive to the modeling of the pdf of the signal to restore.
Nonlinear MMSE: Wiener

The Wiener “filter” consists in finding the linear estimate, \( \hat{x} = Ay \), that minimizes the Mean-Squared Error (MSE)

\[
\delta \left( \frac{1}{N} \|Ay - x\|^2 \right) = \min_A \delta \left( \frac{1}{N} \|Ay - x\|^2 \right)
\]

MSE between \( x \) and \( \hat{x} \)

**Solution**: Requires only the knowledge of the covariance matrix \( \Gamma_x = \delta \{xx^T\} \) of the original signal

\[
x = \Gamma_x (\Gamma_x + \sigma^2 I)^{-1} y
\]

**Note**: Although very popular, linear processing is not well-adapted to the processing of transient signals.

2\( \delta \{x\} = 0 \) — an affine estimate is used, otherwise.

Nonlinear MMSE: One Step Further

**Problem**: Find the optimal processing \( F(\cdot) \) that yields the estimate \( \hat{x} = F(y) \) such that

\[
\delta \left( \frac{1}{N} \|F(y) - x\|^2 \right) \text{ is minimized.}
\]

**Solution**: In the case of AWGN, the posterior expectation \( \hat{x} = \delta \{x|y\} \) can be simplified to (Stein 1981, Raphan & Simoncelli 2007):

\[
\hat{x} = y + \sigma^2 \nabla \log \mathcal{P} \{y\}
\]

convolution with a Gaussian

**Note**: Because \( \mathcal{P} \{y\} = \int \mathcal{P} \{y|x\} \cdot \mathcal{P} \{x\} \, d^Nx \), the optimal MSE processing is infinitely differentiable.

The optimal algorithm only requires the knowledge of the pdf of the observed noisy signal ~ No prior information is needed!

Examples

Assuming a Laplace prior, \( \mathcal{P} \{x\} = \prod_{n=1}^{N} \frac{1}{2} e^{-\lambda|x_n|} \), these statistical approaches yield a pointwise thresholding involving \( T = \lambda \sigma^2 \):

**MAP** \( \hat{x}_n = \text{soft}_T(y_n) \)

**Wiener** \( \hat{x}_n = \frac{y_n}{1 + \frac{T}{\sigma^2} e^{-\lambda y_n}} \cdot \text{erfc} \left( \frac{-y_n + T}{\sigma \sqrt{2}} \right) - e^{\lambda y_n} \cdot \text{erfc} \left( \frac{y_n + T}{\sigma \sqrt{2}} \right) \)

**MMSE** \( \hat{x}_n = y_n - T \cdot e^{-\lambda y_n} \cdot \text{erfc} \left( \frac{-y_n + T}{\sigma \sqrt{2}} \right) + e^{\lambda y_n} \cdot \text{erfc} \left( \frac{y_n + T}{\sigma \sqrt{2}} \right) \)
Regularization Approaches

The signal estimate \( \hat{x} \) is selected as the minimizer of a (convex) regularized cost-functional
\[
J(x, y) = \Psi(x, y) + \lambda \Phi(x)
\]
data-fidelity term penalty

Typical choice of data-fidelity term:
\[
\Psi(x, y) = \| y - x \|^2 \propto \text{negative log-likelihood (AWGN)}
\]

Typical choices of penalty:
- Tikhonov (smoothness prior): \( \Phi(x) = \|Lx\|^2 \);
- Sparsity prior: \( \Phi(x) = \|x\|_0 \sim \|x\|_1 \);
- TV (edge prior): \( \Phi(x) = \| \nabla x \|_1 \).

**Note:** Depending on the choice of data-fidelity and penalty terms, \( J(x, y) \) can be re-interpreted as a statistical prior and its optimization equivalent to a MAP.

No explicit distance minimization between original and denoised signal.

A simple proof

On the one hand (remember that \( y = x + b \))
\[
\mathcal{E} \{ \| F(y) - x \|^2 \} = \mathcal{E} \{ \| F(y) \|^2 \} - 2 \mathcal{E} \{ x^T F(y) \} + \| x \|^2
\]

\[
= \mathcal{E} \{ \| F(y) - y \|^2 \} + \mathcal{E} \{ x^T F(y) \} - N \sigma^2
\]

and on the other hand (Stein’s Lemma)
\[
\mathcal{E} \{ b^T F(y) \} = \int \mathcal{E} \{ b \} b^T F(x + b) \, dN(b) \quad \text{(Gaussian pdf)}
\]
\[
= \sigma^2 \mathcal{E} \{ b \} \text{div} \{ F(x + b) \} \quad \text{(by parts)}
\]

Minimizing \( \mathcal{E} \{ \| F(y) - x \|^2 \} \) yields an algorithm \( F(y) = y + \sigma^2 \nabla \log \mathcal{E} \{ y \} \).

