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

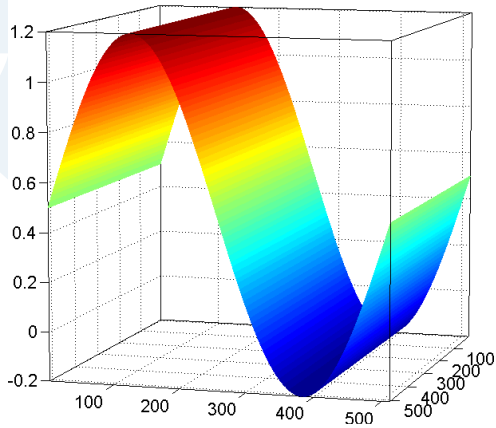
Practical denoising of clipped or overexposed noisy images

Alessandro Foi

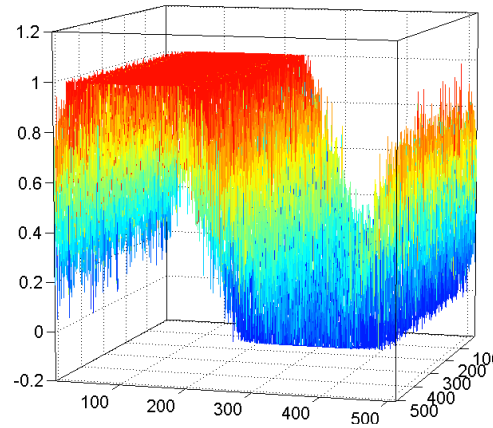
www.cs.tut.fi/~foi

EUSIPCO-2008 - Lausanne, Switzerland, 27.9.2008

Clipped noisy data



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

Preliminaries

Modeling 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

Modeling 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 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

Goal: estimate y from $\tilde{z}$

Modeling raw-data

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

η_{g} *Gaussian* signal-independent component

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

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

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

Foi et al., IEEE TIP, Oct. 2008.

(Generalized) Probability distributions

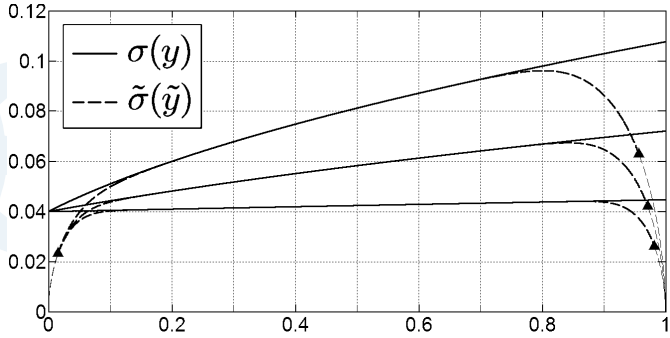
Before clipping:

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

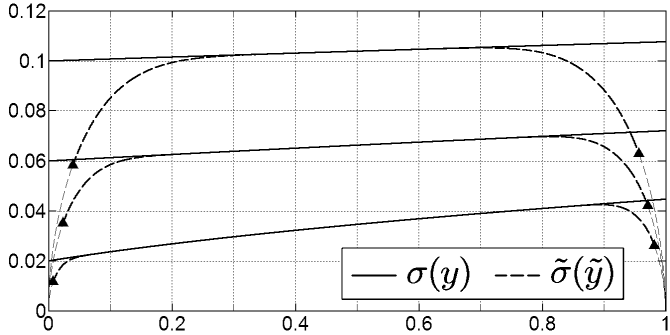
After clipping:

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

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 ⇆ non-clipped

Direct transformations (non-clipped to clipped)

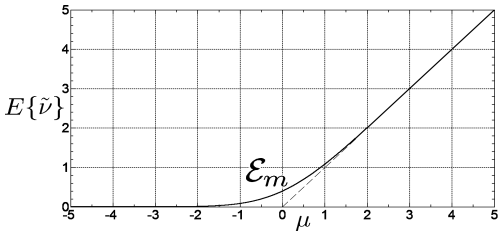
$$\tilde{y} = \mathcal{A}(y,\sigma(y)) = \sigma(y)\,\mathcal{E}_m\left(\tfrac{y}{\sigma(y)}\right) - y + 1 - \sigma(y)\,\mathcal{E}_m\left(\tfrac{1-y}{\sigma(y)}\right)$$

