



TAMPERE UNIVERSITY OF TECHNOLOGY

Department of Signal Processing – Transforms and Spectral Methods Group

1

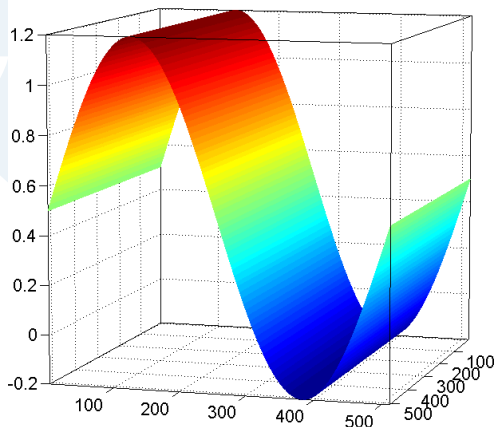
Practical denoising of clipped or overexposed noisy images

Alessandro Foi

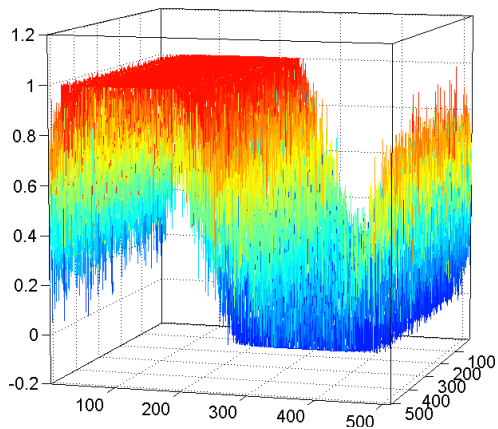
www.cs.tut.fi/~foi

Clipped noisy data

2



original



added noise and then clipped

What is proposed?

Complete framework for the filtering of clipped observations with signal-dependent noise

Ingredients:

A good (transform-based) denoising algorithm for AWGN

Four nonlinear homomorphic transformations:

- 1) variance-stabilizing transformation [and its 3) inverse]
- 2) debiasing
- 4) declipping

A decorative graphic on the left side of the slide. It features a light blue gear partially visible on the left edge. To the left of the gear are two vertical bars: a light blue one on top and a yellow one on the bottom.

Preliminaries

Modeling clipped signal-dependent observations

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

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

$y : X \rightarrow Y \subseteq \mathbb{R}$ unknown original image (deterministic)

$\sigma(y(x)) \xi(x)$ zero-mean random error

$\sigma : \mathbb{R} \rightarrow \mathbb{R}^+$ standard-deviation function (deterministic)

$\xi(x)$ random variable $E \{\xi(x)\} = 0$ $\text{var} \{\xi(x)\} = 1$

$y(x) = E \{z(x)\}$ expectation

$\sigma(y(x)) = \text{std} \{z(x)\}$ standard deviation

Modeling clipped signal-dependent observations

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

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

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

$$\tilde{y}(x) = E\{\tilde{z}(x)\} \quad \text{expectation}$$

$$\tilde{\sigma} : [0, 1] \rightarrow \mathbb{R}^+ \quad \text{standard-deviation function (of expectation)}$$

$$\tilde{\sigma}(\tilde{y}(x)) = \text{std} \{\tilde{z}(x)\} \quad \text{standard deviation}$$

Modeling clipped signal-dependent observations

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

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

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

$$\tilde{y}(x) = E\{\tilde{z}(x)\} \quad \text{expectation}$$

$$\tilde{\sigma} : [0, 1] \rightarrow \mathbb{R}^+ \quad \text{standard-deviation function (of expectation)}$$

$$\tilde{\sigma}(\tilde{y}(x)) = \text{std} \{\tilde{z}(x)\} \quad \text{standard deviation}$$

Goal: estimate y from \tilde{z}

Modeling raw-data

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

$$\sigma(y(x)) \xi(x) = \eta_p(y(x)) + \eta_g(x)$$

η_p *Poissonian* signal-dependent component

η_g *Gaussian* signal-independent component

$$\begin{aligned} (y(x) + \eta_p(y(x)))\chi &\sim \mathcal{P}(\chi y(x)), & \chi > 0 \\ \eta_g(x) &\sim \mathcal{N}(0, b), & b > 0 \end{aligned}$$

