Image Denoising and the SURE-LET Methodology

Thierry Blu[1] **and Florian Luisier**[2]

[1]Department of Electronic Engineering
The Chinese University of Hong Kong

[2]Biomedical Imaging Group
École Polytechnique Fédérale de Lausanne

Statistics and Information Sciences Laboratory
Harvard University, Cambridge, MA

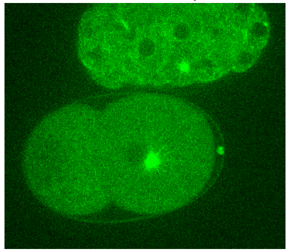
September 2010

Outline

Outline

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,...).

Noise in Images: Measurement Model

- Usual acquisition devices provide signals[1]

$$\mathbf{y} = [y_1, y_2, \ldots, y_N]^{\mathrm{T}}$$

that are corrupted with noise.

- Frequent modeling using an **additive white Gaussian noise (AWGN)** hypothesis

$$\underbrace{\mathbf{y}}_{\text{noisy signal}} = \underbrace{\mathbf{x}}_{\text{original signal}} + \underbrace{\mathbf{b}}_{\text{noise}}$$

where $\mathscr{E}\{\mathbf{b}\} = \mathbf{0}$ and $\mathscr{E}\{\mathbf{b}\mathbf{b}^{\mathrm{T}}\} = \sigma^2\mathbf{Id}$.

- **Signal denoising** consists in finding a "good" candidate $\hat{\mathbf{x}}$ of $\mathbf{x}$ using **the noisy signal $\mathbf{y}$ only**; i.e., find the algorithm $\mathbf{F}$ such that

$$\hat{\mathbf{x}} = \mathbf{F}(\mathbf{y})$$

[1] Images are represented as *vectors*, using lexicographic ordering.

An Abundant Literature

Many approaches available, based on:

1. **Explicit hypotheses on the signal:**
   - Statistics-based: wavelet-domain (Bayesian) inference Donoho *et al.* 1994, Simoncelli *et al.* 1996, Abramovich *et al.* 1998, Vidakovic *et al.* 1998;
   - Regularization: Total Variation (TV) Osher *et al.* 1992;
   - PDE: anisotropic diffusion Perona *et al.* 1990;
2. **Heuristics:**
   - Filtering: Bilateral Filter Tommasi *et al.* 1998;
   - Patch-based: Non-Local Means Buades *et al.* 2005;
   - Any combination of approaches 1 when the hypotheses are not satisfied/checked.

NOTE:

- Some approaches can be either applied in the *signal-domain* or in a *transform-domain*.
- Most approaches involve several *nonlinear* parameters which are often set *empirically*.

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:

$$\mathscr{P}\{\mathbf{y}|\mathbf{x}\} = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}\|^2}{2\sigma^2}\right)$$

- The probability density function (pdf) of the original signal $\mathscr{P}\{\mathbf{x}\}$.

Highly sensitive to the modeling of
the pdf of the signal to restore.

Maximum a Posteriori

The MAP consists in choosing the estimate $\hat{\mathbf{x}}$ that maximizes the *posterior probability density*

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \mathscr{P}\{\mathbf{x}|\mathbf{y}\} = \arg\max_{\mathbf{x}} \mathscr{P}\{\mathbf{y}|\mathbf{x}\} \cdot \mathscr{P}\{\mathbf{x}\}$$

Optimal detector: Given noisy measurements of a signal $\mathbf{x}$ having a finite number of values x_1, x_2, ..., x_K occurring with probabilities p_1, p_2, ..., p_K, the MAP minimizes the error probability

$$\mathscr{P}\{\hat{\mathbf{x}} \neq \mathbf{x}\}$$

NOTE: Description of the prior $\mathscr{P}\{\mathbf{x}\}$ may require many nonlinear parameters.

For signals with large or infinite number of levels, the probabilistic optimality of the MAP becomes irrelevant $\rightsquigarrow$ MMSE instead.

Linear MMSE: Wiener

The Wiener "filter" consists in finding the linear[2] estimate, $\hat{\mathbf{x}} = \hat{\mathbf{A}}\mathbf{y}$, that minimizes the *Mean-Squared Error* (MSE)

$$\underbrace{\mathscr{E}\left\{\frac{1}{N}\|\hat{\mathbf{A}}\mathbf{y}-\mathbf{x}\|^2\right\}}_{\text{MSE between } \hat{\mathbf{x}} \text{ and } \mathbf{x}} = \min_{\mathbf{A}} \mathscr{E}\left\{\frac{1}{N}\|\mathbf{A}\mathbf{y}-\mathbf{x}\|^2\right\}$$

Solution: Requires only the knowledge of the covariance matrix $\mathbf{\Gamma_x} = \mathscr{E}\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$ of the original signal

$$\mathbf{x} = \mathbf{\Gamma_x}\left(\mathbf{\Gamma_x} + \sigma^2\mathbf{Id}\right)^{-1}\mathbf{y}$$

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

[2] if $\mathscr{E}\{\mathbf{x}\} = \mathbf{0}$ — an affine estimate is used, otherwise.

Nonlinear MMSE: Bayesian Least Squares

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

$$\mathscr{E}\left\{\frac{1}{N}\|\mathbf{F}(\mathbf{y})-\mathbf{x}\|^2\right\} \text{ is minimized.}$$

Solution: The posterior expectation (conditional mean):

$$\hat{\mathbf{x}} = \mathscr{E}\{\mathbf{x}|\mathbf{y}\} = \int \mathbf{x}\,\mathscr{P}\{\mathbf{x}|\mathbf{y}\}\,\mathrm{d}^N\mathbf{x} \overset{\text{Bayes}}{=} \frac{1}{\mathscr{P}\{\mathbf{y}\}}\int \mathbf{x}\,\mathscr{P}\{\mathbf{y}|\mathbf{x}\}\cdot\mathscr{P}\{\mathbf{x}\}\,\mathrm{d}^N\mathbf{x}$$

where $\mathscr{P}\{\mathbf{y}\} = \int \mathscr{P}\{\mathbf{y}|\mathbf{x}\}\cdot\mathscr{P}\{\mathbf{x}\}\,\mathrm{d}^N\mathbf{x}$ is the marginal pdf of $\mathbf{y}$.

NOTE: The above integrals often need to be computed numerically.

The Bayesian MMSE algorithm requires the knowledge of the *pdf of the unknown signal* $\rightsquigarrow$ **Choice of prior ?**

Nonlinear MMSE: One Step Further

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

$$\mathscr{E}\left\{\frac{1}{N}\|\mathbf{F}(\mathbf{y})-\mathbf{x}\|^2\right\} \text{ is minimized.}$$

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

$$\hat{\mathbf{x}} = \mathbf{y} + \sigma^2\nabla\log\mathscr{P}\{\mathbf{y}\}$$

NOTE: Because $\mathscr{P}\{\mathbf{y}\} = \overbrace{\int \mathscr{P}\{\mathbf{y}|\mathbf{x}\}\cdot\mathscr{P}\{\mathbf{x}\}\,\mathrm{d}^N\mathbf{x}}^{\text{convolution with a Gaussian}}$, the optimal MSE processing is infinitely differentiable.

The optimal algorithm only requires the knowledge of the *pdf of the observed noisy signal* $\rightsquigarrow$ **No prior information is needed !**

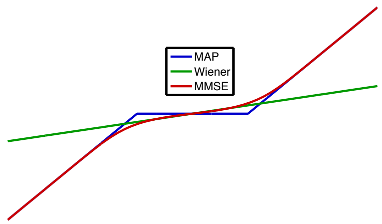
Examples

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

MAP $\hat{x}_n = \mathrm{soft}_T(y_n)$

Wiener $\hat{x}_n = \dfrac{y_n}{1+\frac{T^2}{2\sigma^2}}$

MMSE $\hat{x}_n = y_n - T\dfrac{\mathrm{e}^{-\lambda y_n}\,\mathrm{erfc}\left(\frac{-y_n+T}{\sigma\sqrt{2}}\right) - \mathrm{e}^{\lambda y_n}\,\mathrm{erfc}\left(\frac{y_n+T}{\sigma\sqrt{2}}\right)}{\mathrm{e}^{-\lambda y_n}\,\mathrm{erfc}\left(\frac{-y_n+T}{\sigma\sqrt{2}}\right) + \mathrm{e}^{\lambda y_n}\,\mathrm{erfc}\left(\frac{y_n+T}{\sigma\sqrt{2}}\right)}$



Regularization Approaches

The signal estimate $\hat{\mathbf{x}}$ is selected as the minimizer of a (convex) regularized cost-functional

$$J(\mathbf{x},\mathbf{y}) = \underbrace{\Psi(\mathbf{x},\mathbf{y})}_{\text{data-fidelity term}} + \underbrace{\lambda\,\Phi(\mathbf{x})}_{\text{penalty}}$$

Typical choice of data-fidelity term:

$$\Psi(\mathbf{x},\mathbf{y}) = \|\mathbf{y}-\mathbf{x}\|^2 \propto \text{ negative log-likelihood (AWGN)}$$

Typical choices of penalty:

- Tikhonov (smoothness prior): $\Phi(\mathbf{x}) = \|\mathbf{L}\mathbf{x}\|^2$;
- Sparsity prior: $\Phi(\mathbf{x}) = \|\mathbf{x}\|_{\ell_0} \rightsquigarrow \Phi(\mathbf{x}) = \|\mathbf{x}\|_{\ell_1}$;
- TV (edge prior): $\Phi(\mathbf{x}) = \|\,|\nabla\mathbf{x}|\,\|_{\ell_1}$.

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

No explicit distance minimization between original and denoised signal.

Minimizing $\mathscr{E}\left\{\|\mathbf{F}(\mathbf{y})-\mathbf{x}\|^2\right\}$ yields an algorithm $\mathbf{F}: \mathbf{y} \mapsto \hat{\mathbf{x}}$ that *depends on the probability of* $\mathbf{y}$ *alone*: $\mathbf{F}(\mathbf{y}) = \mathbf{y} + \sigma^2 \nabla \log \mathscr{P}\{\mathbf{y}\}$.

