TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Signal Processing – Transforms and Spectral Methods Group

Removal of signal-dependent noise: the BM3D filter and optimized variance-stabilizing transformations

Alessandro Foi

www.cs.tut.fi/~foi

INRIA Centre de Rennes Bretagne Atlantique, IRISA *April 10, 2009*

Outline

1. Block-Matching and 3D filtering (BM3D) algorithm

Grouping and collaborative filtering, block-based algorithm and shape-adaptive PCA implementation.

2.1 Variance stabilization

Introduction to the problem, examples, counterexamples, main results.

2.2 Optimization of variance stabilizing transformations

Stabilization functional; Optimization by recursive approximate integral stabilization; comparison with AVAS; Optimization by direct search; relaxation of monotonicity, examples.

3. Application to raw-data denoising

Noise modelling for raw-data of imaging sensors; clipping; doubly censored normal distributions; variance stabilization; filtering; debiasing, declipping. Comparison of standard vs. optimized stabilizers.

Block-Matching and 3D filtering (BM3D) denoising algorithm

Generalizes NL-means and overcomplete transform methods.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering", *Proc. SPIE El. Imaging 2006, Image Process.: Algorithms and Systems V*, no. 6064A-30, San Jose (CA), USA, Jan. 2006.

— , "Image denoising by sparse 3D transform-domain collaborative filtering", *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080-2095, Aug. 2007.

Observation model for the image denoising problem

$$z(x) = y(x) + \eta(x), \qquad x \in X \subset \mathbb{Z}^2,$$

$z : X \to \mathbb{R}$	observed noisy image
$y : X \to \mathbb{R}$	unknown original image (grayscale)
$\eta : X \to \mathbb{R}$	i.i.d. Gaussian white noise, $\eta(\cdot) \sim \mathcal{N}\left(0, \sigma^2\right)$

Notation

Given a function $f : X \to \mathbb{R}$, a subset $U \subset X$, and a function $g : U \to \mathbb{R}$, we denote by:

$f_{\|U} : U \to \mathbb{R}$	the *restriction* of f on U, $f_{\|U}(x) = f(x)\ \forall x \in U$;
$g^{\|X} : X \to \mathbb{R}$	the *zero-extension* of g to X, $\left(g^{\|X}\right)_{\|U} = g$ and $g^{\|X}(x) = 0\ \forall x \in X \setminus U$;
$\chi_U = 1_{\|U}{}^{\|X}$	the *characteristic function (indicator)* of U;
$\|U\|$	the *cardinality* of U (i.e. the number of its elements of U);
$\circledast$	the *convolution* operation.

Block-matching

Let $x \in X$ and denote by $\tilde{B}_x \subset \mathbb{Z}^2$ be the square block of size $l \times l$ "centered" at x. Let $\mathbb{B}$ be the collection of all such blocks which are entirely contained in X, $\mathbb{B} = \left\{ \tilde{B}_x : x \in X,\ \tilde{B}_x \subset X \right\}$. Equivalently, define $X_{\mathbb{B}} = \left\{ x \in X : \tilde{B}_x \in \mathbb{B} \right\} = \left\{ x \in X : \tilde{B}_x \subset X \right\} \subset X$.

For each block $\tilde{B}_x \in \mathbb{B}$, (i.e. for each point $x \in X_{\mathbb{B}}$), we look for "similar" blocks $\tilde{B}_{x'}$ whose range distance $d_z(x, x')$ with respect to $\tilde{B}_x$,

$$d_z(x, x') = \left\| z_{|\tilde{B}_x} - z_{|\tilde{B}_{x'}} \right\|_2 ,$$

is smaller than a fixed threshold $\tau_{\text{match}} \geq 0$.

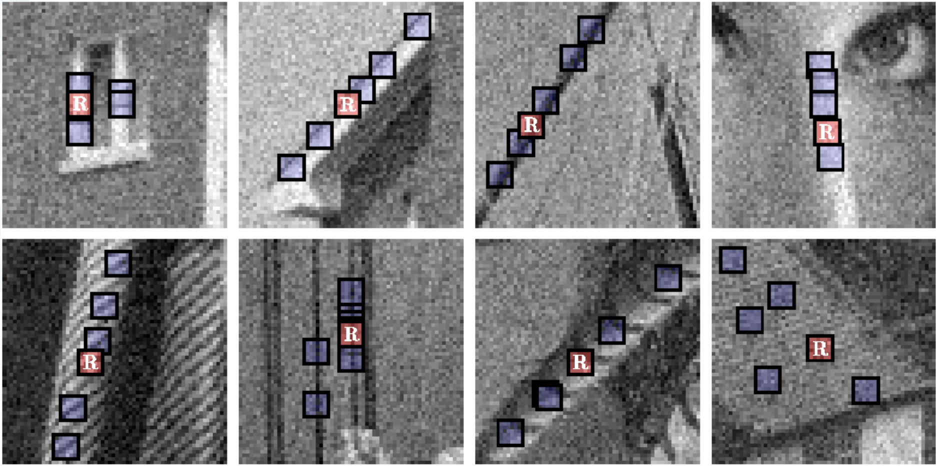
Thus, we construct the set S_x that contains the central points of the found blocks:

$$S_x = \left\{ x' \in X_{\mathbb{B}} : d_z(x, x') \leq \tau_{\text{match}} \right\}.$$

The threshold τ_{match} is the maximum d_z-distance for which two blocks are considered similar.

In case of heavy noise, we embed a coarse prefiltering within d_z (e.g., ℓ^2-distance of thresholded spectra). Otherwise, we need to increase l.

Block-matching



To a fixed "reference" block $\tilde{B}_{x_R} \in \mathbb{B}$ associate a collection (disjoint union) $\widetilde{\mathbb{B}}_{x_R}$ of neighborhoods:

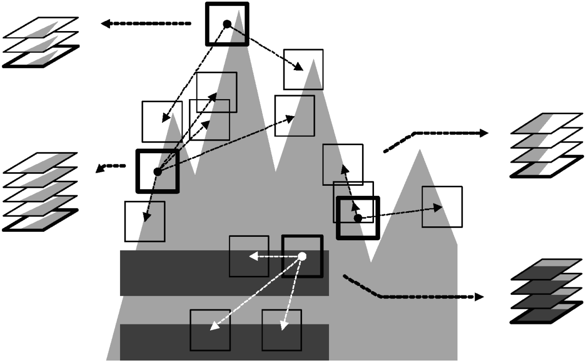
$$\begin{aligned}\widetilde{\mathbb{B}}_{x_R} &= \coprod_{x \in S_{x_R}} \tilde{B}_x = \\ &= \left\{ \left(\tilde{B}_x, x \right) : x \in S_{x_R} \right\} \subset X \times S_{x_R} \subset X \times X.\end{aligned}$$

Group

$$\text{collection of the noisy patches} \quad z_{|\tilde{B}_x},\ \tilde{B}_x \in \widetilde{\mathbb{B}}_{x_R}$$

$$\text{(Compact notation)} \quad \mathbf{Z}_{x_R} : \widetilde{\mathbb{B}}_{x_R} \to \mathbb{R}.$$

The patches can be stacked together into a 3-D data array defined on the square prism $B \times \{1, \ldots, |S_{x_R}|\}$.



Why groups are good and why do we need to be careful

Groups are characterized by both:

- ⋄ *intra*-block correlation between the pixels of each grouped block (natural images);
- ∘ *inter*-block correlation between the corresponding pixels of different blocks (grouped block are similar);

Warnings:

- ⋄ blocks are not necessary flat or smooth but can be anything;
- ∘ “similar” does not mean “identical”.

Goals:

- ⋄ exploit intra-block correlation whenever possible, without smoothing away the unexpected;
- ∘ exploit similarity in the forms in which it exists, without forcing dissimilar blocks to become identical.

Collaborative filtering

- *each grouped block collaborates for the filtering of all others, and vice versa.*
- *provides individual estimates for all grouped blocks (not necessarily equal).*

Realized as shrinkage in a 3-D transform domain.

Typically separable transform: $T^{\text{3D}} = T^{\text{2D}} \circ T^{\text{1D}}$.

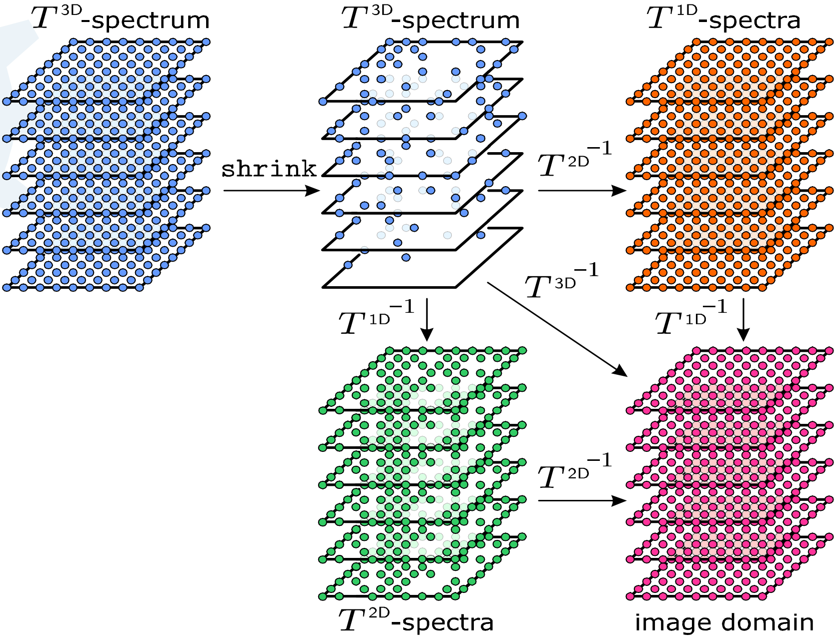
E.g.: 2D-DCT $\circ$ DCT $=$ 3D-DCT
or, restricting h and $|S_{x_R}|$ to powers of two,
biorth. 2D-DWT $\circ$ Haar 1D-DWT
shrinkage: hard-thresholding

$$\boxed{\widehat{\mathbf{Y}}_{x_R} = T^{\text{3D}\,-1}\left(\texttt{shrink}\left(T^{\text{3D}}\left(\mathbf{Z}_{x_R}\right)\right)\right)}$$

The group estimate $\widehat{\mathbf{Y}}_{x_R} : \widetilde{\mathbb{B}}_{x_R} \to \mathbb{R}$ is composed of slices with local block estimates $\hat{y}_{x,x_R} : \tilde{B}_x \to \mathbb{R}$ for each $\tilde{B}_x \in \widetilde{\mathbb{B}}_{x_R}$.

Total variance of $\widehat{\mathbf{Y}}_{x_R}$ can be estimated as $\operatorname{tsvar}\left\{\widehat{\mathbf{Y}}_{x_R}\right\} \approx \sigma^2 N_{x_R}^{\text{har}}$,
$N_{x_R}^{\text{har}}$ is number of coefficients of $T^{\text{3D}}(\mathbf{Z}_{x_R})$ that survive thresholding
(so-called "number of harmonics").

Collaborative filtering



Aggregation

For each reference point $x_R \in X$, grouping and collaborative filtering generate a group $\widehat{\mathbf{Y}}_{x_R}$ of $|S_{x_R}|$ distinct *local* estimates of y.

Overall, we have a highly redundant and rich representation of the original image y composed of the estimates

$$\coprod_{x_R \in X,\ x \in S_{x_R}} \hat{y}_{x,x_R}, \quad \text{where} \quad \hat{y}_{x,x_R} : \tilde{B}_x \to \mathbb{R}.$$

Note: different groups $\mathbf{Z}_{x_R}$ and $\mathbf{Z}_{x'_R}$ can lead to different estimates $\hat{y}_{x,x_R}$ and $\hat{y}_{x,x'_R}$ even when these estimates are defined on the same block $\tilde{B}_x$!

In order to obtain a single *global* estimate $\hat{y}^{\mathrm{ht}} : X \to \mathbb{R}$ defined on the whole image domain, all these local estimates are averaged together using adaptive weights $w_{x_R} > 0$ in the following convex combination:

$$\boxed{\hat{y}^{\mathrm{ht}} = \frac{\sum_{x_R \in X} \sum_{x \in S_{x_R}} w_{x_R} \hat{y}_{x,x_R}{}^{|X}}{\sum_{x_R \in X} \sum_{x \in S_{x_R}} w_{x_R} \chi_{\tilde{B}_x}} \qquad w_{x_R} = \frac{1}{\sigma^2 N_{x_R}^{\mathrm{har}}}.}$$

Wiener filtering stage

Denoising can be improved by performing matching within this estimate and replacing hard-thresholding by empirical Wiener filtering in the collaborative shrinkage.

