Image denoising by averaging, including NL-means algorithm

A. Buades
CNRS - Paris Descartes

J.M Morel
ENS-Cachan

Master Mathematiques / Vision / Aprentissage

ENS Cachan, 26 movember 2010

Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

Noise estimation

Photography II

NLmeans + Transform domain methods

Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

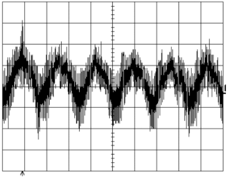
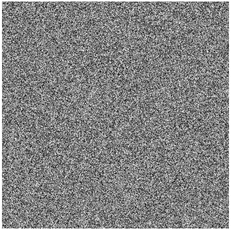
Noise estimation

Photography II

NLmeans + Transform domain methods

Noise

- Images by typing "noise" at google

- Noise : "random, unpredictable, and undesirable signals, or changes in signals, that mask the desired information content".
- Noise : "random fluctuations that do not contain meaningful data or other information".

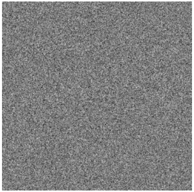
Noise images

- We assume an additive white noise model

$$v(x, y) = u(x, y) + n(x, y)$$

  In practice, we simulate the noise as i.i.d Gaussian variables $n(x, y) \sim N(0, \sigma^2)$


=

+


- Other types of noise are related to this one or reduces to it in certain circumstances.

Averaging

- The principle of most denoising methods is quite simple: Replace the color of a pixel with an average of the nearby pixels colors.

  If X_i are i.i.d of standard deviation σ

  $$Var\,(\frac{X_1 + \cdots + X_m}{m}) = \frac{\sigma^2}{m}$$

  The average reduces the uncertainty by m.

- If $\hat{u}$ denotes the average of N noisy values $v(x_1), \cdots, v(x_N))$ then

$$E\{\|u - \hat{u}\|^2\} \quad = \quad E\{\|u - \frac{1}{N}(u(x_1) + \ldots + u(x_N))\|^2\} + \frac{\sigma^2}{N}$$

Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

Noise estimation

Photography II

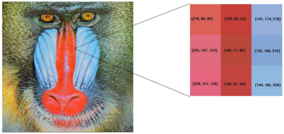
NLmeans + Transform domain methods

Gaussian Filtering

- **Average of the spatially closest pixels**

  As closer pixels are more dependent they should have a similar grey level value.

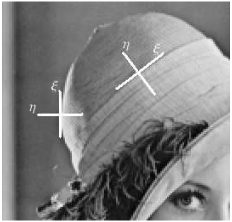
$$M_h u(\mathbf{x}) = \frac{1}{\pi h^2} \int_{B_h(\mathbf{x})} u(\mathbf{y}) d\mathbf{y},$$



The assumption is only valid for homogeneous regions and therefore edges and texture are blurred.

Anisotropic filtering

- **Average of spatially close pixels in the direction of the level line**



The vector $\eta = \frac{Du}{|Du|}$ and $\xi = \frac{Du^{\perp}}{|Du|}$ are respectively orthogonal and parallel to the level line passing trough x.

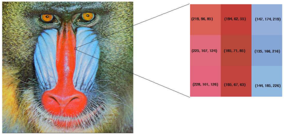
$$AF_h u(\mathbf{x}) = G_h * u_{|l(\vec{\xi})} = \int_{\mathbb{R}} G_h(t) u(\mathbf{x} + t\vec{\xi}) dt,$$

where G_h is the one-dimensional Gauss function of standard deviation h.

The straight edges are well restored while flat and textured regions are degraded.

Neighborhood filter

- **Average of pixels both closer in spatial and grey level distance**



In order to denoise the central red pixel, it would be better to average the color of this pixel with the nearby red pixels and only them, excluding the blue ones.

$$YNF_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{B_\rho(\mathbf{x})} e^{-\frac{|\boldsymbol{u}(\mathrm{y})-\boldsymbol{u}(\mathrm{x})|^2}{\boldsymbol{h}^2}} u(\mathbf{y})d\mathbf{y},$$

where $C(\mathbf{x})$ is a normalizing factor, $B_\rho(\mathbf{x})$ is a ball of center $\mathbf{x}$ and radius ρ and h is the filtering parameter.

