Tutorial T3

Signal-Dependent and Correlated Noise in Direct and Inverse Imaging:

From Modelling to Parameter Estimation and Practical Filtering

Alessandro Foi

TAMPERE UNIVERSITY OF TECHNOLOGY

Tampere University

SPCOM

International Conference on Signal Processing and Communications

July 19 - 23, 2020

Indian Institute of Science, Bangalore

Outline (1/2)

- Prelude: the i.i.d. additive white Gaussian noise model

Outline (1/2)

- Prelude: the i.i.d. additive white Gaussian noise model
- Signal-dependent noise models
  - one-parameter families of distributions;
  - heteroskedasticity; variance function, response function.

Outline (1/2)

- Prelude: the i.i.d. additive white Gaussian noise model
- Signal-dependent noise models
  - one-parameter families of distributions;
  - heteroskedasticity; variance function, response function.
- Estimation of noise-model parameters:
  - theoretical aspects;
  - methods for noise variance-mean curve estimation and fitting;
  - estimation under saturation and clipping.

Outline (1/2)

- Prelude: the i.i.d. additive white Gaussian noise model
- Signal-dependent noise models
  - one-parameter families of distributions;
  - heteroskedasticity; variance function, response function.
- Estimation of noise-model parameters:
  - theoretical aspects;
  - methods for noise variance-mean curve estimation and fitting;
  - estimation under saturation and clipping.
- Variance Stabilizing Transforms (VST)
  - heuristics;
  - theory: existence and finite vs. asymptotic properties;
  - construction and analysis of VST;

Outline (2/2)

- Inverse variance-stabilizing transformations:
  - asymptotic and exact unbiasedness;
  - optimal inverse transformations;

Outline (2/2)

- Inverse variance-stabilizing transformations:
  - asymptotic and exact unbiasedness;
  - optimal inverse transformations;
- Case studies:
  - Iterative VST Denoising
  - VST-based Deblurring (correlated noise)

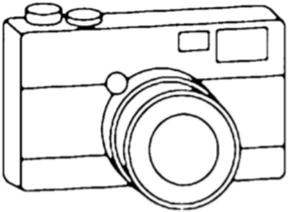
Outline (2/2)

- Inverse variance-stabilizing transformations:
  - asymptotic and exact unbiasedness;
  - optimal inverse transformations;
- Case studies:
  - Iterative VST Denoising
  - VST-based Deblurring (correlated noise)
- Discussion and Q&A.

Warm-up

A simple experiment

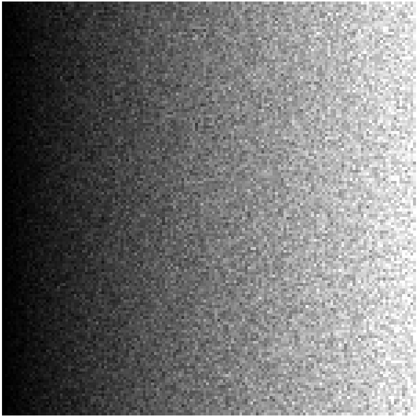
Take photos of a gray scale test ramp



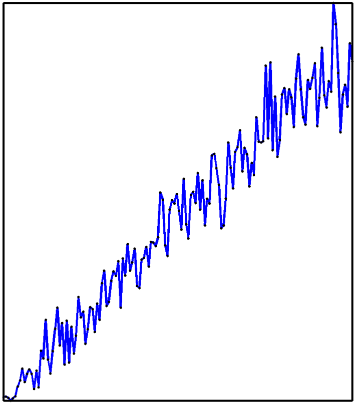
Advice: use a *short* exposure time and *high* ISO value

A simple experiment

Shot #1

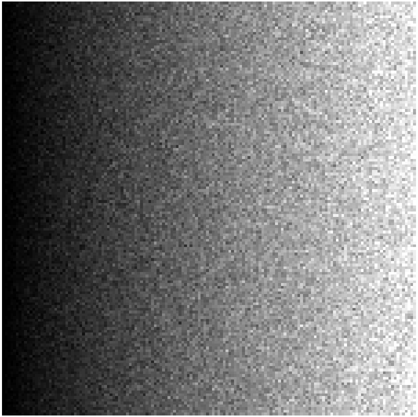


A simple experiment



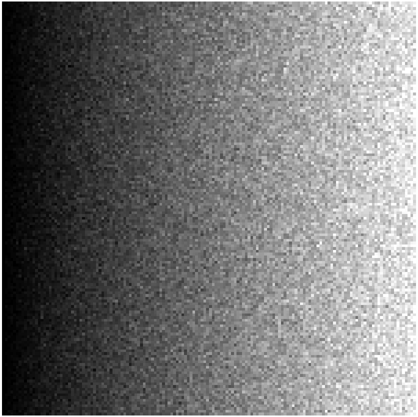
A simple experiment

Shot #2



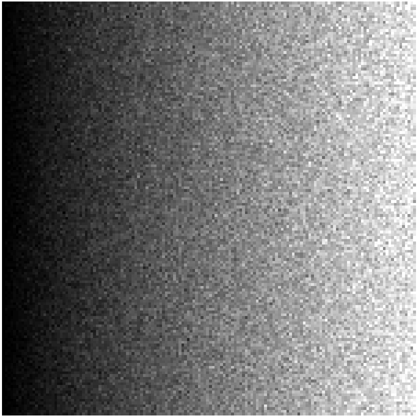
A simple experiment

Shot #3



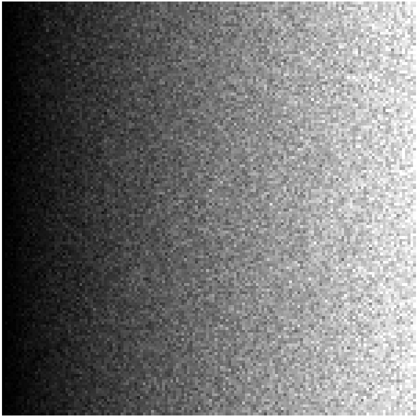
A simple experiment

Shot #4



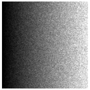
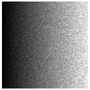
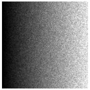
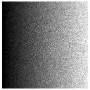
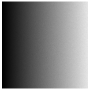
A simple experiment

Shot #5



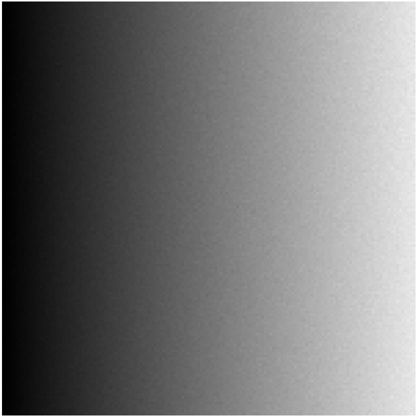
A simple experiment

TAKE MANY MORE SHOTS, AND THEN AVERAGE THEM ALL

$\frac{1}{N}\sum$  $+$  $+$  $+ \cdots +$  $=$ 

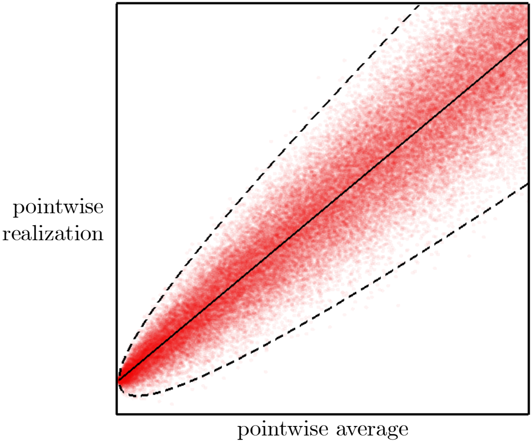
A simple experiment

TAKE MANY MORE SHOTS, AND THEN AVERAGE THEM ALL



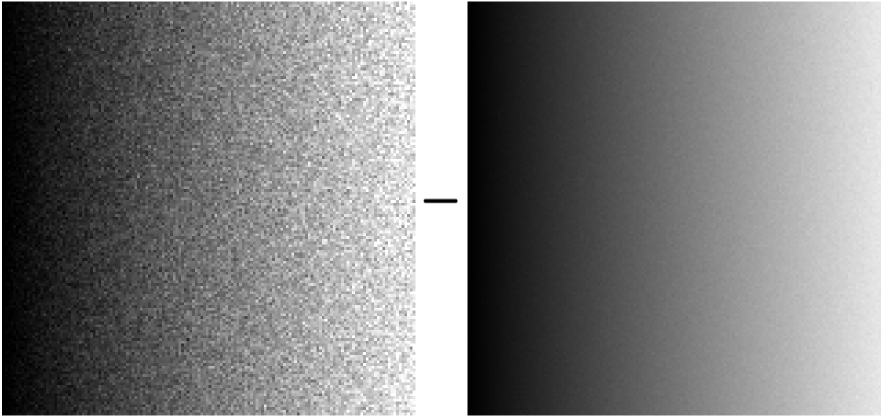
A simple experiment

Scatterplot: average vs realization



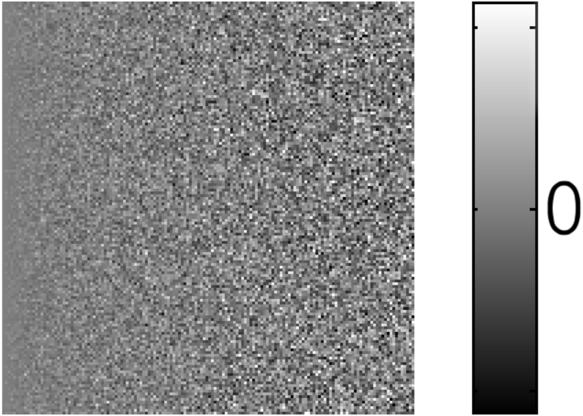
A simple experiment

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



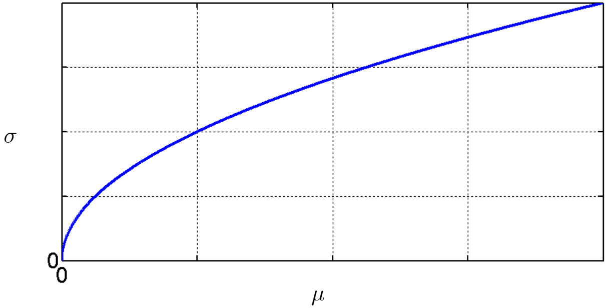
A simple experiment

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



A simple experiment

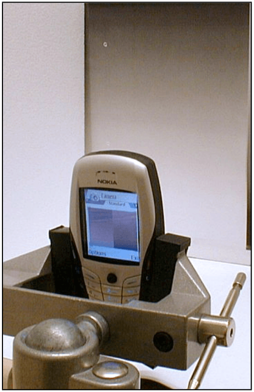
FOR EACH PIXEL, COMPUTE
SAMPLE MEAN AND *SAMPLE STANDARD DEVIATION*
W.R.T. THE VARIOUS SHOTS

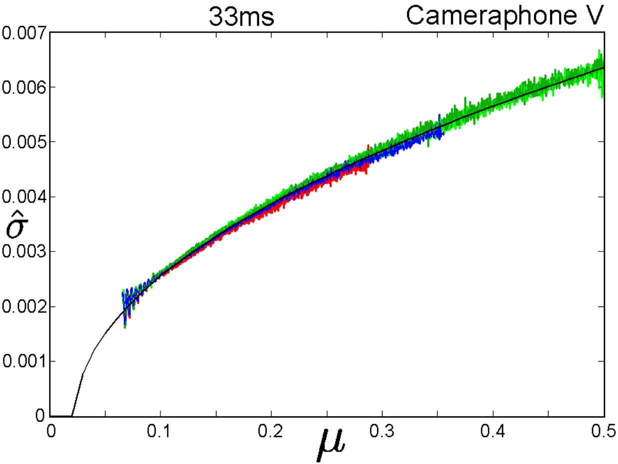


NOISE IS STRONGER WHERE THE AVERAGE IMAGE IS BRIGHTER:
STANDARD-DEVIATION IS A FUNCTION OF MEAN

SIGNAL-DEPENDENT NOISE

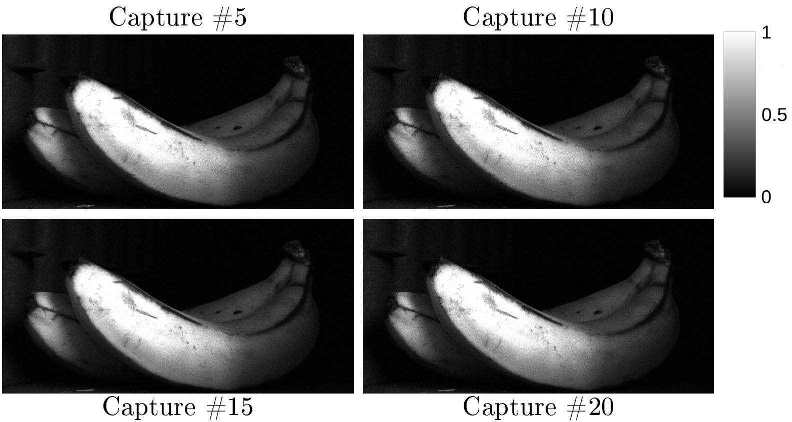
A simple experiment (Nokia 6600)





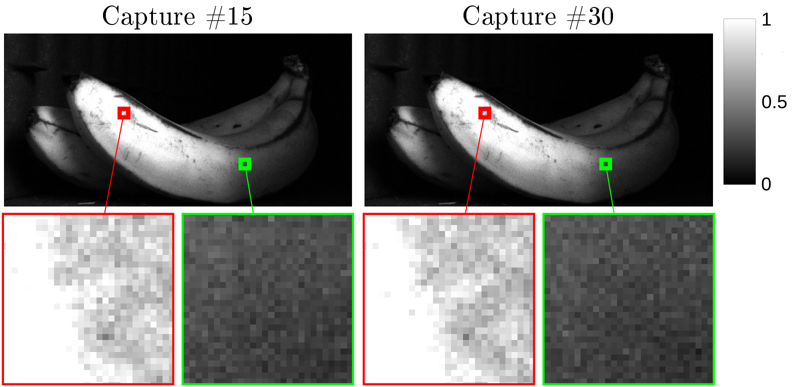
analysis of raw data from cameraphone CMOS sensor (F&al.SensJ2007)

A simple experiment (Samsung S8)



Examples from set of 30 raw images captured under identical settings with a Samsung S5K2L2 CMOS ISOCELL sensor at ISO 1250.

A simple experiment



Sample mean and sample standard deviation

We denote by $\tilde{z}^{(m)}(x)$ the pixel value at coordinate x in the m-th captured frame, modeled as a realization of a random variable $\tilde{z}(x)$.

Sample mean and sample standard deviation

We denote by $\tilde{z}^{(m)}(x)$ the pixel value at coordinate x in the m-th captured frame, modeled as a realization of a random variable $\tilde{z}(x)$.

$$\mathrm{E}\{\tilde{z}(x)\} \approx \frac{1}{M}\sum_{m=1}^{M} \tilde{z}^{(m)}(x) = \mathrm{E}\widehat{\{\tilde{z}(x)\}}$$

$$\mathrm{std}\{\tilde{z}(x)\} \approx \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}\left(\tilde{z}^{(m)}(x) - \frac{1}{M}\sum_{l=1}^{M}\tilde{z}^{(l)}(x)\right)^2} = \mathrm{std}\widehat{\{\tilde{z}(x)\}}$$

Sample mean and sample standard deviation

We denote by $\tilde{z}^{(m)}(x)$ the pixel value at coordinate x in the m-th captured frame, modeled as a realization of a random variable $\tilde{z}(x)$.

$$\mathrm{E}\{\tilde{z}(x)\} \approx \frac{1}{M}\sum_{m=1}^{M}\tilde{z}^{(m)}(x) = \mathrm{E}\widehat{\{\tilde{z}(x)\}}$$

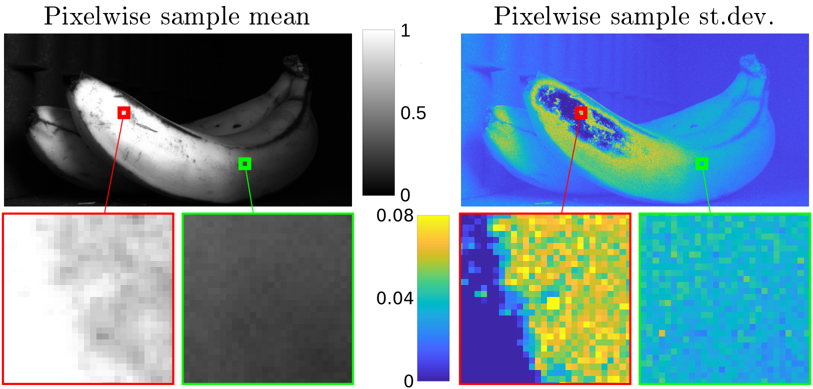
$$\mathrm{std}\{\tilde{z}(x)\} \approx \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}\left(\tilde{z}^{(m)}(x) - \frac{1}{M}\sum_{l=1}^{M}\tilde{z}^{(l)}(x)\right)^2} = \mathrm{std}\widehat{\{\tilde{z}(x)\}}$$

$$\mathrm{E}\widehat{\{\tilde{z}(x)\}} \sim \mathcal{N}\left(\mathrm{E}\{\tilde{z}(x)\}, \frac{1}{M}\mathrm{var}\{\tilde{z}(x)\}\right),$$

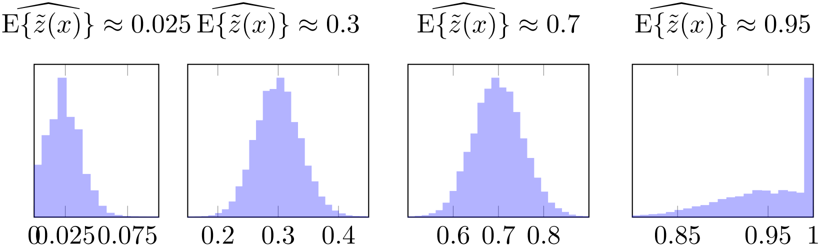
$$\mathrm{std}\widehat{\{\tilde{z}(x)\}} \sim \mathcal{N}\left(\mathrm{std}\{\tilde{z}(x)\}, \frac{2+\kappa}{4M}\mathrm{var}\{\tilde{z}(x)\}\right),$$

κ denotes the excess kurtosis of $\tilde{z}(x)$.

A simple experiment



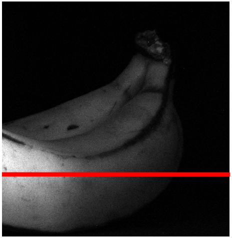
Sample histograms of $\tilde{z}(x)$

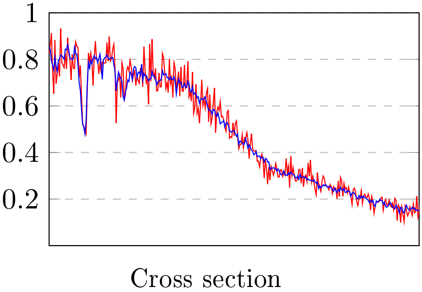


$\sigma^2 = 0.0002$	$\sigma^2 = 0.0014$	$\sigma^2 = 0.0029$	$\sigma^2 = 0.0024$
$\kappa = 0.71$	$\kappa = 0.12$	$\kappa = 0.014$	$\kappa = 0.032$

Below each histogram we report its variance σ^2 and excess kurtosis κ.