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

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

(Generalized) Probability distributions

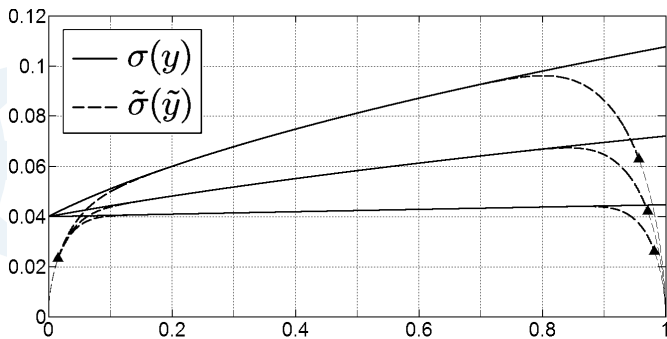
Before clipping:

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

After clipping:

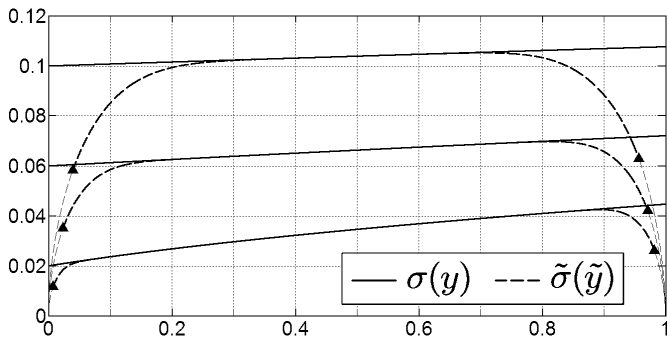
$$\varphi_z(\zeta) = \Phi\left(\frac{-y}{\sigma(y)}\right) \delta_0(\zeta) + \frac{1}{\sigma(y)} \phi\left(\frac{\zeta - y}{\sigma(y)}\right) \chi_{[0,1]} + \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right) \delta_0(1 - \zeta)$$

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



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

$$b = 0.04^2$$



$$a = 0.04^2$$

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

Clipped \Leftrightarrow non-clipped

Direct transformations (non-clipped to clipped)

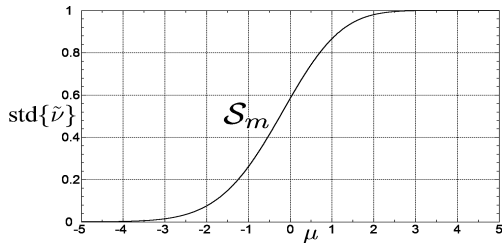
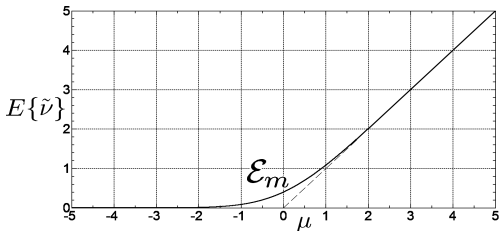
$$\tilde{y} = \mathcal{A}(y, \sigma(y)) = \sigma(y) \mathcal{E}_m\left(\frac{y}{\sigma(y)}\right) - y + 1 - \sigma(y) \mathcal{E}_m\left(\frac{1-y}{\sigma(y)}\right)$$

$$\tilde{\sigma}(\tilde{y}) = \mathcal{B}(y, \sigma(y)) = \sigma(y) \mathcal{S}_m\left(\frac{y}{\sigma(y)}\right) \mathcal{S}_m\left(\frac{1-y}{\sigma(y)}\right)$$

where

$$\mathcal{E}_m(\mu) = \Phi(\mu) \mu + \phi(\mu)$$

$$\mathcal{S}_m^2(\mu) = \Phi(\mu) + \mathcal{E}_m(\mu) \mu - \mathcal{E}_m^2(\mu)$$



Clipped \Leftrightarrow non-clipped

Inverse transformations (clipped to non-clipped)

$$y = \mathcal{C}(\tilde{y}, \tilde{\sigma}(\tilde{y})) = \tilde{y} \mathcal{E}_r\left(\frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right) - \tilde{y} + 1 - (1 - \tilde{y}) \mathcal{E}_r\left(\frac{1-\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right)$$