Problem: we have only one realization of the noisy image $\mathbf{y}$.
Solution: *estimate* $\mathscr{E}\left\{\|\mathbf{F}(\mathbf{y})-\mathbf{x}\|^2\right\}$ from $\mathbf{y}$, instead of $\mathscr{P}\{\mathbf{y}\}$.

MSE estimation

Consider the random variable[a]

$$\text{SURE}(\mathbf{y}) = \frac{1}{N}\|\mathbf{F}(\mathbf{y})-\mathbf{y}\|^2 + \frac{2\sigma^2}{N}\text{div}\left\{\mathbf{F}(\mathbf{y})\right\} - \sigma^2$$

Under the *additive white Gaussian noise* hypothesis, this random variable is an *unbiased estimate of the MSE* Stein et al. 1981

$$\mathscr{E}\left\{\text{SURE}(\mathbf{y})\right\} = \mathscr{E}\left\{\|\mathbf{F}(\mathbf{y})-\mathbf{x}\|^2/N\right\}$$

[a] Divergence operator: $\text{div}\left\{\mathbf{F}(\mathbf{y})\right\} \stackrel{\text{def}}{=} \sum_k \frac{\partial F_k(\mathbf{y})}{\partial y_k}$.

The original signal $\mathbf{x}$ may, or may not be random.
No assumptions on $\mathbf{x}$ are needed.

A simple proof

On the one hand (remember that $\mathbf{y} = \mathbf{x} + \mathbf{b}$)

$$\mathscr{E}\left\{\|\mathbf{F}(\mathbf{y})-\mathbf{x}\|^2\right\} = \mathscr{E}\left\{\|\mathbf{F}(\mathbf{y})\|^2\right\} - 2\underbrace{\mathscr{E}\left\{\mathbf{x}^{\mathrm{T}}\mathbf{F}(\mathbf{y})\right\}}_{\mathscr{E}\{(\mathbf{y}-\mathbf{b})^{\mathrm{T}}\mathbf{F}(\mathbf{y})\}} + \underbrace{\|\mathbf{x}\|^2}_{\mathscr{E}\{\|\mathbf{y}\|^2\}-N\sigma^2}$$

$$= \mathscr{E}\left\{\|\mathbf{F}(\mathbf{y})-\mathbf{y}\|^2\right\} + 2\mathscr{E}\left\{\mathbf{b}^{\mathrm{T}}\mathbf{F}(\mathbf{y})\right\} - N\sigma^2$$

and on the other hand (*Stein's Lemma*)

$$\mathscr{E}\left\{\mathbf{b}^{\mathrm{T}}\mathbf{F}(\mathbf{y})\right\} = \int \underbrace{\mathscr{P}\{\mathbf{b}\}\,\mathbf{b}^{\mathrm{T}}}_{-\sigma^2\nabla\mathscr{P}\{\mathbf{b}\}^{\mathrm{T}}}\mathbf{F}(\mathbf{x}+\mathbf{b})\,\mathrm{d}^N\mathbf{b} \qquad \text{(Gaussian pdf)}$$

$$= \int \sigma^2\,\mathscr{P}\{\mathbf{b}\}\,\text{div}\left\{\mathbf{F}(\mathbf{x}+\mathbf{b})\right\}\mathrm{d}^N\mathbf{b} \qquad \text{(by parts)}$$

$$= \mathscr{E}\left\{\sigma^2\text{div}\left\{\mathbf{F}(\mathbf{y})\right\}\right\}$$

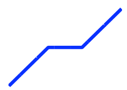
Equivalence SURE-MSE

$\text{SURE}(\mathbf{y})$ has a small variance (law of large numbers: $\propto 1/N$), which implies $\text{SURE}(\mathbf{y}) \approx \mathscr{E}\left\{\text{SURE}(\mathbf{y})\right\}$. Hence

$$\frac{1}{N}\|\mathbf{F}(\mathbf{y})-\mathbf{x}\|^2 \approx \text{SURE}(\mathbf{y})$$

NOTE: The SURE–MSE match worsens when $\mathbf{F}(\mathbf{y})$ is less *regular*; some boundedness of $\text{div}\left\{\mathbf{F}(\mathbf{y})\right\}$ is needed $\rightsquigarrow$ hard-threshold excluded.

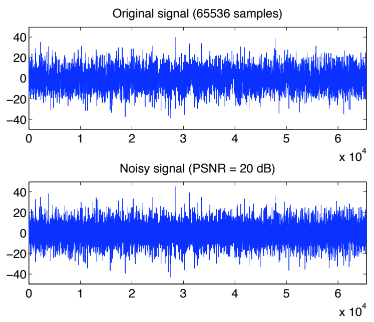
Example Donoho 1995: SURE soft-threshold

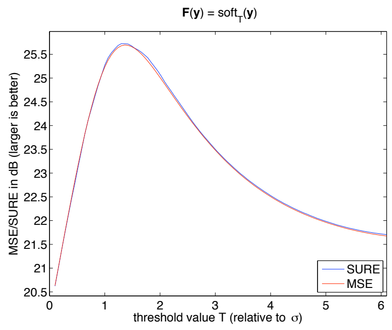


$$\text{SURE}_{\text{soft}} = \frac{1}{N}\Bigg(\overbrace{\sum_{|y_n|\le T} y_n^2 + \sum_{|y_n|\ge T} T^2}^{\|\mathbf{F}(\mathbf{y})-\mathbf{y}\|^2} + 2\sigma^2 \overbrace{\sum_{|y_n|\ge T} 1}^{\text{div}\{\mathbf{F}(\mathbf{y})\}}\Bigg) - \sigma^2$$

Closeness between SURE and MSE

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





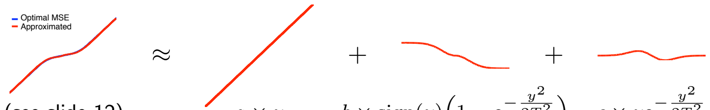
NOTE: The use of the SURE (instead of the MSE) is particularly justified for large data sizes (e.g., images).

Approximation of processings

Functions can often be efficiently approximated onto adapted bases.

Examples of bases: wavelets ($\mathbf{L}^2$ functions), sinc kernels (bandlimited functions), radial basis functions (scattered points interpolation), etc.

The MMSE result $\mathbf{F}(\mathbf{y}) = \mathbf{y} + \sigma^2 \nabla \log \mathscr{P}\{\mathbf{y}\}$ 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.



(see slide 12) $\approx$ $a \times y$ $+$ $b \times \mathrm{sign}(y)\left(1 - \mathrm{e}^{-\frac{y^2}{2T^2}}\right)$ $+$ $c \times y\mathrm{e}^{-\frac{y^2}{2T^2}}$

Linear Expansion of Thresholds

An approximation of the optimal denoising process as a (finite) linear combination of elementary processes

$$\mathbf{F}(\mathbf{y}) = \sum_{k=1}^{K} a_k \mathbf{F}_k(\mathbf{y})$$

The approximation is all the better as the order, K, is larger.

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*.

The idea of LET is that a genuine *approximation* of the optimal processing can be sufficient, while having useful *linear* properties.

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

$$\mathbf{F}(\mathbf{y}) = \sum_{\text{zones}} \overbrace{\gamma_k(\mathbf{y})}^{\text{indicator function of zone } k} \mathbf{A}_k \mathbf{y}$$

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

Choosing the LET basis

Example: A simple threshold tends to segment a signal into large values, and small values. A possible choice[3] for the indicator function of the small values is

$$\gamma(y) = \mathrm{e}^{-\frac{y^2}{2T^2}}$$

Then, a possible LET function is of the form

$$F(y) = \underbrace{\gamma(y) \times ay}_{\text{small } y} + \underbrace{(1-\gamma(y)) \times by}_{\text{large } y}$$

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

NOTE: A practical choice for T is $\sqrt{6}\,\sigma$ (noise), which can be related to a significance level in a statistical test.

[3] for a $\tanh$-based threshold, see Pesquet *et al.* 1997

Recapitulation of the SURE-LET approach

1. Instead of finding an *approximation of the signal* $\mathbf{x}$, find an *approximation of the processing* $\mathbf{F}(\mathbf{y})$ that transforms $\mathbf{y}$ into $\hat{\mathbf{x}}$;
2. Instead of minimizing the MSE between $\hat{\mathbf{x}}$ and $\mathbf{x}$, minimize an (unbiased) *estimate* of this MSE, based on $\mathbf{y}$ alone (SURE);
3. Express $\mathbf{F}(\mathbf{y})$ as a linear decomposition (LET) $\sum_k a_k \mathbf{F}_k(\mathbf{y})$ of basis processings $\mathbf{F}_k(\mathbf{y}) \rightsquigarrow$ 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.
$\rightsquigarrow$ faster algorithm, and better agreement between MSE and SURE.

The SURE minimization

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

$$\sum_{l=1}^{K} \mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{F}_l(\mathbf{y})\, a_l = \mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{y} - \sigma^2 \mathrm{div}\,\{\mathbf{F}_k(\mathbf{y})\}$$

Finally, by stacking the LET coefficients in $\mathbf{a} = [a_1, a_2, \ldots, a_K]^{\mathrm{T}}$, we get

$$\mathbf{a} = \mathbf{M}^{-1}\mathbf{c} \quad \text{where} \quad \left| \begin{array}{l} \mathbf{M} = \left[\mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{F}_l(\mathbf{y})\right]_{1 \le k, l \le K} \\ \mathbf{c} = \left[\mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{y} - \sigma^2 \mathrm{div}\,\{\mathbf{F}_k(\mathbf{y})\}\right]_{1 \le k \le K} \end{array} \right.$$

NOTE: When $\mathbf{M}$ is non-invertible, it means that one LET basis element depends linearly on the other $\mathbf{F}_k \rightsquigarrow$ decrease the LET-order to $K-1$.