Block-Matching

Noise in $\hat{y}^{\mathrm{ht}}$ is significantly attenuated: more accurate matching by replacing the distance d_z by a distance $d_{\hat{y}^{\mathrm{ht}}}$:

$$d_{\hat{y}^{\mathrm{ht}}}\left(x_R, x\right) = \left\| \hat{y}^{\mathrm{ht}}{}_{|\tilde{B}_{x_R}} - \hat{y}^{\mathrm{ht}}{}_{|\tilde{B}_x} \right\|_2 ,$$

The sets S_{x_R} are redefined as

$$S_{x_R} = \left\{ x \in X_{\mathbb{B}} : d_{\hat{y}^{\mathrm{ht}}}\left(x_R, x\right) \leq \tau_{\mathrm{match}} \right\} .$$

These new sets S_{x_R} lead to new collections (disjoint unions) of blocks $\widetilde{\mathbb{B}}_{x_R} = \coprod_{x \in S_{x_R}} \tilde{B}_x$.

Grouping: two groups

$$\mathbf{Z}_{x_R} : \widetilde{\mathbb{B}}_{x_R} \to \mathbb{R}, \quad \text{built by stacking together the noisy patches} \quad z_{|\tilde{B}_x}, \tilde{B}_x \in \widetilde{\mathbb{B}}_{x_R}$$

$$\widehat{\mathbf{Y}}^{\mathrm{ht}}_{x_R} : \widetilde{\mathbb{B}}_{x_R} \to \mathbb{R}, \quad \text{built by stacking together the estimate patches} \quad \widehat{y}^{\mathrm{ht}}_{|\tilde{B}_x}, \tilde{B}_x \in \widetilde{\mathbb{B}}_{x_R}$$

Collaborative Wiener filtering

Group Wiener estimate $$\boxed{\widehat{\mathbf{Y}}_{x_R} = T^{\mathsf{3D}\,-1}\left(\mathbf{W}_{x_R} T^{\mathsf{3D}}\left(\mathbf{Z}_{x_R}\right)\right)}$$

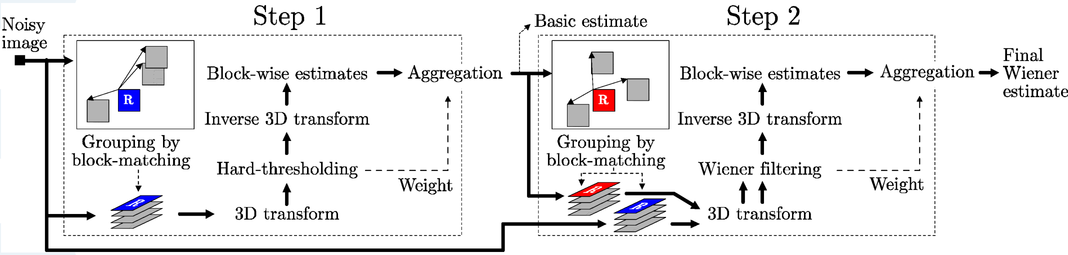
Wiener attenuation factors $$\boxed{\mathbf{W}_{x_R} = \frac{\left(T^{\mathsf{3D}}\left(\widehat{\mathbf{Y}}^{\mathrm{ht}}_{x_R}\right)\right)^2}{\left(T^{\mathsf{3D}}\left(\widehat{\mathbf{Y}}^{\mathrm{ht}}_{x_R}\right)\right)^2+\sigma^2}}$$

Estimate of total variance $\quad \operatorname{tsvar}\left\{\widehat{\mathbf{Y}}_{x_R}\right\} \approx \sigma^2 \left\|\mathbf{W}_{x_R}\right\|_2^2.$

Aggregation

Global estimate $$\boxed{\hat{y}^{\mathrm{wie}} = \frac{\sum_{x_R\in X}\sum_{x\in S_{x_R}} w_{x_R}\hat{y}_{x,x_R}{}^{|X}}{\sum_{x_R\in X}\sum_{x\in S_{x_R}} w_{x_R}\chi_{\tilde{B}_x}}, \qquad w_{x_R} = \frac{1}{\sigma^2\left\|\mathbf{W}_{x_R}\right\|_2^2}.}$$

BM3D flowchart



- ▷ Process overlapping blocks in a raster scan. For each such block, do the following:
  - (a) Use block-matching to find the locations of the blocks that are similar to the currently processed one. Form a 3D array (group) by stacking the blocks located at the obtained locations.
  - (b) Apply a 3-D transform on the formed group.
  - (c) Attenuate the noise by shrinkage the 3-D transform spectrum.
  - (d) invert the 3-D transform to produce filtered grouped blocks.
- ▷ Return the filtered blocks to their original locations in the image domain and compute the resultant filtered image by a weighted average of these filtered blocks (aggregation).

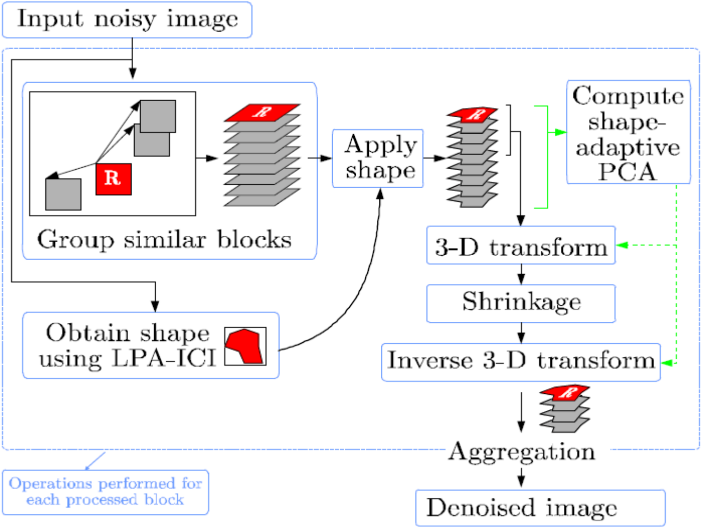
BM3D with Shape-Adaptive PCA (BM3D-SAPCA)

Main ingredients:

- **Local Polynomial Approximation - Intersection of Confidence Intervals (LPA-ICI)** to adaptively select support for 2-D transform;
- **Block-Matching** to enable non-locality;
- **Shape-Adaptive PCA (SA-PCA);**
- **Shape-Adaptive DCT** low-complexity 2-D transform on arbitrarily-shaped domains (when SA-PCA is not feasible).

K. Dabov, 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.

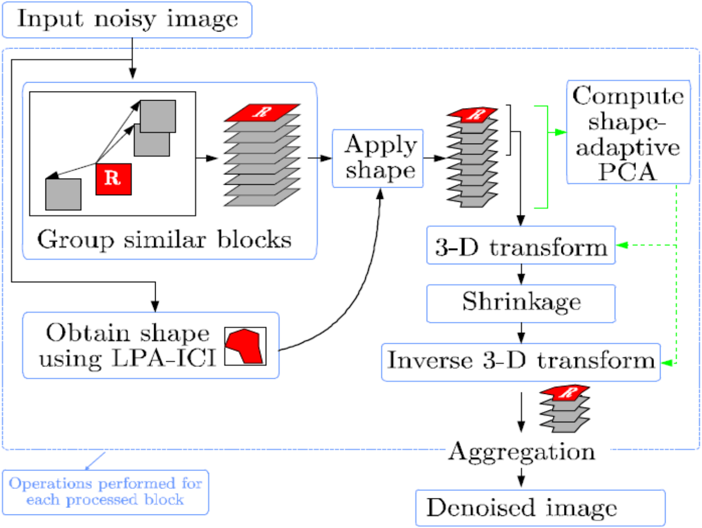
BM3D-SAPCA



At each pixel:

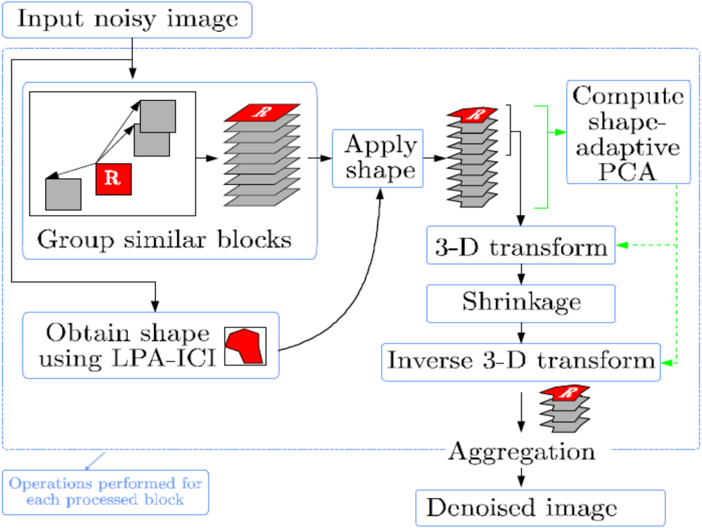
1. Group together square image blocks that are similar to the block centered at the current pixel.

BM3D-SAPCA



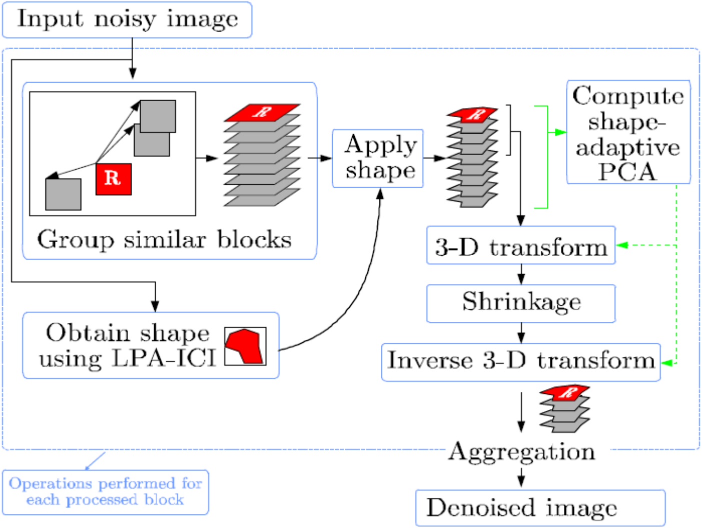
2. Obtain the anisotropic neighborhood at the current pixel using 8-directional LPA-ICI. Apply its shape on each of the grouped blocks, producing a group of adaptive-shape neighborhoods.

BM3D-SAPCA



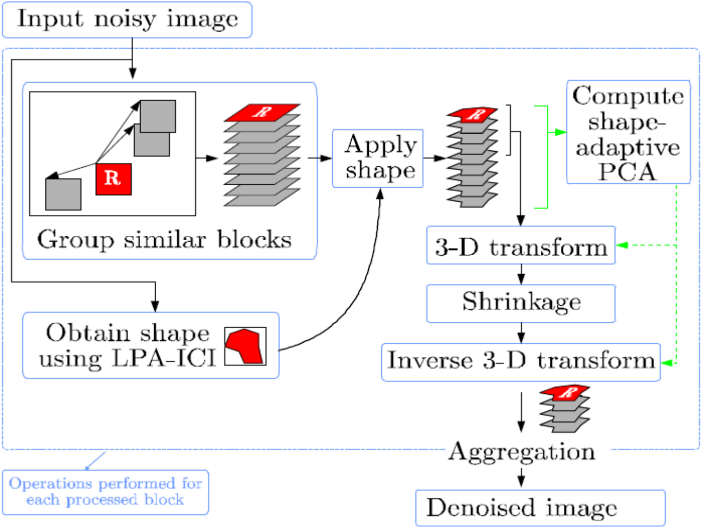
3. Use this group as training data for computing Shape-Adaptive PCA (SVD of the empirical second-moment matrix estimated from the group of similar adaptive-shape neighborhoods).

BM3D-SAPCA



3b. Keep only the eigenvectors (PC) whose corresponding eigenvalues are greater than a threshold proportional to the noise variance (*trimmed PCA*).
The 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.