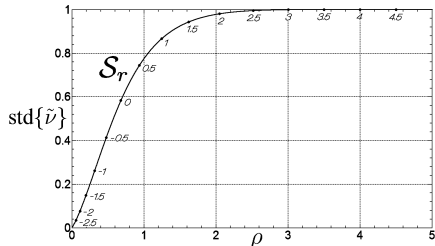
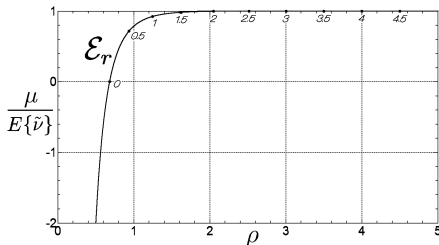
$$\sigma(y) = \mathcal{D}(\tilde{y}, \tilde{\sigma}(\tilde{y})) = \frac{\tilde{\sigma}(\tilde{y})}{\mathcal{S}_r\left(\frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right) \mathcal{S}_r\left(\frac{1-\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right)}$$

where

$$\mathcal{E}_r(\rho) = \mu / \mathcal{E}_m(\mu)$$

$$\mathcal{S}_r(\rho) = \mathcal{S}_m(\mu)$$

$$\rho = \frac{\mathcal{E}_m(\mu)}{\mathcal{S}_m(\mu)}$$



A decorative graphic on the left side of the slide. It features a light blue gear partially visible on the left edge. To the left of the gear are two vertical bars: a light blue one on top and a yellow one on the bottom.

Algorithm

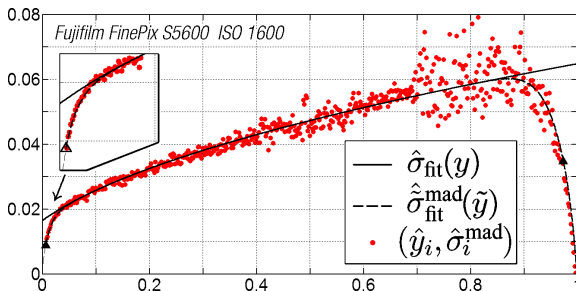
Algorithm: Step 0, Noise estimation

14

If σ (or $\tilde{\sigma}$) is known, go ahead.

Otherwise, estimate noise.

Estimation is possible provided that σ is described by few parameters (e.g., a and b for the raw-data), using algorithm in [Foi et al., IEEE TIP, Oct. 2008].



Algorithm: Step 1, Variance stabilization

15

Idea: transform \tilde{z} to $f(\tilde{z})$ such that $\text{var}\{f(\tilde{z})\}$ is approximately constant.

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

Main result: analytical proof that the integral does not diverge (i.e. f is bounded) if $b \neq 0, -a$ (i.e., for all cases of practical interest).

Algorithm: Step 2, Denoising

Denoise $f(\tilde{z})$ assuming constant variance c^2 .

$$\mathbf{D}_{\text{ho}}(f(\tilde{z})) \approx E\{f(\tilde{z})\}$$

We use BM3D algorithm [Dabov et al., IEEE TIP, Aug. 2007].



Algorithm: Step 3, Debiasing

17

$$\mathbf{D}_{\text{ho}}(f(\tilde{z})) \approx E\{f(\tilde{z})\} \neq f(E\{\tilde{z}\})$$

Algorithm: Step 3, Debiasing

$$\mathbf{D}_{\text{ho}}(f(\tilde{z})) \approx E\{f(\tilde{z})\} \neq f(E\{\tilde{z}\})$$

$$E\{f(\tilde{z})\} = \int_0^1 f(\zeta) \wp_{\tilde{z}}(\zeta) d\zeta = \int_0^1 f(\zeta) \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) d\zeta + f(1) \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right)$$

$$f(E\{\tilde{z}\}) = f\left(\int_0^1 \zeta \wp_{\tilde{z}}(\zeta) d\zeta\right) = f\left(\int_0^1 \zeta \left(\Phi\left(\frac{-y}{\sigma(y)}\right) \delta_0(\zeta) + \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) \chi_{[0,1]} + \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right) \delta_0(1-\zeta)\right) d\zeta\right)$$

Algorithm: Step 3, Debiasing

$$\mathbf{D}_{\text{ho}}(f(\tilde{z})) \approx E\{f(\tilde{z})\} \neq f(E\{\tilde{z}\})$$

$$E\{f(\tilde{z})\} = \int_0^1 f(\zeta) \wp_{\tilde{z}}(\zeta) d\zeta = \int_0^1 f(\zeta) \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) d\zeta + f(1) \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right)$$

$$f(E\{\tilde{z}\}) = f\left(\int_0^1 \zeta \wp_{\tilde{z}}(\zeta) d\zeta\right) = f\left(\int_0^1 \zeta \left(\Phi\left(\frac{-y}{\sigma(y)}\right) \delta_0(\zeta) + \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) \chi_{[0,1]} + \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right) \delta_0(1-\zeta)\right) d\zeta\right)$$