The Oracle minimization

The same LET optimization, by minimizing the MSE $\|\mathbf{F}(\mathbf{y}) - \mathbf{x}\|^2$ instead of the SURE yields, for all $k = 1, 2, \ldots, K$

$$\sum_{l=1}^{K} \mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{F}_l(\mathbf{y})\, a_l = \mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{x}$$

This also boils down to solving a linear system of equations

$$\mathbf{a} = \mathbf{M}^{-1}\mathbf{c}' \quad \text{where} \quad \left| \begin{array}{l} \mathbf{M} = \left[\mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{F}_l(\mathbf{y})\right]_{1 \le k, l \le K} \\ \mathbf{c}' = \left[\mathbf{F}_k(\mathbf{y})^{\mathrm{T}} \mathbf{x}\right]_{1 \le k \le K} \end{array} \right.$$

NOTE: The Oracle computation allows to choose elementary LET processings $\mathbf{F}_k$ 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 $\mathbf{F}(\mathbf{y};\lambda)$ depends on *one* non-linear parameter, λ, approximate it using *two* (or more) LETs

$$\mathbf{F}(\mathbf{y};\lambda) = a_1\mathbf{F}(\mathbf{y};\lambda_1) + a_2\mathbf{F}(\mathbf{y};\lambda_2)$$

where λ_1, λ_2 are fixed: $[\lambda_1, \lambda_2]$ is the expected range of values for λ.

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

$$\mathrm{div}\,\{\mathbf{F}(\mathbf{y})\} = \sum_{n=1}^{N} \frac{\partial F_n(\mathbf{y})}{\partial y_n}$$

may prove too much CPU intensive.

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

$$\mathrm{div}\,\{\mathbf{F}(\mathbf{y})\} \approx \mathbf{b}_0^{\mathrm{T}} \frac{\mathbf{F}(\mathbf{y}+\varepsilon\mathbf{b}_0) - \mathbf{F}(\mathbf{y})}{\varepsilon} \quad \text{(law of large numbers)}$$

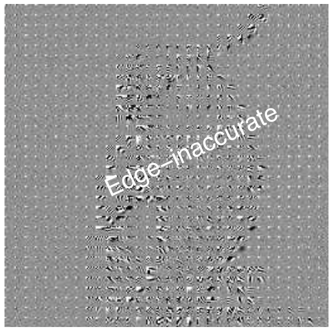
where $\mathbf{b}_0$ is a normalized (unit-variance, zero-mean) Gaussian white noise. ε is some small value compared to the level of noise; typ., $\varepsilon = \sigma/100$.

NOTE: Particularly useful when $\mathbf{F}(\mathbf{y})$ is not obtained explicitely, but through a "black-box" algorithm like TV regularization Ramani *et al.* 2008.

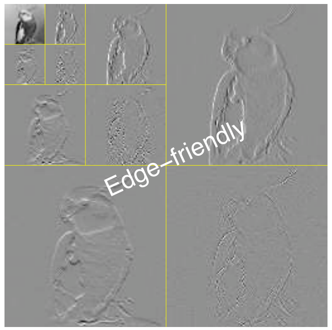
Linear transformations

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

BDCT transformed image (block–size = 8)



Wavelet decomposition (3 iterations)



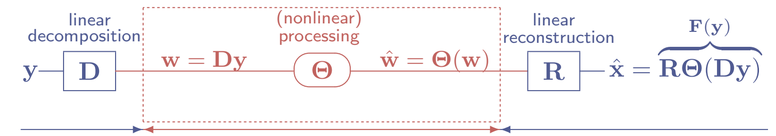
Most generally, a linear transformation maps an image $\mathbf{y}$ onto another image $\mathbf{w}$ through a matrix multiplication $\mathbf{Dy}$. It is assumed that the transformation can be inverted using a matrix $\mathbf{R}$.

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

- Perfect reconstruction: $\mathbf{RD} = \mathbf{Id}$;
- $\mathbf{D}$ 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.



Graphical overview: transform-domain thresholding

SURE-LET methodology: specify a LET basis $\mathbf{F}_k(\mathbf{y})$ as follows

$$\boldsymbol{\Theta}(\mathbf{w}) = \sum_{k=1}^{K} a_k \boldsymbol{\Theta}_k(\mathbf{w}) \rightsquigarrow \mathbf{F}_k(\mathbf{y}) = \mathbf{R}\boldsymbol{\Theta}_k(\mathbf{D}\mathbf{y})$$

Potential issue: efficient computation[4] of the SURE (essentially the $\mathrm{div}\,\{\mathbf{F}_k\}$ term) for this type of processing $\rightsquigarrow$ Monte-Carlo technique.

[4]However, *exact* expression in a number of practical cases (periodic extensions).

Orthonormality

A decomposition is orthonormal iff $\mathbf{D}^{\mathrm{T}}\mathbf{D} = \mathbf{D}\mathbf{D}^{\mathrm{T}} = \mathbf{Id}$. Properties:

- The reconstruction is given by $\mathbf{R} = \mathbf{D}^{\mathrm{T}}$;
- Preservation of the energies: $\|\mathbf{w}\| = \|\mathbf{y}\|$ and $\|\hat{\mathbf{x}} - \mathbf{x}\| = \|\hat{\mathbf{w}} - \mathbf{D}\mathbf{x}\|$;
- Statistical *independence* of the transformed coefficients;

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

If $\mathbf{w}_j = \mathbf{D}_j\mathbf{y}$ for $j = 1, 2, \dots, J$ where $\mathbf{D} = [\mathbf{D}_1; \mathbf{D}_2; \dots; \mathbf{D}_J]$, then the unbiased estimate of $\|\hat{\mathbf{x}} - \mathbf{x}\|^2$ can be written in the *transformed domain*

$$\mathrm{SURE}(\mathbf{y}) = \frac{1}{N}\left(\sum_{j=1}^{J} \|\boldsymbol{\Theta}_j(\mathbf{w}) - \mathbf{w}_j\|^2 + 2\sigma^2 \mathrm{div}\,\{\boldsymbol{\Theta}_j(\mathbf{w})\}\right) - \sigma^2$$

where $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_1; \boldsymbol{\Theta}_2; \dots; \boldsymbol{\Theta}_J]$.

Optimizing the denoising process $\mathbf{F}(\mathbf{y})$ is equivalent to denoising *separately* the denoising processes $\boldsymbol{\Theta}_j$ in the transformed domain.

Simple wavelet thresholding

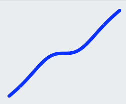
Choice of an orthonormal wavelet transform[5] (e.g., symlet 8). Then, the processing in subband j is a simple thresholding $\hat{w}_{j,n} = \theta_j(w_{j,n})$ for each of the coordinates $n = 1, 2, \dots, N_j$ of $\mathbf{w}_j$, and

$$\mathrm{SURE}_j(\mathbf{w}_j) = \frac{1}{N_j}\left(\sum_{n=1}^{N_j} \left|\theta_j(w_{j,n}) - w_{j,n}\right|^2 + 2\sigma^2\theta_j'(w_{j,n})\right) - \sigma^2$$

SURE-LET simple threshold

A two-parameter zone-selection function

$$\theta_j(w) = a_j w + b_j w \mathrm{e}^{-\frac{w^2}{12\sigma^2}}$$



where a_j and b_j are obtained by minimizing $\mathrm{SURE}_j(\mathbf{w}_j)$.

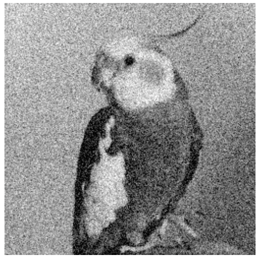
NOTE: SureShrink Donoho 1995 makes the choice $\theta_j(w) = \mathrm{soft}_{T_j}(w)$ and minimizes $\mathrm{SURE}_j(\mathbf{w}_j)$ for T_j.

[5]However, any (non-wavelet) *orthonormal* transform can be used.

In details, a_j, b_j solve the following linear system of equations

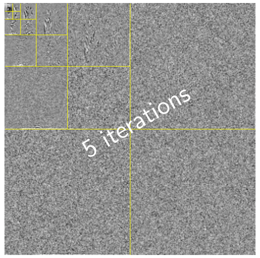
$$\frac{\partial \mathrm{SURE}_j}{\partial a_j} = 0 \rightsquigarrow \sum_{n=1}^{N_j} a_j w_{j,n}^2 + b_j w_{j,n}^2 \mathrm{e}^{-\frac{w_{j,n}^2}{12\sigma^2}} = -N_j\sigma^2 + \sum_{n=1}^{N_j} w_{j,n}^2$$

$$\frac{\partial \mathrm{SURE}_j}{\partial b_j} = 0 \rightsquigarrow \sum_{n=1}^{N_j} a_j w_{j,n}^2 \mathrm{e}^{-\frac{w_{j,n}^2}{12\sigma^2}} + b_j w_{j,n}^2 \mathrm{e}^{-\frac{w_{j,n}^2}{6\sigma^2}} = \sum_{n=1}^{N_j} \left(\frac{7}{6} w_{j,n}^2 - \sigma^2\right)\mathrm{e}^{-\frac{w_{j,n}^2}{12\sigma^2}}$$

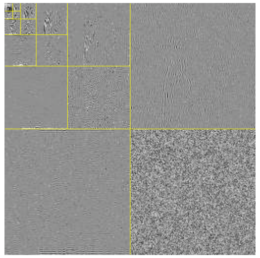


Noisy: PSNR = 18 dB

wavelet decomposition $\longrightarrow$



$\downarrow$ simple thresholding

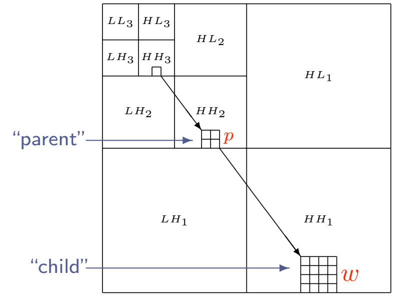


$\longleftarrow$ wavelet reconstruction

Denoised: PSNR = 29.06 dB (SureShrink: PSNR = 28.73 dB)

InterScale wavelet thresholding

The relative locality of the DWT implies that there may be a *spatial correlation* between different wavelet scales: three potential *tree*-structures — LH, HH and HL



Interscale thresholding consists in expressing the denoised estimate as

$$\hat{w}_{j,n} = \theta_j(w_{j,n}, p_{j,n})$$

InterScale wavelet thresholding

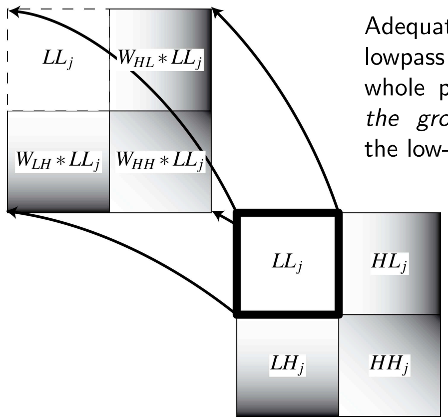
Principle: separate the parent into *large* and *small* coefficients, and within each zone so defined, apply a pointwise thresholding function:

$$\theta_j(w,p) = \underbrace{e^{-\frac{p^2}{12\sigma^2}}\left(a_j w + b_j w e^{-\frac{w^2}{12\sigma^2}}\right)}_{\text{small parents}} + \underbrace{\left(1-e^{-\frac{p^2}{12\sigma^2}}\right)\left(a'_j w + b'_j w e^{-\frac{w^2}{12\sigma^2}}\right)}_{\text{large parents}}$$

NOTE: DWT is orthogonal, hence w and p are *statistically independent* $\rightsquigarrow$ same SURE formula as for the simple threshold case.

