

Advanced Image Processing Based on Spatially Adaptive Nonlocal Image Filtering and Regularization

Vladimir Katkovnik and Alessandro Foi

Department of Signal Processing, Tampere University of Technology, Tampere, Finland

ICIP 2010, Hong Kong

Part I: Basics, Modeling and Algorithms

- 1 From local to nonlocal approximations;
- 2 Nonlocal means, block matching filtering;
- 3 High-order local and nonlocal models;
- 4 Block matching and collaborative filtering (BM3D algorithm);
- 5 Redundancy and multiple model nonlocal approximations;
- 6 Applications: denoising, color image denoising, deblurring, demosaicking;
- 7 Development of BM3D: shape-adaptive patches and adaptive PCA transforms.

Introduction

- In **local** image reconstruction algorithms usually use observations in a neighborhood of a pixel of the interest;

Introduction

- In **local** image reconstruction algorithms usually use observations in a neighborhood of a pixel of the interest;
- In the **nonlocal** techniques, algorithms analyze data “in large” and collects the observations from the whole image looking for similar features;

Introduction

- In **local** image reconstruction algorithms usually use observations in a neighborhood of a pixel of the interest;
- In the **nonlocal** techniques, algorithms analyze data “in large” and collect the observations from the whole image looking for similar features;
- The evolution of the nonlocal techniques from the simple nonlocal means (NL) to the transform domain processing is an evolution to higher-order models;

- In **local** image reconstruction algorithms usually use observations in a neighborhood of a pixel of the interest;
- In the **nonlocal** techniques, algorithms analyze data “in large” and collect the observations from the whole image looking for similar features;
- The evolution of the nonlocal techniques from the simple nonlocal means (NL) to the transform domain processing is an evolution to higher-order models;
- The latter algorithms are redundant, the data are processed by overlapping blocks, and multiple estimates obtained for each pixel are fused (aggregated) into the final image estimates.

Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, “From local kernel to nonlocal multiple-model image denoising”, *Int. J. Computer Vision*, vol. 86, no. 1, pp. 1-32, January 2010. doi:10.1007/s11263-009-0272-7

Table of Classification

	LOCAL	NONLOCAL
POINTWISE	<p>Section 2 (Local pointwise modeling)</p> <p>Signal-independent weights (Sects. 2.1–2.4): Nadaraya-Watson (Cleveland and Devlin 1988; Brown 1963; Savitzky and Golay 1964; Nadaraya 1964; Watson 1964), LPA (Fan and Gijbels 1996; Katkovnik 1976; Loader 1999), Lepski's approach (Lepski et al. 1997; Spokoiny 1998; Polzehl and Spokoiny 2003), LPA-ICI (Goldenshluger and Nemirovski 1997; Katkovnik 1999; Katkovnik et al. 2006; Foi 2005), sliding window transform (Yaroslavsky 1996; Yaroslavsky and Eden 1996)</p> <p>Signal-dependent weights (Sect. 2.5): Yaroslavsky filter (Yaroslavsky 1985), Sigma-filter (Lee 1983), Bilateral filter (Tomasi and Manduchi 1998), kernel regression (Takeda et al. 2007a)</p> <p>Variational formulations (Sect. 2.6): ROF (Rudin et al. 1992, 1993), Anisotropic diffusion (Perona and Malik 1990; Weickert 1996, 1998)</p>	<p>Section 4 (Nonlocal pointwise modeling)</p> <p>Weighted means (Sect. 4.1): neighborhood filter (Buades et al. 2005), NL-means algorithm (Buades et al. 2005), Adaptive Weights Smoothing (AWS) (Polzehl and Spokoiny 2000, 2005), Exemplar-based (Kervrann and Boulanger 2005, 2006, 2008), scale and rotation invariant (Lou et al. 2008a; Zimmer et al. 2008)</p> <p>Higher-order models (Sect. 4.2): NL-means with regression correction (Buades et al. 2006a), kernel regression (Chatterjee and Milanfar 2008)</p> <p>Variational formulations (Sect. 4.3): (Kindermann et al. 2005; Gilboa and Osher 2007a, 2007b; Lou et al. 2008a, 2008b; Elmoataz et al. 2008; Tschumperlé and Brun 2008a, 2008b)</p>
MULTIPOINT	<p>Section 3 (Local multipoint modeling)</p> <p>Overcomplete transform (Öktem et al. 1999, 2001; Egiazarian et al. 2001; Yaroslavsky et al. 2001; Guleryuz 2007; Hel-Or and Shaked 2008); shape-adaptive transform (Foi et al. 2007; Foi 2007); learned bases: adaptive PCA (Muresan and Parks 2003), MS-K-SVD (Mairal et al. 2008); TLS (Hirakawa and Parks 2006), BLS-GSM (Portilla et al. 2003), OAGSM-NC (Hammond and Simoncelli 2008)</p>	<p>Section 5 (Nonlocal multipoint modeling)</p> <p>Single-model approach (Sect. 5.1): Vectorial NL-means (Buades et al. 2005)</p> <p>Multiple-model approach (Sect. 5.2): BM3D (Dabov et al. 2007), Shape-Adaptive BM3D (Dabov et al. 2008), BM3D with Shape-Adaptive PCA (Dabov et al. 2009)</p>

Image denoising: observation model

Let us have independent random observation pairs $\{z_i, x_i\}_{i=1}^n$ given for simplicity in additive form

$$z_i = y_i + \varepsilon_i,$$

where $y_i = y(x_i)$ is a signal of interest, $x_i \in R^2$ and $\varepsilon_i = \varepsilon(x_i)$ is an additive noise.

The denoising problem is to reconstruct $y(x_i)$ from $\{z_i\}_{i=1}^n$.

Variational and heuristic approaches.

Image denoising: variational approach

The **variational approach** formalizes an image reconstruction as an optimization problem

$$\hat{y} = \operatorname{argmin}_y \underbrace{\frac{1}{\sigma^2} \|z - y\|^2}_{\text{fidelity}} + \mu \cdot \underbrace{\text{pen}(y)}_{\text{penalty}}, \quad \mu > 0.$$

The fidelity term follows from a statistical noise model and the penalty is a prior for y .

Parametric and nonparametric formulations.

Typical parametric models are of the form

$$y(x) = \sum_k c_k \phi_k(x).$$

Image denoising: heuristic approach (local means)

- The weighted least square method gives the following criterion

$$J(x^0, C) = \sum_s w_h(x^0 - x_s) e_s^2, \quad e_s = z_s - C_0,$$

where w is a signal-independent window function,

$$w_h(x) = w(x/h),$$

and C_0 is an approximation of $y(x)$ at $x = x^0$.

Image denoising: heuristic approach (local means)

- The weighted least square method gives the following criterion

$$J(x^0, C) = \sum_s w_h(x^0 - x_s) e_s^2, \quad e_s = z_s - C_0,$$

where w is a signal-independent window function,

$$w_h(x) = w(x/h),$$

and C_0 is an approximation of $y(x)$ at $x = x^0$.

- Minimizing J on C_0 :

$$\hat{C}_0(x) = \arg \min_{C_0} J(x^0, C_0) \implies \hat{y}_h(x^0) = \hat{C}_0(x) = \frac{\sum_s w_h(x^0 - x_s) z_s}{\sum_s w_h(x^0 - x_s)}.$$

Image denoising: heuristic approach (local means)

- The weighted least square method gives the following criterion

$$J(x^0, C) = \sum_s w_h(x^0 - x_s) e_s^2, \quad e_s = z_s - C_0,$$

where w is a signal-independent window function,

$$w_h(x) = w(x/h),$$

and C_0 is an approximation of $y(x)$ at $x = x^0$.

- Minimizing J on C_0 :

$$\hat{C}_0(x) = \arg \min_{C_0} J(x^0, C_0) \implies \hat{y}_h(x^0) = \hat{C}_0(x) = \frac{\sum_s w_h(x^0 - x_s) z_s}{\sum_s w_h(x^0 - x_s)}.$$

- Example of signal-independent weight

$$w_h(x^0 - x_s) = e^{-\frac{\|x^0 - x_s\|^2}{h^2}}, \quad h > 0.$$

Image denoising: heuristic approach (local means)

- The weighted least square method gives the following criterion

$$J(x^0, C) = \sum_s w_h(x^0 - x_s) e_s^2, \quad e_s = z_s - C_0,$$

where w is a signal-independent window function,

$$w_h(x) = w(x/h),$$

and C_0 is an approximation of $y(x)$ at $x = x^0$.

- Minimizing J on C_0 :

$$\hat{C}_0(x) = \arg \min_{C_0} J(x^0, C_0) \implies \hat{y}_h(x^0) = \hat{C}_0(x) = \frac{\sum_s w_h(x^0 - x_s) z_s}{\sum_s w_h(x^0 - x_s)}.$$

- Example of signal-independent weight

$$w_h(x^0 - x_s) = e^{-\frac{\|x^0 - x_s\|^2}{h^2}}, \quad h > 0.$$

- Example of signal-dependent weight

$$w_h(x^0 - x_s, y^0 - y_s) = e^{-\frac{\|x^0 - x_s\|^2}{h^2} - \frac{|y^0 - y_s|^2}{\gamma}}, \quad \gamma, h > 0.$$

- Criterion

$$J_{h,x^0}(C) = \sum_s w_h(y^0 - y_s)[z_s - C_0]^2, \quad y^0 = y(x^0),$$

where the weights w_h depend on the distance between the signal values at the observation points y_s and the desirable point $y^0 = y(x^0)$.

Image denoising: heuristic approach (nonlocal means)

- Criterion

$$J_{h,x^0}(C) = \sum_s w_h(y^0 - y_s) [z_s - C_0]^2, \quad y^0 = y(x^0),$$

where the weights w_h depend on the distance between the signal values at the observation points y_s and the desirable point $y^0 = y(x^0)$.

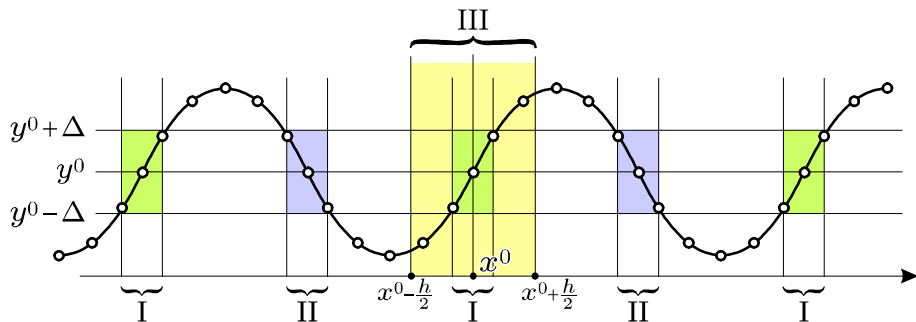
- Minimization, $\min_C J_{h,x^0}(C)$, gives the nonlocal means estimate

$$\hat{y}_h(x^0) = \hat{C}_0(x^0) = \frac{1}{\sum_s w_h(y^0 - y_s)} \sum_s w_h(y^0 - y_s) z_s,$$

where $y^0 = y(x^0)$.

Local versus Nonlocal Supports

Local versus nonlocal supports for zero and first order polynomial fitting: local *III*; nonlocal zero-order model *I* \cup *II*; nonlocal first-order model *I*.



Local Pointwise Approximations

Three key slogans associated with these techniques:

- *Locality*;
- *Anisotropy*;
- *Adaptivity*.

The *locality* means that there is a neighborhood where the image intensity is well approximated some continuous basis functions.

The *anisotropy* means that a good local approximation can be achieved only in a non-symmetric neighborhood.

The *adaptivity* means that both the size and the shape should be data adaptive.

Local Polynomial Approximation

- In a neighborhood of x^0 the Taylor series gives for $y(x_s)$:

$$y(x_s) \simeq y(x^0) - y_{x_1}^{(1)}(x^0)(x_1^0 - x_{1,s}) - y_{x_2}^{(1)}(x^0)(x_2^0 - x_{2,s}) + \\ y_{x_1, x_2}^{(2)}(x^0)(x_1^0 - x_{1,s})(x_2^0 - x_{2,s}) + \frac{1}{2}y_{x_1}^{(2)}(x^0)(x_1^0 - x_{1,s})^2 + \\ \frac{1}{2}y_{x_2}^{(2)}(x^0)(x_2^0 - x_{2,s})^2 \dots$$

- Since the function y and the derivatives are unknown we look for fitting data $y(x_s)$ in the form

$$y(x^0, x_s) \simeq C_0 - C_1(x_1^0 - x_{1,s}) - C_2(x_2^0 - x_{2,s}) + \\ C_{12}(x_1^0 - x_{1,s})(x_2^0 - x_{2,s}) + \frac{1}{2}C_{11}(x_1^0 - x_{1,s})^2 + \frac{1}{2}C_{22}(x_2^0 - x_{2,s})^2 \dots$$

where the coefficients C_0 , C_1 and C_2 give estimates for $y(x^0)$, $y_{x_1}^{(1)}(x^0)$ and $y_{x_2}^{(1)}(x^0)$, and C_{12} , C_{11} , C_{22} give estimates for $y_{x_1, x_2}^{(2)}(x^0)$, $y_{x_1}^{(2)}(x^0)$ and $y_{x_2}^{(2)}(x^0)$.

How to formalize a local fit?

- The weighted least square method gives the following criterion

$$J(x^0, C) = \sum_s w_h(x^0 - x_s) e_s^2,$$
$$e_s = y_s - y(x^0, x_s),$$

where w is a window function, $w_h(x) = w(x/h)$, and $C = (C_0, C_1, \dots)^T$.

- Minimizing J on C :

$$\hat{C}(x^0) = \arg \min_C J(x^0, C).$$

The notation $\hat{C}(x^0)$ emphasizes that the estimate of C depends on x^0 .

The vector parameter $\hat{C}(x^0)$ immediately gives the estimates of the function y and the derivatives $y^{(r)}$:

$$\begin{aligned}\hat{y}(x^0) &= \hat{C}_0(x^0), \\ \hat{y}_{x_1}^{(1)}(x^0) &= \hat{C}_1(x^0), \\ \hat{y}_{x_2}^{(1)}(x^0) &= \hat{C}_2(x^0), \\ &\dots\dots\end{aligned}$$

The conventional windows can be used: *Kaiser*, *Hamming*, *Bartlett*, *Blackman*, *Chebyshev*, etc.

Using the standard notation for multidimensional convolution, the estimates can be represented in the following compact form

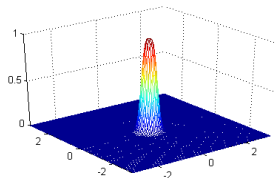
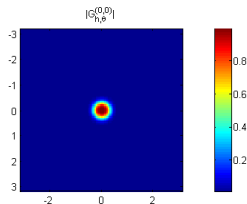
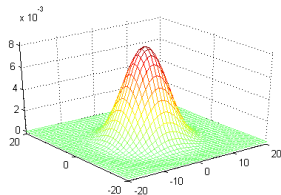
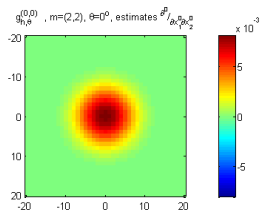
$$\hat{y}_h(x^0) = (g_h \circledast z)(x^0) = \sum_{x_{1,s}, x_{2,s}} g_h(x_1^0 - x_{1,s}, x_2^0 - x_{2,s}) z(x_{1,s}, x_{2,s}),$$

$$\hat{y}_h^{(r)}(x^0) = (g_h^{(r)} \circledast z)(x^0), x^0 \in X$$

2D examples (demo_CreateLPAKernels.m)

The smoothing kernel g_h and its amplitude frequency characteristic $|G_h|$:
Gaussian window, $m = [2, 2]$.

The lowpass filter with a peak of the frequency characteristic at $\bar{\omega} = 0$.

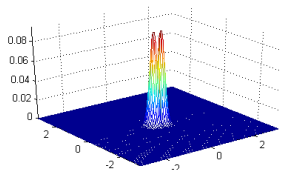
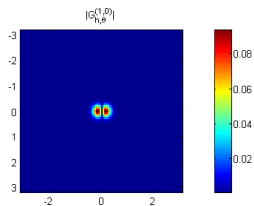
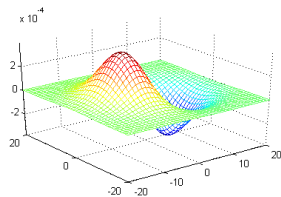
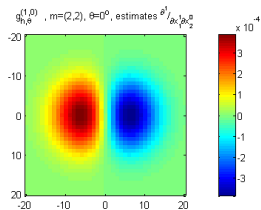


2D examples (cont.)

The differentiation kernel $g_h^{(1,0)}$ and its amplitude frequency characteristic

$|G_h^{(1,0)}|$: Gaussian window, $m = [2, 2]$.

The bandpass filter $|G_h^{(1,0)}| = 0$ at $\bar{\omega} = 0$.

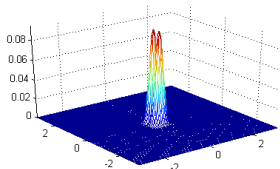
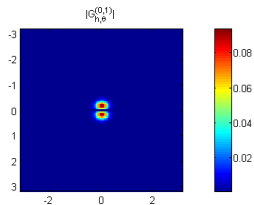
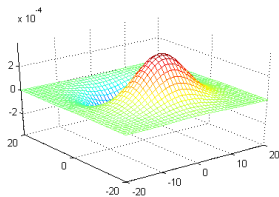
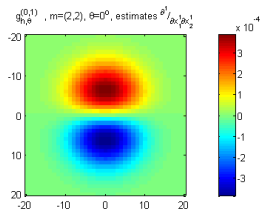


2D examples (cont.)

The differentiation kernel $g_h^{(0,1)}$ and its amplitude frequency characteristic

$|G_h^{(0,1)}|$: Gaussian window, $m = [2, 2]$.

The bandpass filter $|G_h^{(0,1)}| = 0$ at $\bar{\omega} = 0$.

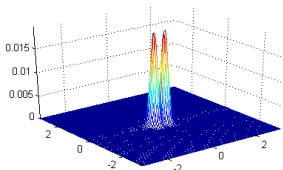
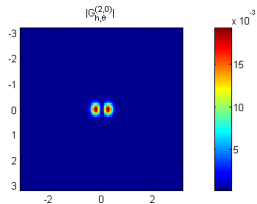
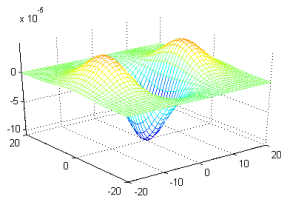
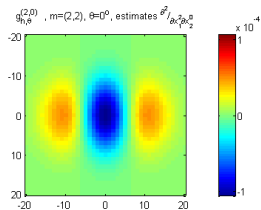


2D examples (cont.)

The differentiation kernel $g_h^{(2,0)}$ and its amplitude frequency characteristic

$|G_h^{(2,0)}|$: Gaussian window, $m = [2, 2]$.

The bandpass filter $|G_h^{(2,0)}| = 0$ at $\bar{\omega} = 0$.

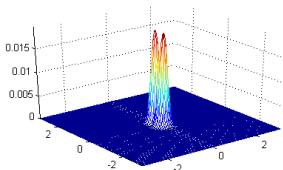
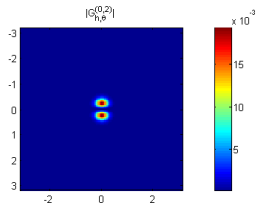
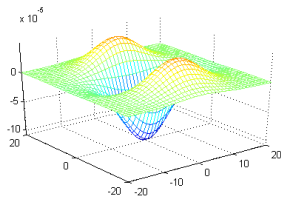
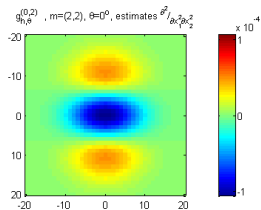


2D examples (cont.)

The differentiation kernel $g_h^{(0,2)}$ and its amplitude frequency characteristic

$|G_h^{(0,2)}|$: Gaussian window, $m = [2, 2]$.

The bandpass filter $|G_h^{(0,2)}| = 0$ at $\bar{\omega} = 0$.

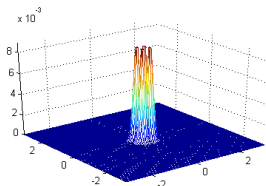
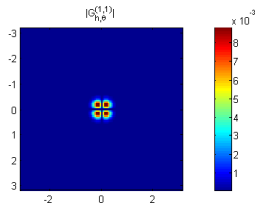
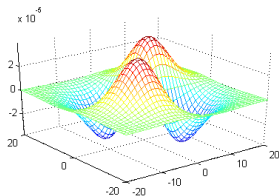
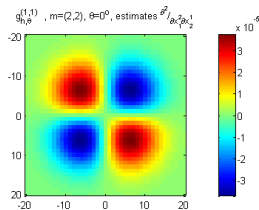


2D examples

The differentiation kernel $g_h^{(1,1)}$ and its amplitude frequency characteristic

$|G_h^{(1,1)}|$: Gaussian window, $m = [2, 2]$.

The bandpass filter $|G_h^{(1,1)}| = 0$ at $\bar{\omega} = 0$.



We use pointwise nonparametric regression methods known under a generic name Lepski's approach and developed by Lepski O., Nemirovski A., Goldenshluger A., Spokoiny V.

- Overall, the algorithm searches for the largest local vicinity of the point of estimation x where the LPA assumptions fit well to the data.*

We use pointwise nonparametric regression methods known under a generic name Lepski's approach and developed by Lepski O., Nemirovski A., Goldenshluger A., Spokoiny V.

- *Overall, the algorithm searches for the largest local vicinity of the point of estimation x where the LPA assumptions fit well to the data.*
- *The estimates are calculated for a few scales and compared.*

We use pointwise nonparametric regression methods known under a generic name Lepski's approach and developed by Lepski O., Nemirovski A., Goldenshluger A., Spokoiny V.

- *Overall, the algorithm searches for the largest local vicinity of the point of estimation x where the LPA assumptions fit well to the data.*
- *The estimates are calculated for a few scales and compared.*
- *The adaptive scale is defined as the largest one of those for which the estimate does not differ significantly from the estimates corresponding to the smaller scales.*

We use pointwise nonparametric regression methods known under a generic name Lepski's approach and developed by Lepski O., Nemirovski A., Goldenshluger A., Spokoiny V.

- *Overall, the algorithm searches for the largest local vicinity of the point of estimation x where the LPA assumptions fit well to the data.*
- *The estimates are calculated for a few scales and compared.*
- *The adaptive scale is defined as the largest one of those for which the estimate does not differ significantly from the estimates corresponding to the smaller scales.*
- *We use this methods in the form known as the intersection of confidence intervals (ICI) rule.*

$$e_{\hat{y}_h(x)} = y(x) - \hat{y}_h(x) = y(x) - (g_h \circledast (y + \varepsilon))(x) =$$

$$\underbrace{y(x) - (g_h \circledast y)(x)}_{\text{bias}} - \underbrace{(g_h \circledast \varepsilon)(x)}_{\text{random error}}.$$

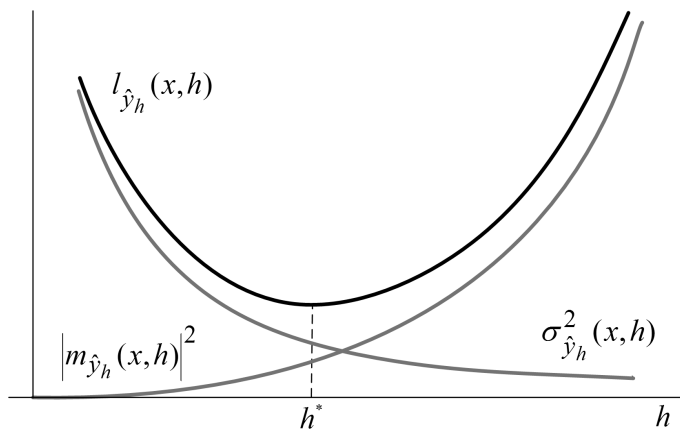
The variance

$$\sigma_{\hat{y}_h}^2 = \sigma^2 \sum_s g_h^2(x_s).$$

The *ICI* rule gives h minimizing *MSE*

$$I_{\hat{y}_h} = E\{e_{\hat{y}_h(x)}^2\} = \text{bias}^2 + \sigma_{\hat{y}_h}^2.$$

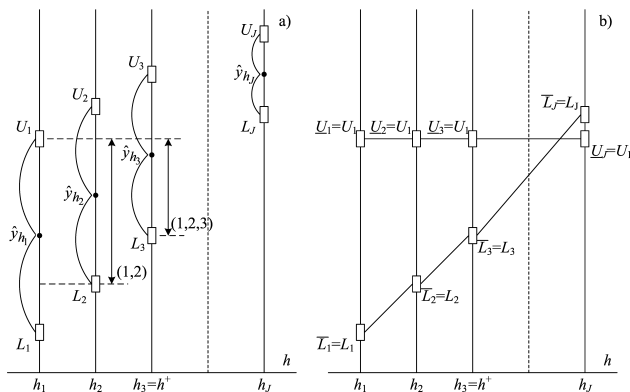
ICI rule (minimum MSE window size)



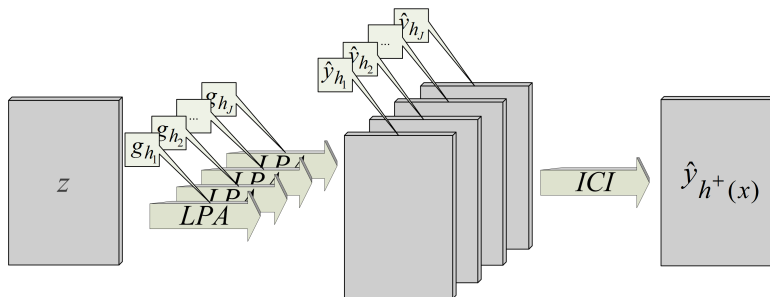
ICI rule (minimum MSE window size)

$$H = \{h_1 < h_2 < \dots < h_J\}.$$

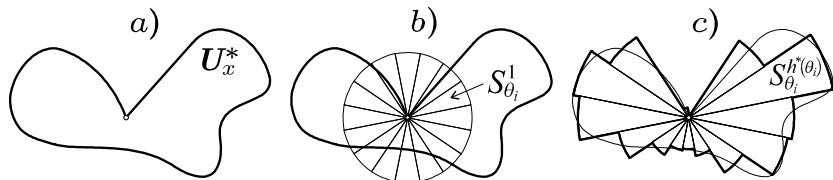
$$D_j = [\hat{y}_{h_j}(x) - \Gamma \cdot \sigma_{\hat{y}_h}(x), \hat{y}_{h_j}(x) + \Gamma \cdot \sigma_{\hat{y}_h}(x)], \quad j = 1, \dots, J,$$



A layout of the adaptive scale LPA-ICI algorithm



Anisotropy: starshaped neighborhood



Consider a ball (disk) of the radius h defining a spherical neighborhood of x ,

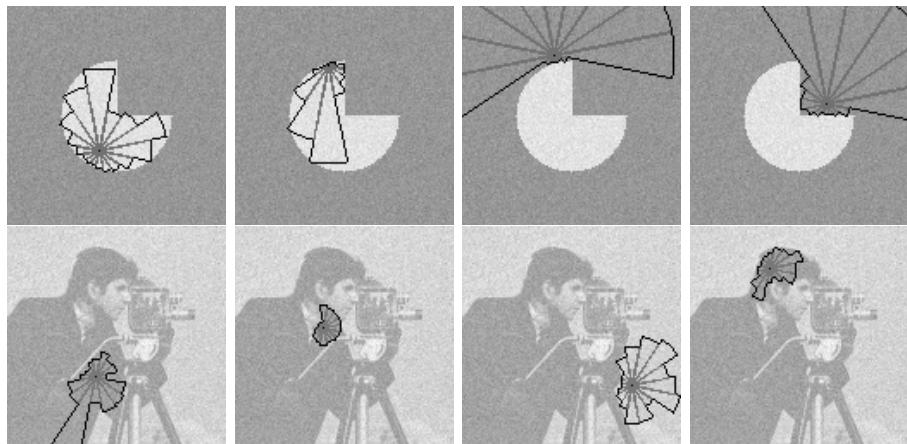
$$\mathcal{B}_h = \{u : \|x - u\| \leq h\}.$$

Introduce a sectorial partition of this ball with K nonoverlapping sectors having x as a common vertex.