Let $h : [0, f(1)] \rightarrow [0, f(1)]$ be the function defined (implicitly varying y) by

$$f(E\{\tilde{z}\}) \xrightarrow{h} E\{f(\tilde{z})\}$$

Algorithm: Step 3, Debiasing

$$\mathbf{D}_{\text{ho}}(f(\tilde{z})) \approx E\{f(\tilde{z})\} \neq f(E\{\tilde{z}\})$$

$$E\{f(\tilde{z})\} = \int_0^1 f(\zeta) \varphi_{\tilde{z}}(\zeta) d\zeta = \int_0^1 f(\zeta) \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) d\zeta + f(1) \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right)$$

$$f(E\{\tilde{z}\}) = f\left(\int_0^1 \zeta \varphi_{\tilde{z}}(\zeta) d\zeta\right) = f\left(\int_0^1 \zeta \left(\Phi\left(\frac{-y}{\sigma(y)}\right) \delta_0(\zeta) + \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) \chi_{[0,1]} + \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right) \delta_0(1-\zeta)\right) d\zeta\right)$$

Let $h : [0, f(1)] \rightarrow [0, f(1)]$ be the function defined (implicitly varying y) by

$$f(E\{\tilde{z}\}) \stackrel{h}{\mapsto} E\{f(\tilde{z})\}$$

Debiasing

$$h^{-1}(\mathbf{D}_{\text{ho}}(f(\tilde{z}))) \approx f(E\{\tilde{z}\})$$

Algorithm: Step 3, Debiasing

$$\mathbf{D}_{\text{ho}}(f(\tilde{z})) \approx E\{f(\tilde{z})\} \neq f(E\{\tilde{z}\})$$

$$E\{f(\tilde{z})\} = \int_0^1 f(\zeta) \wp_{\tilde{z}}(\zeta) d\zeta = \int_0^1 f(\zeta) \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) d\zeta + f(1) \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right)$$

$$f(E\{\tilde{z}\}) = f\left(\int_0^1 \zeta \wp_{\tilde{z}}(\zeta) d\zeta\right) = f\left(\int_0^1 \zeta \left(\Phi\left(\frac{-y}{\sigma(y)}\right) \delta_0(\zeta) + \frac{1}{\sigma(y)} \phi\left(\frac{\zeta-y}{\sigma(y)}\right) \chi_{[0,1]} + \left(1 - \Phi\left(\frac{1-y}{\sigma(y)}\right)\right) \delta_0(1 - \zeta)\right) d\zeta\right)$$

Let $h : [0, f(1)] \rightarrow [0, f(1)]$ be the function defined (implicitly varying y) by

$$f(E\{\tilde{z}\}) \stackrel{h}{\mapsto} E\{f(\tilde{z})\}$$

Debiasing

$$h^{-1}(\mathbf{D}_{\text{ho}}(f(\tilde{z}))) \approx f(E\{\tilde{z}\})$$

Caution: in image processing literature it is always assumed $E\{f(\tilde{z})\} = f(E\{\tilde{z}\})$

Arsenault and Denis, "Integral expression for transforming signal-dependent noise..", *Optics Letters*, 1981.

Prucnal and Saleh, "Transformation of image-signal-dependent noise into...", *Optics Letters*, 1981.

Kasturi, Walkup, and Krile, "Image restoration by transformation of signal-dependent noise...", *App. Optics*, 1983.

Hirakawa and Parks, "Image denoising using total least squares", *IEEE Trans. Image Process.*, 2006.

Algorithm: Step 4, Inversion of f

22

$$f^{-1}(h^{-1}(\mathbf{D}_{\text{ho}}(f(\tilde{z})))) \approx f^{-1}(f(E\{\tilde{z}\})) = E\{\tilde{z}\}$$

This provides an estimate \hat{y} of \tilde{y}

Algorithm: Step 4, Inversion of f

23

$$f^{-1}(h^{-1}(\mathbf{D}_{\text{ho}}(f(\tilde{z})))) \approx f^{-1}(f(E\{\tilde{z}\})) = E\{\tilde{z}\}$$

Algorithm: Step 5, Declipping

Recall $y = \mathcal{C}(\tilde{y}, \tilde{\sigma}(\tilde{y}))$ and define $\hat{y} = f^{-1}(h^{-1}(\mathbf{D}_{\text{ho}}(f(\tilde{z}))))$