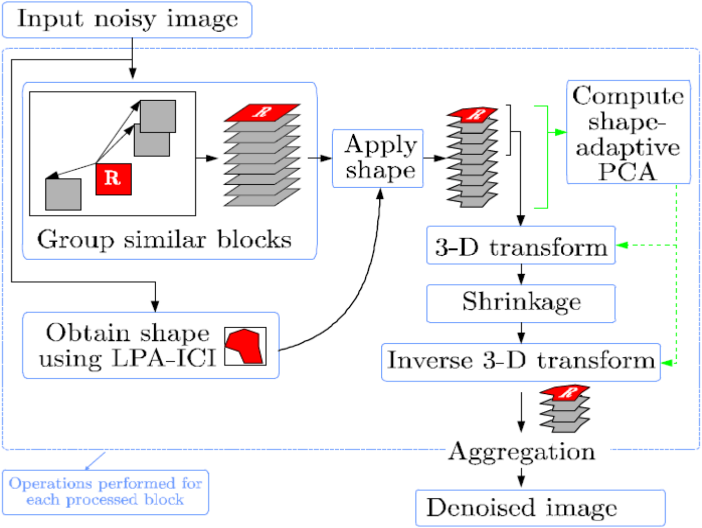
BM3D-SAPCA



4. Apply the 3-D transform on a group of adaptive-shape neighborhoods.

5. Attenuate noise by shrinakage (hard-thresholding or empirical Wiener filtering).

BM3D-SAPCA



6. Apply the inverse 3-D transform to obtain filtered neighborhoods,

7. Return the filtered neighborhoods to their original locations and aggregate in case of overlapping.

BM3D-SAPCA

The scheme is implemented in three iterations:

I: hard-thresholding, BM and PCA on noisy data

II: hard-thresholding, BM and PCA on estimate from I.

III: empirical Wiener filtering, BM and PCA on estimate from II.

Directional varying-scale LPA estimates

$$\hat{y}_{h,\theta_k} = z \circledast g_{h,\theta_k}$$

$$\text{scales:} \quad h \in \{h_1, \dots, h_J\} = H$$

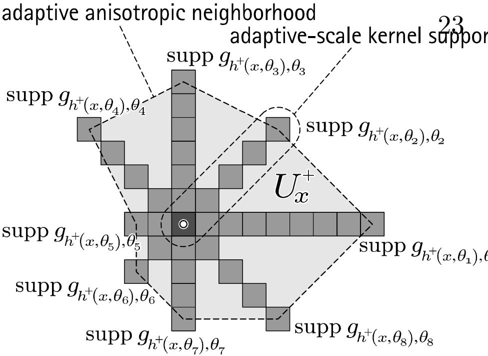
$$\text{directions:} \quad \theta_k = \frac{(k-1)}{4}\pi, \; k = 1, \dots, 8$$

ICI directional adaptive scales

$$\{h^+(x, \theta_k)\}_{k=1}^{8}$$

Adaptive neighborhood of the origin

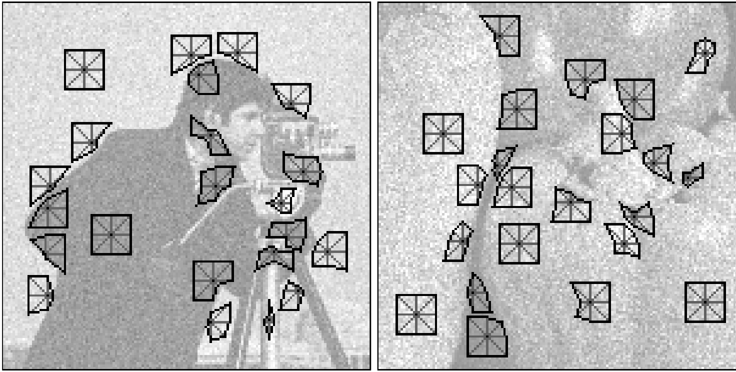
$$U_x^+ = \texttt{polygonal\_hull}\{\operatorname{supp} g_{h^+(x,\theta_k),\theta_k}\}_{k=1}^{8}$$



Adaptive neighborhood of estimation point x

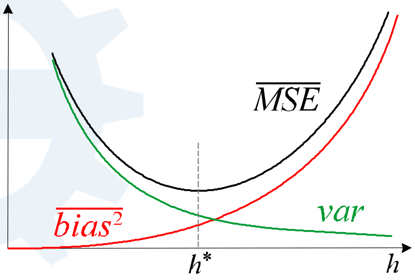
(mirror-translates)

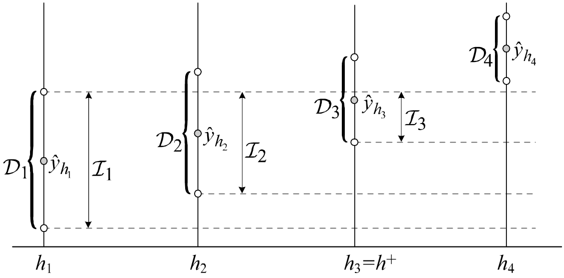
$$\tilde{U}_x^+ = \{v \in X : (x - v) \in U_x^+\}$$



Intersection of Confidence Intervals (*ICI*) (Goldenshluger&Nemirovski, 1997)

(for each fixed direction θ_k)





The estimates $\hat{y}_h(x)$ are calculated for a set $H = \{h_j\}_{j=1}^J$ of increasing scales. The *ICI* rule yields a pointwise adaptive estimate $\hat{y}_{h^+}(x)$, where for every x an adaptive scale $h^+(x) \in H$ is used such that $\hat{y}_{h^+}(x) \approx \hat{y}_{h^*(x)}(x)$.

ICI rule: *Consider the intersection of confidence intervals*

$$\mathcal{I}_j = \bigcap_{i=1}^{j} \mathcal{D}_i, \quad \text{where} \quad \mathcal{D}_i = \left[\hat{y}_{h_i}(x) - \Gamma\sigma_{\hat{y}_{h_i}}, \hat{y}_{h_i}(x) + \Gamma\sigma_{\hat{y}_{h_i}}\right]$$

and $\Gamma > 0$ *is a threshold parameter, and let* j^+ *be the largest of the indexes* j *for which* $\mathcal{I}_j$ *is non-empty,* $\mathcal{I}_{j^+} \neq \varnothing$ *and* $\mathcal{I}_{j^++1} = \varnothing$*. Then,* h^+ *is defined as* $h^+ = h_{j^+}$ *and the adaptive estimate is* $\hat{y}_{h^+}(x)$*.*

Block-matching

Adaptive neighborhoods can be *too small* for reliable matching!

Matching for $\tilde{U}_x^+$ needs to be carried out for a superset.

We use square blocks of size $(2h_{\max}-1)\times(2h_{\max}-1)$ centered at x, $h_{\max}=\max\{H\}$.

$$\text{Adaptive neighborhoods } \tilde{U}_x^+ \quad \forall x\in X$$
$$\text{Blocks } \tilde{B}_x \quad \forall x\in X_{\mathbb{B}}\subsetneq X$$

To every $x\in X$ we associate $x_{\mathbb{B}}\in X_{\mathbb{B}}$ such that $\|\delta_{\mathbb{B}}(x)\|_2$ of $\delta_{\mathbb{B}}(x)=x_{\mathbb{B}}-x$ is minimal.

The mapping $x\mapsto x_{\mathbb{B}}$ and $\delta_{\mathbb{B}}(x)$ are univocally defined (for convex X).

$\delta_{\mathbb{B}}(x)\neq 0$ only for x sufficiently close to the boundary ∂X of X.

Shape-adaptive grouping

For given points x, x_R define the translate of $\tilde{U}^+_{x_R}$

$$\tilde{U}^+_{x,x_R} = \{v \in X : (x-v) \in U^+_{x_R}\} = \left\{v \in X : (x_R - x + v) \in \tilde{U}^+_{x_R}\right\}.$$

$\tilde{U}^+_{x,x_R}$ is an adaptive neighborhood of x which uses the adaptive scales of the "reference point" x_R.

It can happen that $\tilde{U}^+_{x,x_R} \neq \tilde{U}^+_x$.

To a given "reference" point x_R we can now associate not only its own adaptive neighborhood $\tilde{U}^+_{x_R}$, but a collection (disjoint union) $\widetilde{\mathbb{U}}_{x_R}$ of neighborhoods defined as

$$\widetilde{\mathbb{U}}_{x_R} = \coprod_{x+\delta_{\mathbb{B}}(x_R) \in S_{x_R+\delta_{\mathbb{B}}(x_R)}} \tilde{U}^+_{x,x_R} = \left\{\tilde{U}^+_{x,x_R} : x + \delta_{\mathbb{B}}(x_R) \in S_{x_R+\delta_{\mathbb{B}}(x_R)}\right\},$$

where $S_{x_R+\delta_{\mathbb{B}}(x_R)}$ is the result of block-matching for $\tilde{B}_{x_R+\delta_{\mathbb{B}}(x_R)}$.

All neighborhoods in $\widetilde{\mathbb{U}}_{x_R}$ have the same shape, completely determined by adaptive scales $\{h^+(x_R, \theta_k)\}_{k=1}^{8}$ at x_R.

Shape-Adaptive PCA

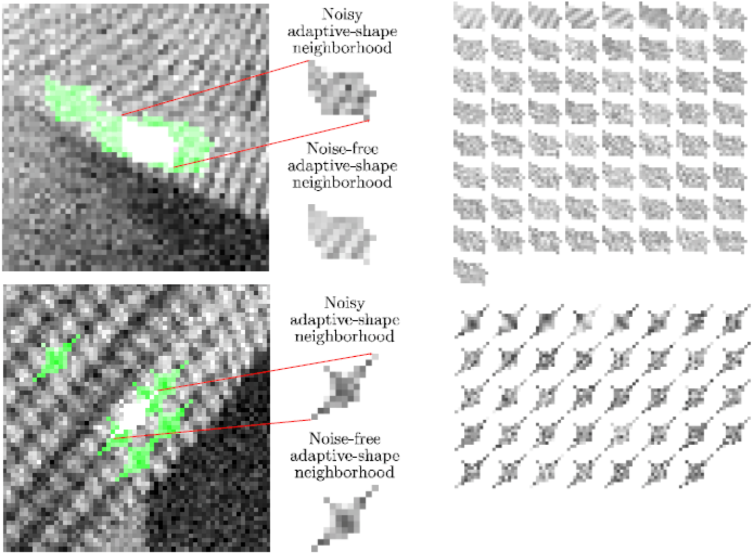
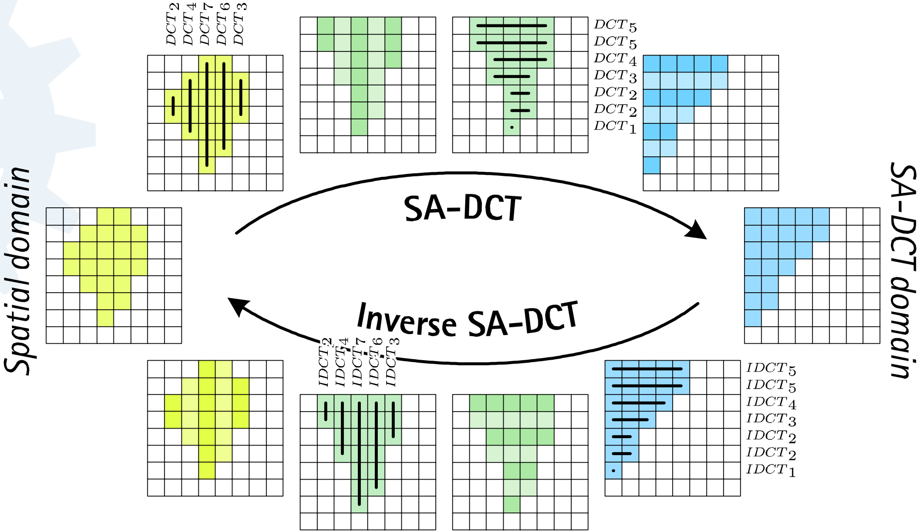


Fig. Illustration of the PCs (listed by decreasing eigenvalue magnitude) for two adaptive-shape neighborhoods. The green overlay shows the grouped similar neighborhoods.

Shape-Adaptive Discrete Cosine Transform (SA-DCT) (Sikora et al., 1995)



Shape-Adaptive Discrete Cosine Transform (SA-DCT) and its inverse. Transformation is computed by cascaded application of one-dimensional varying-length DCT transforms, along the columns and along the rows.

Shape-Adaptive Discrete Cosine Transform (SA-DCT)

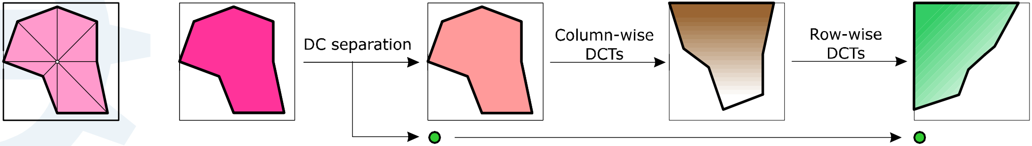
- direct generalization of the classical block-DCT (B-DCT);
- on rectangular domains (e.g., squares) the SA-DCT and B-DCT coincide;
- the same computational complexity as the B-DCT (separable);
- SA-DCT is part of the MPEG-4 standard;
- efficient (low-power) hardware implementations available;

- shape must be coded separately (constitutes some overhead).