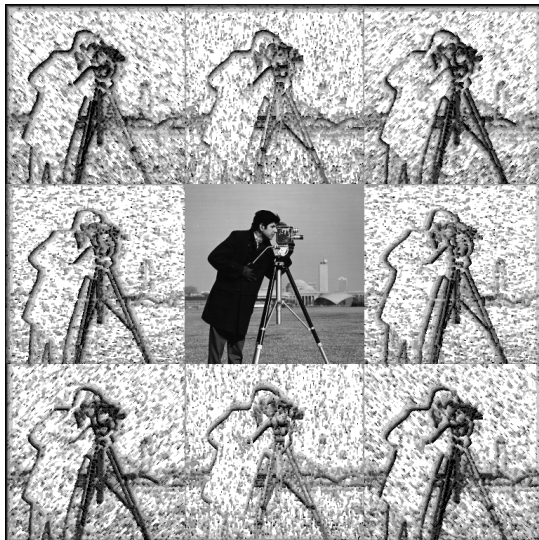
The adaptivity *ICI* technique is used to find the varying scale for each sector independently. We obtain K estimates $\hat{y}_{h,\theta_i}(x)$, $i = 1, \dots, K$, with K nonoverlapping supports $S_{\theta_i}^h$ covering the ball \mathcal{B}_h .

The union of the supports of these sectors is a *starshaped* set.

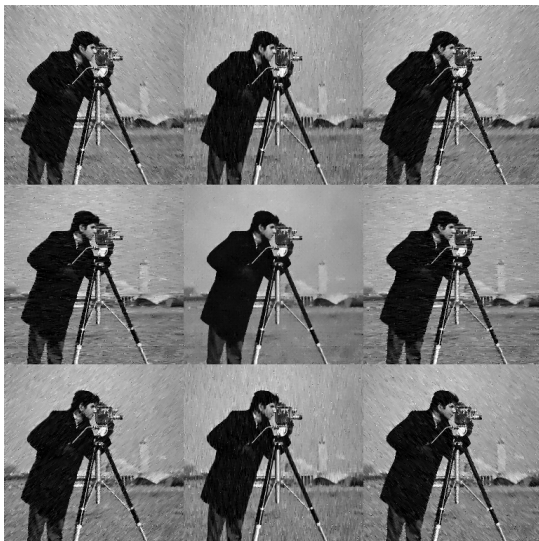
Adaptive size sectorial neighborhoods are obtained by the LPA-ICI algorithm



These are ICI adaptive directional window sizes (scales) computed for 8 directions



These ICI adaptive directional estimates are fused in the final one using the anisotropic multi-window method



Fusing (aggregation) of directional estimates

Using K sectors we obtain K independently derived estimates for each x . The *multi – window (fused)* estimates are exploited to obtain the unique final estimate $\hat{y}(x)$ from the partial directional ones.

With the inverse variances as the weights for the linear fusing it gives

$$\hat{y}(x) = \sum_{j=1}^K \lambda_j \hat{y}_{\theta_j}(x), \quad \hat{y}_{\theta_j}(x) = \hat{y}_{h,\theta_j}(x)|_{h=h^+(x,\theta_j)},$$
$$\lambda_j = \sigma_j^{-2}(x) / \sum_{i=1}^K \sigma_i^{-2}(x),$$

where

$$\hat{y}_{h,\theta_j}(x) = (g_{h,\theta_j} \circledast z)(x), \quad \sigma_j^{-2}(x) = \sigma_{\hat{y},\theta_j}^2(x, h)|_{h=h^+(x,\theta_j)},$$
$$\sigma_{\hat{y},\theta_j}^2(x, h) = \sigma^2 \sum_x g_{h,\theta_j}^2(x).$$

Fusing (aggregation) of directional estimates

The weights λ_j are data-driven adaptive as $\sigma_j^{-2}(x)$ depend on the adaptive pointwise $h^+(x, \theta_j)$.

Assuming that the supports of the kernels $g_{\theta_j, h}$ are not overlapping and neglecting that the kernels have the point x in common, we obtain for the variance of the fused estimate

$$\sigma_{\hat{y}}^2(x) = \sum_{j=1}^K \lambda_j^2 \sigma_j^2(x) = \frac{1}{\sum_{j=1}^K \sigma_j^{-2}(x)}.$$

Shape-Adaptive DCT Filter (Local Multipoint)

This approach to estimation for a point x can be roughly described as the following four stage procedure.

- Stage I (spatial adaptation): For every $x \in X$, define a neighborhood \tilde{U}_x^+ of x where a simple low-order polynomial model fits the data;

Shape-Adaptive DCT Filter (Local Multipoint)

This approach to estimation for a point x can be roughly described as the following four stage procedure.

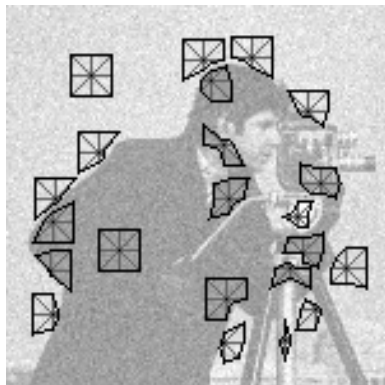
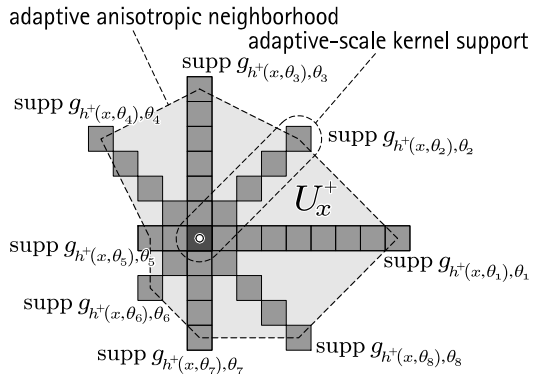
- Stage I (spatial adaptation): For every $x \in X$, define a neighborhood \tilde{U}_x^+ of x where a simple low-order polynomial model fits the data;
- Stage II (order selection): apply some localized transform (parametric series model) to the data on the set \tilde{U}_x^+ , use thresholding operator (model selection procedure) in order to identify the significant (i.e. nonzero) elements of the transform (and thus the order of the parametric model).

- Stage III (approximation): Calculate, by inverse-transformation of the significant elements only, the corresponding estimates $\hat{y}_{\tilde{U}_x^+}(v)$ of the signal for all $v \in \tilde{U}_x^+$. These $\hat{y}_{\tilde{U}_x}$ are calculated for all $x \in X$.

- Stage III (approximation): Calculate, by inverse-transformation of the significant elements only, the corresponding estimates $\hat{y}_{\tilde{U}_x^+}(v)$ of the signal for all $v \in \tilde{U}_x^+$. These $\hat{y}_{\tilde{U}_x}$ are calculated for all $x \in X$.
- Stage IV (aggregation): Let $x \in X$ and $I_x = \{x \in X : x \in \tilde{U}_x^+\}$ be the set of the centers of the neighborhoods which have x as a common point. The final estimate $\hat{y}(x)$ is calculated as an aggregate of $\{\hat{y}_{\tilde{U}_x^+}(x)\}_{x \in I_x}$.

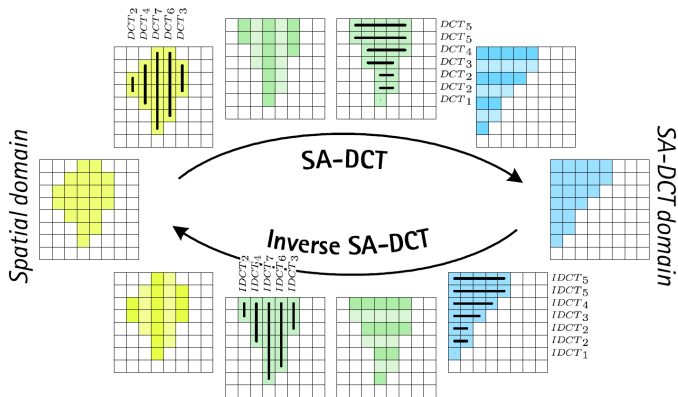
- Stage III (approximation): Calculate, by inverse-transformation of the significant elements only, the corresponding estimates $\hat{y}_{\tilde{U}_x^+}(v)$ of the signal for all $v \in \tilde{U}_x^+$. These $\hat{y}_{\tilde{U}_x}$ are calculated for all $x \in X$.
- Stage IV (aggregation): Let $x \in X$ and $I_x = \{x \in X : x \in \tilde{U}_x^+\}$ be the set of the centers of the neighborhoods which have x as a common point. The final estimate $\hat{y}(x)$ is calculated as an aggregate of $\{\hat{y}_{\tilde{U}_x^+}(x)\}_{x \in I_x}$.
- One key aspect in this procedure is that by demanding the local fit of a low-order polynomial model, we are able to avoid the presence of singularities, discontinuities, or sharp transitions within the transform support \tilde{U}_x^+ . In this way, we increase the sparsity in the transform domain, improving the effectiveness of thresholding.

Illustrations



Shape-Adaptive DCT Transform

Illustration of the shape-adaptive DCT transform and its inverse. Transformation is computed by cascaded application of one-dimensional varying-length DCT transforms, along the columns and along the rows.



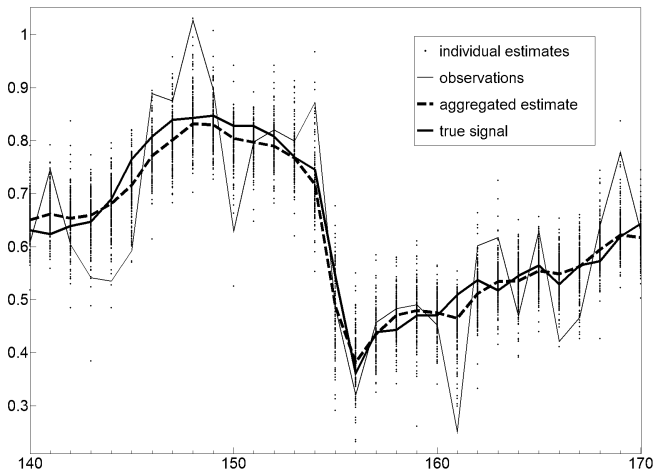
Multipoint Estimate Fusing (fusing-aggregation)

In order to obtain a single global estimate $\hat{y} : X \rightarrow R$ defined on the whole image domain, all the local patch estimates are averaged together using adaptive weights $w_x \in R$ in the following convex combination:

$$\hat{y} = \frac{\sum_{x \in X} w_x \hat{y}_{U_x^+}}{\sum_{x \in X} w_x \chi_{U_x^+}}.$$

Aggregation (multipoint) in action

A cross-section of length 31 pixels from Peppers test-image ($\sigma = 25$).



Implementation of SA-DCT Filter

<http://www.cs.tut.fi/~foi/SA-DCT/>

- demo_SADCT_denoising.m
- demo_SADCT_color_denoising.m
- demo_SADCT_deblurring.m
- demo_SADCT_deblocking.m
- demo_SADCT_inverse_half_toning.m

A. Buades, et al., "A review of image denoising algorithms, with a new one,"
SIAM Multiscale Modeling and Simulation, vol. 4, 2005.

- The nonlocal means (NL-means) as they are introduced by Buades et al. (2005) have been given in a different form where these weights calculated over spatial neighborhoods of the points x^0 and x_s .

A. Buades, et al., "A review of image denoising algorithms, with a new one," SIAM Multiscale Modeling and Simulation, vol. 4, 2005.

- The nonlocal means (NL-means) as they are introduced by Buades et al. (2005) have been given in a different form where these weights calculated over spatial neighborhoods of the points x^0 and x_s .
- These neighborhood-wise differences can be interpreted as more reliable way to estimate $y^0 - y_s$ from the noise samples alone.

A. Buades, et al., "A review of image denoising algorithms, with a new one," SIAM Multiscale Modeling and Simulation, vol. 4, 2005.

- The nonlocal means (NL-means) as they are introduced by Buades et al. (2005) have been given in a different form where these weights calculated over spatial neighborhoods of the points x^0 and x_s .
- These neighborhood-wise differences can be interpreted as more reliable way to estimate $y^0 - y_s$ from the noise samples alone.
- Then, the nonlocal mean estimate is calculated in a pointwise manner as the weighted mean with the weights defined by the proximity measure between the image patches used in the estimate.

NL-means filters (cont.)

- This estimation can be formalized as minimization of the weighted squared residual criterion

$$I_{h,x}(C) = \sum_s w_{h,s}(x^0, x_s) [z_s - C]^2,$$

with, say, Gaussian weights

$$w_{h,s}(x^0, x_s) = e^{-\frac{\sum_{v \in V} (z(x^0+v) - z(x_s+v))^2}{h^2}}$$

defined by the Euclidean distance between the observations z in V -neighborhoods of the points x^0 and x_s , V being a fixed neighborhood of x^0 .

NL-means filters (cont.)

- This estimation can be formalized as minimization of the weighted squared residual criterion

$$I_{h,x}(C) = \sum_s w_{h,s}(x^0, x_s) [z_s - C]^2,$$

with, say, Gaussian weights

$$w_{h,s}(x^0, x_s) = e^{-\frac{\sum_{v \in V} (z(x^0+v) - z(x_s+v))^2}{h^2}}$$

defined by the Euclidean distance between the observations z in V -neighborhoods of the points x^0 and x_s , V being a fixed neighborhood of x^0 .

- The nonlocal means estimate is calculated as the weighted mean

$$\hat{y}_h(x^0) = \sum_s g_{h,s}(x^0) z_s, \quad g_{h,s}(x^0) = \frac{w_{h,s}(x^0, x_s)}{\sum_s w_{h,s}(x^0, x_s)}.$$

We consider nonlocal estimates by use of the transforms enabling the adaptive high-order approximations of the windowed data.

- Let the image be defined on a regular $2D$ grid X . Consider a windowing $C = \{X_r, r = 1, \dots, N_s\}$ of X with N_s blocks (uniform windows) $X_r \subset X$ of size $n_r \times n_r$ such that $\cup_{r=1}^{N_s} X_r = X$.
The noise-free data $y(x)$ and the noisy data $z(x)$ windowed on X_r are arranged in $n_r \times n_r$ blocks denoted as Y_r and Z_r , respectively.

We consider nonlocal estimates by use of the transforms enabling the adaptive high-order approximations of the windowed data.

- Let the image be defined on a regular $2D$ grid X . Consider a windowing $C = \{X_r, r = 1, \dots, N_s\}$ of X with N_s blocks (uniform windows) $X_r \subset X$ of size $n_r \times n_r$ such that $\cup_{r=1}^{N_s} X_r = X$.
The noise-free data $y(x)$ and the noisy data $z(x)$ windowed on X_r are arranged in $n_r \times n_r$ blocks denoted as Y_r and Z_r , respectively.
- Typically, the blocks may overlap and we use transforms in conjunction with the concept of the redundancy of natural signals.

Nonlocal transform domain (cont.)

- Mainly these are the 2-D discrete Fourier and cosine transforms (DFT and DCT), orthogonal polynomials, and wavelet transforms. The transform, denoted as \mathcal{T}_r^{2D} , is applied for each window X_r independently as

$$\theta_r = \mathcal{T}_r^{2D} (Y_r), \quad \left[\theta_r = D_r Y_r D_r^T \right] \quad r = 1, \dots, N_s,$$

where θ_r is the spectrum of Y_r , and D_r are orthonormal matrices

$$D_r D_r^T = D_r^T D_r = I.$$

Nonlocal transform domain (cont.)

- Mainly these are the 2-D discrete Fourier and cosine transforms (DFT and DCT), orthogonal polynomials, and wavelet transforms. The transform, denoted as \mathcal{T}_r^{2D} , is applied for each window X_r independently as

$$\theta_r = \mathcal{T}_r^{2D} (Y_r), \quad \left[\theta_r = D_r Y_r D_r^T \right] \quad r = 1, \dots, N_s,$$

where θ_r is the spectrum of Y_r , and D_r are orthonormal matrices

$$D_r D_r^T = D_r^T D_r = I.$$

- The equality enclosed in square brackets holds when the transform \mathcal{T}_r^{2D} is realized as a separable composition of 1-D transforms, each computed by matrix multiplication against an $n_r \times n_r$ orthogonal matrix D_r .

- The inverse \mathcal{T}_r^{2D-1} of \mathcal{T}_r^{2D} defines the signal from the spectrum as

$$Y_r = \mathcal{T}_r^{2D-1}(\theta_r), \quad \left[Y_r = D_r^T \theta_r D_r \right] \quad r = 1, \dots, N_s.$$

- The inverse \mathcal{T}_r^{2D-1} of \mathcal{T}_r^{2D} defines the signal from the spectrum as

$$Y_r = \mathcal{T}_r^{2D-1}(\theta_r), \quad \left[Y_r = D_r^T \theta_r D_r \right] \quad r = 1, \dots, N_s.$$

- The noisy spectrum of the noisy signal is defined as

$$\tilde{\theta}_r = \mathcal{T}_r^{2D}(Z_r), \quad \left[\tilde{\theta}_r = D_r Z_r D_r^T \right] \quad r = 1, \dots, N_s.$$

- The inverse \mathcal{T}_r^{2D-1} of \mathcal{T}_r^{2D} defines the signal from the spectrum as

$$Y_r = \mathcal{T}_r^{2D-1}(\theta_r), \quad \left[Y_r = D_r^T \theta_r D_r \right] \quad r = 1, \dots, N_s.$$

- The noisy spectrum of the noisy signal is defined as

$$\tilde{\theta}_r = \mathcal{T}_r^{2D}(Z_r), \quad \left[\tilde{\theta}_r = D_r Z_r D_r^T \right] \quad r = 1, \dots, N_s.$$

- The signal y is sparse if it can be well approximated by a small number of non-zero elements of the spectrum θ_r .

Nonlocal transform-domain calculation: single model

Let Z_r be a reference window and Z_s be all others.

The non-local estimate for r -th window is formalized as minimization of the weighted criterion

$$I_{h,Z_r}(\theta) = \sum_s w_h(Z_r, Z_s) \|Z_s - D^T \theta D\|_2^2,$$

with, say, Gaussian weights

$$w_h(Z_r, Z_s) = \exp(-\|Z_r - Z_s\|_2^2 / h^2).$$

The estimate for r -th reference window is

$$\hat{\theta}_r = \arg \min_{\theta} I_{h,Z_r}(\theta).$$

Nonlocal transform-domain calculation: single model (cont.)

The minimum conditions

$$\frac{\partial}{\partial \theta} I_{h, Z_r}(\theta) = 0 \implies \sum_s w_h(Z_r, Z_s)(DZ_s D^T - \theta) = 0 \implies$$

$$\hat{\theta}_r = \frac{\sum_s w_h(Z_r, Z_s) \cdot \tilde{\theta}_s}{\sum_s w_h(Z_r, Z_s)}, \quad \tilde{\theta}_r = DZ_r D^T.$$

The solution is the weighted mean of the noisy spectrums $\tilde{\theta}_s$ versus the weighted mean of the means in the standard NL-means.

Nonlocal transform domain: single model with prior (penalty)

All denoising is because of this weighted mean. It can be essentially improved imposing a prior on the spectrum θ .

- The non-local estimate for r -th window is formalized as minimization of the weighted criterion

$$I_{h,Z_r}(\theta) = \frac{1}{\sigma^2} \sum_s w_h(Z_r, Z_s) \|Z_s - D^T \theta D\|_2^2 + \mu^2 \cdot \text{pen}(\theta).$$

Nonlocal transform domain: single model with prior (penalty)

All denoising is because of this weighted mean. It can be essentially improved imposing a prior on the spectrum θ .

- The non-local estimate for r -th window is formalized as minimization of the weighted criterion

$$I_{h,Z_r}(\theta) = \frac{1}{\sigma^2} \sum_s w_h(Z_r, Z_s) \|Z_s - D^T \theta D\|_2^2 + \mu^2 \cdot \text{pen}(\theta).$$

- $\text{pen}(\theta) = \|\theta\|_0$ is l_0 -norm is equal to a number of nonzero items in the matrix θ . $\|\theta\|_0$ is a measure of the model complexity.

Nonlocal transform domain: single model with prior (penalty)

All denoising is because of this weighted mean. It can be essentially improved imposing a prior on the spectrum θ .

- The non-local estimate for r -th window is formalized as minimization of the weighted criterion

$$I_{h,Z_r}(\theta) = \frac{1}{\sigma^2} \sum_s w_h(Z_r, Z_s) \|Z_s - D^T \theta D\|_2^2 + \mu^2 \cdot pen(\theta).$$

- $pen(\theta) = \|\theta\|_0$ is l_0 -norm is equal to a number of nonzero items in the matrix θ . $\|\theta\|_0$ is a measure of the model complexity.
- Other norms also can be used $\|\theta\|_1 = \sum |\theta_{ij}|$, $\|\theta\|_2 = \sqrt{\sum \theta_{ij}^2}$.

Nonlocal transform domain: single model with prior (penalty)

All denoising is because of this weighted mean. It can be essentially improved imposing a prior on the spectrum θ .

- The non-local estimate for r -th window is formalized as minimization of the weighted criterion

$$I_{h,Z_r}(\theta) = \frac{1}{\sigma^2} \sum_s w_h(Z_r, Z_s) \|Z_s - D^T \theta D\|_2^2 + \mu^2 \cdot \text{pen}(\theta).$$

- $\text{pen}(\theta) = \|\theta\|_0$ is l_0 -norm is equal to a number of nonzero items in the matrix θ . $\|\theta\|_0$ is a measure of the model complexity.
- Other norms also can be used $\|\theta\|_1 = \sum |\theta_{ij}|$, $\|\theta\|_2 = \sqrt{\sum \theta_{ij}^2}$.
- A minimal complexity estimate is calculated as

$$\hat{\theta}_r = \arg \min_{\theta} \frac{1}{\sigma^2} \sum_s w_h(Z_r, Z_s) \|\tilde{\theta}_s - \theta\|_2^2 + \mu^2 \cdot \|\theta\|_0.$$

Nonlocal transform domain: single model with prior, single window

For a single window the minimal complexity model is calculated as

$$\hat{\theta}_r = \arg \min_{\theta} \frac{1}{\sigma^2} \|\tilde{\theta}_r - \theta\|_2^2 + \mu^2 \cdot \|\theta\|_0.$$

It can be shown that this solution gives the so-called hard-thresholding (filtering)

$$\hat{\theta}_r(i, j) = \begin{cases} \tilde{\theta}_r(i, j), & \text{if } |\tilde{\theta}_r(i, j)| \geq \sigma\mu, \\ 0, & \text{if } |\tilde{\theta}_r(i, j)| < \sigma\mu. \end{cases}$$

We use the symbol $Y(\cdot)$ for the thresholding operation

$$\hat{\theta}_r(i, j) = Y(\tilde{\theta}_r(i, j)).$$

When $\hat{\theta}_r$ found the windowed estimate is calculated as

$$\hat{Y}_r = D^T \hat{\theta}_r D.$$

Nonlocal transform domain: biasedness and variance

Let us evaluate the efficiency of the thresholding filtering.

