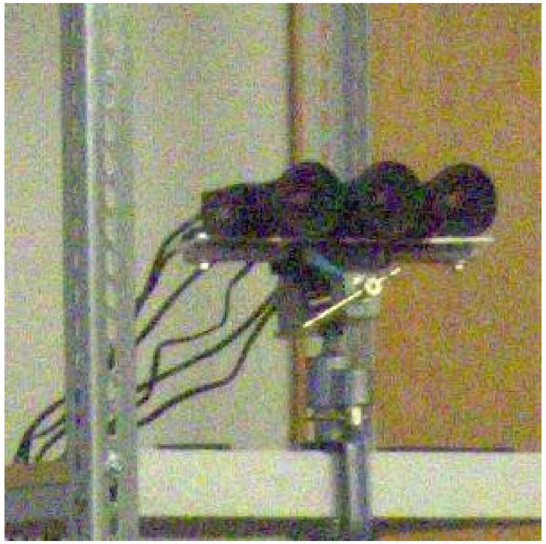
Image Denoising and Enhancement

Karen Egiazarian (TUT, NI)

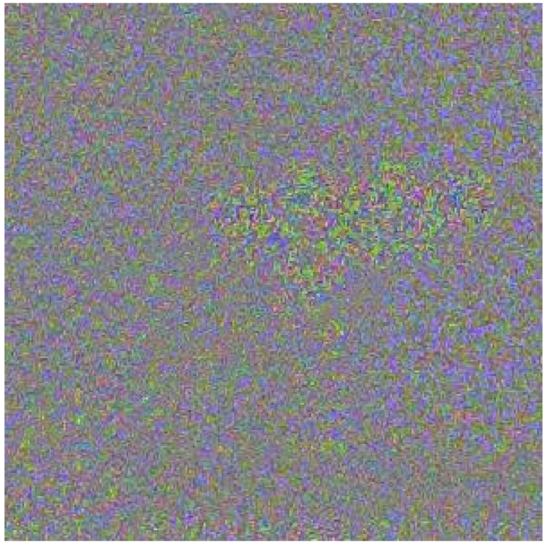
Image denoising: motivating example

- **Images are inevitably corrupted by various degradations and particularly by noise.**
- **Megapixels race: Pixels are getting smaller, and images even noisier**

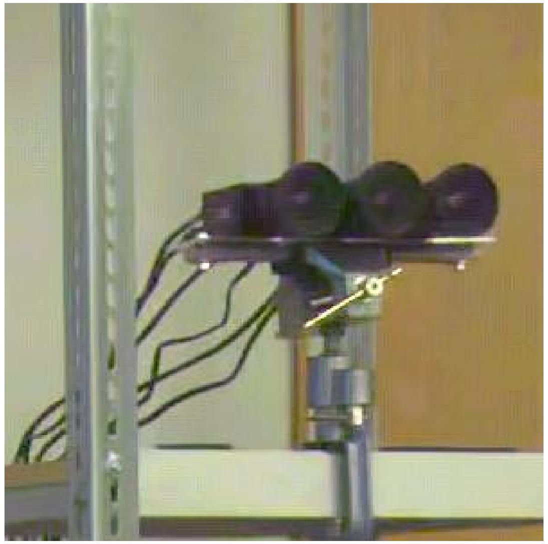
image **noise** **denoised image**



–



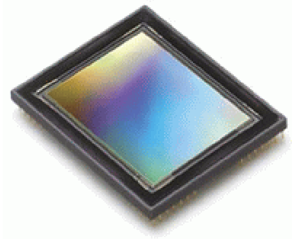
=



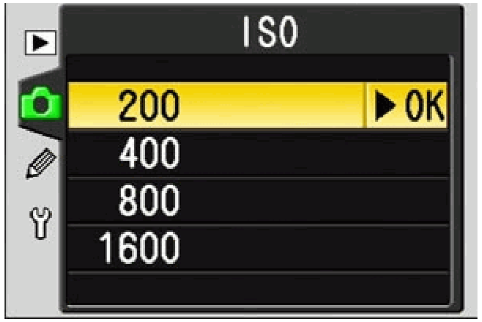
Canon Powershot A590IS ISO 800

Imaging Sensors: Exposure-time/noise trade-off

Digital imaging sensors can have very different performance



Different acquisition settings result in different noise levels in the image



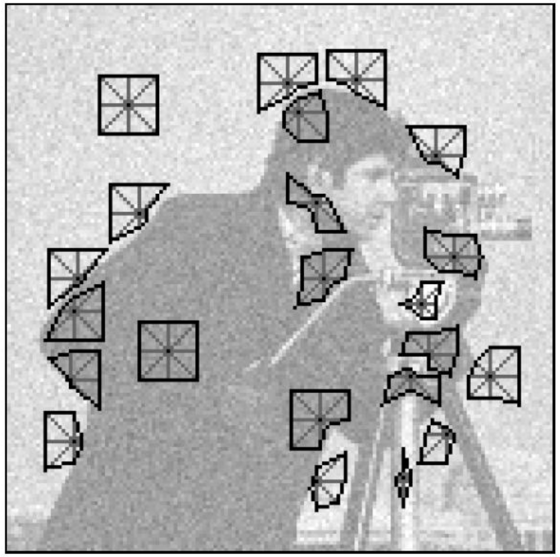
“Exposure-time/noise trade-off “

Outline

- **Intro**
- **Signal-dependent noise modeling and removal for digital imaging sensors**
- **Local polynomial approximations (LPA-ICI)**
- **Advanced image processing techniques:**
  - **- shape-adaptive methods**
  - **- nonlocal transform-based methods**
- **Applications:**
  - **- denoising**
  - **- deblurring**
  - **- deblocking**
  - **- super-resolution/zooming**

Outline

- Signal-dependent noise modeling and removal for digital imaging sensors
- Local polynomial approximations (LPA-ICI)
- Advanced image processing techniques:
  - shape-adaptive methods
  - nonlocal transform-based methods



- Applications:
  - denoising
  - deblurring
  - deblocking
  - super-resolution/zooming

Outline

- **Signal-dependent noise modeling and removal for digital imaging sensors**
- **Local polynomial approximations (LPA-ICI)**
- **Advanced image processing techniques:**

  - shape-adaptive methods

  - nonlocal transform-based methods

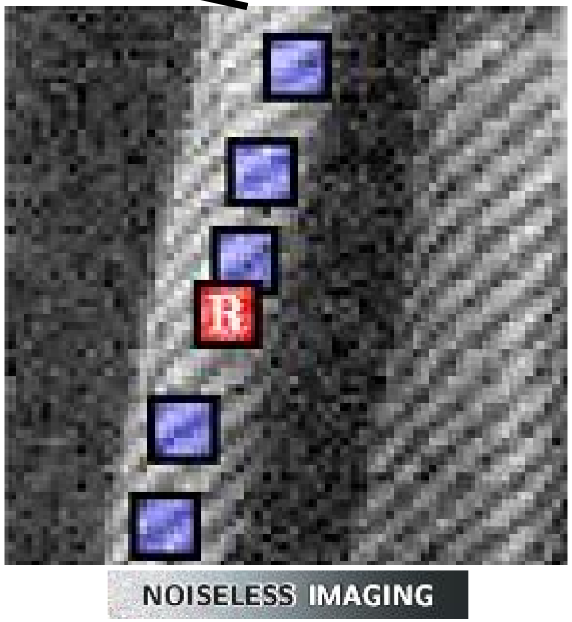
- **Applications:**

  - denoising

  - deblurring

  - deblocking

  - super-resolution/zooming



Outline

- **Signal-dependent noise modeling and removal for digital imaging sensors**
- **Local polynomial approximations equipped with ICI rule (LPA-ICI)**
- **Advanced image processing techniques:**

  - shape-adaptive methods

  - nonlocal transform-based methods

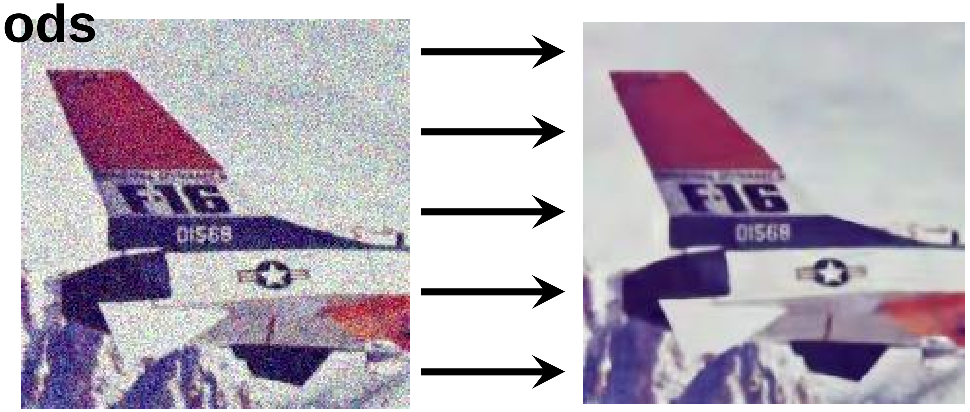
- **Applications:**

  - denoising

  - deblurring

  - deblocking

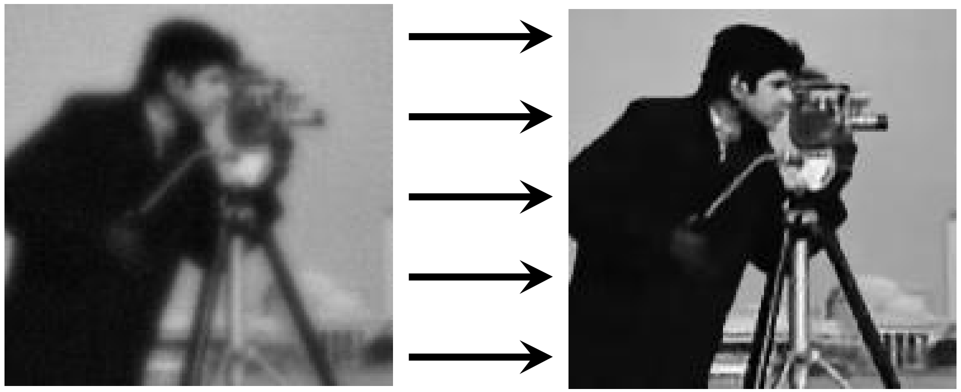
  - super-resolution/zooming



Outline

- **Signal-dependent noise modeling and removal for digital imaging sensors**
- **Local polynomial approximations equipped with ICI rule (LPA-ICI)**
- **Advanced image processing techniques:**
  - shape-adaptive methods
  - nonlocal transform-based methods

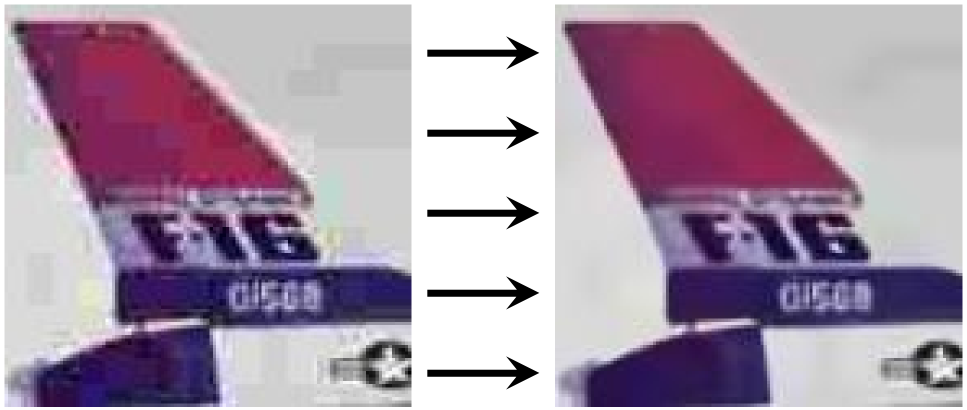
- **Applications:**
  - denoising
  - deblurring
  - deblocking
  - super-resolution/zooming



Outline

- Signal-dependent noise modeling and removal for digital imaging sensors
- Local polynomial approximations (LPA-ICI)
- Advanced image processing techniques:
  - shape-adaptive methods
  - nonlocal transform-based methods

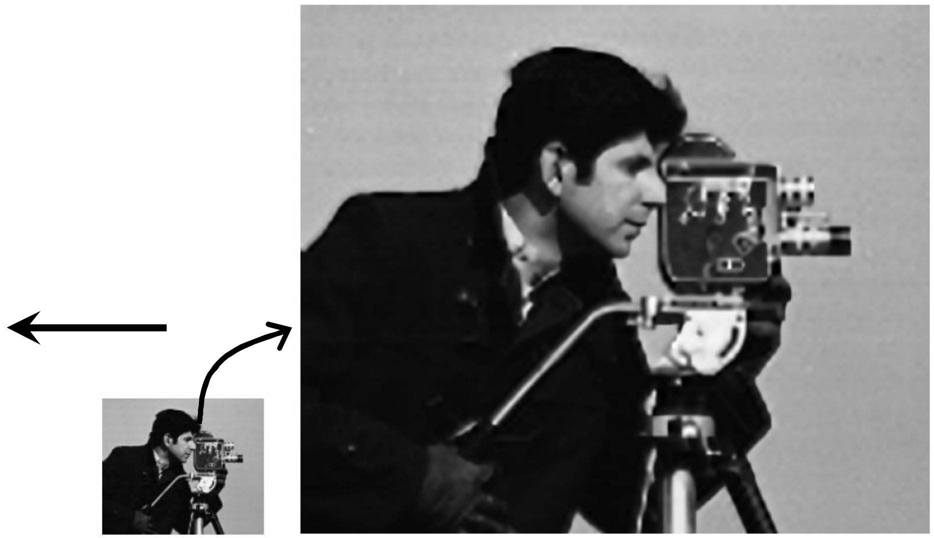
- Applications:
  - denoising
  - deblurring
  - deblocking
  - super-resolution/zooming



Outline

- Signal-dependent noise modeling and removal for digital imaging sensors
- Local polynomial approximations (LPA-ICI)
- Advanced image processing techniques:
  - shape-adaptive methods
  - nonlocal transform-based methods

- Applications:
  - denoising
  - deblurring
  - deblocking
  - super-resolution/zooming



Intro

- Load an image and corrupt with additive white Gaussian noise (AWGN)

```
y = im2double(imread('cameraman.tif'));   % load noise-free image
sigma = 0.2*(max(y(:))-min(y(:))); % define sigma as 20% of range of y
n = sigma * randn(size(y)); % generate 20% noise
z = y + n;    % add noise to obtain noisy image
figure(1);
imshow([y z]);
```

- $z(x) = y(x) + \sigma n(x), x \in \mathbb{Z}^2$ and $n(\cdot) \sim \mathcal{N}(0,1)$
- The goal in *image denoising* is to estimate y from a single realization of z. The statistics of σn can be either known, or have to be estimated (*noise estimation*).

Intro

- If noise samples are independent, then
$$\mathrm{var}\left\{\sum_{i=1}^{N}\lambda_i z(x_i)\right\} = \sum_{i=1}^{N}\lambda_i^2 \mathrm{var}\{z(x_i)\}$$
- Consider a linear smoothing filter implemented as the convolution of z against a blur kernel g:
$$\hat{y}(x) = (z \circledast g)(x) = \sum_{\xi\in\mathbb{Z}^2} z(x-\xi)g(\xi).$$
Then $\mathrm{var}\{\hat{y}(x)\} = \sum_{\xi\in\mathbb{Z}^2}\mathrm{var}\{z(x-\xi)\}\, g^2(\xi) =$
$$= \sigma^2 \sum_{\xi\in\mathbb{Z}^2} g^2(\xi) = \sigma^2 \|g\|_2^2,$$
where $\|g\|_2$ is the ℓ_2 norm of g.
Blur kernels satisfy $g \geqslant 0$ and $\sum_{\xi\in\mathbb{Z}^2} g(\xi) = 1$, therefore $\mathrm{var}\{\hat{y}(x)\} \leqslant \sigma^2$ (*i.e.* noise attenuation).

Intro

- Any discrete uniform blur kernel g has N non-zero samples all equal to $1/N$.
- Then, convolving z with a uniform blur kernel gives

$$\mathrm{var}\{\hat{y}(x)\} = \sigma^2 \sum_{\xi \in \mathbb{Z}^2} g^2(\xi) = \sigma^2 \sum_1^N N^{-2} = \sigma^2/N.$$

This means that the bigger is the kernel, the stronger is the noise attenuation.

Intro

- Multiple noise realizations

```
N_realizations = 300;
Z=zeros(size(y,1),size(y,2),N_realizations);
for jj=1:N_realizations
    n = sigma * randn(size(y)); % generate noise
    z = y + n;     % add noise
    Z(:,:,jj)=z;   % collect in a 3D stack
end
```

- **In real applications, we typically deal with only a single noise realization.** Here we consider many realizations as an easy way to obtain population statistics from sample estimators.

- Pointwise variance (computed along the 3^{rd} dimension)

```
>> var_of_Z=mean((Z-repmat(mean(Z,3),[1 1 size(Z,3)])).^2,3);
```

Intro

- Smoothing with uniform kernel of size $h \times h$ and inspection of variance, bias, and MSE of the estimate $\hat{y}$:

```
for h=1:9;
    clear Yhat;
    g_h=ones(h)/h^2;
    for jj=1:N_realizations
        Yhat(:,:,jj)=conv2(Z(:,:,jj),g_h,'same');
    end
    var_of_Yhat=mean((Yhat-repmat(mean(Yhat,3),[1 1 size(Yhat,3)])).^2,3);
    bias_of_Yhat=mean(Yhat-repmat(y,[1 1 N_realizations]),3);
    MSE_of_Yhat{h}=bias_of_Yhat.^2+var_of_Yhat;
end
```

Note that `var_of_Yhat` is equal to `sigma^2*sum(g_h(:).^2)`

Intro

Poisson distributions

Poisson distributions are discrete integer-valued distributions with non-negative real-valued parameter (mean) $\theta \geq 0$

$$z \sim \mathcal{P}(\theta) \qquad \Pr[z = \zeta|\theta] = e^{-\theta}\frac{\theta^{\zeta}}{\zeta!}, \ \zeta \in \mathbb{N}.$$

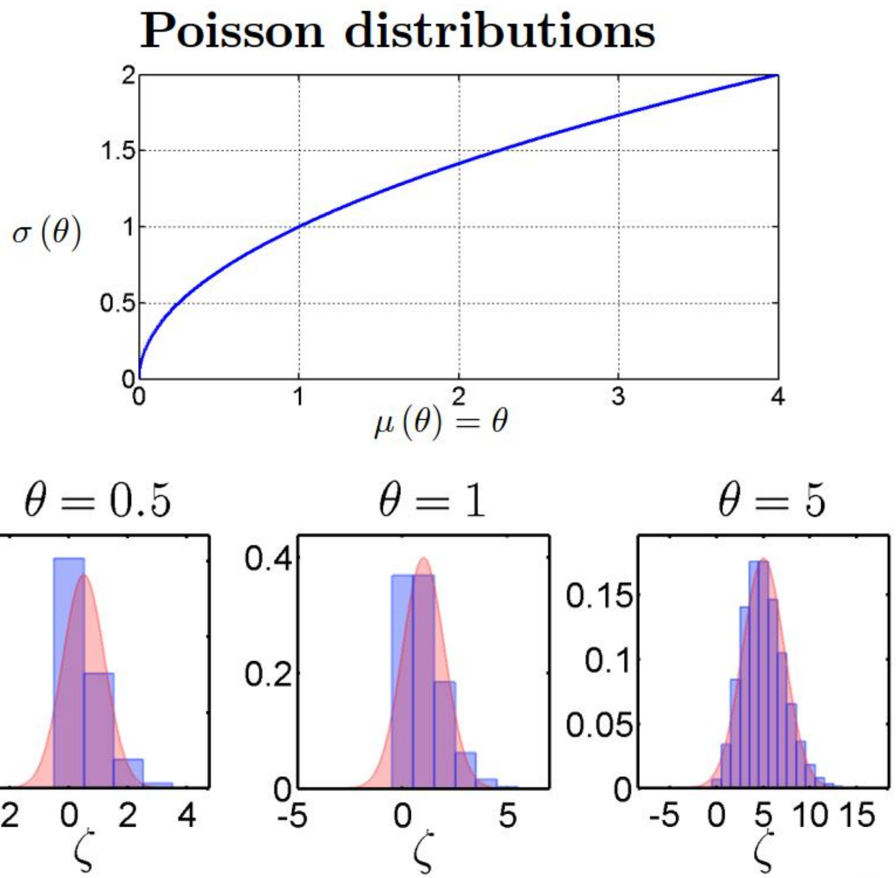
$$\begin{aligned} \mu(\theta) &= E\{z|\theta\} = \theta \\ \sigma^2(\theta) &= \operatorname{var}\{z|\theta\} = \theta = \mu(\theta) \end{aligned}$$

mean and variance coincide and are equal to the parameter θ

Matlab code: z = `poissrnd(theta)`

signal-to-noise ratio (SNR): $\dfrac{\mu(\theta)}{\sigma(\theta)} = \sqrt{\theta} \underset{\theta \to 0}{\longrightarrow} 0 \qquad \dfrac{\mu(\theta)}{\sigma(\theta)} \underset{\theta \to +\infty}{\longrightarrow} +\infty$