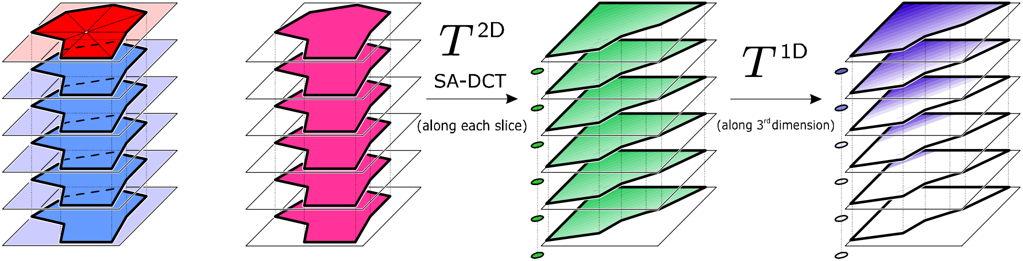
Orthonormal SA-DCT does not have a DC term and works best if applied on zero-mean data: "Orthonormal SA-DCT with DC separation and ΔDC compensation", Kauff et al. 1997.

SA-DCT (forward transform)

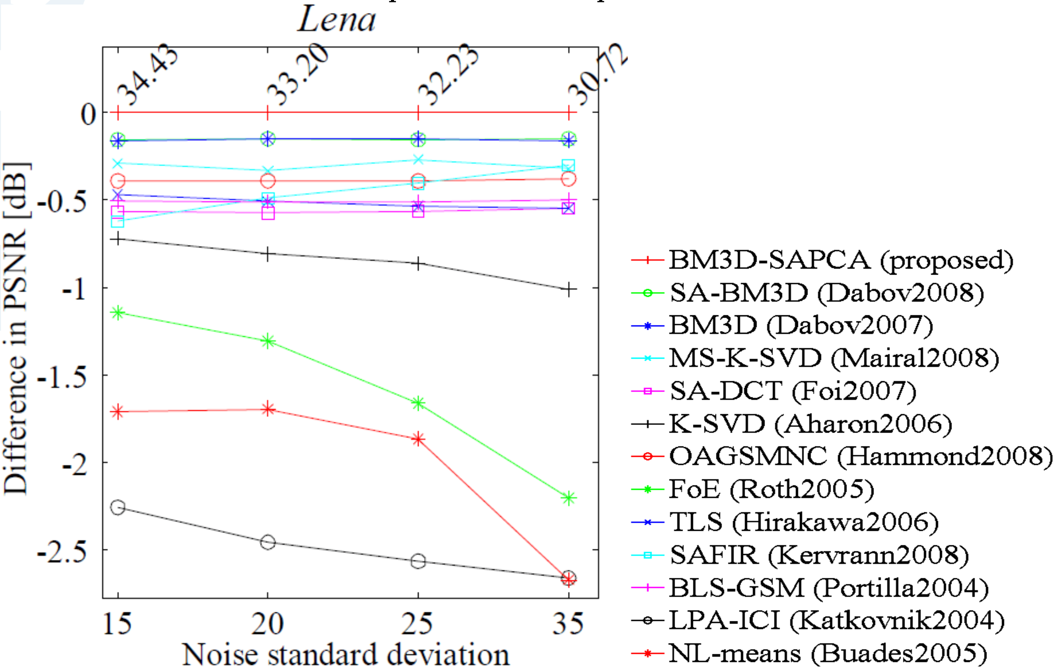
[as used in Pointwise SA-DCT denoising algorithm (Foi et al., IEEE TIP 2007)]



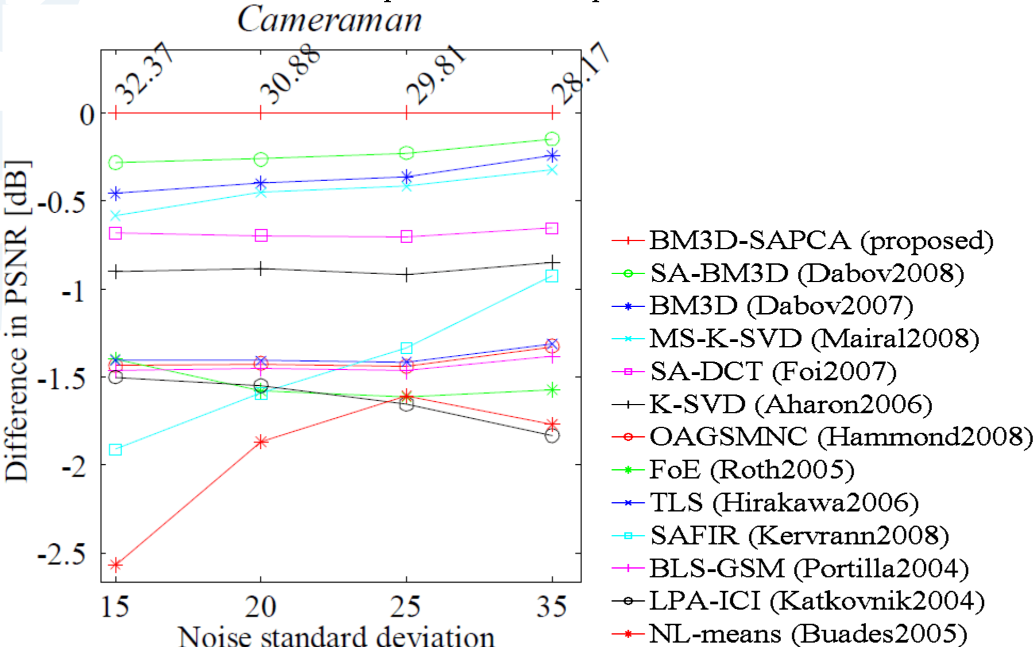
Shape-adaptive collaborative filtering (forward transform)



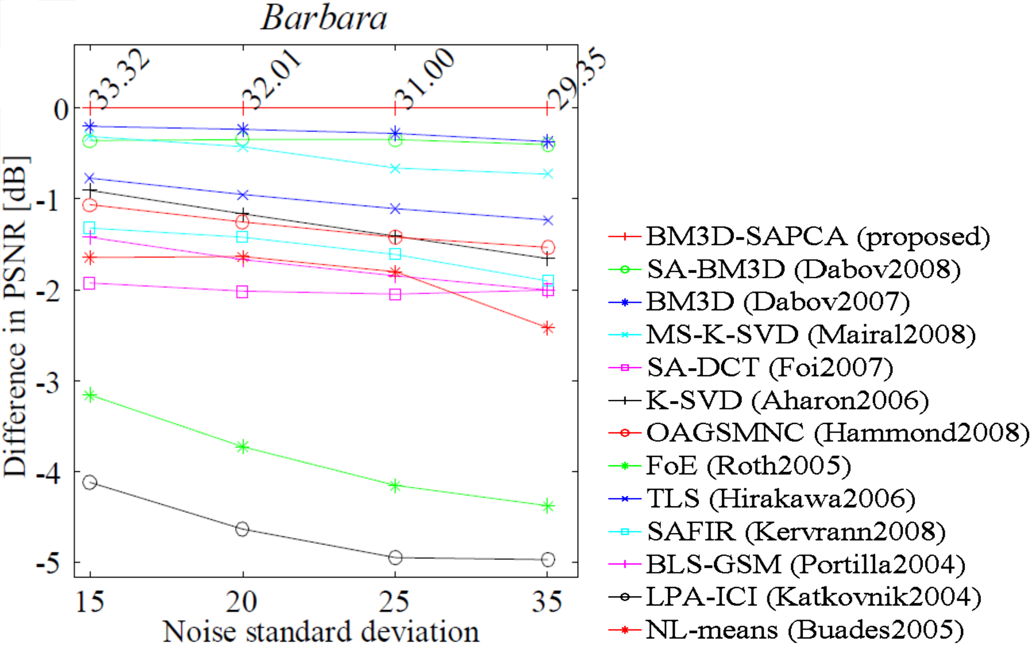
Experimental comparison



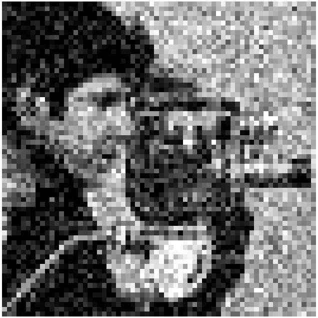
Experimental comparison



Experimental comparison



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)

2.1. Variance stabilization

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 $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_\theta\}$, where $\theta \in \Theta \subseteq \mathbb{R}$ denotes the parameter.

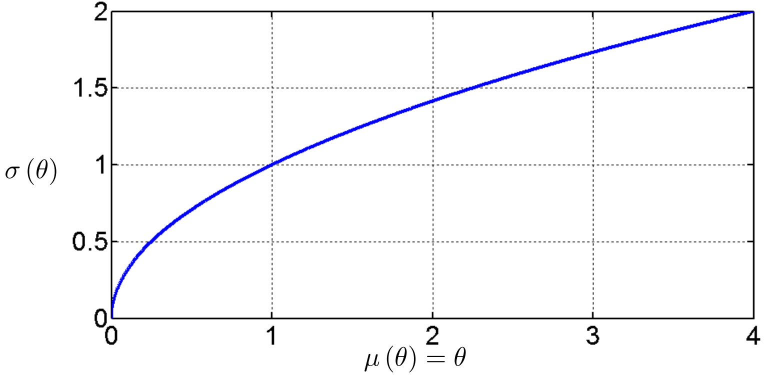
$$\mu(\theta) = E\{z|\theta\} \qquad \text{and} \qquad \sigma(\theta) = \operatorname{std}\{z|\theta\}$$

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

Example:

$\boldsymbol{\mathcal{D}}$ Poisson distributions with mean $\theta \in \Theta = [0, +\infty)$, $\Pr[z = \zeta|\theta] = e^{-\theta}\frac{\theta^\zeta}{\zeta!}$, $\zeta \in \mathbb{N}$.

We have $\mu(\theta) = \theta$ and $\sigma(\theta) = \sqrt{\theta}$.



One-parameter families of distributions

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

Variance stabilization problem

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

- 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.

Variance stabilization is typically impossible to achieve

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:

$$E\{g(z)|\theta\} = \theta g(1) + (1-\theta)g(0)$$

$$\mathrm{var}\{g(z)|\theta\} = E\left\{(g(z) - E\{g(z)|\theta\})^2|\theta\right\} = (g(0)-g(1))^2\theta(1-\theta).$$

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

In practice, we deal with either *approximate* or *asymptotic* stabilization.

Variance stabilization: history and examples

Classic **heuristic** stabilizer as indefinite integral form

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

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

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

then impose $\mathrm{std}\{f(z)|\theta\} = c$ and obtain the indefinite integral (1).

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).

Very rough, but useful as a first guess: nearly all classical stabilizers can be seen as a slight modification of (1).

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

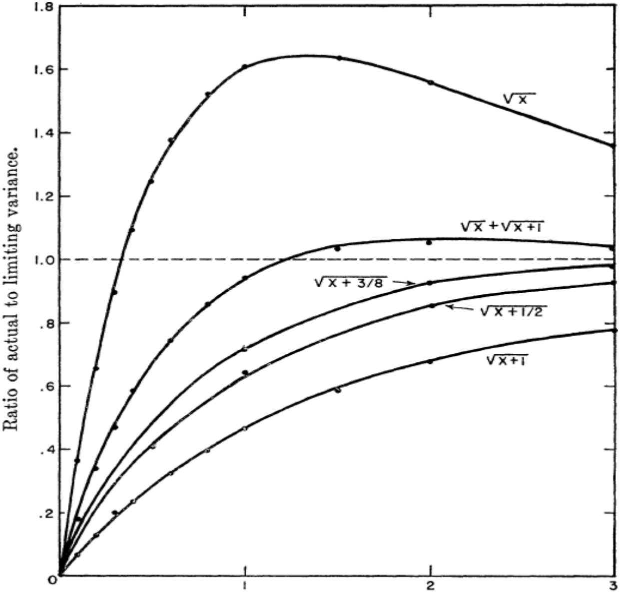


FIG. 2. Stabilization of Poisson variance.

M. Freeman and J. Tukey, "Transformations Related to the Angular and the Square Root", The Annals of Mathematical Statistics, vol. 21, no. 4, pp. 607-611, Dec. 1950.