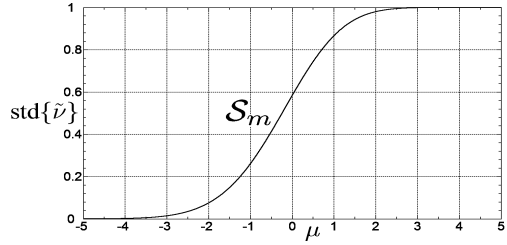
$$\tilde{\sigma}(\tilde{y}) = \mathcal{B}(y,\sigma(y)) = \sigma(y)\,\mathcal{S}_m\left(\tfrac{y}{\sigma(y)}\right)\mathcal{S}_m\left(\tfrac{1-y}{\sigma(y)}\right)$$

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 ⇆ non-clipped

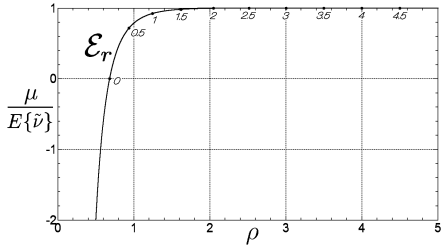
Inverse transformations (clipped to non-clipped)

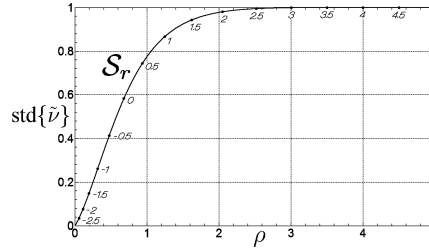
$$\boxed{y = \mathcal{C}(\tilde{y}, \tilde{\sigma}(\tilde{y})) = \tilde{y}\mathcal{E}_r\left(\frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right) - \tilde{y} + 1 - (1-\tilde{y})\,\mathcal{E}_r\left(\frac{1-\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right)}$$

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

where

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





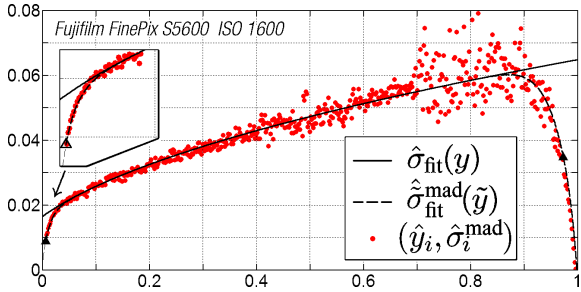
Algorithm

Algorithm: Step 0, Noise estimation

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

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

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

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.