Declipping

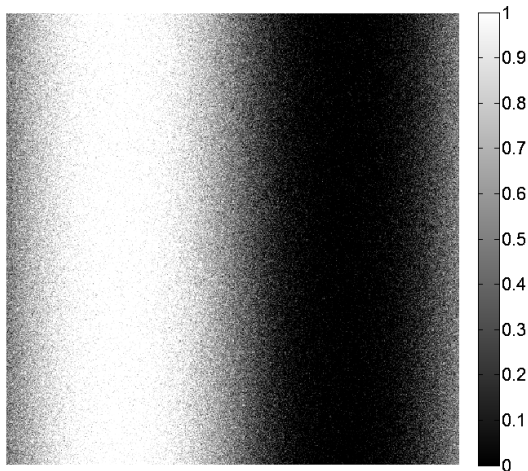
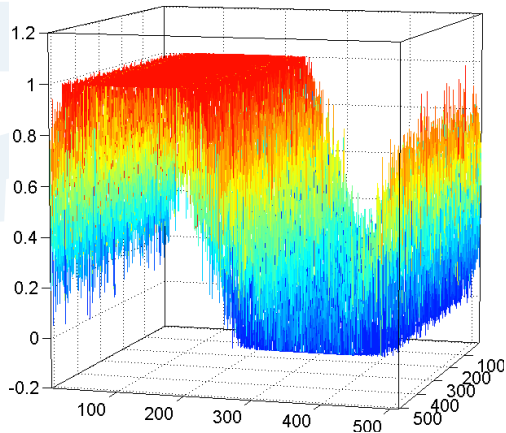
$$\hat{y} = \mathcal{C}(\hat{y}, \tilde{\sigma}(\hat{y}))$$

A decorative graphic on the left side of the slide. It features a light blue gear partially visible on the left edge. To the left of the gear are two vertical bars: a light blue one on top and a yellow one on the bottom.

Experiments

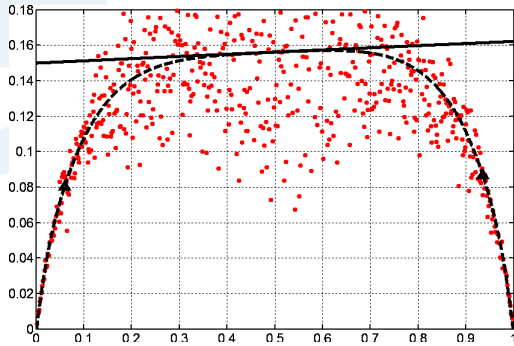
Observations \tilde{z}

25

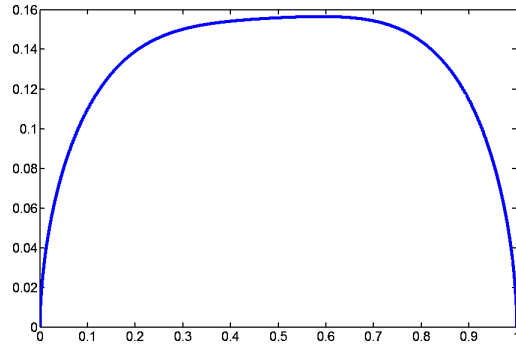


$$y(x_1, x_2) = 0.7 \sin\left(\frac{2\pi}{512}x_1\right) + 0.5$$

Noise estimation

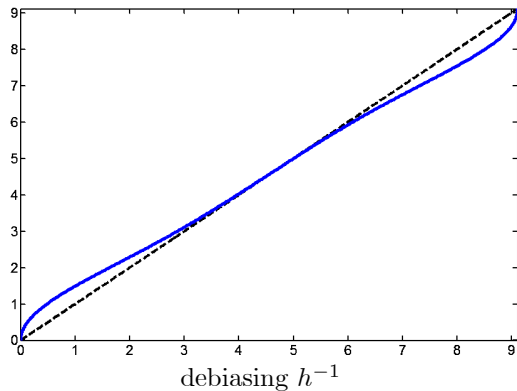
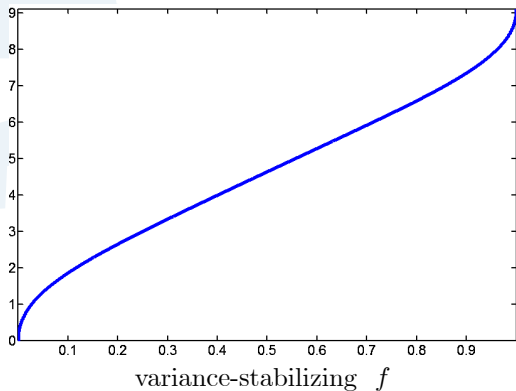