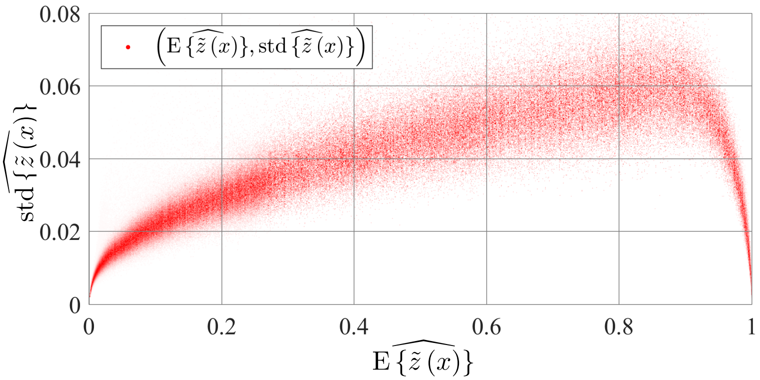
Cross section





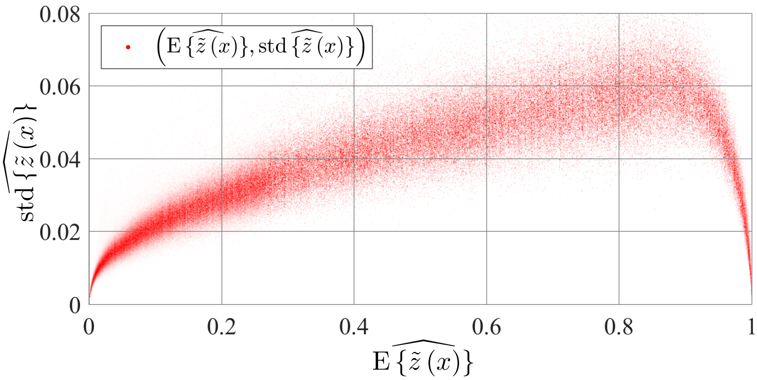
Left: detail from the dataset with highlighted cross section. Right: cross section (red line) plotted against the pixelwise sample mean (blue line).

Mean-St.Dev. Scatterplot



Scatterplot of the pairs $\left(\widehat{\mathrm{E}\{\tilde{z}(x)\}}, \widehat{\mathrm{std}\{\tilde{z}(x)\}}\right)$ drawn as red dots.

Mean-St.Dev. Scatterplot



Scatterplot of the pairs $\left(\widehat{\mathrm{E}\{\tilde{z}(x)\}}, \widehat{\mathrm{std}\{\tilde{z}(x)\}}\right)$ drawn as red dots. The dispersion visible in the scatterplot is described by the distributions of the estimated pairs.

Noise Modeling 101, and Beyond

Additive White Gaussian Noise (AWGN) model

$$z(x) = y(x) + \sigma\xi(x) \qquad x \in X$$

$y : X \to Y \subseteq \mathbb{R}$	unknown original image (deterministic)
$\sigma\xi(x)$	i.i.d. zero-mean random error
$z : X \to Z \subseteq \mathbb{R}$	observed noisy image (random)
$x \in X \subseteq \mathbb{Z}$	coordinate in the image domain

Additive White Gaussian Noise (AWGN) model

$$z(x) = y(x) + \sigma\xi(x) \qquad x \in X$$

$y : X \to Y \subseteq \mathbb{R}$	unknown original image (deterministic)
$\sigma\xi(x)$	i.i.d. zero-mean random error
$z : X \to Z \subseteq \mathbb{R}$	observed noisy image (random)
$x \in X \subseteq \mathbb{Z}$	coordinate in the image domain

$\sigma \in \mathbb{R}^+$	standard deviation of $\sigma\xi(x)$
$\xi(x)$	standard normal random variable
	$\mathrm{E}\{\xi(x)\} = 0 \quad \mathrm{var}\{\xi(x)\} = 1$

Additive White Gaussian Noise (AWGN) model

$$\boxed{z(x) = y(x) + \sigma\xi(x) \qquad x \in X}$$

$y : X \to Y \subseteq \mathbb{R}$	unknown original image (deterministic)
$\sigma\xi(x)$	i.i.d. zero-mean random error
$z : X \to Z \subseteq \mathbb{R}$	observed noisy image (random)
$x \in X \subseteq \mathbb{Z}$	coordinate in the image domain

$\sigma \in \mathbb{R}^+$	standard deviation of $\sigma\xi(x)$
$\xi(x)$	standard normal random variable
	$\mathrm{E}\{\xi(x)\} = 0 \quad \mathrm{var}\{\xi(x)\} = 1$
$\mathrm{E}\{z(x)\} = y(x)$	expectation of z
$\mathrm{std}\{z(x)\} = \sigma \mathrm{std}\{\xi(x)\} = \sigma$	standard deviation of z

Additive White Gaussian Noise (AWGN) model

$$z(x) = y(x) + \sigma\xi(x) \qquad x \in X$$

$y : X \to Y \subseteq \mathbb{R}$	unknown original image (deterministic)
$\sigma\xi(x)$	i.i.d. zero-mean random error
$z : X \to Z \subseteq \mathbb{R}$	observed noisy image (random)
$x \in X \subseteq \mathbb{Z}$	coordinate in the image domain

$\sigma \in \mathbb{R}^+$	standard deviation of $\sigma\xi(x)$
$\xi(x)$	standard normal random variable
	$\mathrm{E}\{\xi(x)\} = 0 \quad \mathrm{var}\{\xi(x)\} = 1$
$\mathrm{E}\{z(x)\} = y(x)$	expectation of z
$\mathrm{std}\{z(x)\} = \sigma \mathrm{std}\{\xi(x)\} = \sigma$	standard deviation of z

!!! Often z, ξ are used to denote both the random variables/processes and their realizations.

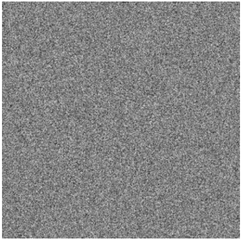
Additive White Gaussian Noise (AWGN) model

z

$=$

y

$+$



$\sigma\xi$

Additive *White* Gaussian Noise (AWGN) model

white:

$$\operatorname{var}\{\mathcal{F}(\sigma\xi)\} = \text{constant} \quad \text{(noise power spectrum is flat)}$$

This nomenclature is perhaps misleading.

What we demand is $\sigma\xi(x)$ to be *independent* and *identically distributed.*

identically distributed:

$$\Pr[\sigma\xi(x_1) < c] = \Pr[\sigma\xi(x_2) < c] \qquad \forall c \in \mathbb{R}$$

independent:

$$\Pr[\sigma\xi(x_1) < c]\Pr[\sigma\xi(x_2) < d] = \Pr[(\sigma\xi(x_1) < c) \cap (\sigma\xi(x_2) < d)] \quad \forall c, d \in \mathbb{R}$$

Additive *White* Gaussian Noise (AWGN) model

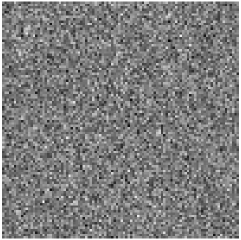
independence implies whiteness:

$$\begin{aligned}
\mathcal{F}(\sigma\xi)(\omega) &= \sum_{x\in X} e^{-2\pi i\omega x}\sigma\xi(x) \\
\operatorname{var}\{\mathcal{F}(\sigma\xi)(\omega)\} &= \sum_{x\in X} \left|e^{-2\pi i\omega x}\right|^2 \operatorname{var}\{\sigma\xi(x)\} = \\
&= \sum_{x\in X} \operatorname{var}\{\sigma\xi(x)\} \qquad \left(=\sigma^2 |X| \quad \text{because id. distr.}\right)
\end{aligned}$$

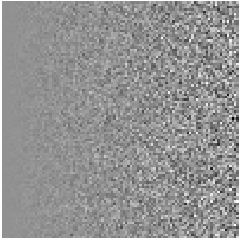
We can have Gaussian white noise that is not i.i.d.!!

How? It suffices to have independent but not identically distributed errors.

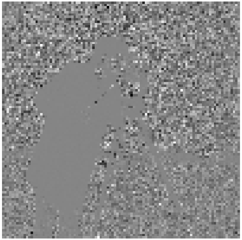
Various examples of Gaussian white noise



i.i.d.

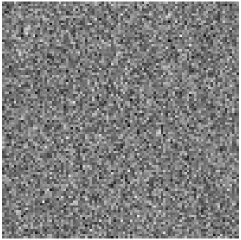


ramp

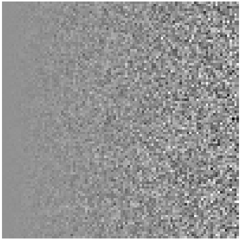


Cameraman

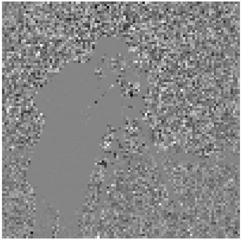
Various examples of Gaussian white noise



i.i.d.



ramp



Cameraman

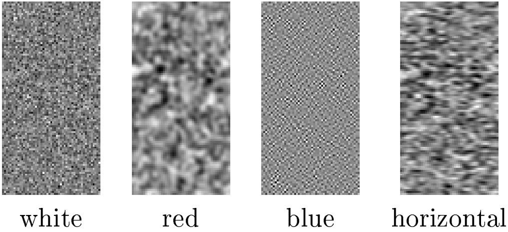
They are all three Gaussian and white, but only
the i.i.d. one is what is typically assumed as AWGN.

:-(

Colored noise

Noise is *colored* when the noise power spectrum is markedly not flat.

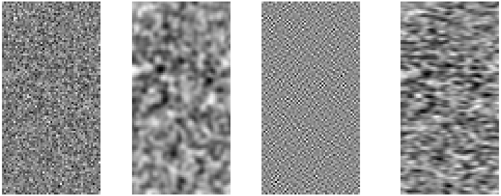
The band with larger variance determines the "color".



Colored noise

Noise is *colored* when the noise power spectrum is markedly not flat.

The band with larger variance determines the "color".



white red blue horizontal

Typically modeled by kernel convolution operator against white noise:

$$\begin{aligned} \mathcal{F}(v \circledast \xi) &= \mathcal{F}(v)\,\mathcal{F}(\xi) \\ \operatorname{var}\{\mathcal{F}(v \circledast \xi)\} &= |\mathcal{F}(v)|^2 \operatorname{var}\{\mathcal{F}(\xi)\} \end{aligned}$$

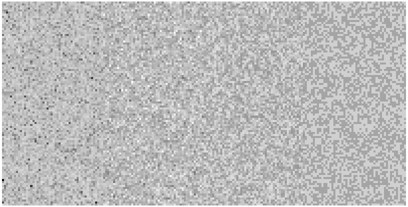
Homoskedasticity vs. Heteroskedasticity

The noise η is **homoskedastic** if different noise samples have same variance:

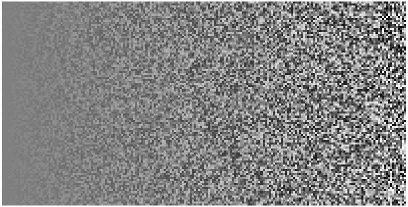
$$\operatorname{var}\{\eta(x')\} = \operatorname{var}\{\eta(x'')\} \quad \forall x', x'' \in X$$

otherwise it is **heteroskedastic** and different noise samples can have different variance:

$$\operatorname{var}\{\eta(x')\} \neq \operatorname{var}\{\eta(x'')\} \quad \text{for some } \ x', x'' \in X.$$



homoskedastic (but *not* ident.distr.)



heteroskedastic

Standard-deviation map

Let $z(x) = y(x) + \eta(x), \quad x \in X,$ with η **heteroskedastic** noise.

Whenever the variance $\mathrm{var}\{\eta\}$ is deterministic, it makes sense to break η into two factors:

$$\eta = \mathrm{std}\{\eta\}\,\xi$$

$$\begin{aligned} \mathrm{std}\{\eta\} : X \to \mathbb{R}^+ & \qquad \text{standard-deviation map (deterministic)} \\ \xi : X \to \mathbb{R} & \qquad \text{homoskedastic noise (random)} \\ & \qquad \mathrm{std}\{\xi\}(x) = 1 \quad \forall x \in X \end{aligned}$$

η $\qquad$ $\mathrm{std}\{\eta\}$ $\qquad$ ξ

Signal-dependent noise

The η noise is *signal-dependent* when the distribution of $\eta(x)$ has some parameter that depends on $y(x)$:

$$\Pr[\eta(x) < c] = F(c, y(x)), \quad \forall x \in X \text{ and } \forall c \in \mathbb{R}$$
$$\text{with } F \text{ functionally independent of } x.$$

The most significant situation arises when the variance of η depends on y, i.e. when the standard-deviation map becomes a function of y:

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

$\sigma : \mathbb{R} \to \mathbb{R}^{+}$	**standard-deviation function or curve** (deterministic),
$\xi(x)$	random variable $E\{\xi(x)\} = 0 \quad \text{var}\{\xi(x)\} = 1$.

Here ξ is homoskedastic noise with unitary variance.
The distribution of $\xi(x)$ may depend on $y(x)$, but what most matters is its variance.

Multiplicative noise

Multiplicative noise is special case of signal-depedent noise where the mean is the direct scaling parameter of the noise distribution.

$$z(x) = y(x) \cdot \eta_{\mathrm{mult}}(x), \qquad x \in X,$$

$$\eta_{\mathrm{mult}} \quad \text{i.i.d. noise}, \quad E\{\eta_{\mathrm{mult}}(x)\} = 1, \quad \mathrm{std}\{\eta_{\mathrm{mult}}(x)\} = c.$$

Rewrite in additive signal-dependent form:

$$\begin{aligned} z(x) &= y(x) + y(x)(\eta_{\mathrm{mult}}(x) - 1) = \\ &= y(x) + y(x)\xi'(x) = \\ &= y(x) + \sigma(y(x))\xi(x), \end{aligned}$$

where $\sigma : \mathbb{R} \to \mathbb{R}^+, \;\; \sigma : y \longmapsto c|y|$

and $\xi(x) = \mathrm{sign}\{y(x)\} c^{-1}\xi'(x) = \mathrm{sign}\{y(x)\} c^{-1}(\eta_{\mathrm{mult}}(x) - 1)$.

We have $E\{\xi(x)\} = 0, \;\; \mathrm{var}\{\xi(x)\} = 1.$

Poisson distributions

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

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

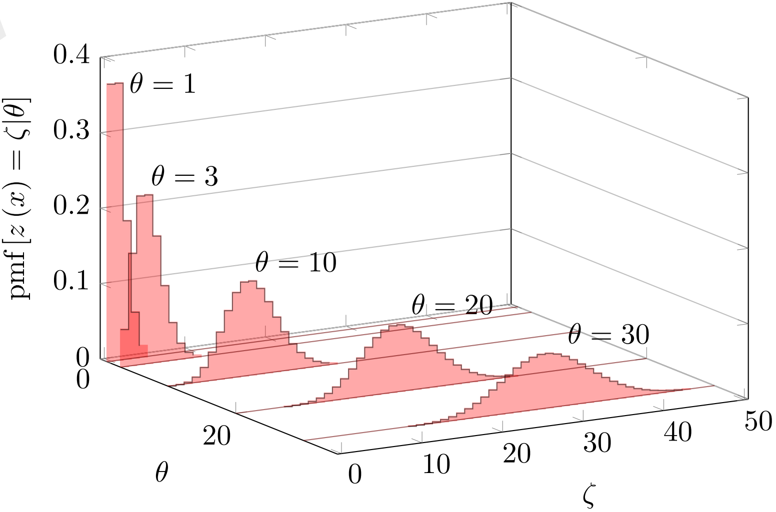
$$\begin{aligned} \mu(\theta) &= E\{z|\theta\} = \theta \\ \sigma^2(\theta) &= \operatorname{var}\{z|\theta\} = \theta = \mu(\theta) \end{aligned}$$

mean and variance coincide and are equal to the parameter θ

Matlab code: z = `poissrnd(theta)`

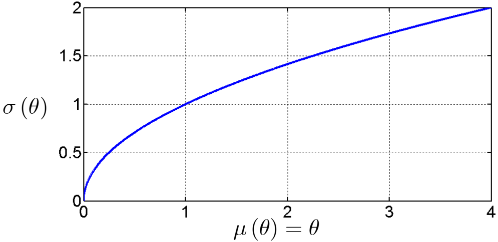
signal-to-noise ratio (SNR): $\frac{\mu(\theta)}{\sigma(\theta)} = \sqrt{\theta} \underset{\theta \to 0}{\longrightarrow} 0 \qquad \frac{\mu(\theta)}{\sigma(\theta)} \underset{\theta \to +\infty}{\longrightarrow} +\infty$

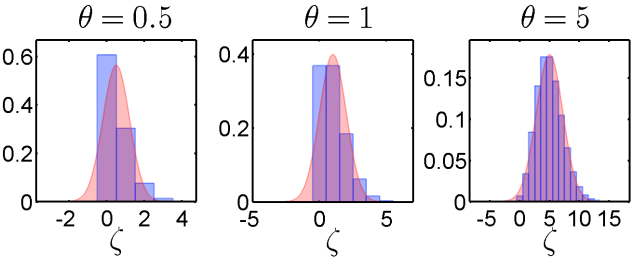
Poisson distributions



Distributions with mean $\theta = 1, 3, 10, 20, 30$.

Poisson distributions





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

Normal approximation of Poisson

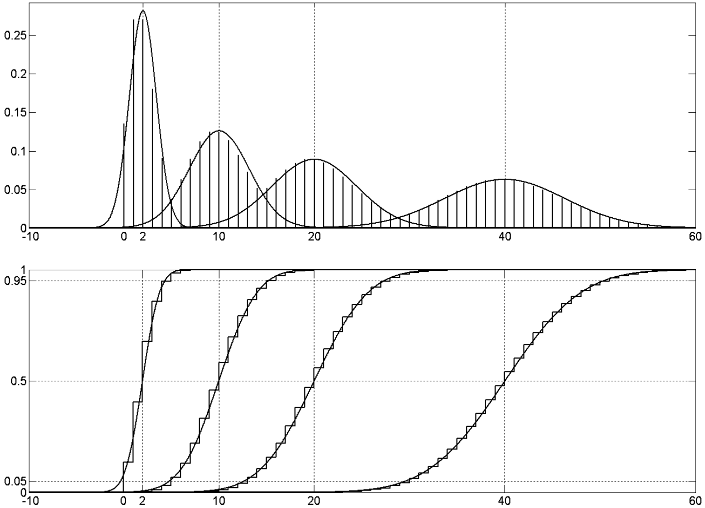
$z \sim \mathcal{P}(\theta)$ means the probability of z $\Pr[z = \zeta|\theta] = e^{-\theta}\dfrac{\theta^{\zeta}}{\zeta!}$, $\zeta \in \mathbb{N}$

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

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

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

Normal approximation of Poisson



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

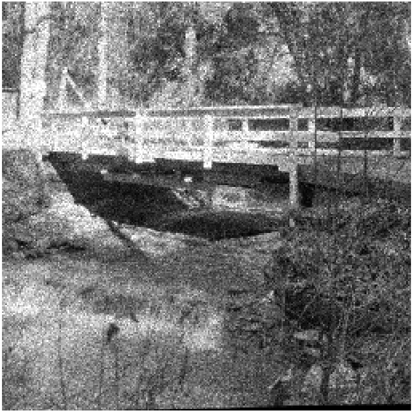
Noise-free image y

y

Noisy Poisson image - peak 256

$z \sim \mathcal{P}(y)$ when peak of y is 256

Noisy Poisson image - peak 64

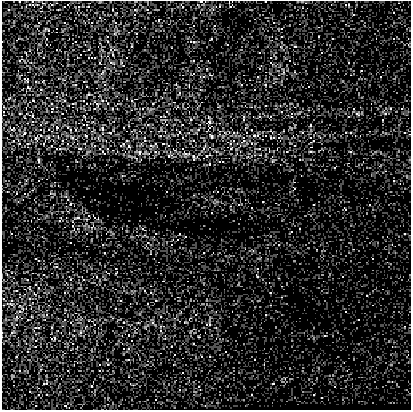


$z \sim \mathcal{P}(y)$ when peak of y is 64

Noisy Poisson image - peak 8

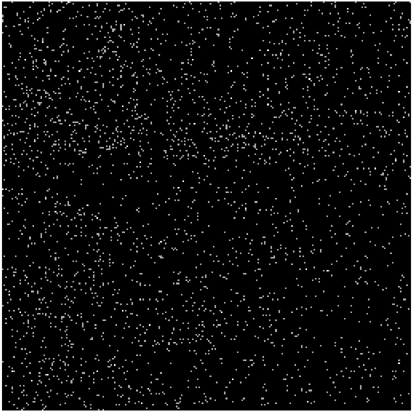
$z \sim \mathcal{P}(y)$ when peak of y is 8

Noisy Poisson image - peak 1



$z \sim \mathcal{P}(y)$ when peak of y is 1

Noisy Poisson image - peak 0.1



$z \sim \mathcal{P}(y)$ when peak of y is 0.1

Scaled Poisson distributions

Scaled Poisson distributions with scale parameter $\chi > 0$ and mean $\theta \geq 0$

$$z\chi \sim \mathcal{P}(\theta\chi) \qquad \Pr[z = \zeta|\theta] = e^{-\theta\chi}\frac{(\theta\chi)^{\zeta\chi}}{(\zeta\chi)!}, \quad \zeta\chi \in \mathbb{N}, \quad \theta \in [0, +\infty).$$

Discrete taking values that are nonnegative integer multiples of $\frac{1}{\chi}$.