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: $\sqrt{z} + \sqrt{z+1}$

Starck, Murtagh, and Bijaoui, 1998: generalization of Anscombe for linear combinations of Poisson variates.

All these results enjoy asymptotic optimality, but good stabilization for small θ is not achieved.

Fryzlewicz, Nason, et al. 2004-2008: wavelet-Fisz transforms that return spectra having approximately constant variance.

Kolaczyk 1999: threshold-correcting schemes.

Variance stabilization: three milestone works

- Curtiss 1943: general asymptotic theorems are proved.
  - gave theoretical support to empirical stabilizers that were already used (and also to others yet to appear).
- Efron 1981: existence of transformations for exact variance stabilization and/or perfect normalization.
  - formalizes sufficient conditions for existence of exact stabilizers ("general transformation families" framework), and provides their analytical expressions.
  - results are nonparametric and nonasymptotic.
  - difficult to use in practice (assumes too much smoothness and invertibilities of parametrized mappings).
- Tibshirani 1986: AVAS procedure for regression
  - approximate variance stabilizing transformations are iteratively computed by recursive application of the integral stabilizer (iterative refinement of the stabilizer) [Tibshirani fails to successfully use Efron's stabilizers on data]
  - developed for data-driven application, hints about potential use for random variables.
  - nonparametric and nonasymptotic.

2.2. Optimization of variance-stabilizing transformations

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

Motivation

With so many transformations, which one is the best?

This question remains largely unanswered.

- It is typically impossible to achieve simultaneously good stabilization for all parameter values (see Freeman & Tukey): thus, when a stabilizer appears to be better than another for some values of the parameter, it is likely that for other values it is actually worse. In this sense, there might be no "best stabilizer".

- No objective criterion for assessing the goodness of a stabilizer has ever been formulated. Simply demanding $\mathrm{std}\{f(z)\,|\theta\}$ to be as close as possible to c is too vague and ambiguous.

Variance stabilization as a minimization problem

Let

$$e_f(\theta) = \sigma_f(\theta) - c$$

be the local error because of inexact stabilization (where locality is intended by the conditioning on θ) and define a global cost functional as

$$C_f = \int |e_f(\theta)|\, d\theta. \tag{2}$$

We may formulate the variance stabilization problem as the solution of

$$\operatorname{argmin}_f C_f \tag{3}$$

Variance stabilization is exact only when $C_f = 0$ for some f.

Minimization needs to be constrained to some particular class of functions, such as strictly monotone, Lipschitz, smooth functions, etc.

Variance stabilization as a minimization problem

We have seen that it makes little sense to aim at *exact* variance stabilization simultaneously for *all* parameter values.

We consider a separable *weighted cost functional (stabilization functional)* of the form

$$C_f = \int_{\Theta} w_\theta\left(\theta\right) w_e\left(e_f\left(\theta\right)\right) d\theta, \tag{4}$$

where the weight functions w_θ and w_e provide different weighting for the different values of θ and different stabilization errors $e_f\left(\theta\right)$, respectively.

In particular, we design special weights w_e that *favor approximate stabilization while ignoring very large stabilization errors.*

Stabilization functional

Let $\gamma_\mathrm{u}, \gamma_\mathrm{l} \le 1$, $r'_\mathrm{u}, r'_\mathrm{l} \ge 0$, $r''_\mathrm{u} \ge r'_\mathrm{u}$, $r''_\mathrm{l} \ge r'_\mathrm{l}$, $o_\mathrm{u}, o_\mathrm{l} \ge 1$ be some real constants and χ_Ω be the characteristic (indicator) function of a set Ω.

We define the weights w_e as

$$w_e\left(e_f(\theta)\right) = \left|\varphi\left(\overline{e_f}(\theta)\right)\overline{e_f}(\theta)\right|,$$

where

$$\begin{aligned}
\overline{e_f}(\theta) &= \overline{\sigma_f}(\theta) - c = \max\left\{-r''_\mathrm{l}, \min\left\{r''_\mathrm{u}, e_f(\theta)\right\}\right\},\\
\overline{\sigma_f}(\theta) &= \max\left\{c - r''_\mathrm{l}, \min\left\{c + r''_\mathrm{u}, \sigma_f(\theta)\right\}\right\},
\end{aligned}$$

and with the function φ given by

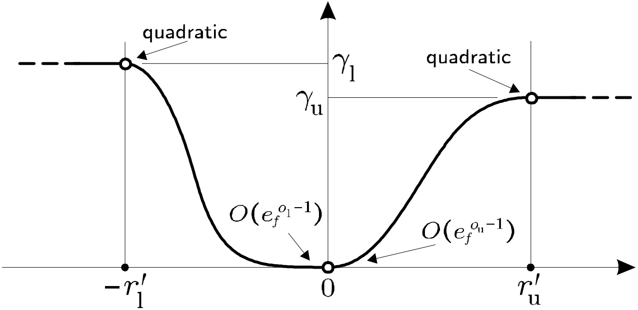
$$\begin{aligned}
\varphi(e_f) &= \gamma_\mathrm{u}\cdot\chi_{[0,+\infty)}(e_f)\left\{\left[1-\left(\frac{e_f - r'_\mathrm{u}}{r'_\mathrm{u}}\right)^2\right]^{(o_\mathrm{u}-1)}\chi_{(-\infty,r'_\mathrm{u})}(e_f) + \chi_{[r'_\mathrm{u},+\infty)}(e_f)\right\} +\\
&\quad +\gamma_\mathrm{l}\cdot\chi_{(-\infty,0)}(e_f)\left\{\left[1-\left(\frac{e_f + r'_\mathrm{l}}{r'_\mathrm{l}}\right)^2\right]^{(o_\mathrm{l}-1)}\chi_{(-r'_\mathrm{l},+\infty)}(e_f) + \chi_{(-\infty,-r'_\mathrm{l}]}(e_f)\right\}.
\end{aligned}$$

Stabilization functional

The clipped argument $\overline{e_f}(\theta)$ cannot distinguish stabilization errors larger than r''_{l},r''_{u}, while the multiplication against the function φ increases the order of the stabilization errors from 1 to o_{l},o_{u}. Note that for a positive (resp. negative) argument, the function φ has a zero of order $o_{\mathrm{u}} - 1$ ($o_{\mathrm{l}} - 1$) at zero and becomes constant (with quadratic-smooth joint) equal to γ_{u} (γ_{l}) starting from r'_{u} (r'_{l}).

Thus, the cost functional (4) takes the form

$$C_f = \int_{\Theta} w_\theta(\theta) \left|\varphi\left(\overline{e_f}(\theta)\right) \overline{e_f}(\theta)\right| d\theta.$$



The function φ.

Iterative integral algorithm for optimizing f

0. Initialize

$f_0(z) = z$ (identity) or an arbitrary (non-optimal) stabilizer
f monotone increasing

Iterate the following three stages:

1. Compute statistics

$\vartheta_k(\theta) = \text{med}\{f_k(z)|\theta\} = f_k(\text{med}\{z|\theta\})$
$\sigma_k(\theta) = \text{std}\{f_k(z)|\theta\}$

2. Compute stabilization refinement

$r_k(z) = \int^z I_k(\theta)\, d[\vartheta_k(\theta)]$ (integration with respect to the median)

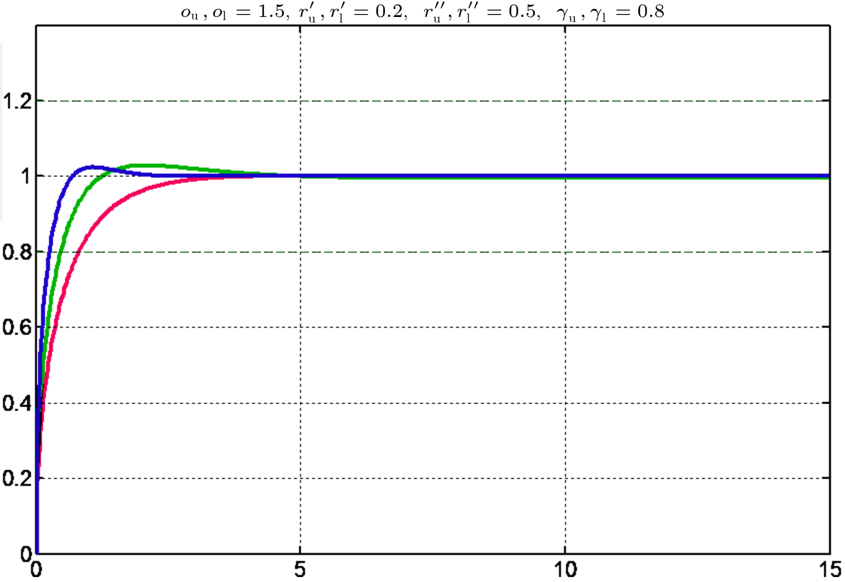
where

$$I_k(\theta) = 1 - \frac{w_\theta(\theta)\,\varphi(\overline{e_k}(\theta))\,\overline{e_k}(\theta)}{\overline{\sigma_k}(\theta)},$$
$$\overline{e_k}(\theta) = \overline{\sigma_k}(\theta) - c = \max\{-r''_{\text{l}}, \min\{r''_{\text{u}}, e_k(\theta)\}\}$$
$$\overline{\sigma_k}(\theta) = \max\{c - r''_{\text{l}}, \min\{c + r''_{\text{u}}, \sigma_{fk}(\theta)\}\},$$

3. Compose

$f_{k+1}(z) = r_k(f_k(z))$

Optimization of Poisson stabilizer (iterative integral)[52]

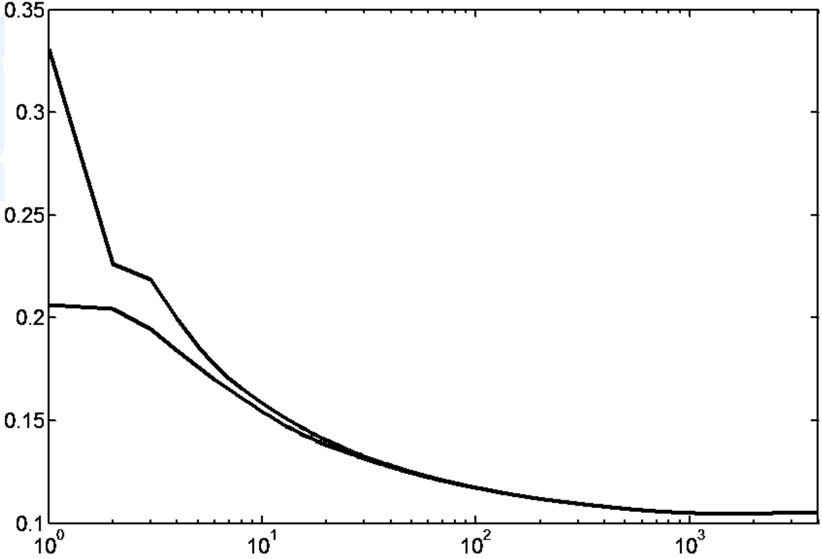


Standard-deviation functions $\sigma(\theta)$

Green: $f_0 = \sqrt{z} + \sqrt{z+1}$ Red: $f_0 = 2\sqrt{z + 3/8}$ Blue: optimized

Optimization of Poisson stabilizer (iterative integral)

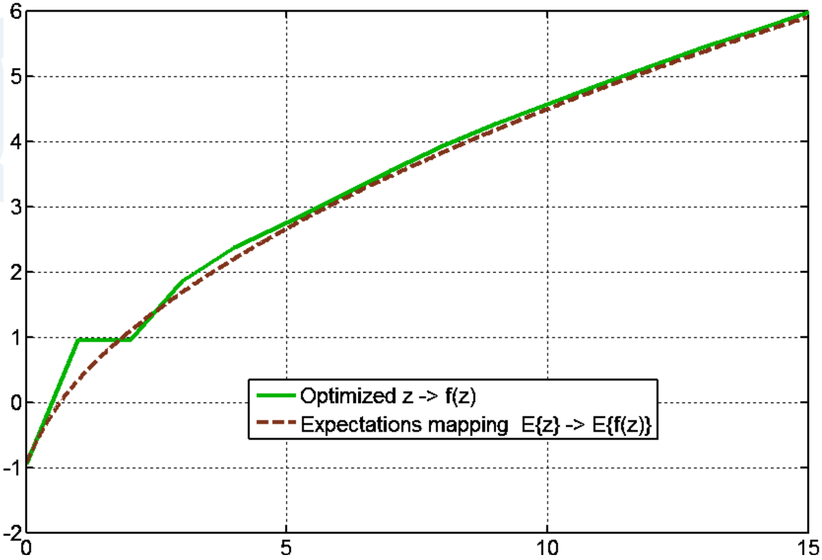
$o_u, o_l = 1.5,\ r'_u, r'_l = 0.2,\ \ r''_u, r''_l = 0.5,\ \ \gamma_u, \gamma_l = 0.8$



stabilization functional vs. iterations (log scale)
$f_0 = \sqrt{z} + \sqrt{z+1}$ (lower) and $f_0 = 2\sqrt{z + 3/8}$ (higher)

Optimization of Poisson stabilizer (iterative integral)

$o_{\mathrm{u}}, o_{\mathrm{l}} = 1.5,\ r'_{\mathrm{u}}, r'_{\mathrm{l}} = 0.2,\ \ r''_{\mathrm{u}}, r''_{\mathrm{l}} = 0.5,\ \ \gamma_{\mathrm{u}}, \gamma_{\mathrm{l}} = 0.8$



variance-stabilizer f and the mapping $E\{z|\theta\} \mapsto E\{f(z)|\theta\}$

stabilization functional = 01.051

Optimization by iterative integral vs. direct search

Convergence of the iterative integral algorithm was verified experimentally, up to the numerical precision of the algorithm, in extensive tests.