For the noisy data we have (before filtering)

$$E\{\tilde{\theta}_r\} = E\{D_r Z_r D_r^T\} = D_r E\{Z_r\} D_r^T = D_r Y_r D_r^T = \theta_r,$$

thus $\theta_r - E\{\tilde{\theta}_r\} = 0$.

For the hard-thresholded data

$$E\{\hat{\theta}_r(i,j)\} = \begin{cases} \theta_r(i,j), & \text{if } |\tilde{\theta}_r(i,j)| \geq \sigma\mu, \\ 0, & \text{if } |\tilde{\theta}_r(i,j)| < \sigma\mu, \end{cases} \implies$$

$$\theta_r(i,j) - E\{\hat{\theta}_r(i,j)\} = \begin{cases} 0, & \text{if } |\tilde{\theta}_r(i,j)| \geq \sigma\mu, \\ \theta_r, & \text{if } |\tilde{\theta}_r(i,j)| < \sigma\mu. \end{cases}$$

Nonlocal transform domain: biasedness and variance (cont.)

The variance of the thresholded data

$$\Delta_r \theta = \tilde{\theta}_r - \theta_r = D_r Z_r D_r^T - D_r Y_r D_r^T = D_r E_r D_r^T;$$

$$S_{var} = \sum_{i,j} \text{var}\{\tilde{\theta}_r(i,j)\} = \sum_{i,j} E(\Delta_r \theta(i,j))^2 = \sigma^2 n_r^2.$$

$$S_{var,thr} = \sum_{i,j} E(\hat{\theta}_r(i,j))^2 = \sigma^2 (n_r^2 - n_{r,0}^2) = \sigma^2 N_{har}^{x_r},$$

where $n_{r,0}^2$ is a number of $|\hat{\theta}_r| = 0$, $N_{har}^{x_r}$ is the number of retained (non-zero) coefficients after hard-thresholding.

Nonlocal transform domain with prior: single model, many windows

It can be verified that

$l_{h,Z_r}(\theta) = \frac{1}{\sigma^2} \sum_s w_h(Z_r, Z_s) \|Z_s - D^T \theta D\|_2^2 + \mu^2 \cdot \text{pen}(\theta)$ can be rewritten as

$$l_{h,Z_r}(\theta) = \frac{1}{\sigma^2} \sum_s w_h(Z_r, Z_s) \|\bar{\theta}_r - \theta\|_2^2 + \mu^2 \cdot \text{pen}(\theta) + \text{const},$$

where

$$\bar{\theta}_r = \sum_s w_h(Z_r, Z_s) \tilde{\theta}_s / \sum_s w_h(Z_r, Z_s).$$

Then the minimization of $l_{h,Z_r}(\theta)$ and estimation for the reference block becomes

$$\hat{\theta} = \arg \min_{\theta} \|\bar{\theta}_r - \theta\|_2^2 + \mu_r^2 \text{pen}(\theta),$$

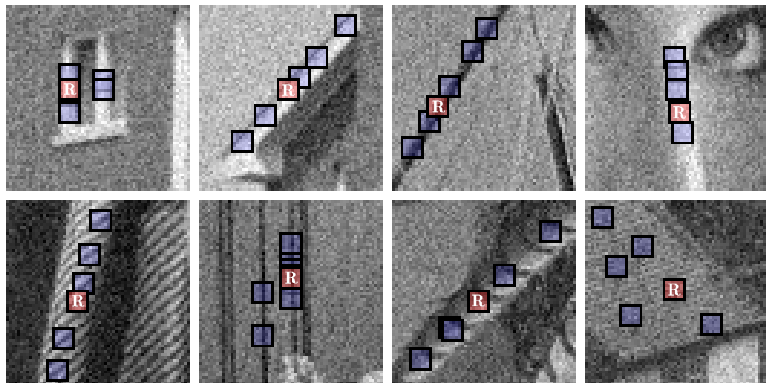
$$\hat{Y}_r = \mathcal{T}^{2D-1}(\hat{\theta}).$$

where $\mu_r^2 = \sigma^2 \mu^2 / \sum_s w_h(Z_r, Z_s)$.

If $\text{pen}(\theta) = \|\theta\|_0$ the estimate $\hat{\theta}$ is a thresholded version of $\bar{\theta}_r$.

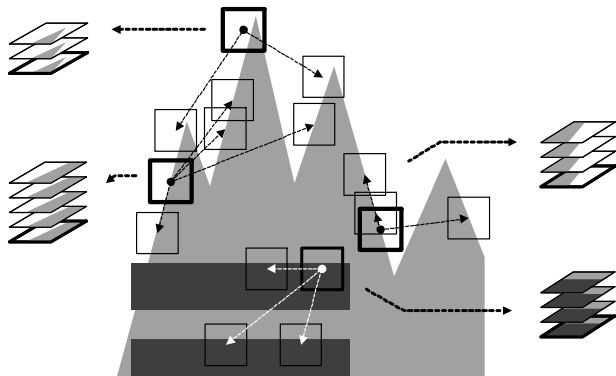
Block matching and 3D filtering (BM3D): nonlocal filtering with multiple models

K. Dabov, et al., "Image denoising by sparse 3-D transform-domain collaborative filtering," IEEE Trans. IP, vol. 16, no. 8, 2007.



Stage I (block matching, grouping)

The grouping is a concept of collecting similar d -dimensional fragments of a given signal into a $d + 1$ -dimensional data structure that we term “*group*.”



Stage I (block matching, grouping)

- Distance between the blocks:

$$d(Z_r, Z_s) = \|Z_r - Z_s\|_p,$$

where typically $p = 2$.

Stage I (block matching, grouping)

- Distance between the blocks:

$$d(Z_r, Z_s) = \|Z_r - Z_s\|_p,$$

where typically $p = 2$.

- Grouping rule:

$$S_r^{\text{ht}} = \{s : d(Z_r, Z_s) \leq \tau\}.$$

$\tau \geq 0$ is a (dis)similarity threshold.

Stage I (block matching, grouping)

- Distance between the blocks:

$$d(Z_r, Z_s) = \|Z_r - Z_s\|_p,$$

where typically $p = 2$.

- Grouping rule:

$$S_r^{\text{ht}} = \{s : d(Z_r, Z_s) \leq \tau\}.$$

$\tau \geq 0$ is a (dis)similarity threshold.

- Group (stacking represented as disjoint union):

$$\mathbf{Z}_{S_r^{\text{ht}}} = \coprod_{s \in S_r^{\text{ht}}} Z_s.$$

Stage II (3D collaborative filtering)

- Collaborative filtering is realized as shrinkage in transform domain. Assuming $(2 + 1)$ -dimensional groups of similar signal fragments are already formed, the approach comprises of the following steps.

Stage II (3D collaborative filtering)

- Collaborative filtering is realized as shrinkage in transform domain. Assuming $(2 + 1)$ -dimensional groups of similar signal fragments are already formed, the approach comprises of the following steps.
- Apply a $(2 + 1)$ -dimensional linear transform on a group.

Stage II (3D collaborative filtering)

- Collaborative filtering is realized as shrinkage in transform domain. Assuming $(2 + 1)$ -dimensional groups of similar signal fragments are already formed, the approach comprises of the following steps.
- Apply a $(2 + 1)$ -dimensional linear transform on a group.
- Shrink (e.g. by soft- and hard-thresholding, Wiener filtering) the transform coefficients to attenuate the noise.

Stage II (3D collaborative filtering)

- Collaborative filtering is realized as shrinkage in transform domain. Assuming $(2 + 1)$ -dimensional groups of similar signal fragments are already formed, the approach comprises of the following steps.
- Apply a $(2 + 1)$ -dimensional linear transform on a group.
- Shrink (e.g. by soft- and hard-thresholding, Wiener filtering) the transform coefficients to attenuate the noise.
- **Invert the transform to produce estimates of all grouped fragments.**

Stage II (3D collaborative filtering)

- Collaborative filtering is realized as shrinkage in transform domain. Assuming $(2 + 1)$ -dimensional groups of similar signal fragments are already formed, the approach comprises of the following steps.
- Apply a $(2 + 1)$ -dimensional linear transform on a group.
- Shrink (e.g. by soft- and hard-thresholding, Wiener filtering) the transform coefficients to attenuate the noise.
- Invert the transform to produce estimates of all grouped fragments.
- **Return the filtered fragments to their original locations.**

Collaborative filtering implementation

The collaborative filtering of $\mathbf{Z}_{S_{xr}}^{\text{ht}}$ is realized by hard-thresholding in 3D transform domain.

The 3D linear transform, denoted $\mathcal{T}_{3D}^{\text{ht}}$, is expected to take advantage of the sparsity for the true signal group $\mathbf{Y}_{S_{xr}}^{\text{ht}}$.

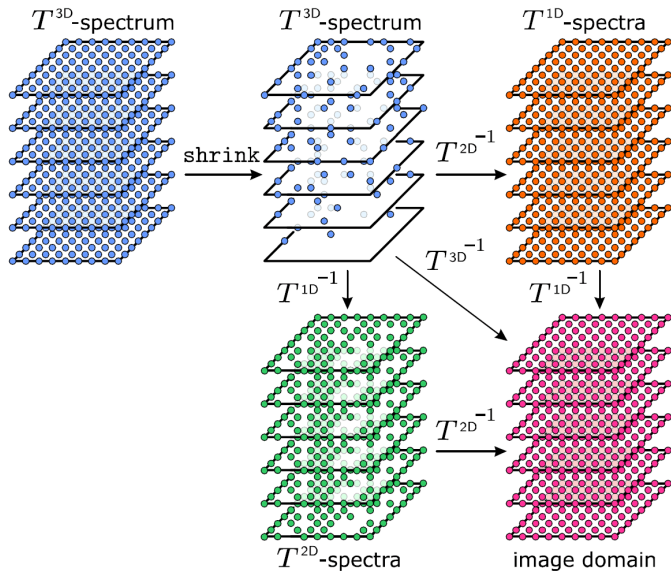
This allows for effective noise attenuation by hard-thresholding, followed by inverse transform that yields a 3D array of block-wise estimates

$$\hat{\mathbf{Y}}_{S_{xr}}^{\text{ht}} = \mathcal{T}_{3D}^{\text{ht}^{-1}} \left(\Upsilon \left(\mathcal{T}_{3D}^{\text{ht}} \left(\mathbf{Z}_{S_{xr}}^{\text{ht}} \right) \right) \right),$$

where Υ is a hard-threshold operator with threshold $\mu_{3D}\sigma$.

It is assumed that the $\mathcal{T}_{3D}^{\text{ht}}$ transform has a DC-term which is never thresholded.

Sparsity of Collaborative 3D Hard-Thresholding



Stage III (fusion/aggregation of filtered fragments)

The estimates of *Stage II* are heavily overlapping.

Calculate the pointwise estimates using the variances of the estimates in the fragments.

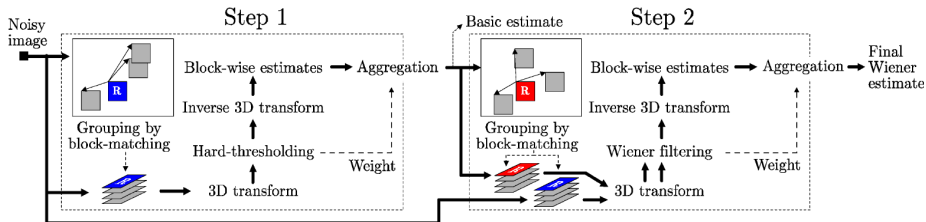
The final estimate \hat{y} is computed by a weighted average of the block-wise estimates $\hat{Y}_{x \in S_{x_r}^{\text{ht}}}^{\text{ht}, x_r}$ using the special weights $w_{x_r}^{\text{ht}}$

$$\hat{y}^{\text{basic}}(x) = \frac{\sum_{x_r \in X} \sum_{x_m \in S_{x_r}^{\text{ht}}} w_{x_r}^{\text{ht}} \hat{Y}_{x_m}^{\text{ht}, x_r}(x)}{\sum_{x_r \in X} \sum_{x_m \in S_{x_r}^{\text{ht}}} w_{x_r}^{\text{ht}} \chi_{x_m}(x)}, \forall x \in X,$$

$$w_{x_r}^{\text{ht}} = \frac{1}{\sigma^2 N_{\text{har}}^{x_r}}$$

where $\chi_{x_m} : X \rightarrow \{0, 1\}$ is the characteristic function of the square support of a block located at $x_m \in X$, and the block-wise estimates $\hat{Y}_{x_m}^{\text{ht}, x_r}$ are zero-padded outside of their support, where $N_{\text{har}}^{x_r} \geq 1$ is the number of retained (non-zero) coefficients after hard-thresholding.

BM3D Algorithm Flowchart



Step 1 – Hard thresholding

Step 2 – Wiener filtering

Features of BM3D's Wiener-filtering step

- Block Matching is carried out on the basic estimate \hat{y}^{basic} (output of the hard-thresholding step).

Features of BM3D's Wiener-filtering step

- Block Matching is carried out on the basic estimate $\hat{\mathbf{y}}^{\text{basic}}$ (output of the hard-thresholding step).
- Hard thresholding is replaced by Wiener filtering. We define the empirical Wiener shrinkage coefficients as

$$\mathbf{W}_{S_{x_r}^{\text{wie}}} = \frac{\left| \mathcal{T}_{3\text{D}}^{\text{wie}} \left(\hat{\mathbf{Y}}_{S_{x_r}^{\text{wie}}}^{\text{basic}} \right) \right|^2}{\left| \mathcal{T}_{3\text{D}}^{\text{wie}} \left(\hat{\mathbf{Y}}_{S_{x_r}^{\text{wie}}}^{\text{basic}} \right) \right|^2 + \sigma^2}, \quad \hat{\mathbf{Y}}_{S_{x_r}^{\text{wie}}}^{\text{wie}} = \mathcal{T}_{3\text{D}}^{\text{wie}^{-1}} \left(\mathbf{W}_{S_{x_r}^{\text{wie}}} \mathcal{T}_{3\text{D}}^{\text{wie}} \left(\mathbf{Z}_{S_{x_r}^{\text{wie}}} \right) \right).$$

Features of BM3D's Wiener-filtering step

- Block Matching is carried out on the basic estimate \hat{y}^{basic} (output of the hard-thresholding step).
- Hard thresholding is replaced by Wiener filtering. We define the empirical Wiener shrinkage coefficients as

$$\mathbf{W}_{S_{x_r}^{\text{wie}}} = \frac{\left| \mathcal{T}_{3\text{D}}^{\text{wie}} \left(\hat{\mathbf{Y}}_{S_{x_r}^{\text{wie}}}^{\text{basic}} \right) \right|^2}{\left| \mathcal{T}_{3\text{D}}^{\text{wie}} \left(\hat{\mathbf{Y}}_{S_{x_r}^{\text{wie}}}^{\text{basic}} \right) \right|^2 + \sigma^2}, \quad \hat{\mathbf{Y}}_{S_{x_r}^{\text{wie}}}^{\text{wie}} = \mathcal{T}_{3\text{D}}^{\text{wie}^{-1}} \left(\mathbf{W}_{S_{x_r}^{\text{wie}}} \mathcal{T}_{3\text{D}}^{\text{wie}} \left(\mathbf{z}_{S_{x_r}^{\text{wie}}} \right) \right).$$

- The weights in the aggregation are calculated as

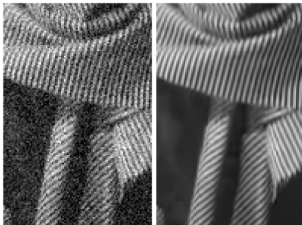
$$\hat{y}^{\text{final}}(x) = \frac{\sum_{x_r \in X} \sum_{x_m \in S_{x_r}^{\text{wie}}} w_{x_r}^{\text{wie}} \hat{\mathbf{Y}}_{S_{x_m}^{\text{wie}}}^{\text{wie}}}{\sum_{x_r \in X} \sum_{x_m \in S_{x_r}^{\text{wie}}} w_{x_r}^{\text{wie}} \chi_{x_m}(x)}, \quad \forall x \in X, \quad w_{x_r}^{\text{wie}} = \sigma^{-2} \left\| \mathbf{W}_{S_{x_r}^{\text{wie}}} \right\|_2^{-2}$$

Examples of BM3D denoising performance

(a) *Lena* (PSNR 32.08 dB)



(b) *Barbara* (PSNR 30.73 dB)



(c) *Camerman* (PSNR 29.45 dB)



(d) *Man* (PSNR 29.62 dB)



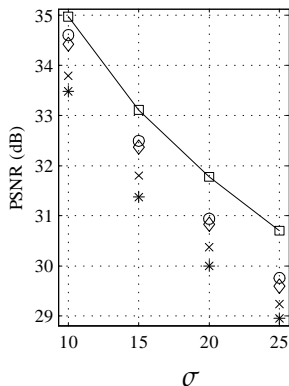
(e) *Boats* (PSNR 29.91 dB)



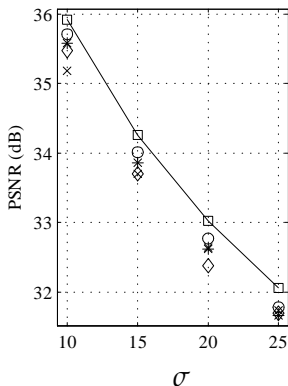
(f) *Couple* (PSNR 29.72 dB)



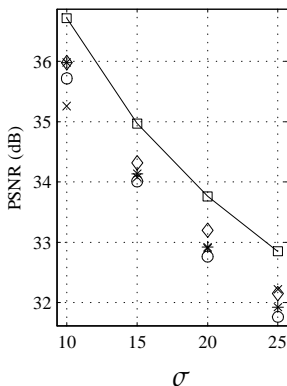
Nonlocal BM3D vs SA-DCT: value of nonlocality



(a) *Barbara*



(b) *Lena*



(c) *House*

PSNR as a function of σ . BM3D ("squares"), BLS-GSM (circles), exemplar-based denoising ("x"), K-SVD denoising ("diamonds"), and Pointwise SA-DCT denoising ("stars").

Dependency of the output PSNR (dB) on the used transforms

Transform $\sigma = 25$	Boats			Lena		
	\mathcal{T}_{2D}^{ht}	\mathcal{T}_{2D}^{wie}	\mathcal{T}_{1D}	\mathcal{T}_{2D}^{ht}	\mathcal{T}_{2D}^{wie}	\mathcal{T}_{1D}
Haar	29.31	29.84	29.91	31.24	31.93	32.08
Db2	29.22	29.83	29.90	31.19	31.97	32.06
Db4	29.34	29.88	29.89	31.31	32.01	32.06
Db6	29.30	29.86	29.89	31.28	31.98	32.06
Bior1.3	29.42	29.87	29.90	31.35	31.96	32.06
Bior1.5	29.43	29.88	29.90	31.37	31.97	32.06
WHT	29.22	29.84	29.88	31.24	32.00	32.07
DCT	29.35	29.91	29.88	31.42	32.08	32.07
DST	29.33	29.91	29.79	31.36	31.97	31.92
DC+rand	29.07	29.75	29.88	31.06	31.88	32.06
DC-only	-	-	28.03	-	-	30.65

Computational complexity

Quality vs. speed scalability

Approx. execution time for a 256×256 grayscale image
on 1.5-GHz Celeron M (Matlab)

Fast profile	0.7 sec.
Normal profile	4.1 sec.

Fast profile results in about 0.2-dB PSNR loss vs. Normal profile.

Extensive independent benchmarking demonstrates the superiority of the BM3D algorithms

PSNR/MSSIM

- <http://www.stanford.edu/~slansel/DenoiseLab/>
- <http://www.cs.utoronto.ca/~strider/Denoise/Benchmark/>

Perceptual quality (human subjects)

- Van der Weken, D., E. Kerre, E. Vansteenkiste, and W. Philips, "Evaluation of fuzzy image quality measures using a multidimensional scaling framework", Proc. 2nd Int. Workshop Video Process. Quality Metrics Consum. Electron., VPQM2006, Scottsdale, AZ, Jan. 2006.
- Vansteenkiste, E., D. Van der Weken, W. Philips, and E. Kerre, "Perceived image Quality Measurement of state-of-the-art Noise Reduction Schemes", Lecture Notes in Computer Science 4179 - ACIVS 2006, pp. 114-124, Springer, Sep. 2006.

Performance bounds for the image denoising problem

- <http://users.soe.ucsc.edu/~priyam/bounds/>

Is Denoising Dead?

Priyam Chatterjee, *Student Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

Abstract—Image denoising has been a well studied problem in the field of image processing. Yet researchers continue to focus attention on it to better the current state-of-the-art. Recently proposed methods take different approaches to the problem and yet their denoising performances are comparable. A pertinent question then to ask is whether there is a theoretical limit to denoising performance and, more importantly, are we there yet? As camera manufacturers continue to pack increasing numbers of pixels per unit area, an increase in noise sensitivity manifests itself in the form of a noisier image. We study the performance bounds for the image denoising problem. Our work in this paper estimates a lower bound on the mean squared error of the denoised result and compares the performance of current state-of-the-art denoising methods with this bound. We show that despite the phenomenal recent progress in the quality of denoising algorithms, some room for improvement still remains for a wide class of general images, and at certain signal-to-noise levels. Therefore, image denoising is not dead—yet.

Index Terms—Bayesian Cramér–Rao lower bound (CRLB), bias, bootstrapping, image denoising, mean squared error.

I. INTRODUCTION

IMAGE denoising has been a well-studied problem in the image processing community and continues to attract

erature on such performance limits exists for some of the more complex image processing problems such as image registration [7], [8] and super-resolution [9]–[12]. Performance limits to object or feature recovery in images in the presence of point-wise degradation has been studied by Treibitz *et al.* [13]. In their work, the authors study the effects of noise among other degradations and formulate expressions for the optimal filtering parameters that define the resolution limits to recovering any given feature in the image. While their study is practical, it does not define statistical performance limits to denoising general images. In [14], Voloshynovskiy *et al.* briefly analyze the performance of MAP estimators for the denoising problem. However, our bounds are developed in a much more general setting and, to the best of our knowledge, no comparable study currently exists for the problem of denoising. The present study will enable us to understand how well the state-of-the-art denoising algorithms perform as compared to these limits. From a practical perspective, it will also lead to understanding the fundamental limits of increasing the number of sensors in the imaging system with acceptable image quality being made possible by noise suppression algorithms.

Before we analyze image denoising statistically, we first de-

Is Denoising Dead?

Privam Chatterjee, *Student Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

despite the phenomenal recent progress in the quality of denoising algorithms, some room for improvement still remains for a wide class of general images, and at certain signal-to-noise levels. Therefore, image denoising is not dead—yet.

denoising problem. Our work in this paper estimates a lower bound on the mean squared error of the denoised result and compares the performance of current state-of-the-art denoising methods with this bound. We show that despite the phenomenal recent progress in the quality of denoising algorithms, some room for improvement still remains for a wide class of general images, and at certain signal-to-noise levels. Therefore, image denoising is not dead—yet.

Index Terms—Bayesian Cramér–Rao lower bound (CRLB), bias, bootstrapping, image denoising, mean squared error.

I. INTRODUCTION

IMAGE denoising has been a well-studied problem in the image processing community and continues to attract

does not define statistical performance limits to denoising general images. In [14], Voloshynovskiy *et al.* briefly analyze the performance of MAP estimators for the denoising problem. However, our bounds are developed in a much more general setting and, to the best of our knowledge, no comparable study currently exists for the problem of denoising. The present study will enable us to understand how well the state-of-the-art denoising algorithms perform as compared to these limits. From a practical perspective, it will also lead to understanding the fundamental limits of increasing the number of sensors in the imaging system with acceptable image quality being made possible by noise suppression algorithms.

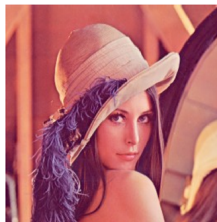
Before we analyze image denoising statistically, we first de-

Color denoising

Exploit structural correlation in luminance-chrominance space

Key idea: the structures (e.g., objects, edges, details) which determine the spatial adaptivity are the same across all color channels.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/\sqrt{6} & 0 & -1/\sqrt{6} \\ 1/(3\sqrt{2}) & -\sqrt{2}/3 & 1/(3\sqrt{2}) \end{bmatrix} \cdot \begin{bmatrix} y_r \\ y_g \\ y_b \end{bmatrix} \quad \text{opponent color transformation}$$



RGB



Y



U



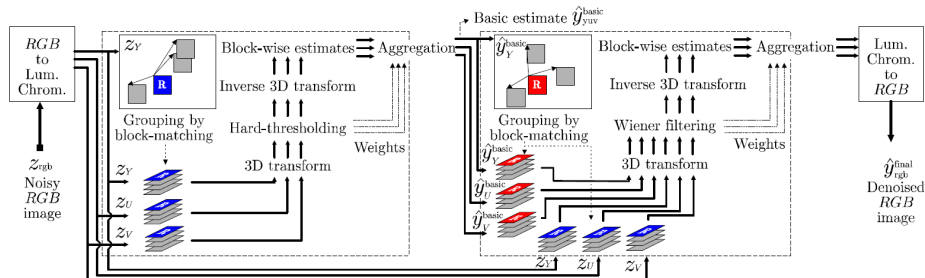
V

The method is implemented after transformation to a *luminance-chrominance color-space* (e.g., opponent, YUV, YCbCr).

Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images", IEEE TIP, vol. 16, no. 5, 2007.