PDEs filtering and enhancement

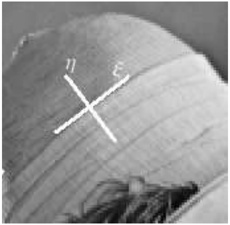
- The heat equation.

$$u_t = \Delta u.$$

The heat equation is an isotropic diffusion.

$$\Delta u = u_{\xi\xi} + u_{\eta\eta}$$

where $\xi = Du^{\perp}/|Du|$ and $\eta = Du/|Du|$.



$$u_{\eta\eta} = D^2u(\frac{Du}{|Du|}, \frac{Du}{|Du|}),$$

$$u_{\xi\xi} = D^2u(\frac{Du^{\perp}}{|Du|}, \frac{Du^{\perp}}{|Du|}),$$

PDEs filtering and enhancement

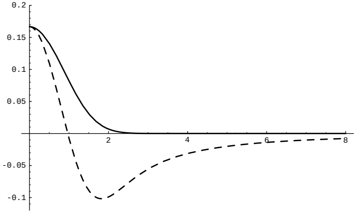
- The convolution with a gaussian kernel G_h is such that
$$u - G_h * u = -h^2 \Delta u + o(h^2),$$
for h small enough.

- The image method noise of an anisotropic filter AF_h is
$$u(\mathbf{x}) - AF_h u(\mathbf{x}) = -\frac{1}{2} h^2 u_{\xi\xi} + o(h^2),$$

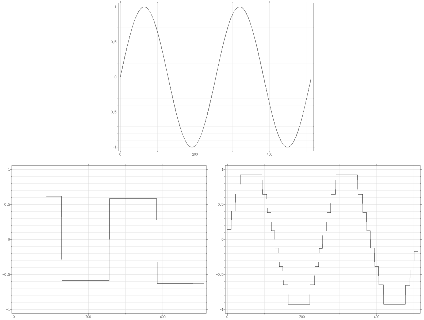
Neighborhood filters and PDEs

Theorem

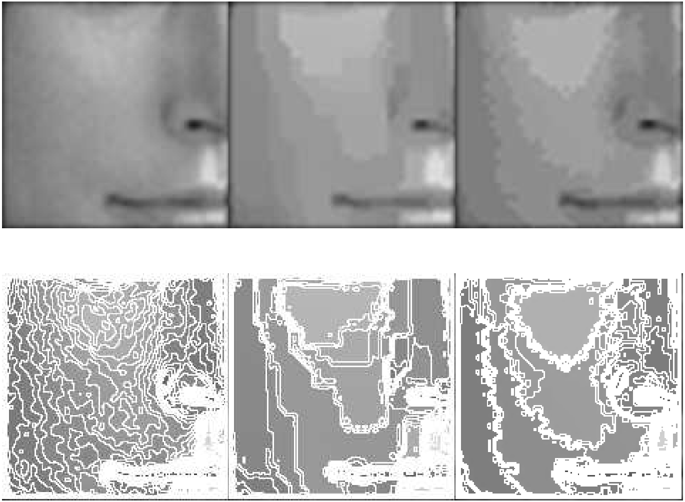
$$YNF_{h,\rho}u(\mathbf{x}) - u(\mathbf{x}) \simeq \left[\tilde{g}(\frac{\rho}{h}\,|Du(\mathbf{x})|)\ u_{\xi\xi}(\mathbf{x}) + \tilde{f}(\frac{\rho}{h}\,|Du(\mathbf{x})|)\ u_{\eta\eta}(\mathbf{x})\right]\rho^2$$



$\hat{g}$ —— $\hat{f}$ - - -



Singularities are created due to the transition of smoothing to enhancement. The number of enhanced regions strongly depends upon the ratio $\frac{\rho}{h}$.

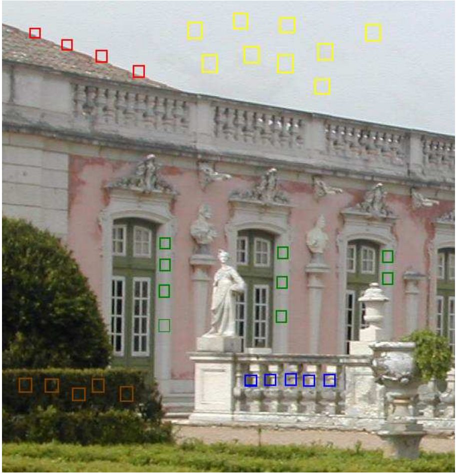


The level lines of the Perona-Malik filter and the neighborhood filter tend to group creating flat zones.