**Problem:** we have only one realization of the noisy image \( y \).

**Solution:** estimate \( \mathcal{E} \{ \| F(y) - x \|^2 \} \) from \( y \), instead of \( \mathcal{E} \{ y \} \).

**MSE estimation**

Consider the random variable \( \mathcal{E} \{ \text{SURE}(y) \} \)
\[
\mathcal{E} \{ \text{SURE}(y) \} = \mathcal{E} \{ \| F(y) - x \|^2 \} - N \sigma^2
\]

Under the *additive white Gaussian noise* hypothesis, this random variable is an *unbiased estimate of the MSE* Stein et al. 1981
\[
\mathcal{E} \{ \text{SURE}(y) \} = \mathcal{E} \{ \| F(y) - x \|^2 \} - N \sigma^2
\]

**Divergence operator:** \( \text{div} \{ F(y) \} \) \( \overset{\text{def}}{=} \sum \frac{\partial F(y)}{\partial y} \)

The original signal \( x \) may, or may not be random. _No assumptions on \( x \) are needed._

**Example** Donoho 1995: SURE soft-threshold
\[
\text{SURE}_{\text{soft}} = \frac{1}{N} \left( \sum_{|n| \leq T} y_n^2 + \sum_{|n| > T} T^2 + 2 \sigma^2 \left( \sum_{|n| > T} 1 \right) - \sigma^2 \right)
\]
Closeness between SURE and MSE

Processing a noisy signal (left) with several lengths, using several different pointwise thresholding functions.

**Note:** The use of the SURE (instead of the MSE) is particularly justified for large data sizes (e.g., images).

Approximation of processes

Functions can often be efficiently approximated onto adapted bases.

**Examples of bases:** wavelets (L^2 functions), sinc kernels (bandlimited functions), radial basis functions (scattered points interpolation), etc.

The MMSE result $F(y) = y + \sigma^2 \nabla \log P\{y\}$ indicates that the optimal processing is slowly varying. It can thus, in principle, be represented on a basis of few functions — e.g., the identity and spline/Gaussian functions.

$\text{Optimal MSE} \approx a \times y + b \times \text{sign}(y) \left(1 - e^{-\frac{y^2}{2 \sigma^2}} \right) + c \times ye^{-\frac{y^2}{2 \sigma^2}}$

(see slide 12)

Choosing the LET basis

Based on Wiener theory, homogenous (Gaussian, zero-mean) images are optimally denoised by linear transformations.

By segmenting/partitioning a non-homogenous image into homogenous zones, the "optimal" denoising process can thus be expressed as a sum of linear processes within each zone

$F(y) = \sum_{k} \gamma_k \mathbf{A}_k y$

The indicator function of zone $k$

Hence, the choice of a LET basis essentially amounts to choosing a "good" (MSE-wise) segmentation algorithm.

The linear space approximation will prove particularly useful when combined with a quadratic objective functional (e.g., MSE or SURE), as the optimization boils down to solving a linear system of equations.
Choosing the LET basis

**Example:** A simple threshold tends to segment a signal into large values, and small values. A possible choice\(^1\) for the indicator function of the small values is

\[ \gamma(y) = e^{-\frac{y^2}{2\sigma^2}} \]

Then, a possible LET function is of the form

\[ F(y) = \gamma(y) \times ay + (1 - \gamma(y)) \times by \]

where

- small \( y \) restricts
- large \( y \) depends linearly on the other

The coefficients \( a \) and \( b \) characterize the linear behavior of the processing in each zone.

**Note:** A practical choice for \( T \) is \( \sqrt{3}\sigma \) (noise), which can be related to a significance level in a statistical test.

\(^1\)for a tanh-based threshold, see Pesquet et al. 1997

The SURE minimization

By restricting \( F(y) \) to be of the LET form \( \sum_k a_k F_k(y) \), the SURE becomes a quadratic expression, in function of the \( a_k \)'s. Its minimization yields, for all \( k = 1, 2, \ldots, K \)

\[ \sum_{l=1}^K F_k(y)^T F_l(y) a_l = F_k(y)^T y - \sigma^2 \text{div} \{ F_k(y) \} \]

Finally, by stacking the LET coefficients in \( a = [a_1, a_2, \ldots, a_K]^T \), we get

\[ a = M^{-1} c \]

where

\[ M = [F_k(y)^T F_l(y)]_{1 \leq k, l \leq K} \]
\[ c = [F_k(y)^T y - \sigma^2 \text{div} \{ F_k(y) \}]_{1 \leq k \leq K} \]

**Note:** When \( M \) is non-invertible, it means that one LET basis element depends linearly on the other \( F_k \) decrease the LET-order to \( K = 1 \).