estimation and fitting $\hat{a} = 0.0038$ $\hat{b} = 0.022$

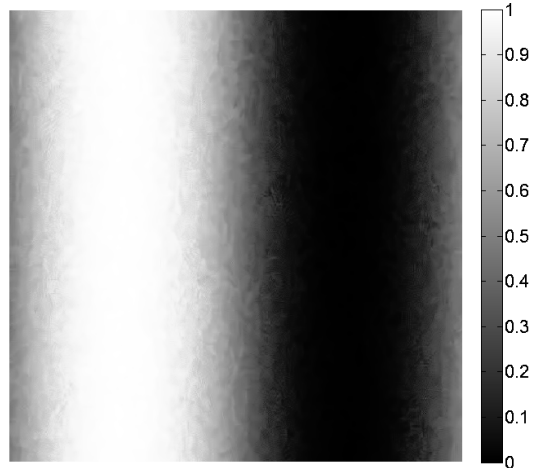
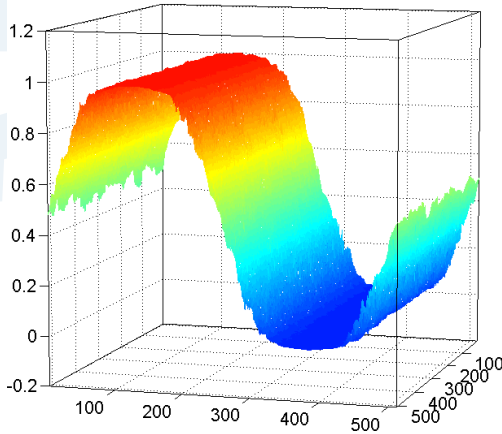


st.dev.-function $\tilde{\sigma}$

f and h^{-1}

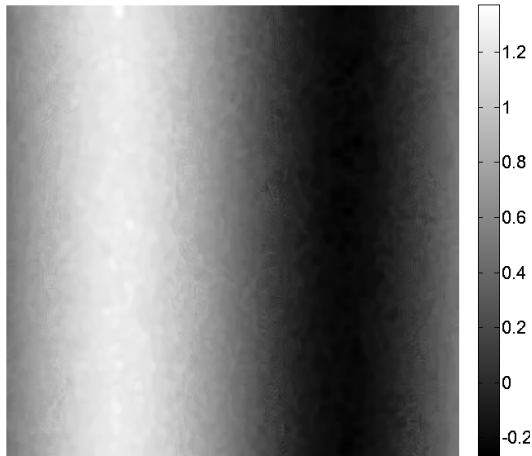
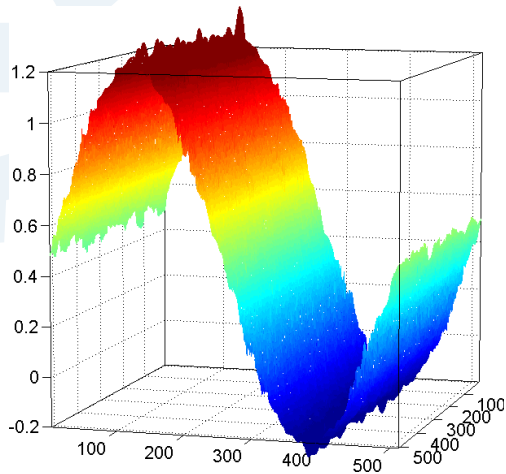


Denoised estimate before declipping



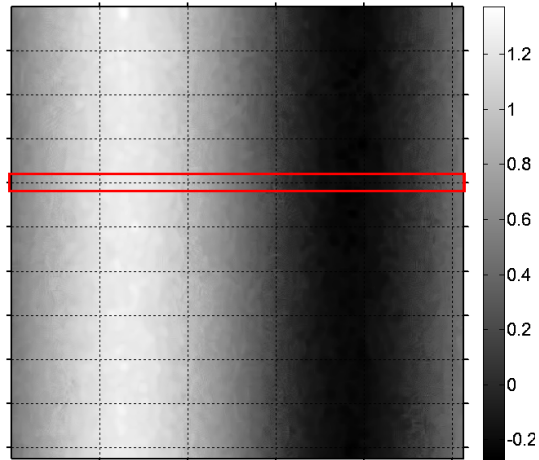
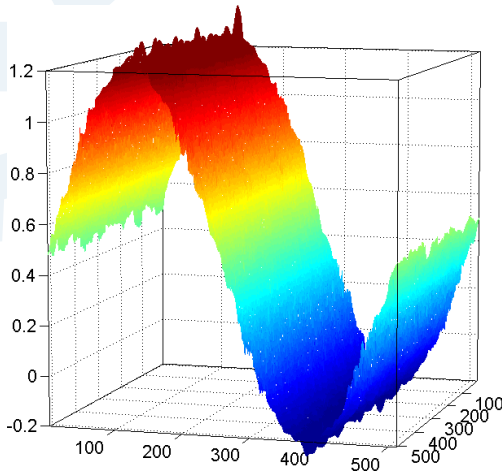
Declipped estimate