Color Image Denoising with C-BM3D

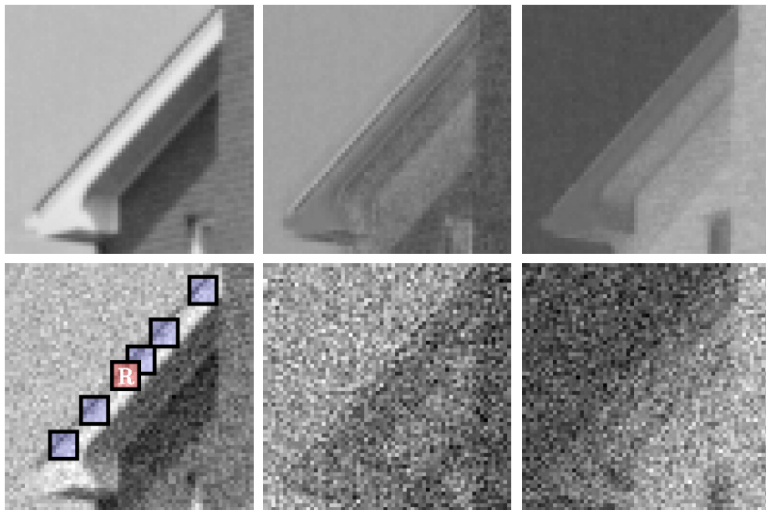
The same grouping defined for the luminance channel is used for all three color channels.



Flowchart of C-BM3D.

K. Dabov, et al., "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," Proc. IEEE Int. Conf. Image Process., ICIP 2007, San Antonio, TX, USA, 2007.

Color Image Denoising Example



Top row contains noise-free Y, Cb, and Cr channels and the bottom row contains corresponding noisy ones ($\sigma = 22$).

Color Image Denoising Example



On the left: noisy ($\sigma = 35$) House and a fragment of it; on the right: the corresponding denoised image (PSNR=31.58 dB) and fragment.

Image Denoising with BM3D

<http://www.cs.tut.fi/~foi/GCF-BM3D/>

(1) BM3D.m

[PSNR, **y_est**] = BM3D(y, z, **sigma**, profile, print_to_screen)

(2) CBM3D.m (color version)

[PSNR, **yRGB_est**] = CBM3D(yRGB, zRGB, **sigma**, profile, print_to_screen, colorspace).

Image Deblurring: setting of the problem

Deblurring problem: reconstruct y_i from $\{z_i, x_i\}_{i=1}^n$,

$$z_i = (y \circledast v)(x_i) + \sigma \varepsilon_i,$$

where $\varepsilon_i = \varepsilon(x_i)$ i.i.d. $\mathcal{N}(0, 1)$.

Standard approach: regularized inverse

$$\hat{y} = \arg \min_y \|z - (y \circledast v)\|_2^2 + \mu \cdot \text{pen}(y), \lambda > 0.$$

An invariant μ is a principal limitation.

In what follows we use $V(f) = \mathcal{F}\{v\}$ and $Z(f) = \mathcal{F}\{z\}$, $Y(f) = \mathcal{F}\{y\}$.

BM3D based deblurring

Our approach is based on two inverse estimates:

- Regularized inverse with small regularization parameter complemented by BM3D filtering: in the frequency domain it is calculated as

$$\hat{Y}^{RI}(f) = \frac{V^*(f)}{|V(f)|^2 + \alpha^2} Z(f),$$

BM3D based deblurring

Our approach is based on two inverse estimates:

- Regularized inverse with small regularization parameter complemented by BM3D filtering: in the frequency domain it is calculated as

$$\hat{Y}^{RI}(f) = \frac{V^*(f)}{|V(f)|^2 + \alpha^2} Z(f),$$

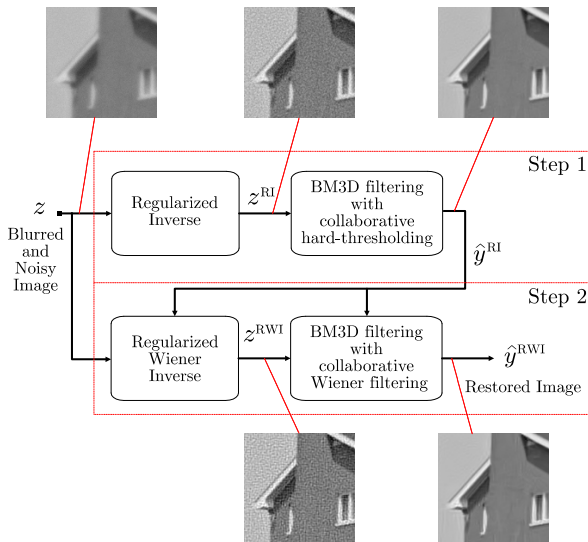
- Wiener inverse complemented by BM3D filtering: in the frequency domain it is calculated as

$$\hat{Y}^{RWI}(f) = \frac{V^*(f)}{|V(f)|^2 + n_1 n_2 \sigma^2 / |\hat{Y}^{RI}(f)|^2} Z(f).$$

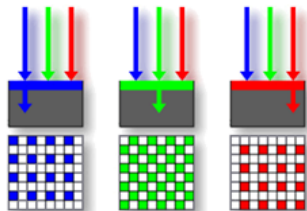
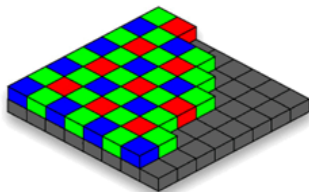
This filtering implements (imitates) effects of the varying adaptive regularization parameter μ .

K. Dabov, et al., "Image restoration by sparse 3D transform-domain collaborative filtering," Proc. SPIE Electronic Imaging '08, no. 6812-07, San Jose, 2008.

Image Deblurring with BM3D (BM3DDEB.m)



Color Filter Array



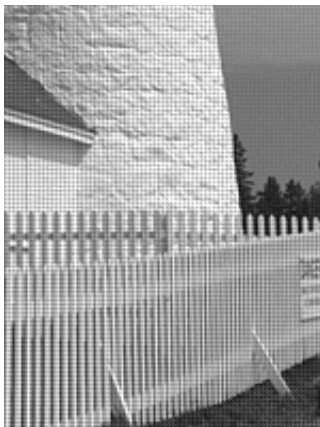
Bayer color filter array (CFA)

High correlation between channels. It is likely that color channels are going to have similar texture and edge locations.

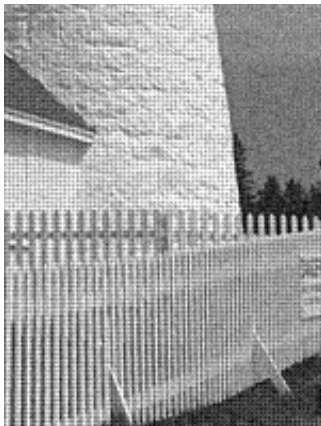
L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," IEEE Trans. IP., vol. 14, no. 12, 2005.

Paliy, et al. "Denoising and Interpolation of Noisy Bayer Data with Adaptive Cross-Color Filters", SPIE-IS&T Electronic Imaging, Visual Communications and Image Processing 2008, vol. 6822, San Jose, 2008.

Noise-Free CFA Demosaicking



Noisy CFA Demosaicking



How to deal with noise?

- Denoising AFTER demosaicking.

No adequate noise model! Interpolation changes statistical model of the noise in a complex and hardly computable form;

Filter should rely only on constraints reflecting the general a priori knowledge about the image structure.

- JOINT demosaicking-denoising. (Paliy et al., Zhang et. al)

Modify demosaicking method to be robust against noise;

Design is difficult! Combining antagonistic procedures. Denoising \sim smoothing, while demosaicking/interpolation \sim reconstruction of high-frequency details.

- Can we do Denoising BEFORE demosaicking?

Mosaic structure violates assumptions about local smoothness of the natural images on which filters were relying;

Split(R, G_1, G_2, B) \rightarrow Denoise \rightarrow Combine - this leads to smoothing fine details.

Modern Cross-Color Denoising Approaches

What has changed in novel algorithms?

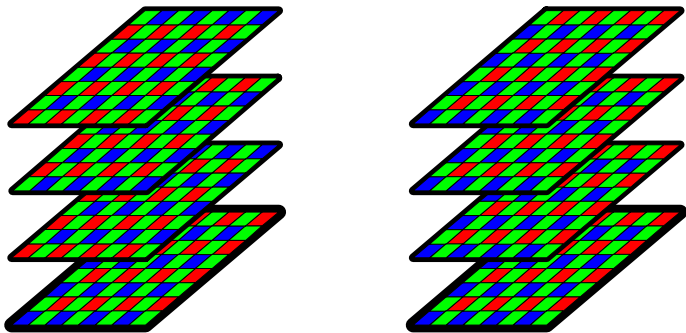
Local smoothing is not critical anymore. Filters exploiting non-local similarity of small image patches.

Zhang, et al., "PCA-based Spatial Adaptive Denoising of CFA Images for Single-Sensor Digital Cameras," IEEE Trans. IP, vol. 18, no. 4, 2009.

A. Danielyan, et al., "Cross-color BM3D filtering of noisy raw data", Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2009, Tuusula, Finland, pp. 125-129, 2009.





Foi, A., "Clipped noisy images: heteroskedastic modeling and practical denoising", Signal Processing, vol. 89, no. 12, pp. 2609-2629, Dec. 2009.

BM3D with color-constrained grouping







Block grouping in BM3D modeling: unconstrained (left) and color-constrained (right)

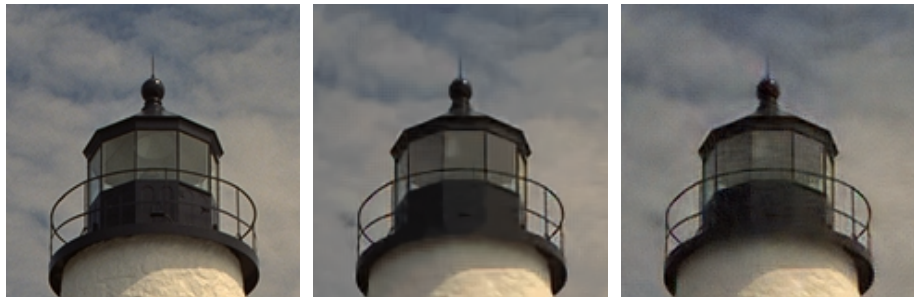
Experiment 1. Denoising and demosaicking (GN).

		Proposed	Zhang, 2009	Proposed	Zhang, 2009
σ		5/255		12/255	
	R	37.8	36.8	33.9	32.6
	G	39.1	38.0	34.6	33.2
	B	37.5	36.6	33.8	32.6
	R	32.7	31.7	29.6	28.5
	G	34.5	33.2	30.5	29.3
	B	32.8	31.9	29.8	28.7
	R	35.7	34.8	31.8	30.9
	G	36.8	35.9	32.5	31.6
	B	36.3	35.4	32.6	31.6
	R	37.7	37.3	34.4	33.9
	G	39.3	38.7	35.5	34.8
	B	38.2	37.6	34.6	33.9

Experiment 2. Gaussian-Poissonian Noise and clipping

		Proposed	Zhang, 2009
$\sigma^2(y) = ay + b$		$a = 0.004, b = 0.02^2$	
	R	34.1	32.7
	G	34.9	33.5
	B	34.3	33.1
	R	29.5	28.3
	G	30.4	29.2
	B	29.7	28.5
	R	32.0	31.0
	G	32.6	31.7
	B	32.7	31.7
	R	34.2	33.7
	G	35.4	34.7
	B	34.8	34.2

Gaussian noise



From left to right: ground truth, proposed denoising + interpolation (Zhang, 2005), denoising (Zhang, 2009) + interpolation (Zhang, 2005), Gaussian noise, $\sigma = 12/255$.

Signal-dependent noise



From left to right: ground truth, proposed denoising + interpolation (Zhang, 2005), denoising (Zhang, 2009) + interpolation (Zhang, 2005), signal-dependent noise ($a = 0.004$, $b = 0.02$).

BM3D algorithms with adaptive-shape neighborhoods

BM3D algorithm with SA-DCT (SA-BM3D)

- The algorithm uses grouping of adaptive-shape neighborhoods whose surrounding square supersets have been found similar by a block-matching procedure.
- The data defined on these grouped neighborhoods is stacked together, resulting in 3-D structures which are generalized cylinders with adaptive-shape cross sections.
- These 3-D groups are characterized by a high correlation along all the three dimensions.
- A 3-D decorrelating transform is computed as a separable composition of the Shape-Adaptive DCT (SA-DCT) and a 1-D orthonormal transform.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "A Nonlocal and Shape-Adaptive Transform-Domain Collaborative Filtering," Proc. 2008 Int. Workshop on Local and Non-Local Approximation in Image Processing, LNLA 2008, Lausanne, Switzerland, 2008.

BM3D algorithm with SA-DCT (SA-BM3D)

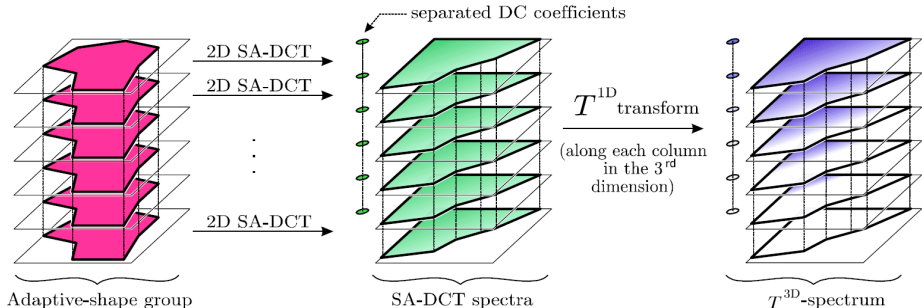


Illustration of applying 3D transform on a group of shape-adaptive neighborhoods.

BM3D algorithm with SA-DCT (SA-BM3D)

BM3D
27.93



SA-DCT
27.51



SA-BM3D
27.95

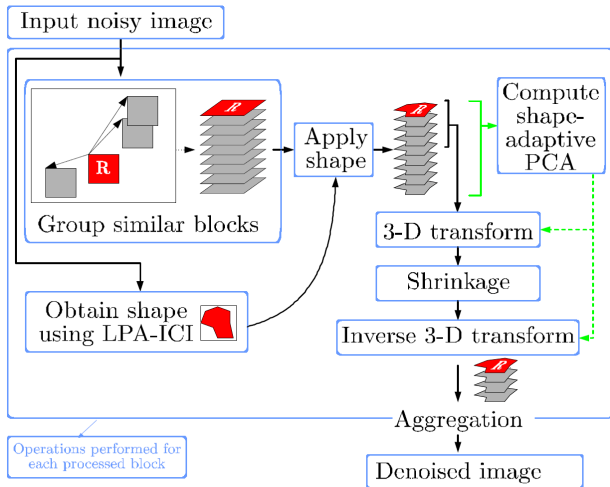


BM3D's good reconstruction of textures and regular image structures and SA-DCT's good reconstruction of sharp edges of varying curvature and image singularities.

BM3D algorithm with shape-adaptive PCA (BM3D-SAPCA)

- A proper selection of the transform is crucial element for ensuring the success of the transform-based methods.
- This problem, known under different names as best basis, dictionary, or prior selection, has been a subject of intensive study from the very beginning of the development and application of estimation/approximation methods. In particular, the use of bases adaptive to the data at hand is of special interest.
- This latest version of BM3D algorithm incorporating a shape-adaptive Principal Component Analysis (PCA) as part of the 3-D transform. For a 3-D group of adaptive-shape image patches, a shape-adaptive PCA basis is obtained by eigenvalue decomposition of an empirical second-moment matrix computed from these patches.
- Overall 3-D transform is a separable composition of the PCA (applied on each image patch) and a fixed orthogonal 1-D transform in the third dimension.

Flowchart of BM3D-SAPCA



K. Dabov, et al., "BM3D Image Denoising with Shape-Adaptive Principal Component Analysis", Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09), Saint-Malo, France, April 2009

Principal Components of PCA in BM3D-SAPCA

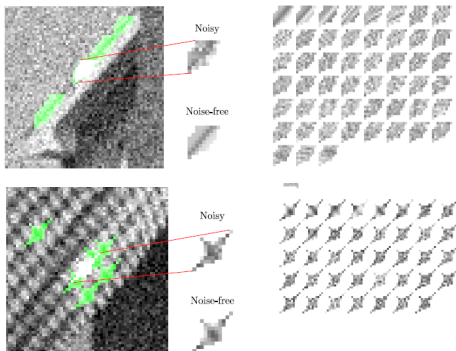


Illustration of the PCs (shown on the right side). The green overlay shows the found similar neighborhoods used to form a 3-D group.

The PCs are listed in decreasing magnitude of their corresponding eigenvalues. The first few PCs have the strongest similarity with the noise-free signal in the neighborhood.

Each of the 2-D neighborhoods in the group is vectorized with vectors a_s of the length N_e . The sample estimate of the covariance matrix $N_e \times N_e$ is calculated as

$$\hat{C} = [a_1, a_2, \dots, a_{N_g}] \cdot [a_1, a_2, \dots, a_{N_g}]^T,$$

where N_g is a number of the patches in the group.
The PCA eigenvalue decomposition yields

$$U^T \hat{C} U = S = \text{diag}\{s_1, \dots, s_{N_e}\}.$$

The largest eigenvalues such that $s_j > th \cdot \sigma^2$ are selected and the corresponding orthonormal columns of U are used as a set of orthonormal bases.

Illustrations: from BM3D to BM3D-SAPCA



Original



Noisy, $\sigma = 35$



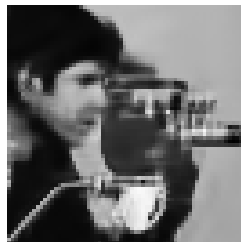
BM3D (27.82, 0.8207)



P.SADCT (27.51, 0.8143)



SA-BM3D (28.02, 0.8228)



BM3D-SAPCA (28.16, 0.8269)

The use of a data-driven adaptive transforms for the collaborative filtering results in a further improvement of the denoising performance, especially in preserving

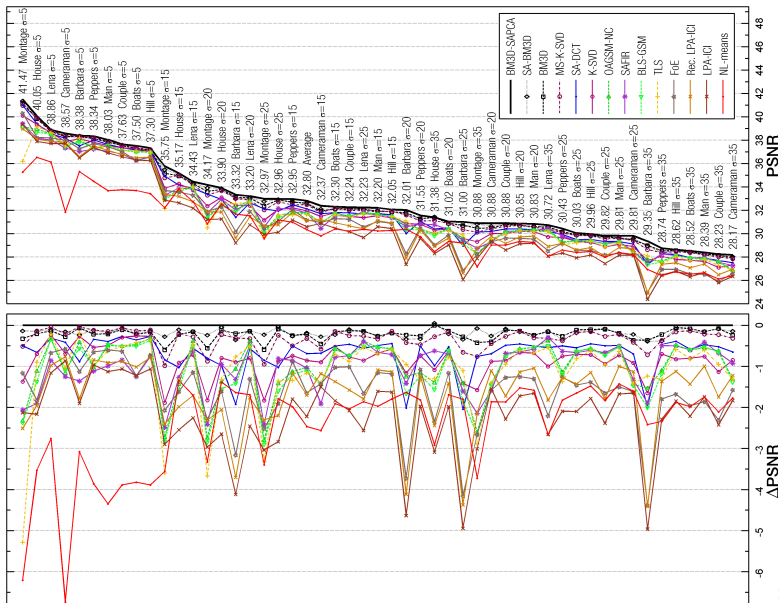
Experimental comparison

The algorithms are applied on a set of 10 different test images corrupted by additive white Gaussian noise with standard deviations $\sigma = 5, 15, 20, 25, 35$. The comparison is made in terms of both PSNR and mean structural-similarity index map (MSSIM).

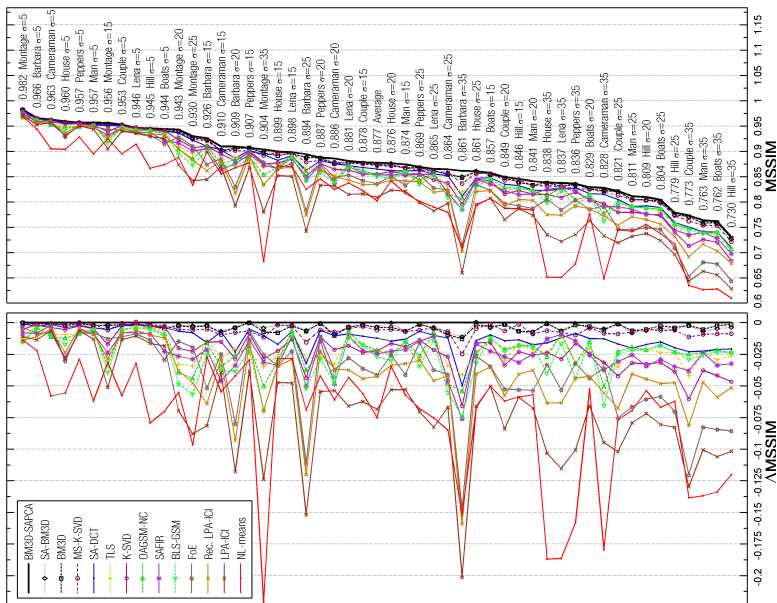
We can see that three algorithms based on the collaborative filtering paradigm occupy the top-three places also in this comparison.

Z. Wang, et al. "Image quality assessment: From error visibility to structural similarity," IEEE Trans. IP. 13, no. 4, 2004.

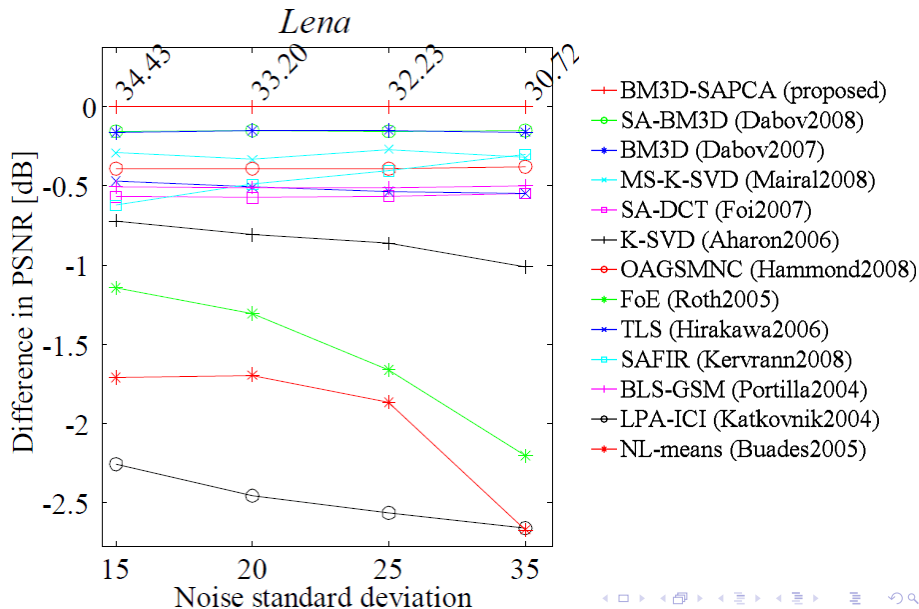
Experimental comparison (PSNR)



Experimental comparison (MSSIM)

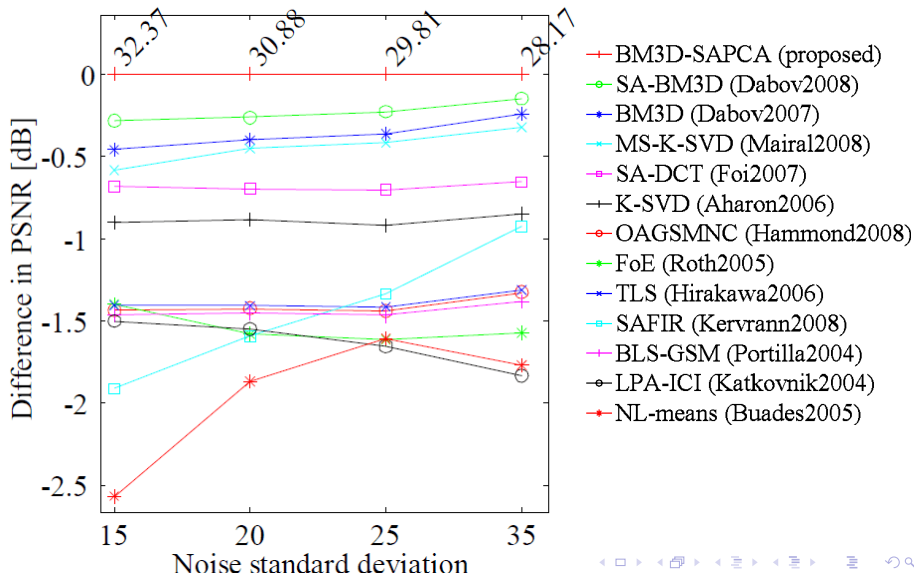


Experimental comparison (PSNR)



Experimental comparison (PSNR)

Cameraman



- Non-Gaussian Image Processing;
- Video Processing with BM3D;
- Compressive Sensing;
- Image Resizing;
- Video Super-Resolution;
- BM3D Joint Denoising-Sharpening through Alpha-Rooting

One-parameter families of distributions

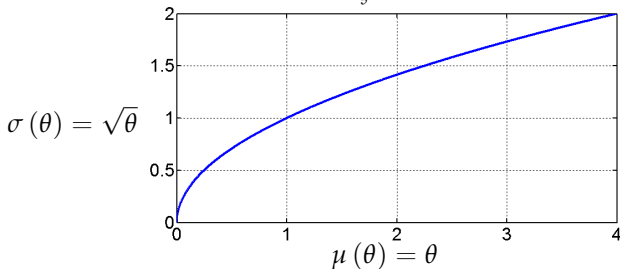
Let $z \in Z \subseteq \mathbb{R}$ be a random variable distributed according to a one-parameter family of distributions $\mathcal{D} = \{\mathcal{D}_\theta\}$, where $\theta \in \Theta \subseteq \mathbb{R}$ denotes the parameter.

$$\mu(\theta) = E\{z|\theta\} \quad \text{and} \quad \sigma(\theta) = \text{std}\{z|\theta\}$$

conditional expectation and standard deviation of z
given as functions of the parameter θ .