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



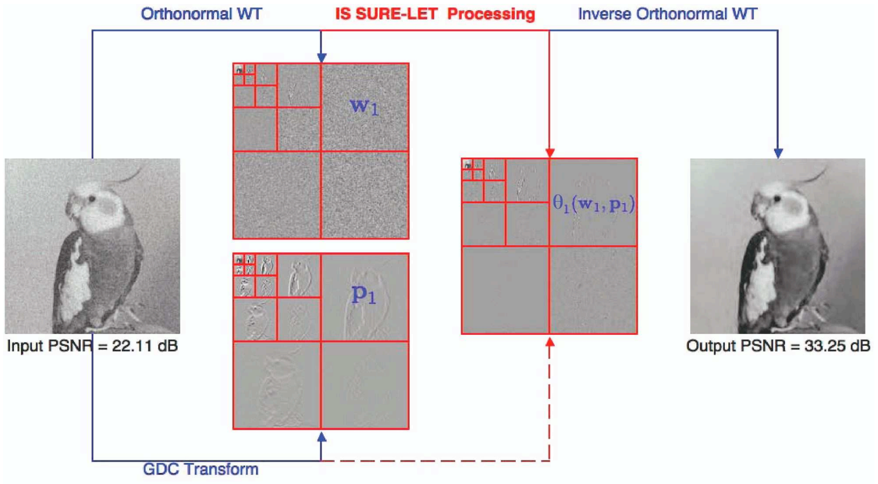
Adequate high-pass filtering of the lowpass LL_j — which contains the whole parent tree: W *compensates the group-delay* difference between the low-pass and the high-pass band.

GDC filter formula

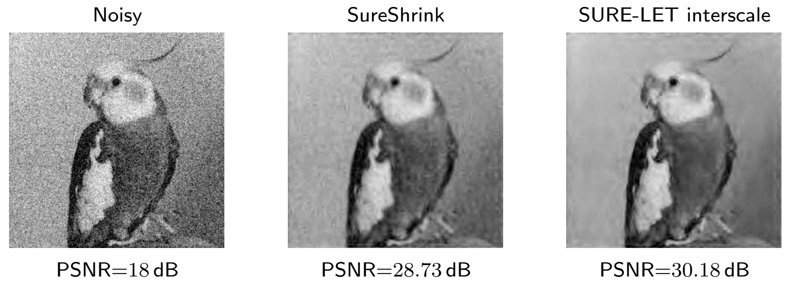
$$W(z^2) = (1 + z^{-2})G(z^{-1})G(-z^{-1})$$

where $G(z)$ = wavelet filter.

Overview of the interscale SURE-LET denoising



Example of result



Best non-redundant transform-domain algorithm.

Extension to multichannel denoising

Direct generalization by replacing:

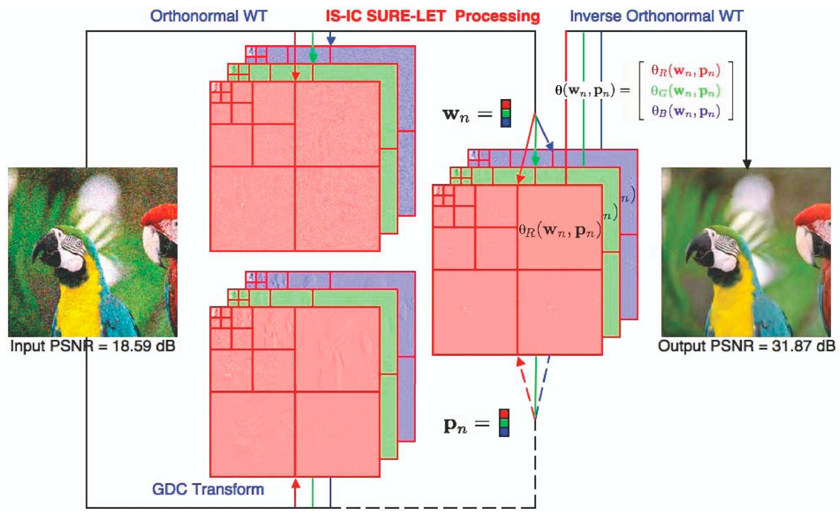
- *scalar*-valued by *vector*-valued wavelet coefficients;
- *scalar*-valued by *matrix*-valued LET parameters.

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

$$\begin{aligned}
\boldsymbol{\theta}_j(\mathbf{w},\mathbf{p}) = {} & \underbrace{\gamma(\mathbf{p}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{p})\gamma(\mathbf{w}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{w})}_{\text{small parents and small coefficients}}\,\mathbf{a}_{1,j}^{\mathrm{T}}\mathbf{w} \\
& + \underbrace{\left(1-\gamma(\mathbf{p}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{p})\right)\gamma(\mathbf{w}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{w})}_{\text{large parents and small coefficients}}\,\mathbf{a}_{2,j}^{\mathrm{T}}\mathbf{w} \\
& + \underbrace{\gamma(\mathbf{p}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{p})\left(1-\gamma(\mathbf{w}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{w})\right)}_{\text{small parents and large coefficients}}\,\mathbf{a}_{3,j}^{\mathrm{T}}\mathbf{w} \\
& + \underbrace{\left(1-\gamma(\mathbf{p}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{p})\right)\left(1-\gamma(\mathbf{w}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{w})\right)}_{\text{large parents and large coefficients}}\,\mathbf{a}_{4,j}^{\mathrm{T}}\mathbf{w}
\end{aligned}$$

NOTE: Automatically selects the best color space (color images).

Overview of the multichannel SURE-LET denoising



Undecimated wavelet denoising

Limitations of non-redundant transformations

- High sensitivity to shifts $\rightsquigarrow$ inconsistent reconstruction of edges
- Low design flexibility $\rightsquigarrow$ poor directional sensitivity

Solution: increase the redundancy

Shifts: Cycle-Spinning Coifman 1995, Undecimated DWT Guo 1995;
Rotations: Steerable Pyramid Simoncelli 1995, Complex DWT Kingsbury 1998;
Edges: Curvelets Candès 2002; etc. . .

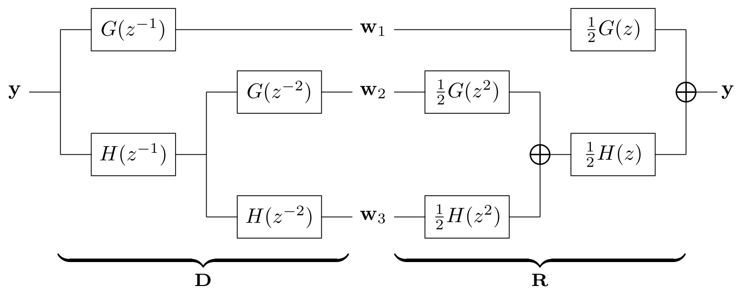
Redundancy vs orthonormality

Although it is still possible to have $\mathbf{R} = \mathbf{D}^{\mathrm{T}}$ (*tight frame*)

- $\mathbf{RD} = \mathbf{Id}$ but $\mathbf{DR} \neq \mathbf{Id}$
- Energies: $\|\mathbf{w}\| = \|\mathbf{y}\|$ (if tight frame) but $\|\hat{\mathbf{x}} - \mathbf{x}\| \neq \|\hat{\mathbf{w}} - \mathbf{Dx}\|$;
- Statistical *dependence* of the transformed coefficients;

In addition, redundancy brings about a higher computational cost.

Two iterations of a 1D UDWT



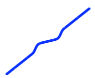
Perfect reconstruction condition: $\mathbf{RD} = \mathbf{Id}$

NOTE: same lowpass and highpass filters, $H(z)$ and $G(z)$, as in the non-redundant WT case.

Undecimated simple wavelet thresholding

Hard-like[6] thresholding rule

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



$$\theta_j(w) = a_j w + b_j w \left(1 - \mathrm{e}^{-\left(\frac{w}{3\sigma}\right)^8}\right)$$

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

Noisy	SureShrink	SURE-LET

PSNR=18 dB	PSNR=28.73 dB	PSNR=31.91 dB

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;

Orthonormal discrete symlet 8 transform

Noisy	SURE-LET interscale	SURE-LET multivariate

PSNR=18 dB	PSNR=30.18 dB	PSNR=30.65 dB

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

Noisy	SURE-LET	SURE-LET multivariate

PSNR=18 dB	PSNR=31.91 dB	PSNR=32.22 dB

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)

Noisy	SURE-LET UWT Haar	SURE-LET Multivariate Dictionary

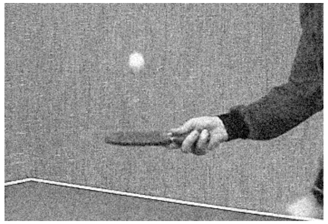
PSNR=18 dB	PSNR=25.90 dB	PSNR=28.80 dB

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;

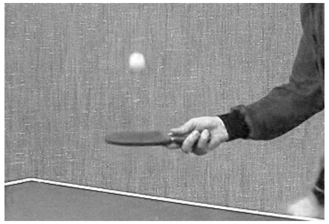
Orthonormal discrete symlet 8 transform

Noisy



PSNR=22.11 dB

Multiframe SURE-LET



PSNR=30.85 dB

Noise Variance Estimation

The most popular approach for estimating the variance σ^2 of the AWGN for wavelet-based denoising algorithms: **MAD estimator** Donoho 1995

$$\hat{\sigma} = 1.4826 \,\mathrm{med}\left\{|\mathbf{y} - \mathrm{med}\{\mathbf{y}\}|\right\},\ y_n \in HH$$

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

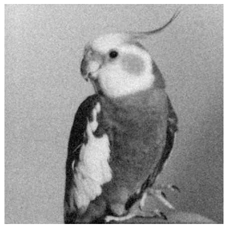
Proposed approach: **Eigenfilter-based design** Vaidyanathan *et al.* 1987

1. Find $\mathbf{h}_{\text{opt}} = \arg\min_{\mathbf{h}\in\mathbb{R}^M} \|\mathbf{h} * \mathbf{y}\|^2$ subject to $\|\mathbf{h}\|^2 = 1$
   $\rightsquigarrow$ Eigenvector corresponding to the smallest eigenvalue of the autocorrelation matrix $\boldsymbol{\Gamma}_{\mathbf{y}} = \left[\sum_{n=1}^{N} y_{n-i} y_{n-j}\right]_{1\leq i,j\leq M}$
2. Noise variance robustly estimated from the filtered residual $(\mathbf{h}_{\text{opt}} * \mathbf{y})$, as the mode of the smoothed histogram of the local noise variances computed inside blocks of given size (typically, 25×25).