However, its limit does not necessarily coincide with the minimizer of the stabilization functional.

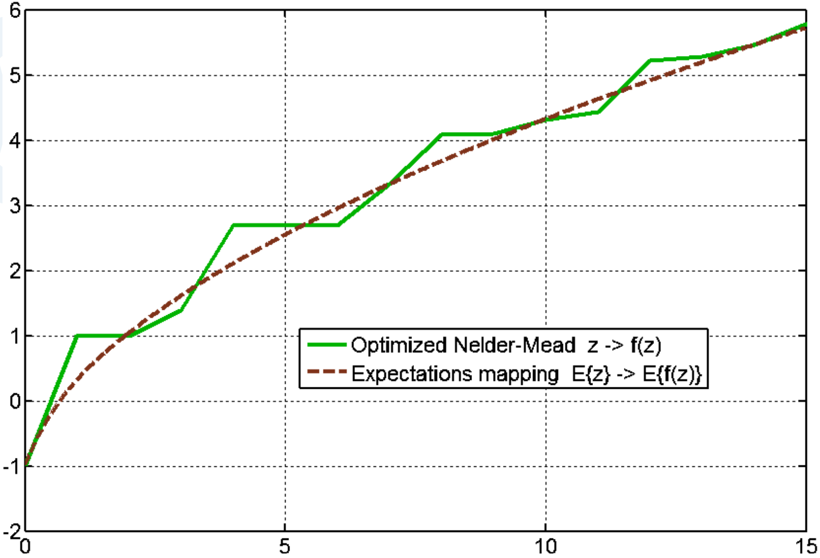
— computational aspects involved in the evalutation of the integrals
— unless the class of distributions and allowed stabilizers are reduced to non-interesting cases, a proof of minimization seems very difficult to achieve (similar situation as for AVAS algorithm)

A practical way to circumvent these issues is to solve the minimization by direct search, which is particularly feasible for discrete distributions.

We use Nelder-Mead downhill simplex algorithm.

Optimization by direct search

$o_{\mathrm{u}}\,, o_{\mathrm{l}} = 1.5,\ r'_{\mathrm{u}}\,, r'_{\mathrm{l}} = 0.2,\ \ r''_{\mathrm{u}}\,, r''_{\mathrm{l}} = 0.5,\ \ \gamma_{\mathrm{u}}\,, \gamma_{\mathrm{l}} = 0.8$

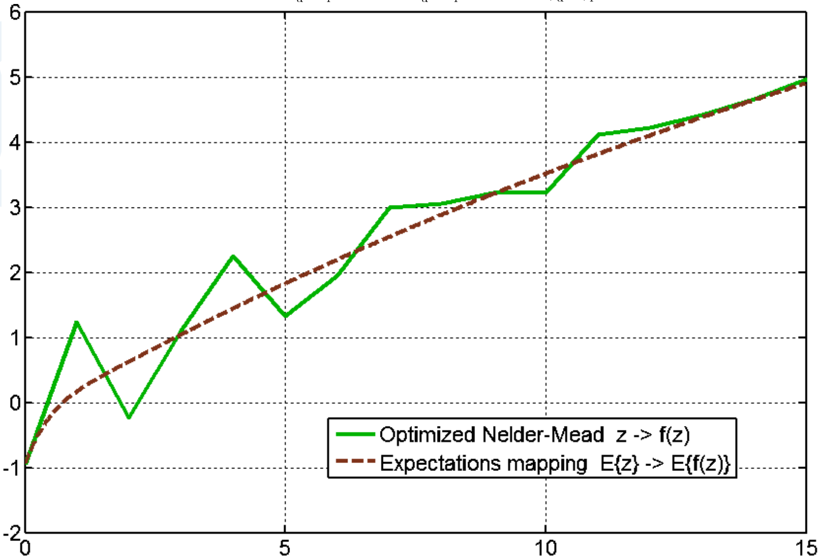


variance-stabilizer f and the mapping $E\{z|\theta\} \mapsto E\{f(z)|\theta\}$

stabilization functional $= 0.096$

Optimization by direct search: relaxing monotonicity

$o_{\mathrm{u}}, o_{\mathrm{l}} = 1.5,\ r'_{\mathrm{u}}, r'_{\mathrm{l}} = 0.2,\ \ r''_{\mathrm{u}}, r''_{\mathrm{l}} = 0.5,\ \ \gamma_{\mathrm{u}}, \gamma_{\mathrm{l}} = 0.8$

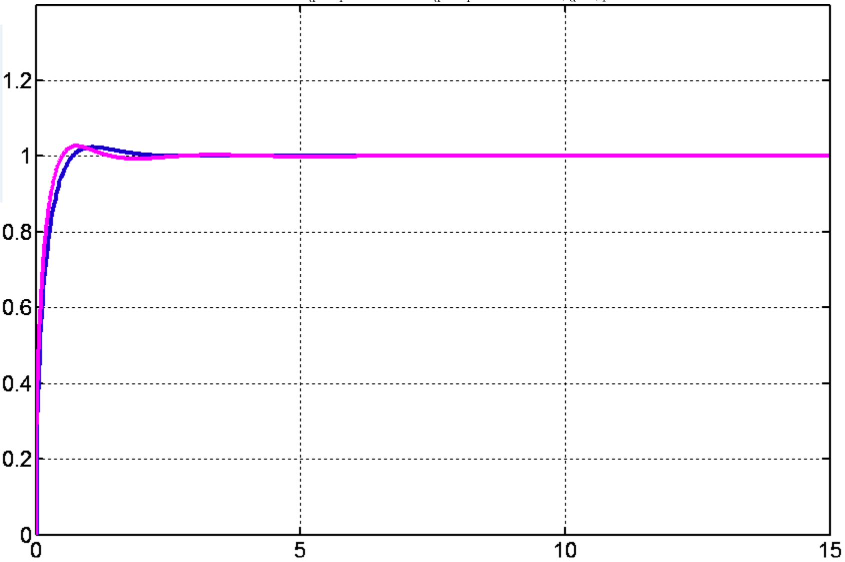


variance-stabilizer f and the mapping $E\{z|\theta\} \mapsto E\{f(z)|\theta\}$

stabilization functional $= 0.079$

Optimization by direct search

$o_u\,, o_l\; = 1.5,\; r'_u\,, r'_l\; = 0.2,\;\; r''_u\,, r''_l\; = 0.5,\;\; \gamma_u\,, \gamma_l\; = 0.8$



Standard-deviation functions $\sigma(\theta)$

Purple: optimized by direct-search non-monotonic Blue: optimized by recursive integral

3. Application to raw-data denoising

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.
Foi, A., "Practical denoising of clipped or overexposed noisy images", *Proc. 16th European Signal Process. Conf., EUSIPCO 2008*, Lausanne, Switzerland, August 2008.
Foi, A., "Clipped or overexposed noisy images: heteroskedastic modeling and practical denoising", preprint, submitted to *Signal Processing*.

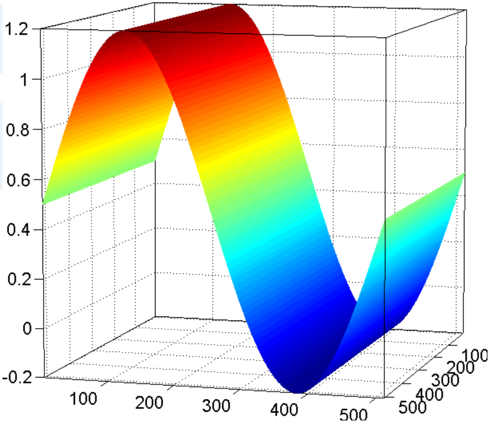
Raw data as clipped signal-dependent observations

$$\tilde{z}(x) = \max\{0, \min\{z(x), 1\}\}, \qquad x \in X \subset \mathbb{Z}^2,$$

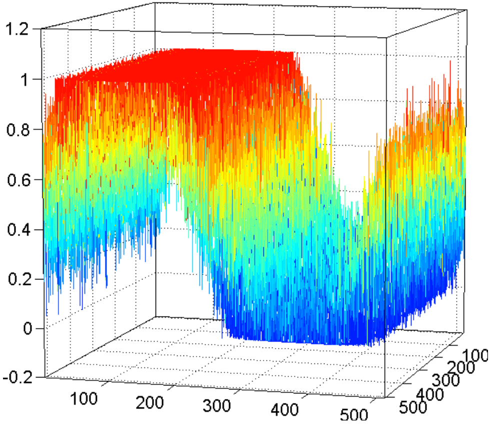
$$z(x) = y(x) + \sigma(y(x))\,\xi(x)$$

$y : X \to Y \subseteq \mathbb{R}$	unknown original image (deterministic)
$\sigma(y(x))\,\xi(x)$	zero-mean random error
$\sigma : \mathbb{R} \to \mathbb{R}^+$	standard-deviation function (deterministic)
$\xi(x)$	random variable $E\{\xi(x)\} = 0 \quad \mathrm{var}\{\xi(x)\} = 1$
$y(x) = E\{z(x)\}$	expectation
$\sigma(y(x)) = \mathrm{std}\{z(x)\}$	standard deviation

Clipped noisy data



original



added noise and then clipped

Raw data as clipped signal-dependent observations

$$\tilde{z}(x) = \max\{0, \min\{z(x), 1\}\}, \qquad x \in X \subset \mathbb{Z}^2,$$

$$z(x) = y(x) + \sigma(y(x))\,\xi(x)$$

$$\boxed{\tilde{z}(x) = \tilde{y}(x) + \tilde{\sigma}(\tilde{y}(x))\,\tilde{\xi}(x)}$$

$\tilde{y}(x) = E\{\tilde{z}(x)\}$	expectation
$\tilde{\sigma} : [0,1] \to \mathbb{R}^+$	standard-deviation function (of expectation)
$\tilde{\sigma}(\tilde{y}(x)) = \mathrm{std}\{\tilde{z}(x)\}$	standard deviation

Modeling raw-data signal-dependence before clipping

The random error before clipping is composed of two mutually independent parts:

$$\sigma\left(y\left(x\right)\right)\xi\left(x\right)=\eta_{\mathrm{p}}\left(y\left(x\right)\right)+\eta_{\mathrm{g}}\left(x\right)$$

η_{p}	*Poissonian* signal-dependent component	(photonic)
η_{g}	*Gaussian* signal-independent component	(everything else)

$$\begin{aligned}\big(y\left(x\right)+\eta_{\mathrm{p}}\left(y\left(x\right)\right)\big)\chi &\sim \mathcal{P}\left(\chi y\left(x\right)\right), &\quad \chi>0\\ \eta_{\mathrm{g}}\left(x\right) &\sim \mathcal{N}\left(0,b\right), &\quad b>0\end{aligned}$$

$$\sigma^2(y(x))=ay(x)+b,\qquad a=\chi^{-1}$$

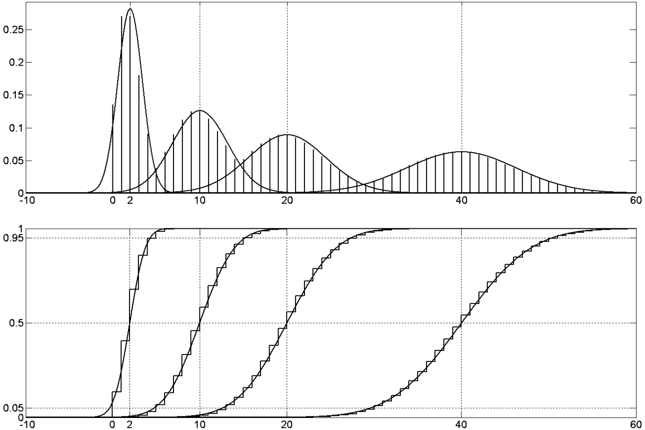
$\mathcal{N}\left(\mu,\sigma^2\right)$ normal (=Gaussian) distribution with mean μ and variance σ^2

$\mathcal{P}\left(\lambda\right)$ Poisson distribution with mean (and variance) λ

Normal approximation of Poisson variates

$\zeta \sim \mathcal{P}(\lambda)$ means the probability $\Pr[\zeta = k] = e^{-\lambda}\dfrac{\lambda^k}{k!}, k \in \mathbb{N}$.