29

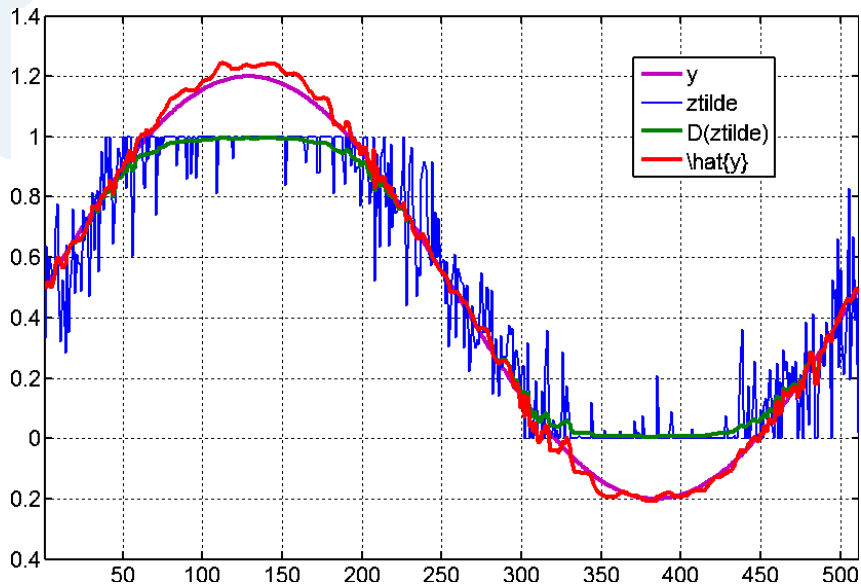


Declipped estimate

30



Cross section

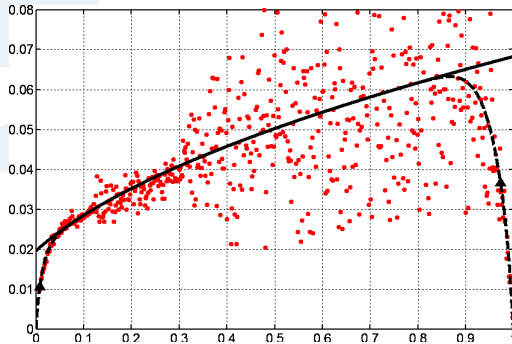


Observations \tilde{z}

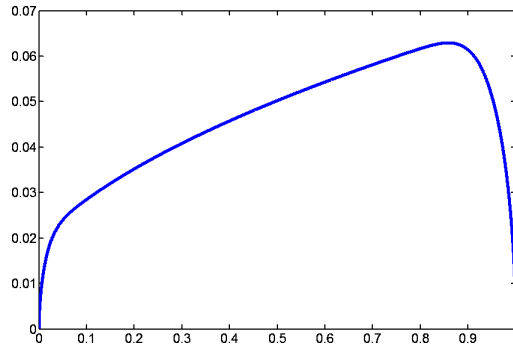


Raw-data from Fujifilm FinePix S9600, ISO 1600

Noise estimation

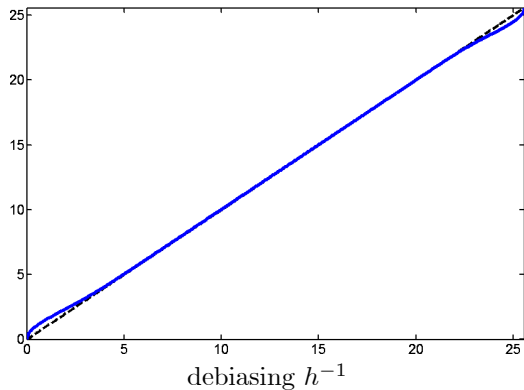
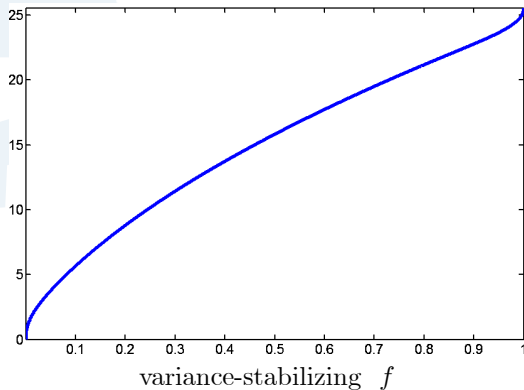