Recapitulation of the SURE-LET approach

1. Instead of finding an approximation of the signal \( x \), find an approximation of the processing \( F(y) \) that transforms \( y \) into \( \hat{x} \);
2. Instead of minimizing the MSE between \( \hat{x} \) and \( x \), minimize an (unbiased) estimate of this MSE, based on \( y \) alone (SURE);
3. Express \( F(y) \) as a linear decomposition (LET) \( \sum_k a_k F_k(y) \) of basis processings \( F_k(y) \sim \) linear system of equations (fast, unique).

**Note:** The number \( K \) of elementary processings is chosen very small (usually, \( K < 200 \)), compared to the number of pixels \( N \).

The Oracle minimization

The same LET optimization, by minimizing the MSE \( ||F(y) - x||^2 \)

instead of the SURE yields, for all \( k = 1, 2, \ldots, K \)

\[ \sum_{l=1}^K F_k(y)^T F_l(y) a_l = F_k(y)^T x \]

This also boils down to solving a linear system of equations

\[ a = M^{-1} c' \]

where

\[ M = [F_k(y)^T F_l(y)]_{1 \leq k, l \leq K} \]
\[ c' = [F_k(y)^T x]_{1 \leq k \leq K} \]

**Note:** The Oracle computation allows to choose elementary LET processings \( F_k \) that are likely to yield more efficient denoising results.
A strategy for evaluating algorithms

How to evaluate the potential of an algorithm, that usually involves a number of non-linear parameters?

- Approximate the resulting algorithm as a LET; i.e., transfer the non-linear degrees of freedom to linear parameters;
- Probe the efficiency of the algorithm through Oracle minimization.

Example: If the algorithm \( F(y; \lambda) \) depends on one non-linear parameter, \( \lambda \), approximate it using two (or more) LETs

\[
F(y; \lambda) = a_1 F(y; \lambda_1) + a_2 F(y; \lambda_2)
\]

where \( \lambda_1, \lambda_2 \) are fixed; \([\lambda_1, \lambda_2]\) is the expected range of values for \( \lambda \).

Monte-Carlo divergence estimation

The computation of the divergence term in the SURE may be impractical when \( N \) is large: a direct application of the formula

\[
div \{ F(y) \} = \sum_{n=1}^{N} \frac{\partial F_n(y)}{\partial y_n}
\]

may prove too much CPU intensive.

An alternative is to use a consequence of Stein’s Lemma

\[
div \{ F(y) \} \approx b_0 F(y + \varepsilon b_0) - F(y)
\]

(law of large numbers)

where \( b_0 \) is a normalized (unit-variance, zero-mean) Gaussian white noise. \( \varepsilon \) is some small value compared to the level of noise; typ., \( \varepsilon = \sigma/100 \).

NOTE: Particularly useful when \( F(y) \) is not obtained explicitly, but through a “black-box” algorithm like TV regularization Ramani et al. 2008.

Linear transformations

In order to exploit their strong local correlations, it is advantageous to represent the pixels in another domain: Discrete Cosine Transform (DCT), Block DCT, Wavelet Transform, etc.

Most generally, a linear transformation maps an image \( y \) onto another image \( w \) through a matrix multiplication \( Dw \). It is assumed that the transformation can be inverted using a matrix \( R \).

Desirable properties (not all of them can be satisfied at once):

- Perfect reconstruction: \( RD = Id \);
- \( D \) yields a sparse/decorrelated image representation;
- Shift, scale, rotation invariance;
- Orthonormality.

Example: undecimated wavelet transforms/BDCT are shift-invariant, but are not orthogonal.

Processing images expressed in a sparse representation considerably increases denoising efficiency.
Simple wavelet thresholding

Choice of an orthonormal wavelet transform can be used (e.g., symlet 8). Then, the processing in subband $j$ is a simple thresholding $w_{j,n} = \theta_j(w_{j,n})$ for each of the coordinates $n = 1, 2, \ldots, N_j$ of $w_j$, and

\[
\text{SURE}_j(w_j) = \frac{1}{N_j} \left( \sum_{n=1}^{N_j} |\theta_j(w_{j,n}) - w_{j,n}|^2 + 2\sigma^2 \theta_j'(w_{j,n}) \right) - \sigma^2
\]

SURE-LET simple threshold

A two-parameter zone-selection function

\[
\theta_j(w) = a_j w + b_j w^{\frac{2}{\sqrt{1 + 2\sigma^2}}}
\]

where $a_j$ and $b_j$ are obtained by minimizing $\text{SURE}_j(w_j)$.

**NOTE:** SureShrink (Donoho 1995) makes the choice $\theta_j(w) = \text{soft}_{T_j}(w)$ and minimizes $\text{SURE}_j(w_j)$ for $T_j$.

$^5$However, any (non-wavelet) orthonormal transform can be used.