Intro



Discrete Poisson $\mathcal{P}(\theta)$ (blue) and continuous normal approximation $\mathcal{N}(\theta, \theta)$ (red)

Intro

Normal approximation of Poisson

$z \sim \mathcal{P}(\theta)$ means the probability of z $\Pr[z = \zeta|\theta] = e^{-\theta}\frac{\theta^{\zeta}}{\zeta!}$, $\zeta \in \mathbb{N}$

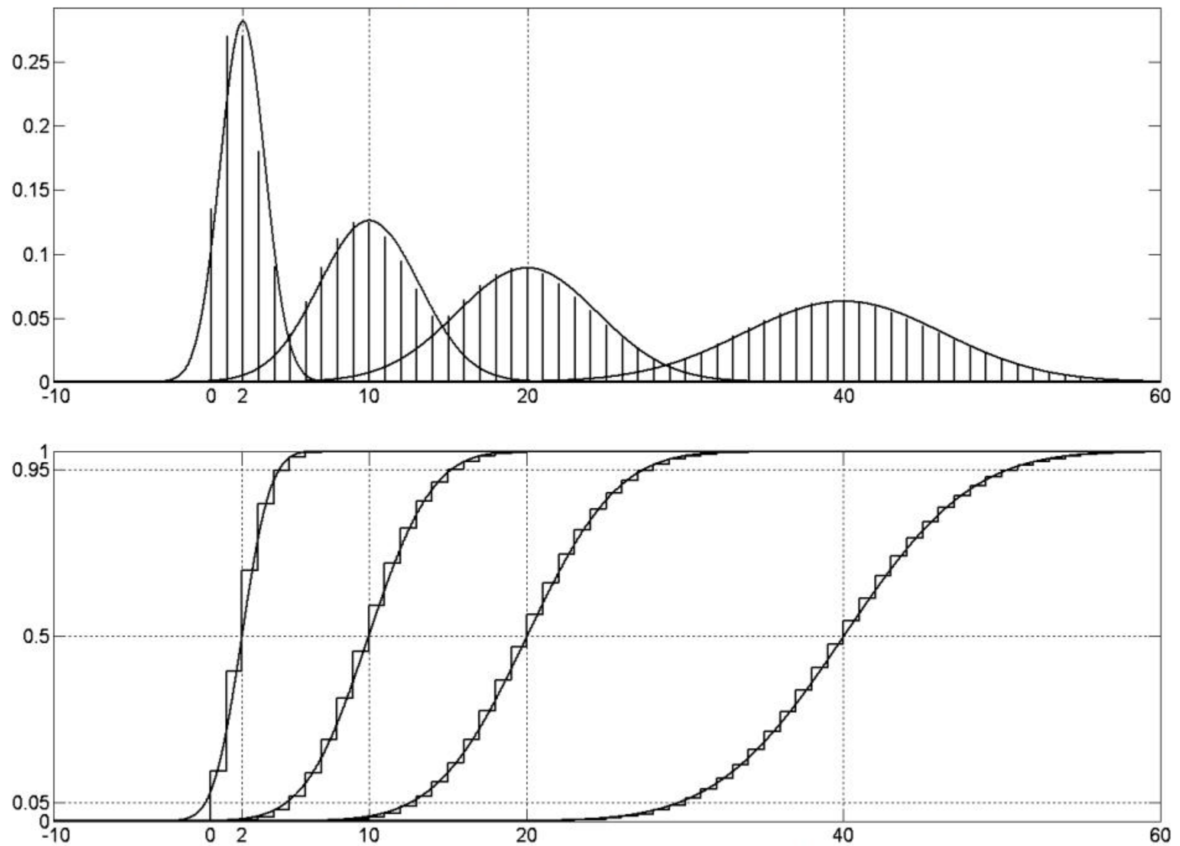
$z \sim \mathcal{N}(\mu, \sigma^2)$ means the probability density of z is $\wp(\zeta|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\zeta-\mu)^2}{2\sigma^2}}$,

$$\mathcal{P}(\theta) \underset{\theta \to +\infty}{\longrightarrow} \mathcal{N}(\theta, \theta)$$

Matlab code: `z = z + sqrt(theta).*randn(size(theta))`

Intro

Normal approximation of Poisson



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

Intro

Poissonian noise

Let $y : X \to Y \subseteq \mathbb{R}^+$ original image (deterministic, possibly unknown)

$\chi > 0$ scaling factor

$$\boxed{z(x)\,\chi \sim \mathcal{P}(\chi y(x)), \qquad \forall x \in X.}$$

$$E\{z(x)\,\chi\} = \chi E\{z(x)\} = \chi y(x) \quad \Longrightarrow \quad E\{z(x)\} = y(x),$$

$$\operatorname{var}\{z(x)\,\chi\} = \chi^2 \operatorname{var}\{z(x)\} = \chi y(x) \quad \Longrightarrow \quad \operatorname{var}\{z(x)\} = \frac{y(x)}{\chi}.$$

This can be rewritten in the usual form as

$$\boxed{z(x) = y(x) + \sqrt{\frac{y(x)}{\chi}}\xi(x), \qquad \forall x \in X,}$$

where $E\{\xi(x)\} = 0$ and $\operatorname{var}\{\xi(x)\} = 1$.

The term $\sqrt{\frac{y(x)}{\chi}}\xi(x)$ is the so-called **Poissonian noise**.

Intro

Scaled Poisson observations

$\chi = 1000$

$\chi = 300$

$\chi = 100$

Intro

Scaled Poisson observations

$\chi = 50$

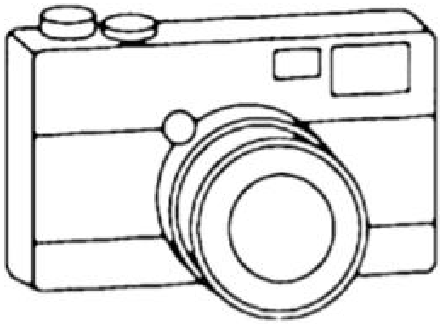
$\chi = 10$

$\chi = 1$

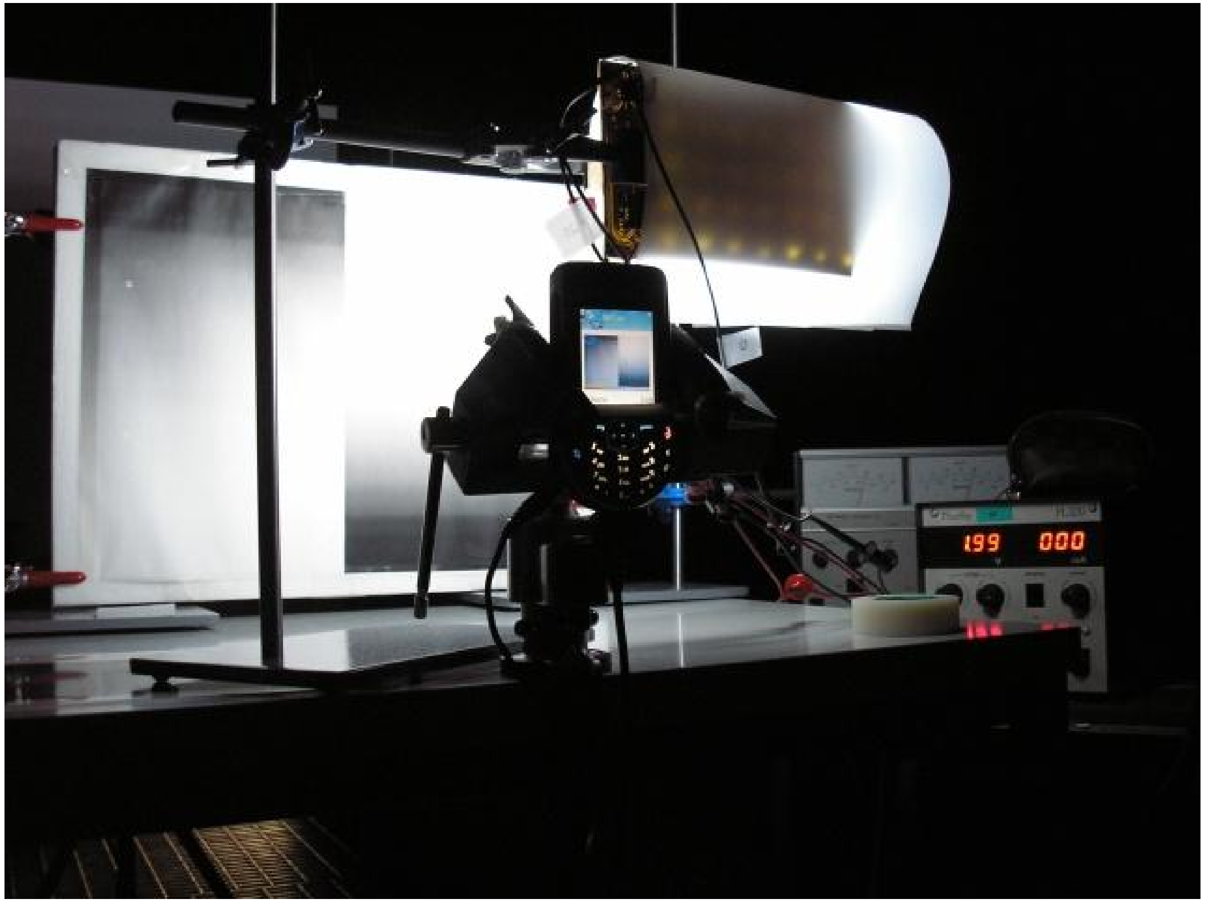
Statistical analysis of raw data

A simple experiment

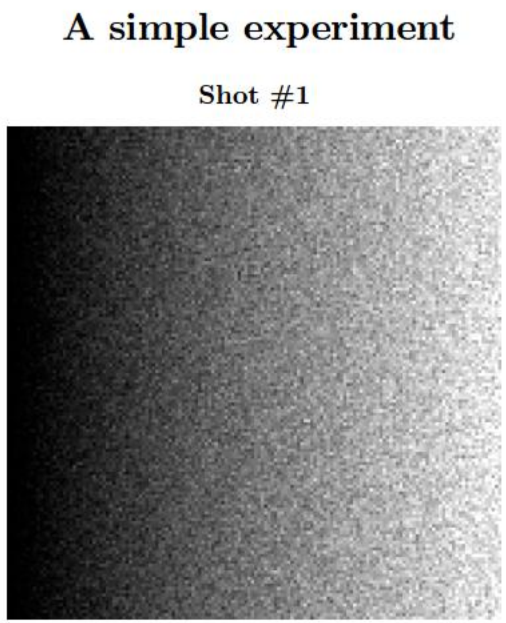
Take photos of a gray scale test ramp

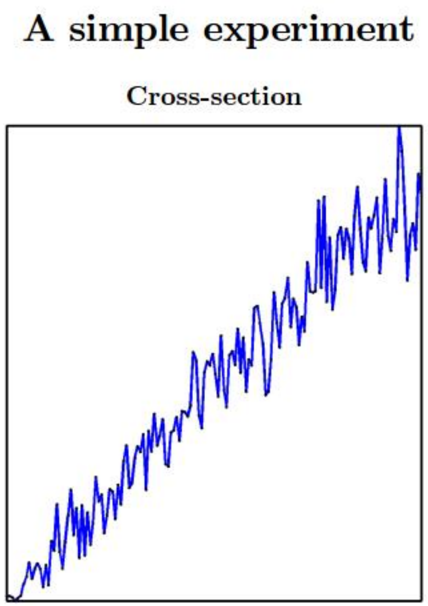


Statistical analysis of raw data



Statistical analysis of raw data

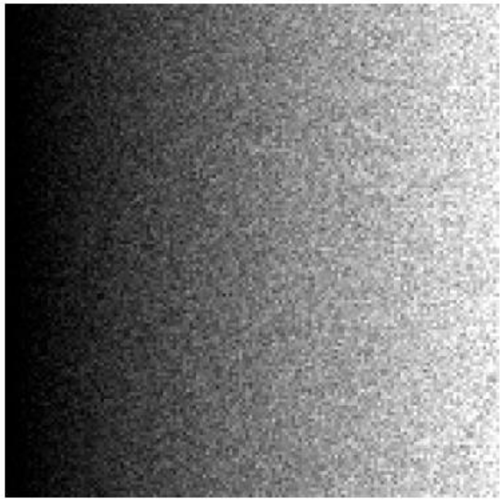




Statistical analysis of raw data

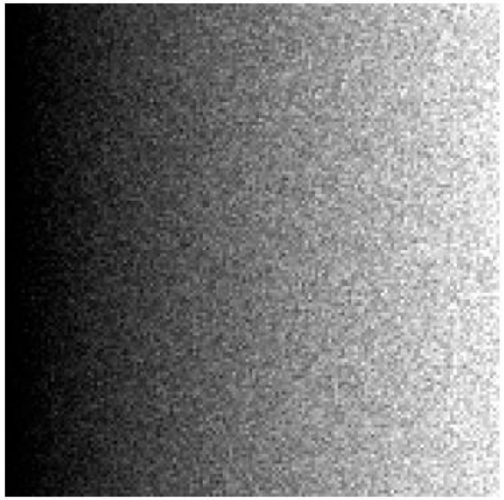
A simple experiment

Shot #2



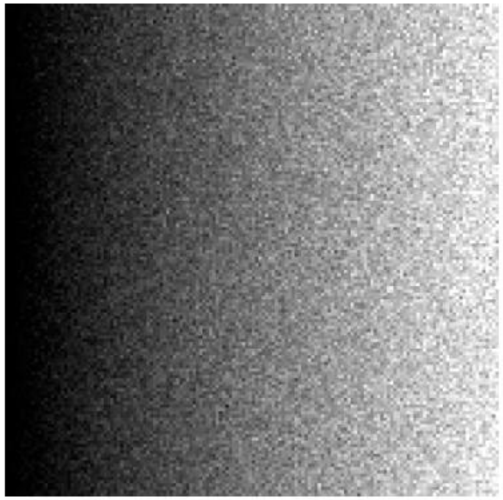
A simple experiment

Shot #3



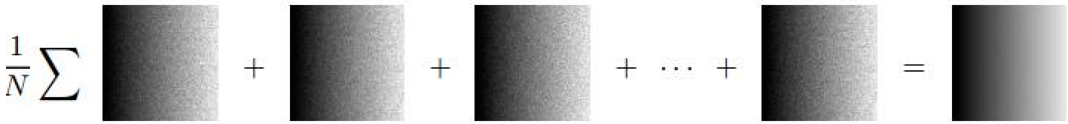
A simple experiment

Shot #4

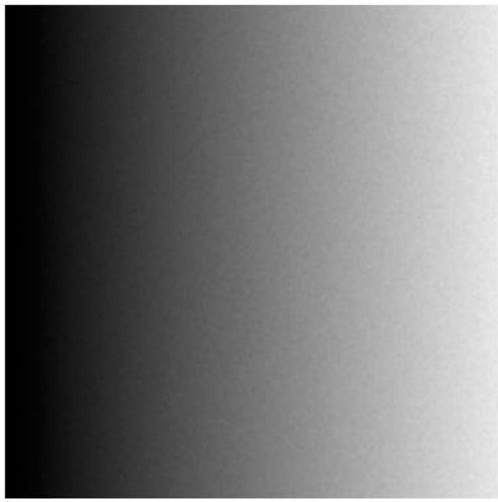


Statistical analysis of raw data

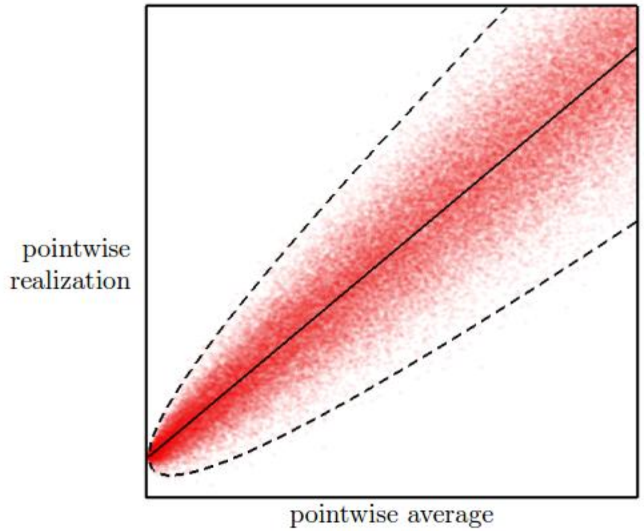
TAKE MANY MORE SHOTS, AND THEN AVERAGE THEM ALL



TAKE MANY MORE SHOTS, AND THEN AVERAGE THEM ALL

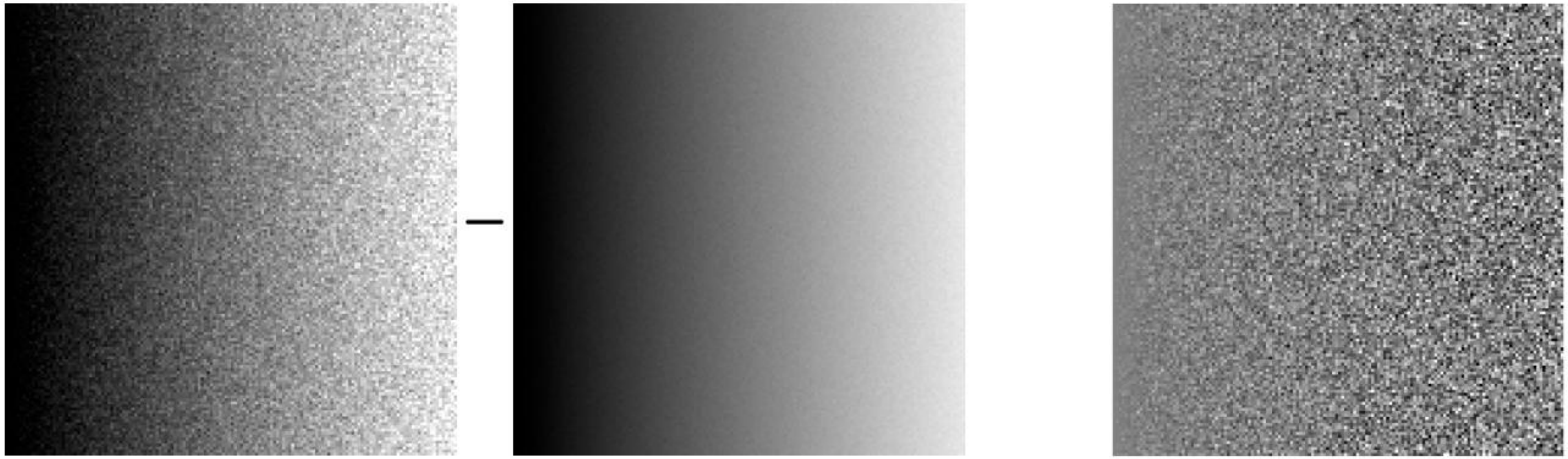


Scatterplot: average vs realization



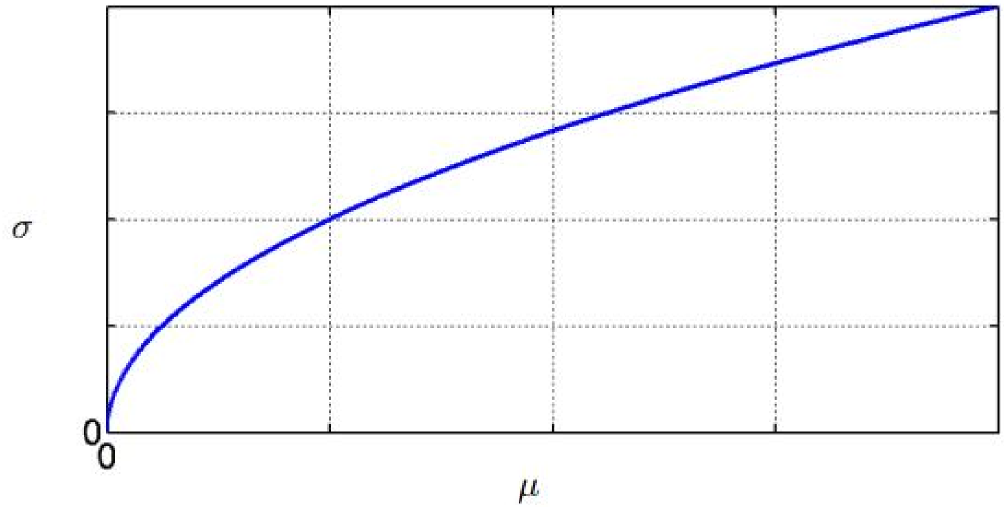
Statistical analysis of raw data

SUBTRACT THE *AVERAGE OF ALL SHOTS* FROM *ANY OF THE SHOTS*



Statistical analysis of raw data

FOR EACH PIXEL, COMPUTE
SAMPLE MEAN AND *SAMPLE STANDARD DEVIATION*
W.R.T. THE VARIOUS SHOTS



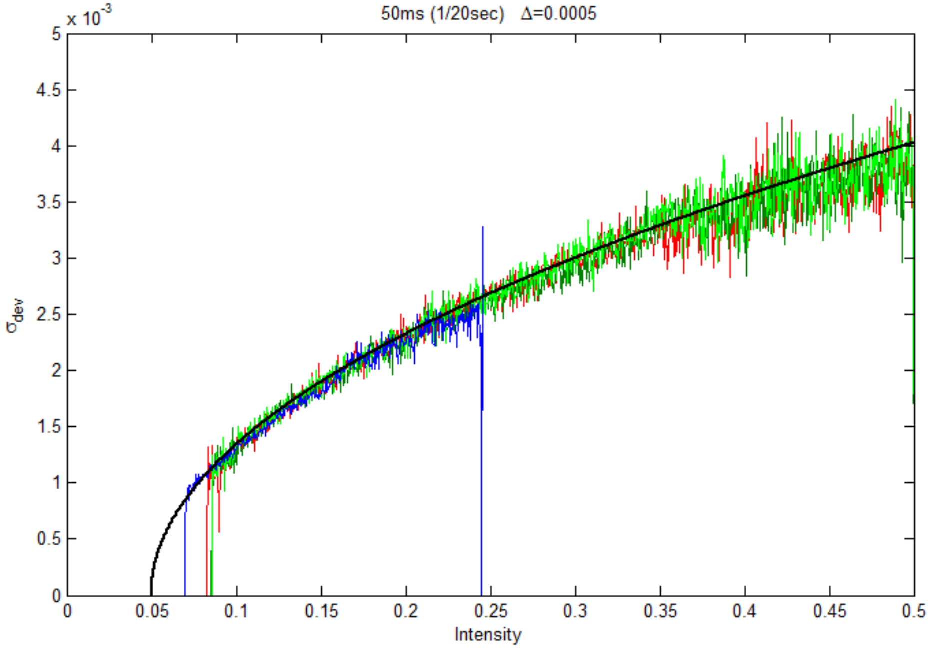
NOISE IS STRONGER WHERE THE AVERAGE IMAGE IS BRIGHTER:
STANDARD-DEVIATION IS A FUNCTION OF MEAN