Outline

- Noise. Averaging
- Local smoothing filters
- Image autosimilarity. NLmeans
- Movie denoising
- Photography
- Noise estimation
- Photography II
- NLmeans + Transform domain methods

Image autosimilarity



Groups of similar windows in a digital image, long range interaction. First used by Efros and Leung for texture synthesis.

Efros Leung Algorithm

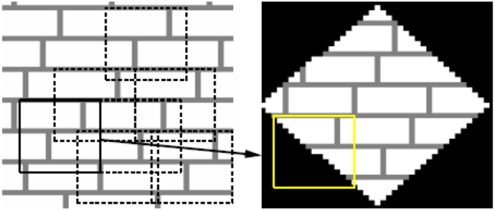
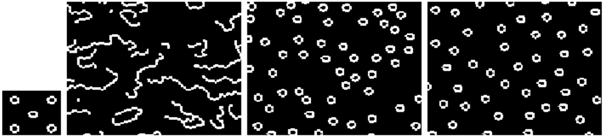


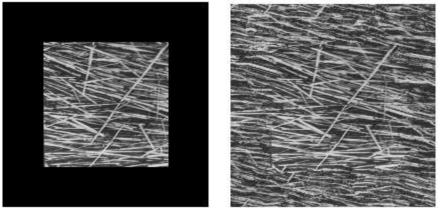
Figure 1. Algorithm Overview. Given a sample texture image (left), a new image is being synthesized one pixel at a time (right). To synthesize a pixel, the algorithm first finds all neighborhoods in the sample image (boxes on the left) that are similar to the pixel's neighborhood (box on the right) and then randomly chooses one neighborhood and takes its center to be the newly synthesized pixel.

Efros Leung Examples

- Texture synthesis



- Interpolation



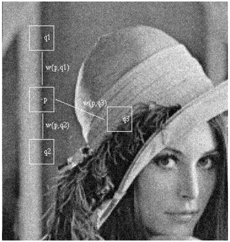
NL-means

- **NL-means filter. Average of pixels with a similar configuration in a whole Gaussian neighborhood.**

$$NL_h[u](\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{\Omega} e^{-\frac{1}{h^2} \int_{\mathbb{R}^2} G_a(t)|u(\mathbf{x}+t)-u(\mathbf{y}+t)|^2 dt} \; u(\mathbf{y}) \; d\mathbf{y},$$

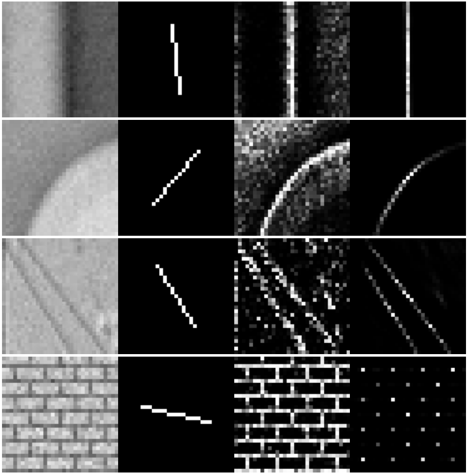
where G_a is a Gaussian kernel of standard deviation a and h acts as a filtering parameter.

Non Local: Pixels of the whole image take part of the previous average.



Markovian hypothesis: Pixels with a similar neighborhood have a similar grey level value.

Average configuration



Methods evaluation

We want to remove as much noise as possible, preserving all the original information and without any artifact.

- **Preservation of original information.** Features in $n(D_h, v) = v - D_h v$ are removed from v. We call this difference *method noise* when v is non or slightly noisy.

> *For every denoising algorithm, the method noise must be zero if the image contains no noise and should be in general an image of independent zero-mean random variables.*

Methods evaluation

- **No artifacts** The transformation of a white noise into any correlated signal creates structure and artifacts.