**Orthonormality**

A decomposition is orthonormal if $DD' = DD'' = D' = D'' = I_d$. Properties:

- The reconstruction is given by $R = D'W$;
- Preservation of the energies: $\|w\| = \|y\|$ and $\|x - \hat{x}\| = \|w - Dx\|$;
- Statistical independence of the transformed coefficients;

**NOTE:** An orthonormal decomposition is automatically non-redundant.

If $w_j = D_j y$ for $j = 1, 2, \ldots, J$ where $D = [D_1; D_2; \ldots; D_J]$, then the unbiased estimate of $\|x - \hat{x}\|^2$ can be written in the transformed domain

\[
\text{SURE}(y) = \frac{1}{N} \left( \sum_{j=1}^{J} \|\Theta_j(w) - w_j\|^2 + 2\sigma^2 \text{div} \{\Theta_j(w)\} \right) - \sigma^2
\]

where $\Theta = [\Theta_1; \Theta_2; \ldots; \Theta_J]$.

Optimizing the denoising process $F(y)$ is equivalent to denoising separately the denoising processes $\Theta_j$ in the transformed domain.
InterScale wavelet thresholding

**Principle:** separate the parent into large and small coefficients, and within each zone so defined, apply a pointwise thresholding function:

\[
\theta_j(w, p) = e^{-\frac{p^2}{2\sigma^2}} \left( a_jw + b_jw^{-\frac{1}{2}} \right) + (1 - e^{-\frac{p^2}{2\sigma^2}}) \left( a'_jw + b'_jw^{-\frac{1}{2}} \right)
\]

- small parents
- large parents

**Note:** DWT is orthogonal, hence \(w\) and \(p\) are statistically independent

**Problem:** the wavelet coefficients are not exactly aligned from band to band (filtering and downsampling effect). How to obtain a parent aligned exactly with its child?

Parent/child alignment: Group-Delay Compensation

Adequate high-pass filtering of the lowpass \(LL_j\) — which contains the whole parent tree: \(W\) compensates the group-delay difference between the low-pass and the high-pass band.

GDC filter formula:

\[
W(z^2) = (1 + z^{-2})G(z^{-1})G(-z^{-1})
\]

where \(G(z) = \) wavelet filter.
**Overview of the interscale SURE-LET denoising**

![Diagram of denoising process]

**Example of result**

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td></td>
</tr>
<tr>
<td>SureShrink</td>
<td>28.73 dB</td>
</tr>
<tr>
<td>SURE-LET</td>
<td>30.18 dB</td>
</tr>
</tbody>
</table>

Best non-redundant transform-domain algorithm.

**Extension to multichannel denoising**

Direct generalization by replacing:
- **scalar-valued by vector-valued wavelet coefficients;**
- **scalar-valued by matrix-valued LET parameters.**

Assuming \( Q = \text{covariance matrix of the noise, and } \gamma(x) = \exp(-x/12) \)

\[
\theta_j(w, p) = \gamma(p^TQ^{-1}p)\gamma(w^TQ^{-1}w)a_{j,w}^T
\]

- small parents and small coefficients
- large parents and small coefficients
- small parents and large coefficients
- large parents and large coefficients

**NOTE:** Automatically selects the best color space (color images).
Undecimated wavelet denoising

Limitations of non-redundant transformations

- High sensitivity to shifts \( \leadsto \) inconsistent reconstruction of edges
- Low design flexibility \( \leadsto \) poor directional sensitivity

Solution: increase the redundancy

Shifts: Cycle-Spinning Coifman 1995, Undecimated DWT Gao 1995; Rotation: Steerable Pyramid Simonsen 1995, Complex DWT Kingsbury 1998; Edges: Curvelets Candès 2000; etc...

Redundancy vs orthonormality

Although it is still possible to have \( R = D^T \) (tight frame)

- \( RD = \text{Id} \) but \( DR \neq \text{Id} \)
- Energies: \( \|w\| = \|y\| \) (if tight frame) but \( \|\hat{x} - x\| \neq \|\hat{w} - Dx\| \)
- Statistical dependence of the transformed coefficients;
In addition, redundancy brings about a higher computational cost.

Undecimated simple wavelet thresholding

Hard-like\(^6\) thresholding rule

In each wavelet subband \( j \), the noisy coefficients are thresholded using

\[
\theta_j(w) = a_j w + b_j w \left( 1 - e^{-\frac{w^2}{\sigma^2}} \right)
\]

where \( (a_j, b_j) \) change from subband to subband — i.e., two parameters per subband.

The optimal set of parameters \( \{a_j, b_j\} \) is then found by minimizing the global image-domain SURE.

NOTE: Contrary to the nonredundant case, it is not possible to optimize the SURE separately in each subband.

\(^6\)Hard threshold cannot be optimized using SURE, for not being differentiable.