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

mean is equal to the parameter θ and coincides with the variance times χ.

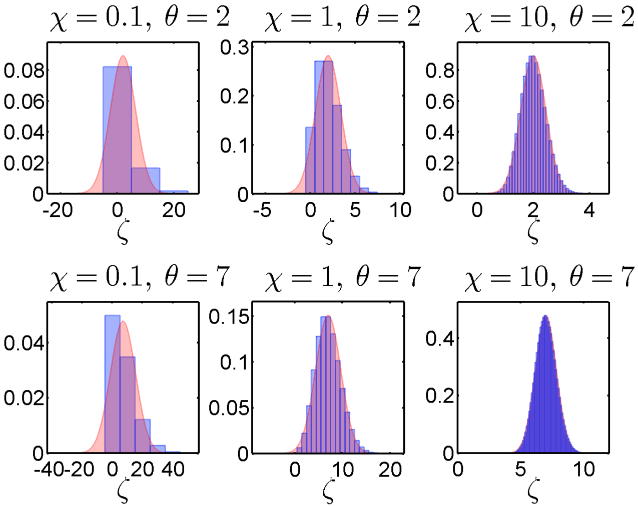
The scale parameter χ controls the relative strength of the noise: SNR $\frac{\mu(\theta)}{\sigma(\theta)} = \sqrt{\chi\theta}$.

Matlab code: z = `poissrnd(chi*theta)/chi`

Normal approximation for large θ: $z \sim \mathcal{N}(\theta, \theta/\chi)$

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

Scaled Poisson distributions



small χ is detremental when θ varies on a narrow range of values

Poissonian noise

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

$\chi > 0$ scaling factor

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

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

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

This can be rewritten in the usual form as

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

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**.

Scaled Poisson observations

$\chi = 1000$

Scaled Poisson observations

$\chi = 300$

Scaled Poisson observations

$\chi = 100$

Scaled Poisson observations

$\chi = 50$

Scaled Poisson observations

$\chi = 10$

Scaled Poisson observations

$\chi = 1$

One-parameter families of distributions

A one-parameter family of distributions $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_\theta\}$ is a collection of distributions, each of which is identified bit the value of a univariate parameter $\theta \in \Theta \subseteq \mathbb{R}$.

Let $z \in Z \subseteq \mathbb{R}$ be a random variable distributed according to a one-parameter family of distributions $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_\theta\}$.

For each individual $\theta \in \Theta$: $\mathcal{D}_\theta$ is a distribution, $z|\theta \sim \mathcal{D}_\theta$, $z|\theta \in Z_\theta \subseteq Z$

$\mu(\theta) = E\{z|\theta\}$ conditional expectation of z expressed as function of θ.
$\sigma(\theta) = \mathrm{std}\{z|\theta\}$ conditional standard deviation of z expressed as function of θ.

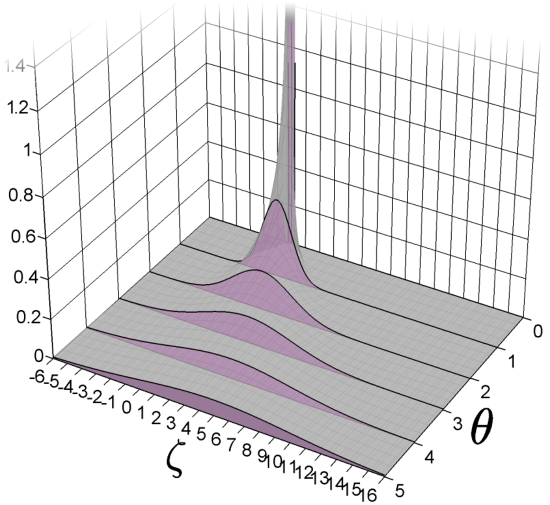
Poisson example:

$\Theta = [0, +\infty) \subset \mathbb{R}$
$\mathcal{D}_\theta$ is one Poisson distribution with parameter $\theta \in \Theta$
$Z_\theta = \{0, 1, 2, \ldots\} = \mathbb{N}$
$\mu(\theta) = \theta$
$\sigma(\theta) = \theta$

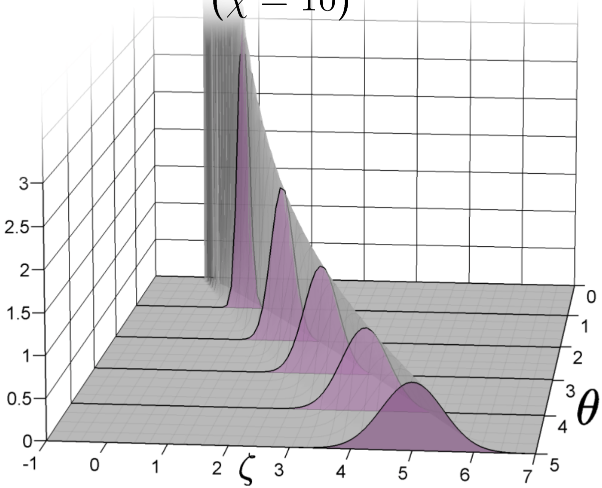
One-parameter families of distributions: examples

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

Multiplicative Gaussian noise pdf $[z|\theta](\zeta)$ $(\chi = 1)$



Multiplicative Gaussian noise pdf $[z|\theta](\zeta)$ $(\chi = 10)$



Poisson-Gaussian noise

Each observed pixel intensity value $z(x)$, $x \in X$, is composed of a scaled Poisson and an additive Gaussian component:

$$z(x) = \alpha p(x) + n(x),$$

where $p(x) \sim \mathcal{P}(y(x))$, $y(x)$ is the unknown noise-free pixel intensity, $\alpha > 0$ is a gain or scaling parameter, and $n(\cdot) \sim \mathcal{N}\left(0, \sigma^2\right)$.

Poisson-Gaussian noise is defined as

$$\eta(x) = z(x) - \alpha y(x).$$

Signal-dependent standard deviation:

$$\text{std}\left\{z(x) | y(x)\right\} = \sqrt{\alpha^2 y(x) + \sigma^2}.$$

Rician-distributed data

Let $z \sim \mathcal{R}(\nu, \sigma)$ be the realization of a random variable with Rician p.d.f. with parameters $\nu \geq 0$ and $\sigma > 0$,

$$p(z|\nu, \sigma) = \frac{z}{\sigma^2} e^{-\frac{z^2+\nu^2}{2\sigma^2}} I_0\left(\frac{z\nu}{\sigma^2}\right), \qquad z \geq 0, \tag{1}$$

where I_n denotes the modified Bessel function of order n.

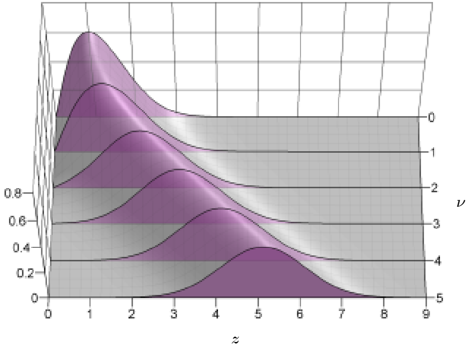
Equivalently, $z = \sqrt{(c_r\nu + \sigma\eta_r)^2 + (c_i\nu + \sigma\eta_i)^2}$,
where c_r and c_i are arbitrary constants such that $0 \leq c_r, c_i \leq 1 = c_r^2 + c_i^2$, and $\eta_r, \eta_i \sim \mathcal{N}(0, 1)$.

Observation model for magnitude magnetic resonance (MR) images/volumes:

$z(x) \sim \mathcal{R}(\nu(x), \sigma)$, $\quad x \in X \subset \mathbb{Z}^d$, $d = 2, 3$ (pixel or voxel coordinates).

$\nu : X \to \mathbb{R}^+$ is the unknown original (noise-free) signal

$z : X \to \mathbb{R}^+$ is the raw magnitude MR data.



The one-parameter family of Rician p.d.f.'s $\mathcal{R}(\nu, 1)$ for $\nu \in [0, 5]$.

The parameter σ is assumed as fixed. Thus, z is treated as distributed according to a one-parameter family of Rician distributions, parametrized with respect to ν: $\mathcal{R}(\cdot, \sigma)$.
Assuming $\sigma = 1$ is not a serious restriction: $z \sim \mathcal{R}(\nu, \sigma)$ iff $\lambda z \sim \mathcal{R}(\lambda\nu, \lambda\sigma)$ $\forall \lambda > 0$.
Thanks to this scaling we can carry out all analysis for $\sigma = 1$, and then apply it to other cases $\sigma > 0$ upon simple linear rescaling of data and parameters.
Given $f : \mathbb{R}^+ \to \mathbb{R}$, we have that $\operatorname{var}\{f(z) | \nu, \sigma\} = \operatorname{var}\{f_\lambda(w) | \lambda\nu, \lambda\sigma\}$, where $z \sim \mathcal{R}(\nu, \sigma)$, $w = \lambda z \sim \mathcal{R}(\lambda\nu, \lambda\sigma)$ and $f_\lambda(w) = f(w/\lambda)$ $\forall w \in \mathbb{R}^+$.

Mean and variance of Rician data

The mean and variance of $z \sim \mathcal{R}(\nu, \sigma)$ are, respectively,

$$\mu = E\{z|\nu,\sigma\} = \sigma\sqrt{\frac{\pi}{2}}L\left(-\frac{\nu^2}{2\sigma^2}\right), \tag{2}$$

$$s^2 = \mathrm{var}\{z|\nu,\sigma\} = 2\sigma^2 + \nu^2 - \frac{\pi\sigma^2}{2}L^2\left(-\frac{\nu^2}{2\sigma^2}\right), \tag{3}$$

where $L(x) = e^{x/2}\left[(1-x)I_0\left(-\frac{x}{2}\right) - xI_1\left(-\frac{x}{2}\right)\right]$.

For large values of ν we have

$$E\{z|\nu,\sigma\} \approx \nu + \frac{\sigma^2}{2\nu}, \quad \mathrm{var}\{z|\nu,\sigma\} \approx \sigma^2 - \frac{\sigma^4}{2\nu^2}. \tag{4}$$

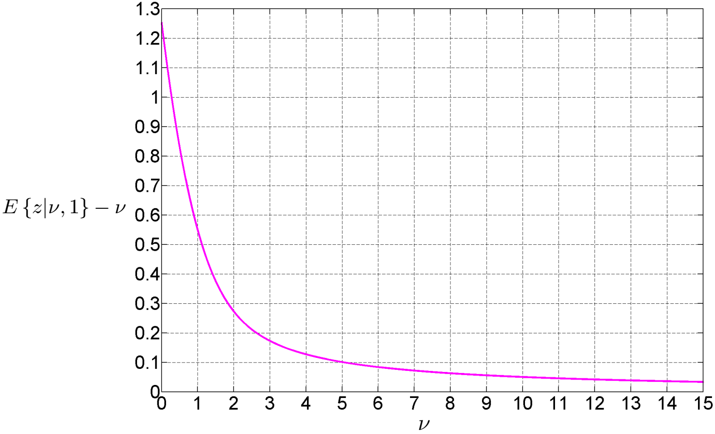
Two crucial issues follow from (2) and (3):

(3) implies that the noise variance is not uniform over the data.

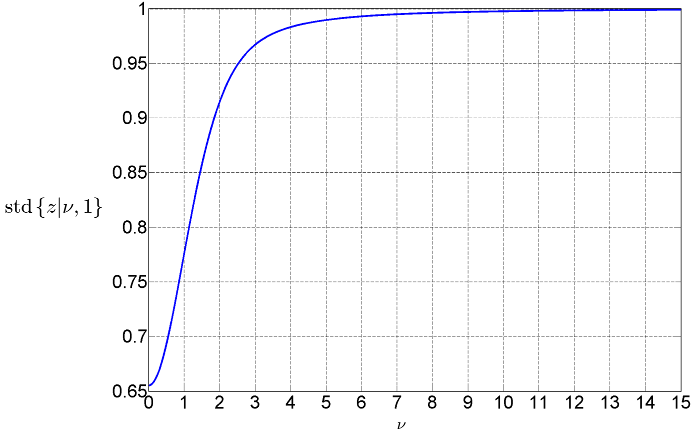
the expectation (2) differs essentially from the parameter of interest, namely ν.

The former problem is addressed by the (forward) variance-stabilizing transformation applied to the data before prior to filtering, wheareas the latter is addressed by the inverse transformation applied upon filtering, which is designed so to directly provide an estimate of ν out of the filtered transformed data.

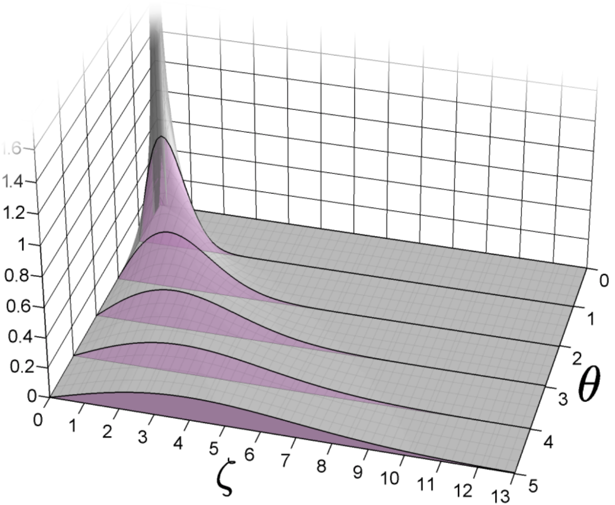
Mean of Rician data



Standard deviation of Rician data

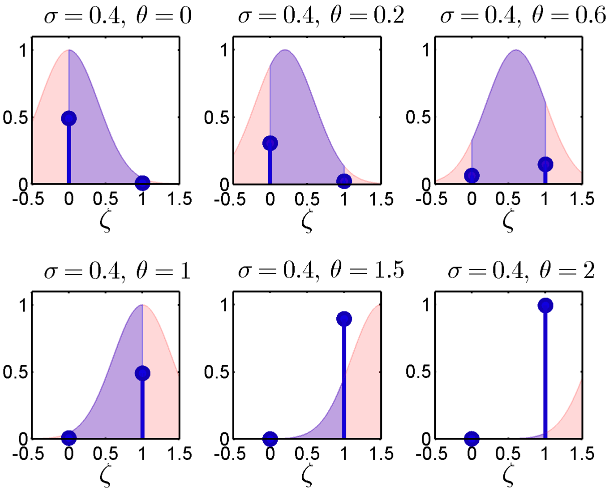


Rayleigh pdf $[z|\theta](\zeta)$



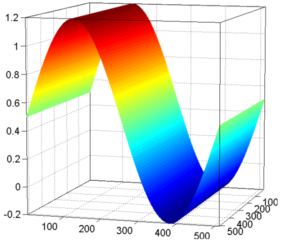
Doubly censored normal

(clipping from below and above)

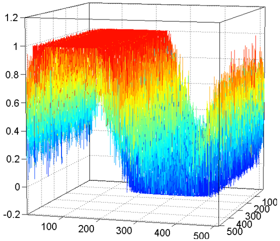


Underlying normal p.d.f. (uncensored) drawn in red

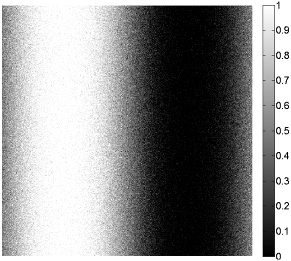
Doubly censored normal as a model for clipped noisy data

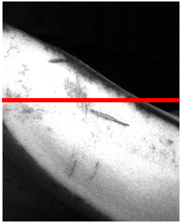


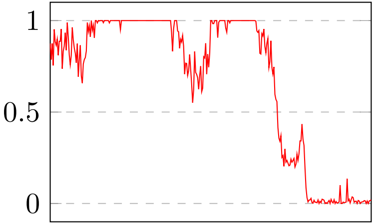
original



added AWGN and then clipped







Cross section

Raw data as clipped signal-dependent observations

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

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

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

Raw data as clipped signal-dependent observations

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

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

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

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

Modeling raw-data signal-dependence before clipping

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

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

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

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

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

Variance is an **affine** function of mean.

Higher-order models (e.g., quadratic functions) are also possible and allow to better capture nonlinearities in sensor response.

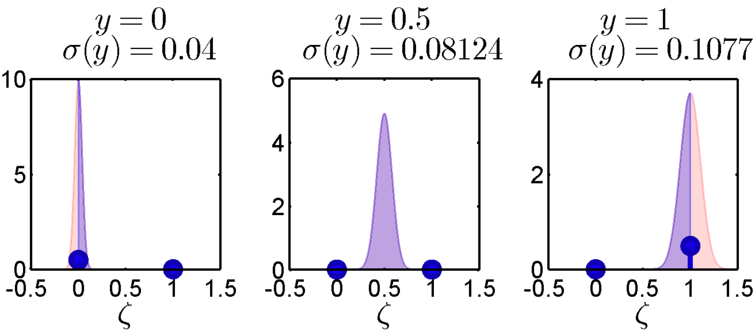
Heteroskedastic normal approximation

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

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

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

(Generalized) Probability distributions



Before clipping : $\wp_z(\zeta|y) = \frac{1}{\sigma(y)}\phi\left(\frac{\zeta - y}{\sigma(y)}\right)$

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

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

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

Expectations and variances

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

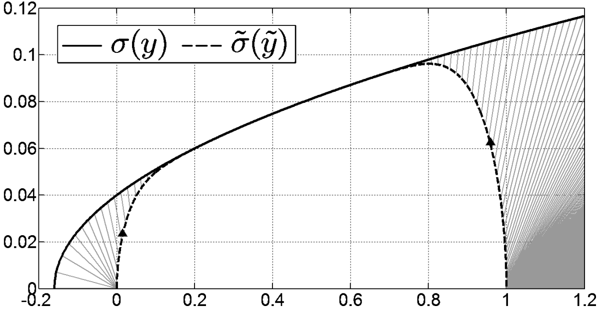
$$\begin{aligned}\operatorname{var}\{\tilde{z}|y\} = \tilde{\sigma}^2(\tilde{y}) = \Phi\left(\tfrac{y}{\sigma(y)}\right)\left(y^2 - 2\tilde{y}y + \sigma^2(y)\right) + \\ + \tilde{y}^2 - \Phi\left(\tfrac{y-1}{\sigma(y)}\right)\left(y^2 - 2\tilde{y}y + 2\tilde{y} + \sigma^2(y) - 1\right) + \\ + \sigma(y)\,\phi\left(\tfrac{y-1}{\sigma(y)}\right)(2\tilde{y}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

These equations are "universal", in the sense that the are valid for any variance function $\sigma^2(y)$, including non-affine ones.