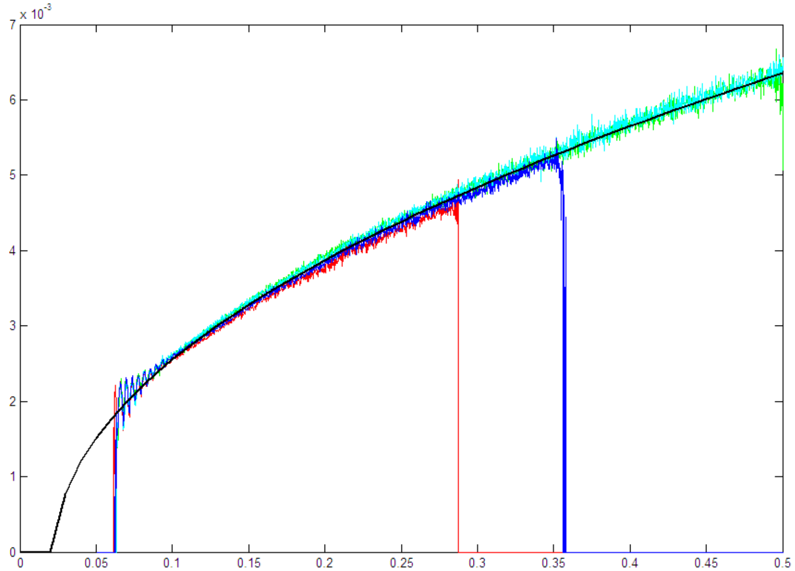
SIGNAL-DEPENDENT NOISE

Statistical analysis of raw data

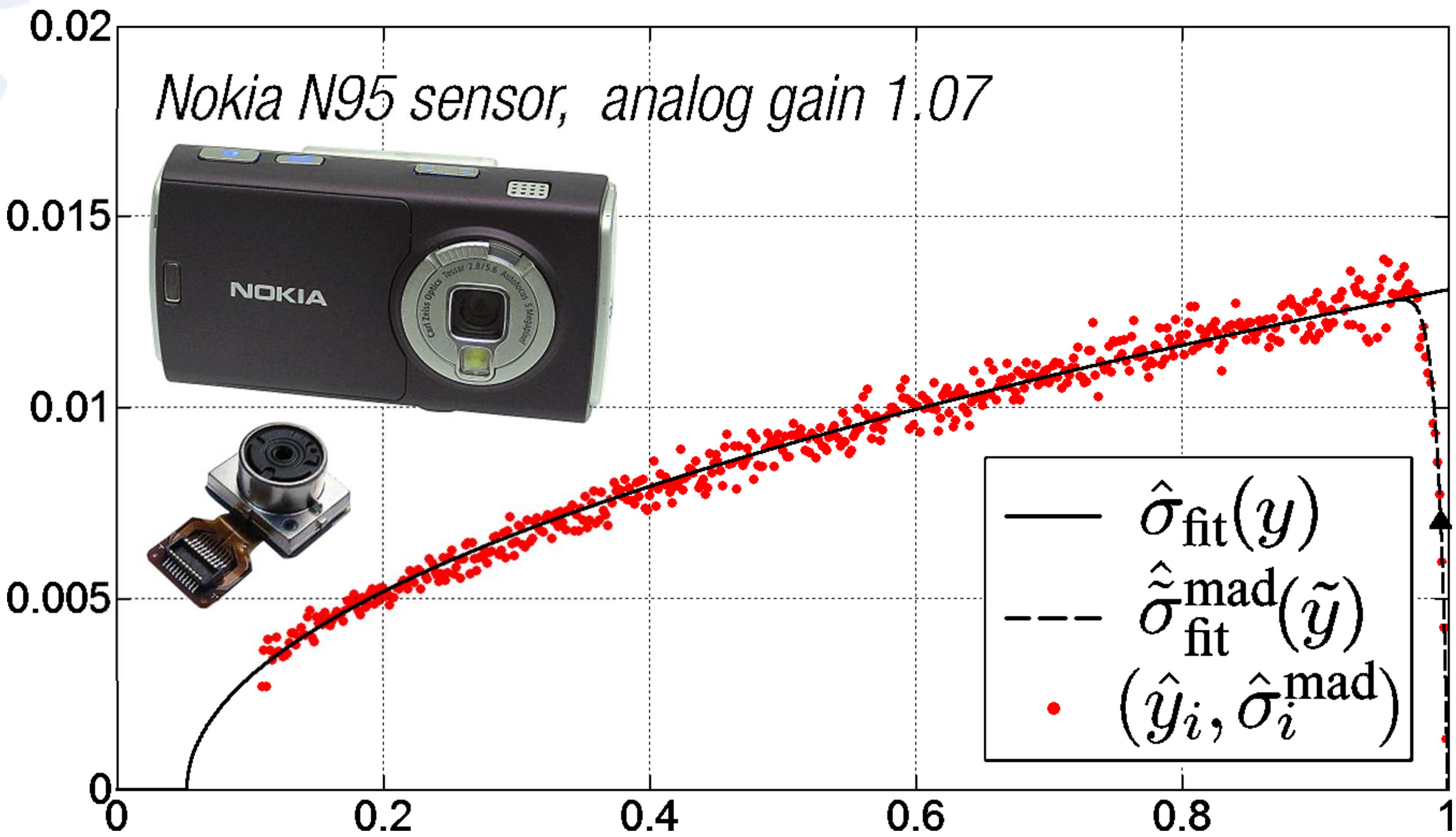
The analysis of experimental data demonstrates that:

1. The model of noise is close to the Poissonian one
2. Model parameters depend neither on the color channel nor on the exposure time

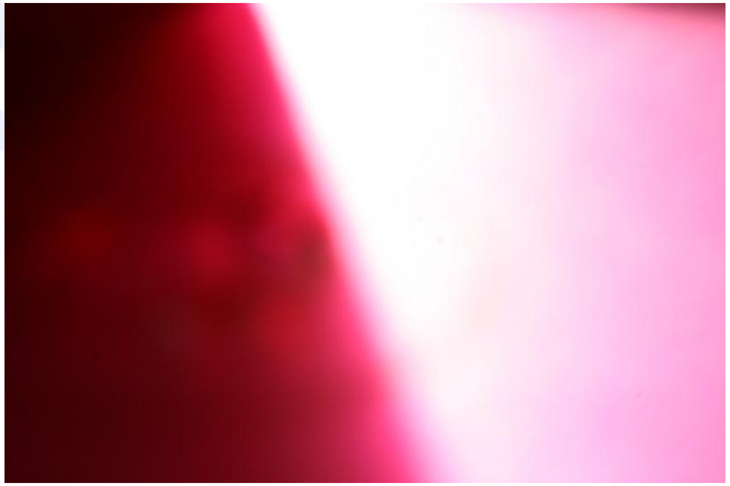


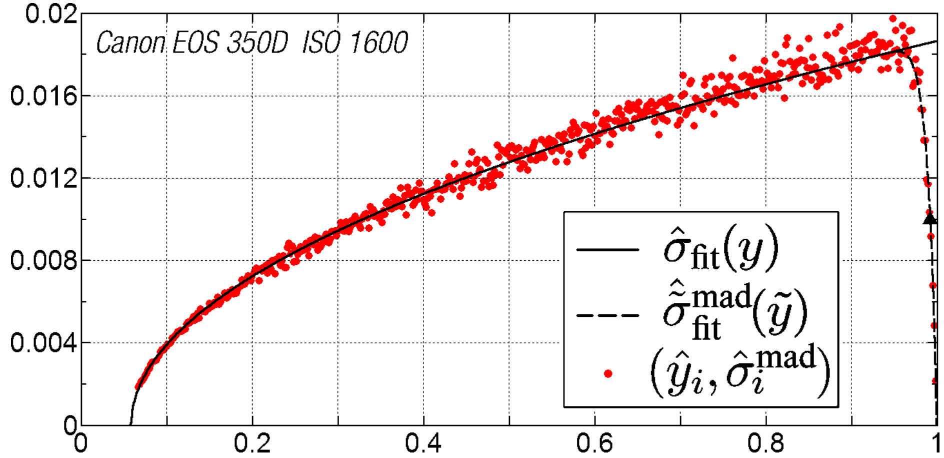


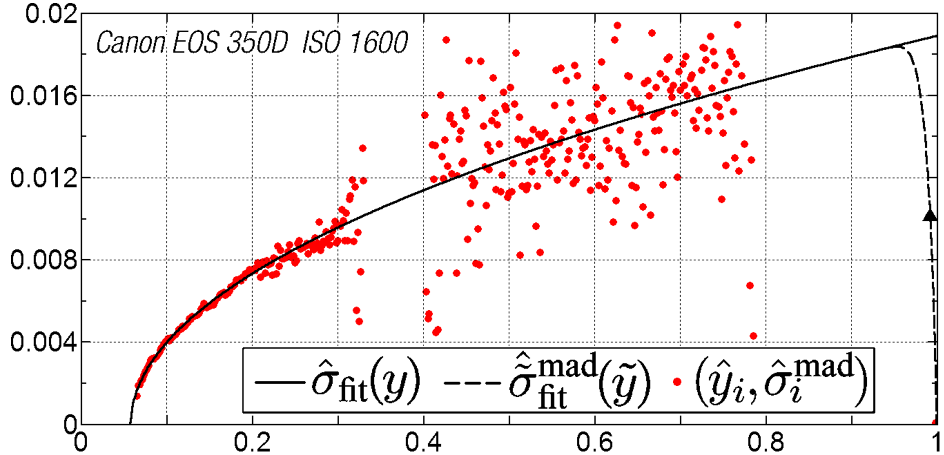
Parametric signal-dependent noise-modelling: Poissonian-Gaussian with clipping



Parametric signal-dependent noise-modelling: automatic estimation from single-image raw-data (*http://www.cs.tut.fi/~foi/sensornoise.html*)







Practical modeling for raw data: idea

- Model photon-to-electron conversion using Poisson distributions (signal dependent);
- Model the other noise sources as signal-independent and Gaussian (central-limit theorem);
- Exploit normal approximation of Poisson distributions;
- The acquisition/dynamic range is limited: too dark or too bright signals are clipped;
- There can be a pedestal;
- Spatial dependencies can be ignored for normal operating conditions (go for independent noise).

Eventually, only two parameters are sufficient to describe the noise model where the raw data is described as clipped signal-dependent observations.

Variance stabilization

Variance-stabilization problem

Find a function $f : Z \rightarrow \mathbb{R}$ such that the transformed variable $f(z)$ has constant standard deviation, say, equal to 1, $\mathrm{std}\{f(z)|\theta\} = 1$.

such f is a **variance-stabilizing transformation (VST)**

f should be independent of θ

Benefits:

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

Variance stabilization

VSTs are often exploited for the removal of signal-dependent noise through the following three-step procedure:

1. Noise variance is stabilized by applying a VST f to the data; this produces a signal in which the noise can be treated as additive with unitary variance.
2. Noise is removed using a conventional denoising algorithm – denoted by Φ – for additive homoskedastic noise (e.g., additive white Gaussian noise).
3. An inverse transformation is applied to the denoised signal, obtaining the estimate of the signal of interest.

Denoising algorithms attempt to estimate the expectation, thus, $D = \Phi\left(f\left(z\right)\right)$ can be treated as an approximation of $E\{f(z)|\theta\}$.

Inversion for Poisson stabilized by Anscombe Mäkitalo, Foi (TIP, 2011)

Let z be Poisson distributed data.
Applying the Anscombe transform yields $f(z) = 2\sqrt{z + \frac{3}{8}}$.
After filtering of $f(z)$ we obtain $D = \Phi(f(z))$, which we treat as an approximation of $E\{f(z)|\theta\}$.

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

Asymptotically unbiased inverse: $\mathcal{I}_B(D) = \left(\frac{D}{2}\right)^2 - \frac{1}{8}$. Typically used in applications.

Exact unbiased inverse: $\mathcal{I}_C : E\{f(z) \mid y\} \longmapsto E\{z \mid y\}$.

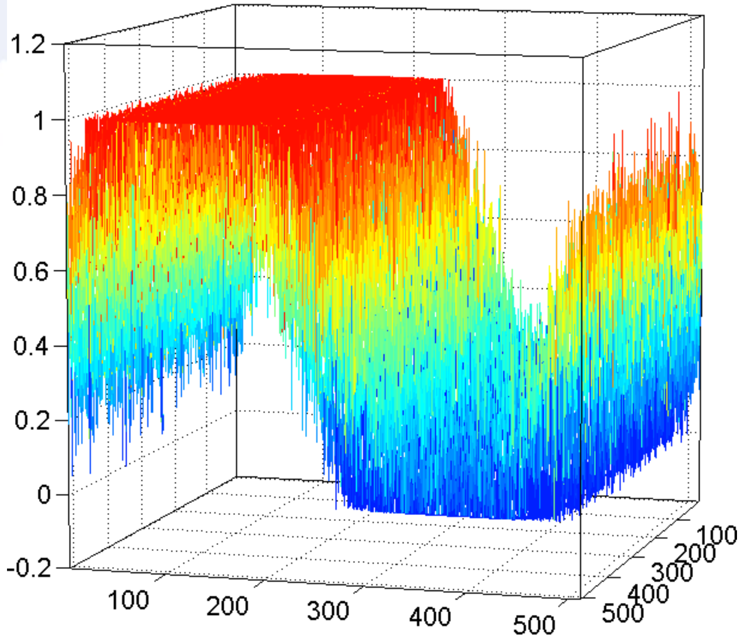
We have discrete Poisson probabilities $P(z \mid y)$, so

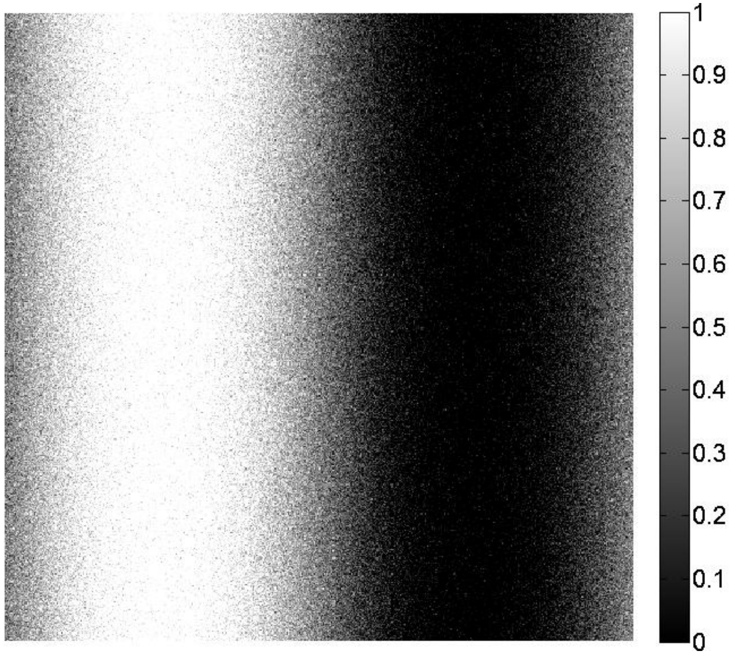
$$E\{f(z) \mid y\} = \sum_{z=0}^{+\infty} f(z) P(z \mid y) = 2 \sum_{z=0}^{+\infty} \left(\sqrt{z + \frac{3}{8}} \cdot \frac{y^z e^{-y}}{z!} \right).$$

The definition of $\mathcal{I}_C$ is implicit, but we can have a closed form approximation as

$$\mathcal{I}_C(D) \cong \frac{1}{4}D^2 + \frac{1}{4}\sqrt{\frac{3}{2}}D^{-1} - \frac{11}{8}D^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}D^{-3} - \frac{1}{8}$$