  > *A denoising algorithm must transform a white noise image into a white noise image (with lower variance).*

  $$E\{D_h n(i) \cdot D_h n(j)\} = 0 \quad \text{for } i \neq j.$$

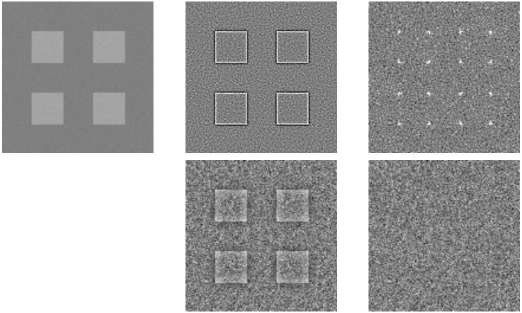
  and

  $$Var\{D_h n(i)\} << \sigma^2 \quad \text{for } i \in I.$$

- **Visual comparison** Visual inspection of denoised image.

Evaluation: Method noise

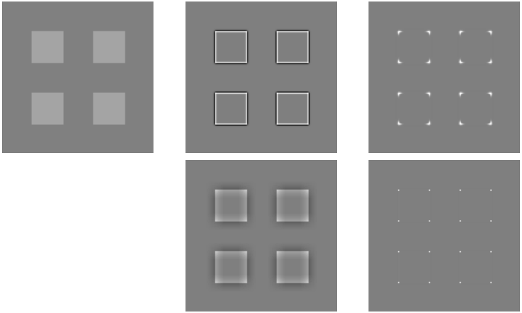
Method noise of the different denoising methods on a simple geometrical image.



Parameters are fixed in order to remove exactly an energy σ^2 ($\sigma = 2.5$).

Evaluation: Method noise

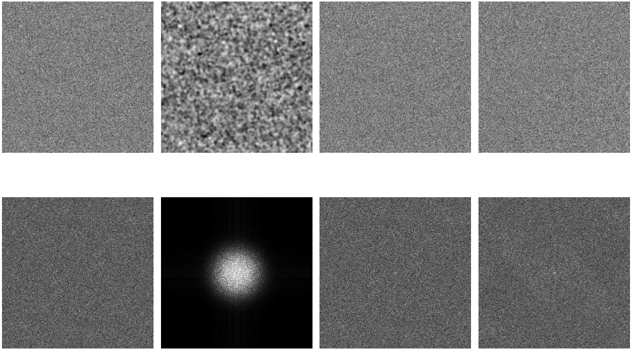
Method noise of the different denoising methods on a simple geometrical image.



Same parameters applied with the noise free image.

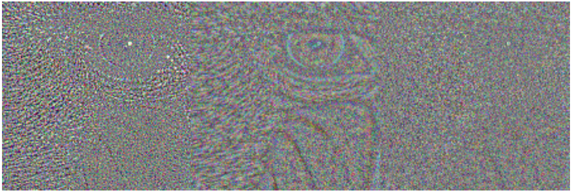
Evaluation: Noise to noise

The transformation of a white noise into any correlated signal creates structure and artifacts.



Evaluation: Visual quality

- Restored images and removed noise by the anisotropic filter, the neighborhood filter and the NL-means.



Evaluation: Visual quality

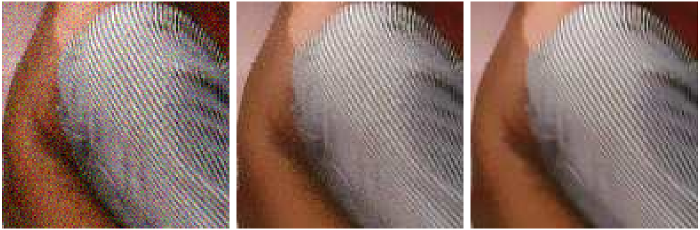
- Restored images and removed noise by the Gaussian filter, the anisotropic filter, the neighborhood filter and the NL-means.





Evaluation: Visual quality

- Restored images and removed noise by the neighborhood filter and the NL-means.



Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

Noise estimation

Photography II

NLmeans + Transform domain methods

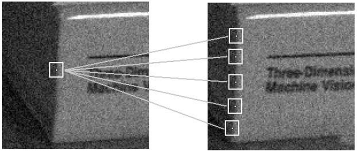
Extension to films

- More samples to average but inconvenient of motion.
- All state of the art movie filters are motion compensated. Motion is explicitly estimated and motion compensated movie yields a new stationary data on which an average filter is applied.
- Static vs Motion compensated neighborhood filter with a OFC based algorithm.

The details are better preserved and the boundaries less blurred with motion compensation.