(F.SigPro2009)

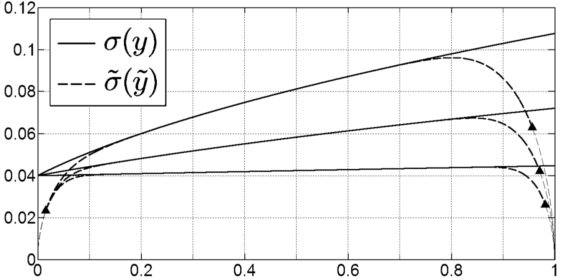
Expectations and variances

$$y = E\{z|y\}, \quad \sigma(y) = \text{std}\{z|y\}, \qquad \tilde{y} = E\{\tilde{z}|y\}, \quad \tilde{\sigma}(\tilde{y}) = \text{std}\{\tilde{z}|y\}.$$

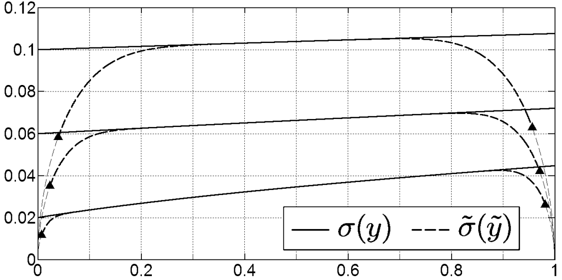


(F.SigPro2009)

Standard-deviation function $\sigma(y) = \sqrt{0.01y + 0.04^2}$ (solid line) and the corresponding standard-deviation curve $\tilde{\sigma}(\tilde{y})$ (dashed line).
The gray segments illustrate the mapping $\sigma(y) \longmapsto \tilde{\sigma}(\tilde{y})$.
The small black triangles ▲ indicate points $(\tilde{y}, \tilde{\sigma}(\tilde{y}))$ which correspond to $y = 0$ and $y = 1$.

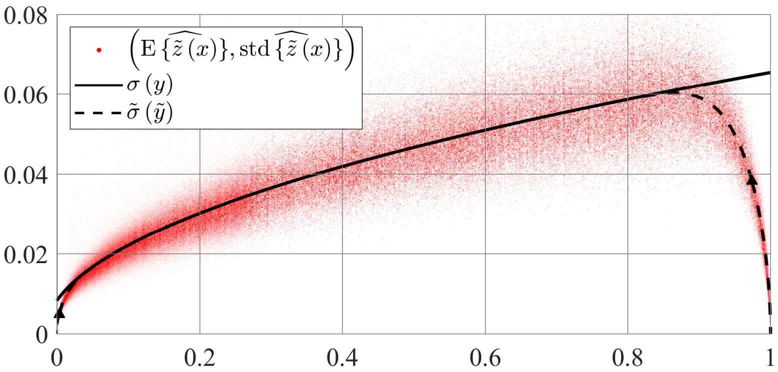


$$a = 0.02^2, 0.06^2, 0.10^2$$
$$b = 0.04^2$$



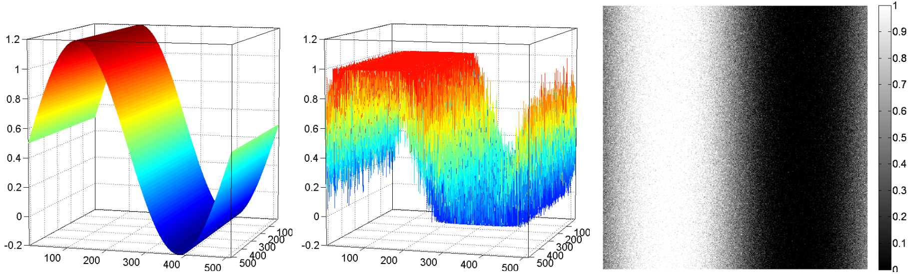
$$a = 0.04^2$$
$$b = 0.02^2, 0.06^2, 0.10^2$$

The model does indeed fit the data
(Samsung S8)



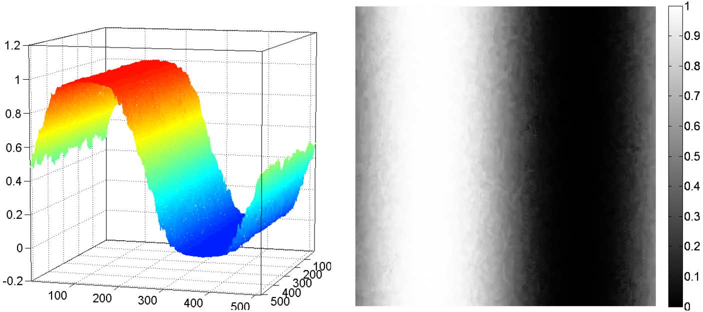
Scatterplot of the pairs $\left(\mathrm{E}\{\widehat{\tilde{z}(x)}\}, \mathrm{std}\{\widehat{\tilde{z}(x)}\}\right)$ drawn as red dots, and estimated clipped (black dashed line) and non-clipped (black continuous line) noise standard deviation curves. The small black triangles ▲ indicate points $(\tilde{y}, \tilde{\sigma}(\tilde{y}))$ which correspond to $y = 0$ and $y = 1$.

Declipping example



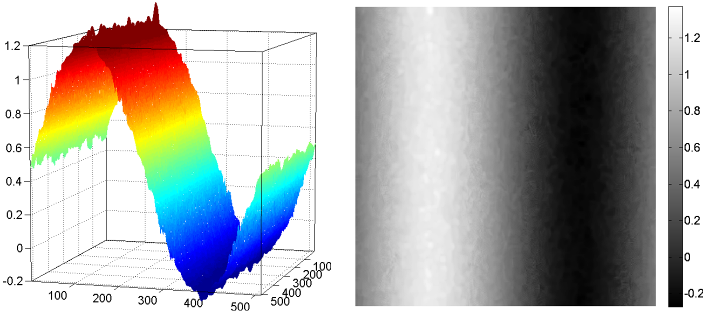
left: original (range [-0.2, 1.2]) center+right: noisy and clipped (range [0, 1])

Declipping example



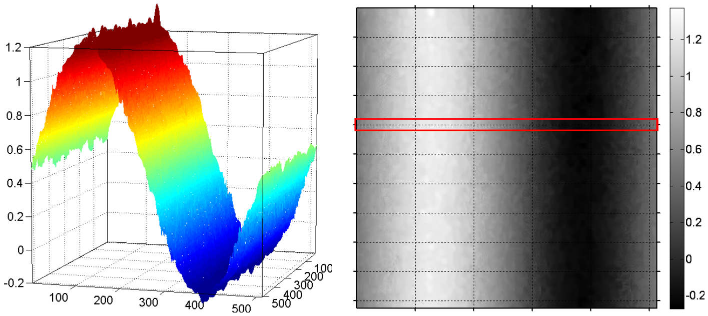
Denoising clipped data (range [0, 1])

Declipping example



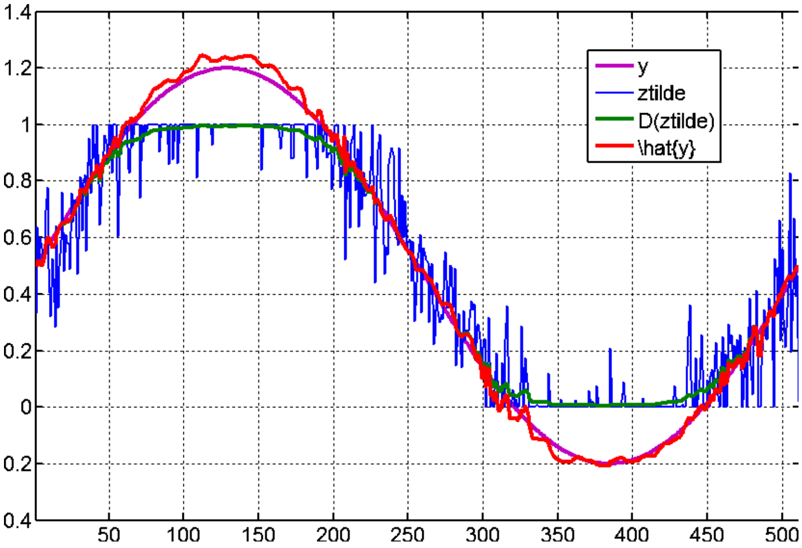
Denoising and declipping.

Declipping example



Denoising and declipping.

Declipping example



Cross-sections of observations and estimates.

Principles scatterplot fitting (1/2)

Goal: estimate the standard-deviation function (e.g., a, b).

Principles scatterplot fitting (1/2)

Goal: estimate the standard-deviation function (e.g., a, b).
Approach: build a scatterplot (mean, st.dev), fit a curve.

Principles scatterplot fitting (1/2)

Goal: estimate the standard-deviation function (e.g., a, b).
Approach: build a scatterplot (mean, st.dev), fit a curve.

Bivariate conditional PDF model for scatterpoints:

$$\operatorname{pdf}\left[\left(\hat{y}_i, \hat{\sigma}_i\right) | \tilde{y}_i = \tilde{y}\right] = \operatorname{pdf}\left[\hat{y}_i | \tilde{y}_i = \tilde{y}\right] \operatorname{pdf}\left[\hat{\sigma}_i | \tilde{y}_i = \tilde{y}\right].$$

Examples: product of univariate Gaussian PDFs (F.et al., 2008), a product of Gaussian-Cauchy mixtures (Azzari&F., 2014).

Principles scatterplot fitting (1/2)

Goal: estimate the standard-deviation function (e.g., a, b).
Approach: build a scatterplot (mean, st.dev), fit a curve.
Bivariate conditional PDF model for scatterpoints:

$$\text{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right] = \text{pdf}\left[\hat{y}_i|\tilde{y}_i = \tilde{y}\right]\text{pdf}\left[\hat{\sigma}_i|\tilde{y}_i = \tilde{y}\right].$$

Examples: product of univariate Gaussian PDFs (F.et al., 2008), a product of Gaussian-Cauchy mixtures (Azzari&F., 2014).

$$\frac{1}{2\pi\sqrt{c_i d_i}}\frac{1}{\tilde{\sigma}^2(\tilde{y})}\,e^{-\frac{1}{2\tilde{\sigma}^2(\tilde{y})}\left(\frac{(\hat{y}_i-\tilde{y})^2}{c_i}+\frac{(\hat{\sigma}_i-\tilde{\sigma}(\tilde{y}))^2}{d_i}\right)}$$

Principles scatterplot fitting (1/2)

Goal: estimate the standard-deviation function (e.g., a, b).
Approach: build a scatterplot (mean, st.dev), fit a curve.

Bivariate conditional PDF model for scatterpoints:

$$\text{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right] = \text{pdf}\left[\hat{y}_i|\tilde{y}_i = \tilde{y}\right]\text{pdf}\left[\hat{\sigma}_i|\tilde{y}_i = \tilde{y}\right].$$

Examples: product of univariate Gaussian PDFs (F.et al., 2008), a product of Gaussian-Cauchy mixtures (Azzari&F., 2014).

Posterior likelihood function L with the prior density pdf $[y]$:

$$L(a, b) = \prod_{i=1}^{N} \int \text{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right]\text{pdf}\left[y\right] dy.$$

Principles scatterplot fitting (1/2)

Goal: estimate the standard-deviation function (e.g., a, b).
Approach: build a scatterplot (mean, st.dev), fit a curve.

Bivariate conditional PDF model for scatterpoints:

$$\mathrm{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right] = \mathrm{pdf}\left[\hat{y}_i|\tilde{y}_i = \tilde{y}\right]\mathrm{pdf}\left[\hat{\sigma}_i|\tilde{y}_i = \tilde{y}\right].$$

Examples: product of univariate Gaussian PDFs (F.et al., 2008), a product of Gaussian-Cauchy mixtures (Azzari&F., 2014).

Posterior likelihood function L with the prior density $\mathrm{pdf}[y]$:

$$L(a,b) = \prod_{i=1}^{N} \int \mathrm{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right]\mathrm{pdf}[y]\, dy.$$

Non-informative non-negative prior for y (typical for raw data):

$$L(a,b) = \prod_{i=1}^{N} \int_0^{\infty} \mathrm{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right] dy\,.$$

Principles scatterplot fitting (1/2)

Goal: estimate the standard-deviation function (e.g., a, b).
Approach: build a scatterplot (mean, st.dev), fit a curve.

Bivariate conditional PDF model for scatterpoints:

$$\mathrm{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right] = \mathrm{pdf}\left[\hat{y}_i|\tilde{y}_i = \tilde{y}\right]\mathrm{pdf}\left[\hat{\sigma}_i|\tilde{y}_i = \tilde{y}\right].$$

Examples: product of univariate Gaussian PDFs (F.et al., 2008), a product of Gaussian-Cauchy mixtures (Azzari&F., 2014).

Posterior likelihood function L with the prior density $\mathrm{pdf}[y]$:

$$L(a, b) = \prod_{i=1}^{N} \int \mathrm{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right]\mathrm{pdf}[y]\,dy.$$

Non-informative non-negative prior for y (typical for raw data):

$$L(a, b) = \prod_{i=1}^{N} \int_0^{\infty} \mathrm{pdf}\left[(\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right]dy\,.$$

Estimation of curve parameters: $\left(\hat{a}, \hat{b}\right) = \mathrm{argmax}_{a,b} L(a, b)\,.$

Principles scatterplot fitting (2/2)

Classical scheme for building scatterplots from a single image

Employ some local or nonlocal low-pass (for mean) and high-pass filtering (for standard deviation);
E.g., split image into wavelet approximation and detail coefficients.

Challenge: ignore edges or high-frequency texture

Principles scatterplot fitting (2/2)

Classical scheme for building scatterplots from a single image

Employ some local or nonlocal low-pass (for mean) and high-pass filtering (for standard deviation);
E.g., split image into wavelet approximation and detail coefficients.

Challenge: ignore edges or high-frequency texture

1. Partitioning of the codomain to pair mean and st.dev. estimates (*conditioning*)

Principles scatterplot fitting (2/2)

Classical scheme for building scatterplots from a single image

Employ some local or nonlocal low-pass (for mean) and high-pass filtering (for standard deviation);
E.g., split image into wavelet approximation and detail coefficients.

Challenge: ignore edges or high-frequency texture

1. Partitioning of the codomain to pair mean and st.dev. estimates (*conditioning*)

2. Use wavelet approximation coefficients to estimate *conditional* expectations

Principles scatterplot fitting (2/2)

Classical scheme for building scatterplots from a single image

Employ some local or nonlocal low-pass (for mean) and high-pass filtering (for standard deviation);
E.g., split image into wavelet approximation and detail coefficients.

Challenge: ignore edges or high-frequency texture

1. Partitioning of the codomain to pair mean and st.dev. estimates (*conditioning*)

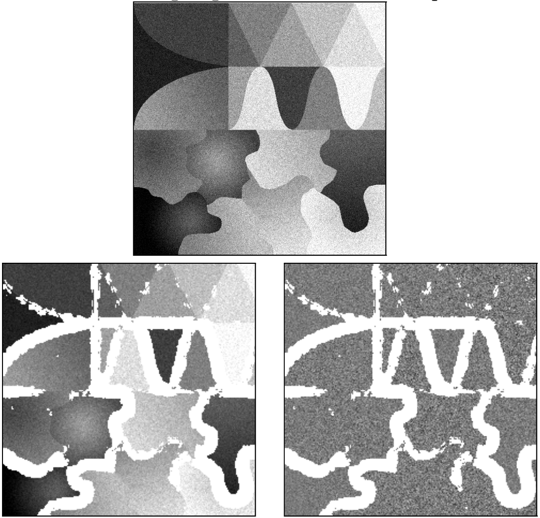
2. Use wavelet approximation coefficients to estimate *conditional* expectations

3. Use wavelet detail coefficients to estimate *conditional* standard-deviation.

It is crucial to use robust sample estimators, such as the Median Absolute Deviation (MAD), Inter-Quantile Range (IQR), or NoiseNet (Uss et al., 2018).

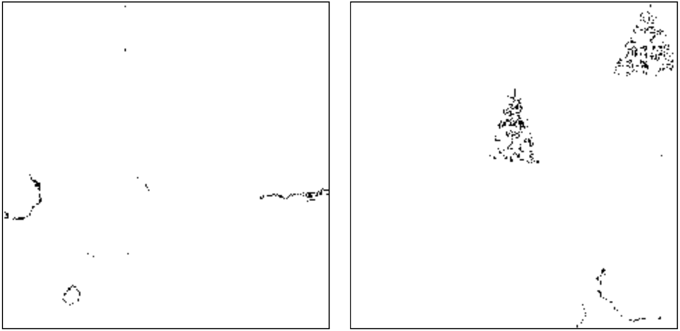
Signal separation

removal of strong edges and wavelet decomposition



(F.&al.TIP2008)

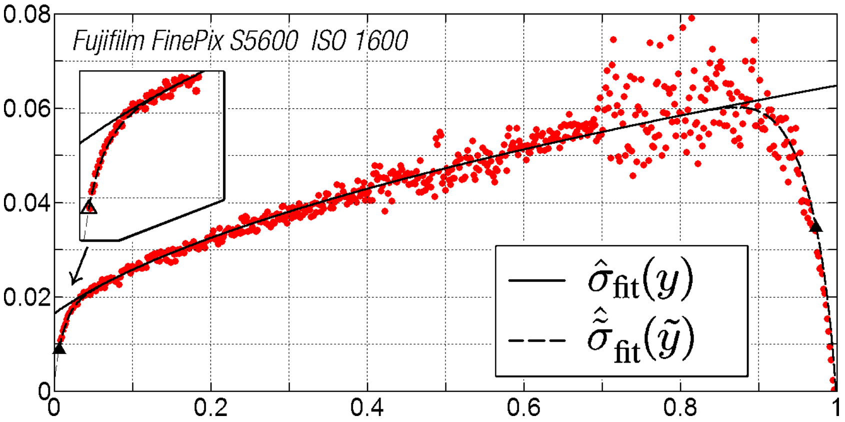
Codomain partitioning (level sets)



two level sets for different intervals of the codomain partition

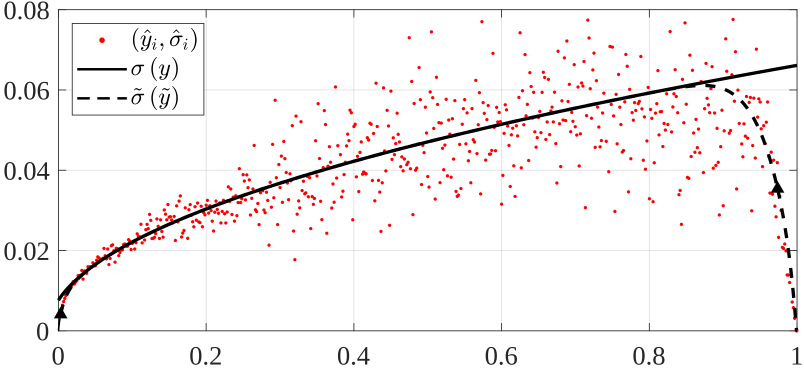
(F.&al.SensJ2007,F.&al.TIP2008)

Model does indeed fit the data
(Fujifilm FinePix)



(F.&al.TIP2008)

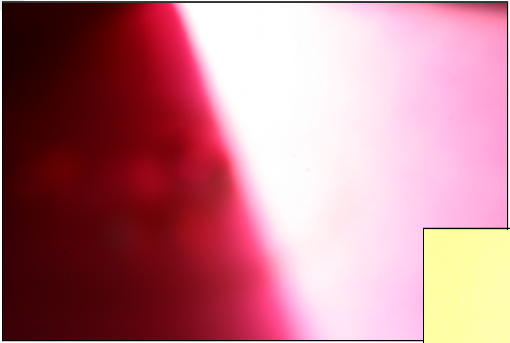
The model does indeed fit the data
(Samsung S8)

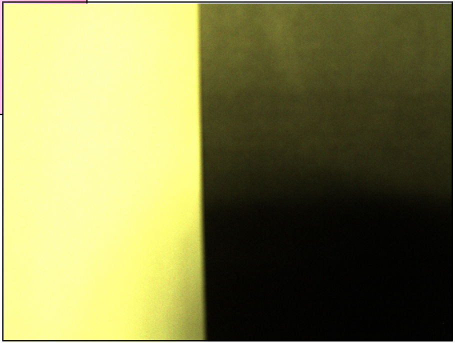


Noise standard deviation (black curve) $\sigma(y)$ estimated (Azzari&F., 2014) from one image of the 30 images from the dataset. We show also the estimate of the clipped standard deviation (dashed curve) $\tilde{\sigma}(\hat{y})$ and the scatterplot used for the fitting.
$\hat{a} = 4.315 \cdot 10^{-3}$, $\hat{b} = 5.814 \cdot 10^{-5}$.