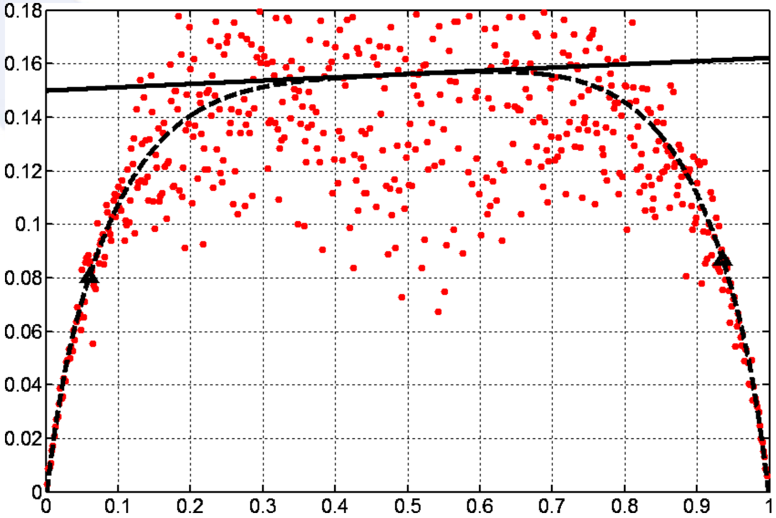
Experiment: clipped noisy data

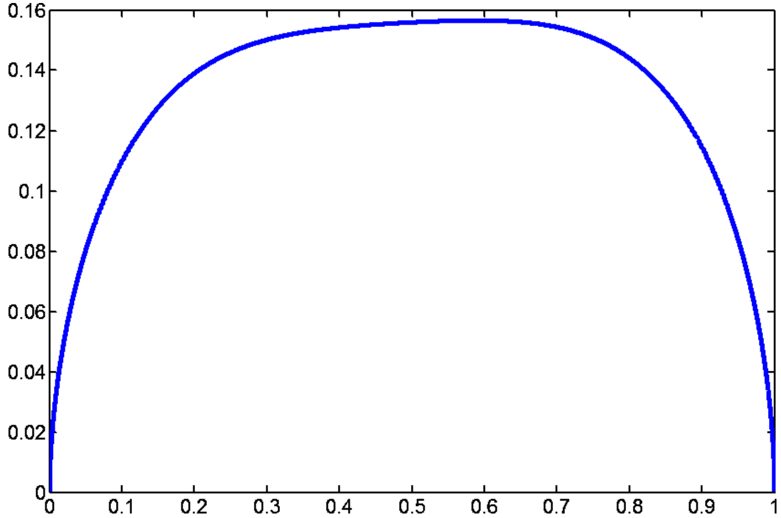




Original image : $y(x1, x2) = 0.7 \sin(2\pi\, x1/512) + 0.5$

Experiment: Noise Estimation

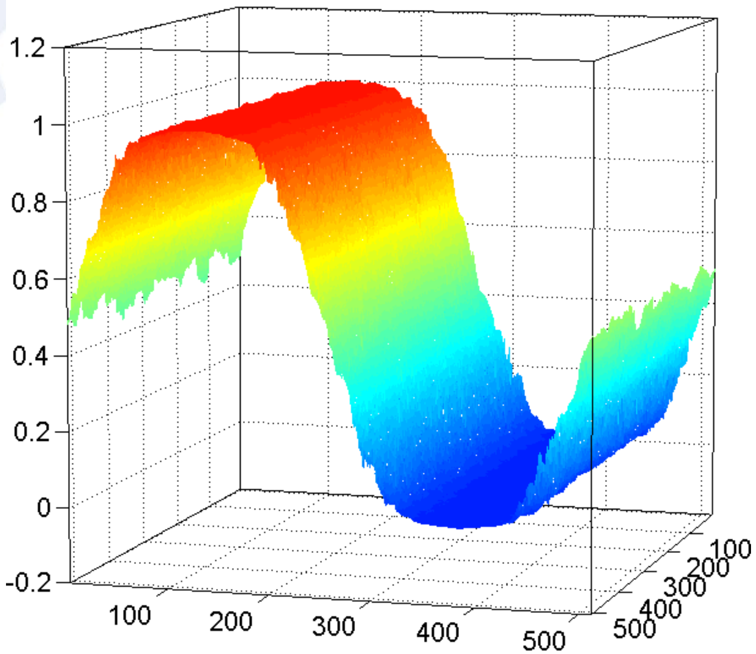


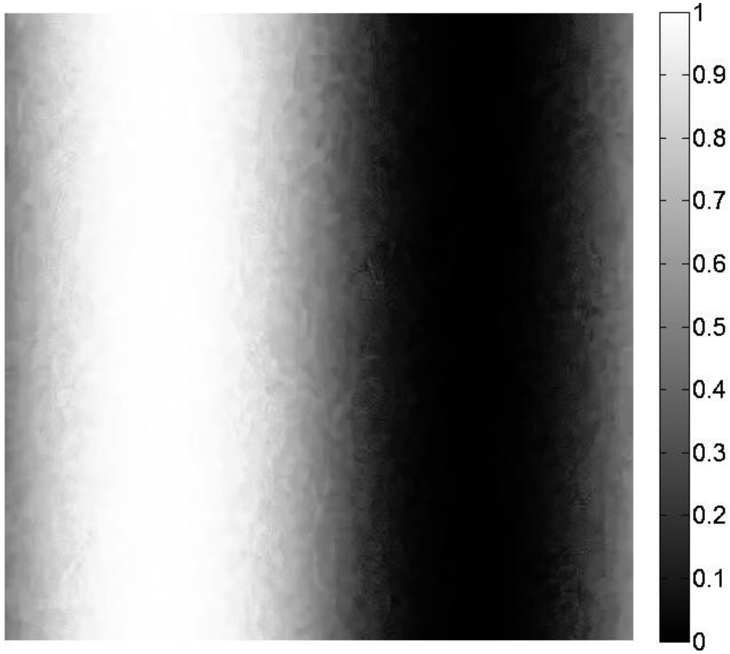


estimation and fitting a = 0.0038, b = 0.022

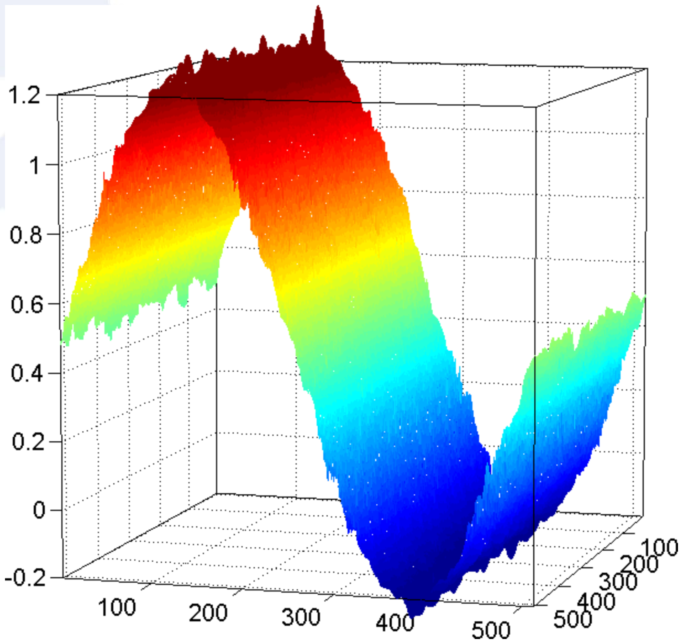
st.dev.-function .σ

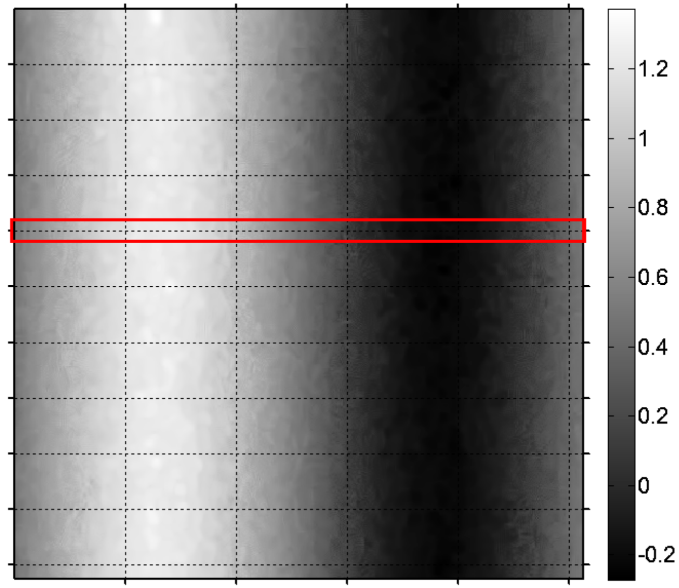
Experiment: denoised estimate after variance stabilization before declipping



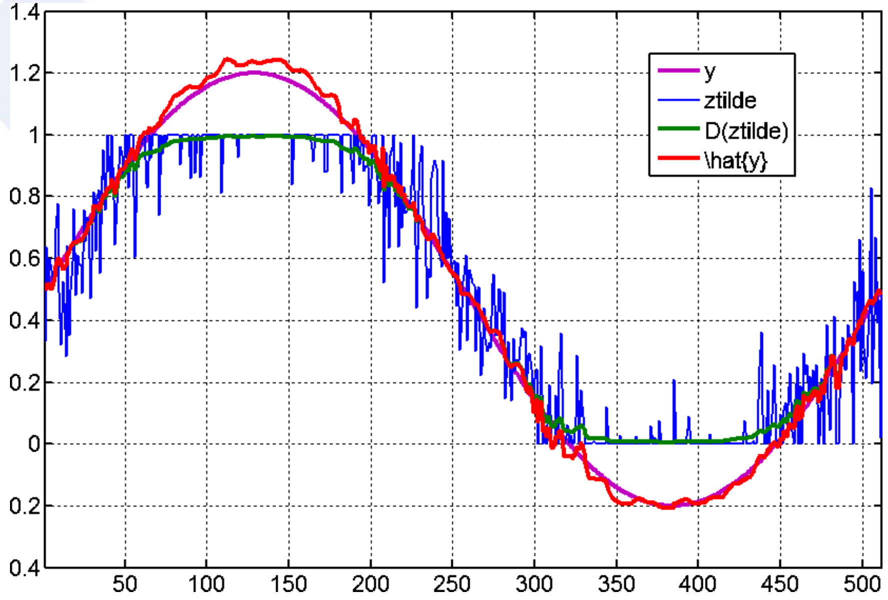


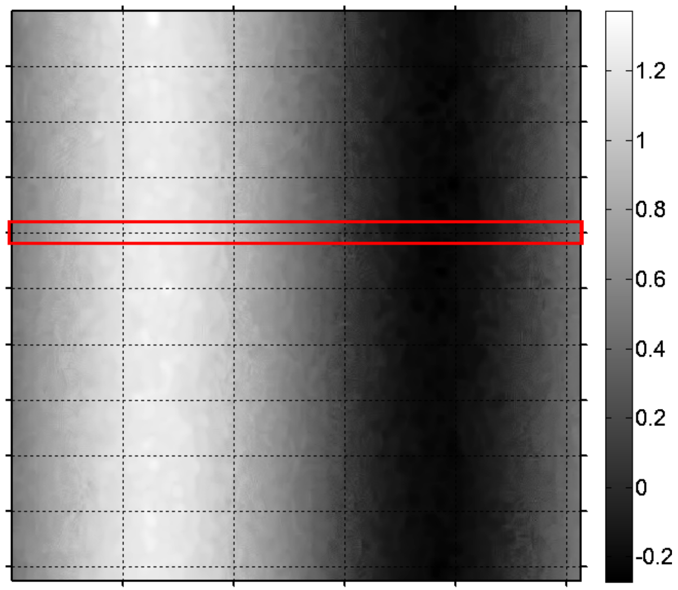
Experiment: declipped estimate





Experiment: declipped estimate (crosssection)



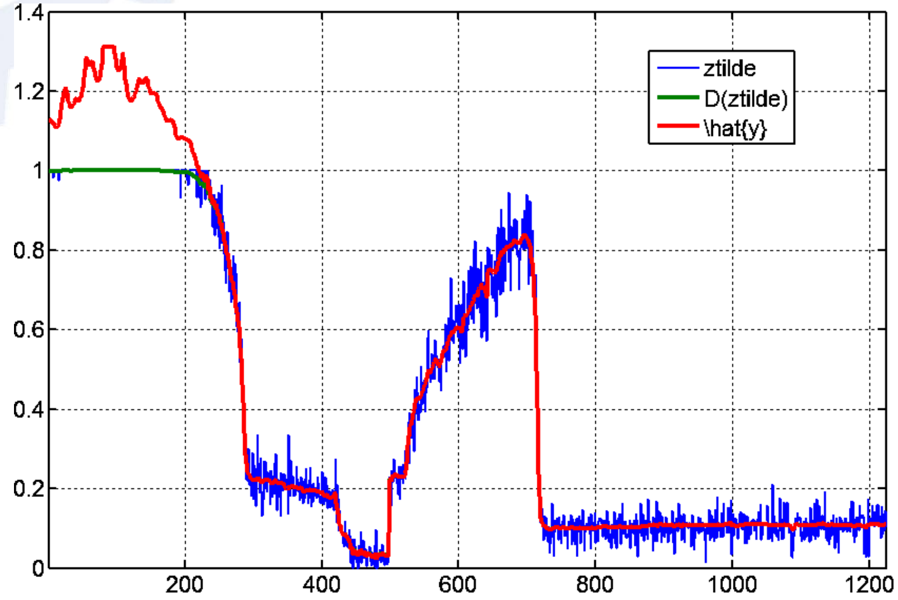


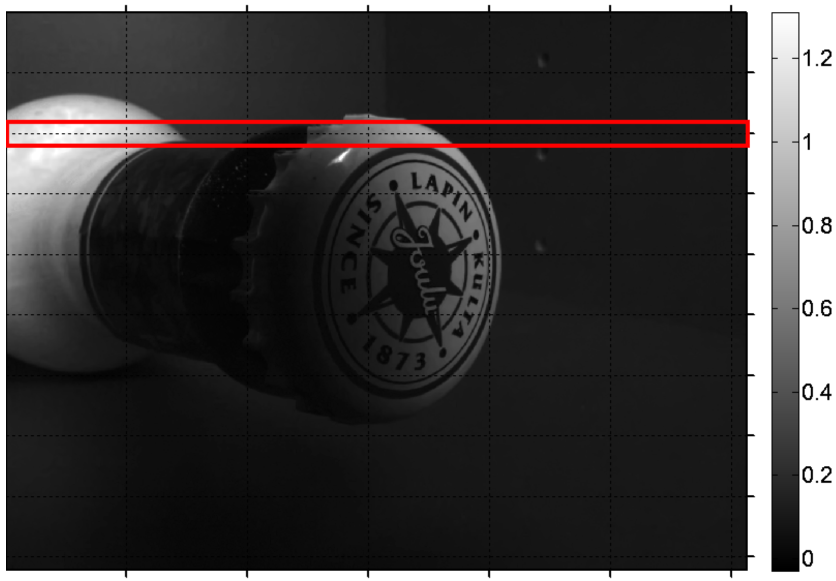
Real experiment: (Raw-data from Fujiflm FinePix S9600, ISO 1600)

Real experiment: Denoising before declipping

Real experiment: Denoising after declipping

Real experiment: Denoising after declipping (crossection)





LASIP
www.cs.tut.fi/~lasip/

- ***Local Approximation Signal and Image Processing (LASIP) Project***

LASIP project is dedicated to investigations in a wide class of novel efficient adaptive signal processing techniques.

LASIP

LPA estimates, bias and variance, and asymptotic *MSE*

The observation model is $z = y + \eta$, where y is the true signal and η is noise. Let $\hat{y}_h$ denote the LPA estimate and the LPA kernel corresponding to different values of a scale parameter h:

$$\hat{y}_h = z \circledast g_h \qquad \text{where} \quad g_h = g\left(\cdot/h\right)$$

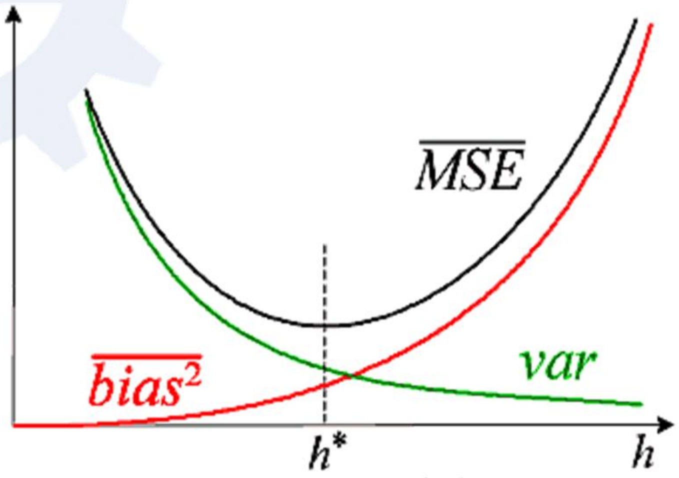
Bias: $b_{\hat{y}_h(x)} = y\left(x\right) - \left(y \circledast g_h\right)\left(x\right)$ (η zero-mean and independent)

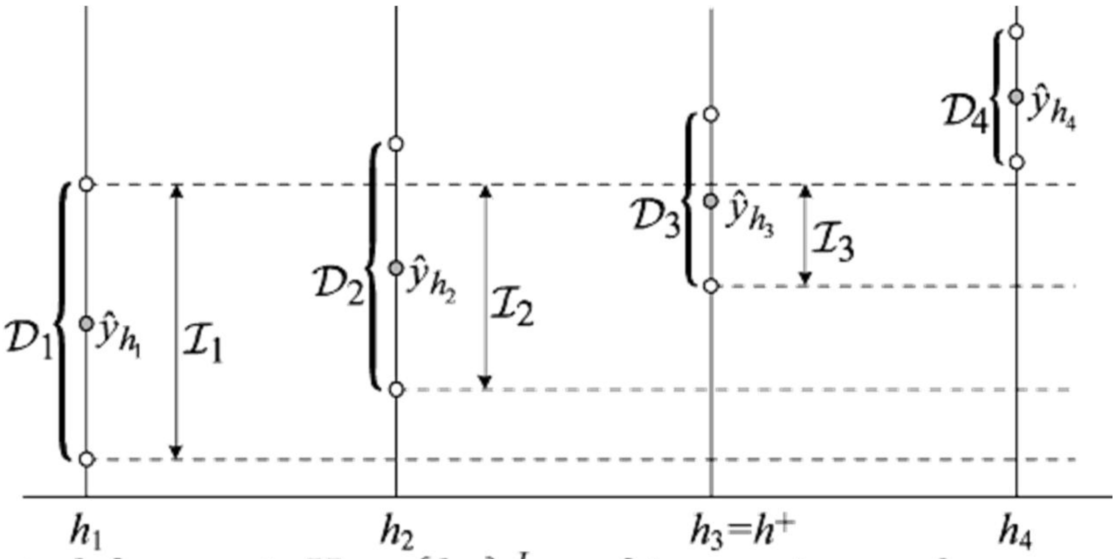
Variance: $\sigma^2_{\hat{y}_h(x)} = \left(\sigma_z^2 \circledast g_h^2\right)\left(x\right)$ (if $\sigma_z^2 \equiv \sigma^2$ then $\sigma^2_{\hat{y}_h(x)} \equiv \sigma_z^2 \left\|g_h\right\|_2^2$)

The following asymptotic expressions for the bias, variance and MSE of hold:

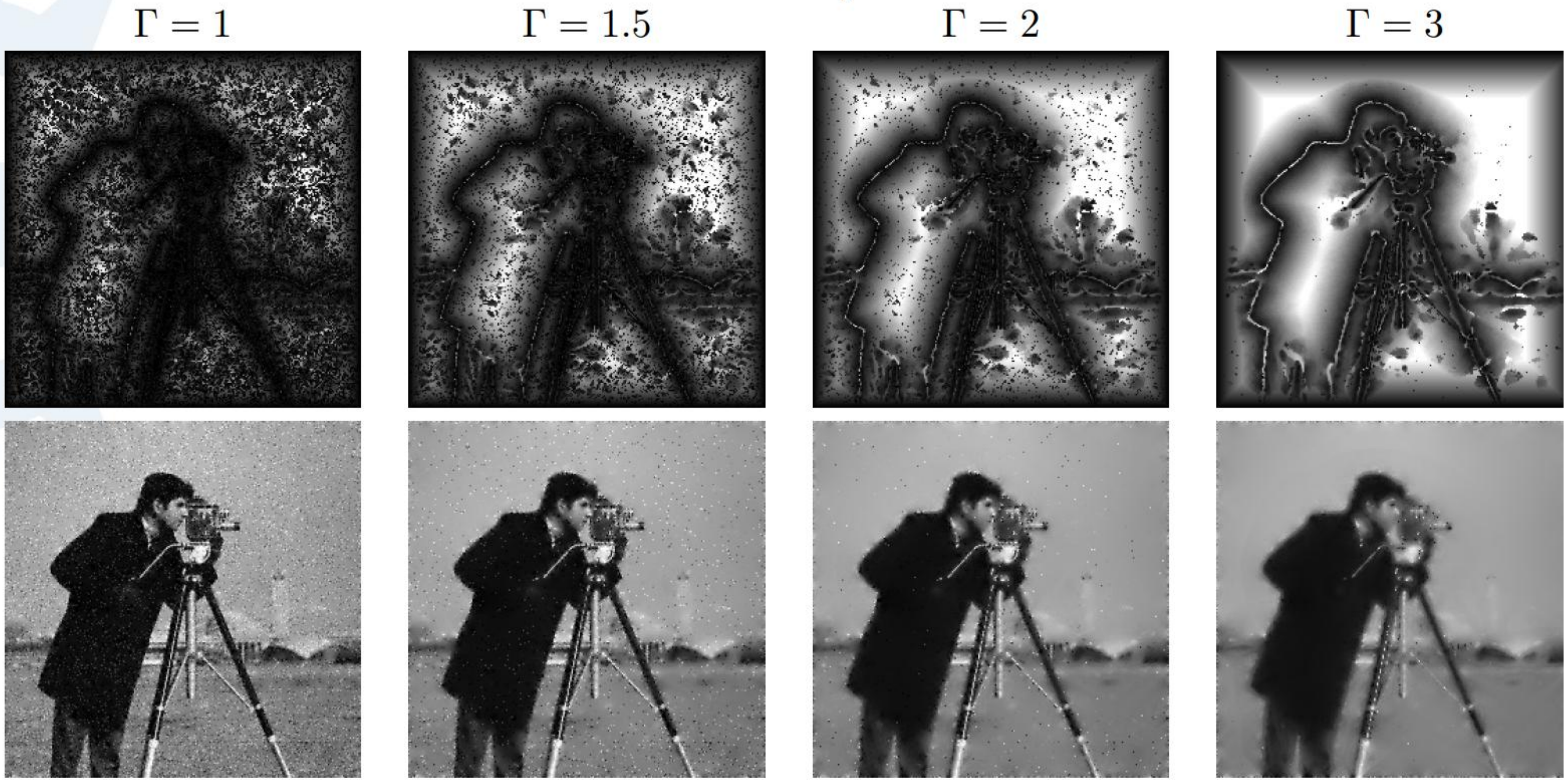
$$b_{\hat{y}_h} = ch^a, \qquad \sigma^2_{\hat{y}_h} = dh^{-b}, \qquad l_{\hat{y}_h(x)} = c^2h^{2a} + dh^{-b}.$$