$\zeta \sim \mathcal{N}(\mu, \sigma^2)$ means the probability density of z is $\wp(\zeta) = \dfrac{1}{\sigma\sqrt{2\pi}} e^{-\dfrac{(\zeta-\mu)^2}{2\sigma^2}}, \quad \zeta \in \mathbb{R}$.



p.d.f. (top) and c.d.f. (bottom) for $\mathcal{P}(\lambda)$ and $\mathcal{N}(\lambda, \lambda)$, $\lambda = 2, 10, 20, 40$.

Heteroskedastic normal approximation

$$\tilde{z}(x) = \max\{0, \min\{z(x), 1\}\}, \qquad x \in X \subset \mathbb{Z}^2,$$

$$z(x) = y(x) + \sigma(y(x))\,\xi(x)$$

$$\boxed{\sigma(y(x))\,\xi(x) = \sqrt{ay(x) + b}\,\xi(x), \quad \xi(x) \sim \mathcal{N}(0,1)}$$

(Generalized) Probability distributions

Before clipping:

$$\wp_z(\zeta) = \frac{1}{\sigma(y)}\,\phi\!\left(\frac{\zeta-y}{\sigma(y)}\right)$$

After clipping:

$$\wp_{\tilde{z}}(\zeta) = \Phi\!\left(\frac{-y}{\sigma(y)}\right)\delta_0(\zeta) + \frac{1}{\sigma(y)}\,\phi\!\left(\frac{\zeta-y}{\sigma(y)}\right)\chi_{[0,1]} + \left(1-\Phi\!\left(\frac{1-y}{\sigma(y)}\right)\right)\delta_0(1-\zeta)$$

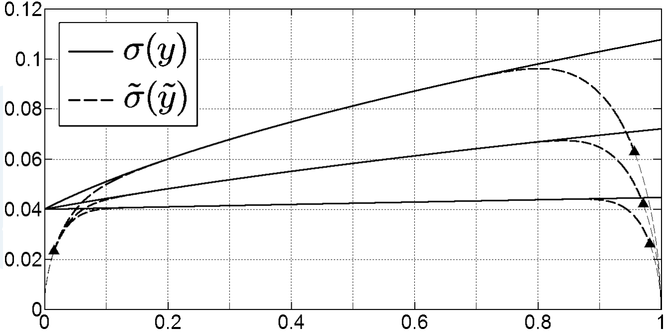
ϕ and Φ are p.d.f. and c.d.f. of $\mathcal{N}(0,1)$

δ_0 is Dirac delta function $\quad$ $\chi_{[0,1]}$ is characteristic (=indicator) function of interval $[0,1]$

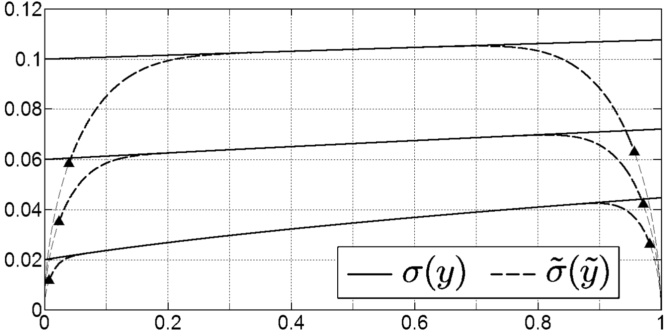
Expectations and variances

$$E\{\tilde{z}|y\} = \tilde{y} = \Phi\left(\tfrac{y}{\sigma(y)}\right) y - \Phi\left(\tfrac{y-1}{\sigma(y)}\right)(y-1) + \sigma(y)\,\phi\left(\tfrac{y}{\sigma(y)}\right) - \sigma(y)\,\phi\left(\tfrac{y-1}{\sigma(y)}\right),$$

$$\begin{aligned}\operatorname{var}\{\tilde{z}|y\} = \tilde{\sigma}^2(\tilde{y}) = \Phi\left(\tfrac{y}{\sigma(y)}\right)\left(y^2 - 2\tilde{y}y + \sigma^2(y)\right) + \\ + \tilde{y}^2 - \Phi\left(\tfrac{y-1}{\sigma(y)}\right)\left(y^2 - 2\tilde{y}y + 2\tilde{y} + \sigma^2(y) - 1\right) + \\ + \sigma(y)\,\phi\left(\tfrac{y-1}{\sigma(y)}\right)(2\tilde{y} - y - 1) - \sigma(y)\,\phi\left(\tfrac{y}{\sigma(y)}\right)(2\tilde{y} - y).\end{aligned}$$

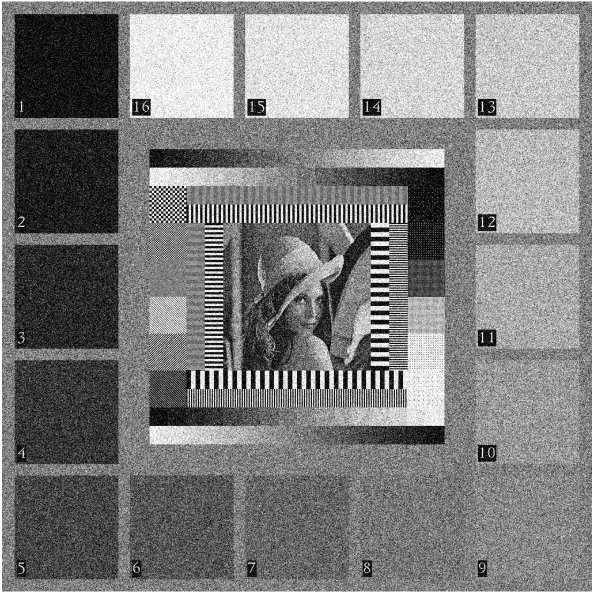


$$a = 0.02^2, 0.06^2, 0.10^2$$
$$b = 0.04^2$$



$$a = 0.04^2$$
$$b = 0.02^2, 0.06^2, 0.10^2$$

Noisy image $\tilde{z}$



PSNR=15.00dB noise parameters $a = 0,\ \ b = 0.2^2$

Denoising

Denoising heteroskedaskic data using variance-stabilization and conventional denoising algorithm for AWGN.

Main stages:

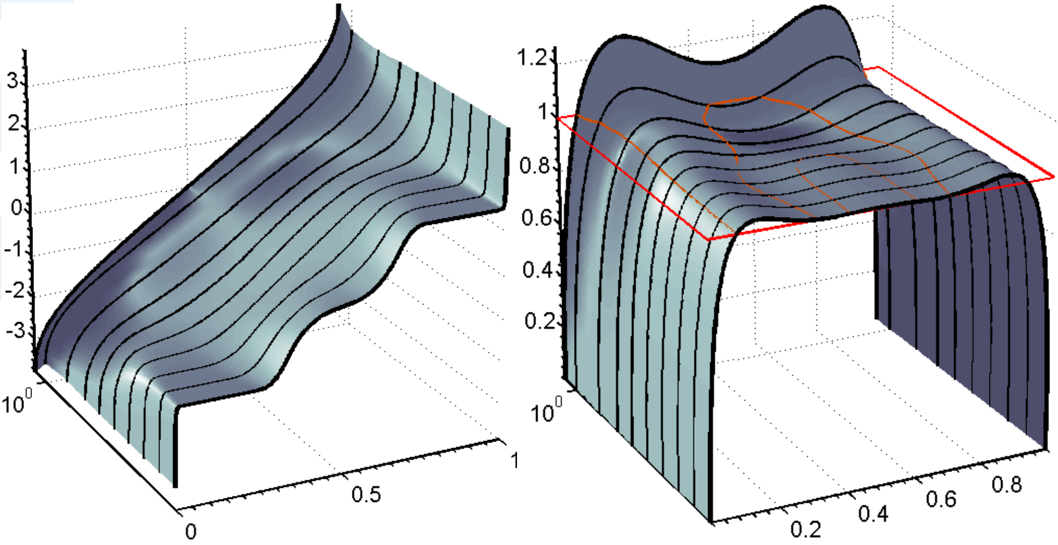
1. variance-stabilization
2. denoising (BM3D public code for AWGN from www.cs.tut.fi/~foi/GCF-BM3D/)
3. inversion of the stabilizer (from $E\{f(\tilde{z})\} \mapsto y$)

We compare two alternatives stabilizers:

$$f_0(t) = \int_{t_0}^{t} \frac{c}{\tilde{\sigma}(\tilde{y})} d\tilde{y}, \qquad t, t_0 \in [0,1]$$

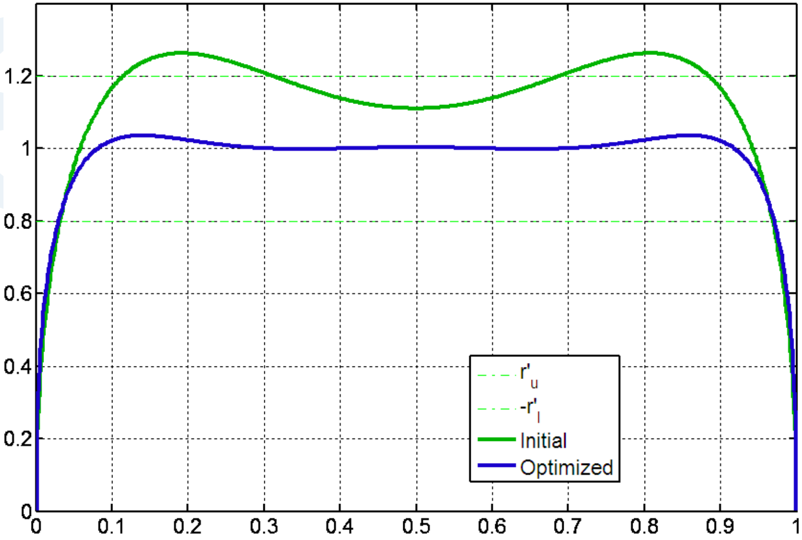
f_{2000} optimization by iterative integral.

Variance stabilization



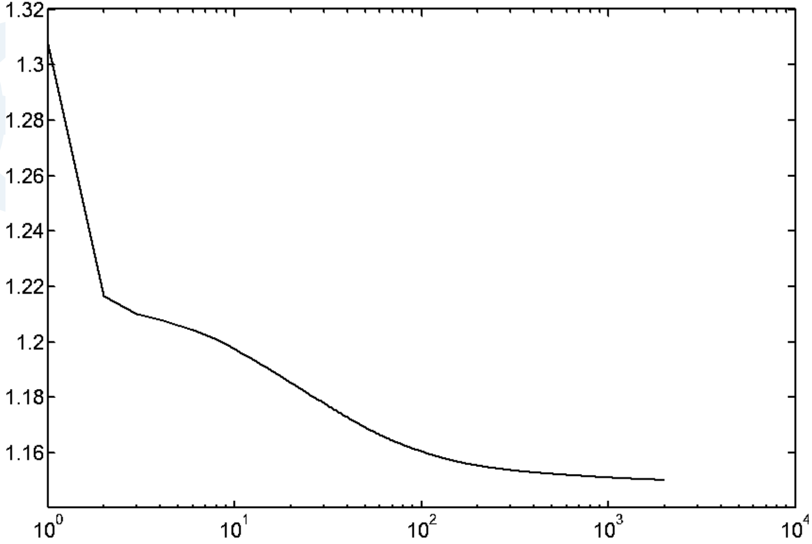
f_k $\quad$ $\mathrm{std}\left\{f_k\left(\tilde{z}\right)|\tilde{y}\right\}$