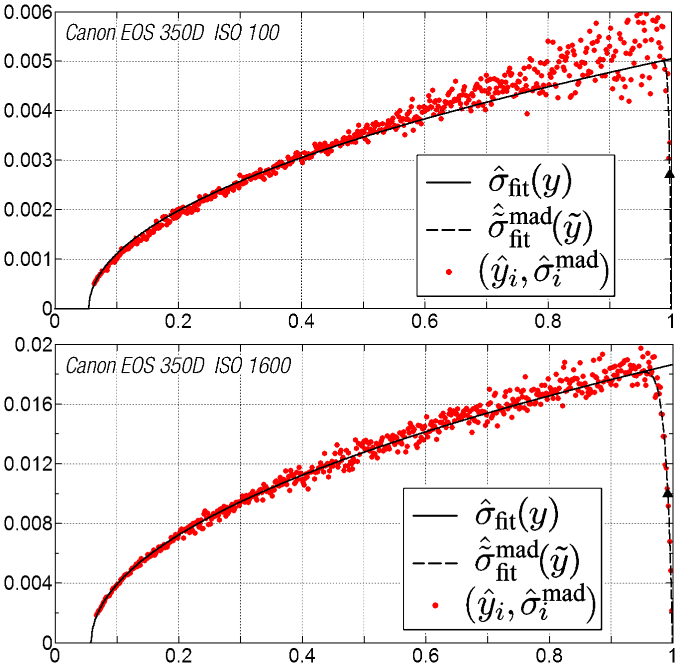
Noise estimation: easy examples

smooth targets with full codomain



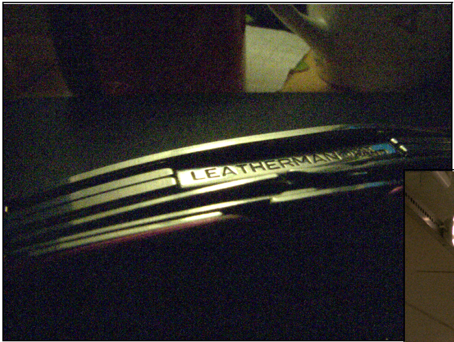


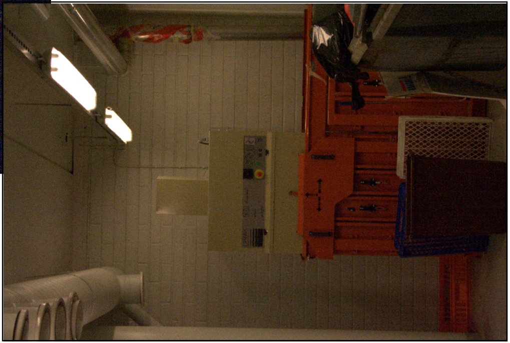
Noise estimation: easy examples



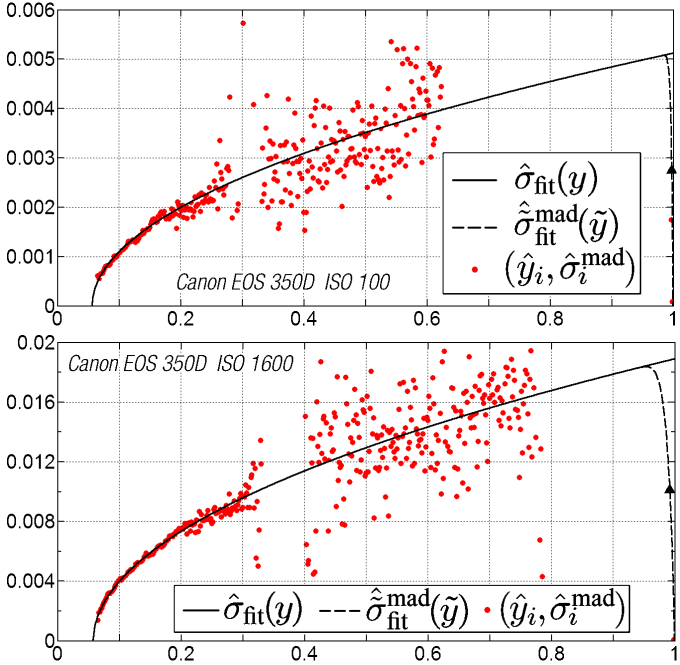
Importance of a good parametric model

complex targets with incomplete/sparse codomain

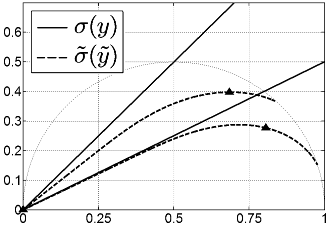




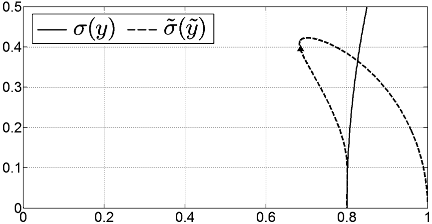
Importance of a good parametric model



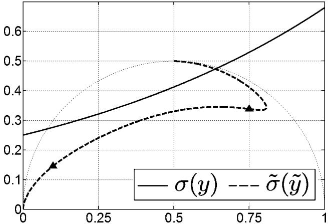
Limit cases and patologies



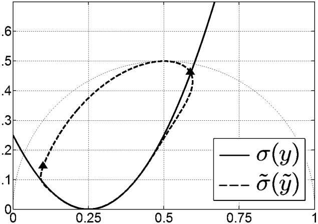
$\sigma(y) = \alpha y, \quad \alpha = 1, \frac{1}{2}$



$\sigma(y) = \sqrt{5y - 4}$



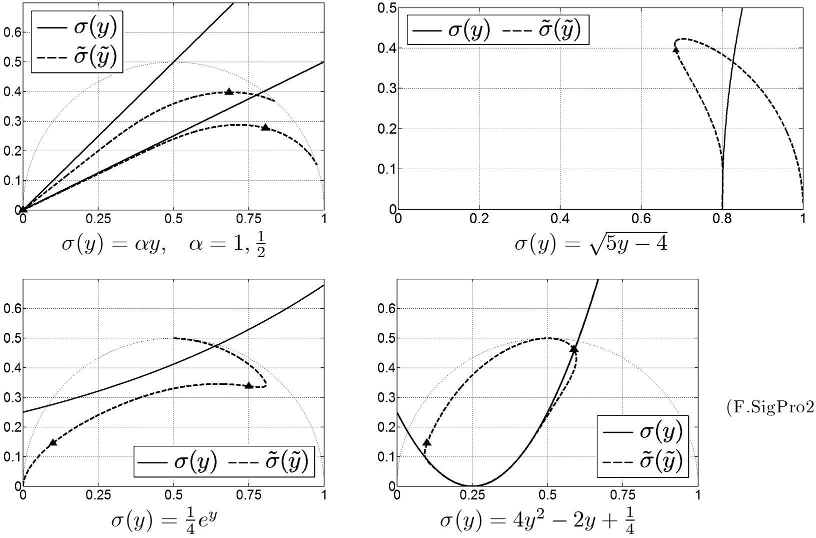
$\sigma(y) = \frac{1}{4} e^y$



$\sigma(y) = 4y^2 - 2y + \frac{1}{4}$

(F.SigPro2009)

Limit cases and patologies



(F.SigPro2009)

The semicircular envelope corresponds to $\{0,1\}$ binary distributions, which have mean $\tilde{y}$ and standard deviation $\sqrt{\tilde{y}\,(1-\tilde{y})}$.

Variance Stabilizing Transforms (VST)

Variance Stabilization: Motivation

Signal-dependent errors are particularly undesirable because

- basic data analysis and processing methods (such as those studied in earlier courses),
- standard statistical procedures implemented in computing environments (Matlab, R, Mathematica, etc.),
- off-the-shelf algorithms,

are typically designed and implemented for *identically distributed errors.*

Variance stabilization attempts to make the variance of the errors to be the same.

Variance-stabilization problem

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

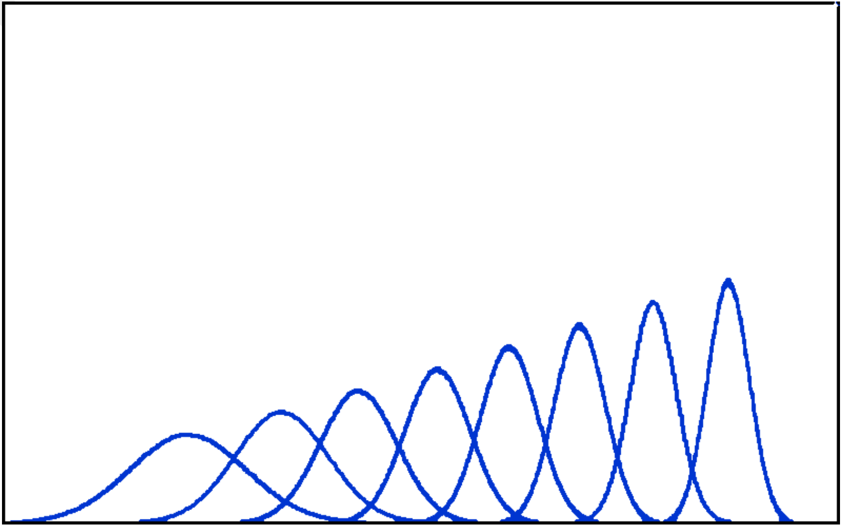
such f is a **variance-stabilizing transformation (VST)**

f should be independent of θ

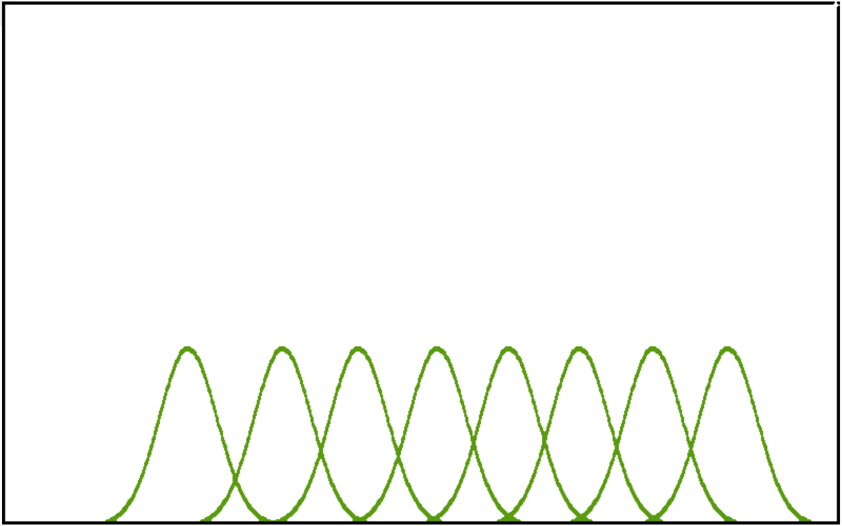
Benefits:

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

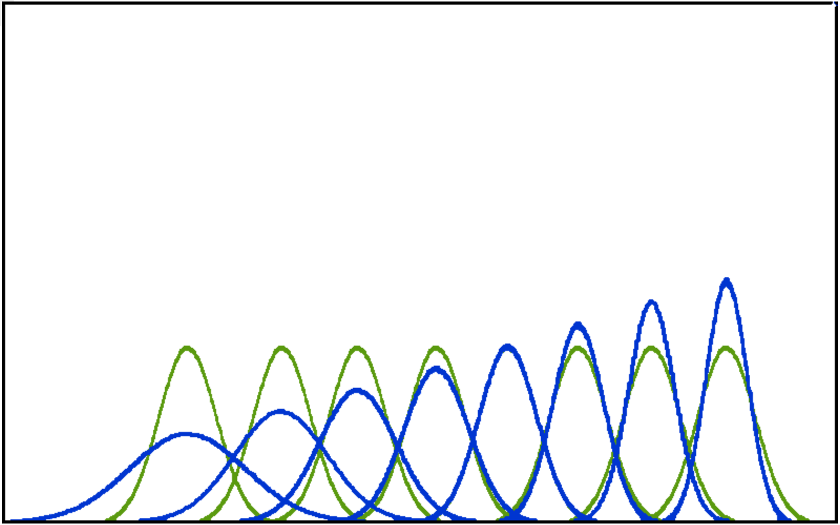
Variance Stabilization: Heuristics



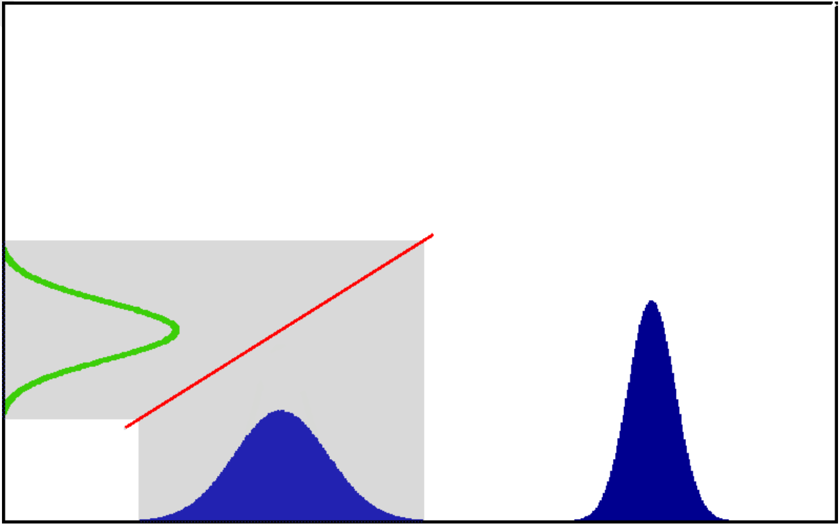
Variance Stabilization: Heuristics



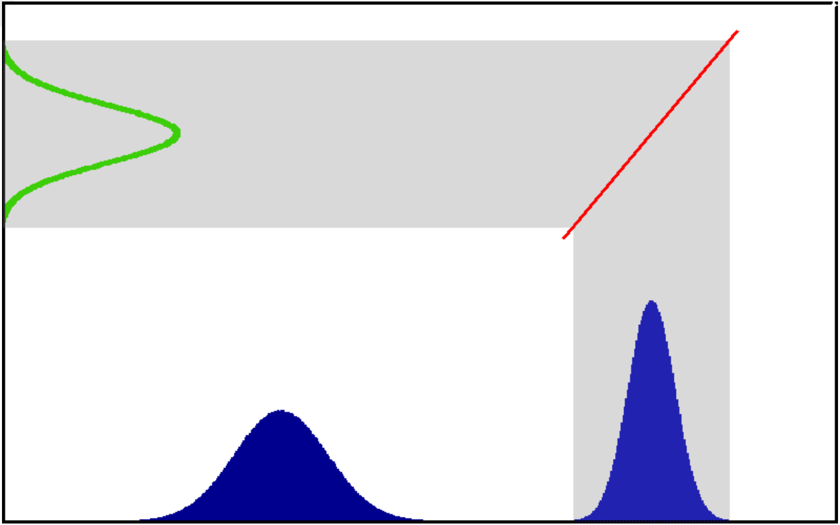
Variance Stabilization: Heuristics



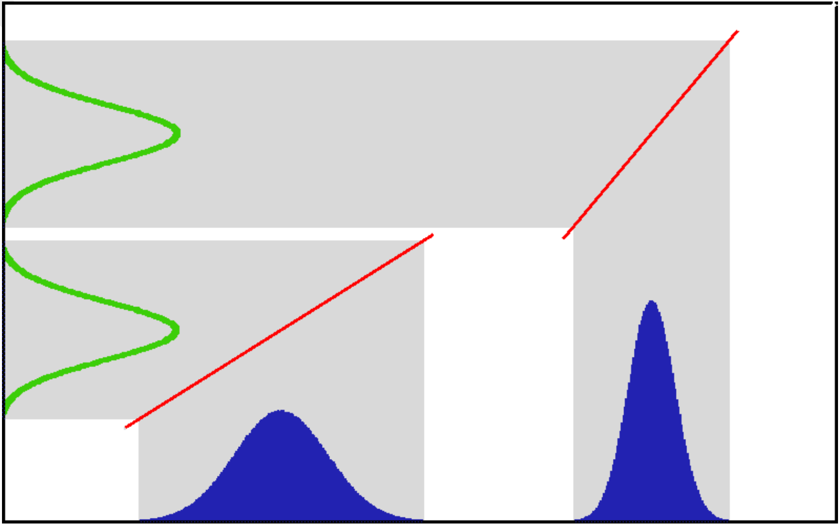
Variance Stabilization: Heuristics



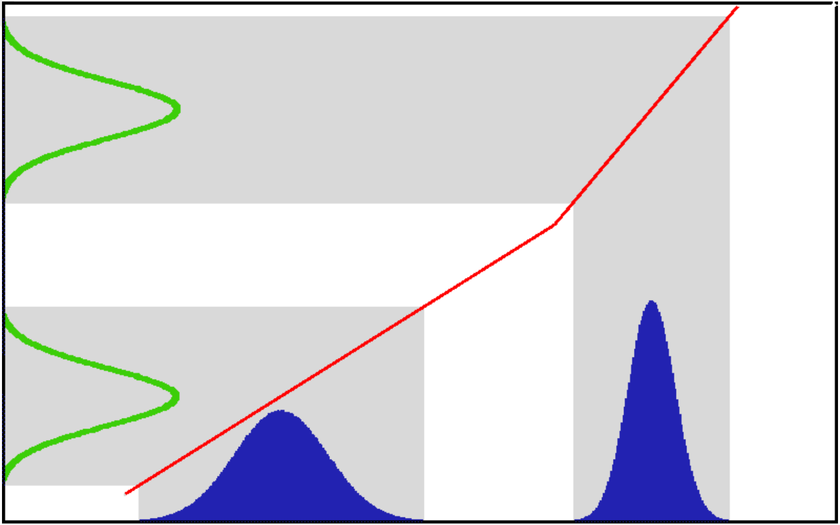
Variance Stabilization: Heuristics



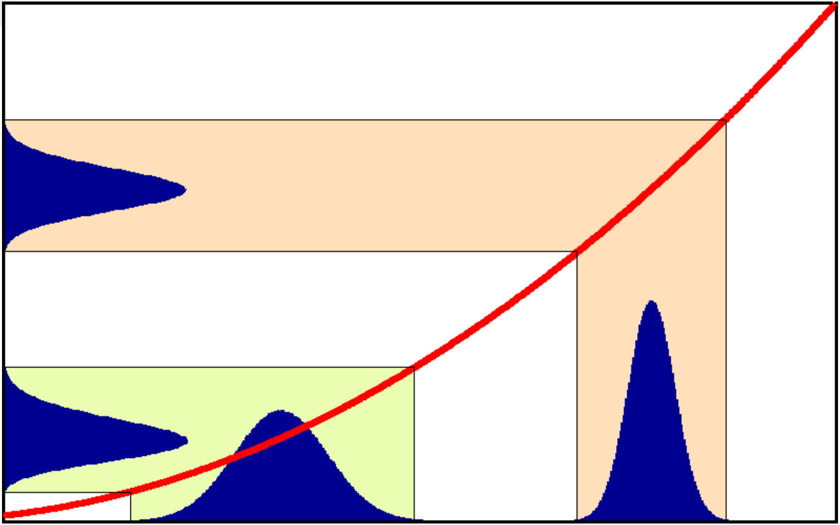
Variance Stabilization: Heuristics



Variance Stabilization: Heuristics



Variance Stabilization: Heuristics



Variance Stabilization: Heuristics

Classic **heuristic** stabilizer as indefinite integral form

$$f(z) = \int^{z} \frac{1}{\sigma(\theta)} d\mu(\theta). \tag{5}$$

Idea: consider a local first-order expansion of f at $\mu(\theta)$
(i.e., assume $\sigma(\theta)$ locally constant),

$$f(z) \simeq f(\mu(\theta)) + (z - \mu(\theta)) \frac{\partial f}{\partial z}(\mu(\theta)),$$

We have

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

then impose $\operatorname{std}\{f(z)|\theta\} = 1$ and obtain the indefinite integral (5).