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



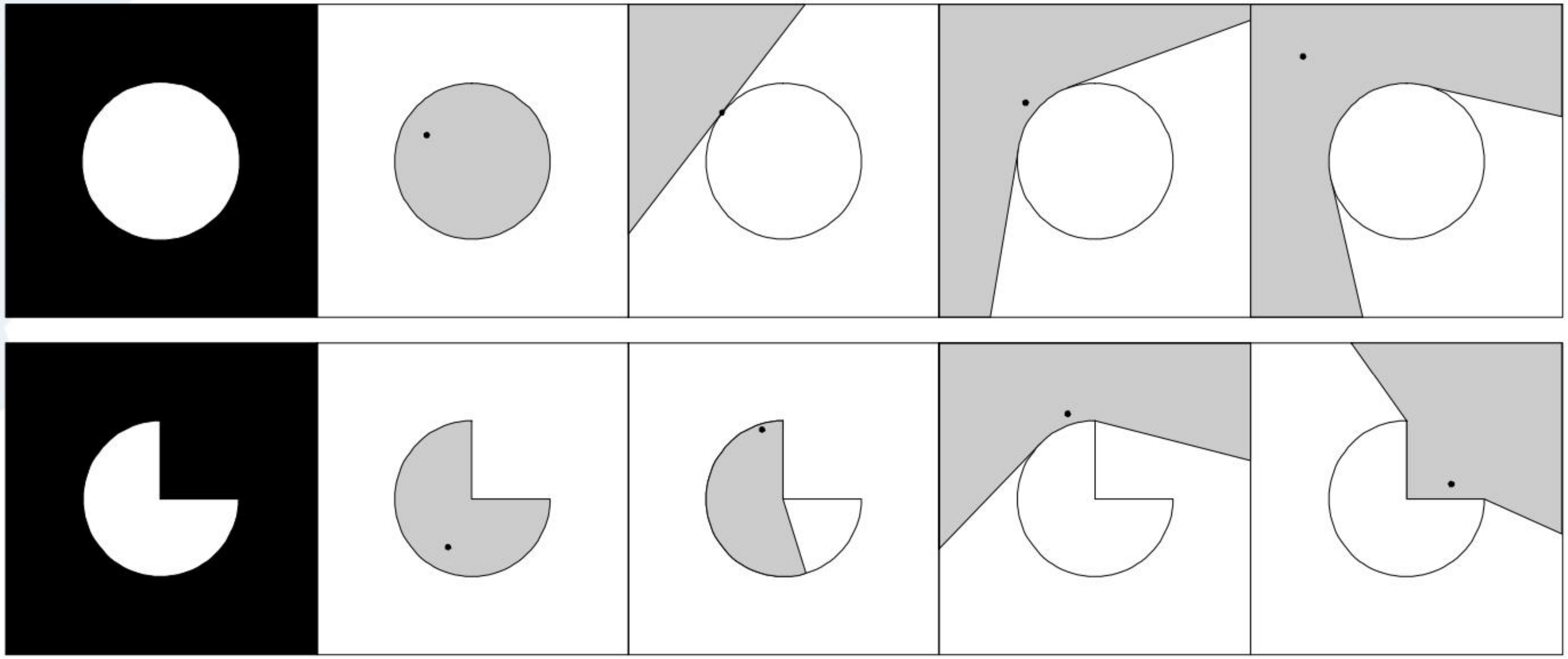


The estimates $\hat{y}_h(x)$ are calculated for a set $H = \{h_j\}_{j=1}^{J}$ of increasing scales. The *ICI* rule yields a pointwise adaptive estimate $\hat{y}_{h^+}(x)$, where for every x an adaptive scale $h^+(x) \in H$ is used; $h^+(x) \approx h^*(x)$. The *ICI* rule is as follows. *Consider the intersection of confidence intervals* $\mathcal{I}_j = \bigcap_{i=1}^{j} \mathcal{D}_i$, *where* $\mathcal{D}_i = [\hat{y}_{h_i}(x) - \Gamma\sigma_{\hat{y}_{h_i}}, \hat{y}_{h_i}(x) + \Gamma\sigma_{\hat{y}_{h_i}}]$ *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* $\hat{y}_{h^+}(x)$.



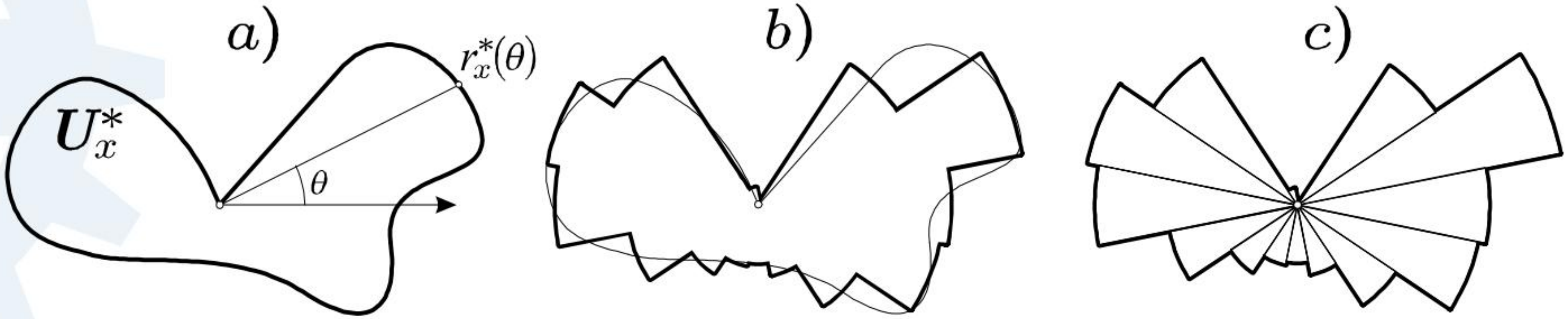
Adaptive scales and adaptive-scale estimates obtained for different values of Γ. The adaptive scales are represented using a darker shade of gray for the smaller scales, black being the smallest scale (which corresponds to a Dirac-delta estimate), and white being the maximum scale (corresponding to a kernel whose support is a disc of radius 35 pixels).

Anisotropy: motivation



In some cases the geometry of symmetric kernels is not sufficient to adapt to the image structure. Goal: adapt to the image using approximations of starshaped supports.

Anisotropic estimator based on directional adaptive-scale: idea



Piecewise constant approximation of $r_x^*(\theta)$ and its representation by adaptive-size sectors.

Directional LPA

The window is characterized by a direction θ and is denoted as w_θ.

The polynomials are expressed with respect to a θ-rotated coordinate system:

$$\mathcal{M} = \left\{ \varphi : \varphi\left(u_1, u_2\right) = \sum_{i,j}^{m} c_{i,j} u_1^i u_2^j \right\},$$

$$(u_1, u_2) = (v_1 \cos\theta + v_2 \sin\theta, v_2 \cos\theta - v_1 \sin\theta) = \mathbf{U}_\theta v.$$

Typically, w_θ is obtained by rotating a “basic” window $w = w_0$ through an angle θ, $w_\theta = w\left(\mathbf{U}_\theta v\right)$. When also a scale parameter h is exploited, the resulting estimates and kernels are denoted as $w_{h,\theta}$, $g_{h.\theta}$, respectively.

LASIP: HOW LPA-ICI WORKS

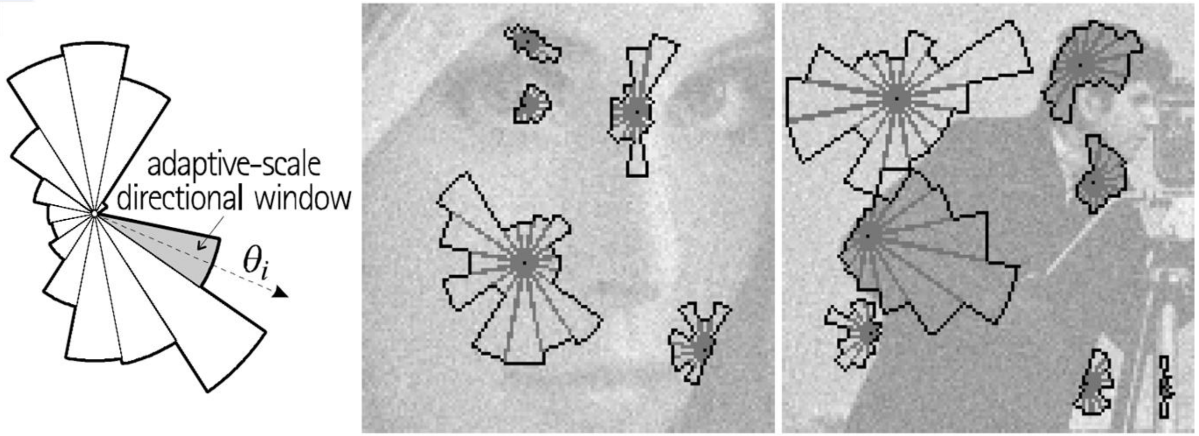
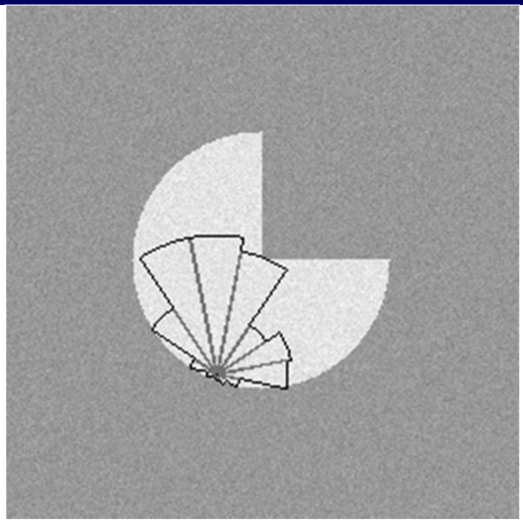
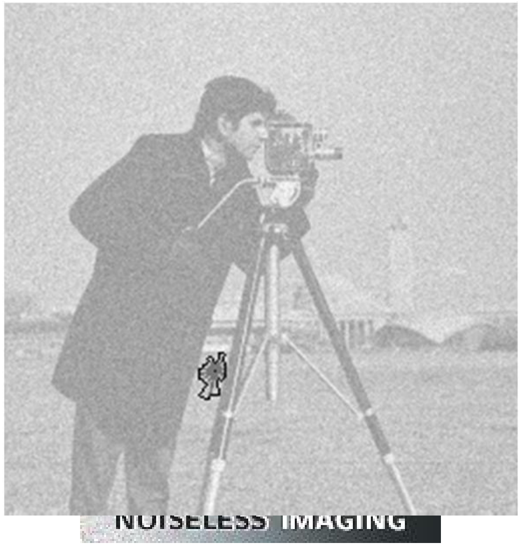
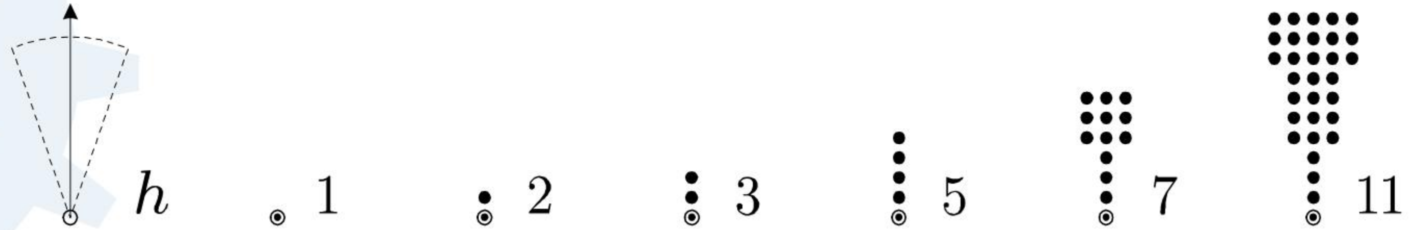


Figure 1: Anisotropic local approximations achieved by combining a number of adaptive-scale directional windows. The examples show some of these windows selected by the directional *LPA-ICI* for the noisy *Lena* and *Cameraman* images.

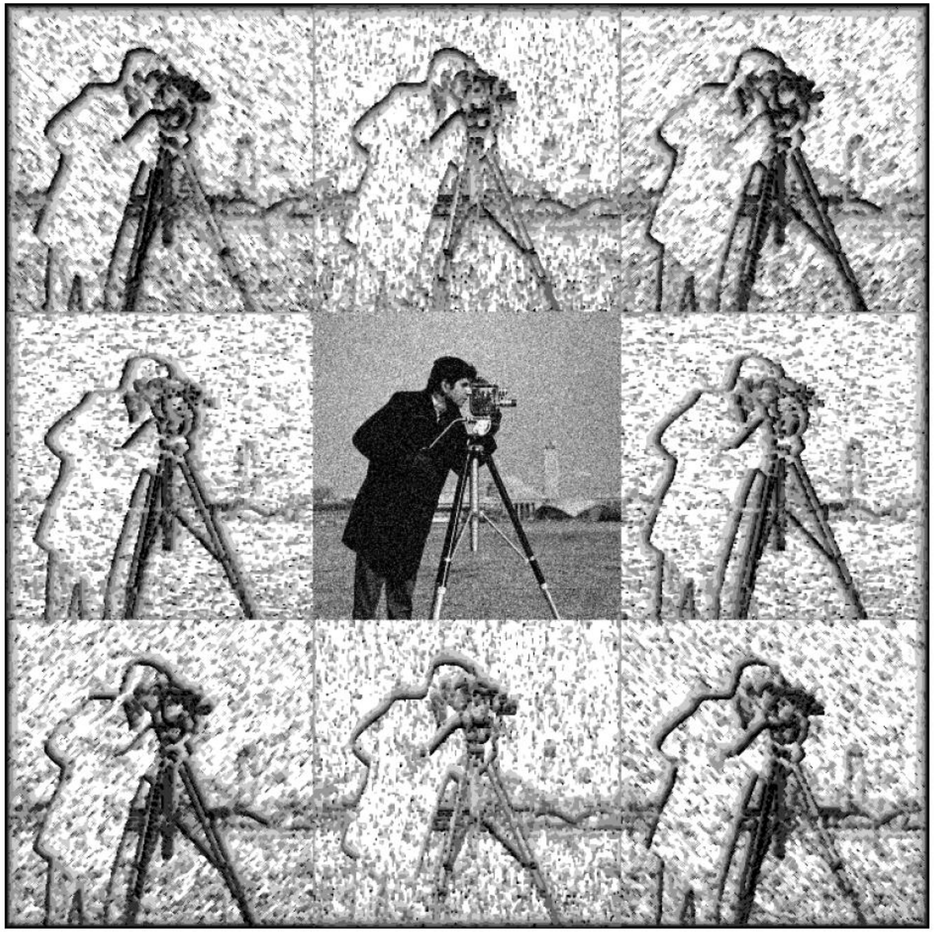




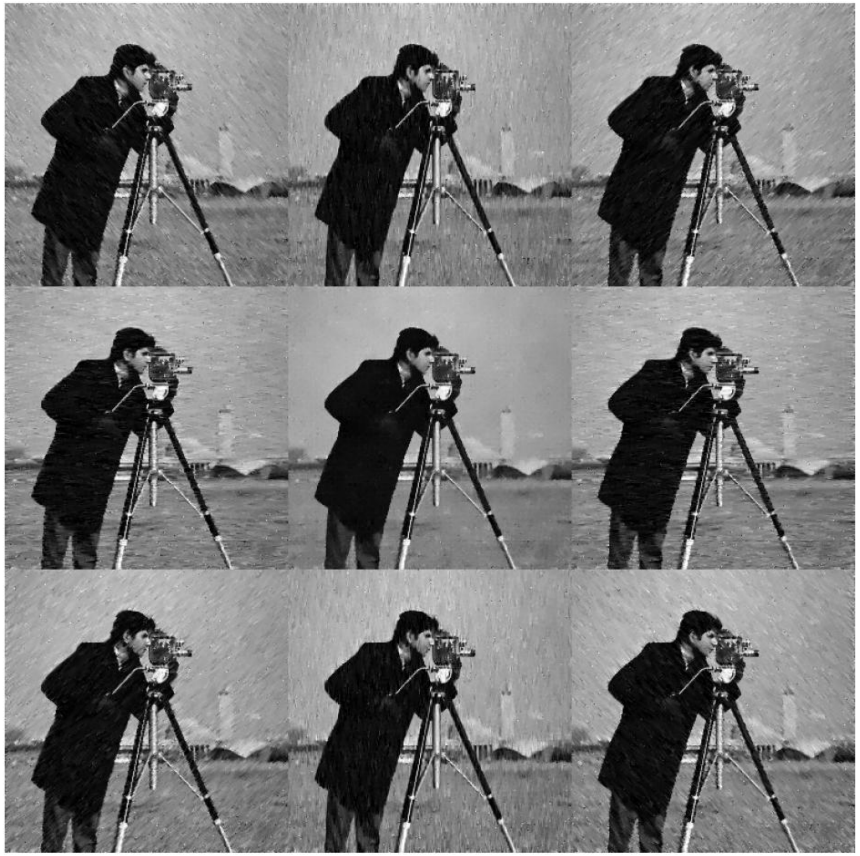
Anisotropic LPA-ICI: Kernels used in practice



The supports of the discrete kernels $g_{h_j,\pi/2}$, $h_j = 1, 2, 3, 5, 7, 11$. The origin pixel is marked with a circle.



Smaller scales are represented using a darker shade of gray.



Clockwise from top-left, the adaptive-scale estimates $\hat{y}_{h^+(x,\theta_k)}(x)\ \forall x$, $\theta_k = \frac{7\pi}{4}, \frac{3\pi}{2}, \frac{5\pi}{4}, \pi, \frac{3\pi}{4}, \frac{\pi}{2}, \frac{\pi}{4}, 0$, and, in the center, the fused anisotropic estimate $\hat{y}$.

The adaptive-scale directional estimates $\hat{y}_{h^+(x,\theta_i),\theta_i}(x)$ are "fused" into the final *anisotropic estimate* $\hat{y}$ by the convex linear combination

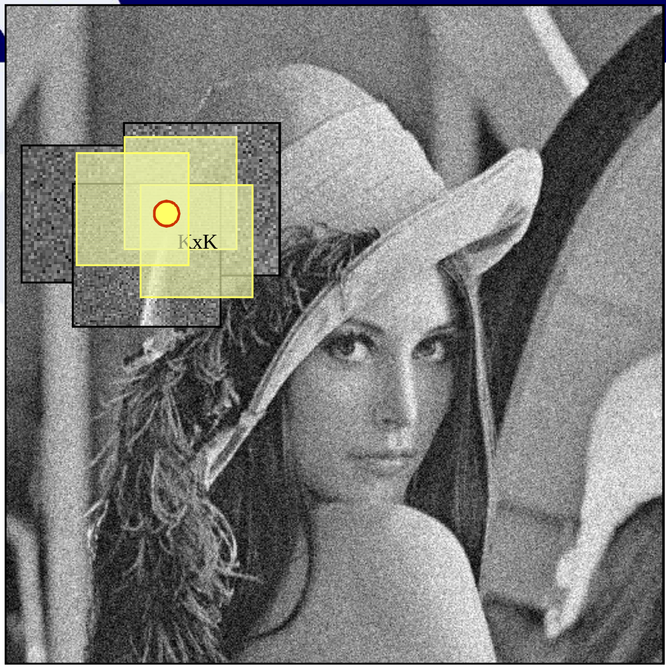
$$\begin{aligned} \hat{y}(x) &= \sum\nolimits_i \lambda(x,\theta_i)\,\hat{y}_{h^+(x,\theta_i),\theta_i}(x), \\ \lambda(x,\theta_i) &= \sigma^{-2}_{\hat{y}_{h^+(x,\theta_i),\theta_i}(x)} \Big/ \sum\nolimits_j \sigma^{-2}_{\hat{y}_{h^+(x,\theta_j),\theta_j}(x)}, \end{aligned} \tag{15}$$

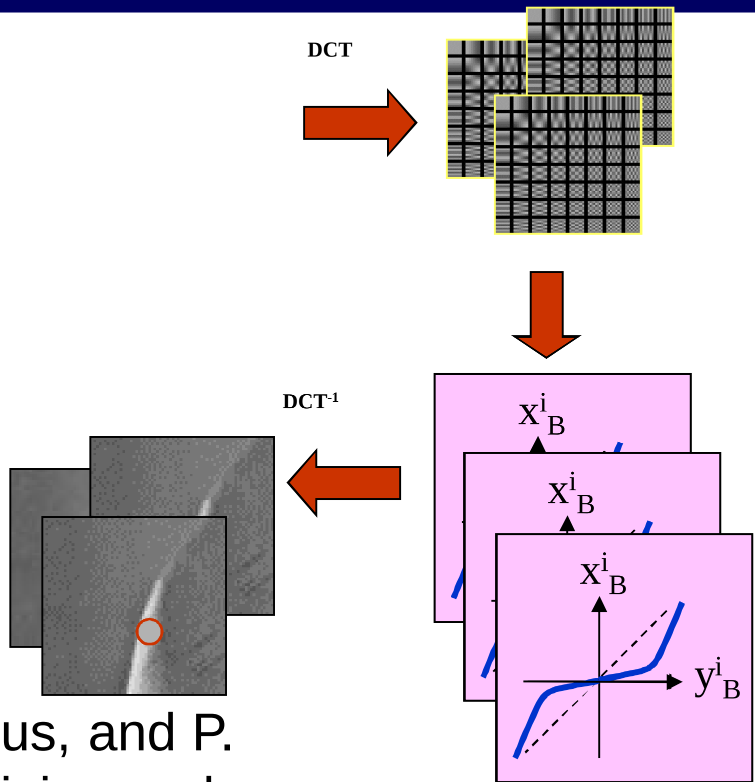
where the inverse of the variance of the adaptive estimates is used as the weighting factor.