Undecimated pointwise wavelet thresholding

Undecimated discrete symlet 8 transform

Noisy

SureShrink

SURE-LET

PSNR=18 dB

PSNR=28.73 dB

PSNR=31.15 dB

NOTE: Surprisingly, it is the simplest wavelet type (Haar) that works best. Smallest support?
Undecimated pointwise wavelet thresholding

Undecimated discrete Haar wavelet transform

<table>
<thead>
<tr>
<th>Noisy</th>
<th>SureShrink</th>
<th>SURE-LET</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=18 dB</td>
<td>PSNR=28.73 dB</td>
<td>PSNR=31.91 dB</td>
</tr>
</tbody>
</table>

**Note:** Surprisingly, it is the simplest wavelet type (Haar) that works best. Shortest support?

**Extensions**

- **Multivariate** wavelet thresholding: taking into account both interscale and **local** wavelet dependencies;
- Thresholding (possibly multivariate) in a **dictionary** of transforms;
- **Multiframe** video denoising: involving motion compensation;

Undecimated discrete Haar wavelet transform

<table>
<thead>
<tr>
<th>Noisy</th>
<th>SURE-LET</th>
<th>SURE-LET multivariate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=18 dB</td>
<td>PSNR=31.91 dB</td>
<td>PSNR=32.22 dB</td>
</tr>
</tbody>
</table>

**Extensions**

- **Multivariate** wavelet thresholding: taking into account both interscale and **local** wavelet dependencies;
- Thresholding (possibly multivariate) in a **dictionary** of transforms.
- **Multiframe** video denoising: involving motion compensation;

**Dictionary of two transforms (UWT Haar & 12 × 12-BDCT)**

<table>
<thead>
<tr>
<th>Noisy</th>
<th>SURE-LET UWT Haar</th>
<th>SURE-LET Multivariate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=18 dB</td>
<td>PSNR=25.90 dB</td>
<td>PSNR=28.80 dB</td>
</tr>
</tbody>
</table>
Extensions

- **Multivariate** wavelet thresholding: taking into account both interscale and local wavelet dependencies;
- Thresholding (possibly multivariate) in a dictionary of transforms.
- Multiframe video denoising: involving motion compensation;

Orthogonal discrete symlet 8 transform

<table>
<thead>
<tr>
<th>Noisy</th>
<th>Multiframe SURE-LET</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=22.11 dB</td>
<td>PSNR=30.85 dB</td>
</tr>
</tbody>
</table>

Noise Variance Estimation

Overview of the Proposed Approach

Noisy Input: $\sigma = 10$

# $h_{opt} = \arg \min_h \| h * y \|_2$ subject to $\| h \|_2 = 1$

$\sim$ Eigenvector corresponding to the smallest eigenvalue of the autocorrelation matrix $\Gamma_y = \sum_{n=1}^{N} y_{n-i}y_{n-j}$ $1 \leq i,j \leq M$

1. Noise variance robustly estimated from the filtered residual ($h_{opt} * y$), as the mode of the smoothed histogram of the local noise variances computed inside blocks of given size (typically, $25 \times 25$).

Performance of the Proposed Approach

<table>
<thead>
<tr>
<th>Cameraman</th>
<th>Mandrill</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\sigma} = 1.4826 \text{ med } {</td>
<td>y - \text{med }{y}</td>
</tr>
</tbody>
</table>

- Simple and accurate for relatively high levels of noise;
- Inaccurate for moderate to low levels of noise.

Proposed approach: **Eigenfilter-based design** Vaidyanathan et al. 1987

<table>
<thead>
<tr>
<th>1.</th>
<th>Find $h_{opt} = \arg \min_h | h * y |_2$ subject to $| h |_2 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>Noise variance robustly estimated from the filtered residual ($h_{opt} * y$), as the mode of the smoothed histogram of the local noise variances computed inside blocks of given size (typically, $25 \times 25$).</td>
</tr>
</tbody>
</table>

Thierry Blu and Florian Luisier Image Denoising and the SURE-LET Methodology 46 / 80
Protocol for Fair Comparisons

- Denoising of a representative set of standard grayscale/color images and video sequences, corrupted by simulated AWGN at 8 different powers $\sigma \in [5, 10, 15, 20, 25, 30, 50, 100]$ (assumed to be known).
- PSNR results averaged over 10 different noise realizations for each noise standard deviation.
- Parameters of each method set according to the values given in the corresponding referred papers or optimized in the MMSE sense if not explicitly provided.