Known and used already in the 1930's (e.g., Tippett 1934, Bartlett 1936), often rediscovered in signal processing (e.g., Prucnal&Saleh 1981, Arsenault&Denis 1981, Kasturi et al. 1983, Hirakawa&Parks 2006).

Very rough, but useful as a first guess: nearly all classical stabilizers can be seen as a slight modification of (5).

Exact variance stabilization is typically impossible to achieve

Positive result: multiplicative noise

$$f(z) = \log|z|$$

Negative result: Bernoulli

Binary samples $z \in \{0, 1\}$ of the Bernoulli distribution with parameter $\theta = E\{z|\theta\}$ cannot be stabilized to the same constant variance for different values of θ:

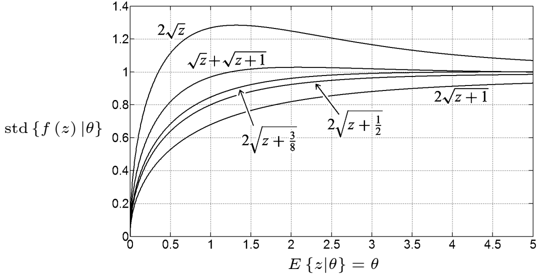
$$E\{g(z)|\theta\} = \theta g(1) + (1-\theta) g(0)$$

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

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

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

Classical variance stabilization for Poisson



$$f(z) = \int^z \frac{1}{\sigma(\theta)} d\mu(\theta) = \int^z \frac{1}{\sqrt{\theta}} d\mu(\theta) = 2\sqrt{z}.$$

Bartlett 1936: $2\sqrt{z+\frac{1}{2}}$

Anscombe 1948: $2\sqrt{z+\frac{3}{8}}$ (Anscombe attributes it to A.H.L. Johnson)

Freeman&Tukey 1950: $\sqrt{z}+\sqrt{z+1}$

In the same way stabilizers were derived for the Binomial and Negative Binomial distribution families ("angular" transformations based on the arcsin and hyperbolic arcsin).

Variance stabilization for Poisson and related

Murtagh, Starck, and Bijaoui, 1995: Generalized Anscombe transformation (GAT) for Poisson-Gaussian noise.

GAT is a family of VSTs parametrized by the Poisson gain α and the Gaussian std σ:

$$f_{\alpha,\sigma}(z) = \begin{cases} \frac{2}{\alpha}\sqrt{\alpha z + \frac{3}{8}\alpha^2 + \sigma^2}, & z \geq -\frac{3}{8}\alpha - \frac{\sigma^2}{\alpha} \\ 0, & z < -\frac{3}{8}\alpha - \frac{\sigma^2}{\alpha} \end{cases}.$$

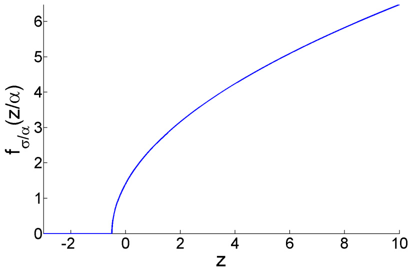
Asymptotically accurate stabilization for large y: $\operatorname{var}\{f_{\alpha,\sigma}(z) \mid y\} = 1 + \mathcal{O}(y^{-2})$
Poor stabilization for small y.

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

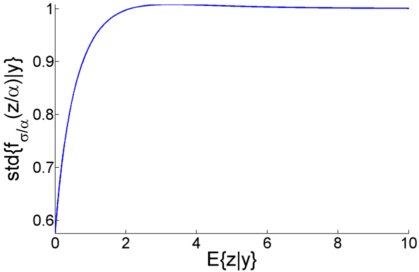
Zhang, Fadili, and Starck, 2008: Generalization of Anscombe for filtered (i.e. for linear combinations of) Poisson-Gaussian variates.

All these results enjoy some form of asymptotic optimality, but good stabilization for small θ is never achieved.

Generalized Anscombe transformation



(a) GAT for $\sigma = 0.357$ $(\alpha = 1)$



(b) Stabilized standard deviation. obtained with the GAT in (a)

Variance stabilization: three milestone works

- Curtiss 1943: general asymptotic theorems are proved (and later Bar-Lev&Enis 1990: alternative formulation)
  - gave theoretical support to empirical stabilizers that were already used (and also to others yet to appear).
- Efron 1981: existence of transformations for exact variance stabilization and/or perfect normalization.
  - formalizes sufficient conditions for existence of exact stabilizers ("general transformation families" framework), and provides their analytical expressions.
  - results are nonparametric and nonasymptotic.
  - difficult to use in practice (assumes too much smoothness and invertibility of parametrized mappings).
- Tibshirani 1986: AVAS procedure for regression
  - approximate variance stabilizing transformations are iteratively computed by recursive application of the integral stabilizer (iterative refinement of the stabilizer)
  - developed for data-driven application, hints about potential use for random variables.
  - nonparametric and nonasymptotic.

Exact stabilization for general transformation families (Efron 1981)

Exact stabilization is possible at least for some special classes of distribution families.

General scaled transformation family:

$$z = g^{-1}\left(p\left(\theta\right) + q\left(\theta\right) w\right),$$

where $w \sim \mathcal{N}(0,1)$ and g, p and q are smooth functions.

General transformation family has $q\left(\theta\right) \equiv q$.

Let z follow a general transformation family, pdf $[z|\theta]$ be the conditional p.d.f. of z, and $\vartheta\left(\theta\right) = \operatorname{med}\left\{z|\theta\right\}$ be the conditional median of z given θ. The *exact* VST f can be computed as:

$$f\left(z\right) = \int^{z} \frac{\text{pdf}\left[z|\theta\right]\left(\vartheta\right)}{\phi\left(0\right)} d\vartheta \qquad \text{(integration w.r.t. median)},$$

where ϕ is the p.d.f. of the standard normal $\mathcal{N}(0,1)$.

Optimization of VSTs: Motivation

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

- There is no universal objective criterion for assessing the goodness of a stabilizer. Simply demanding $\mathrm{std}\{f(z)|\theta\}$ to be as close as possible to 1 is vague and ambiguous.

- Common stabilizing transformations are often based on coarse asymptotics, developed between the 1930's and 1950's without leveraging any numerical optimization.

(F.2009)

Variance Stabilization as a minimization problem

Let

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

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

$$F(f) = \int |e_f(\theta)|\, d\theta. \tag{6}$$

We may formulate the variance stabilization problem as the solution of

$$\operatorname*{argmin}_f F(f) \tag{7}$$

Variance stabilization is exact only when $F(f) = 0$ for some f.

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

Variance Stabilization as a minimization problem

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

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

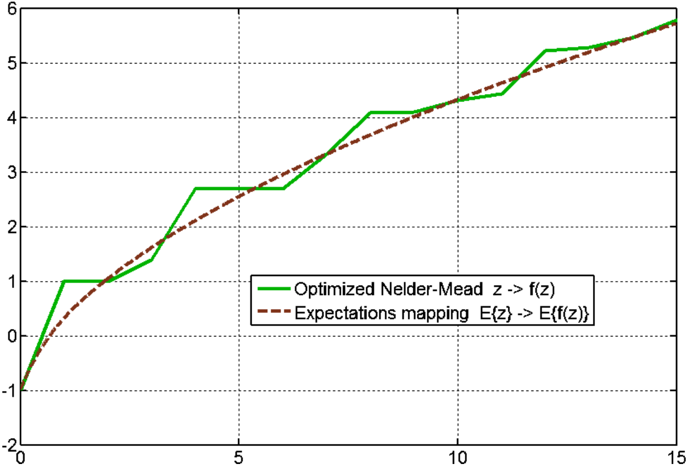
$$F(f) = \int_{\Theta} w_\theta(\theta)\, w_e(e_f(\theta))\, d\theta, \tag{8}$$

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

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

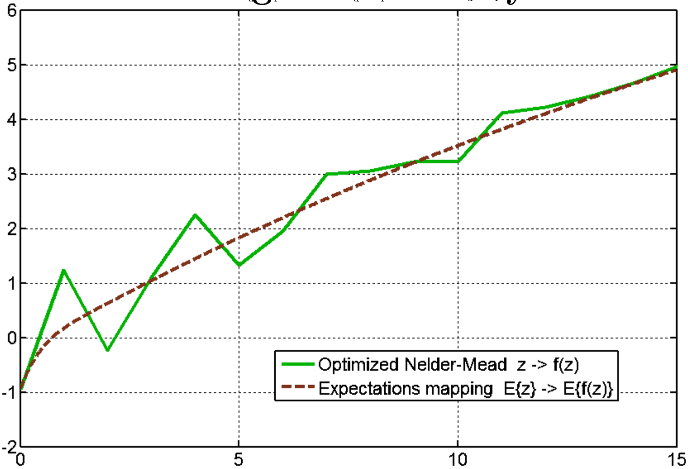
Optimization by direct search (F.2009)

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



variance-stabilizer f and the mapping $E\{z|\theta\} \mapsto E\{f(z)|\theta\}$
stabilization functional $F(f) = 0.096$

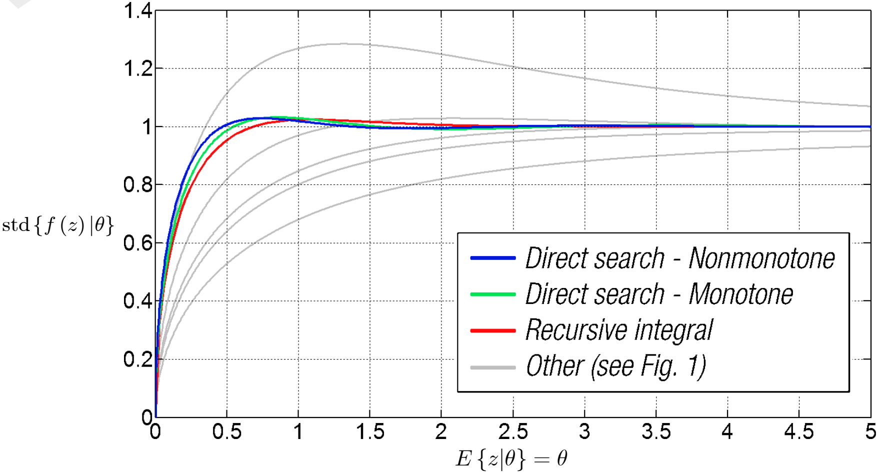
Optimization by direct search: relaxing monotonicity



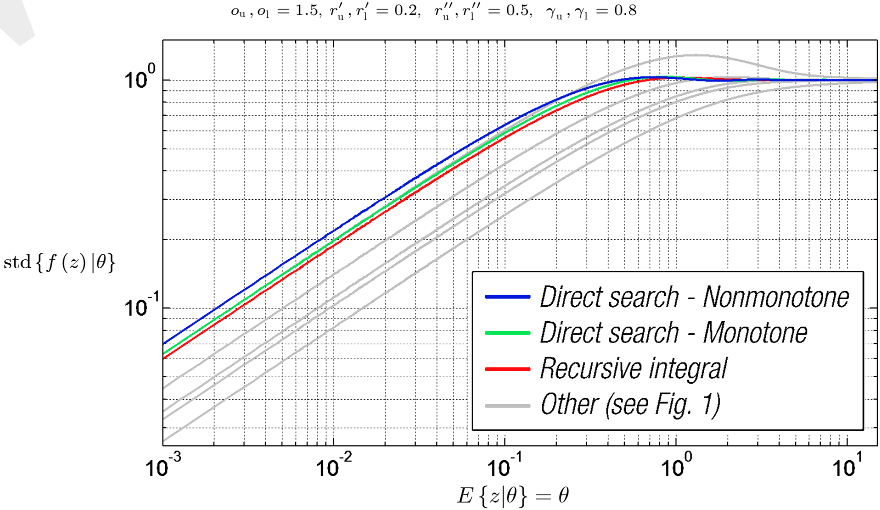
variance-stabilizer f and the mapping $E\{z|\theta\} \mapsto E\{f(z)|\theta\}$
stabilization functional $F(f) = 0.079$

Stabilization accuracy for Poisson data

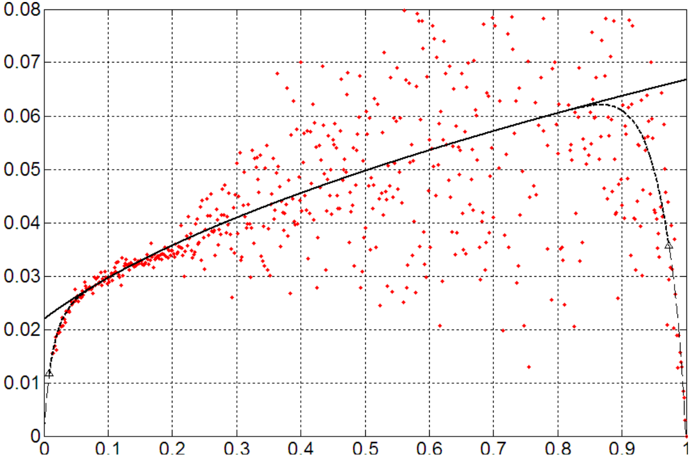
$o_u, o_l = 1.5,\ r'_u, r'_l = 0.2,\ \ r''_u, r''_l = 0.5,\ \ \gamma_u, \gamma_l = 0.8$



Stabilization accuracy for Poisson data



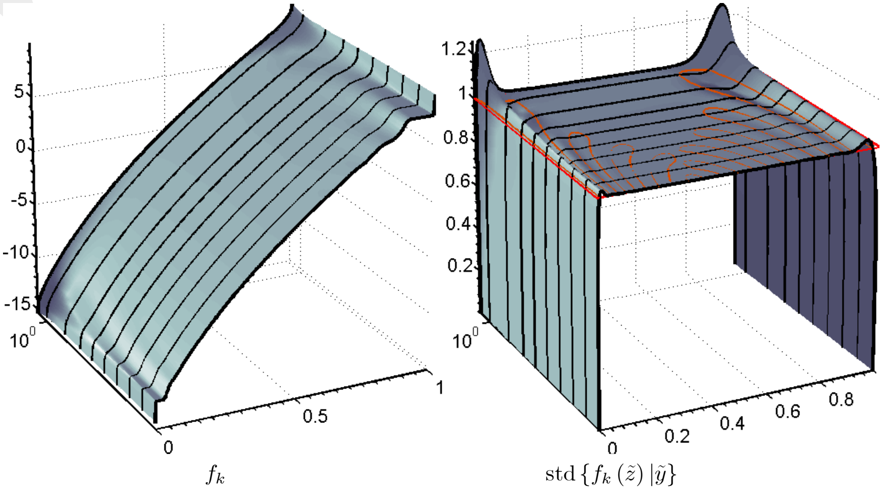
Optimization of VST for raw data (F.2009)



a = 0.003978 b =0.0004787

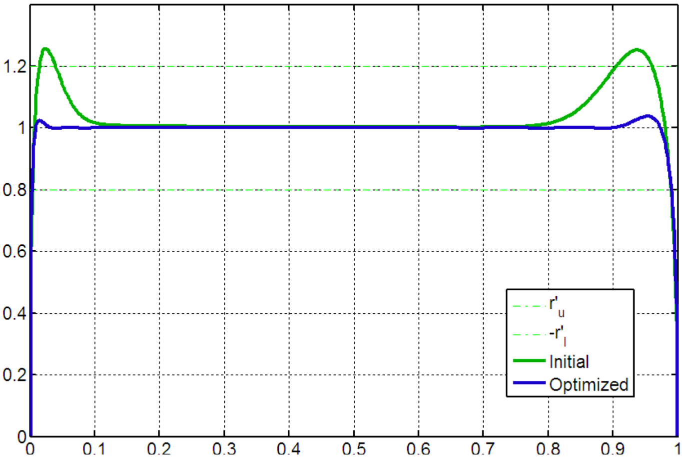
(F.&al.2008)

Optimization of VST for raw data (F.2009)



(F.2009)

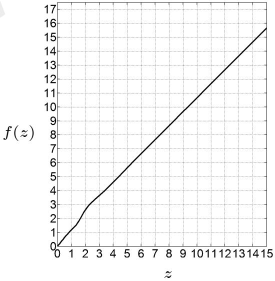
Optimization of VST for raw data (F.2009)

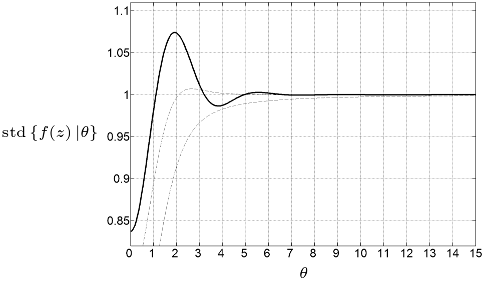


$\text{std}\{f_0(\tilde{z})|\tilde{y}\}$ vs $\text{std}\{f_{2000}(\tilde{z})|\tilde{y}\}$

(F.2009)

Optimization of VST for Rician data (F.2009)

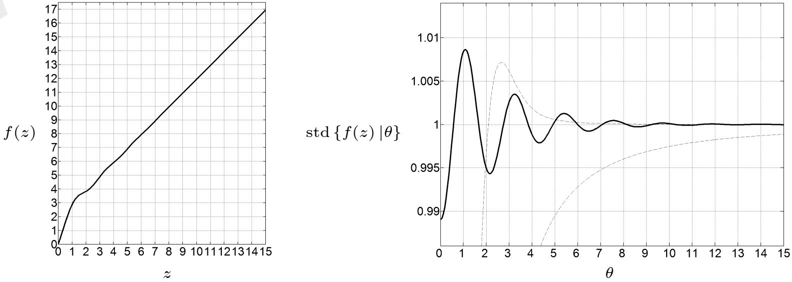




$$\lambda_{\text{asympt}} = 1,\ \lambda_{\text{smooth}} = 10^{-2},\ \lambda_{\text{inverse}} = 10^{-\frac{1}{2}}.$$

(F.ISBI2011)

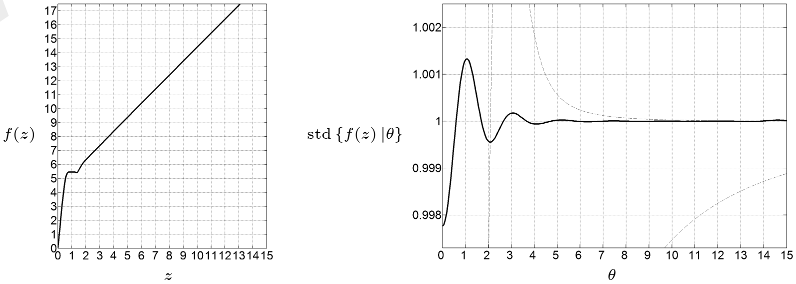
Optimization of VST for Rician data (F.2009)



$\lambda_{\text{asympt}} = 1$, $\lambda_{\text{smooth}} = 10^{-4}$, $\lambda_{\text{inverse}} = 0$.

(F.ISBI2011)

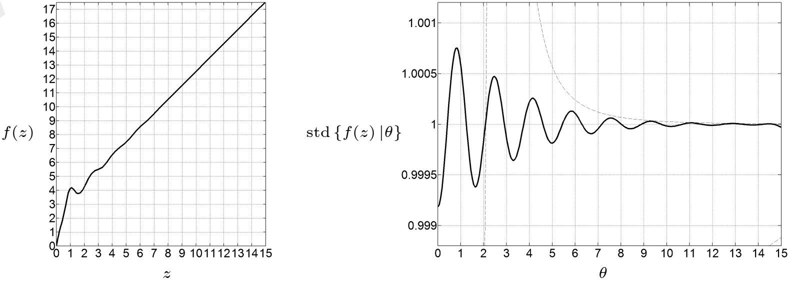
Optimization of VST for Rician data (F.2009)



$$\lambda_{\text{asympt}} = 1,\ \lambda_{\text{smooth}} = 10^{-6},\ \lambda_{\text{inverse}} = 10^{-\frac{5}{2}}.$$

(F.ISBI2011)

Optimization of VST for Rician data **(F.2009)**



$\lambda_{\text{asympt}} = 1$, $\lambda_{\text{smooth}} = 10^{-8}$, $\lambda_{\text{inverse}} = 0$.