Noise Variance Estimation

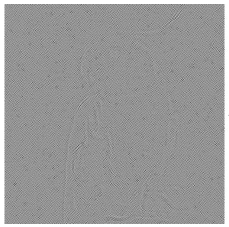
Overview of the Proposed Approach

Noisy Input: $\sigma = 10$



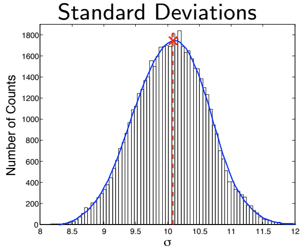
$* \,\mathbf{h}_{\text{opt}} =$

Residual



$\rightsquigarrow$

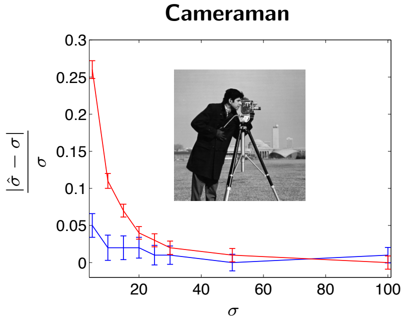
Distribution of the Local Standard Deviations

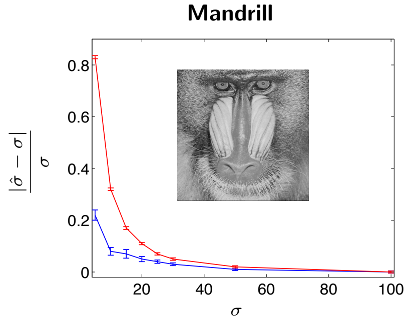


Estimated σ: $\hat{\sigma} = 10.09$

Noise Variance Estimation

Performance of the Proposed Approach





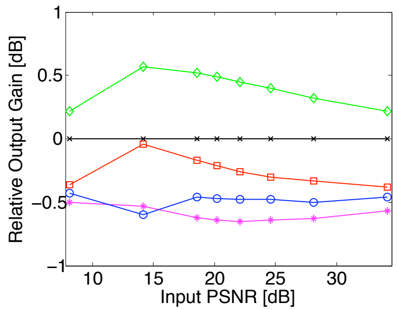
Proposed Approach MAD Estimator

Protocol for Fair Comparisons

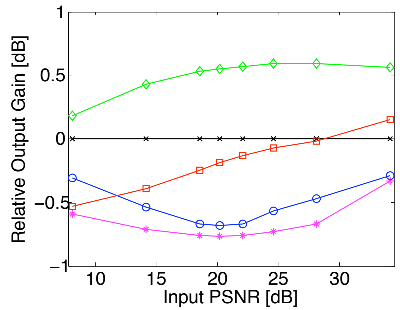
- Denoising of a representative set of standard grayscale/color images and video sequences, corrupted by simulated AWGN at 8 different powers $\sigma \in [5, 10, 15, 20, 25, 30, 50, 100]$ (assumed to be known).
- PSNR results averaged over 10 different noise realizations for each noise standard deviation.
- 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

Peppers 256×256



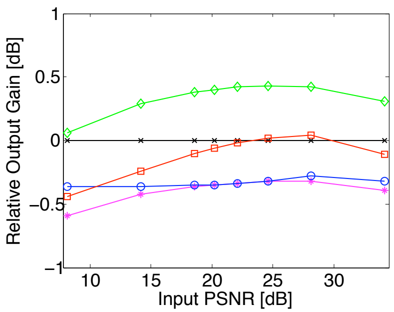
House 256×256



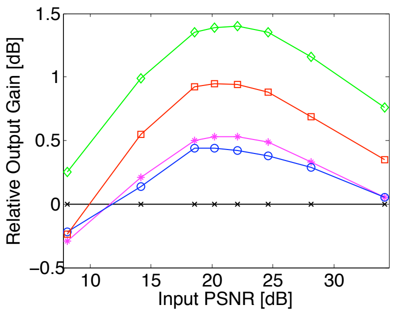
Interscale SURE-LET (baseline)
Multivariate SURE-LET
BiShrink Sendur & Selesnick 2002
ProbShrink Pižurica *et al.* 2006
BLS-GSM Portilla *et al.* 2003

The Non-Redundant Case: PSNR Comparisons

Lena 512×512



Barbara 512×512



Interscale SURE-LET (baseline)
Multivariate SURE-LET
BiShrink Sendur & Selesnick 2002
ProbShrink Pižurica *et al.* 2006
BLS-GSM Portilla *et al.* 2003

The Non-Redundant Case: Visual Comparisons

Original

Average SSIM[1]: 1.000

Noisy

Average SSIM: 0.284

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

The Non-Redundant Case: Visual Comparisons

Original

Average SSIM: 1.000

Multivariate SURE-LET

Average SSIM: 0.894

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

The Non-Redundant Case: Visual Comparisons

BiShrink

Average SSIM: 0.877

Multivariate SURE-LET

Average SSIM: 0.894

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

The Non-Redundant Case: Visual Comparisons

ProbShrink

Average SSIM: 0.882

Multivariate SURE-LET

Average SSIM: 0.894

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

The Non-Redundant Case: Visual Comparisons

BLS-GSM

Average SSIM: 0.888

Multivariate SURE-LET

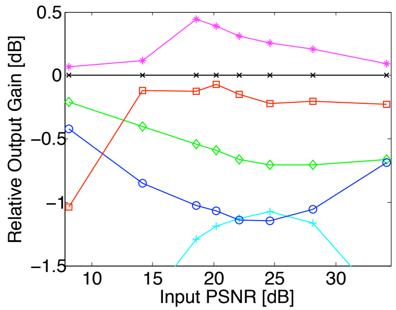
Average SSIM: 0.894

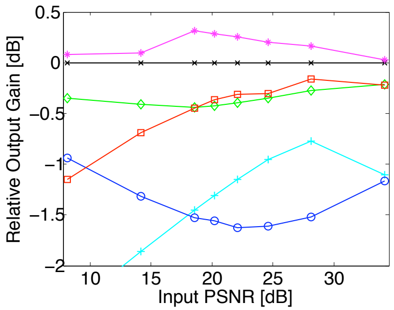
[1]Structural Similarity Index Map Wang, Bovik, Sheikh & Simoncelli 2004

The Redundant Case: PSNR Comparisons

Peppers 256×256 

Coco 256×256 



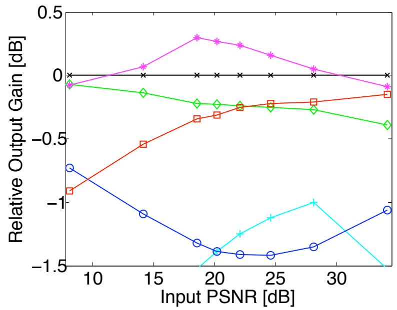


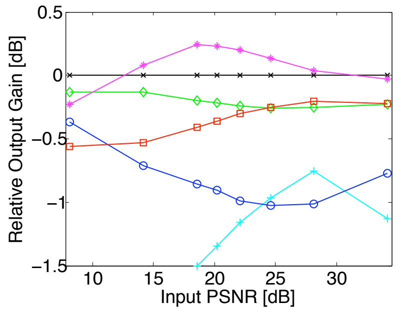
Multivariate SURE-LET (baseline)
NLmeans Buades *et al.* 2005
BLS-GSM Portilla *et al.* 2003
BM3D Dabov *et al.* 2007
Fast TV Chambolle 2004
K-SVD Elad & Aharon 2006