The Non-Redundant Case: PSNR Comparisons

| Image | Original | Noisy | Average SSIM
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena 512 × 512</td>
<td></td>
<td></td>
<td><strong>1.000</strong></td>
</tr>
<tr>
<td>Barbara 512 × 512</td>
<td></td>
<td></td>
<td><strong>0.284</strong></td>
</tr>
</tbody>
</table>

1Structural Similarity Index Map Wang, Bovik, Sheikh & Simoncelli 2004
The Non-Redundant Case: Visual Comparisons

Original

Multivariate SURE-LET

Average SSIM: 1.000
Average SSIM: 0.894

1Structural Similarity Index Map Wang, Bovik, Sheikh & Simoncelli 2004

BiShrink

Multivariate SURE-LET

Average SSIM: 0.877
Average SSIM: 0.894

1Structural Similarity Index Map Wang, Bovik, Sheikh & Simoncelli 2004

ProbShrink

Multivariate SURE-LET

Average SSIM: 0.882
Average SSIM: 0.894

1Structural Similarity Index Map Wang, Bovik, Sheikh & Simoncelli 2004

BLS-GSM

Multivariate SURE-LET

Average SSIM: 0.888
Average SSIM: 0.894

1Structural Similarity Index Map Wang, Bovik, Sheikh & Simoncelli 2004
The Redundant Case: PSNR Comparisons

Peppers 256 × 256
-1.0 0 0.5
Input PSNR [dB] Relative Output Gain [dB]

Coco 256 × 256
-1.0 0 0.5
Input PSNR [dB] Relative Output Gain [dB]

Multivariate SURE-LET (baseline)  
NLmeans Buades et al. 2005  
BLS-GSM Portilla et al. 2003  

BM3D Dabov et al. 2007  
Fast TV Chambolle 2004  
K-SVD Elad & Aharony 2006  

The Redundant Case: Visual Comparisons

Original  
Average SSIM: 1.000

Noisy  
Average SSIM: 0.263

The Redundant Case: Visual Comparisons

Original Multivariate SURE-LET  
Average SSIM: 1.000  

Noisy  
Average SSIM: 0.739
The Redundant Case: Visual Comparisons

- **NLmeans**
  - Average SSIM: 0.662

- **Multivariate SURE-LET**
  - Average SSIM: 0.739

- **Fast TV**
  - Average SSIM: 0.704

- **Multivariate SURE-LET**
  - Average SSIM: 0.739

- **BLS-GSM**
  - Average SSIM: 0.732

- **Multivariate SURE-LET**
  - Average SSIM: 0.739

- **K-SVD**
  - Average SSIM: 0.711

- **Multivariate SURE-LET**
  - Average SSIM: 0.739
The Redundant Case: Visual Comparisons

**BM3D**

**Multivariate SURE-LET**

Average SSIM: 0.754

Average SSIM: 0.739

Color Images: PSNR Comparisons

**Peppers 512 x 512**

Multichannel SURE-LET (baseline)

Non-redundant multichannel

SURE-LET

**Mandrill 512 x 512**

Peppers 512 x 512

Multichannel SURE-LET (baseline)

ProbShrink-YUV Pizurica et al. 2005

ProbShrink-MB Pizurica et al. 2006

CBM3D Dabov et al. 2007

Color Images: Visual Comparisons

Original

**Average SSIM: 1.000**

Noisy

**Average SSIM: 0.221**
Frame-by-Frame PSNR Comparisons

**Flowers** at PSNR = 24.61 dB

- Output PSNR in [dB]
  - Frame no
  - 20
  - 27
  - 29
  - 30

**Bus** at PSNR = 20.17 dB

- Output PSNR in [dB]
  - Frame no
  - 26
  - 27
  - 29

Visual Comparison

Noisy Input

- PSNR = 20.17 dB

Multiframe SURE-LET (CS = 5)

- PSNR = 31.62 dB

More Realistic Measurement Model

Most light intensity measurements $y = [y_1 \ldots y_N]^T$ are more accurately modeled as a vector $z$ of independent Poisson random variables degraded by independent AWGN $b$:

$$y = z + b,$$ where $z \sim P(x)$ and $b \sim \mathcal{N}(0, \sigma^2 I_d)$

This model accounts for:

- Random nature of photon emission/detection
- Signal-dependent degradation;
- Thermal instabilities of the electronic devices
- Signal-independent noise.

Only few denoising algorithms consider this hybrid measurement model.

Two Main Approaches for Poisson Intensity Estimation

- Variance-stabilizing transform (VST):
  Design a transform $T$ such that $T(y) \sim \mathcal{N}(0, 1)$
  - Ancombe and its extension to Poisson-Gaussian noise
  - Direct handling of Poisson statistics:
    - Almost exclusively in a Bayesian framework
    - Multiscale Bayesian model
    - Hypothesis testing
    - Penalized likelihood

Potential of purely data-driven, prior-free MMSE techniques remains under-exploited.
The Unnormalized Haar Wavelet Transform

Denoising by *interscale* thresholding of the unnormalized Haar-wavelet coefficients: set $s_0 = y$, then for $j = 1, 2, \ldots, J$

$$
\begin{array}{ccc}
1 - z^{-1} & 1 & \theta(d^j, s^j) \\
1 + z^{-1} & 2 & s^j
\end{array}
\xrightarrow{\delta^j} \begin{array}{cc}
1 - \frac{z}{2} & 1 \\
1 + \frac{z}{2} & 2
\end{array} \xrightarrow{} \chi^j
$$

Haar conservation properties:

- **Error energy**: $\text{MSE} = \frac{2 - J}{N^2} \|\tilde{z}^j - \chi^j\|^2 + \sum_{j=1}^{J} \frac{2 - j}{N} \|\delta^j - \delta^j\|^2$

- **Statistics**: $s^j \sim \mathcal{P}(c^j) + \mathcal{N}(0, \sigma^2_j \text{Id})$, where $\sigma^2_j = 2^j \sigma^2$

Allows independent processing of each wavelet subband.