(F.ISBI2011)

Optimization of rational polynomial VST

To effectively regularize the optimization, we can also seek the solution within a specific class of functions.

Poisson-Gaussian VST optimization

Find stabilizer by optimizing the coefficients of polynomials $P(z)$ and $Q(z)$ in

$$f_{1,\sigma}(z) = 2\sqrt{\frac{\sum_{i=0}^{N} p_i z^i}{\sum_{i=0}^{M} q_i z^i}} = 2\sqrt{\frac{P(z)}{Q(z)}}, \tag{9}$$

Constrain polynomials such that the VST necessarily approaches the GAT asymptotically. In this way, the optimized VST always attains good asymptotic stabilization:

$$\frac{P(z)}{Q(z)} - z - \frac{3}{8} - \sigma^2 \to 0 \text{ as } z \to +\infty \tag{10}$$

at a rate of $\mathcal{O}(z^{-1})$. For $N = 3$ we have

$$f_{1,\sigma}(z) = 2\sqrt{\frac{p_3 z^3 + p_2 z^2 + p_1 z + p_0}{p_3 z^2 + [p_2 - p_3 (3/8 + \sigma^2)] z + 1}}, \tag{11}$$

which depends solely on $\{p_i\}_{i=0}^{3}$.

(MF.TIP2014)

Optimization of rational polynomial VST for Poisson-Gaussian noise

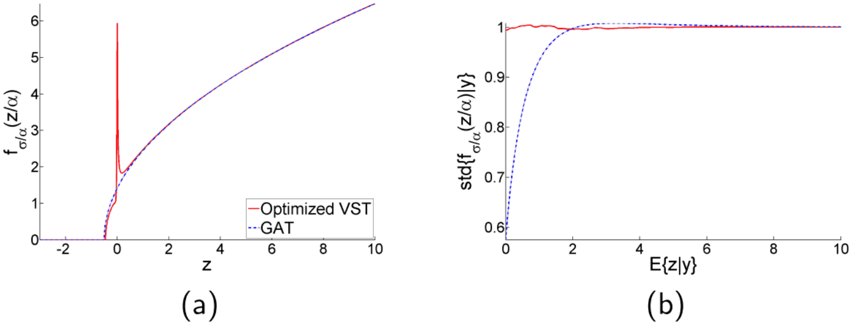


Figure: (a) Optimized rational VST $f_{1,\sigma}(z)$ and the GAT, for $\sigma = 0.357$ ($\alpha = 1$). (b) Stabilized standard deviation obtained with the VSTs in (a).

(MF.TIP2014)

Signal-dependent noise estimation via VST

Goal: estimate the standard-deviation function.

Idea: Different standard-deviation functions are typically stabilized by different VSTs: finding a VST that stabilizes the data can be equivalent to finding the standard-deviation function.

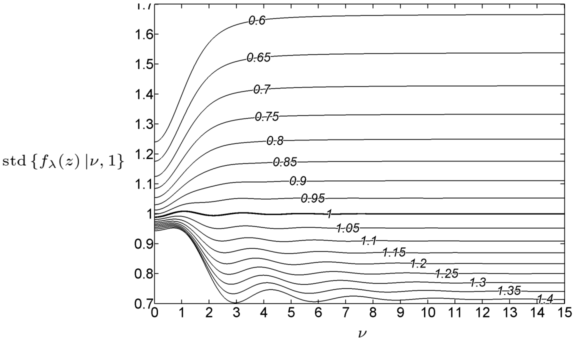
Challenges:

- stabilization is typically inaccurate even when the standard-deviation function is known;
- detecting noise-parameter mismatch

The generic algorithm iterates the following steps:

1. Apply VST $f_{\hat{\sigma}}$ based on current estimate $\hat{\sigma}$ of st.dev. function σ.
2. Assess stabilization of $f_{\hat{\sigma}}(z)$:
   If unable to improve stabilization further, the current $\hat{\sigma}$ is the final estimate; else, modify $\hat{\sigma}$ and go to 1.

Rice: Noise-level mismatch



(F.ISBI2011)

Standard deviation of the transformed data std $\{f_\lambda(z)\,|\nu,1\}$, for different values of λ, as indicated by the italic numbers superimposed on the curves. Stabilizer f on page 98.

The stabilizer f_λ is asymptotically affine for large z, with derivative approaching $\frac{1}{\lambda}$. Thus,

$$\operatorname{std}\{f_{\lambda\sigma}(z)\,|\sigma\nu,\sigma\} = \operatorname{std}\{f_\lambda(z)\,|\nu,1\} \underset{\nu\to+\infty}{\longrightarrow} \frac{1}{\lambda}. \qquad (12)$$

In other words, for large ν, *the stabilized standard deviation is approximately equal to the reciprocal of the under- or over-estimation factor.*

Rice: Noise-level estimation

General iterative scheme based on variance stabilization aimed at estimating the value of the σ parameter from a single realization z.

Let $\mathfrak{E}$ denote an estimator of the standard deviation σ of the homoskedastic noise corrupting a signal. Popular examples for estimating σ of AWGN in natural images are the median or mean absolute deviation of the high-pass filtered signal:

$$\mathfrak{E}_{\text{MedianAD}}\{z\} = \text{med}\{|H\{z\}|\}/\Phi^{-1}(3/4),$$

$$\mathfrak{E}_{\text{MeanAD}}\{z\} = \text{mean}\{|H\{z\}|\}\sqrt{\pi/2},$$

where $H\{z\} = z \circledast w_{\text{hi}}$, and w_{hi} is a high-pass convolutional kernel having zero mean and unit L^2-norm,

$$\int w_{\text{hi}} = 0, \qquad \int |w_{\text{hi}}|^2 = 1,$$

such as, e.g., a wavelet function.

(F.ISBI2011)

Rice: Iterative scheme for estimating σ

The proposed scheme is expressed by the following recursive system:

$$\begin{cases} \hat{\sigma}_0 &= \mathfrak{E}\{z\}, \\ \hat{\sigma}_{k+1} &= \mathfrak{E}\{f_{\hat{\sigma}_k}(z)\}\,\hat{\sigma}_k, \qquad k \geq 0. \end{cases} \tag{13}$$

The idea of this recursion originates from (12). The estimate $\hat{\sigma}_k$ is used to define a variance-stabilizing transformation for z. If the estimated value $\hat{\sigma}_k$ is correct, then the transformation $f_{\hat{\sigma}_k}$ successfully stabilizes the data and when $\mathfrak{E}$ is applied to the stabilized data it should return a value $\mathfrak{E}\{f_{\hat{\sigma}_k}(z)\}$ close to 1. If the estimated value $\hat{\sigma}_k$ is not correct (e.g., an under-estimate of σ), then the stabilization is not accurate, being roughly the inverse of the mis-estimation ratio, $\mathfrak{E}\{f_{\hat{\sigma}_k}(z)\} \approx \frac{\sigma}{\hat{\sigma}_k}$. Hence, we correct the current estimate $\hat{\sigma}_k$ by multiplying it with $\mathfrak{E}\{f_{\hat{\sigma}_k}(z)\}$. Observe that if $\mathfrak{E}\{f_{\hat{\sigma}}(z)\} = 1$ for some value $\hat{\sigma}$, then this $\hat{\sigma}$ is a fixed point for (13) and we want the sequence $\hat{\sigma}_k$ to converge to such $\hat{\sigma}$. The system (13) is initialized by the estimator $\mathfrak{E}$ applied on the non-stabilized data z.

Under very general conditions, the iterative scheme (13) is guaranteed to converge with exponential rate to an accurate and stable estimate $\hat{\sigma}$ of the true value σ.

(F.ISBI2011)

Standard-deviation contours in Poisson-Gaussian noise

Let $z_{\alpha,\sigma}$ be a Poisson-Gaussian image with (true) parameters α, σ.

Let B be an image block, with $p_B(y)$ being the probability density of y over this block.

Let $\hat{\alpha}, \hat{\sigma}$ be (possibly erroneous) estimates of α, σ.

Consider the VST $f_{\hat{\alpha},\hat{\sigma}}$ (such as GAT or an optimized VST).

Denote the average standard deviation of $f_{\hat{\alpha},\hat{\sigma}}(z_{\alpha,\sigma})$ over B as

$$F_B(\hat{\alpha}, \hat{\sigma}) := \mathfrak{E}_B\{f_{\hat{\alpha},\hat{\sigma}}(z_{\alpha,\sigma})\} = \int \operatorname{std}\{f_{\hat{\alpha},\hat{\sigma}}(z_{\alpha,\sigma})\,|y\}\, p_B(y)\, dy.$$

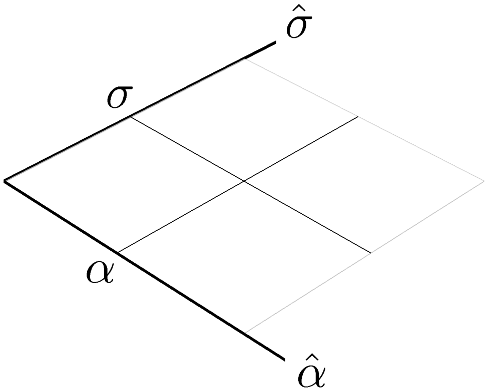
$F_B(\hat{\alpha}, \hat{\sigma})$ is a bivariate function of the parameter estimates $\hat{\alpha}, \hat{\sigma}$.

Under some simplifying assumptions, the unitary standard-deviation contours $F_B(\hat{\alpha}, \hat{\sigma}) = 1$ are smooth curves in a neighbourhood of the true parameter values (α, σ).

We apply the results by devising a VST-based algorithm for estimating α and σ.

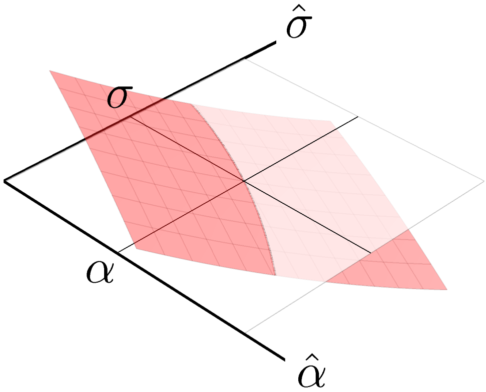
(M.&F.TIP2014)

$(\hat{\alpha}, \hat{\sigma})$ plane and the true parameters (α, σ)



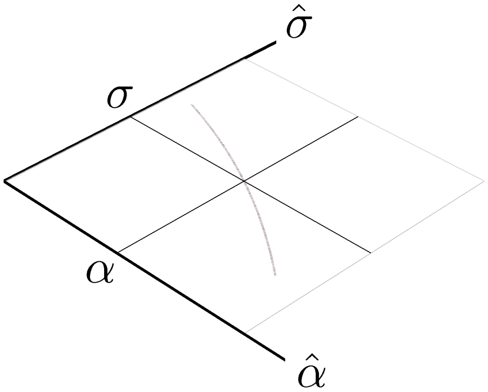
(M.&F.TIP2014)

$(\hat{\alpha}, \hat{\sigma})$ plane and $F_B(\hat{\alpha}, \hat{\sigma}) - 1$



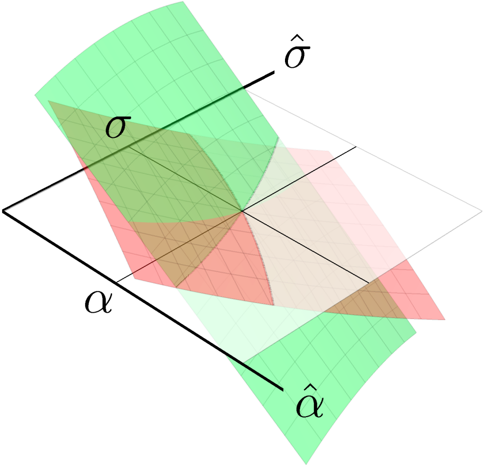
(M.&F.TIP2014)

Unitary contour of $F_B(\hat{\alpha}, \hat{\sigma})$



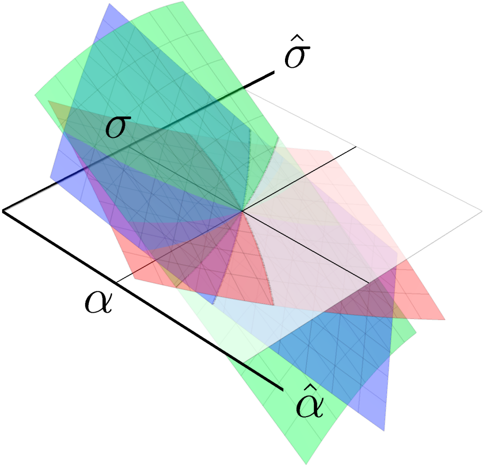
(M.&F.TIP2014)

$F_B(\hat{\alpha}, \hat{\sigma}) - 1$ for different blocks B



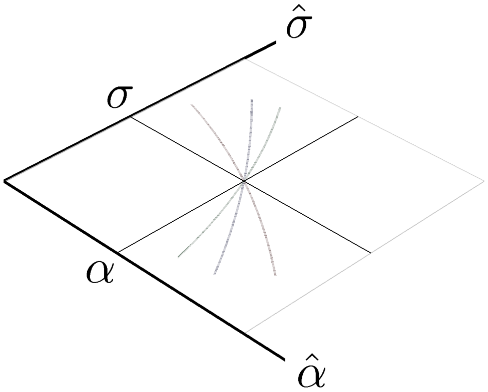
(M.&F.TIP2014)

$F_B(\hat{\alpha}, \hat{\sigma}) - 1$ for different blocks B



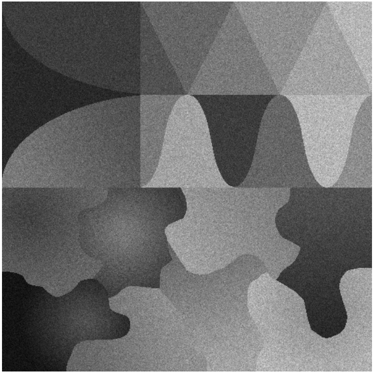
(M.&F.TIP2014)

Intersecting contours $F_B(\hat{\alpha}, \hat{\sigma}) = 1$

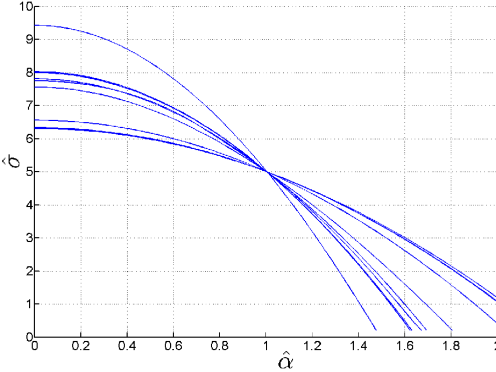


(M.&F.TIP2014)

Standard deviation contours: Example (GAT)



(a) peak 120, $\alpha = 1$, $\sigma = 5$



(b) GAT contours

Ten standard deviation contours $F_B(\hat{\alpha}, \hat{\sigma}) = 1$ computed from ten randomly selected 32×32 blocks B of the 512×512 image (a).

Standard deviation contours: Propositions

- We assume two ideal hypotheses:

1. We can achieve exact stabilization with the correct noise parameters θ:
$$\operatorname{std}\left\{f_{\alpha,\sigma}\left(z_{\alpha,\sigma}\right)|y\right\} = 1 \;\forall y \geq 0. \tag{14}$$

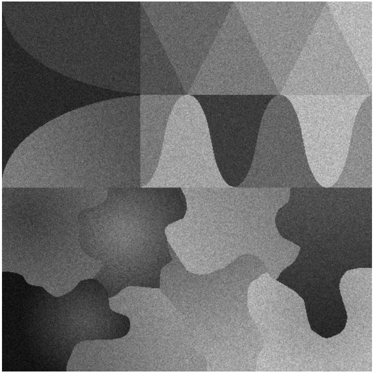
2. For any VST $f_{\hat{\alpha},\hat{\sigma}}$ and any choice of parameters $(\hat{\alpha},\hat{\sigma})$ and α,σ, the approximation
$$\operatorname{std}\left\{f_{\hat{\alpha},\hat{\sigma}}\left(z_{\alpha,\sigma}\right)|y\right\} \approx \operatorname{std}\left\{z_{\alpha,\sigma}|y\right\} f'_{\hat{\alpha},\hat{\sigma}}\left(E\left\{z_{\alpha,\sigma}|y\right\}\right) \tag{15}$$
holds exactly.

Proposition 1. The mean standard deviation of the stabilized image block $f_{\hat{\alpha},\hat{\sigma}}\left(z_{\alpha,\sigma}\right)$ can now be written as
$$\mathfrak{E}_B\left\{f_{\hat{\alpha},\hat{\sigma}}\left(z_{\alpha,\sigma}\right)\right\} = \int \frac{\operatorname{std}\left\{z_{\alpha,\sigma}|y\right\}}{\operatorname{std}\left\{z_{\hat{\alpha},\hat{\sigma}}|y\right\}} p_B\left(y\right) dy. \tag{16}$$

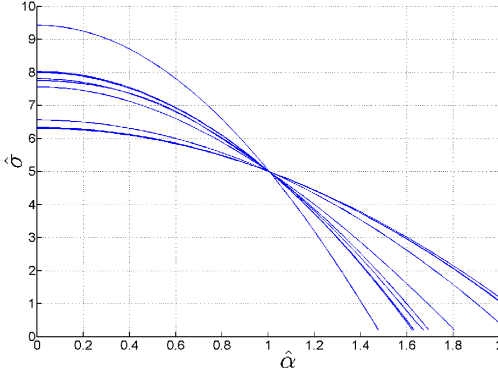
Proposition 2. Given the assumptions in Proposition 1, $F_B\left(\hat{\alpha},\hat{\sigma}\right)$ has a well-behaving (locally smooth and simple) unitary contour near the true parameter values α, σ.

(M.&F.TIP2014)

Standard deviation contours: Example (GAT)



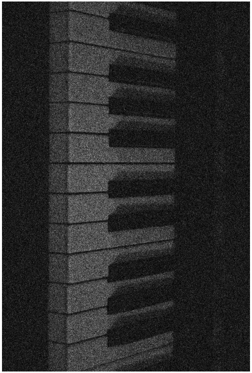
(a) peak 120, $\alpha = 1$, $\sigma = 5$



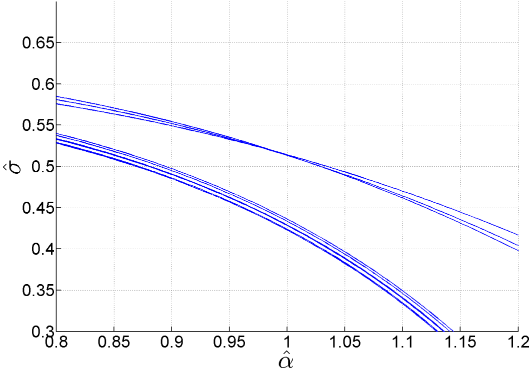
(b) GAT contours

Ten standard deviation contours $F_B(\hat{\alpha}, \hat{\sigma}) = 1$ computed from ten randomly selected 32×32 blocks B of the 512×512 image (a).