θ_k	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$	$\hat{y}$
ISNR (dB)	4.13	3.57	4.08	3.56	4.11	3.44	4.07	3.55	8.07
SNR (dB)	18.52	17.96	18.47	17.95	18.50	17.83	18.46	17.95	22.46
MAE (ℓ^1)	10.67	11.55	10.80	11.59	10.69	11.70	10.82	11.58	6.44
RMSE (ℓ^2)	15.90	16.95	16.00	16.98	15.93	17.21	16.01	16.98	10.10
MAX (ℓ^∞)	131.6	114.7	124.2	117.0	112.6	142.5	114.4	125.9	85.3

Criteria values for the denoising of the *Cameraman* image using 8 directional adaptive estimates. The fused estimate is much better than each of the directional ones.

Sliding DCT denoising



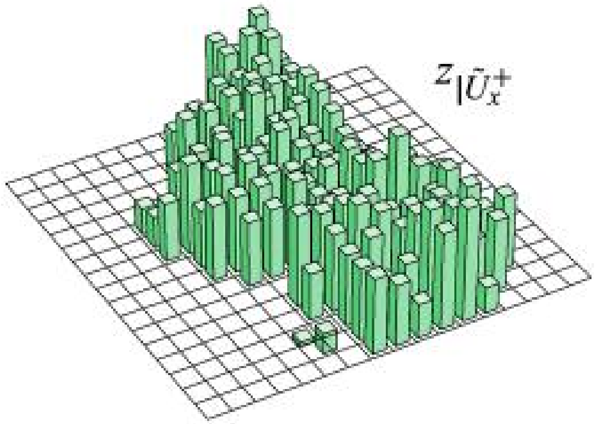


K. Egiazarian, J. Astola, M. Helsingius, and P. Kuosmanen (1999) “Adaptive denoising and lossy compression of images in transform domain”, J. Electronic Imaging

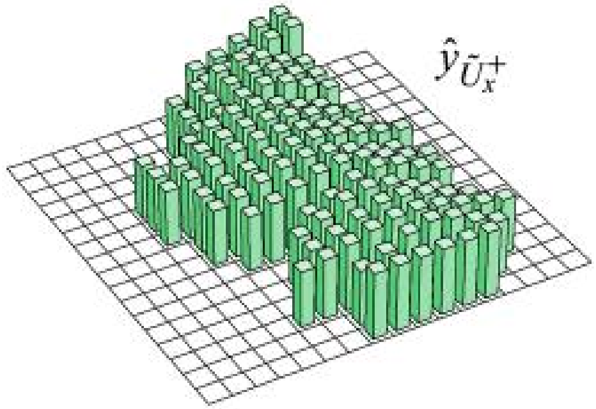
Shape-adaptive DCT image filtering

By demanding the local fit of a polynomial model, we are able to avoid the presence of singularities or discontinuities within the transform support. In this way, we ensure that data are represented sparsely in the transform domain, significantly improving the effectiveness of shrinkage (e.g., thresholding).

noisy image and adaptive-shape neighborhood

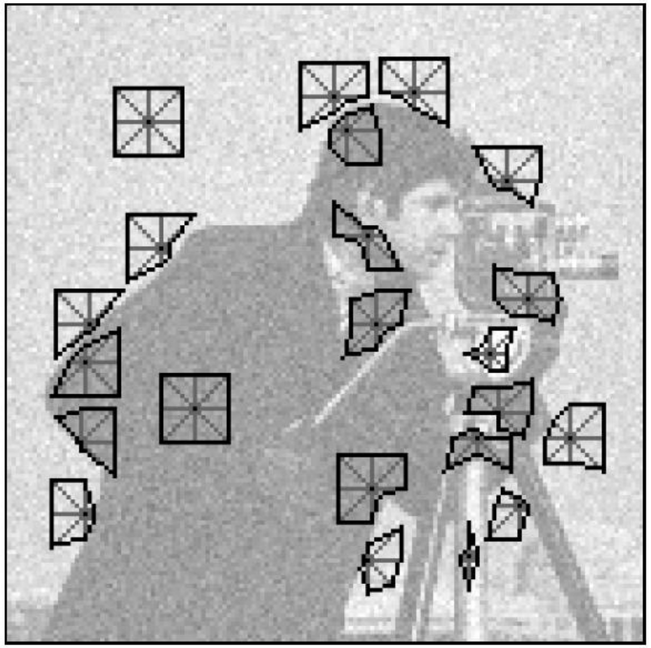


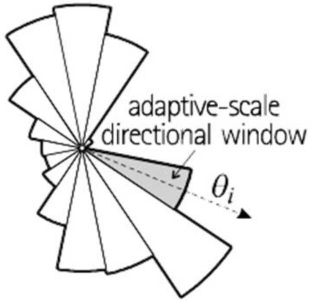
noisy data extracted from the neighborhood

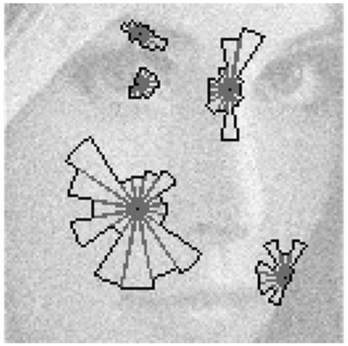


after hard-thresholding in SA-DCT domain

Shape-adaptation: use directional LPA-ICI

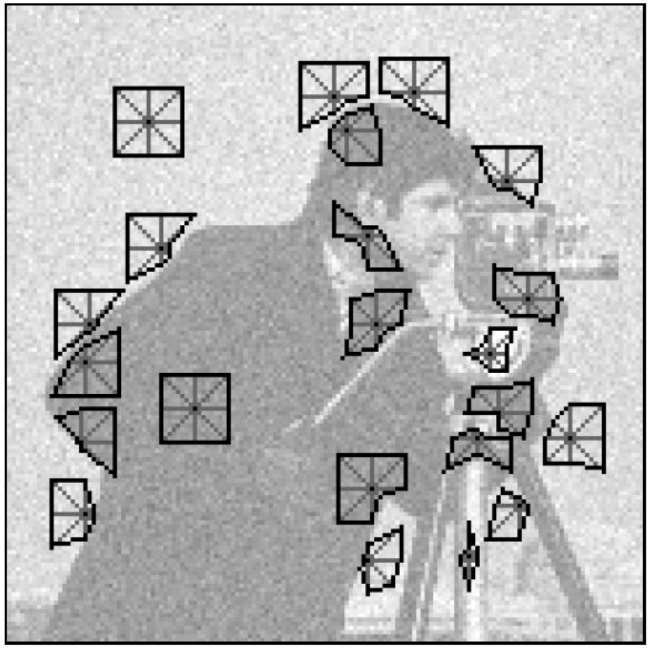


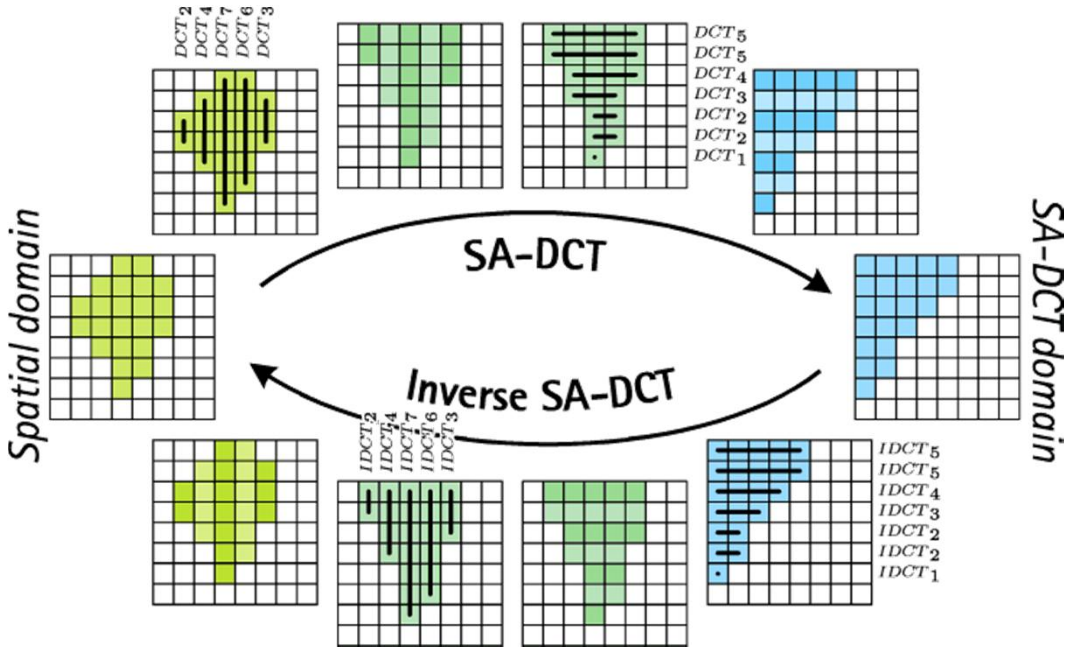




Shape-adaptive DCT image filtering

Pointwise SA-DCT: anisotropic neighborhoods





Shape-adaptive DCT image filtering

- **Direct generalization of the classical block-DCT (B-DCT);**
- **On rectangular domains (e.g., squares) the SA-DCT and B-DCT coincide;**
- **Comparable computational complexity as the separable B-DCT (fast algorithms);**
- **SA-DCT is part of the MPEG-4 standard;**
- **Efficient (low-power) hardware implementations available.**

Before our work on SA-DCT filtering, the SA-DCT had been used only for image and video compression.

Pointwise SA-DCT: denoising results

A fragment of Cameraman: noisy observation (σ=25, PSNR=20.14dB), BLS-GSM estimate (Portilla et al.) (PSNR=28.35dB), and the proposed Pointwise SA-DCT estimate (PSNR=29.11dB).

Pointwise SA-DCT: deblocking results

JPEG coded Cameraman with 2 different quality levels and the results of post-filtering using SA-DCT

Pointwise SA-DCT: deblurring results

Images blurred & noisy are deblurred & denoised by SA-DCT filter.

Pointwise SA-DCT: extension to color, motivation

Luminance-chrominance decompositions: structural correlation

R
G
B

color transformation

$$\mathbf{A}_{opp}=\begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\\ \frac{1}{\sqrt{6}} & 0 & \frac{-1}{\sqrt{6}}\\ \frac{1}{3\sqrt{2}} & \frac{-\sqrt{2}}{3} & \frac{1}{3\sqrt{2}}\end{bmatrix}$$

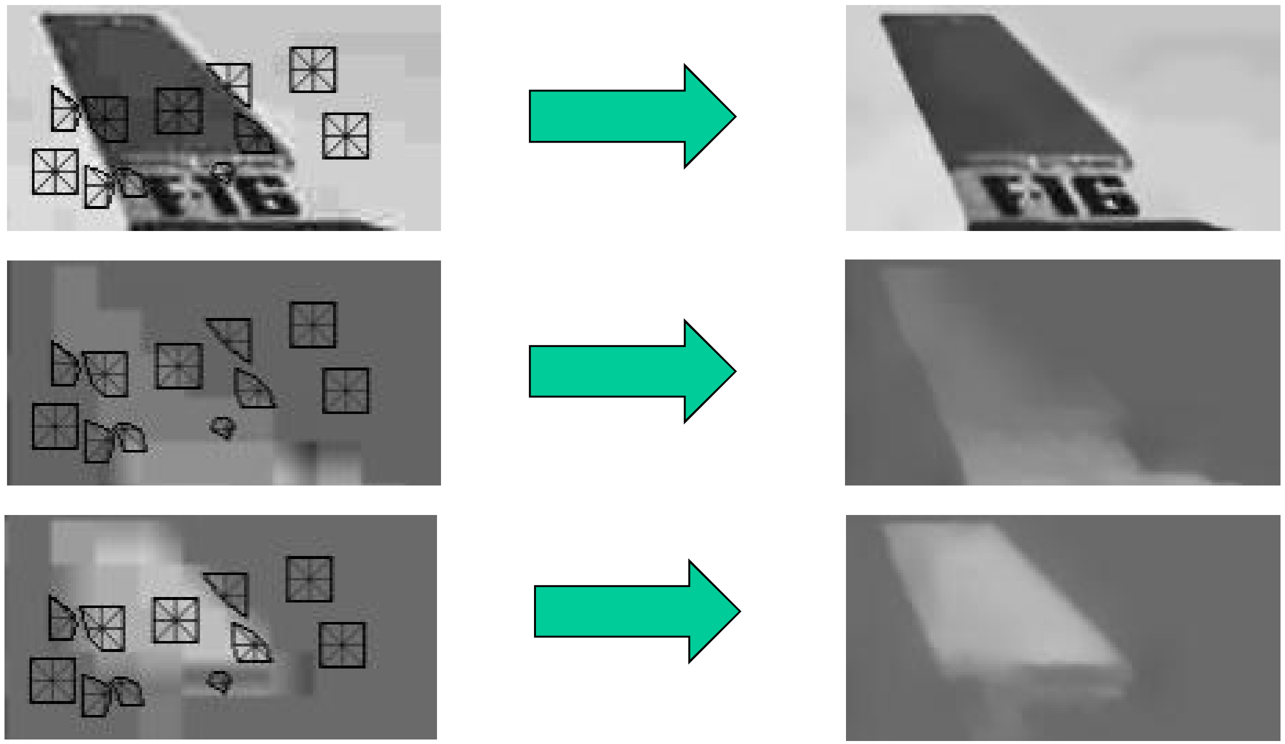
Y

U

V

Pointwise SA-DCT: structural contraint in luminance-chrominance space

Use for all three channels the adaptive neighborhoods defined by the anisotropic LPA-ICI for the luminance channel.



Pointwise SA-DCT: deblocking results

JPEG-compressed
(Q=10, 0.25bpp, PSNR=26.87dB)

Pointwise SA-DCT deblocking
(PSNR=28.30dB)

Pointwise SA-DCT: deblocking results

Pointwise SA-DCT: denoising results

Fragments of the noisy F-16 (σ=30, PSNR=18.59dB), of ProbShrink-MB (Pizurica et al.) estimate (PSNR=30.50dB), and of Pointwise SA-DCT estimate (PSNR=31.59dB).

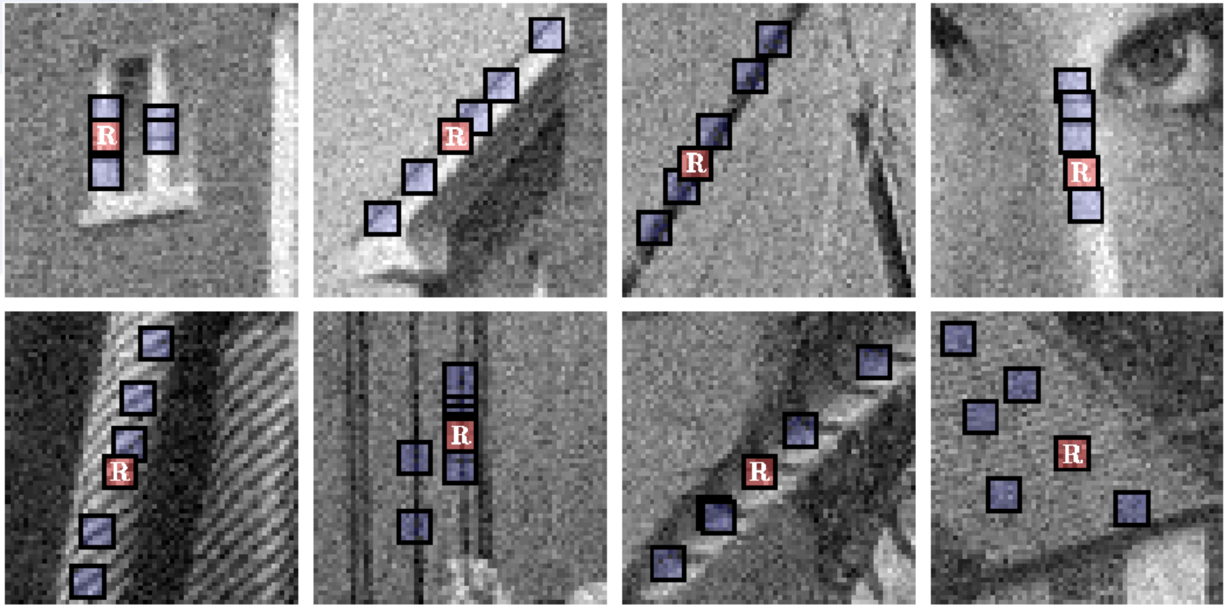
Block-Matching and 3D filtering (BM3D) denoising algorithm

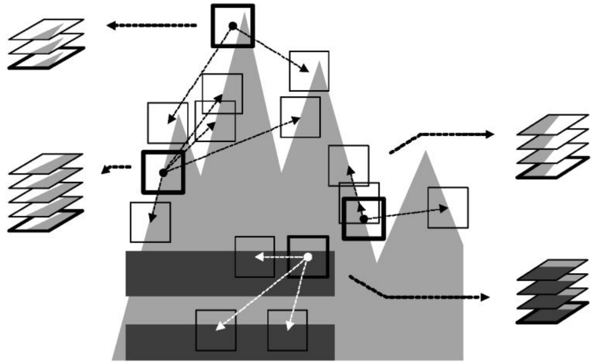
- Generalizes NL-means and overcomplete transform methods
- Current state-of-the-art denoising method

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3D filtering”, Proc. SPIE Electronic 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.

Block-matching and grouping



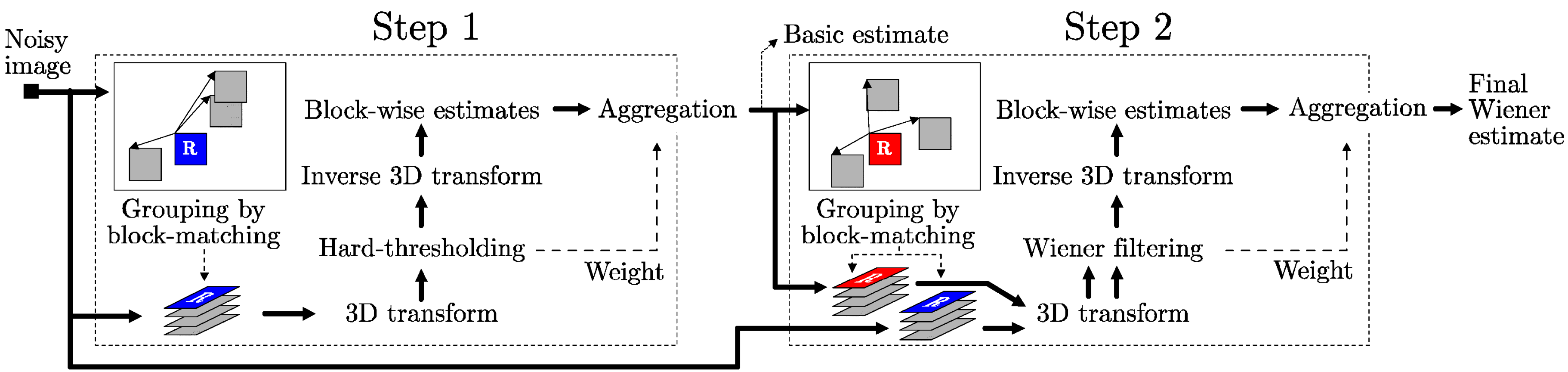


Groups are characterized by both:

- 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);

BM3D: 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.