---

**PURE: Poisson-Gaussian Unbiased Risk Estimate**

Let $y = z + b$ with $z \sim \mathcal{P}(x)$ independent of $b \sim \mathcal{N}(0, \sigma^2 \text{Id})$. Let $f(y) = [f_n(y)]_{1 \leq n \leq N}$ such that $\delta \{\partial f_n(y)/\partial y_n\} < +\infty$. Then,

$$
\text{PURE} = \frac{1}{N} \left( \|f(y)\|^2 - 2y^T f(y) + 2\sigma^2 \text{div} \{f^T(y)\} \right) + \frac{1}{N} \left( y^T - 1 \right)^2 - \sigma^2
$$

is an unbiased estimate of the expected MSE; i.e.,

$$
\delta \{\text{PURE} \} = \frac{1}{N} \delta \{\|f(y) - x\|^2\}
$$

Notation: $f^T(y) = [f_n(y - e_n)]_{1 \leq n \leq N}$, where $(e_n)_{1 \leq n \leq N}$ is the canonical basis of $\mathbb{R}^N$.

---

**Sketch of proof**: Need to estimate

$$
\delta \left\{ \|f(y) - x\|^2 \right\} = \sum_{n} \left( \delta \{f_n^2(y)\} - 2 \delta \{x_n f_n(y)\} + x_n^2 \right)
$$

1. **Poisson’s Lemma** (Hudson 1978, Tsui & Press 1982):

   $$
   \delta \{x_n f_n(y)\} = \delta \{x_n f_n(z + b)\} = \delta \{z_n f_n(z + b - e_n)\}
   $$

2. **Stein’s Lemma** (Stein 1981):

   $$
   \delta \{z_n f_n(z + b - e_n)\} = \delta \{y_n f_n(y - e_n)\} - \delta \{b_n f_n(z + b - e_n)\} = \delta \{y_n f_n(y - e_n)\} - \sigma^2 \delta \{f_n(y - e_n)/\partial y_n\}
   $$

3. Notice that: $x_n^2 = \delta \{x_n y_n\}$

4. **SURE**

   $$
   \delta \{y_n (y_n - 1)\} - \sigma^2
   $$

---

**Interscale Haar-Wavelet-Domain PURE**

Let $\theta(d, s) = \theta^j(d^j, s^j)$ be an estimate of the noise-free wavelet coefficients $\hat{\theta} = \delta^j$. Define $\hat{\theta}^d(d, s)$ and $\hat{\theta}^s(d, s)$ by

$$
\hat{\theta}^d(d, s) = \theta_n(d + e_n, s - e_n) \\
\hat{\theta}^s(d, s) = \theta_n(d - e_n, s - e_n)
$$

Then the random variable $1$

$$
\text{PURE}_j = \frac{1}{N} \left( \|\hat{\theta}(d, s)\|^2 + \|d\|^2 - 1^T s - N \sigma^2 \right)
$$

is an unbiased estimate of the expected MSE for the $j$th subband; i.e.,

$$
\delta \{\text{PURE}_j\} = \delta \{\text{MSE}_j\}
$$

1 A similar result for pure Poisson noise can be found in Hirakawa et al. 2009.
PURE for Arbitrary Nonlinear Processing

**Problem:** PURE is time-consuming to compute for an arbitrary nonlinear processing due to the term: $f'(y) = [f_n(y - \theta_n)]_{n \leq N}$.

**Solution:** First-order Taylor series approximation of $f'(y)$ given by $f'(y) \approx f(y) - \partial f(y)$, where $\partial f(y) = [\partial f_n(y)]_{n \leq N}$.

Consequently, provided that each $f_n$ varies slowly, PURE is well-approximated by

$$\hat{\text{PURE}} = \frac{1}{N} \{ ||f(y)||^2 - 2y^T(f(y) - \partial f(y)) + 2\sigma^2 \text{div} \{ f(y) - \partial f(y) \} \} + \frac{1}{N} \{ ||y||^2 - 1^T y \} - \sigma^2$$

**Example:** Undecimated Haar Thresholding

Undecimated Haar filterbank: $H(z) = \frac{1}{\sqrt{2}}(1 + z^{-1})$ and $G(z) = \frac{1}{\sqrt{2}}(1 - z^{-1})$
Some PSNR Comparisons