Extension to films

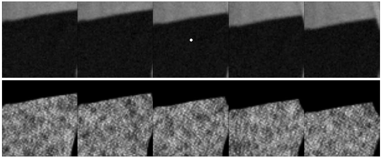
- One of the major difficulties in motion estimation is the ambiguity of trajectories, the so called *aperture problem*.

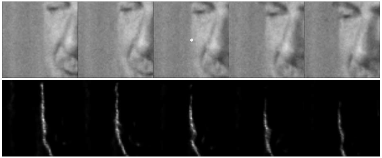


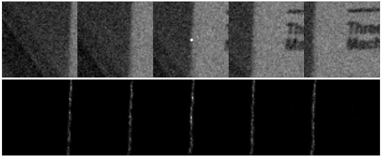
At most pixels, there are several options for the displacement vector. Motion estimate algorithms have to select one by some additional criterion thus loosing many interesting candidates.

- The NL-means simply looks for the resembling pixels, no matter where they lie in the movie.

Probability distributions in movement



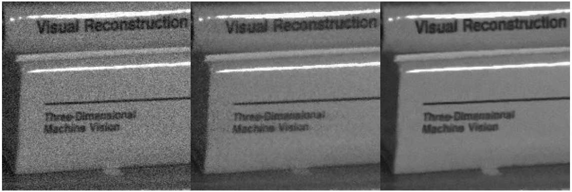


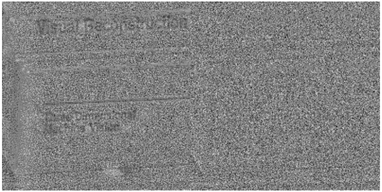


The algorithm looks for the pixels with a more similar configuration even they have moved (movie).

Comparison

- Comparison experiment between the motion compensated neighborhood filter and the NL-means.





- Dr. Mabuse sequence.

Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

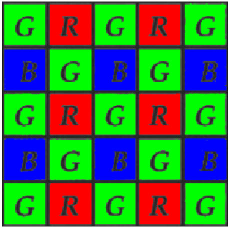
Noise estimation

Photography II

NLmeans + Transform domain methods

Photography Denoising I: CFA

- Photon counting process, obscurity noise, quantification, ... approximated by an additive signal dependent white noise with variance $a + bu$.
- Color filter array

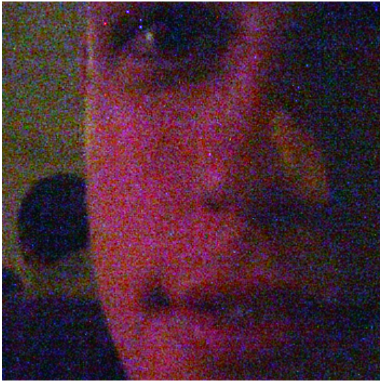


Photography Denoising I: CFA

- Noise at CCD sensors is approximately white and additive but signal dependent.

Photography Denoising I: CFA

- Noise after white balance, demosaicking, color correction, gamma correction and compression.



Photography Denoising I: CFA

Let $f(x)$ be the CFA output and $x \in \Omega_{\boldsymbol{u}}$,

$$NL[f](x) = \frac{1}{C_u(x)} \int_{\Omega_{\boldsymbol{u}} \cap B(x,t)} e^{-d(x,y)/h(x)} \, f(y) \, dy, \tag{1}$$

with $u \in \{r, g, g', b\}$.

The red and blue pixels can be compared with all red and blue pixels, while green pixels will be compared only to green pixels in the same CFA position (g or g'). For each point x, the non-local denoising algorithm averages pixels of the same channel with a similar neighborhood in $f(x)$.

The value of the filtering parameter h depends on the noise standard deviation at x and it is set taking also into account the white balance and tone curve.

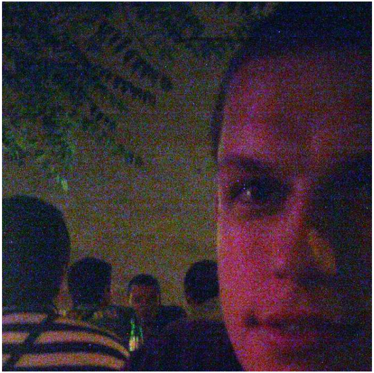
$$h(x) = k \cdot wb_u \cdot std_u(f(x)) \cdot TC'(y)$$

for $y = wb_u \cdot f(x)$, $x \in \Omega_u$ and where $TC'(\cdot)$ denotes the derivative of the tone curve function.