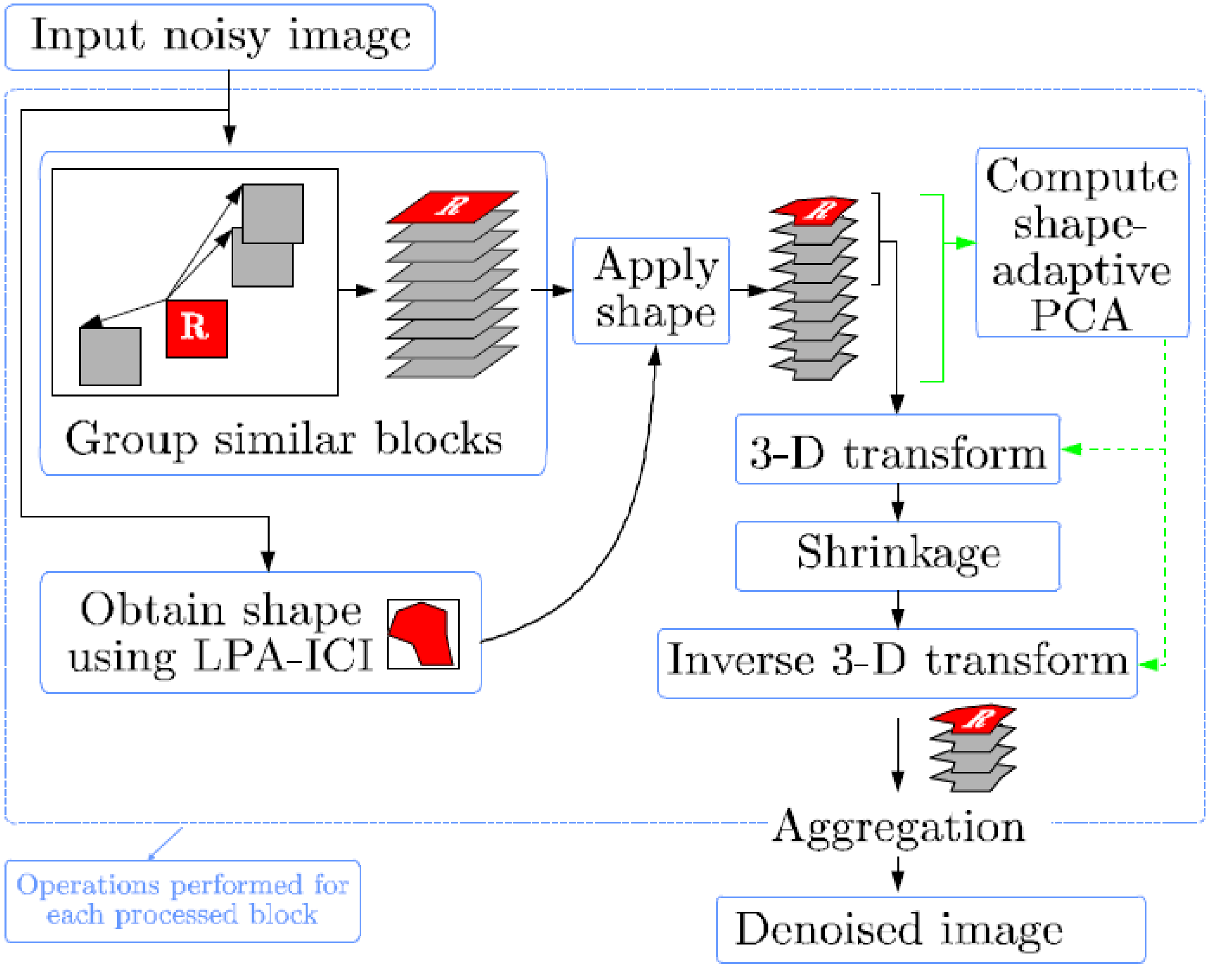
BM3D with Shape-Adaptive PCA (BM3D-SAPCA)

Main ingredients:

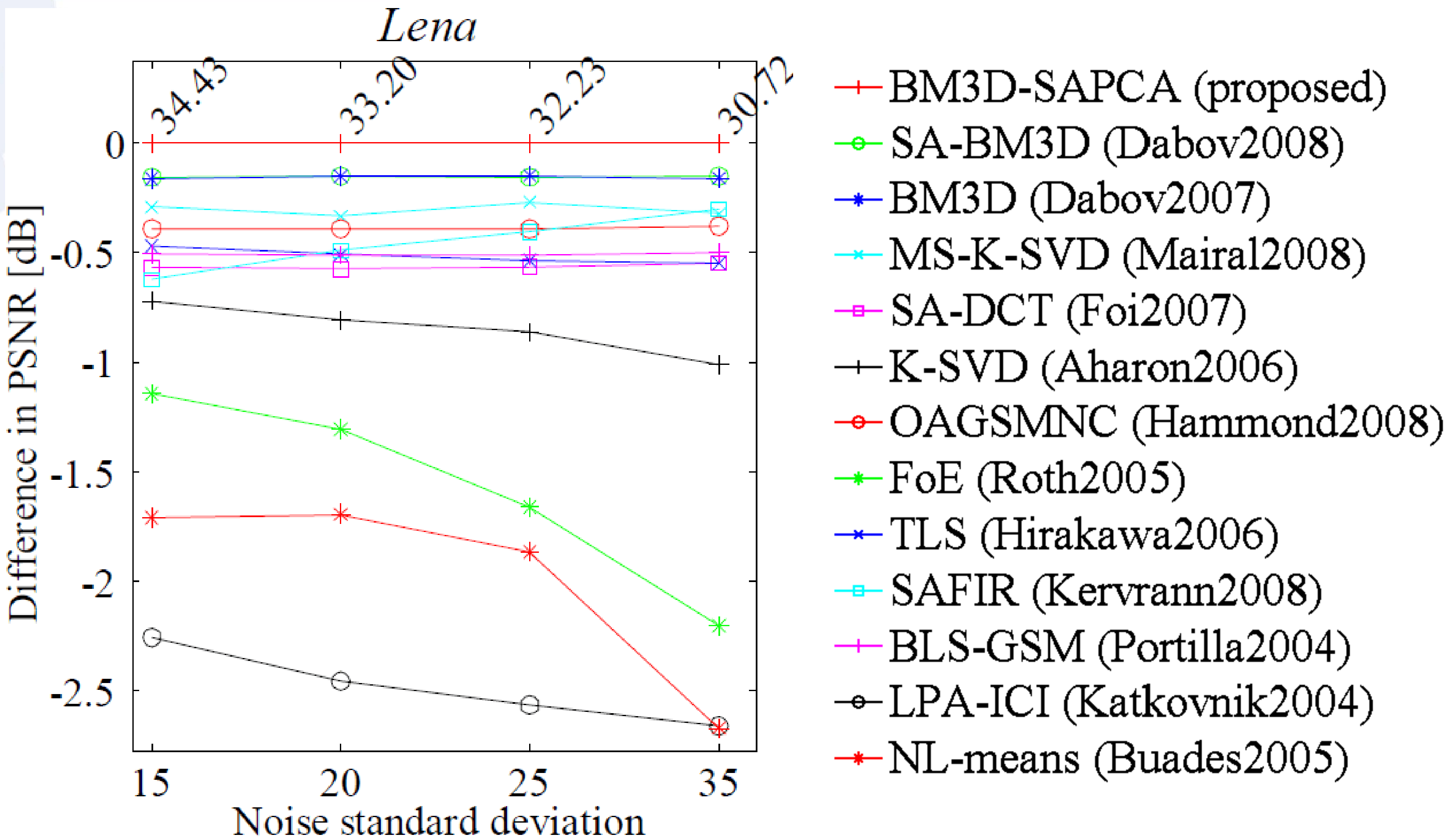
- Local Polynomial Approximation - Intersection of Confidence Intervals (LPA-ICI) to adaptively select support for 2-D transform;
- Block-Matching to enable non-locality;
- Shape-Adaptive PCA (SA-PCA);
- Shape-Adaptive DCT low-complexity 2-D transform on arbitrarily-shaped domains (when SA-PCA is not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

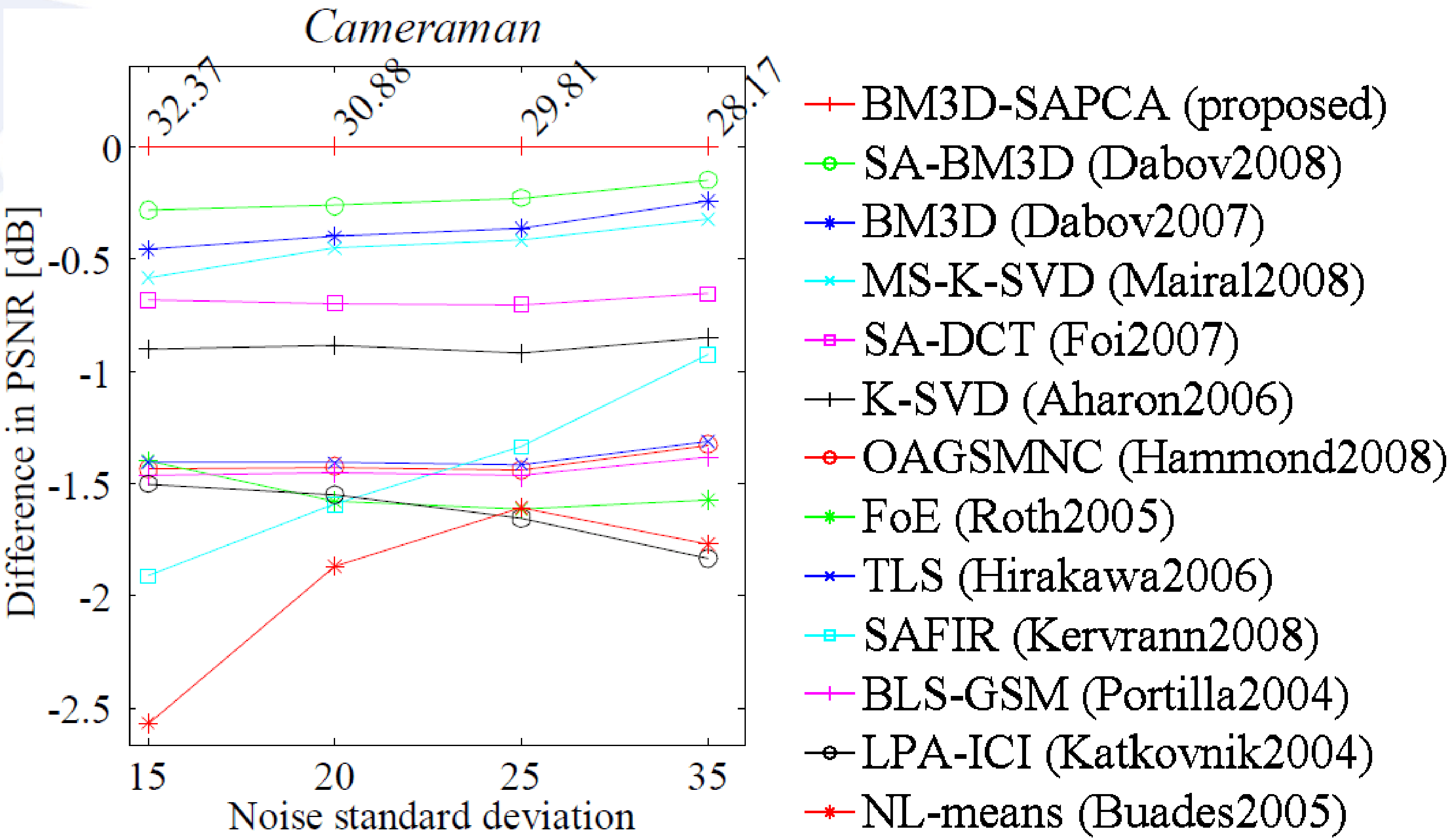
BM3D-SAPCA


Input noisy image
Group similar blocks
Apply shape
Compute shape-adaptive PCA
3-D transform
Shrinkage
Obtain shape using LPA-ICI
Inverse 3-D transform
Aggregation
Operations performed for each processed block
Denoised image

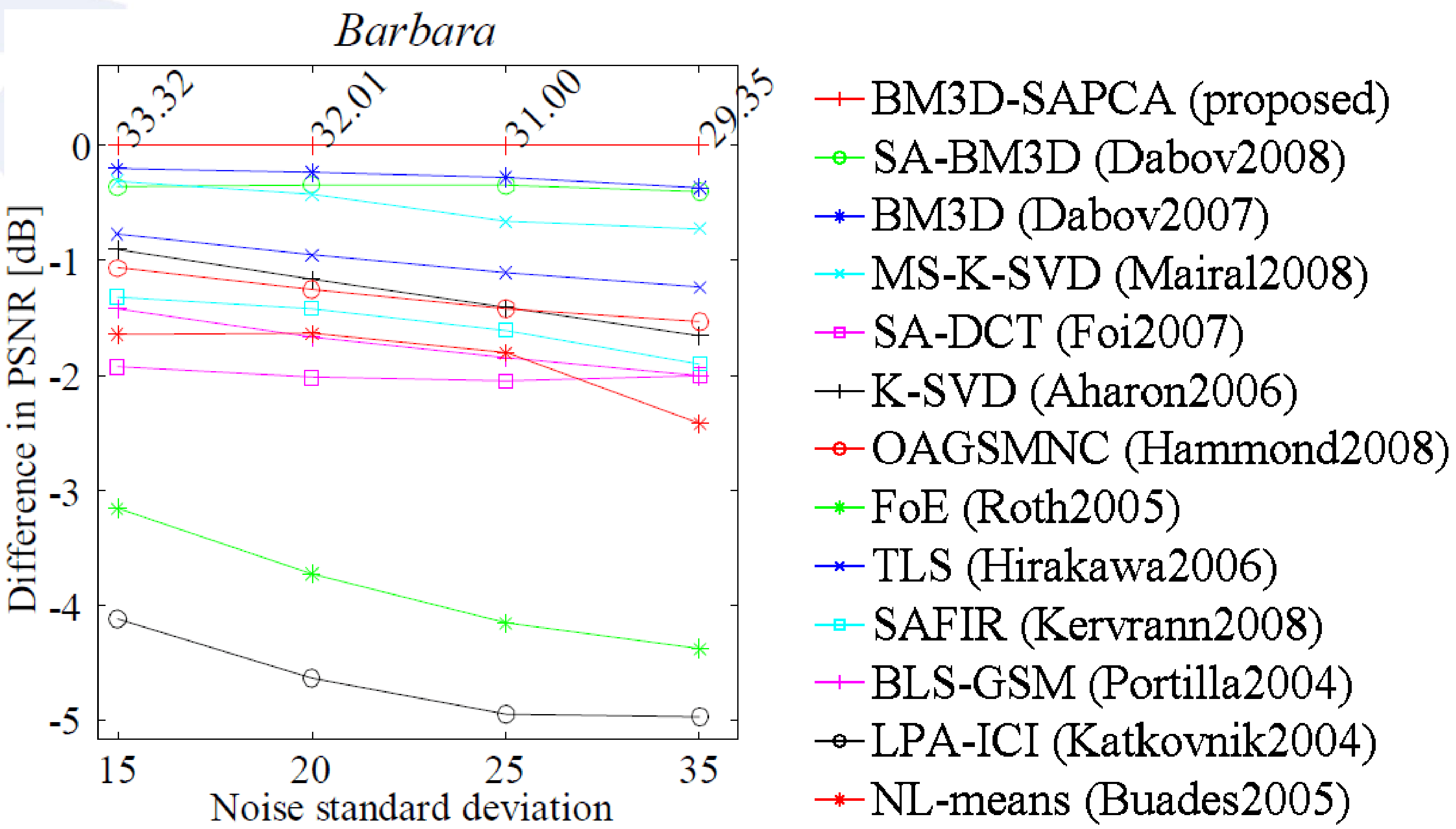
Comparison of BM3D-SAPCA with other filters



Comparison of BM3D-SAPCA with other filters



Comparison of BM3D-SAPCA with other filters



Comparison of BM3D-SAPCA with other filters (PSNR, SSIM)

Original

Noisy, $\sigma = 35$

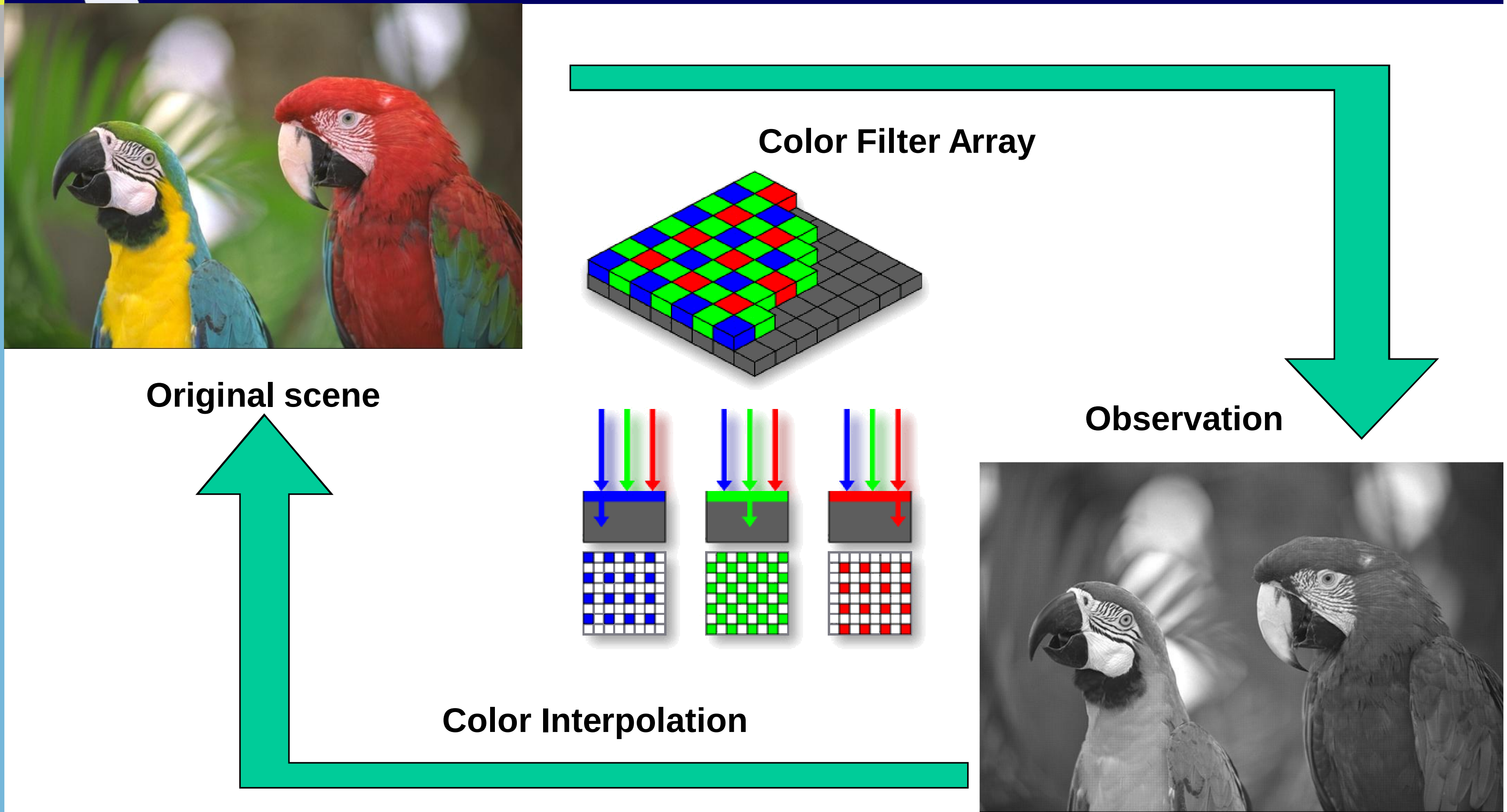
BM3D (27.82, 0.8207)

P.SADCT (27.51, 0.8143)

SA-BM3D (28.02, 0.8228)