**Cameraman** 256 x 256

**Cells** 512 x 512

<table>
<thead>
<tr>
<th>Input PSNR [dB]</th>
<th>Relative Output Gain [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.2</td>
</tr>
<tr>
<td>10</td>
<td>0.5</td>
</tr>
<tr>
<td>15</td>
<td>1.0</td>
</tr>
<tr>
<td>20</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Haar PURE-LET (baseline)

Haar PURE-LET

G cycle-spins

Redundant PURE-LET

Anscome + BLS-GSM Portilla et al. 2003

Platelet Willett & Nowak 2007

PH-HMT Lefkimmiatis et al. 2009

Some Visual Comparisons

**Original**

**Redundant PURE-LET**

Average SSIM: 1.000

Average SSIM: 0.543

**Haar-Fisz**

**Redundant PURE-LET**

Average SSIM: 0.445

Average SSIM: 0.543
Fluorescence Microscopy

A fluorescence microscope is an imaging system that performs:

- Excitation of fluorescent constituents of a specimen;
- Focusing/filtering of the fluorescent light emitted from the specimen;
- Amplification/quantification of the light received at the ocular.

Combined with protein tagging (e.g., with GFP), fluorescence microscopy allows to image selected fine structures of living cells.
Noise in Fluorescence Microscopy

Three main sources:

- **Photon-counting noise**: major source of noise due to the random nature of photon emission/detection (signal-dependent);
- **Measurement noise**: thermal instabilities of the various electronic devices (signal-independent);
- **Other**: autofluorescence and bleaching (reduced by short exposure and low fluorophore concentration).

~ Measurement model: scaled Poisson distributed degraded by AWGN

\[
y \sim \alpha \mathcal{P}(x) + \mathcal{N}(\mu, \sigma^2)
\]

- \(\alpha\): detector gain
- \(\mu\): detector offset
- \(\sigma^2\): AWGN variance

Experiments: 2D Sample

Specifications:

- 512 \times 512 image acquired on a confocal microscope at the Imaging Center of the IGBMC, France;
- *C. elegans* embryo labeled with 3 fluorescent dyes;
- Each channel has been processed independently.

Raw Data

UWT PURE-LET

Experiments: 3D Sample

Specifications:

- 1024 \times 1024 \times 64 volume of confocal microscopy images;
- Fibroblast cells labeled with *DIO* and 100nm fluorescent beads;
- Voxel resolution: 0.09 \times 0.09 \times 0.37 \mu m^3.

Raw Data

Multislice Haar PURE-LET

Noise Parameters Estimation

Affine relationship between sample-mean and sample-variance:

\[
\begin{align*}
\mu_y & \quad \text{def} \quad \mathbb{E}\{y\} = \alpha x + \mu \\
\sigma^2_y & \quad \text{def} \quad \text{Var}\{y\} = \alpha^2 x + \sigma^2
\end{align*}
\]

\[
\sigma^2_y = \alpha \mu_y + \sigma^2 - \alpha \mu \beta
\]

Simple estimation procedure: (similar to Lee 1989, Boulanger et al. 2007)

1. Compute \(\mu_y\) and \(\sigma^2_y\) in many small regions of the noisy image.
2. Perform a robust linear regression on the set of points \((\mu_y, \sigma^2_y)\).
3. Identify \(\alpha\) as the slope of the fitted line and \(\beta\) as the ordinate at \(\mu_y = 0\).
4. \(\sigma^2\) and \(\mu\) can be estimated independently in signal-free regions and cross-checked with \(\beta\).
Experiments: 3D Sample

Specifications:
- 1024 × 1024 × 64 volume of confocal microscopy images;
- Fibroblast cells labeled with DiO and 100nm fluorescent beads;
- Voxel resolution: 0.09 × 0.09 × 0.37μm³.

3D Median Filter

Multislice Haar PURE-LET

Experiments: 2D Timelapse Sequence

Specifications:
- 448 × 512 × 100 image sequence of confocal microscopy images;
- C. elegans embryos labeled with GFP;

Raw Data

Multiframe Haar PURE-LET

Conclusion

Presentation of a generic methodology for building signal/image denoising algorithms.

Advantages:
- Does not require hypotheses on the signal, only on the noise (SURE/PURE);
- No parameters to tune;
- Fast, non-iterative (SURE/PURE + LET);
- Natural construction of multivariate/redundant thresholding rules.

Although they involve only simple thresholding operations in a transformed domain (single step, no training, no block-matching, no direction/edge detection), the proposed algorithms reach the state of the art in image/video denoising.

Main References


Internet links

- Authors: thierry.blu@m4x.org and florian.luisier@a3.epfl.ch
- Papers: www.ee.cuhk.edu.hk/~tblu/ and bigwww.epfl.ch/
- Demos: bigwww.epfl.ch/
  - Orthonormal grayscale and color image denoising
- Software: bigwww.epfl.ch/
  - Matlab implementations of SURE-LET algorithms
  - PURE-LET denoising plugin for ImageJ