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

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

Algorithm: Step 3, Debiasing

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

Algorithm: Step 3, Debiasing

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

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

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

Algorithm: Step 3, Debiasing

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

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

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

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

$$f(E\{\tilde{z}\}) \overset{h}{\longmapsto} E\{f(\tilde{z})\}$$

Algorithm: Step 3, Debiasing

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

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

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

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

$$f(E\{\tilde{z}\}) \overset{h}{\longmapsto} E\{f(\tilde{z})\}$$

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

Algorithm: Step 3, Debiasing

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

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

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

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

$$f(E\{\tilde{z}\}) \overset{h}{\longmapsto} E\{f(\tilde{z})\}$$

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

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

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

This provides an estimate $\widehat{\tilde{y}}$ of $\tilde{y}$

Algorithm: Step 4, Inversion of f

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

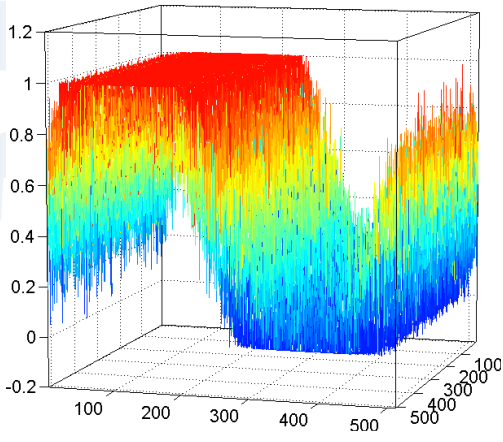
Algorithm: Step 5, Declipping

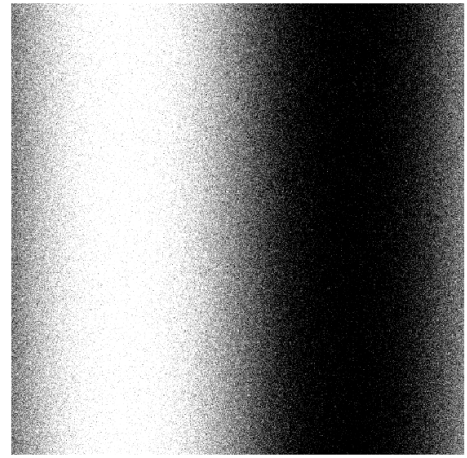
Recall $y = \mathcal{C}(\check{y}, \check{\sigma}(\check{y}))$ and define $\widehat{\check{y}} = f^{-1}\big(h^{-1}(\mathbf{D}_{\mathsf{ho}}(f(\tilde{z})))\big)$

Declipping $\boxed{\hat{y} = \mathcal{C}\left(\widehat{\check{y}}, \check{\sigma}\left(\widehat{\check{y}}\right)\right)}$