Example: \mathcal{D} Poisson distributions with mean $\theta \in \Theta = [0, +\infty)$,

$$\Pr[z = \zeta|\theta] = e^{-\theta} \frac{\theta^\zeta}{\zeta!}, \quad \zeta \in \mathbb{N}.$$



One-parameter families of distributions

\mathcal{D}_θ	$\mu(\theta)$	$\sigma(\theta)$
Poisson		
$\Pr[z = \zeta \theta] = e^{-\theta} \frac{\theta^\zeta}{\zeta!}, \zeta \in \mathbb{N}, \theta \in [0, +\infty)$	θ	$\sqrt{\theta}$
Scaled Poisson (scale $\chi > 0$)		
$\Pr[z = \frac{\zeta}{\chi} \theta] = e^{-\theta} \frac{\theta^\zeta}{\zeta!}, \zeta \in \mathbb{N}, \theta \in [0, +\infty)$	$\frac{\theta}{\chi}$	$\frac{\sqrt{\theta}}{\chi} = \sqrt{\frac{\mu(\theta)}{\chi}}$
Binomial (n trials)		
$\Pr[z = \zeta \theta] = \binom{n}{\zeta} \theta^\zeta (1-\theta)^{n-\zeta}, \zeta \in \mathbb{N}, \theta \in [0, 1]$	$n\theta$	$\sqrt{n\theta(1-\theta)} = \sqrt{\frac{\mu(\theta)(n-\mu(\theta))}{n}}$
Scaled binomial (n trials, scale n)		
$\Pr[z = \frac{\zeta}{n} \theta] = \binom{n}{\zeta} \theta^\zeta (1-\theta)^{n-\zeta}, \zeta \in \mathbb{N}, \theta \in [0, 1]$	θ	$\sqrt{\frac{\theta(1-\theta)}{n}}$
Negative binomial (exponent k)		
$\Pr[z = \zeta \theta] = \frac{\Gamma(\zeta+k)}{\zeta! \Gamma(k)} \left(\frac{\theta}{\theta+k}\right)^\zeta \left(\frac{k+\theta}{k}\right)^{-k}, \zeta \in \mathbb{N}, \theta \in [0, +\infty)$	θ	$\sqrt{\frac{\theta(\theta+k)}{k}}$
Scaled negative binomial (exponent k, scale $\chi > 0$)		
$\Pr[z = \frac{\zeta}{\chi} \theta] = \frac{\Gamma(\zeta+k)}{\zeta! \Gamma(k)} \left(\frac{\theta}{\theta+k}\right)^\zeta \left(\frac{k+\theta}{k}\right)^{-k}, \zeta \in \mathbb{N}, \theta \in [0, +\infty)$	$\frac{\theta}{\chi}$	$\sqrt{\frac{\theta(\theta+k)}{\chi^2 k}} = \sqrt{\frac{\mu(\theta)(\mu(\theta)\chi+k)}{\chi^k}}$
Multiplicative normal (scale $\chi > 0$)		
$\text{pdf}[z \theta](\zeta) = \frac{\chi}{\theta\sqrt{2\pi}} e^{-\frac{(\zeta-\theta)^2 \chi^2}{2\theta^2}}$	θ	$\frac{\theta}{\chi}$
Doubly censored normal with standard-deviation $s(\theta)$		
$\text{pdf}[z \theta](\zeta) = \Phi\left(\frac{-\gamma}{\sigma(y)}\right) \delta_0(\zeta) + \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-\gamma}{\sigma(y)}\right) \chi_{[0,1]} + \left(1 - \Phi\left(\frac{1-\gamma}{\sigma(y)}\right)\right) \delta_0(1-\zeta)$		

Find a function $f : Z \rightarrow \mathbb{R}$ such that the transformed variable $f(z)$ has constant standard deviation, say, equal to c , $\text{std}\{f(z) | \theta\} = c$.

- the (conditional) standard deviation does not depend anymore on the distribution parameter;
- heteroskedastic z turns into a homoskedastic $f(z)$.

Constraints:

- !!! f should be independent of θ ;
- !!! avoid pathological solutions (e.g., f identically constant);
- require, e.g., f to be monotone strictly increasing;
- the conditional distributions of $f(z)$ possibly not too bad.

Positive result: multiplicative normal

$$f(z) = \log |z|$$

Negative result: Bernoulli

Binary samples $z \in \{0, 1\}$ of the Bernoulli distribution with parameter $\theta = E\{z|\theta\}$ cannot be stabilized to the same constant variance for different values of θ :

$$\begin{aligned} E\{g(z) | \theta\} &= \theta g(1) + (1 - \theta) g(0) \\ \text{var}\{g(z) | \theta\} &= E\left\{(g(z) - E\{g(z) | \theta\})^2 | \theta\right\} = \\ &= (g(0) - g(1))^2 \theta(1 - \theta). \end{aligned}$$

Exact stabilization is not possible for Poisson, Binomial, and most other families used in applications.

Variance stabilization: history and examples

Classic **heuristic** stabilizer as indefinite integral form

$$f(z) = \int^z \frac{1}{\sigma(\theta)} d\mu(\theta). \quad (**)$$

Idea: consider a local first-order expansion of f at $\mu(\theta)$
(i.e., assume $\sigma(\theta)$ locally constant),

$$f(z) \simeq f(\mu(\theta)) + (z - \mu(\theta)) \frac{\partial f}{\partial z}(\mu(\theta)),$$

We have

$$\text{std}\{f(z) | \theta\} \simeq \frac{\partial f}{\partial z}(\mu(\theta)) \sigma(\theta),$$

then impose $\text{std}\{f(z) | \theta\} = c$ and obtain the indefinite integral (**).

Known and used already in the 1930's (e.g., Tippett 1934, Bartlett 1936), often rediscovered in signal processing (e.g., Prucnal&Saleh 1981, Arsenault&Denis 1981, Kasturi et al. 1983, Hirakawa&Parks 2006).

Variance stabilization: Poisson

$$f(z) = \int^z \frac{1}{\sigma(\theta)} d\mu(\theta) = \int^z \frac{1}{\sqrt{\theta}} d\mu(\theta) = 2\sqrt{z}.$$

Bartlett 1936: $2\sqrt{z + \frac{1}{2}}$

Anscombe 1948: $2\sqrt{z + \frac{3}{8}}$ (Anscombe attributes the result to A.H.L. Johnson)

Freeman&Tukey 1950: $\sqrt{z} + \sqrt{z + 1}$

In the same way stabilizers were derived for the Binomial and Negative Binomial distribution families (“angular” transformations based on the arcsin and hyperbolic arcsin).

Variance stabilization: Poisson

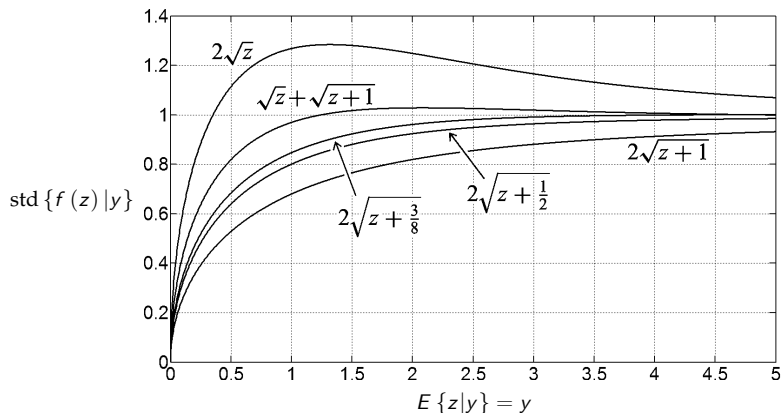


Figure: Conditional standard-deviation $\text{std}\{f(z)|y\}$ of the transformed Poisson variables z with parameter y after stabilization by five root-type transformations.

Denoising Poisson-count images

One approach is to use the following three-step procedure:

One approach is to use the following three-step procedure:

- Stabilize the noise variance by applying the Anscombe transformation.

One approach is to use the following three-step procedure:

- Stabilize the noise variance by applying the Anscombe transformation.
- Denoise with an algorithm designed for AWGN.

One approach is to use the following three-step procedure:

- Stabilize the noise variance by applying the Anscombe transformation.
- Denoise with an algorithm designed for AWGN.
- Apply an inverse transformation to the denoised image.

- We observe pixel values $z_i, i = 1, \dots, N$ (=noisy data).

- We observe pixel values $z_i, i = 1, \dots, N$ (=noisy data).
- We consider each z_i to be an independent random Poisson variable, whose mean y_i we want to estimate.

- We observe pixel values $z_i, i = 1, \dots, N$ (=noisy data).
- We consider each z_i to be an independent random Poisson variable, whose mean y_i we want to estimate.
- Variance is data-dependent:

$$E\{z_i | y_i\} = y_i = \text{var}\{z_i | y_i\}.$$

- Poisson noise is defined as

$$\eta_i = z_i - E\{z_i \mid y_i\}.$$

- Poisson noise is defined as

$$\eta_i = z_i - E\{z_i \mid y_i\}.$$

- Thus, we have

$$\text{var}\{\eta_i \mid y_i\} = \text{var}\{z_i \mid y_i\} = y_i.$$

- Poisson noise is defined as

$$\eta_i = z_i - E\{z_i \mid y_i\}.$$

- Thus, we have

$$\text{var}\{\eta_i \mid y_i\} = \text{var}\{z_i \mid y_i\} = y_i.$$

- We want to remove this data-dependence by a variance-stabilizing transform.

- Forward Anscombe transformation:

$$f(z) = 2\sqrt{z + \frac{3}{8}}.$$

Anscombe transformation

- Forward Anscombe transformation:

$$f(z) = 2\sqrt{z + \frac{3}{8}}.$$

- Applying Anscombe f to Poisson distributed data produces a signal whose noise is asymptotically additive standard normal (i.e. of unitary variance).

Anscombe transformation

- Forward Anscombe transformation:

$$f(z) = 2\sqrt{z + \frac{3}{8}}.$$

- Applying Anscombe f to Poisson distributed data produces a signal whose noise is asymptotically additive standard normal (i.e. of unitary variance).
- In other words, the transformation is both (asymptotically) variance-stabilizing and normalizing.

Inverse Anscombe transformation

- After applying the forward Anscombe transformation we denoise the signal $f(z)$ with e.g. BM3D, SAFIR or BLS-GSM, thus obtaining a signal D .

Inverse Anscombe transformation

- After applying the forward Anscombe transformation we denoise the signal $f(z)$ with e.g. BM3D, SAFIR or BLS-GSM, thus obtaining a signal D .
- We consider D to be an estimate of $E\{f(z) \mid y\}$.

Inverse Anscombe transformation

- After applying the forward Anscombe transformation we denoise the signal $f(z)$ with e.g. BM3D, SAFIR or BLS-GSM, thus obtaining a signal D .
- We consider D to be an estimate of $E\{f(z) | y\}$.
- We need to apply an inverse transformation to D in order to obtain the wanted estimate of y .

Inverse Anscombe transformation

- The direct algebraic inverse of Anscombe f is

$$\mathcal{I}_A(D) = f^{-1}(D) = \left(\frac{D}{2}\right)^2 - \frac{3}{8}.$$

- The direct algebraic inverse of Anscombe f is

$$\mathcal{I}_A(D) = f^{-1}(D) = \left(\frac{D}{2}\right)^2 - \frac{3}{8}.$$

- This leads to a biased estimate of y , because f is nonlinear:

$$E\{f(z) \mid y\} \neq f(E\{z \mid y\}).$$

- The direct algebraic inverse of Anscombe f is

$$\mathcal{I}_A(D) = f^{-1}(D) = \left(\frac{D}{2}\right)^2 - \frac{3}{8}.$$

- This leads to a biased estimate of y , because f is nonlinear:

$$E\{f(z) \mid y\} \neq f(E\{z \mid y\}).$$

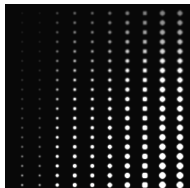
- Another possibility is to use

$$\mathcal{I}_B(D) = \left(\frac{D}{2}\right)^2 - \frac{1}{8},$$

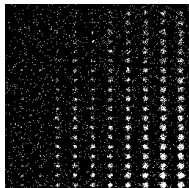
which provides *asymptotical unbiasedness* for large counts.

Asymptotically unbiased inverse

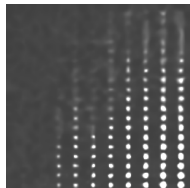
- Spots (intensity range $[0.03 \ 5.02]$), denoised with BM3D and inverted with the asymptotically unbiased inverse.



original



noisy



estimate

Exact unbiased inverse

- Applying the asymptotically unbiased inverse to high-count data gives good results, but for low-count data it produces a biased estimate.

Mäkitalo, M., and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising", to appear in IEEE Trans. Image Process., 2010.

Exact unbiased inverse

- Applying the asymptotically unbiased inverse to high-count data gives good results, but for low-count data it produces a biased estimate.
- Assuming the denoising was successful (D can be treated as $E\{f(z) | y\}$), we can solve the problem by finding an inverse transformation \mathcal{I}_C that maps the values $E\{f(z) | y\}$ to the desired values $E\{z | y\}$:

$$\mathcal{I}_C : E\{f(z) | y\} \mapsto E\{z | y\}.$$

Mäkitalo, M., and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising", to appear in IEEE Trans. Image Process., 2010.

Exact unbiased inverse

- Applying the asymptotically unbiased inverse to high-count data gives good results, but for low-count data it produces a biased estimate.
- Assuming the denoising was successful (D can be treated as $E\{f(z) | y\}$), we can solve the problem by finding an inverse transformation \mathcal{I}_C that maps the values $E\{f(z) | y\}$ to the desired values $E\{z | y\}$:

$$\mathcal{I}_C : E\{f(z) | y\} \mapsto E\{z | y\}.$$

- For any given y , $E\{z | y\} = y$, but we also need to compute the values of $E\{f(z) | y\}$.

Mäkitalo, M., and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising", to appear in IEEE Trans. Image Process., 2010.

- Expected value:

$$E\{f(z) \mid y\} = \int_{-\infty}^{+\infty} f(z)p(z \mid y) dz.$$

- Expected value:

$$E\{f(z) | y\} = \int_{-\infty}^{+\infty} f(z)p(z | y) dz.$$

- We have discrete Poisson probabilities $P(z | y)$, so

$$E\{f(z) | y\} = \sum_{z=0}^{+\infty} f(z)P(z | y).$$

Exact unbiased inverse

- Expected value:

$$E\{f(z) | y\} = \int_{-\infty}^{+\infty} f(z)p(z | y) dz.$$

- We have discrete Poisson probabilities $P(z | y)$, so

$$E\{f(z) | y\} = \sum_{z=0}^{+\infty} f(z)P(z | y).$$

- Explicitly:

$$E\{f(z) | y\} = 2 \sum_{z=0}^{+\infty} \left(\sqrt{z + \frac{3}{8}} \cdot \frac{y^z e^{-y}}{z!} \right).$$

Exact unbiased inverse

- In practice we do the summation over specific values of z to keep the error very small.

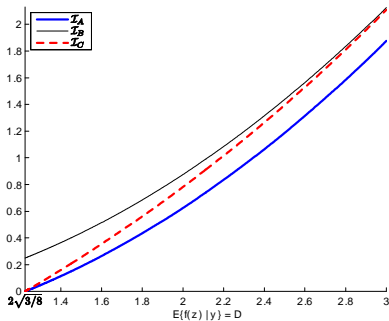
- In practice we do the summation over specific values of z to keep the error very small.
- We compute the values $E\{f(z) \mid y\}$ for a limited set of values y : for arbitrary values of y we use linear interpolation and for large values of y we approximate \mathcal{I}_C by \mathcal{I}_B .

Exact unbiased inverse

- In practice we do the summation over specific values of z to keep the error very small.
- We compute the values $E\{f(z) \mid y\}$ for a limited set of values y : for arbitrary values of y we use linear interpolation and for large values of y we approximate \mathcal{I}_C by \mathcal{I}_B .
- Matlab files implementing the exact unbiased inverse are available online at <http://www.cs.tut.fi/~foi/invansc>.

Exact unbiased inverse

- At low counts the asymptotically unbiased inverse actually leads to a larger bias than the algebraic inverse:



- The exact unbiased inverse assumes that the denoising is perfectly successful: we treat D as $E\{f(z) \mid y\}$.

- The exact unbiased inverse assumes that the denoising is perfectly successful: we treat D as $E\{f(z) \mid y\}$.
- Now assume instead that the pointwise MSE of D as an estimate of $E\{f(z) \mid y\}$ is

$$\varepsilon^2 = E \left\{ (D - E\{f(z) \mid y\})^2 \right\}.$$

- The exact unbiased inverse assumes that the denoising is perfectly successful: we treat D as $E\{f(z) | y\}$.
- Now assume instead that the pointwise MSE of D as an estimate of $E\{f(z) | y\}$ is

$$\varepsilon^2 = E \left\{ (D - E\{f(z) | y\})^2 \right\}.$$

- In practice the distribution of D is unknown. For simplicity, assume normality:

$$D \sim \mathcal{N} (E\{f(z) | y\}, \varepsilon^2).$$

- The exact unbiased inverse assumes that the denoising is perfectly successful: we treat D as $E\{f(z) | y\}$.
- Now assume instead that the pointwise MSE of D as an estimate of $E\{f(z) | y\}$ is

$$\varepsilon^2 = E \left\{ (D - E\{f(z) | y\})^2 \right\}.$$

- In practice the distribution of D is unknown. For simplicity, assume normality:

$$D \sim \mathcal{N} (E\{f(z) | y\}, \varepsilon^2).$$

- Formally this implies that D is an unbiased estimate of $E\{f(z) | y\}$, however also unknown estimation-bias errors can be considered as contributors of ε^2 : the symmetry of the distribution about $E\{f(z) | y\}$ reflecting our uncertainty about the sign of the bias.

- By treating D as the data, the maximum likelihood (ML) inverse is defined as

$$\mathcal{I}_{\text{ML}}(D) = \arg \max_y p(D | y),$$

$$p(D | y) = \frac{1}{\sqrt{2\pi\epsilon^2}} e^{-\frac{1}{2\epsilon^2}(D - E\{f(z)|y\})^2}.$$

- By treating D as the data, the maximum likelihood (ML) inverse is defined as

$$\mathcal{I}_{\text{ML}}(D) = \arg \max_y p(D | y),$$

$$p(D | y) = \frac{1}{\sqrt{2\pi\epsilon^2}} e^{-\frac{1}{2\epsilon^2}(D - E\{f(z)|y\})^2}.$$

- Under the given assumptions,

$$\mathcal{I}_{\text{ML}}(D) = \begin{cases} \mathcal{I}_C(D), & \text{if } D \geq 2\sqrt{3/8} \\ 0, & \text{if } D < 2\sqrt{3/8}. \end{cases}$$

Thus, exact unbiased inverse is a ML inverse.

- By treating D as the data, the maximum likelihood (ML) inverse is defined as

$$\mathcal{I}_{\text{ML}}(D) = \arg \max_y p(D | y),$$

$$p(D | y) = \frac{1}{\sqrt{2\pi\varepsilon^2}} e^{-\frac{1}{2\varepsilon^2}(D - E\{f(z)|y\})^2}.$$

- Under the given assumptions,

$$\mathcal{I}_{\text{ML}}(D) = \begin{cases} \mathcal{I}_C(D), & \text{if } D \geq 2\sqrt{3/8} \\ 0, & \text{if } D < 2\sqrt{3/8}. \end{cases}$$

Thus, exact unbiased inverse is a ML inverse.

- $\mathcal{I}_{\text{ML}}(D)$ is independent of ε .

- By treating D as the data, the maximum likelihood (ML) inverse is defined as

$$\mathcal{I}_{\text{ML}}(D) = \arg \max_y p(D | y),$$

$$p(D | y) = \frac{1}{\sqrt{2\pi\varepsilon^2}} e^{-\frac{1}{2\varepsilon^2}(D - E\{f(z)|y\})^2}.$$

- Under the given assumptions,

$$\mathcal{I}_{\text{ML}}(D) = \begin{cases} \mathcal{I}_C(D), & \text{if } D \geq 2\sqrt{3/8} \\ 0, & \text{if } D < 2\sqrt{3/8}. \end{cases}$$

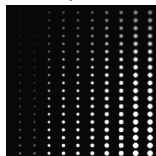
Thus, exact unbiased inverse is a ML inverse.

- $\mathcal{I}_{\text{ML}}(D)$ is independent of ε .
- Valid for any unimodal distribution with mode at $E\{f(z) | y\}$.

Test images

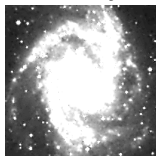
- All images 256x256, both low-count and high-count images included. The images are from Zhang, Fadili and Starck: Wavelets, ridgelets, and curvelets for Poisson noise removal (2008).

Spots



[0.03, 5.02]

Galaxy



[0, 5]

Ridges



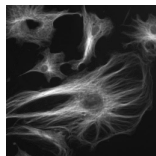
[0.05, 0.85]

Barbara



[0.93, 15.73]

Cells



[0.53, 16.93]

- Three steps: forward Anscombe transformation, denoising (BM3D/SAFIR/BLS-GSM), and the exact unbiased inverse.

- Three steps: forward Anscombe transformation, denoising (BM3D/SAFIR/BLS-GSM), and the exact unbiased inverse.
- The same is also done for the asymptotically unbiased inverse.

- Three steps: forward Anscombe transformation, denoising (BM3D/SAFIR/BLS-GSM), and the exact unbiased inverse.
- The same is also done for the asymptotically unbiased inverse.
- We evaluate the performance by normalized mean integrated square error (NMISE):

$$\frac{1}{\bar{N}} \sum_{i:y_i>0} ((\hat{y}_i - y_i)^2 / y_i), \quad (1)$$

where \hat{y}_i are the estimated intensities, y_i the respective true values, and the sum is computed over the \bar{N} pixels in the image for which $y_i > 0$.

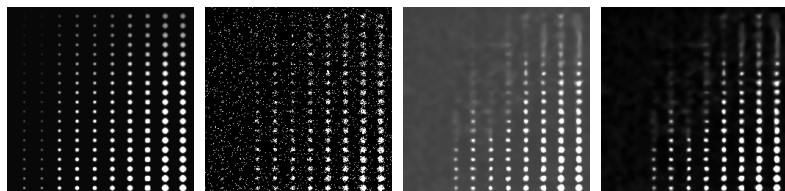
- Three steps: forward Anscombe transformation, denoising (BM3D/SAFIR/BLS-GSM), and the exact unbiased inverse.
- The same is also done for the asymptotically unbiased inverse.
- We evaluate the performance by normalized mean integrated square error (NMISE):

$$\frac{1}{\bar{N}} \sum_{i:y_i>0} ((\hat{y}_i - y_i)^2 / y_i), \quad (1)$$

where \hat{y}_i are the estimated intensities, y_i the respective true values, and the sum is computed over the \bar{N} pixels in the image for which $y_i > 0$.

- For each image we do five replications and present the average NMISE.

- Spots $[0.03, 5.02]$ denoised with BM3D.



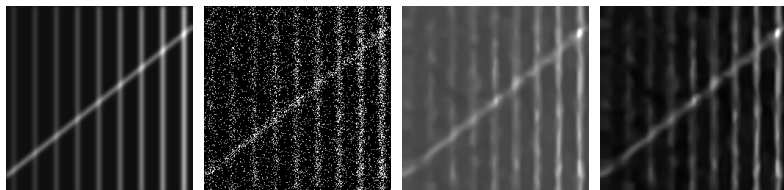
original

noisy

asymptotical

exact

- Ridges $[0.05, 0.85]$ denoised with BM3D.



original

noisy

asymptotical

exact

Numerical results (NMISE)

	Asymptotically unbiased inverse				Exact unbiased inverse			Other algorithms	
	WT	BM3D	SAFIR	BLS-GSM	BM3D	SAFIR	BLS-GSM	PH-HMT	MS-VST
Spots [0.03, 5.02]	2.34	1.7395	1.7495	2.0370	0.0365	0.0384	0.2024	0.048	0.069
Galaxy [0, 5]	0.15	0.1025	0.1110	0.1253	0.0299	0.0301	0.0385	0.030	0.035
Ridges [0.05, 0.85]	0.83	0.7018	0.7252	0.7694	0.0128	0.0173	0.0332	-	0.017
Barbara [0.93, 15.73]	0.26	0.0880	0.1178	0.1122	0.0880	0.1178	0.1123	0.159	0.17
Cells [0.53, 16.93]	0.095	0.0660	0.0683	0.0718	0.0649	0.0671	0.0707	0.082	0.078

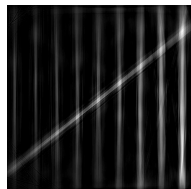
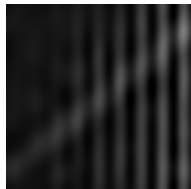
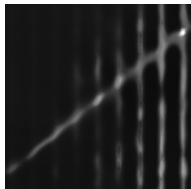
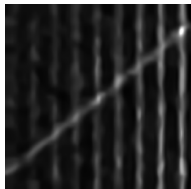
- We compare to two algorithms specifically designed for Poisson noise removal: MS-VST (Zhang, Fadili, Starck 2008) and PH-HMT (Lefkimmiatis, Maragos, Papandreou 2009).