The Redundant Case: PSNR Comparisons

Boat 512×512 

Goldhill 512×512 





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 & Aharon 2006

The Redundant Case: Visual Comparisons

Original

Average SSIM: 1.000

Noisy

Average SSIM: 0.263

The Redundant Case: Visual Comparisons

Original

Average SSIM: 1.000

Multivariate SURE-LET

Average SSIM: 0.739

The Redundant Case: Visual Comparisons

NLmeans

Average SSIM: 0.662

Multivariate SURE-LET

Average SSIM: 0.739

The Redundant Case: Visual Comparisons

Fast TV

Average SSIM: 0.704

Multivariate SURE-LET

Average SSIM: 0.739

The Redundant Case: Visual Comparisons

BLS-GSM

Average SSIM: 0.732

Multivariate SURE-LET

Average SSIM: 0.739

The Redundant Case: Visual Comparisons

K-SVD

Average SSIM: 0.711

Multivariate SURE-LET

Average SSIM: 0.739

The Redundant Case: Visual Comparisons

BM3D

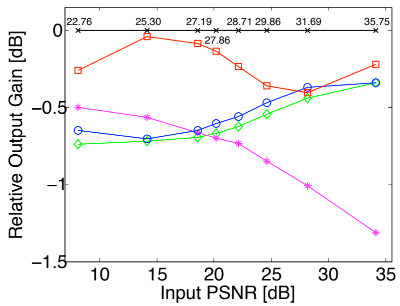
Average SSIM: 0.754

Multivariate SURE-LET

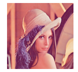
Average SSIM: 0.739

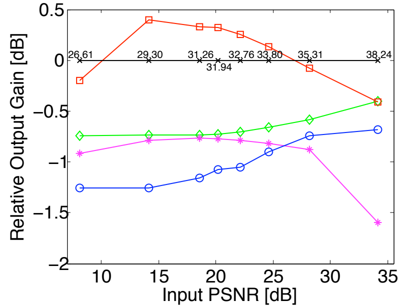
Color Images: PSNR Comparisons

Girl 256×256



Lena 512×512

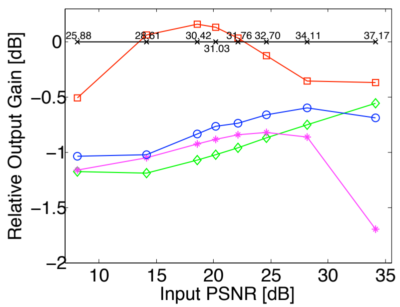




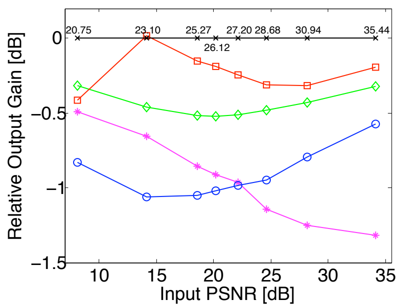
Multichannel SURE-LET (baseline)
Non-redundant multichannel SURE-LET
ProbShrink-YUV Pižurica *et al.* 2005
ProbShrink-MB Pižurica *et al.* 2006
CBM3D Dabov *et al.* 2007

Color Images: PSNR Comparisons

Peppers 512×512



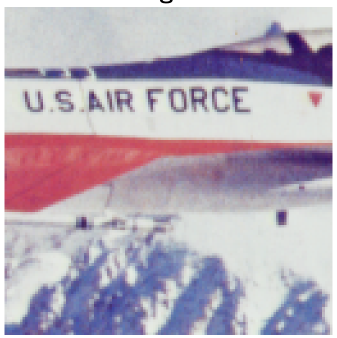
Mandrill 512×512



Multichannel SURE-LET (baseline)
Non-redundant multichannel SURE-LET
ProbShrink-YUV Pižurica *et al.* 2005
ProbShrink-MB Pižurica *et al.* 2006
CBM3D Dabov *et al.* 2007

Color Images: Visual Comparisons

Original



Average SSIM: 1.000

Noisy

Average SSIM: 0.221

Color Images: Visual Comparisons

Original

Average SSIM: 1.000

Multichannel SURE-LET

Average SSIM: 0.872

Image Denoising Methods
The SURE-LET Methodology
SURE-LET Algorithmics
Algorithm Comparisons
Extension to Poisson-Gaussian Denoising
Grayscale Image Denoising
Color Image Denoising
Video Denoising

Color Images: Visual Comparisons

ProbShrink-MB

Average SSIM: 0.825

Multichannel SURE-LET

Average SSIM: 0.872

Thierry Blu and Florian Luisier Image Denoising and the SURE-LET Methodology 56 / 80

Image Denoising Methods
The SURE-LET Methodology
SURE-LET Algorithmics
Algorithm Comparisons
Extension to Poisson-Gaussian Denoising
Grayscale Image Denoising
Color Image Denoising
Video Denoising

Color Images: Visual Comparisons

ProbShrink-YUV

Average SSIM: 0.841

Multichannel SURE-LET

Average SSIM: 0.872

Thierry Blu and Florian Luisier Image Denoising and the SURE-LET Methodology 56 / 80

Image Denoising Methods
The SURE-LET Methodology
SURE-LET Algorithmics
Algorithm Comparisons
Extension to Poisson-Gaussian Denoising
Grayscale Image Denoising
Color Image Denoising
Video Denoising

Color Images: Visual Comparisons

CBM3D

Average SSIM: 0.882

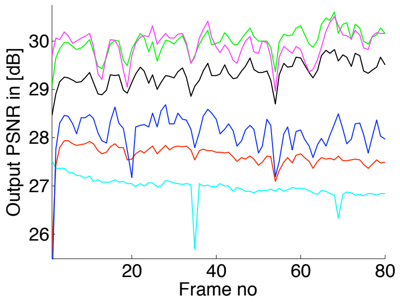
Multichannel SURE-LET

Average SSIM: 0.872

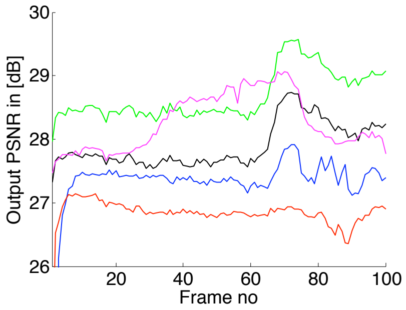
Thierry Blu and Florian Luisier Image Denoising and the SURE-LET Methodology 56 / 80

Frame-by-Frame PSNR Comparisons

Flowers at PSNR $= 24.61\,\mathrm{dB}$



Bus at PSNR $= 20.17\,\mathrm{dB}$



Multiframe SURE-LET (OWT)
Multiframe SURE-LET ($\mathrm{CS} = 5$)
SEQWT Pižurica *et al.* 2004
WRSTF Zlokolica *et al.* 2006
Real-time WRSTF Jovanov *et al.* 2009
VBM3D Dabov *et al.* 2007

Visual Comparison

Noisy Input

PSNR $= 20.17\,\mathrm{dB}$

Multiframe SURE-LET ($\mathrm{CS} = 5$)

PSNR $= 31.62\,\mathrm{dB}$

More Realistic Measurement Model

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

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

This model accounts for:

- Random nature of photon emission/detection
  $\rightsquigarrow$ **signal-dependent** degradation;
- Thermal instabilities of the electronic devices
  $\rightsquigarrow$ **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 $\mathrm{T}(y) - \mathrm{T}(x) \xrightarrow[x \to +\infty]{\mathcal{D}} \mathcal{N}(0, 1)$
  - Anscombe and its extension to Poisson-Gaussian noise Murtagh *et al.* 1995;
  - Haar-Fisz Fryzlewicz & Nason 2004;
  - Multiscale VST Jansen 2006, Fadili *et al.* 2008.
- **Direct handling of Poisson statistics:**
  Almost exclusively in a Bayesian framework
  - Multiscale Bayesian model Nowak *et al.* 1999;
  - Hypothesis testing Kolaczyk 1999, Fadili *et al.* 2007;
  - Penalized likelihood Sardy *et al.* 2004, Willett & Nowak 2007.

Potential of purely data-driven, prior-free MMSE techniques remains under-exploited.

PURE: Poisson-Gaussian Unbiased Risk Estimate

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

$$\begin{aligned}\text{PURE} &= \frac{1}{N}\left(\|\mathbf{f}(\mathbf{y})\|^2 - 2\mathbf{y}^{\mathrm{T}}\mathbf{f}^-(\mathbf{y}) + 2\sigma^2\mathrm{div}\left\{\mathbf{f}^-(\mathbf{y})\right\}\right) + \\ &\quad \frac{1}{N}\left(\|\mathbf{y}\|^2 - \mathbf{1}^{\mathrm{T}}\mathbf{y}\right) - \sigma^2\end{aligned}$$

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

$$\mathscr{E}\{\text{PURE}\} = \frac{1}{N}\mathscr{E}\left\{\|\mathbf{f}(\mathbf{y}) - \mathbf{x}\|^2\right\}$$

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

PURE: Poisson-Gaussian Unbiased Risk Estimate

Sketch of proof: Need to estimate

$$\mathscr{E}\left\{\|\mathbf{f}(\mathbf{y}) - \mathbf{x}\|^2\right\} = \sum_n \left(\mathscr{E}\left\{f_n^2(\mathbf{y})\right\} - 2\underbrace{\mathscr{E}\left\{x_n f_n(\mathbf{y})\right\}}_{1} + \underbrace{x_n^2}_{2}\right)$$

1 Poisson's Lemma Hudson 1978, Tsui & Press 1982:

$$\begin{aligned}\mathscr{E}\{x_n f_n(\mathbf{y})\} &= \mathscr{E}\{x_n f_n(\mathbf{z} + \mathbf{b})\} \\ &= \mathscr{E}\{z_n f_n(\mathbf{z} + \mathbf{b} - \mathbf{e}_n)\}\end{aligned}$$

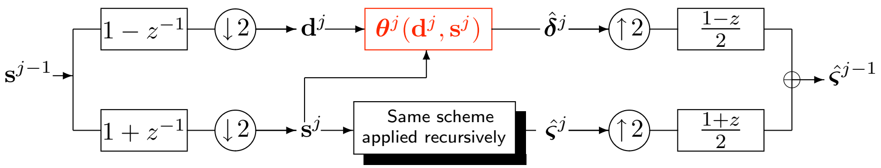
Stein's Lemma Stein 1981:

$$\begin{aligned}\mathscr{E}\{z_n f_n(\mathbf{z} + \mathbf{b} - \mathbf{e}_n)\} &= \mathscr{E}\{y_n f_n(\mathbf{y} - \mathbf{e}_n)\} - \mathscr{E}\{b_n f_n(\mathbf{z} + \mathbf{b} - \mathbf{e}_n)\} \\ &= \mathscr{E}\{y_n f_n(\mathbf{y} - \mathbf{e}_n)\} - \sigma^2\mathscr{E}\{\partial f_n(\mathbf{y} - \mathbf{e}_n)/\partial y_n\}\end{aligned}$$

2 Notice that: $x_n^2 = \mathscr{E}\{x_n y_n\} \overset{1}{=} \mathscr{E}\{y_n(y_n - 1)\} - \sigma^2$

The Unnormalized Haar Wavelet Transform

Denoising by **interscale** thresholding of the unnormalized **Haar-wavelet** coefficients: set $\mathbf{s}_0 = \mathbf{y}$, then for $j = 1, 2, \ldots, J$



Haar conservation properties:

- **Error energy:** $\text{MSE} = \dfrac{2^{-J}}{N}\|\hat{\boldsymbol{\varsigma}}^J - \boldsymbol{\varsigma}^J\|^2 + \displaystyle\sum_{j=1}^{J}\dfrac{2^{-j}}{N}\|\hat{\boldsymbol{\delta}}^j - \boldsymbol{\delta}^j\|^2$
- **Statistics:** $\mathbf{s}^j \sim \mathcal{P}(\boldsymbol{\varsigma}^j) + \mathcal{N}(\mathbf{0}, \sigma_j^2\mathbf{Id})$, where $\sigma_j^2 = 2^j\sigma^2$

Allows independent processing of each wavelet subband.