estimation and fitting $\hat{a} = 0.0043$ $\hat{b} = 0.00038$



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

f and h^{-1}



Denoised estimate before declipping

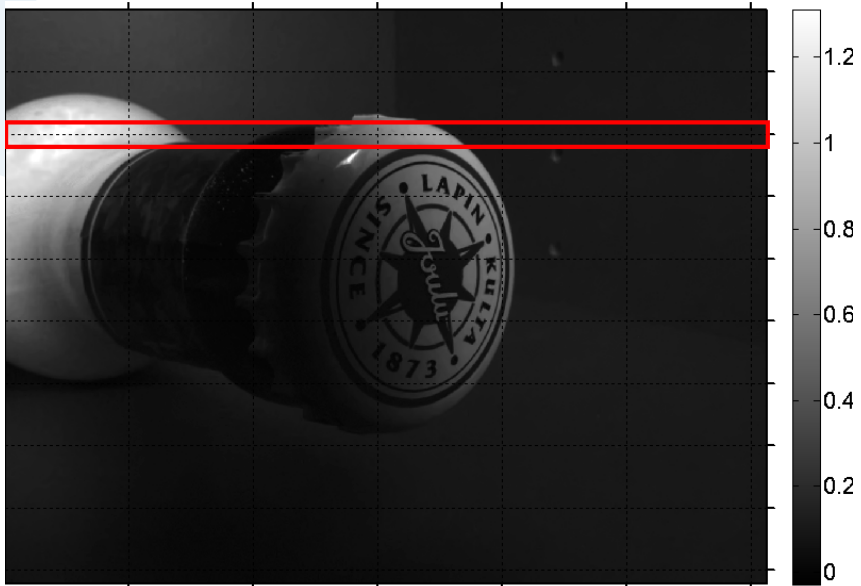
35



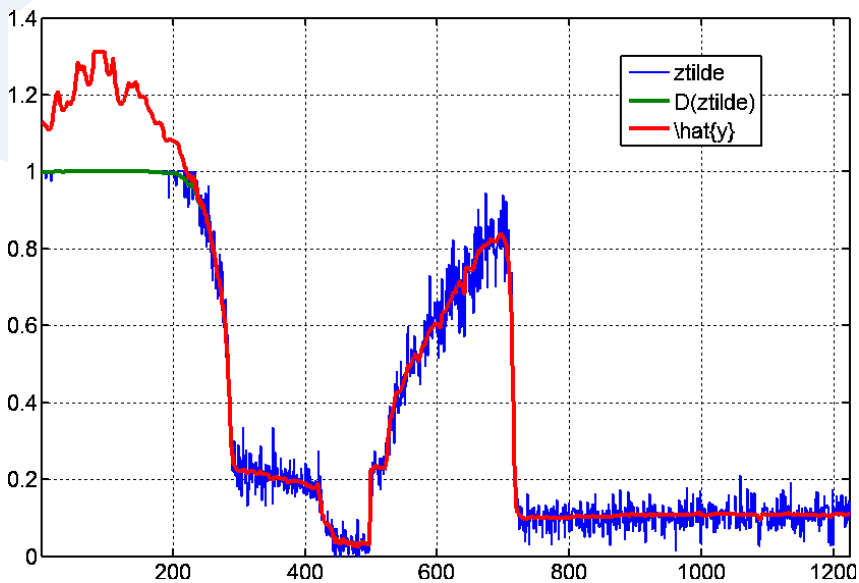
Declipped estimate



Declipped estimate



Cross section



Noise estimation

A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data”, to appear in *IEEE Trans. Image Process.* (October 2008)


Multiframe raw-data

G. Boracchi and A. Foi, “Multiframe raw-data denoising based on block-matching and 3-D filtering for low-light imaging and stabilization”, *Proc. LNLA 2008*, August 2008.

Matlab software


40

www.cs.tut.fi/~foi/sensornoise.html



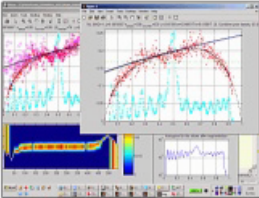
Matlab software Help

ClipPoisGaus_stdEst2D
Poissonian-Gaussian noise estimation for single-image raw-data




ver. 1.12, released 17 June 2008
(for Matlab 7.0 or later)

Fully automatic estimation of noise parameters from a single image with clipped or non-clipped data corrupted by signal-dependent noise.



ClipPoisGaus_denoising
Denoising of clipped images (e.g., raw-data)



ver. 1.13, released 24 June 2008
(for Matlab 7.0 or later)

Fully automatic denoising and debiasing of clipped images with Poissonian-Gaussian noise using variance-stabilization and homoskedastic filtering.