Numerical results (NMISE)

	Asymptotically unbiased inverse				Exact unbiased inverse			Other algorithms	
	WT	BM3D	SAFIR	BLS-GSM	BM3D	SAFIR	BLS-GSM	PH-HMT	MS-VST
Spots [0.03, 5.02]	2.34	1.7395	1.7495	2.0370	0.0365	0.0384	0.2024	0.048	0.069
Galaxy [0, 5]	0.15	0.1025	0.1110	0.1253	0.0299	0.0301	0.0385	0.030	0.035
Ridges [0.05, 0.85]	0.83	0.7018	0.7252	0.7694	0.0128	0.0173	0.0332	-	0.017
Barbara [0.93, 15.73]	0.26	0.0880	0.1178	0.1122	0.0880	0.1178	0.1123	0.159	0.17
Cells [0.53, 16.93]	0.095	0.0660	0.0683	0.0718	0.0649	0.0671	0.0707	0.082	0.078

- We compare to two algorithms specifically designed for Poisson noise removal: MS-VST (Zhang, Fadili, Starck 2008) and PH-HMT (Lefkimmiatis, Maragos, Papandreou 2009).
- BM3D + exact unbiased inverse gives the best result for all five test images.

Comparisons

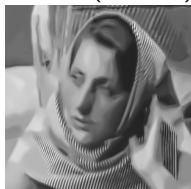


BM3D (0.0128)

SAFIR (0.0173)

BLS-GSM (0.0332)

MS-VST (0.017)



BM3D (0.0880)

SAFIR (0.1178)

BLS-GSM (0.1123)

MS-VST (0.17)

- The improvement from the asymptotically unbiased inverse to the exact unbiased inverse is significant for low-count images.

Summary

- The improvement from the asymptotically unbiased inverse to the exact unbiased inverse is significant for low-count images.
- Denoising with Anscombe and exact unbiased inverse is competitive with MS-VST and PH-HMT, which are specifically designed for Poisson noise removal.

- The improvement from the asymptotically unbiased inverse to the exact unbiased inverse is significant for low-count images.
- Denoising with Anscombe and exact unbiased inverse is competitive with MS-VST and PH-HMT, which are specifically designed for Poisson noise removal.
- Even though most of the improvement is due to the exact unbiased inverse, the choice of the denoising algorithm does also matter: BM3D outperforms all other filters.

- The improvement from the asymptotically unbiased inverse to the exact unbiased inverse is significant for low-count images.
- Denoising with Anscombe and exact unbiased inverse is competitive with MS-VST and PH-HMT, which are specifically designed for Poisson noise removal.
- Even though most of the improvement is due to the exact unbiased inverse, the choice of the denoising algorithm does also matter: BM3D outperforms all other filters.
- Approach is not limited to 2-D data.

- The improvement from the asymptotically unbiased inverse to the exact unbiased inverse is significant for low-count images.
- Denoising with Anscombe and exact unbiased inverse is competitive with MS-VST and PH-HMT, which are specifically designed for Poisson noise removal.
- Even though most of the improvement is due to the exact unbiased inverse, the choice of the denoising algorithm does also matter: BM3D outperforms all other filters.
- Approach is not limited to 2-D data.
- Matlab files implementing the exact unbiased inverse are available online at <http://www.cs.tut.fi/~foi/invansc>

Raw-data observation model

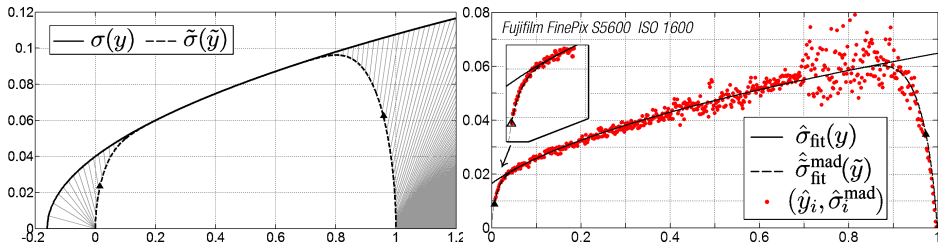
For imaging sensors (CCD or CMOS)

$$\text{var}\{z(x)\} = ay(x) + b, \quad y(x) \geq b/a, \quad a, b > 0.$$

For clipped data

$$\tilde{z}(x) = \max\{0, \min\{z(x), 1\}\}, \quad \tilde{y}(x) = E\{\tilde{z}(x)\} \neq y(x),$$

with standard deviation curves:



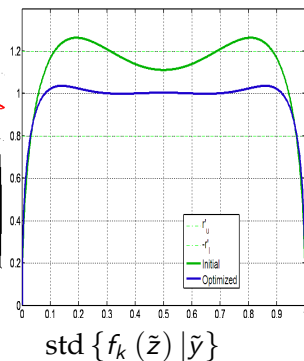
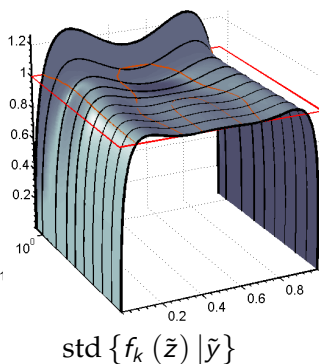
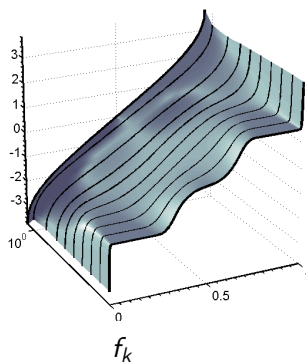
It is important to compute the functions \tilde{y} and $\tilde{\sigma}$ given σ and y , and vice versa.

ClipPoisGaus toolbox for Matlab

<http://www.cs.tut.fi/~foi/sensornoise.html>

- 1 Fully automatic estimation of noise parameters from a single image with clipped or non-clipped data corrupted by signal-dependent noise.
 - 2 Fully automatic denoising and debiasing of clipped images with Poissonian-Gaussian noise using variance-stabilization and homoskedastic filtering.
- Foi, A., "Clipped noisy images: heteroskedastic modeling and practical denoising", Signal Processing, vol. 89, no. 12, pp. 2609-2629, December 2009.
 - Foi, A., M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single image raw-data", IEEE Trans. Image Process., vol. 17, no. 10, pp. 1737-1754, October 2008.

Optimization of variance stabilization for raw data

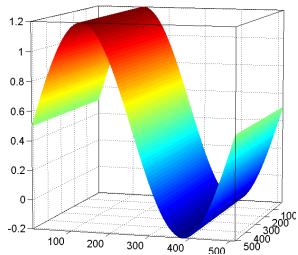


- Foi, A., "Direct optimization of nonparametric variance-stabilizing transformations", Proc. 8èmes Rencontres de Statistiques Mathématiques, CIRM Luminy, Marseille, France, December 2008.
- Foi, A., "Optimization of variance-stabilizing transformations", preprint, submitted.

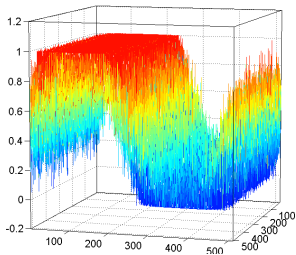
Algorithm consists from the following steps:

- 1 Noise estimation (optional);
- 2 Design and apply variance-stabilizing transformation;
- 3 Denoise the transformed data using some denoising algorithm for homoskedastic noise (e.g., BM3D);
- 4 Apply the exact unbiased inverse of the variance stabilizing transform;
- 5 Perform declipping.

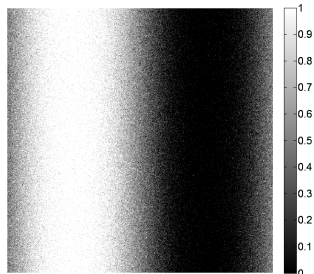
Examples



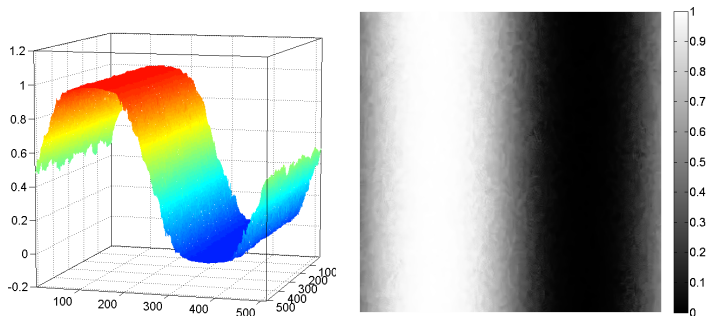
left: original (range $[-0.2, 1.2]$)



center+right: noisy and clipped (range $[0, 1]$)

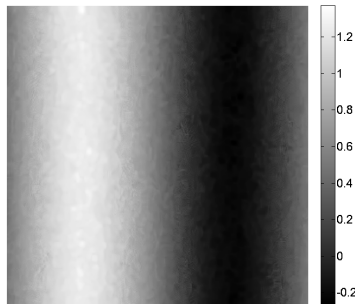
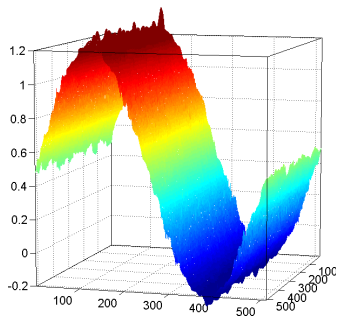


Examples



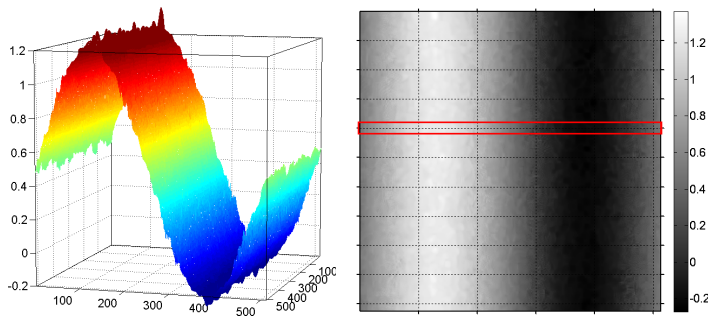
Denoising clipped data (range $[0, 1]$)

Examples



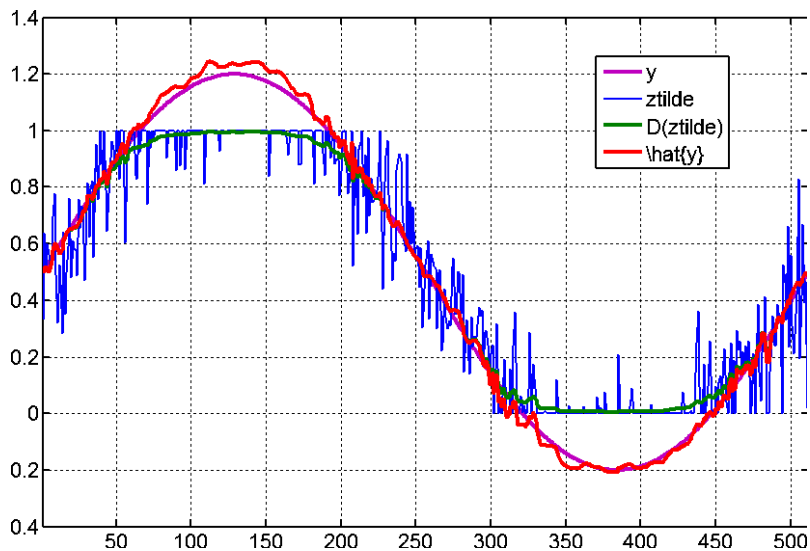
Denoising and declipping.

Examples



Denoising and declipping.

Examples



Cross-sections of observations and estimates.

Declipping: numerical results

Noise parameters	S1 $a=0.004, b=0.02^2$			S2 $a=1/30, b=0.1^2$			S3 $a=0, b=0.2^2$											
	<i>Testpat</i>			<i>Man</i>			<i>Testpat</i>			<i>Man</i>								
	<i>Testpat</i>	<i>Man</i>		<i>Testpat</i>	<i>Man</i>		<i>Testpat</i>	<i>Man</i>		<i>Testpat</i>	<i>Man</i>							
Noisy \tilde{z}	26.83			27.51			16.70			17.07			14.99			14.86		
	$\mathbf{D}(\tilde{z})$	\hat{y}	\hat{y}_0^1	$\mathbf{D}(\tilde{z})$	\hat{y}	\hat{y}_0^1	$\mathbf{D}(\tilde{z})$	\hat{y}	\hat{y}_0^1	$\mathbf{D}(\tilde{z})$	\hat{y}	\hat{y}_0^1	$\mathbf{D}(\tilde{z})$	\hat{y}	\hat{y}_0^1	$\mathbf{D}(\tilde{z})$	\hat{y}	\hat{y}_0^1
BM3D [5]	39.03	41.20	42.29	33.75	33.75	33.76	28.49	31.53	32.33	27.72	27.99	28.03	26.45	29.36	30.15	25.26	26.32	26.44
TLS [18]	32.50	31.32	32.95	33.48	33.46	33.49	25.54	25.69	26.78	27.28	27.36	27.50	23.90	24.26	25.23	24.72	25.24	25.54
K-SVD [7]	35.49	34.29	36.43	33.05	33.01	33.05	26.03	26.66	27.48	26.08	26.06	26.15	23.99	24.77	25.36	23.47	23.80	23.96
BLS-GSM [20]	33.26	28.78	33.72	33.58	33.57	33.59	24.64	22.52	25.63	26.85	26.96	27.04	22.89	21.45	23.99	24.16	24.79	24.96
SA-DCT hom.[11]	37.86	37.66	39.96	33.57	33.56	33.57	26.83	27.20	28.87	27.37	27.61	27.66	24.49	24.65	26.26	24.80	25.81	25.91
SA-DCT het. [12]	38.57	35.85	41.40	33.52	33.53	33.54	27.91	22.53	30.95	27.69	28.07	28.13	25.97	19.94	28.94	25.25	25.92	26.82

$PSNR$ (dB) values for the denoised $\mathbf{D}(\tilde{z})$, denoised and declipped \hat{y} , and range-constrained estimates.

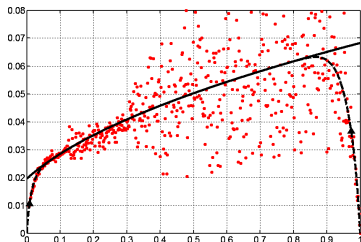
Examples: raw data



Raw data from Fujifilm FinePix S9600, ISO 1600

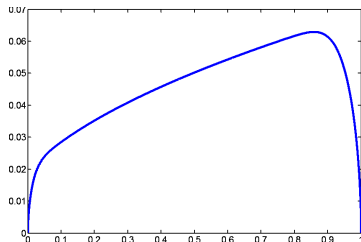
Examples: raw data

Noise estimation



estimation and fitting

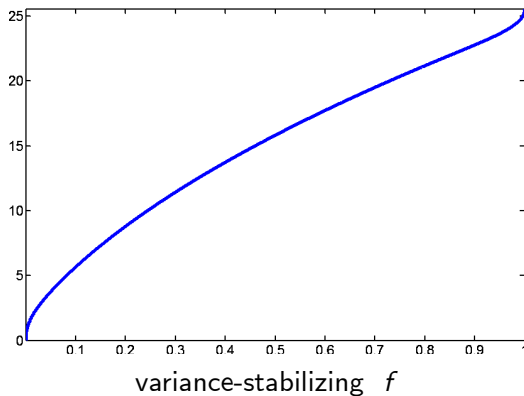
$$\hat{a} = 0.0043 \quad \hat{b} = 0.00038$$



st.dev.-function $\tilde{\sigma}$

Examples: raw data

Variance-stabilizing transformation



Examples: raw data

Denoising



Denoised estimate

Examples: raw data

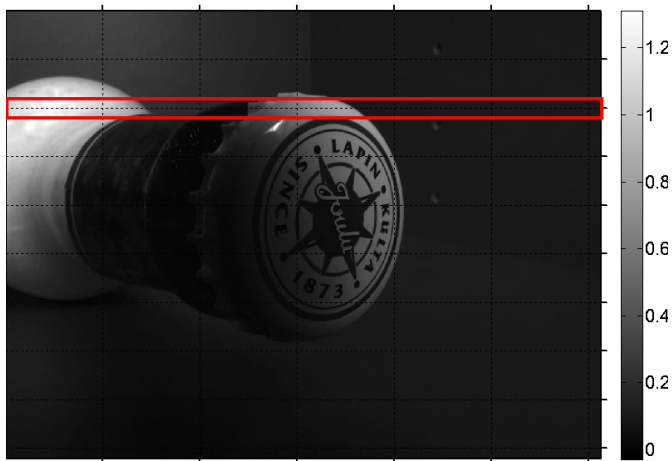
Declipping



Declipped estimate

Examples: raw data

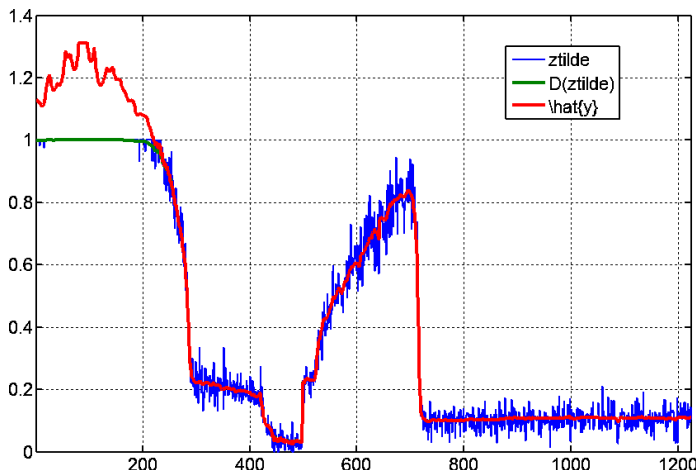
Declipping



Declipped estimate

Examples: raw data

Declipping



It implements a video denoising method that enables highly effective noise attenuation at near real-time execution times.

It is based on enhanced sparse representation in local 3D transform domain. As the noisy video is processed in block-wise manner, the sparsity enhancement is achieved by grouping 2D fragments similar to the current one into a 3D data array that we call group.

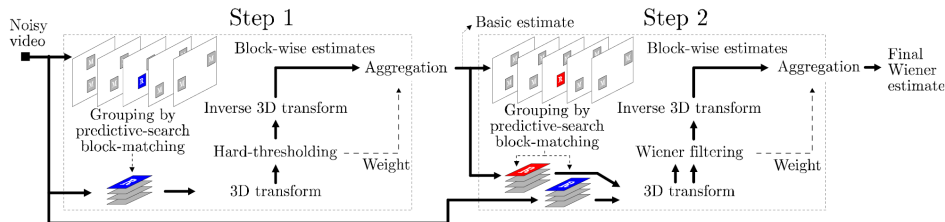
K. Dabov, et al., "Video denoising by sparse 3D transform-domain collaborative filtering," Proc. 15th European Signal Processing Conference, EUSIPCO 2007, Poznan, Poland, September 2007.

Video Processing with V-BM3D

- The grouping is realized as a predictive-search block-matching, similar to techniques used for motion estimation;
- For each formed group, we apply collaborative filtering in order to take advantage of the correlation between grouped blocks. We realize this filtering by a 3D transform-domain shrinkage (hard-thresholding and Wiener filtering). The collaborative filtering produces estimates of all grouped blocks.
- Since these estimates are overlapping in general, we aggregate them by a weighted average in order to form a non-redundant video estimate.
- A significant improvement of this approach is the use of 2-step algorithm where an intermediate estimate is produced by grouping and collaborative hard-thresholding and then used for improving the grouping and for applying collaborative empirical Wiener filtering.
- The experimental results demonstrate the state-of-the-art denoising performance and subjective visual quality.

Video Processing with V-BM3D

Flowchart of the algorithm



Flowchart of the V-BM3D video-denoising algorithm.

The operation enclosed by dashed lines are repeated for each reference block. Grouping is illustrated by showing a reference block marked with 'R' and the matched ones in a temporal window of 5 frames.

Video Processing with V-BM3D

Predictive-search block-matching

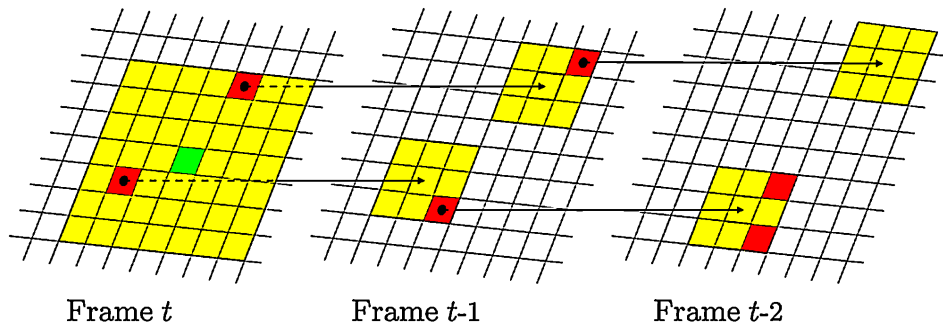


Illustration of the proposed *predictive-search block-matching*.

Each pixel represents a block located at it; grey pixels denote blocks that are part of the search neighborhood; red pixels denote blocks matched as similar.

Video Processing with V-BM3D

Predictive-search block-matching

- For frame t , which contains the reference block, the search neighborhood is (non-adaptive) fixed around the reference block.
- For frame $t \pm 1$, the blocks matched as similar to in the frame t are used to determine the centers of the small 3×3 neighborhoods, whose union forms the overall adaptive neighborhood for this frame.
- For frame $t \pm 2$, the motion compensated predictive search uses the motion vectors (shown as white arrows) between the matched blocks of the previous two frames to determine the centers of the 3×3 neighborhoods.

BUS!

Flowers!

Man!

Compressive sensing (CS): conventional approach

- The basic setting: an unknown signal of interest is observed (sensed) through a limited number linear functionals;
- These observations can be considered as an incomplete portion of the spectrum of the signal with respect to a given linear transform;
- It is assumed that the signal can be represented sparsely with respect to a different relevant basis (e.g., wavelets);
- The algorithms rely on convex optimization with a penalty expressed by the ℓ_0 or ℓ_1 norms which is exploited to enable the assumed sparsity;
- It results in parametric modeling of the solution and in problems that are then solved by mathematical programming algorithms.

Candes, E., J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information", IEEE Trans. Inf. Theory, vol. 52, no. 2, 2006.

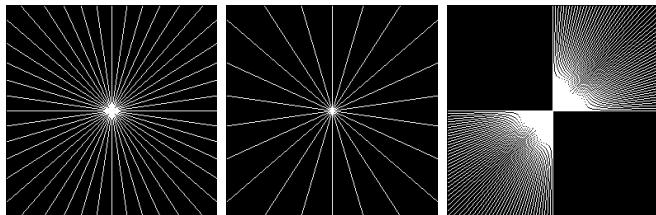
Compressive sensing (CS): our approach

- We replace the traditional parametric modeling used in CS by a nonparametric one.
- The nonparametric modeling is implemented by the use of spatially adaptive filters.

The logic behind of this approach is as follows.

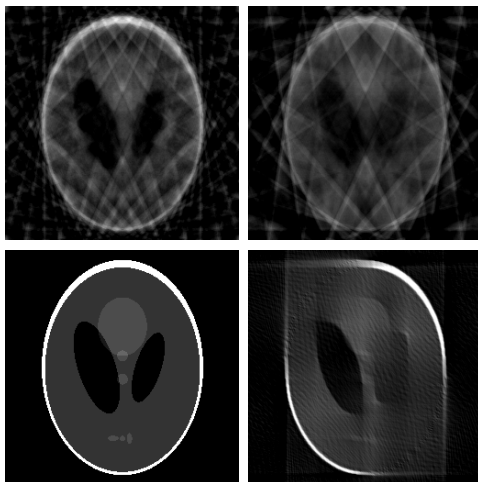
- The regularization imposed by the ℓ_0 or ℓ_1 norms (or by more general criteria) is essentially only as a tool for design of some nonlinear filtering.
- Let us replace this implicit regularization by explicit filtering, exploiting spatially adaptive filters sensitive to image features and details.
- Then, CS signal reconstruction is realized by a recursive algorithm based on spatially adaptive image denoising.

Examples: observations



Sample domain Ω for the FFT spectrum: 22 radial lines, 11 radial lines, 90 degrees limited-angle with 61 radial lines.

Examples: reconstructions



Clockwise from top-left: back-projection estimates for 22 radial lines, 11 radial lines, 61 radial lines with limited-angle (90 degrees), and original phantom (unknown and shown here only as a reference). For all three experiments, the estimates obtained after convergence of the algorithm coincide with the original image.

Observation model

Let y and $\theta = \mathcal{T}\{y\}$ be, respectively, the image intensity and its 2-D transform.

If all elements of the spectrum θ are given then the signal can be recovered by inverting the transform, $y = \mathcal{T}^{-1}\{\theta\}$.

In CS problems only a small portion of the spectrum is available, which makes the reconstruction of y an ill-posed problem.

Observation model (cont)

Introduce a sampling operator as the characteristic function $S = \chi_{\Omega}$ (with values 0 or 1) of the available portion Ω of the spectrum.