Interscale Haar-Wavelet-Domain PURE

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

$$\begin{cases}\theta_n^+(\mathbf{d}, \mathbf{s}) = \theta_n(\mathbf{d} + \mathbf{e}_n, \mathbf{s} - \mathbf{e}_n) \\ \theta_n^-(\mathbf{d}, \mathbf{s}) = \theta_n(\mathbf{d} - \mathbf{e}_n, \mathbf{s} - \mathbf{e}_n)\end{cases}$$

Then the random variable[1]

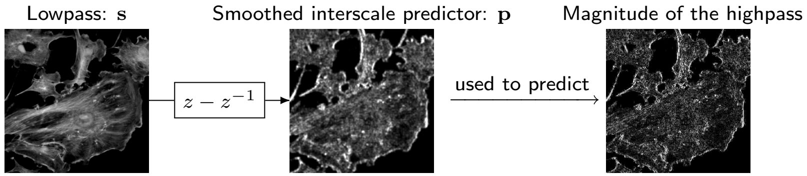
$$\begin{aligned}\text{PURE}_j = \frac{1}{N_j}\Big(&\|\boldsymbol{\theta}(\mathbf{d}, \mathbf{s})\|^2 + \|\mathbf{d}\|^2 - \mathbf{1}^{\mathrm{T}}\mathbf{s} - N_j\sigma_j^2 \\ &-\mathbf{d}^{\mathrm{T}}\left(\boldsymbol{\theta}^-(\mathbf{d}, \mathbf{s}) + \boldsymbol{\theta}^+(\mathbf{d}, \mathbf{s})\right) - \mathbf{s}^{\mathrm{T}}\left(\boldsymbol{\theta}^-(\mathbf{d}, \mathbf{s}) - \boldsymbol{\theta}^+(\mathbf{d}, \mathbf{s})\right) \\ &+\sigma_j^2\left(\mathrm{div}_{\mathbf{d}}\left\{\boldsymbol{\theta}^-(\mathbf{d}, \mathbf{s}) + \boldsymbol{\theta}^+(\mathbf{d}, \mathbf{s})\right\} + \mathrm{div}_{\mathbf{s}}\left\{\boldsymbol{\theta}^-(\mathbf{d}, \mathbf{s}) - \boldsymbol{\theta}^+(\mathbf{d}, \mathbf{s})\right\}\right)\Big)\end{aligned}$$

is an **unbiased estimate of the expected MSE** for the jth subband; i.e.,

$$\mathscr{E}\{\text{PURE}_j\} = \mathscr{E}\{\text{MSE}_j\}$$

[1] A similar result for pure Poisson noise can be found in Hirakawa *et al.* 2009.

Interscale Haar-Wavelet-Domain LET



$$\theta_n(\mathbf{d}_n, \mathbf{s}_n) = \underbrace{\gamma_n(p_n^2)\gamma_n(d_n^2)}_{\text{small predictor and small coefficient}} a_1 d_n + \underbrace{\overline{\gamma}_n(p_n^2)\gamma_n(d_n^2)}_{\text{large predictor and small coefficient}} a_2 d_n +$$

$$\underbrace{\gamma_n(p_n^2)\overline{\gamma}_n(d_n^2)}_{\text{small predictor and large coefficient}} a_3 d_n + \underbrace{\overline{\gamma}_n(p_n^2)\overline{\gamma}_n(d_n^2)}_{\text{large predictor and large coefficient}} a_4 d_n +$$

$$\underbrace{\gamma_n(p_n^2) a_5 \tilde{d}_n + \overline{\gamma}_n(p_n^2) a_6 \tilde{d}_n}_{\text{sign consistency enhancement}}$$

where $\gamma_n(x) = \mathrm{e}^{-\frac{|x|}{12(|s_n|+\sigma^2)}}$ and $\overline{\gamma}_n(x) = 1 - \gamma_n(x)$.

PURE for Arbitrary Nonlinear Processing

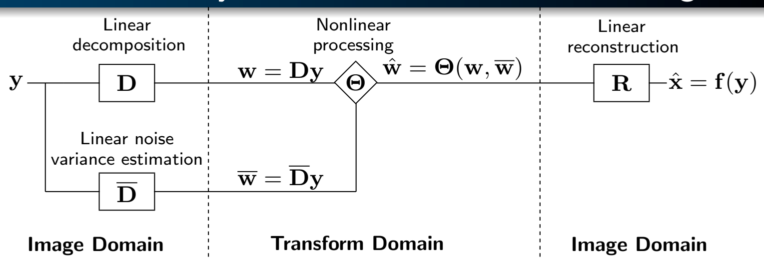
Problem: PURE is time-consuming to compute for an arbitrary nonlinear processing due to the term: $\mathbf{f}^{-}(\mathbf{y}) = [f_n(\mathbf{y} - \mathbf{e}_n)]_{1 \le n \le N}$.

Solution: First-order Taylor series approximation of $\mathbf{f}^{-}(\mathbf{y})$ given by $\mathbf{f}^{-}(\mathbf{y}) \simeq \mathbf{f}(\mathbf{y}) - \partial\mathbf{f}(\mathbf{y})$, where $\partial\mathbf{f}(\mathbf{y}) = [\frac{\partial f_n(\mathbf{y})}{\partial y_n}]_{1 \le n \le N}$.

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

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

PURE for Arbitrary Transform-Domain Processing

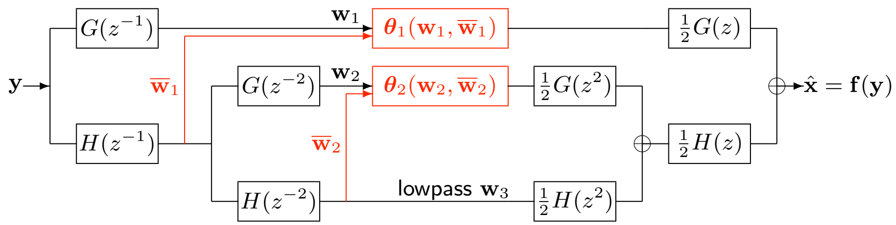


For *pointwise* processing $\boldsymbol{\Theta}(\mathbf{w}, \overline{\mathbf{w}}) = [\theta_l(w_l, \overline{w}_l)]_{1 \le l \le L}$, $\widehat{\text{PURE}}$ becomes:

$$\begin{aligned}\widehat{\text{PURE}} = {} & \frac{1}{N}\|\mathbf{f}(\mathbf{y}) - \mathbf{y}\|^2 + \frac{2}{N}\left(\boldsymbol{\Theta}_1(\mathbf{w}, \overline{\mathbf{w}})^{\mathrm{T}}(\mathbf{D} \bullet \mathbf{R}^{\mathrm{T}})\mathbf{y} + \boldsymbol{\Theta}_2(\mathbf{w}, \overline{\mathbf{w}})^{\mathrm{T}}(\overline{\mathbf{D}} \bullet \mathbf{R}^{\mathrm{T}})\mathbf{y}\right) + \\ & \frac{2\sigma^2}{N}\left(\mathrm{diag}\,\{\mathbf{D}\mathbf{R}\}^{\mathrm{T}}\boldsymbol{\Theta}_1(\mathbf{w}, \overline{\mathbf{w}}) + \mathrm{diag}\,\{\overline{\mathbf{D}}\mathbf{R}\}^{\mathrm{T}}\boldsymbol{\Theta}_2(\mathbf{w}, \overline{\mathbf{w}})\right) - \\ & \frac{2\sigma^2}{N}\Big(\mathrm{diag}\,\{(\mathbf{D} \bullet \mathbf{D})\mathbf{R}\}^{\mathrm{T}}\boldsymbol{\Theta}_{11}(\mathbf{w}, \overline{\mathbf{w}}) - \mathrm{diag}\,\{(\overline{\mathbf{D}} \bullet \overline{\mathbf{D}})\mathbf{R}\}^{\mathrm{T}}\boldsymbol{\Theta}_{22}(\mathbf{w}, \overline{\mathbf{w}}) - \\ & 2\,\mathrm{diag}\,\{(\mathbf{D} \bullet \overline{\mathbf{D}})\mathbf{R}\}^{\mathrm{T}}\boldsymbol{\Theta}_{12}(\mathbf{w}, \overline{\mathbf{w}})\Big) - \frac{1}{N}\mathbf{1}^{\mathrm{T}}\mathbf{y} - \sigma^2\end{aligned}$$

Example: Undecimated Haar Thresholding

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



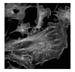
Subband-adaptive thresholding function:

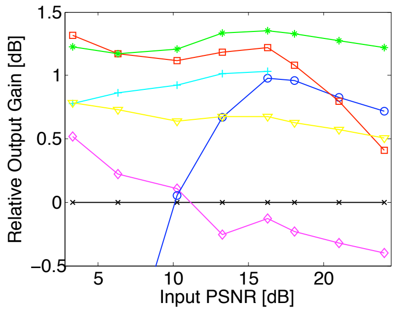
$$\theta_j(w, \overline{w}) = a_{j,1} \cdot w + a_{j,2} \cdot w \exp\left(-\left(\frac{w}{3t_j(\overline{w})}\right)^8\right)$$

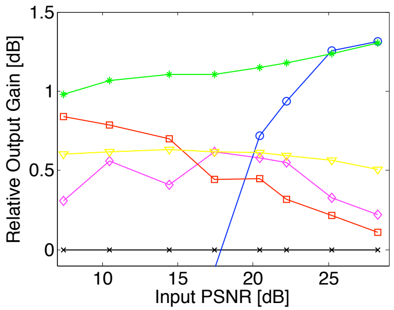
with signal-dependent threshold: $t_j(\overline{w}) = \sqrt{2^{-j/2}|\overline{w}| + \sigma^2}$