Experiments

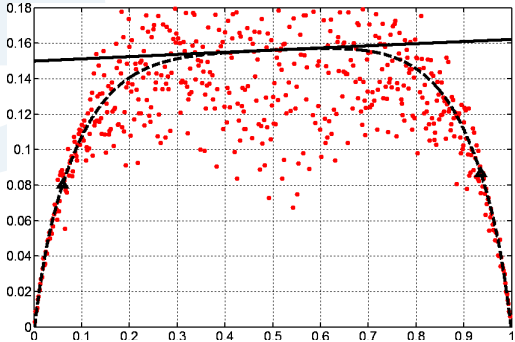
Observations $\tilde{z}$



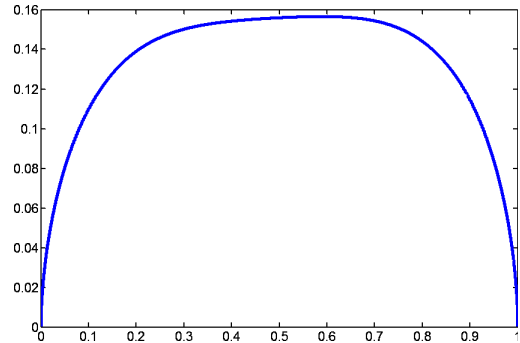


$$y\left(x_1, x_2\right) = 0.7 \sin\left(\frac{2\pi}{512} x_1\right) + 0.5$$

Noise estimation

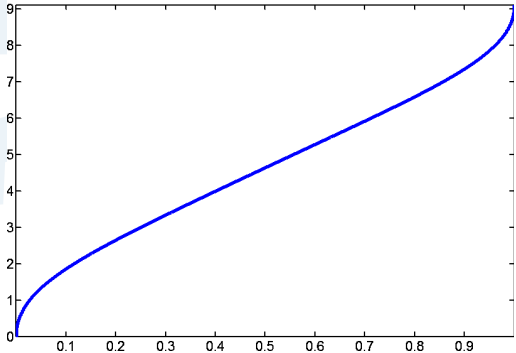


estimation and fitting $\hat{a} = 0.0038 \quad \hat{b} = 0.022$

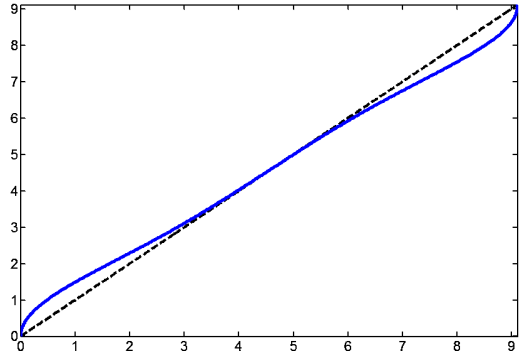


st.dev.-function $\tilde{\sigma}$

f **and** h^{-1}

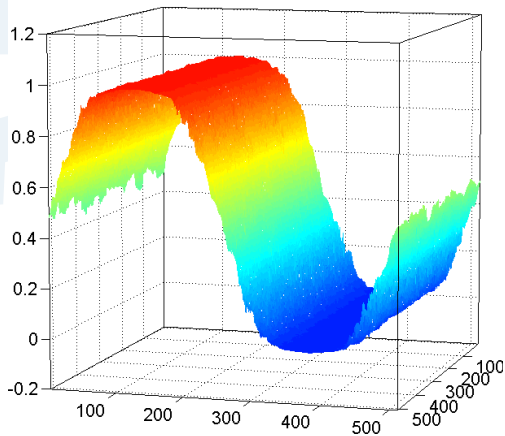


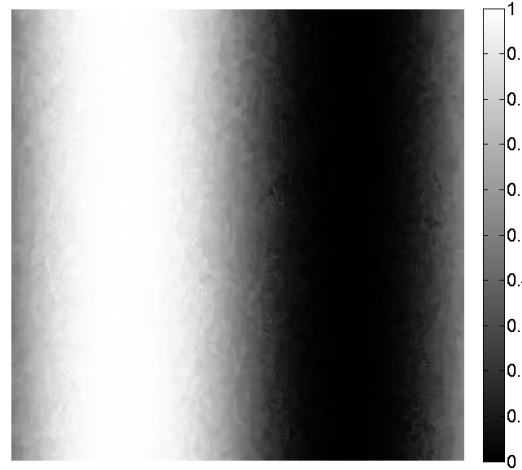