Thus, the pointwise products $S.*\theta$ and $(1 - S).*\theta$ produce a decomposition of the spectrum in two complementary parts

$$\theta_1 = S.*\theta, \quad \theta_2 = (1 - S).*\theta,$$

with the equation

$$\theta = \theta_1 + \theta_2 = S.*\theta + (1 - S).*\theta.$$

Here, θ_1 and θ_2 are the observed (known) and the unobserved (unknown) part of θ , respectively.

The goal is reconstruct y (or equivalently θ_2) from the available data θ_1 .

K. Egiazarian et al., "Compressed sensing image reconstruction via recursive spatially adaptive filtering," ICIP, 2007.

Recursive algorithm

Given an estimate $\hat{\theta}_2^{(k)}$ of θ_2 , we define the estimate $\hat{\theta}^{(k)}$ of θ as $\hat{\theta}^{(k)} = \theta_1 + \hat{\theta}_2^{(k)}$.

The algorithm is as follows:

$$\hat{\theta}_2^{(k)} = \hat{\theta}_2^{(k-1)} - \gamma_k [\hat{\theta}_2^{(k-1)} - (1 - S) .* \mathcal{T} (\Phi (\mathcal{T}^{-1} (\theta_1 + \hat{\theta}_2^{(k-1)})))] + (1 - S) .* \eta_k], \quad k = 1, 2, \dots, \quad \hat{\theta}_2^{(0)} = 0.$$

Each iteration ($k \geq 1$) comprises of the following steps:

- Image-domain estimate filtering

$$\Phi (\mathcal{T}^{-1} (\theta_1 + \hat{\theta}_2^{(k-1)}));$$

- Excitation by random η_k .

The recursive algorithm can be treated as the Robbins-Monro stochastic approximation procedure for the equation

$$\hat{\theta}_2 - (1 - S) .* \mathcal{T} (\Phi (\mathcal{T}^{-1} (\theta_1 + \hat{\theta}_2))) = 0.$$

The noise η_k serves as a generator of the spectrum θ_2 features and for acceleration of the convergence.

If $\Phi = I$ then any $\hat{\theta}_2$ satisfies this equation.

Compressed sensing for image Upsampling (Resizing)

We do not know which blurring and decimation operators have been used to obtain the given image.

Instead, we assume that the blurring kernel is the low-pass analysis filter of a wavelet transform.

Hence, we seek for a high-resolution image whose wavelet approximation coefficients in the lower resolution subband decomposition coincide to the pixel values of the given low-resolution image.

A high-resolution image is reconstructed by alternating two procedures: spatially adaptive filtering and projection on the observation-constrained subspace.

The Block Matching and 3D filtering (BM3D) technique is used to suppress ringing, and reconstruct missing wavelet detail coefficients.

Image Resizing with BM3D Examples

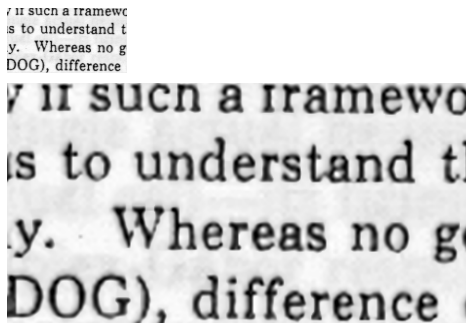


Upsampling with m-stage algorithm: Cameraman, 4 times



Upsampling with m-stage algorithm:Lighthouse, 8 times.

Image Resizing with BM3D Examples



Upsampling of fragments of the images with m-stage algorithm: Text, 4 times.

Image and Video Super-Resolution with BM3D

The classical SR approach is based on three steps:

- 1) registration of the LR images to a HR coordinate grid,
- 2) warping of the LR images onto that grid by interpolation,
- 3) fusion of the warped images into the final HR image.

An additional deblurring step is sometimes considered to compensate the blur existing in the LR frames.

- The novel algorithm is developed generalizing VBM3D.m for super-resolution imaging.

Danielyan, A., A. Foi, V. Katkovnik, and K. Egiazarian, "Image and video superresolution via spatially adaptive filtering," in Proc. 2008 Int. Workshop on Local and Non-Local Approximation in Image Processing, LNLA 2008, Lausanne, Switzerland, 2008.

Danielyan, A., A. Foi, V. Katkovnik, and K. Egiazarian, "Spatially adaptive filtering as regularization in inverse imaging: compressive sensing, upsampling, and super-resolution", in Super-Resolution Imaging (P. Milanfar, ed.) CRC Press / Taylor & Francis, 2010.

Image and Video Super-Resolution with BM3D

Preliminaries

- Let $\{\mathcal{T}_m\}_{m=0}^M$ be a family of orthonormal transforms of increasing sizes $x_m^h \times x_m^v$, $x_m^h < x_{m+1}^h$, $x_m^v < x_{m+1}^v$, such that for any pair m, m' with $m < m'$, up to a scaling factor $\beta_{m,m'} = \sqrt{\frac{x_{m'}^h x_{m'}^v}{x_m^h x_m^v}}$, the whole \mathcal{T}_m -spectrum can be considered as a smaller portion of the $\mathcal{T}_{m'}$ -spectrum.

Image and Video Super-Resolution with BM3D

Preliminaries

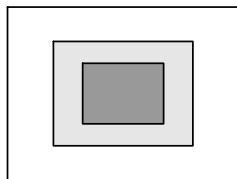
- Let $\{\mathcal{T}_m\}_{m=0}^M$ be a family of orthonormal transforms of increasing sizes $x_m^h \times x_m^v$, $x_m^h < x_{m+1}^h$, $x_m^v < x_{m+1}^v$, such that for any pair m, m' with $m < m'$, up to a scaling factor $\beta_{m,m'} = \sqrt{\frac{x_{m'}^h x_{m'}^v}{x_m^h x_m^v}}$, the whole \mathcal{T}_m -spectrum can be considered as a smaller portion of the $\mathcal{T}_{m'}$ -spectrum.
- This means that the supports Ω_m of the \mathcal{T}_m -transform coefficients form a nested sequence of subsets of Ω_M , i.e. $\Omega_0 \subset \dots \subset \Omega_M$, where Ω_M is a complete set of the coefficients.

Image and Video Super-Resolution with BM3D

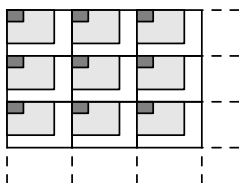
Preliminaries

- Let $\{\mathcal{T}_m\}_{m=0}^M$ be a family of orthonormal transforms of increasing sizes $x_m^h \times x_m^v$, $x_m^h < x_{m+1}^h$, $x_m^v < x_{m+1}^v$, such that for any pair m, m' with $m < m'$, up to a scaling factor $\beta_{m,m'} = \sqrt{\frac{x_{m'}^h x_{m'}^v}{x_m^h x_m^v}}$, the whole \mathcal{T}_m -spectrum can be considered as a smaller portion of the $\mathcal{T}_{m'}$ -spectrum.
- This means that the supports Ω_m of the \mathcal{T}_m -transform coefficients form a nested sequence of subsets of Ω_M , i.e. $\Omega_0 \subset \dots \subset \Omega_M$, where Ω_M is a complete set of the coefficients.
- The examples of such $\{\mathcal{T}_m\}_{m=0}^M$ families are DCT and DFT transforms of different sizes, discrete wavelet transforms associated to one common scaling function, as well as block-wise DCT, DFT and WT. Sets Ω_m are commonly referred to as lower-resolution, low-frequency, or coarser-scale subbands of the \mathcal{T}_M -spectrum.

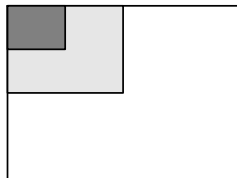
Image and Video Super-Resolution: nested spectrum sets



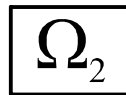
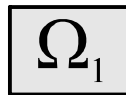
(a) DFT



(b) Block DFT



(c) DWT, DCT



Nested support subsets.

- For $m < m'$ the three operators are defined:
the restriction operator $|_{\Omega_{m,m'}}$ that, from a given $\mathcal{T}_{m'}$ -spectrum, extracts a smaller portion defined on Ω_m , which can be considered as the \mathcal{T}_m -spectrum of a smaller image;

Image and Video Super-Resolution: basic operators

- For $m < m'$ the three operators are defined:
 - the restriction operator $|_{\Omega_{m,m'}}$ that, from a given $\mathcal{T}_{m'}$ -spectrum, extracts a smaller portion defined on Ω_m , which can be considered as the \mathcal{T}_m -spectrum of a smaller image;
 - the zero-padding operator $\mathcal{U}_{m,m'}$ that expands a \mathcal{T}_m -spectrum defined on Ω_m to the $\mathcal{T}_{m'}$ -spectrum defined on the superset $\Omega_{m'} \supset \Omega_m$ by introducing zeros in the complementary $\Omega_{m'} \setminus \Omega_m$;

- For $m < m'$ the three operators are defined:
 - the restriction operator $|_{\Omega_{m,m'}}$ that, from a given $\mathcal{T}_{m'}$ -spectrum, extracts a smaller portion defined on Ω_m , which can be considered as the \mathcal{T}_m -spectrum of a smaller image;
 - the zero-padding operator $\mathcal{U}_{m,m'}$ that expands a \mathcal{T}_m -spectrum defined on Ω_m to the $\mathcal{T}_{m'}$ -spectrum defined on the superset $\Omega_{m'} \supset \Omega_m$ by introducing zeros in the complementary $\Omega_{m'} \setminus \Omega_m$;
 - the projection operator $P_{m,m'}^\perp$ that zeroes all coefficients of $\mathcal{T}_{m'}$ -spectrum on Ω_m .

- For $m < m'$ the three operators are defined:
 - the restriction operator $|_{\Omega_{m,m'}}$ that, from a given $\mathcal{T}_{m'}$ -spectrum, extracts a smaller portion defined on Ω_m , which can be considered as the \mathcal{T}_m -spectrum of a smaller image;
 - the zero-padding operator $\mathcal{U}_{m,m'}$ that expands a \mathcal{T}_m -spectrum defined on Ω_m to the $\mathcal{T}_{m'}$ -spectrum defined on the superset $\Omega_{m'} \supset \Omega_m$ by introducing zeros in the complementary $\Omega_{m'} \setminus \Omega_m$;
 - the projection operator $P_{m,m'}^\perp$ that zeroes all coefficients of $\mathcal{T}_{m'}$ -spectrum on Ω_m .
- Note that $\mathcal{U}_{m,m'}(A)|_{\Omega_m} = A$ for any \mathcal{T}_m -spectrum A , and $B = P_{m,m'}^\perp(B) + \mathcal{U}_{m,m'}(B|_{\Omega_m})$ for any $\mathcal{T}_{m'}$ -spectrum B .

Observation model and superresolution as compressed sensing problem

Let a sequence of low-resolution images be given $\{y_{\text{low } r}\}_{r=1}^R$ of the sizes $x_0^h \times x_0^v$, and assume each $y_{\text{low } r}$ being obtained from the subband of the corresponding T_M spectra of original higher-resolution images $\{y_{\text{hi } r}\}_{r=1}^R$ of size $x_M^h \times x_M^v$ as follows:

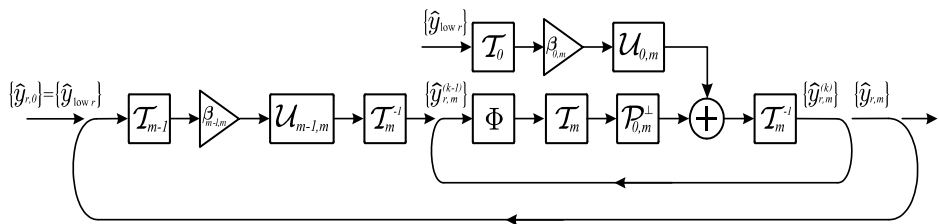
$$y_{\text{low } r} = \mathcal{T}_0^{-1} \left(\beta_{0,M}^{-1} \mathcal{T}_M (y_{\text{hi } r})|_{\Omega_{0,M}} \right),$$

where the scaling factor $\beta_{0,M}$ ensures that the means of $y_{\text{hi } r}$ and $y_{\text{low } r}$ are equal to each other.

The superresolution problem is to reconstruct $\{y_{\text{hi } r}\}_{r=1}^R$ from $\{y_{\text{low } r}\}_{r=1}^R$. For $R = 1$, the observation model corresponds to the image upsampling problem.

$$\left\{ \begin{array}{l}
 \hat{y}_{r,0} = y_{\text{low } r}, \quad r = 1, \dots, R, \\
 \text{for } m = 1 : M \\
 \hat{y}_{r,m}^{(0)} = \mathcal{T}_m^{-1} \left(\mathcal{U}_{m-1,m} \left(\beta_{m-1,m} \mathcal{T}_{m-1} \left(\hat{y}_{r,m-1} \right) \right) \right), \quad r = 1, \dots, R, \\
 \\
 \text{for } k = 1 : k_{\text{final}} \\
 \hat{y}_{r,m}^{(k)} = \mathcal{T}_m^{-1} \left(\begin{array}{l}
 \mathcal{U}_{0,m} \left(\beta_{0,m} \mathcal{T}_0 \left(y_{\text{low } r} \right) \right) + \\
 + \mathcal{P}_{0,m}^{\perp} \left(\mathcal{T}_m \left(\Phi \left[\left\{ \hat{y}_{r,m}^{(k-1)} \right\}_{r=1}^R \right] \right) \right) \end{array} \right), \\
 r = 1, \dots, R, \\
 \text{end for } k \\
 \hat{y}_{r,m} = \hat{y}_{r,m}^{(k_{\text{final}})}, \quad r = 1, \dots, R, \\
 \text{end for } m \\
 \hat{y}_r = \hat{y}_{r,M}^{(k_{\text{final}})}, \quad r = 1, \dots, R.
 \end{array} \right.$$

Flowchart of the algorithm



Experiments: Image and Video Super-Resolution

In all these experiments, the LR image is obtained from the HR one by first blurring using a 3×3 uniform kernel, shifted and then decimating by factor 3. It gives a set of nine shifted sampled LR versions of HR blurred image.

The comparison is done vs: M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the Non-Local-Means to Super-Resolution Reconstruction", IEEE TIP, Vol. 18, No. 1, Jan. 2009.

Image and Video Super-Resolution with BM3D: Examples



Nearest neighbor



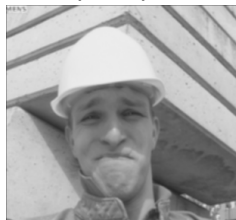
Super-resolved



Upsampled



Ground truth



Ground truth (blurred)

Image and Video Super-Resolution with BM3D Examples: Suzie



Nearest neighbor



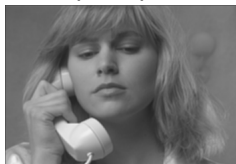
Super-resolved



Upsampled



Ground truth



Ground truth (blurred)

Super-resolution results for the 23rd frame of Foreman



Clockwise from top left: pixel-replicated low resolution image; original image (ground truth); super-resolved by proposed algorithm, super-resolved by Protter et al. algorithm.

Super-resolution results for the 23rd frame of Suzie



Clockwise from top left: pixel-replicated low resolution image; original image (ground truth); super-resolved by proposed algorithm, super-resolved by Protter et al. algorithm.

Super-resolution results for the 23rd frame of Miss America

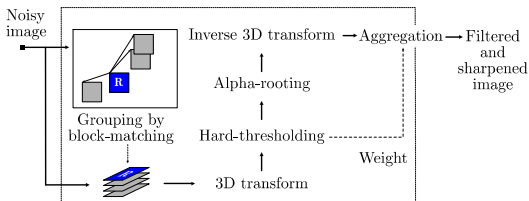


Clockwise from top left: pixel-replicated low resolution image; original image (ground truth); super-resolved by proposed algorithm, super-resolved by Protter et al. algorithm.

	Nearest neighbor	Protter et al.	Proposed
	PSNR	PSNR	PSNR
Foreman	29.0	32.9	35.0
Suzie	30.3	33.0	34.2
Miss America	32.0	34.74	37.0

Mean over all 30 frames

BM3D Denoising and Alpha-Root Sharpener



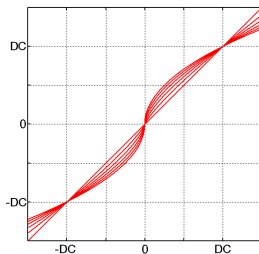
- 1 Use block-matching to find the locations of the blocks in z that are similar to the currently processed one.
 - 1 Apply a 3D transform on the formed group;
 - 2 Attenuate the noise by hard-thresholding the 3D transform spectrum;
 - 3 Apply alpha rooting on the hard-thresholded 3D transform spectrum and invert the 3D transform to produce filtered grouped blocks;
- 2 Return the filtered blocks to their original locations in the image domain and compute the resultant filtered image as a weighted average of these filtered blocks.

Collaborative sharpening by Alpha Rooting

Except for the alpha rooting and a modification of the aggregation weights, both described below, the rest of the steps of the algorithm are taken without modification from the first step of BM3D.

Given a transform spectrum of a signal, which contains a DC coefficient denoted as $\hat{\theta}_s(1, 1)$, the alpha rooting (Aghagolzadeh&Ersoy, 1992) is performed as

$$\hat{\theta}_s^{sh}(i, j) = \begin{cases} \text{sign} \hat{\theta}_s(i, j) |\hat{\theta}_s(1, 1)| \left| \frac{\hat{\theta}_s(i, j)}{\hat{\theta}_s(1, 1)} \right|^{\frac{1}{\alpha}}, & \text{if } \hat{\theta}_s(i, j) \neq 0 \\ \hat{\theta}_s(i, j), & \text{otherwise.} \end{cases}$$



Collaborative sharpening by Alpha Rooting

Aggregation weights for sharpening

Variance of sharpened coefficients (using first order approximations)

$$\begin{aligned} \text{var} \left\{ \hat{\theta}_s^{sh}(i, j) \right\} &\approx \omega_{i,j} \sigma^2 = \\ &= \left(1 - \frac{1}{\alpha} \right)^2 |\hat{\theta}_s(1, 1)|^{-\frac{2}{\alpha}} |\hat{\theta}_s(i, j)|^{\frac{2}{\alpha}} \sigma^2 + \frac{1}{\alpha^2} |\hat{\theta}_s(i, j)|^{\frac{2}{\alpha}-2} |\hat{\theta}_s(1, 1)|^{2-\frac{2}{\alpha}} \sigma^2. \end{aligned}$$

The total variance of the thresholded and sharpened group $\hat{\mathbf{Y}}_{x_R}^{\text{sharp}}$ is approximated as

$$\text{tsvar} \left\{ \hat{\mathbf{Y}}_{x_R}^{\text{sharp}} \right\} = \sigma^2 + \sum_{\hat{\theta}_s(i,j) \neq 0, (i,j) \neq (1,1)} \omega_{i,j} \sigma^2.$$

Hence, the aggregation weights are $w_{x_R} = \frac{1}{\text{tsvar} \left\{ \hat{\mathbf{Y}}_{x_R}^{\text{sharp}} \right\}}.$

BM3D sharpening experiments

Noisy *House*, $\sigma = 10$



BM3D sharpening experiments

BM3D-SH3D, $\alpha = 1.2$



BM3D sharpening experiments

BM3D-SH3D, $\alpha = 1.4$



BM3D sharpening experiments

BM3D-SH3D, $\alpha = 1.6$



BM3D sharpening experiments

BM3D-SH3D, $\alpha = 1.8$



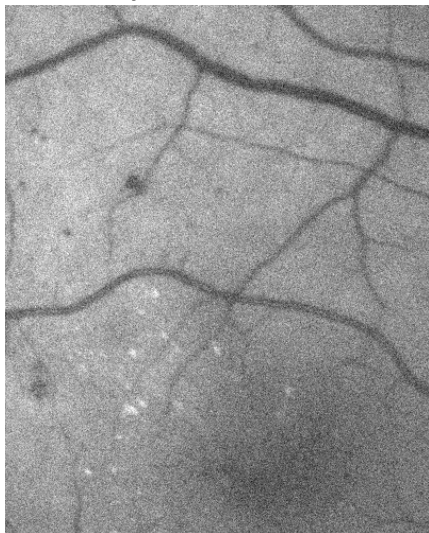
BM3D sharpening experiments

BM3D-SH3D, $\alpha = 2.0$



BM3D sharpening experiments

Noisy *Fundus* $\sigma = 20$



BM3D-SH3D



Part III: Variational BM3D Formulation. Motivation

- Block Matching and 3-D collaborative Filtering (BM3D) algorithm (Dabov, Foi, Katkovnik, and Egiazarian, IEEE TIP, 2007) is currently recognized as one of the best performing denoising algorithms.
- A family of algorithms of this type has been developed for various applications: video, demosaicking, deblurring, super-resolution, etc.
- All these algorithms are based on overcomplete windowed image modeling and special nonlocal nonparametric techniques.
- Recently, a special prior has been proposed allowing to reformulate the multi-stage hard-thresholding BM3D denoising as global minimization of a special energy criterion (V. Katkovnik and K. Egiazarian, "Nonlocal image deblurring: variational formulation with nonlocal collaborative l_0 -norm imaging," LNLA 2009, Tuusula, Finland, 2009).

- Better understanding BM3D as an universal improved image modeling technique.
- Novel recursive deblurring algorithms have been developed based on this nonlocal collaborative l_0 -norm prior.
- Simulation experiments demonstrate a very good performance of this novel deblurring algorithm.

Variational Formulation

- Suppose we have independent random observation pairs $\{z_i, x_i\}$,

$$z_i = y_i + \varepsilon_i,$$

where $y_i = y(x_i)$ is a signal of interest, $x_i \in \mathbb{R}^2$, and $\varepsilon_i = \varepsilon(x_i)$ is an additive noise, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

The denoising problem is to reconstruct $y(x_i)$ from $\{z_i\}$:

- Variational approach

$$\hat{y} = \arg \min_y \underbrace{\|y - z\|_2^2 / \sigma^2}_{\text{fidelity}} + \underbrace{\lambda \cdot \text{pen}(y)}_{\text{penalty}},$$

- Heuristic (or semi-heuristic) approach as an alternative to the variational one.
- Examples: local and nonlocal nonparametric regression methods, BM3D, etc.

Conventional penalties

- Quadratic penalties (Tikhonov A.N. and V.Y. Arsenin, 1977)

$$\text{pen}(y) = \int |y|^2 dx, \quad \text{pen}(y) = \int |Ly|^2 dx.$$

These penalties mean that the solution is penalized with respect to:

$$\min_y \int |y|^2 dx \rightarrow y^0 = 0, \quad \min_y \int \|\nabla y\|_2^2 dx \rightarrow \nabla y^0 = 0,$$

$$\min_y \int |Ly|^2 dx \rightarrow Ly^0 = 0;$$

- Total variation (Rudin, Osher, and Fatemi, 1992).

$$\text{pen}(y) = \int \|\nabla y\|_2 dx \rightarrow \nabla y^0 = 0,$$

where Δy is a vector-gradient of y . This penalty allows discontinuous solutions and preserve edges while filtering out high-frequency oscillations.

Conventional penalties (cont.)

Complexity penalty is formulated usually for spectrum representations of the image as $\theta = \mathcal{T}\{y\}$, where \mathcal{T} stands for orthonormal or overcomplete transforms. This penalty is calculated as

$$\text{pen}(\theta) = \|\theta\|_0,$$

where the l_0 -norm gives a number of active spectrum elements different from zero. It enables penalization with respect to $\|\theta\|_0 = 1$;

Nonlocal penalization with respect to a desirable image (Kindermann, Osher, and Jones, 2005)

$$\text{pen}(y) = \int g\left(\frac{|y^{TRUE}(x) - y(v)|^2}{h^2}\right) w(|x - v|) dx dv,$$

where $w > 0$ is a window, and g is a differentiable function, $g(0) = 0$. It gives $\min_y \text{pen}(y) \implies y^0 = y^{TRUE}$.

Block-wise imaging: windowing

- Let the signals be defined on the regular 2-D grid X .

Consider a windowing $C = \{X_r, r = 1, \dots, N_s\}$ of X with N_s blocks (uniform windows, patches) $X_r \subset X$ such that $\bigcup_{r=1}^{N_s} X_r = X$.

Each $x \in X$ belongs to at least one subset X_r .

Block-wise imaging: windowing

- Let the signals be defined on the regular 2-D grid X .

Consider a windowing $C = \{X_r, r = 1, \dots, N_s\}$ of X with N_s blocks (uniform windows, patches) $X_r \subset X$ such that $\bigcup_{r=1}^{N_s} X_r = X$.

Each $x \in X$ belongs to at least one subset X_r .

- The noise-free data Y and the noisy data Z windowed on X_r are arranged in blocks denoted as Y_r and Z_r , respectively.

The blocks are overlapping and therefore some of the elements may belong to more than one block.

Block-wise imaging: transform domain representations

- We use transforms (orthonormal series) of pixels in the blocks. The transform, denoted as \mathcal{T}_r^{2D} , is applied for each window X_r independently as

$$\theta_r = \mathcal{T}_r^{2D} (Y_r), \quad \left[= D_r Y_r D_r^T \right] \quad r = 1, \dots, N_s,$$

where θ_r is the spectrum of Y_r .

Block-wise imaging: transform domain representations

- We use transforms (orthonormal series) of pixels in the blocks. The transform, denoted as \mathcal{T}_r^{2D} , is applied for each window X_r independently as

$$\theta_r = \mathcal{T}_r^{2D} (Y_r), \quad \left[= D_r Y_r D_r^T \right] \quad r = 1, \dots, N_s,$$

where θ_r is the spectrum of Y_r .