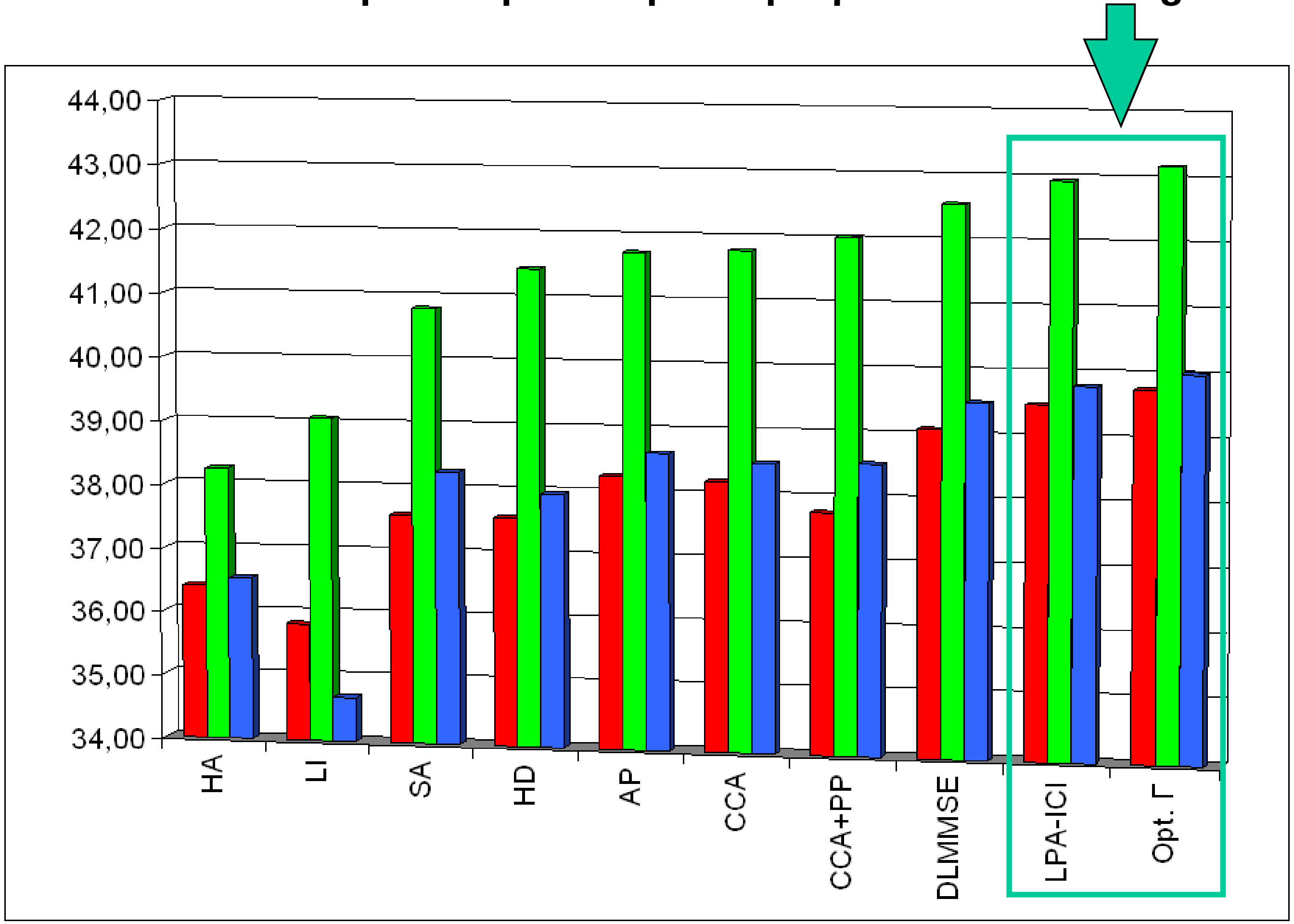
BM3D-SAPCA (28.16, 0.8269)

Interpolation for Bayer Pattern

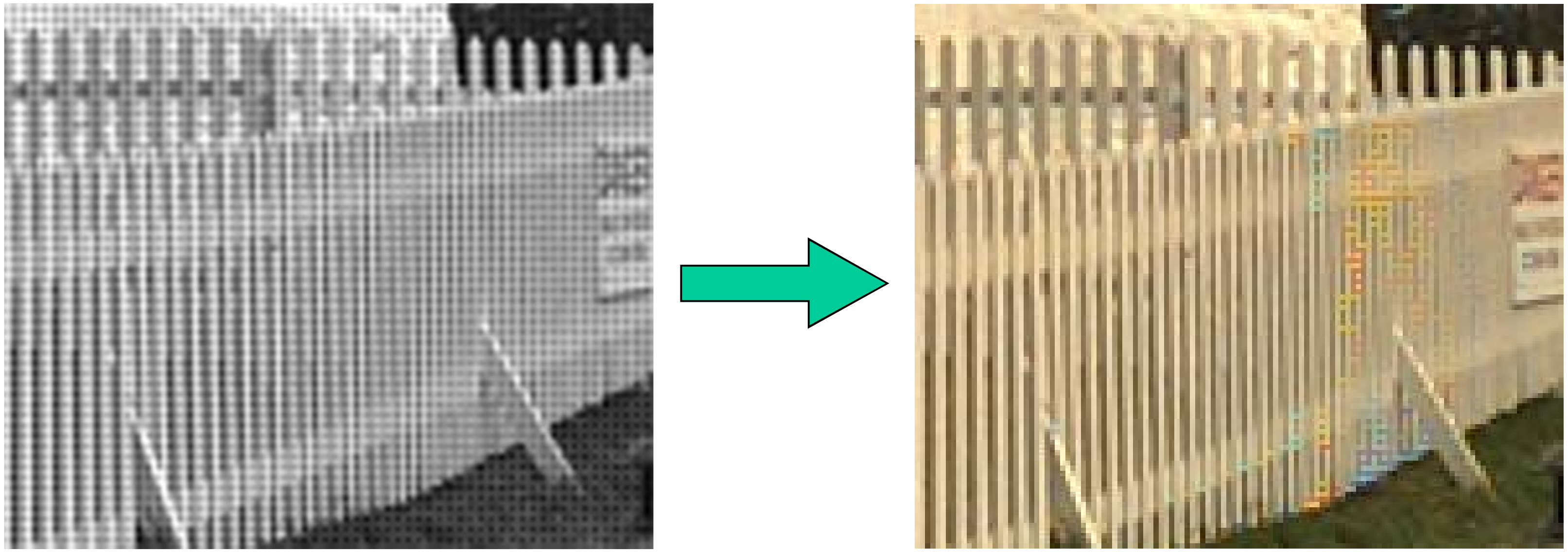


Competitiveness with state-of-the-art techniques

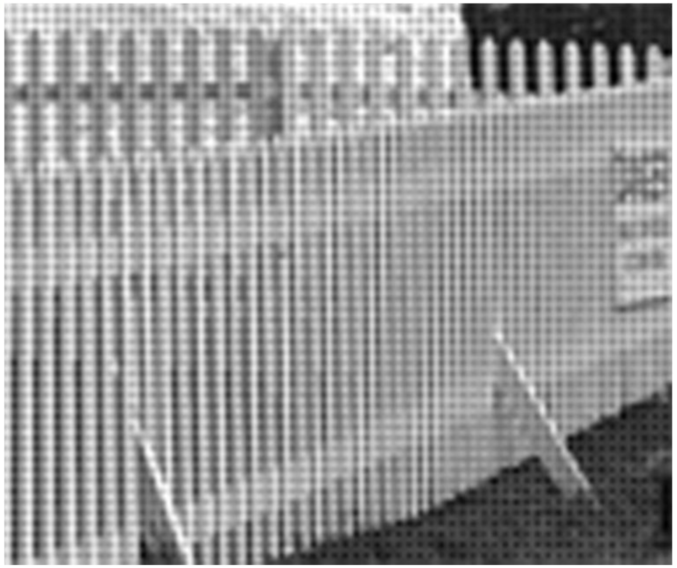
The proposed CFAI technique adapts to spatial properties of an image



Conventional Approach for Noiseless Data (Hamilton-Adams)



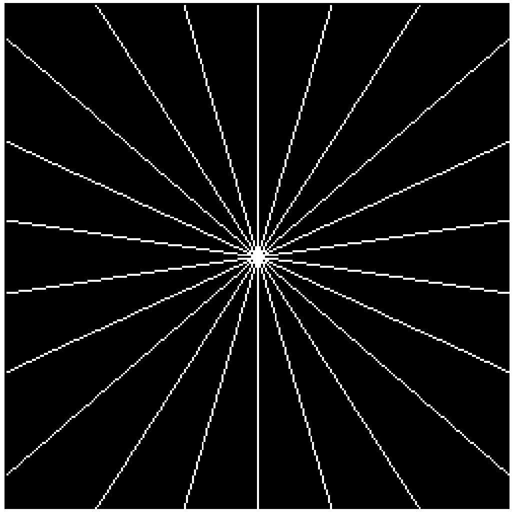
Proposed Approach for Noiseless Data (Spatially-Adaptive LPA-ICI)

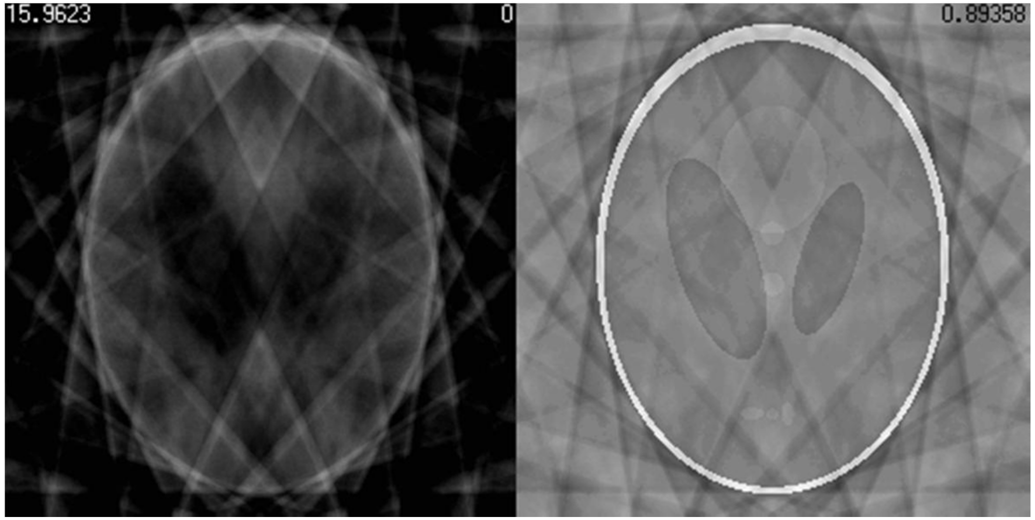


Compressed Sensing Image Reconstruction via Recursive BM3D

Egiazarian, K., A. Foi, and V. Katkovnik, "**Compressed Sensing Image Reconstruction via Recursive Spatially Adaptive Filtering,** *ICIP 2007*

Simulation of Radon reconstruction from sparse projections (approximating Radon projections as radial lines in FFT domain: Sparse projections: 11 radial lines)

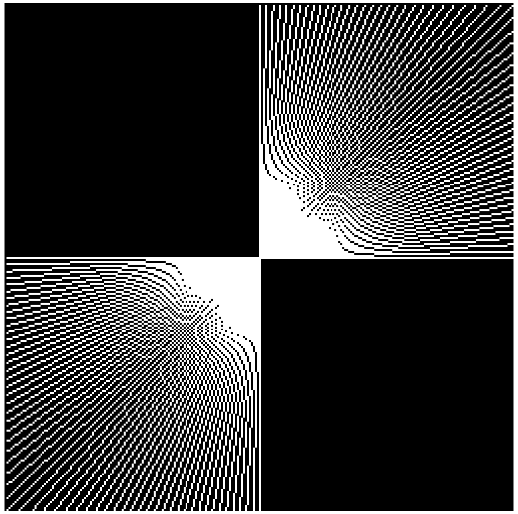


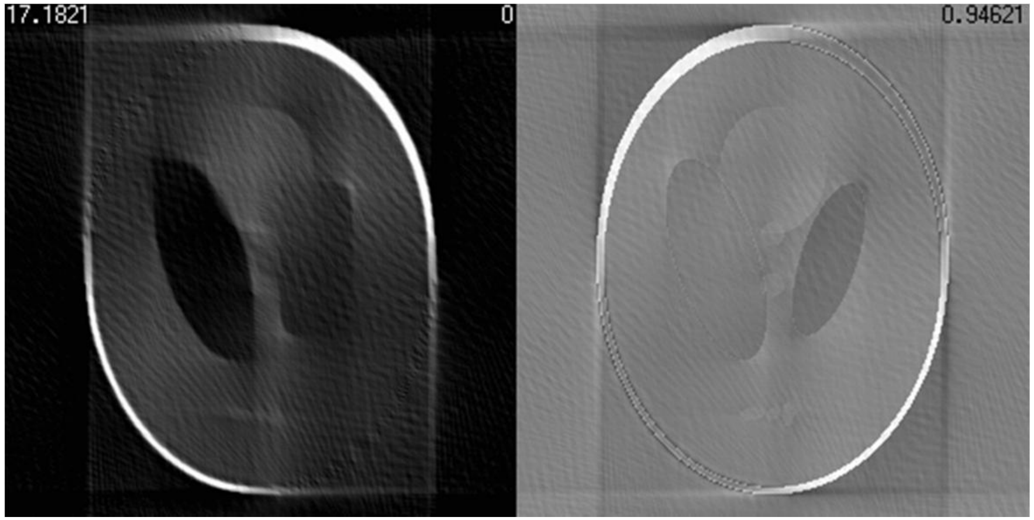


Compressed Sensing Image Reconstruction via Recursive BM3D

Egiazarian, K., A. Foi, and V. Katkovnik, "**Compressed Sensing Image Reconstruction via Recursive Spatially Adaptive Filtering,** *ICIP 2007*

Simulation of Radon reconstruction from sparse projections (approximating Radon projections as Limited-angle in FFT domain)





BM3D for upsampling and super-resolution

Image **upsampling** or **zooming**, can be de.ned as the process of resampling a single low-resolution (LR) image on a high-resolution grid.

The process of combining a sequence of undersampled and degraded low-resolution images in order to produce a single high-resolution image is commonly referred to as a **Super-resolution** (SR) reconstruction.

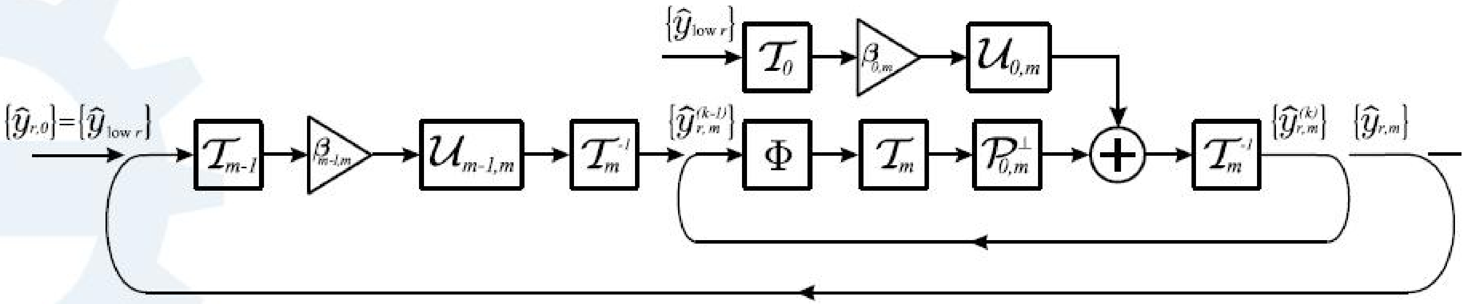
Modern SR methods (e.g., Protter et al. 2008, Ebrahimi and Vrscay 2008) are based on the nonlocal means (NLM) filtering paradigm (Buades-Coll-Morel, 2005).

• No explicit registration: one-to-one pixel mapping between frames is replaced by a one-to-many mapping.

The BM3D and V-BM3D algorithms share with the NLM the idea of exploiting nonlocal similarity between blocks. However, in (V-)BM3D a more powerful transform-domain modeling is used.

BM3D based superresolution

Multistage iterative reconstruction



$$
\begin{cases}
\hat{y}_{r,0} = y_{\text{low }r} & \text{(algorithm input)} \\
\hat{y}_{r,m} = \hat{y}_{r,m}^{(k_{\text{final }m})} & \text{(stage input)} \\
\hat{y}_{r,m}^{(0)} = \mathcal{T}_m^{-1}\left(\mathcal{U}_{m-1,m}\left(\beta_{m-1,m}\mathcal{T}_{m-1}\left(\hat{y}_{r,m-1}\right)\right)\right) & \\
\hat{y}_{r,m}^{(k)} = \mathcal{T}_m^{-1}\left(\mathcal{U}_{0,m}\left(\beta_{0,m}\mathcal{T}_0\left(y_{\text{low }r}\right)\right) + \mathcal{P}_{0,m}^{\perp}\left(\mathcal{T}_m\left(\Phi\left(r, \left\{\hat{y}_{r,m}^{(k-1)}\right\}_{r=1}^{R}, \sigma_{k,m}\right)\right)\right)\right) &
\end{cases}
$$

m	stage number	$m = 1, \ldots, M$
k	iteration number	$k = 0, \ldots, k_{\text{final }m}$
$\hat{y}_{r,m}^{(k)}$	estimate for $\hat{y}_r$ on iter. k of stage m	
$\mathcal{T}_m$	transform	
Φ	spatially adaptive filter (V-BM3D)	
$\sigma_{k,m}$	parameter controlling the strength of the filter	$\sigma_{k,m} = \sigma_{k,m-1} - \Delta_m$

Image upsampling x 4 (pixel replication)

Image upsampling x 4 in wavelet domain (Danielyan et al. EUSIPCO 2008)

Video superresolution comparison with (Protter et. al.)

Nearest neighbor Ground truth Protter et. al. Proposed

1. M. Protter, M. Elad, H. Takeda, and P. Milanfar, .Generalizing the Non-Local-Means to Super-Resolution Reconstruction., IEEE Trans. Image Process., 2008.
2. A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, .Image upsampling via spatially adaptive block-matching filtering, EUSIPCO2008, Lausanne, Switzerland, Aug. 2008.

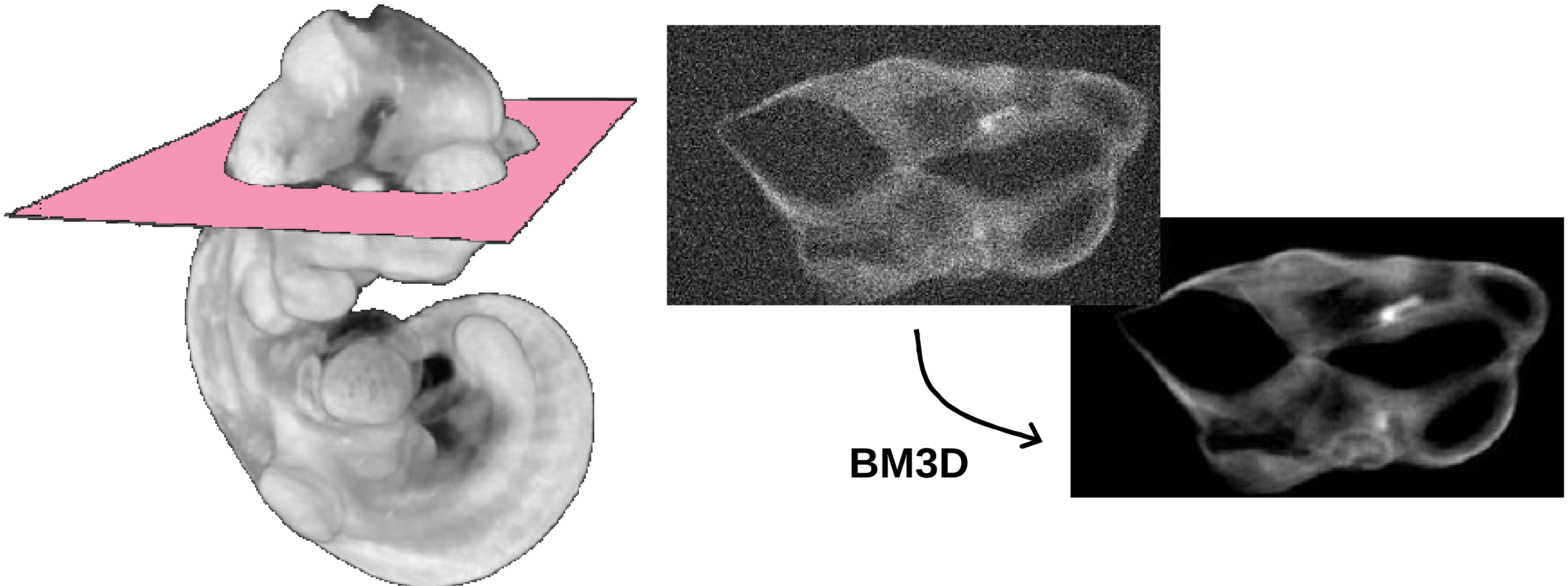
Examples: Video denoising using V-BM3D

Examples: Video denoising using V-BM3D

Examples: Video denoising using V-BM3D

Conclusions

Our algorithms have been licensed to major digital camera manufacturers and are already in use by various research institutes for processing and enhancing their images.



Tomographic reconstruction of mouse embryo with BM3D filtering of axial slices (Harvard Medical School, Boston MA, 2010)

Conclusions

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.

IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 19, NO. 4, APRIL 2010 895

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

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 pointwise 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

ISELESS IMAGING

Thanks to collaborators and former students:

Jaakko Astola (TUT, Finland)
Leonid Yaroslavsky (Israel) –Sliding DCT denoising (1997-2000)
Dmitiy Paliy (NOKIA, Finland) LPA-ICI CFAI (2005-2007)
Rusen Oktem (Berkeley, USA) Sliding DCT
Aram Danielyan (Noiseless Imaging)
Enrique Sánchez-Monge (Noiseless Imaging) super-resolution using BM3D (2008-)
Alessandro Foi (TUT, Finland) SA DCT, BM3D,…
Vladimir Katkovnik (TUT, Finland) LPA-ICI, SA-DCT, BM3D,…
Kostadin Dabov (Apple, USA), BM3D
Dmytro Rusanovsky (LG, USA) Video BM3D
and many-many others…