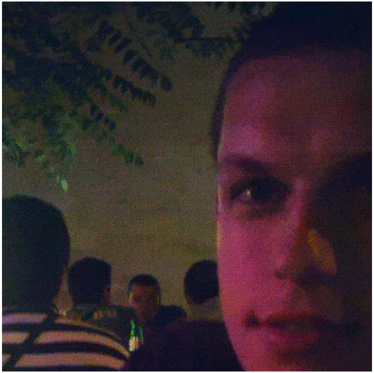
Phography Denoising I: CFA

Phography Denoising I: CFA

Phography Denoising I: CFA



Phography Denoising I: CFA



Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

Noise estimation

Photography II

NLmeans + Transform domain methods

Literature: Uniform white noise

The most part of the literature applies only to additive and white signal independent noise.

- Median of absolute derivatives.
- Median of wavelet coefficients at finest scale or DCT high frequency coefficients.
- Median of variance of small patches of derivative image.
- Actually it works better by using large patches and a small percentile ($p = 1\%$).

Literature: Signal dependent white noise

- Divide the range adaptively taking into account the grey level histogram of the image, in such a way, each bin contains the same number of samples.

	der	robust der	var der w small	var der w large
Uniform	1.81	2.87	1.58	0.75
Adaptive	1.66	1.87	1.36	0.73

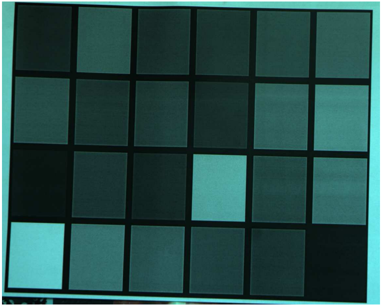
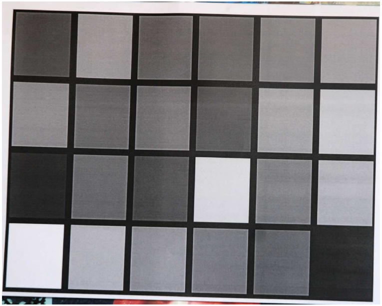
Signal dependent noise of $\sigma = \sqrt{8 + 2u}$ is added to 100 images and algorithms are applied with a uniform and adaptive splitting of the grey level range

$\rightarrow$ Signal dependent noise can be estimated, at least in simulated tests.

$\rightarrow$ In order to compare in real images we need a ground truth.

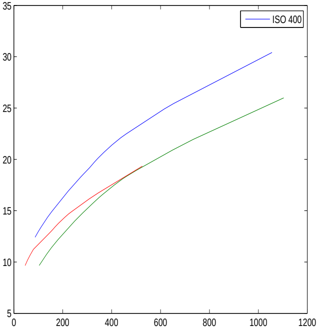
Ground truth estimation on real data

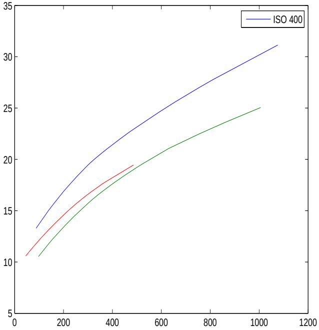
- Fix the camera and take a burst of images.
- Compute temporal average and standard deviation.
- Divide the gray level range adaptively into n bins and compute the median of standard deviations inside each bin.

Testing on raw data (ISO 400)

Comparison of "ground truth" and single image noise estimation (w=15x15, p=0.005):

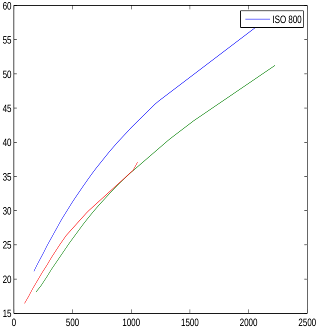


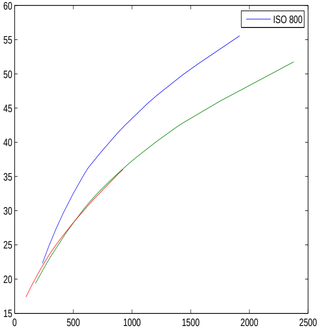


Blue channel is noisier than the red and green channels.