- The inverse $\mathcal{T}_r^{2D^{-1}}$ of \mathcal{T}_r^{2D} defines the signal from the spectrum as

$$Y_r = \mathcal{T}_r^{2D^{-1}} (\theta_r), \quad \left[= D_r^T \theta_r D_r \right] \quad r = 1, \dots, N_s.$$

The noisy spectrum of the noisy signal is defined as

$$\tilde{\theta}_r = \mathcal{T}_r^{2D} (Z_r), \quad \left[= D_r Z_r D_r^T \right] \quad r = 1, \dots, N_s.$$

Group-wise collaborative penalty

It is assumed that there is a similarity between some of the blocks and the similar blocks are clustered in "groups".

Initially, the penalty for the r -th group can be defined as

$$\text{pen}_r(\{\theta_j\}_{j \in K_r}, \{\vartheta_{r,j}\}_{j \in K_r}) = \left(\sum_{j \in K_r} \|\theta_j - \vartheta_{r,j}\|_2^2 \right) + \lambda_r \|\{\vartheta_{r,j}\}_{j \in K_r}\|_0,$$

where

$$K_r = \{j : \|Y_r - Y_j\|_2^2 \leq h\}$$

Here, $\{\vartheta_{r,j}\}_{j \in K_r}$ is a set of the models for all j -th blocks included in the r -th group, and $\{\theta_j\}_{j \in K_r}$ are the corresponding noise-free block spectra in this group.

Group-wise collaborative penalty (cont.)

Let us treat the collection of 2-D block spectra $\{\theta_{r,j}\}_{j \in K_r}$ as 3-D array, where j is the index used for the third dimension.

Applying a 1-D orthonormal transform \mathcal{T}^{1D} with respect to j we arrive to a 3-D group spectrum

$$\Omega_r^Y = \mathcal{T}^{1D}(\{\theta_{r,j}\}_{j \in K_r}).$$

Replace the set K_r of the 2-D spectrum-approximations $\{\theta_{r,j}\}_{j \in K_r}$ with this joint 3-D spectrum Ω_r^Y .

The l_0 -norm $\|\{\theta_{r,j}\}_{j \in K_r}\|_0$ in the penalty is replaced with the norm in this 3-D spectrum space defined as

$$\|\Omega_r\|_{l_0} = \sum_{k,l} \mathbf{1}(\Omega_r(k,l) \neq 0).$$

This 3-D spectrum representation is used as a joint collaborative model of the signal clustered in the r -th group:

$$\text{pen}_r(\Omega_r^Y, \Omega_r) = \underbrace{\|\Omega_r^Y - \Omega_r\|_2^2}_{\text{Accuracy}} + \underbrace{\lambda_r \|\Omega_r\|_0}_{\text{Complexity}}.$$

Global penalty in spectrum domain

Let us go further and introduce the global penalty as the weighted mean of the group-wise penalties

$$\begin{aligned} PEN(\{\Omega_r^Y\}, \{\Omega_r\}) &= \sum_r g_r \cdot pen_r(\Omega_r^Y, \Omega_r) = \\ &\sum_r g_r \left(\|\Omega_r^Y - \Omega_r\|_2^2 + \lambda_r \|\Omega_r\|_0 \right), \end{aligned}$$

with the group weights g_r calculated as

$$g_r = \frac{1/\|\Omega_r\|_0}{\sum_r 1/\|\Omega_r\|_0},$$

where the spectrum Ω_r is an estimate for the spectrums Ω_r^Y in the r -th group, and $\|\Omega_r\|_0$ is the l_0 -norm penalty for this estimate.

In the global penalty the group-wise ones are weighed with the weights inversely proportional to the complexity of the group-wise models.

This rule perfectly corresponds to the idea of sparse image modeling with a low-complexity model as the main goal.

Global penalty in signal domain

In the signal/image domain the global penalty can be represented using the block-wise true signals Y_j and the signal approximations $Y_{r,j}$ in the following form

$$PEN(\{Y_j\}, \{\Omega_r\}) = \sum_r g_r \left(\sum_{j \in K_r} \|Y_j - Y_{r,j}\|_2^2 + \lambda_r \|\Omega_r\|_0 \right),$$

where

$$Y_{r,j} = \mathcal{T}^{2D-1}(\theta_{r,j}), \Theta_r = \{\theta_{r,j}\}_{j \in K_r} = \mathcal{T}^{1D-1}(\Omega_r).$$

Deblurring: variational formulation

- Suppose we have independent random observation pairs $\{z_i, x_i\}$ given in the form

$$z_i = (w * y)(x_i) + \sigma \varepsilon_i,$$

where $z_i = z(x_i)$ and $y_i = y(x_i)$ are noisy observations and signal of interest, respectively, $\varepsilon_i \sim \mathcal{N}(0, 1)$.

The deblurring problem is to reconstruct $y(x_i)$ from noisy observations $\{z_i\}$.

Deblurring: variational formulation

- Suppose we have independent random observation pairs $\{z_i, x_i\}$ given in the form

$$z_i = (w * y)(x_i) + \sigma \varepsilon_i,$$

where $z_i = z(x_i)$ and $y_i = y(x_i)$ are noisy observations and signal of interest, respectively, $\varepsilon_i \sim \mathcal{N}(0, 1)$.

The deblurring problem is to reconstruct $y(x_i)$ from noisy observations $\{z_i\}$.

- Some of the methods used:
 - (1) Decouple of deblurring and denoising, e.g. BM3D deblurring.
 - (2) Variational approach

$$J = \underbrace{\| (w * y) - z \|_2^2 / \sigma^2}_{\text{Fidelity}} + \underbrace{\lambda \cdot \text{pen}(y)}_{\text{Penalty}}.$$

Deblurring by global energy minimization

Let \mathbf{Y} , \mathbf{Z} , \mathbf{Y}_j , $\hat{\mathbf{Y}}_{r,j}$ be the lexicographical vector representations of the corresponding images/blocks Y , Z , Y_j , $\hat{Y}_{r,j}$.

The vectors \mathbf{Y}_j are projections of the vector \mathbf{Y} , which can be defined through the projection matrices P_j ,

$$\mathbf{Y}_j = P_j \mathbf{Y}.$$

Here P_j are binary matrices with items (0,1).

Then the deblurring problem can be formulated as the variational problem:

$$\begin{aligned}\hat{\mathbf{Y}} &= \arg \min_{\mathbf{Y}, \{\Omega_r\}} J, \\ J &= \|\mathbf{Z} - A\mathbf{Y}\|_2^2 / \sigma^2 + \lambda \cdot PEN(\{\mathbf{Y}_j\}, \{\Omega_r\}).\end{aligned}$$

For solution we exploit a recursive alternative minimization of J on $\{\Omega_r\}$ and \mathbf{Y} .

Minimization in spectrum domain

If \mathbf{Y} and g_r are given the minimization on $\{\Omega_r\}$ concerns the penalty term $PEN(\{\mathbf{Y}_j\}, \{\Omega_r\})$ only.

With a fixed g_r the minimization is reduced to scalar calculations independent for each element of Ω_r :

$$\hat{\Omega}_r(k, l) = \arg \min_{x \in \mathbb{R}} \left(\Omega_r^Y(k, l) - x \right)^2 + \lambda_r \cdot \mathbf{1}(x \neq 0) .$$

This solution is the hard-thresholding of $\Omega_r^Y(k, l)$ calculated as

$$\hat{\Omega}_r(k, l) = \Omega_r^Y(k, l) \cdot \mathbf{1} \left(|\Omega_r^Y(k, l)| \geq \sqrt{\lambda_r} \right) .$$

When $\hat{\Omega}_r(k, l)$ are found the signal estimates are calculated as

$$\{\hat{\theta}_{r,j}\}_{j \in K_r} = \mathcal{T}^{1D^{-1}}(\hat{\Omega}_r), \hat{Y}_{r,j} = \mathcal{T}^{2D^{-1}}(\hat{\theta}_{r,j}) .$$

Minimization in signal domain

Consider minimization of J on \mathbf{Y} provided $\{\Omega_r\}_r$ are given as $\{\hat{\Omega}_r\}_r$.

The spectrums Ω_r^Y depend on \mathbf{Y} and this dependence should be taken into considerations:

$$J = \|\mathbf{Z} - A\mathbf{Y}\|_2^2 / \sigma^2 + \mu \cdot \sum_r g_r \cdot \left(\sum_{j \in K_r} \|P_j \mathbf{Y} - \hat{\mathbf{Y}}_{r,j}\|_2^2 + \lambda_r \|\Omega_r\|_0 \right).$$

Differentiation on \mathbf{Y} gives after some manipulations the estimate of \mathbf{Y} :

$$\hat{\mathbf{Y}} = \Phi^{-1} \cdot \left(A^T \mathbf{Z} / \sigma^2 + \mu \cdot \sum_r g_r \sum_{j \in K_r} P_j^T \hat{\mathbf{Y}}_{r,j} \right),$$

$$\Phi = A^T A / \sigma^2 + \mu \cdot \sum_r g_r \sum_{j \in K_r} P_j^T P_j.$$

Note, that the matrices $P_j^T P_j$ and $W = \sum_r g_r \sum_{j \in K_r} P_j^T P_j$ are diagonal.

Matrix DEB-NEM algorithm

Recursive calculations based on the above matrix formulas result in the following algorithm:

1: Initialization: $\hat{\mathbf{Y}}^{(0)}$ and $g_r^{(0)} = 1$;

2: For every $t = 0, 1, \dots$

- Calculate the windowed signals $\hat{Y}_r^{(t)}$, the groupings

$$K_r^{(t)} = \{j : 1(\|\hat{Y}_r^{(t)} - \hat{Y}_j^{(t)}\|_2^2 \leq h)\}$$

and the spectra $\tilde{\theta}_{r,j}^{(t)} = \mathcal{T}_r^{2D}(\hat{Y}_j^{(t)})$, $j \in K_r^h$, for all groups r ;

- Calculate the group-wise "noisy" spectrums $\tilde{\Omega}_r^{\hat{\mathbf{Y}}^{(t)}}$, the 3D spectrum thresholded estimates $\hat{\Omega}_r^{(t)}$ and the windowed signal estimates $\hat{Y}_{r,j}^{(t)}$;
- Calculate the complexity $\|\hat{\Omega}_r^{(t)}\|_0$ of the group models and the weights

$$g_r^{(t)} = \frac{1/\|\hat{\Omega}_r^{(t)}\|_0}{\sum_r 1/\|\hat{\Omega}_r^{(t)}\|_0};$$

Matrix DEB-NEM algorithm (cont.)

Update the signal estimate $\hat{\mathbf{Y}}^{(t+1)}$ using

$$\hat{\mathbf{Y}}^{(t+1)} = \Phi^{-1} \cdot \left(A^T \mathbf{z} / \sigma^2 + \mu \cdot \sum_r \mathbf{g}_r^{(t)} \sum_{j \in K_r^{(t)}} P_j^T \hat{\mathbf{Y}}_{r,j}^{(t)} \right),$$
$$\Phi = A^T A / \sigma^2 + \mu \cdot \sum_r \mathbf{g}_r^{(t)} \sum_{j \in K_r^{(t)}} P_j^T P_j;$$

Continue until convergence.

"Ideal" collaborative penalty

Minimization on $\{\Omega_r\}$ yields

$$\min_{\{\Omega_r\}} PEN(\{\Omega_r^Y\}, \{\Omega_r\}) =$$
$$\min_{\{\Omega_r\}_r} \sum_r g_r \left(\sum_{j \in K_r^h} \|\Omega_r^Y - \Omega_r\|_2^2 + \lambda_r \|\Omega_r\|_0 \right) \Rightarrow \Omega_r^0;$$

Minimization on Y yields

$$\min_Y PEN(\{Y_j\}, \{\Omega_r^0\}) \Rightarrow \mathbf{Y}^0,$$
$$\mathbf{Y}^0 = \left(\sum_r g_r \sum_{j \in K_r^h} \mathbf{P}_j^T \mathbf{P}_j \right)^{-1} \left(\sum_r g_r \sum_{j \in K_r^h} \mathbf{P}_j^T \mathbf{Y}_{r,j} \right),$$
$$Y_{r,j} = \mathcal{T}^{2D-1}(\theta_{r,j}), \quad \{\theta_{r,j}\}_{j \in K_r^h} = \mathcal{T}^{1D-1}(\Omega_r^0).$$



Smoothed Y^0 images obtained for different λ_r .

Links between NEM and BM3D for denoising

- The basic hard-thresholding thresholding BM3D algorithm can be interpreted as an alternative minimizer of the global penalty.
- Let Ω_r^Y in the global penalty be replaced by the noisy Ω_r^Z , and this global penalty be minimized on $\{\Omega_r\}$:

$$\begin{aligned} \min_{\{\Omega_r\}} \text{PEN}(\{\Omega_r^Z\}, \{\Omega_r\}) = \\ \min_{\{\Omega_r\}_r} \sum_r g_r \left(\sum_{j \in K_r^h} \|\Omega_r^Z - \Omega_r\|_2^2 + \lambda_r \|\Omega_r\|_0 \right) \implies \hat{\Omega}_r; \end{aligned}$$

Links between BM3D penalty and BM3D algorithm for denoising (cont)

Minimization on \mathbf{Y} gives the BM3D aggregation.

The vectorized representation of the signals gives the estimate of Y in the form

$$\hat{\mathbf{Y}} = \Phi^{-1} \sum_r g_r \sum_{j \in K_r^h} \mathbf{P}_j^T \hat{\mathbf{Y}}_{r,j},$$
$$\Phi = \sum_r g_r \sum_{j \in K_r^h} \mathbf{P}_j^T \mathbf{P}_j, \quad g_r = \frac{1}{\sigma^2 \|\Omega_r\|_0},$$

which is identical to used in BM3D for aggregation of the estimates obtained by the hard-thresholding.

Frequency-domain deblurring algorithm

First, we rewrite the equation for $\hat{\mathbf{Y}}$ as a set of the linear equations

$$\left(A^T A / \sigma^2 + \mu \cdot W \right) \hat{\mathbf{Y}} = A^T \mathbf{Z} / \sigma^2 + \mu \cdot \sum_r g_r \sum_{j \in K_r} P_j^T \hat{\mathbf{Y}}_{r,j},$$

and solve these equations with respect to \mathbf{Y} using the recursive procedure

$$\hat{\mathbf{Y}}_{(k+1)} = \hat{\mathbf{Y}}_{(k)} - \alpha_k \left[\left(A^T A / \sigma^2 + \mu \cdot W \right) \hat{\mathbf{Y}}_{(k)} - A^T \mathbf{Z} / \sigma^2 - \mu \cdot \tilde{\mathbf{Y}} \right],$$

where

$$\tilde{\mathbf{Y}} = \sum_r g_r \sum_{j \in K_r} P_j^T \hat{\mathbf{Y}}_{r,j}, \quad k = 1, \dots, L.$$

Frequency domain algorithm (cont.)

With above assumptions the vectorization of the convolution is valid

$$\mathbf{u} = \mathbf{A}\mathbf{y}, \mathbf{A} = \mathbf{w} \otimes \mathbf{w},$$

where w is a blur PSF and \otimes stands for the Kronecker product.

Assuming that the blur is shift invariant circular, \mathbf{A} is a structured Toeplitz matrix and the discrete Fourier transform (DFT) can be used for the matrix calculations:

$$\mathbf{A}^T \cdot \mathbf{z} = \text{col}(\mathcal{F}^{-1}\{\mathcal{F}\{w\}^* \cdot \mathcal{F}\{Z\}\}),$$

$$\mathbf{A}^T \mathbf{A} \cdot \hat{\mathbf{Y}}_{(k)} = \text{col}(\mathcal{F}^{-1}\{|\mathcal{F}\{w\}|^2 \cdot \mathcal{F}\{Y_{(k)}\}\}).$$

Frequency domain algorithm (cont.)

Then, the recursive algorithm can be implemented without variable vectorization for variables organized as image-size matrices

$$Y_{(k+1)} = Y_{(k)} - \alpha_k [\mathcal{F}^{-1}\{|\mathcal{F}\{w\}|^2 \cdot \mathcal{F}\{Y_{(k)}\}\} / \sigma^2 + \mu \cdot (\tilde{W} \circ Y_{(k)}) - \mathcal{F}^{-1}\{\mathcal{F}\{w\}^* \cdot \mathcal{F}\{Z\}\} / \sigma^2 - \mu \cdot \tilde{Y}],$$
$$k = 1, \dots, L,$$

where

$$\tilde{W} = \mathit{reshape}_{n \times m}[\mathit{diag}\{W\}], \quad \tilde{Y} = \mathit{reshape}_{n \times m}[\tilde{\mathbf{Y}}],$$

and $\tilde{W} \circ Y_{(k)}$ means the element-wise product of two matrices.

The DEB-NEM algorithm is organized as it is presented in the previous section with

$$Y^{(t+1)} = \mathit{reshape}_{n \times m}[\hat{\mathbf{Y}}^{(t+1)}]$$

calculated according to the above recursive procedure.

Matrix DEB-NEM algorithm: experiments

In our experiments, we use 9×9 uniform kernel (boxcar) blur PSF.

The noise is white zero-mean Gaussian with blurred-signal-to-noise-ratio $BSNR = 40$ dB.

The parameters of the algorithm are fixed as $\lambda/\sigma = 10$ and $\mu\sigma = 1$.

The main goal of these experiments is to check a potential of the proposed penalty function in the deblurring problem.

Table: Initial (DEBBM3D) $PSNR$ and $ISNR$ values given with the index 0 and final (DEB-NEM) after 10 iterations (in dB).

	$PSNR_0$ [$ISNR_0$]	$PSNR_{10}$ [$ISNR_{10}$]
cameraman, 64^2	23.02 [7.18]	26.65 [10.81]
lena, 64^2	30.81 [6.41]	34.35 [9.95]
barbara, 64^2	25.27 [7.93]	26.27 [8.93]

Visual improvements for cameraman

Cameraman fragment: true and blurred noisy images (left).

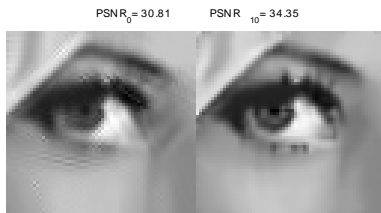
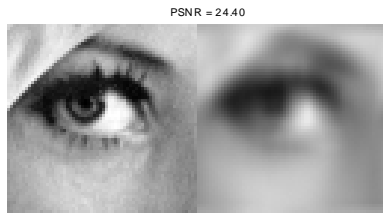
Cameraman fragment: initialization and 10-th iteration of DEB-NEM reconstruction (right)



Visual improvements for Lena

Lena fragment: true and blurred noisy images (left).

Lena fragment: initialization and 10-th iteration of DEB-NEM reconstruction (right)



Frequency domain DEB-NEM: experiments

Fixed parameters: $\mu/\sigma^2 = 2$, $\lambda = 6\sigma \cdot 2.7$, $N_1 = 8$, $N_2 = 32$, $N_{step} = 2$
(for "Barbara" $\mu/\sigma^2 = 8$, $\lambda = \sigma \cdot 2.7$).

Table: *ISNR* values for initial BM3D-DEB given with the index 0, for DEB-NEM after 20 iterations (in dB), and best results by other methods.

	<i>ISNR</i> ₀ (<i>DEBBM3D</i>)	<i>ISNR</i> ₂₀	Other
Cameraman, 256 ²	8.4	9.92	9.1 (Portilla)
Lena, 512 ²	7.8	8.91	8.52 (Chantas)
Barbara, 512 ²	5.9	6.05	6.29 (Babacan)
House, 256 ²	10.9	12.8	10.74 (Portilla)
Boats, 512 ²	8.5	9.55	
Checkerboard, 256 ²	23.3	48.3	
Phantom, 256 ²	12.3	21.4	17.86 (Oliveira)



Cameramen test image: true (a), blurred noisy (b), DEB-NEM reconstruction after 20 iterations (c), DEBBM3D reconstruction (d).



A fragment of Lena test image: true (a), blurred noisy (b), DEB-NEM reconstruction after 20 iterations (c), DEBBM3D reconstruction (d).

Currently we develop improved versions of the presented algorithm based on *variable splitting proximal operator* and *augmented Lagrangian* techniques.

The principal intention and result of our research is demonstrating the power of the image modeling based on block matching and collaborative 3-D filtering.

A. Danielyan, V. Katkovnik and K. Egiazarian, "Image deblurring by augmented Lagrangian with BM3D frame prior", Proc. 3rd Workshop on Information Theoretic Methods in Science and Engineering , WITMSE 2010, Tampere 2010.

THANK YOU

References

<http://www.cs.tut.fi/~foi>

<http://www.cs.tut.fi/~lasip>

<http://sp.cs.tut.fi/groups/trans/>

- Danielyan, A., V. Katkovnik and K. Egiazarian, "Image deblurring by augmented Lagrangian with BM3D frame prior", Proc. 3rd Workshop on Information Theoretic Methods in Science and Engineering , WITMSE 2010, Tampere 2010.
- Katkovnik, V., and K. Egiazarian, "Nonlocal image deblurring: variational formulation with nonlocal collaborative l0-norm imaging," LNLA 2009, Tuusula, Finland, 2009
- Mäkitalo, M., and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising", IEEE Trans. Image Process., 2010. doi:10.1109/TIP.2010.2056693
- Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising", Int. J. Computer Vision, vol. 86, no. 1, pp. 1-32, January 2010. doi:10.1007/s11263-009-0272-7
- Foi, A., "Clipped noisy images: heteroskedastic modeling and practical denoising", Signal Processing, vol. 89, no. 12, pp. 2609-2629, December 2009. doi:10.1016/j.sigpro.2009.04.035
- Danielyan, A., A. Foi, V. Katkovnik, and K. Egiazarian, "Spatially adaptive filtering as regularization in inverse imaging: compressive sensing, upsampling, and super-resolution", in Super-Resolution Imaging (P. Milanfar, ed.) CRC Press / Taylor & Francis, 2010. ISBN: 978-1-4398-1930-2.
- Foi, A., "Optimization of variance-stabilizing transformations", preprint, submitted. (online at <http://www.cs.tut.fi/~foi/optvst>)
- Danielyan, A., and A. Foi, "Noise variance estimation in nonlocal transform domain", Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2009, Tuusula, Finland, pp. 41-45, August 2009. doi:10.1109/LNLA.2009.5278404
- Danielyan, A., M. Vehviläinen, A. Foi, V. Katkovnik, and K. Egiazarian, "Cross-color BM3D filtering of noisy raw data", Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2009, Tuusula, Finland, pp. 125-129, August 2009. doi:10.1109/LNLA.2009.5278395
- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D Image Denoising with Shape-Adaptive Principal Component Analysis", Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09), Saint-Malo, France, April 2009.

- Danielyan, A., A. Foi, V. Katkovnik, and K. Egiazarian, "Image and video super-resolution via spatially adaptive block-matching filtering", Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2008, Lausanne, Switzerland, August 2008.
- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "A nonlocal and shape-adaptive transform-domain collaborative filtering", Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2008, Lausanne, Switzerland, August 2008.
- Katkovnik, V., A. Foi, K. Dabov, and K. Egiazarian, "Spatially adaptive support as a leading model-selection tool for image filtering", Proc. First Workshop Inf. Th. Methods Sci. Eng., WITMSE, Tampere, August 2008.
- Foi, A., M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single image raw-data", IEEE Trans. Image Process., vol. 17, no. 10, pp. 1737-1754, October 2008.
- Paliy, D., A. Foi, R. Bilcu, V. Katkovnik, "Denoising and interpolation of noisy Bayer data with adaptive cross-color filters", SPIE-IS&T Electronic Imaging, Visual Communications and Image Processing 2008, vol. 6822, no. 6822-1K, San Jose (CA), USA, January 2008.
- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," Proc. SPIE Electronic Imaging '08, vol. 6812, no. 6812-1D, San Jose (CA), USA, January 2008.
- Foi, A., "Spatially adaptive local approximations in signal and image processing: varying-scale polynomials, anisotropic adaptation, shape-adaptive transforms", Lecture notes of the tutorial given at the 15th European Signal Process. Conf., EUSIPCO 2007, Poznan, September 2007.
- Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images", IEEE Trans. Image Process., vol. 16, no. 5, pp. 1395-1411, May 2007. doi:10.1109/TIP.2007.891788
- Egiazarian, K., A. Foi, and V. Katkovnik, "Compressed Sensing Image Reconstruction via Recursive Spatially Adaptive Filtering", Proc. IEEE Int. Conf. Image Process., ICIP 2007, San Antonio (TX), USA, pp. 549-552, September 2007.

- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Joint image sharpening and denoising by 3D transform-domain collaborative filtering," Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2007, Moscow, Russia, September 2007.
- Dabov, K., A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," Proc. 15th European Signal Processing Conference, EUSIPCO 2007, Poznan, Poland, September 2007.
- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," Proc. IEEE Int. Conf. Image Process., ICIP 2007, San Antonio, TX, USA, September 2007.
- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080-2095, August 2007. doi:10.1109/TIP.2007.901238
- Foi, A., Dabov, K., V. Katkovnik, and K. Egiazarian, "Shape-Adaptive DCT for denoising and image reconstruction", Proc. SPIE Electronic Imaging 2006, Image Processing: Algorithms and Systems V, 6064A-18, San Jose, CA, January 2006.
- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," Proc. SPIE Electronic Imaging '06, no. 6064A-30, San Jose, California, USA, January 2006.
- Katkovnik, V., K. Egiazarian, and J. Astola, Local Approximation Techniques in Signal and Image Processing, SPIE Press, Monograph Vol. PM157, September 2006.
- Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Directional varying scale approximations for anisotropic signal processing", Proc. of XII European Signal Process. Conf., EUSIPCO 2004, pp. 101-104, 2004.