Some PSNR Comparisons

Cameraman 256×256 

Cells 512×512 



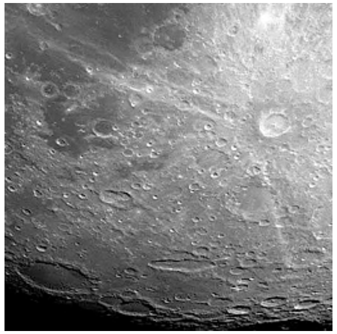


Haar PURE-LET (baseline)
Haar PURE-LET
(5 cycle-spins)
Redundant PURE-LET

Haar-Fisz Fryzlewicz & Nason 2004
Anscombe+BLS-GSM Portilla *et al.* 2003
Platelet Willett & Nowak 2007
PH-HMT Lefkimmiatis *et al.* 2009

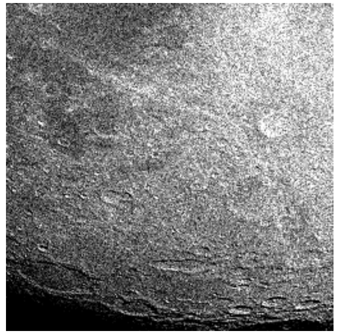
Some Visual Comparisons

Original



Average SSIM: 1.000

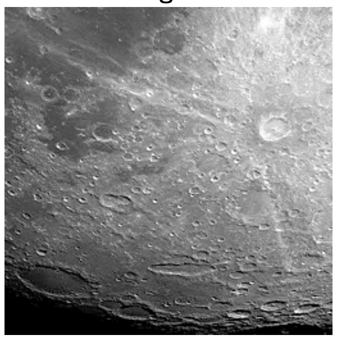
Noisy



Average SSIM: 0.385

Some Visual Comparisons

Original



Average SSIM: 1.000

Redundant PURE-LET

Average SSIM: 0.543

Some Visual Comparisons

Haar-Fisz

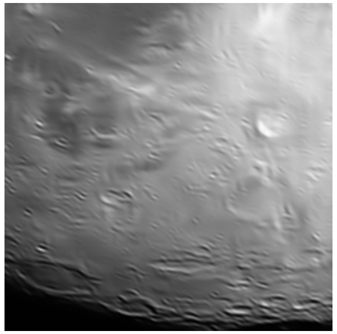
Average SSIM: 0.445

Redundant PURE-LET

Average SSIM: 0.543

Some Visual Comparisons

Anscombe+BLS-GSM



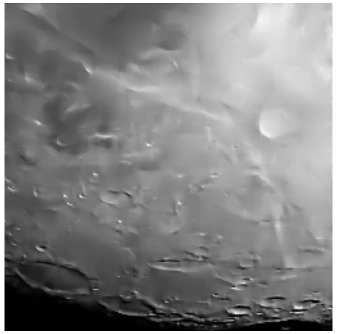
Average SSIM: 0.432

Redundant PURE-LET

Average SSIM: 0.543

Some Visual Comparisons

Platelet



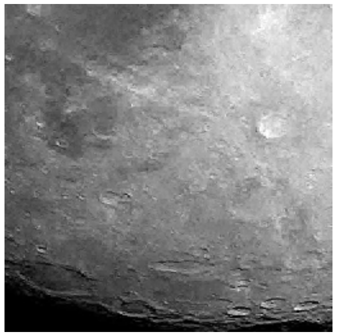
Average SSIM: 0.420

Redundant PURE-LET

Average SSIM: 0.543

Some Visual Comparisons

Haar PURE-LET



Average SSIM: 0.520

Redundant PURE-LET

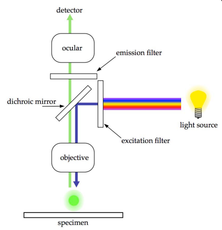
Average SSIM: 0.543

Fluorescence Microscopy

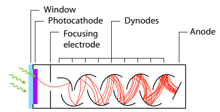
A fluorescence microscope is an imaging system that performs:

- 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.



Optical description



Photomultiplier tube

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).

$\rightsquigarrow$ **Measurement model:** scaled Poisson rdv degraded by AWGN

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

α: detector gain $\quad \mu$: detector offset $\quad \sigma^2$: AWGN variance

Noise Parameters Estimation

Affine relationship between sample-mean and sample-variance:

$$\left.\begin{array}{rclcl} \mu_y & \stackrel{\text{def}}{=} & \mathscr{E}\{y\} & = & \alpha x + \mu \\ \sigma_y^2 & \stackrel{\text{def}}{=} & \mathrm{Var}\{y\} & = & \alpha^2 x + \sigma^2 \end{array}\right\} \rightarrow \sigma_y^2 = \alpha\mu_y + \underbrace{\sigma^2 - \alpha\mu}_{\beta}$$

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

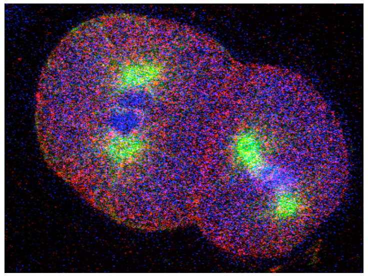
1. Compute μ_y and σ_y^2 in many small regions of the noisy image.
2. Perform a robust linear regression on the set of points (μ_y, σ_y^2).
3. Identify α as the slope of the fitted line and β as the ordinate at $\mu_y = 0$.
4. σ^2 and μ can be estimated independently in signal-free regions and cross-checked with β.

Experiments: 2D Sample

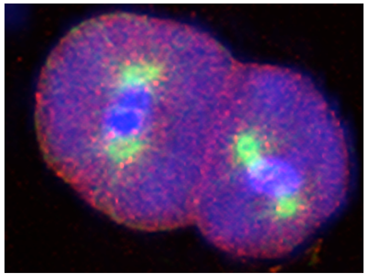
Specifications:

- 512×512 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

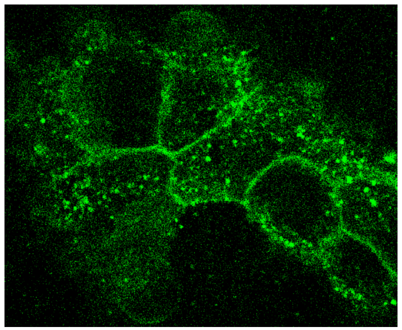


Experiments: 3D Sample

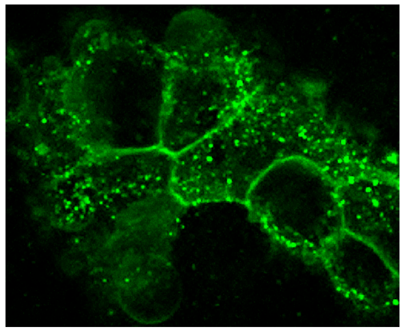
Specifications:

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

Raw Data



Multislice Haar PURE-LET

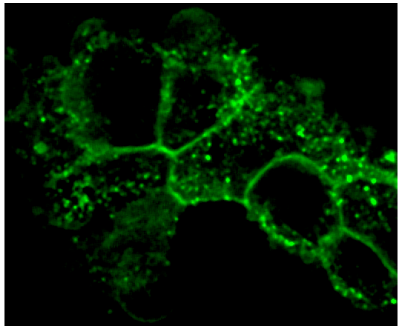


Experiments: 3D Sample

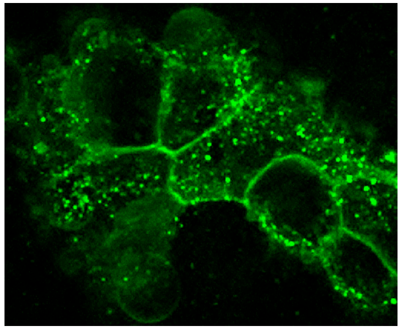
Specifications:

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

3D Median Filter



Multislice Haar PURE-LET

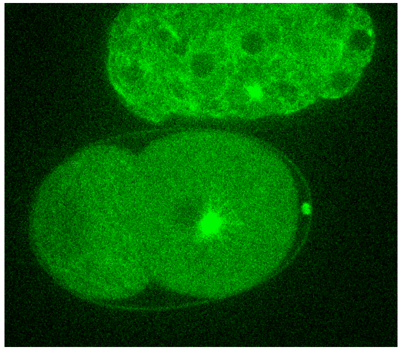


Experiments: 2D Timelapse Sequence

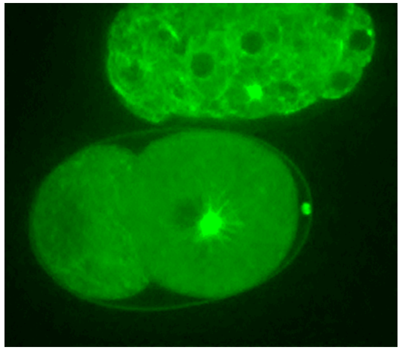
Specifications:

- $448 \times 512 \times 100$ 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

Luisier *et al.*, "Image Denoising in Mixed Poisson-Gaussian Noise", *IEEE Transactions on Image Processing*. To appear (2010).

Luisier *et al.*, "Fast Interscale Wavelet Denoising of Poisson-Corrupted Images", *Signal Processing*, Vol. 90 (2), pp. 415-427, February 2010.

Luisier *et al.*, "SURE-LET for Orthonormal Wavelet-Domain Video Denoising", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 20 (6), pp. 913-919, June 2010.

Luisier *et al.*, "SURE-LET Multichannel Image Denoising: Interscale Orthonormal Wavelet Thresholding", *IEEE Transactions on Image Processing*, Vol. 17 (4), pp. 482-492, April 2008.

Blu *et al.*, "The SURE-LET Approach to Image Denoising", *IEEE Transactions on Image Processing*, Vol. 16 (11), pp. 2778-2786, November 2007.

Luisier *et al.*, "A New SURE Approach to Image Denoising: Interscale Orthonormal Wavelet Thresholding", *IEEE Transactions on Image Processing*, Vol. 16 (3), pp. 593-606, March 2007. **Young Author Best Paper Award 2009**.

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