variance-stabilizing f



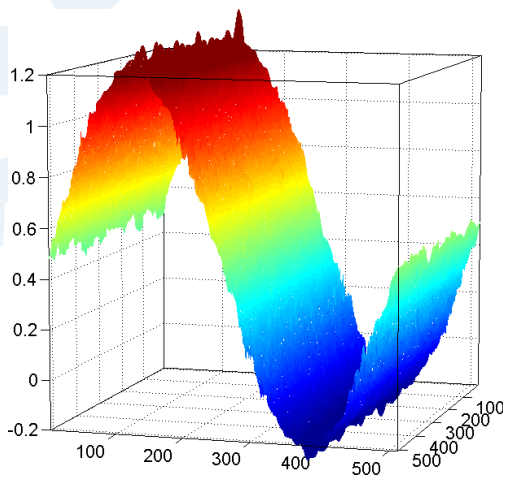
debiasing h^{-1}

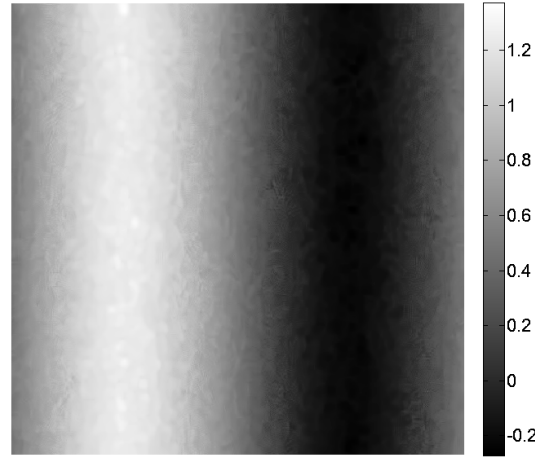
Denoised estimate before declipping



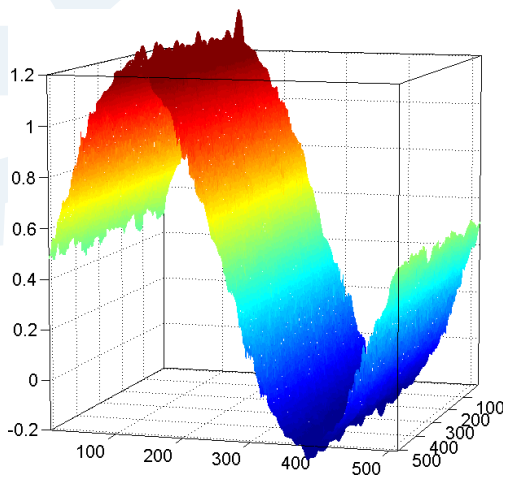


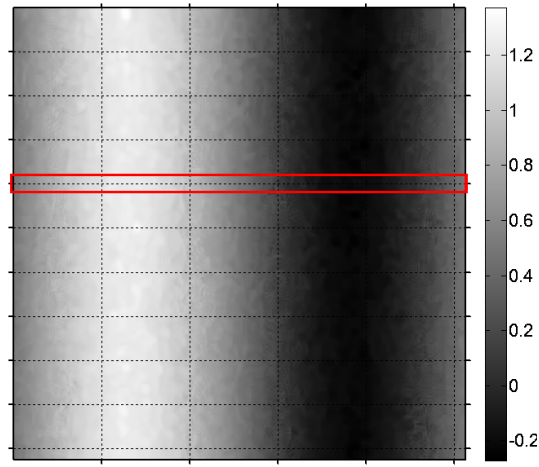
Declipped estimate



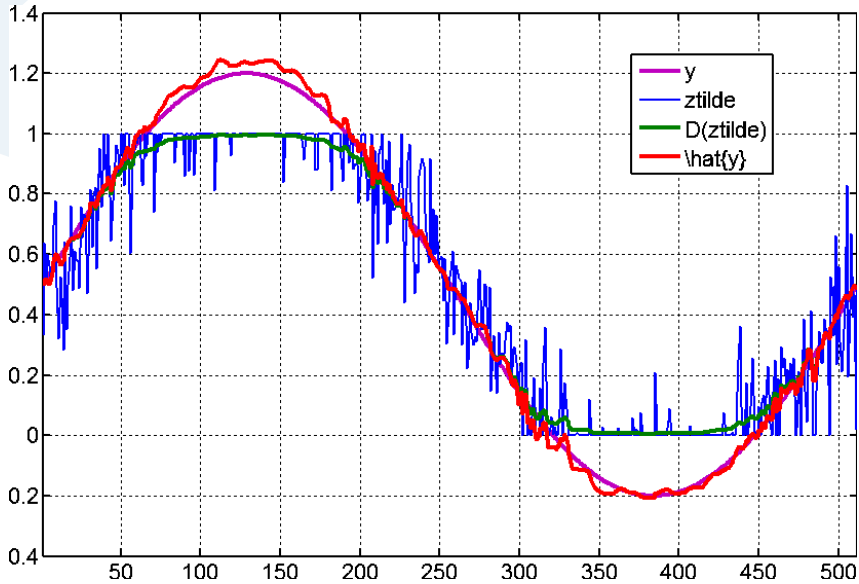


Declipped estimate





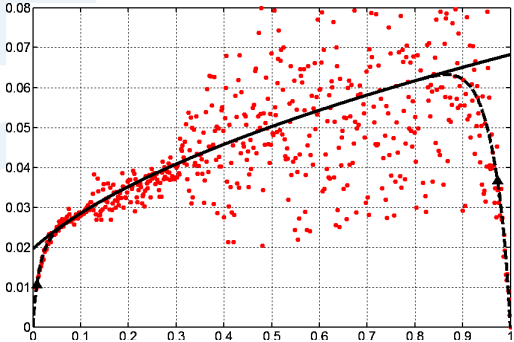
Cross section



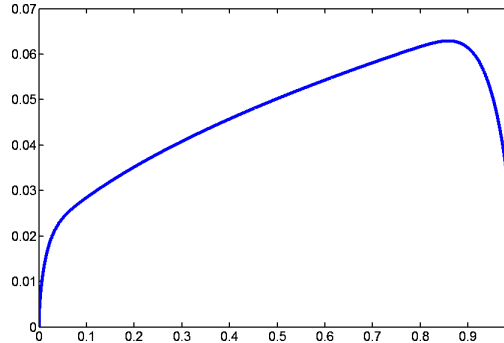
Observations $\tilde{z}$

Raw-data from Fujifilm FinePix S9600, ISO 1600

Noise estimation