Testing on raw data (ISO 800)

Comparison of "ground truth" and single image noise estimation (w=15x15, p=0.005):

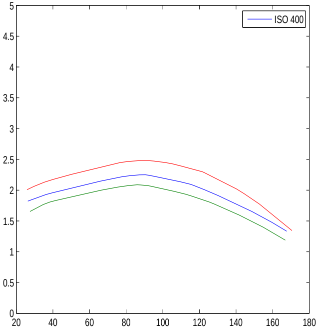


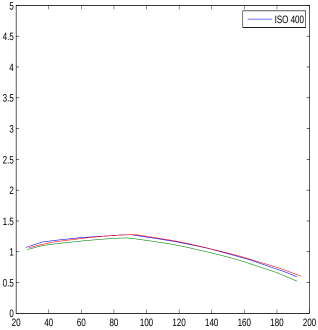


Blue channel is noisier than the red and green channels.

Testing on jpeg data (ISO 400)

Comparison of "ground truth" and single image noise estimation (w=15x15, p=0.005):

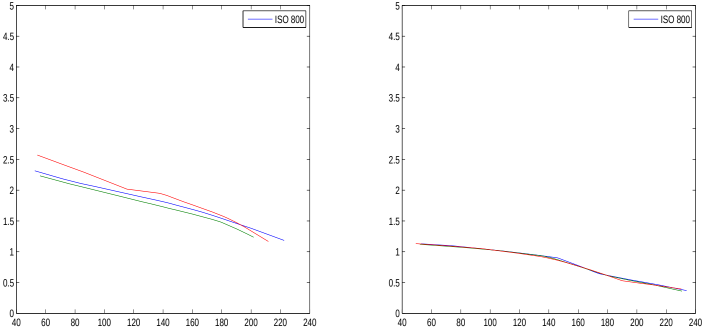




Red channel gets noisier than the blue one because of white balance.

Testing on jpeg data (ISO 800)

Comparison of "ground truth" and single image noise estimation (w=15x15, p=0.005):



Red channel gets noisier than the blue one because of white balance.

Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

Noise estimation

Photography II

NLmeans + Transform domain methods

Phograpy Denoising II: Final image

- Correlated noise and artifacts.
- Estimated standard deviation is not realist. Image is hardly modified.

Phography Denoising II: Final image

A classical multiresolution decomposition of u and u_2 is applied and the resulting images are filtered by the NL-means algorithm with noise estimation at each scale.

function out = multiresolution (int &i, Image input) {
if $i <$niterations **then**
  sampled $\leftarrow$ input $\downarrow 2$
  difference $\leftarrow$ input - sampled $\uparrow 2$
  i++;
  aux $\leftarrow$ multiresolution(i, sampled);
  input $\leftarrow$ aux $\uparrow 2$ + difference;
end if
estimatenoise(input);
out = denoise(input);
}

Phography Denoising II: Final image

Outline

Noise. Averaging

Local smoothing filters

Image autosimilarity. NLmeans

Movie denoising

Photography

Noise estimation

Photography II

NLmeans + Transform domain methods

Wavelet thresholding

- Let $\mathcal{B} = \{\psi_{j,k}\}_{(j,k)}$ be an orthonormal wavelets basis,

$$HWT = \sum_{\{(j,k) \mid |\langle v, \psi_{j,k} \rangle| > \tau\}} \langle v, \psi_{j,k} \rangle \psi_{j,k}$$

  The procedure is based on the idea that the image is represented with a small set of large wavelet coefficients while noise is distributed across small coefficients.

- The noise reduction is assured by the cancelation of degraded coefficients mainly due to noise. τ is taken over the maximum of noise coefficients $|\langle n, \psi_{j,k} \rangle|$.

- Consequences:
  - Gibbs phenomenon due to cancelation of coefficients near edges.
  - Spurious wavelets seen in the image

Wavelet thresholding

Hybrid methods: BM3D and PCA

PCA For each block

- Find similar blocks
- Construct adapted basis by Principal Component Analysis.
- Perform a thresholding in this basis.

BM3D For each block

- Find similar blocks
- Construct a 3D block and use 3D DCT transform.
- Perform a thresholding in this basis.

Hybrid methods: BM3D and PCA

Hybrid methods: BM3D and PCA

Hybrid methods: BM3D and PCA