Variance stabilization



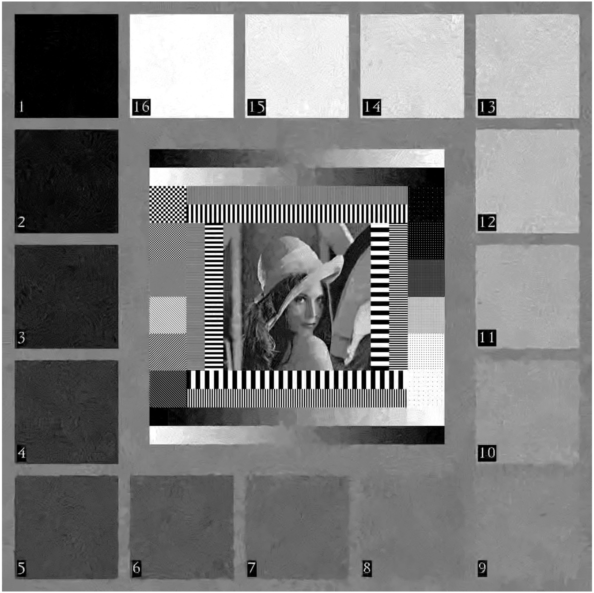
$\mathrm{std}\left\{f_0\left(\tilde{z}\right)|\tilde{y}\right\}$ vs $\mathrm{std}\left\{f_{2000}\left(\tilde{z}\right)|\tilde{y}\right\}$

Variance stabilization



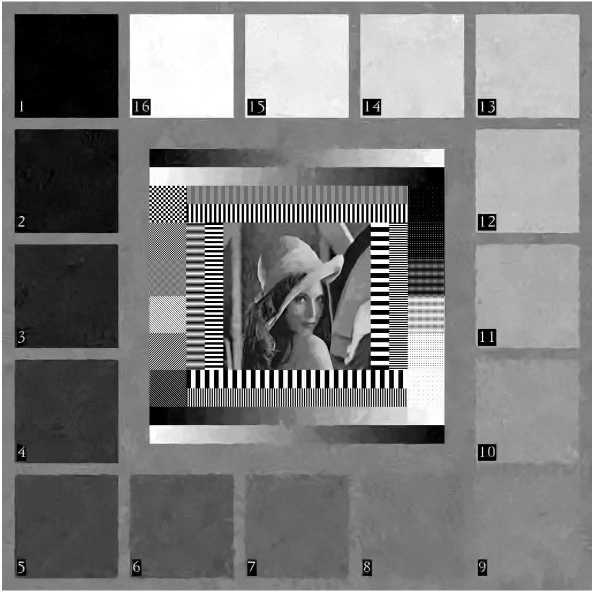
Convergence of the iterative integral algorithm

Denoised using f_0 as stabilizer



PSNR=29.37

Denoised using f_{2000} as stabilizer

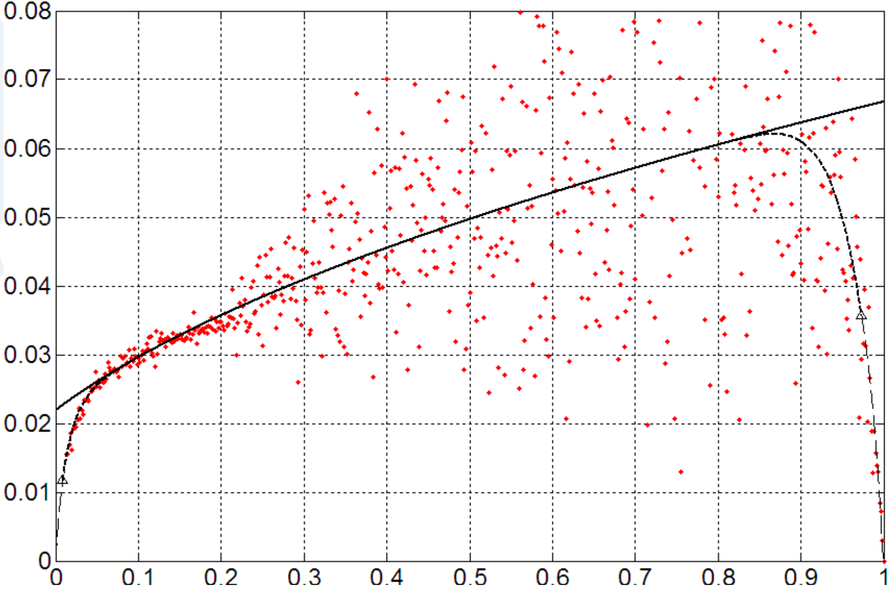


PSNR=30.67 [1.3dB gain]

Noisy raw-data image z

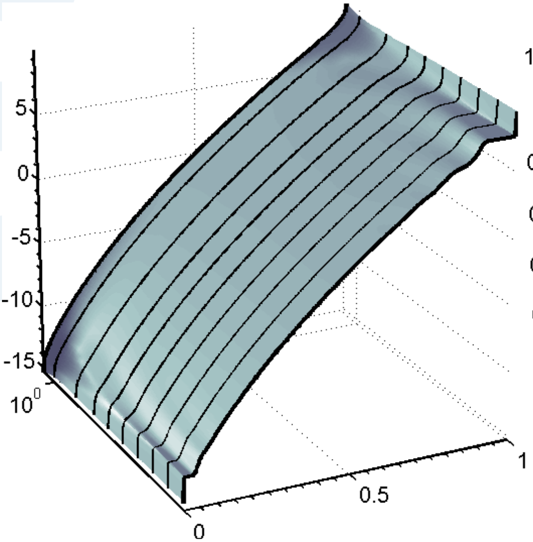
Fujifilm FinePix S9600 (green channel)

Noise estimation

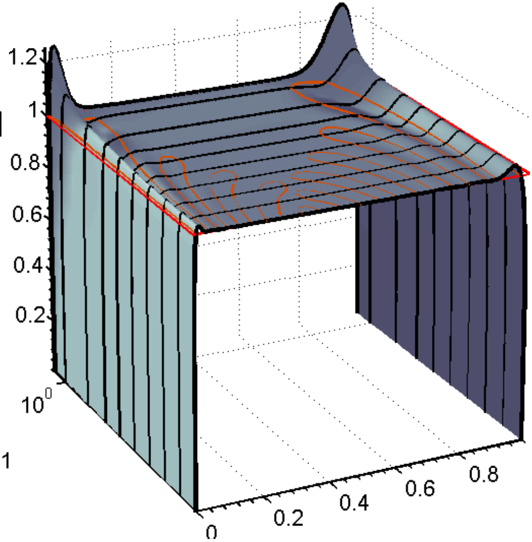


$a = 0.003978$ $b = 0.0004787$

Variance stabilization

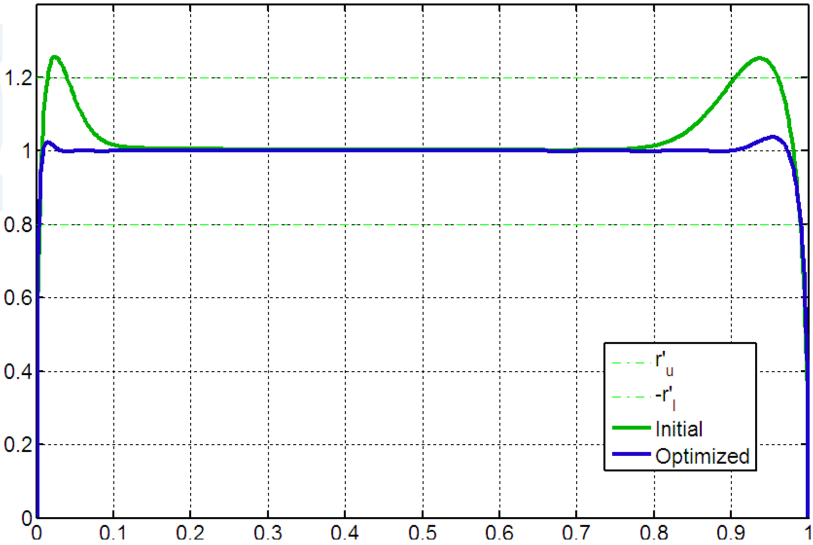


f_k



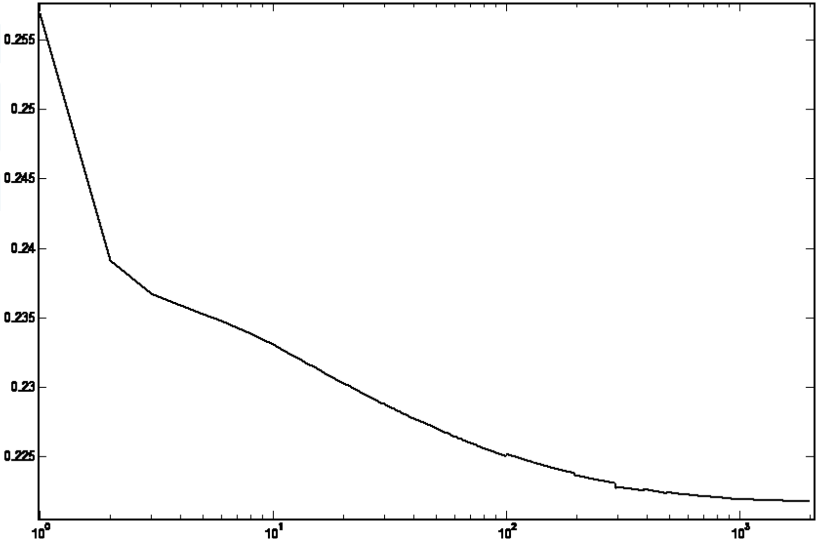
$\mathrm{std}\left\{f_k\left(\tilde{z}\right)|\tilde{y}\right\}$

Variance stabilization



$$\mathrm{std}\left\{f_0\left(\tilde{z}\right)|\tilde{y}\right\} \quad \text{vs} \quad \mathrm{std}\left\{f_{2000}\left(\tilde{z}\right)|\tilde{y}\right\}$$

Variance stabilization



Convergence of the iterative integral algorithm

Denoised using f_0 as stabilizer

Denoised using f_{2000} as stabilizer

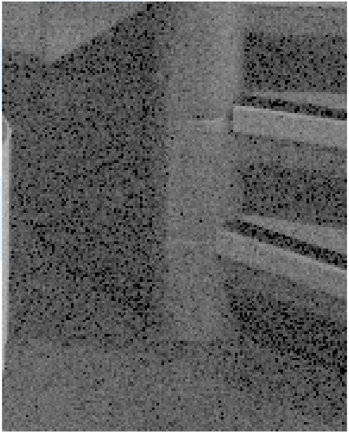
Denoised using f_0 as stabilizer

(gamma-corrected)

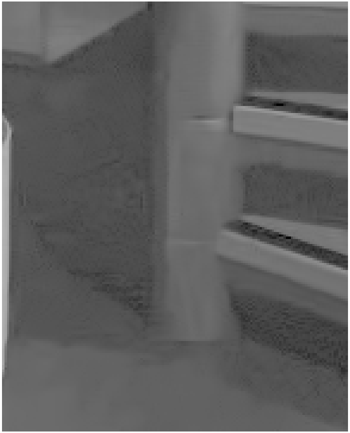
Denoised using f_{2000} as stabilizer

(gamma-corrected)

Comparison of fragments (1/3)



noisy



using f_0

using $\boldsymbol{f_{2000}}$

(all gamma-corrected)

Comparison of fragments (2/3)

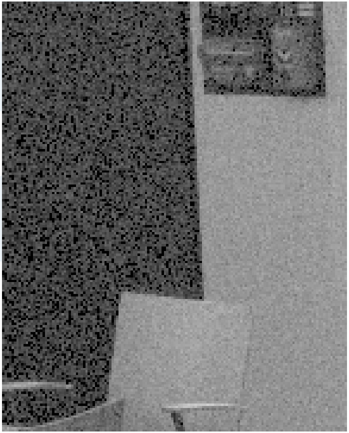
noisy

using f_0

using f_{2000}

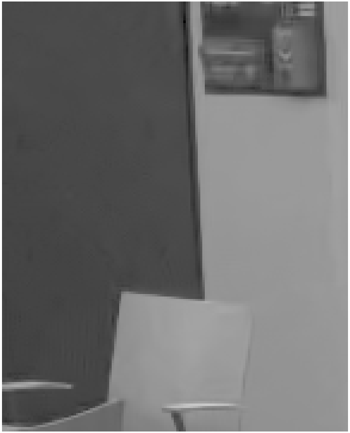
(all gamma-corrected)

Comparison of fragments (3/3)



noisy

using f_0



using f_{2000}

(all gamma-corrected)

Thank you!

LNLA 2009

2009 International Workshop on Local and Non-Local Approximation in Image Processing

August 19-21, 2009, Gustavelund Conference Hotel, Tuusula, Finland (near Helsinki)

`http://sp.cs.tut.fi/ticsp/lnla09`

Tampere International Center for Signal Processing