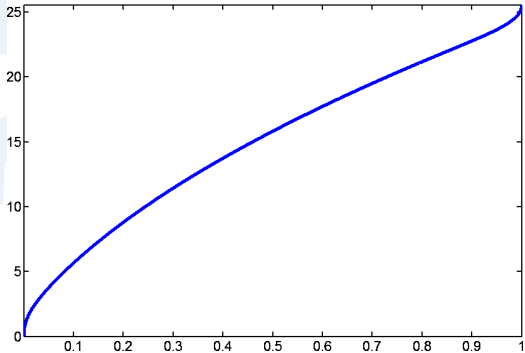


estimation and fitting $\hat{a} = 0.0043 \quad \hat{b} = 0.00038$

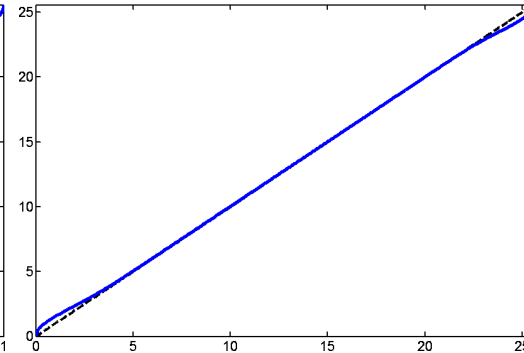


st.dev.-function $\tilde{\sigma}$

f **and** h^{-1}



variance-stabilizing f



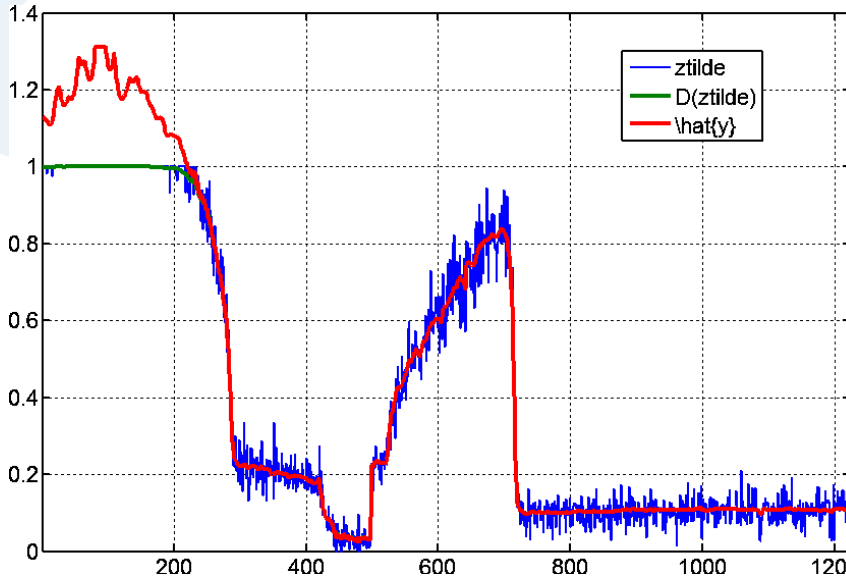
debiasing h^{-1}

Denoised estimate before declipping

Declipped estimate

Declipped estimate

Cross section



Related works

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

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