Standard deviation contours: Example (GAT)



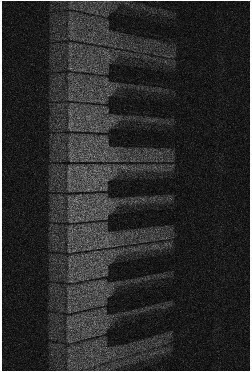
(a) peak 120, $\alpha = 1$, $\sigma = 5$



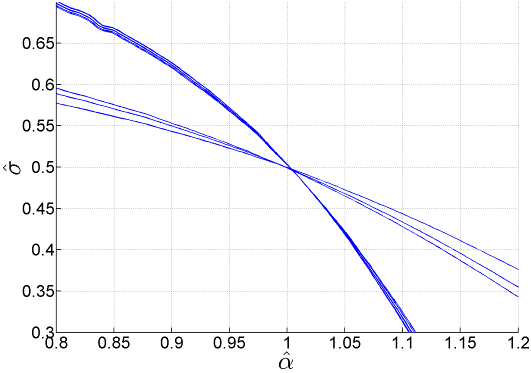
(b) GAT contours

Ten standard deviation contours $F_B(\hat{\alpha}, \hat{\sigma}) = 1$ computed from ten randomly selected 32×32 blocks B of the 1193×795 image (a).

Standard deviation contours: Example (Opt.VST)



(a) peak 120, $\alpha = 1$, $\sigma = 5$



(b) GAT contours

Ten standard deviation contours $F_B(\hat{\alpha}, \hat{\sigma}) = 1$ computed from ten randomly selected 32×32 blocks B of the 1193×795 image (a).

Application to parameter estimation

- The contours $F_B(\hat{\alpha}, \hat{\sigma}) = 1$ corresponding to different stabilized blocks B are locally smooth in the $(\hat{\alpha}, \hat{\sigma})$ plane.
- Typically different blocks yield differently oriented curves intersecting each other.
- The intersection has coordinates (α, σ), i.e. the true parameters.
- A cost functional measuring the lack of stabilization is minimized at the intersection.

(M.&F.TIP2014)

Parameter estimation algorithm

1. Initialize the estimates $\hat{\alpha}$ and $\hat{\sigma}$.
2. Choose M random blocks B_m, $m = 1, \ldots, M$ from the noisy image $z_{\alpha,\sigma}$.
3. Apply a VST $f_{\hat{\alpha},\hat{\sigma}}(z_{\alpha,\sigma})$ to each block.
4. Compute an estimate $F_{B_m}(\hat{\alpha}, \hat{\sigma}) = \mathfrak{E}_{B_m}\{f_{\hat{\alpha},\hat{\sigma}}(z_{\alpha,\sigma})\}$ for the standard deviation of each stabilized block, using any AWGN standard deviation estimator $\mathfrak{E}$.
5. Optimize $\hat{\alpha}$ and $\hat{\sigma}$ so to minimize the difference between $F_{B_m}(\hat{\alpha}, \hat{\sigma})^2$ and 1 (target variance) over the M blocks.

- We implement the proposed approach in Matlab, using the optimized VSTs (or GAT for comparison), and minimizing the cost functional

$$C(\hat{\alpha}, \hat{\sigma}) = \underset{m=1,\ldots,M}{\text{mean}} \left| F_{B_m}(\hat{\alpha}, \hat{\sigma})^2 - 1 \right|.$$

- $\mathfrak{E}$ is sample standard deviation of wavelet detail coefficients.
- We estimate $F_{B_m}(\hat{\alpha}, \hat{\sigma})$ from $M = 2000$ randomly selected 32×32 image blocks.

(M.&F.TIP2014)

Experiments

Root Histogram-Weighted Normalized MSE (RHWNMSE) : $\sqrt{\int_{\mathbb{R}^+} p(\xi) \frac{\left(\sqrt{\alpha^2\xi+\sigma^2}-\sqrt{\hat{\alpha}^2\xi+\hat{\sigma}^2}\right)^2}{\alpha^2\xi+\sigma^2} d\xi}$

Table: Average RHWNMSE (± std) over 10 noise realizations for *Piano* image:

Peak	α	σ	Opt. VST	GAT	Scatterplot
2	0.5	0.2	0.042 ± 0.002	0.286 ± 0.008	$\mathbf{0.024 \pm 0.009}$
2	2.5	0.2	$\mathbf{0.007 \pm 0.005}$	0.676 ± 0.007	0.056 ± 0.016
10	0.5	1.0	$\mathbf{0.006 \pm 0.003}$	0.021 ± 0.002	0.011 ± 0.007
10	2.5	1.0	$\mathbf{0.005 \pm 0.004}$	0.013 ± 0.005	0.016 ± 0.008
30	0.5	3.0	$\mathbf{0.006 \pm 0.003}$	$\mathbf{0.006 \pm 0.003}$	0.016 ± 0.007
30	2.5	3.0	$\mathbf{0.005 \pm 0.003}$	0.008 ± 0.002	0.014 ± 0.006

- Combined with the optimized VSTs, the algorithm yields results that are competitive with the results obtained with scatterplot method (Foi et al., 2008).
- The optimized VSTs plays an important role in the estimation performance for the low-intensity cases.
  - The GAT is inherently unable to accurately stabilize regions with low mean intensity; this violates our assumption that std $\{f_\theta(z_\theta)\,|y\} = 1\ \forall y \geq 0$.
  - Optimized VSTs provide highly accurate stabilization also for low intensities.

VST-based Denoising and
the Exact Unbiased Inverse

Three steps: stabilization, denoising, and inversion

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 $E\{f(z)|\theta\}$.

Exact unbiased inverse (M.&F.TIP2011)

Since f is necessarily a nonlinear mapping, we may have

$$E\{f(z)|\theta\} \neq f(E\{z|\theta\}),$$

and, thus,

$$f^{-1}(E\{f(z)|\theta\}) \neq E\{z|\theta\},$$

which means that the inverse transformation applied after denoising (Step 3.) should not coincide with the algebraic inverse of f, as this would introduce bias in the estimation of $E\{z|\theta\}$ from the observed z.

Exact unbiased inverse (M.&F.TIP2011)

Since f is necessarily a nonlinear mapping, we may have

$$E\{f(z)|\theta\} \neq f(E\{z|\theta\}),$$

and, thus,

$$f^{-1}(E\{f(z)|\theta\}) \neq E\{z|\theta\},$$

which means that the inverse transformation applied after denoising (Step 3.) should not coincide with the algebraic inverse of f, as this would introduce bias in the estimation of $E\{z|\theta\}$ from the observed z. The problem of bias in variance-stabilized denoising is solved by the *exact unbiased inverse* that is defined by the mapping

$$\mathcal{I}_f : E\{f(z)|\theta\} \longmapsto E\{z|\theta\} = \mu.$$

This definition assumes that the mapping $E\{z|\theta\} \mapsto E\{f(z)|\theta\}$ is invertible. In particular, we require this mapping to be strictly increasing, or, equivalently, that $E\{f(z)|\theta\}$ is strictly increasing with θ. This condition supplants the traditional requirement of invertibility of f, which instead we may allow to be nonmonotone.

Exact unbiased inverse (M.&F.TIP2011)

Since f is necessarily a nonlinear mapping, we may have

$$E\{f(z)|\theta\} \neq f(E\{z|\theta\}),$$

and, thus,

$$f^{-1}(E\{f(z)|\theta\}) \neq E\{z|\theta\},$$

which means that the inverse transformation applied after denoising (Step 3.) should not coincide with the algebraic inverse of f, as this would introduce bias in the estimation of $E\{z|\theta\}$ from the observed z. The problem of bias in variance-stabilized denoising is solved by the *exact unbiased inverse* that is defined by the mapping

$$\mathcal{I}_f : E\{f(z)|\theta\} \longmapsto E\{z|\theta\} = \mu.$$

This definition assumes that the mapping $E\{z|\theta\} \mapsto E\{f(z)|\theta\}$ is invertible. In particular, we require this mapping to be strictly increasing, or, equivalently, that $E\{f(z)|\theta\}$ is strictly increasing with θ. This condition supplants the traditional requirement of invertibility of f, which instead we may allow to be nonmonotone.

$\mathcal{I}_f(D)$ is a ML estimate of θ under modest hypotheses.

Inversion for Poisson stabilized by Anscombe

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

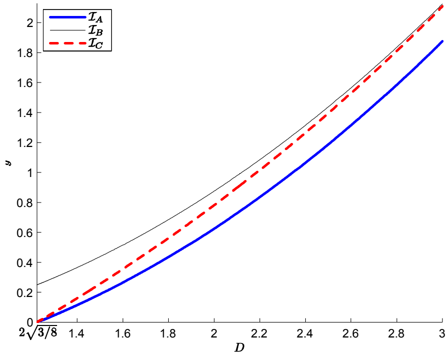
We have discrete Poisson probabilities $P(z \mid y)$, so

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

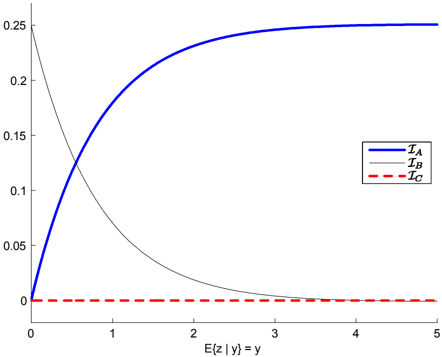
The definition of $\mathcal{I}_C$ is implicit, but we can have a closed form approximation as

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

Inversion for Poisson stabilized by Anscombe



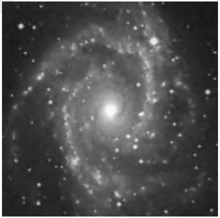
inverse transformations

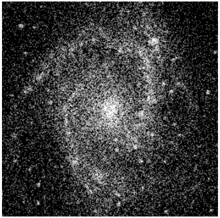


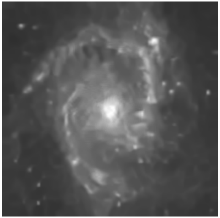
bias

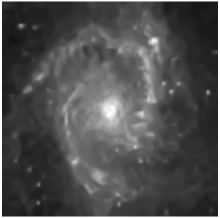
(M.&F.TIP2011)

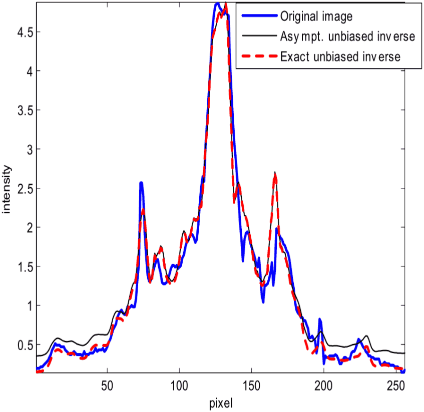
Inversion for Poisson stabilized by Anscombe


original


noisy


Ansc.+BM3D+Asy.Unb.Inv.


Ansc.+BM3D+Ex.Unb.Inv.



Cross section

Exact unbiased inverse of Generalized Anscombe Transform for Poisson-Gaussian noise

Without loss of generality, we can fix $\alpha = 1$ and use scaling for $\alpha \neq 1$.

The EUI of GAT is constructed analogous to the EUI of the Anscombe transformation:

$$\mathcal{I}_\sigma : E\left\{f_\sigma\left(z\right) \mid y, \sigma\right\} \longmapsto E\left\{z \mid y, \sigma\right\}.$$

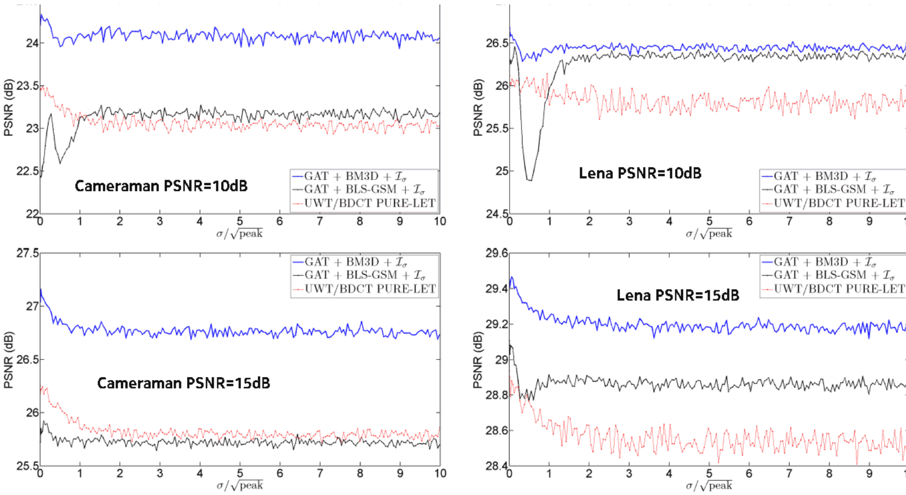
$$E\{f_\sigma\left(z\right) \mid y, \sigma\} = \int_{-\infty}^{+\infty} f_\sigma\left(z\right) p\left(z \mid y, \sigma\right)\, dz =$$

$$= \int_{-\infty}^{+\infty} 2\sqrt{z + \frac{3}{8} + \sigma^2} \sum_{k=0}^{+\infty} \left(\frac{y^k e^{-y}}{k!\sqrt{2\pi\sigma^2}} e^{-\frac{(z-k)^2}{2\sigma^2}}\right) dz.$$

Closed form approximation:

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

Consistency of GAT+EUI at fixed input PSNR from pure Gaussian to pure Poisson



Correlated Signal-Dependent Noise Model (1/2)

Model 1. Noise Scaling Post Correlation

$$\begin{aligned} &z(x) = y(x) + \sigma(y(x))\,\eta(x), \\ &\eta = \nu \circledast g, \qquad \nu(\cdot) \sim \mathcal{N}(0,1), \qquad \sigma : y \to \mathbb{R}^{+}, \end{aligned}$$

where σ is a generic standard deviation function.

Correlated Signal-Dependent Noise Model (1/2)

Model 1. Noise Scaling Post Correlation

$$z(x) = y(x) + \sigma(y(x))\,\eta(x),$$
$$\eta = \nu \circledast g, \qquad \nu(\cdot) \sim \mathcal{N}(0,1), \qquad \sigma : y \to \mathbb{R}^{+},$$

where σ is a generic standard deviation function.

$$\begin{aligned} \mathrm{E}\{z\} =& y, \\ \operatorname{var}\{z\} =& \operatorname{var}\{\sigma(y)\,\nu \circledast g\} = \sigma^2(y)\operatorname{var}\{\nu \circledast g\} = \\ & \sigma^2(y)\,\|g\|_2^2 = \sigma^2(\mathrm{E}\{z\})\,\|g\|_2^2\,. \end{aligned}$$

Correlated Signal-Dependent Noise Model (1/2)

Model 1. Noise Scaling Post Correlation

$$z(x) = y(x) + \sigma(y(x))\,\eta(x),$$
$$\eta = \nu \circledast g, \qquad \nu(\cdot) \sim \mathcal{N}(0,1)\,, \qquad \sigma : y \to \mathbb{R}^+,$$

where σ is a generic standard deviation function.

$$\begin{aligned} \mathrm{E}\{z\} =& y, \\ \mathrm{var}\{z\} =& \mathrm{var}\{\sigma(y)\,\nu \circledast g\} = \sigma^2(y)\,\mathrm{var}\{\nu \circledast g\} = \\ & \sigma^2(y)\,\|g\|_2^2 = \sigma^2(\mathrm{E}\{z\})\,\|g\|_2^2\,. \end{aligned}$$

$$\text{PSD} \qquad \mathrm{var}\{\mathcal{F}[z]\} \approx |\mathcal{F}[g]|^2 \left\|\sigma^2(y)\right\|_1\,.$$

Correlated Signal-Dependent Noise Model (2/2)

Model 2. Noise Scaling Prior to Correlation

$$z'(x) = y(x) + \sigma(y(x))\,\nu(x),$$
$$z(x) = (z' \circledast g)\,(x).$$

Correlated Signal-Dependent Noise Model (2/2)

Model 2. Noise Scaling Prior to Correlation

$$z'(x) = y(x) + \sigma(y(x))\,\nu(x),$$
$$z(x) = (z' \circledast g)\,(x).$$

$$\mathrm{E}\{z\} \approx \mathrm{E}\{z'\}\,\|g\|_1 = y\,\|g\|_1\,,$$
$$\mathrm{var}\{z\} \approx \mathrm{var}\{z'\}\,\|g\|_2^2 = \sigma^2(y(x))\,\|g\|_2^2\,,$$

where the approximations become accurate in large smooth areas of the image where the intensity changes gradually.

Correlated Signal-Dependent Noise Model (2/2)

Model 2. Noise Scaling Prior to Correlation

$$z'(x) = y(x) + \sigma(y(x))\,\nu(x),$$
$$z(x) = (z' \circledast g)(x).$$

$$\mathrm{E}\{z\} \approx \mathrm{E}\{z'\}\,\|g\|_1 = y\,\|g\|_1\,,$$
$$\mathrm{var}\{z\} \approx \mathrm{var}\{z'\}\,\|g\|_2^2 = \sigma^2(y(x))\,\|g\|_2^2\,,$$

where the approximations become accurate in large smooth areas of the image where the intensity changes gradually.

$$\mathrm{var}\{z\} \approx \sigma^2\left(\frac{\mathrm{E}\{z\}}{\|g\|_1}\right)\|g\|_2^2$$

Correlated Signal-Dependent Noise Model (2/2)

Model 2. Noise Scaling Prior to Correlation

$$z'(x) = y(x) + \sigma(y(x))\,\nu(x),$$
$$z(x) = (z' \circledast g)(x).$$

$$\mathrm{E}\{z\} \approx \mathrm{E}\{z'\}\,\|g\|_1 = y\,\|g\|_1\,,$$
$$\mathrm{var}\{z\} \approx \mathrm{var}\{z'\}\,\|g\|_2^2 = \sigma^2(y(x))\,\|g\|_2^2\,,$$

where the approximations become accurate in large smooth areas of the image where the intensity changes gradually.

$$\mathrm{var}\{z\} \approx \sigma^2\left(\frac{\mathrm{E}\{z\}}{\|g\|_1}\right)\|g\|_2^2$$

PSD $$\mathrm{var}\{\mathcal{F}[z]\} = |\mathcal{F}[g]|^2\,\left\|\sigma^2(y)\right\|_1 \approx |\mathcal{F}[g]|^2\left\|\sigma^2\left(\frac{\mathrm{E}\{z\}}{\|g\|_1}\right)\right\|_1.$$

Correlated Signal-Dependent Noise Model (2/2)

Model 2. Noise Scaling Prior to Correlation

$$z'(x) = y(x) + \sigma(y(x))\,\nu(x),$$
$$z(x) = (z' \circledast g)(x).$$

$$\mathrm{E}\{z\} \approx \mathrm{E}\{z'\}\,\|g\|_1 = y\,\|g\|_1\,,$$
$$\mathrm{var}\{z\} \approx \mathrm{var}\{z'\}\,\|g\|_2^2 = \sigma^2(y(x))\,\|g\|_2^2\,,$$

where the approximations become accurate in large smooth areas of the image where the intensity changes gradually.

$$\mathrm{var}\{z\} \approx \sigma^2\left(\frac{\mathrm{E}\{z\}}{\|g\|_1}\right)\|g\|_2^2$$

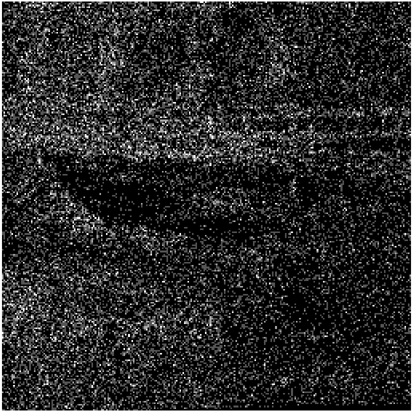
PSD $$\mathrm{var}\{\mathcal{F}[z]\} = |\mathcal{F}[g]|^2\,\left\|\sigma^2(y)\right\|_1 \approx |\mathcal{F}[g]|^2\left\|\sigma^2\left(\frac{\mathrm{E}\{z\}}{\|g\|_1}\right)\right\|_1.$$

Thus, both Model 1 and Model 2 express the variance of z as a function of its expectation, where the main differences consist merely in a scaling of the variables, and this scaling is determined by the ℓ_1 and ℓ_2 norms of the convolution kernel g.

Case studies

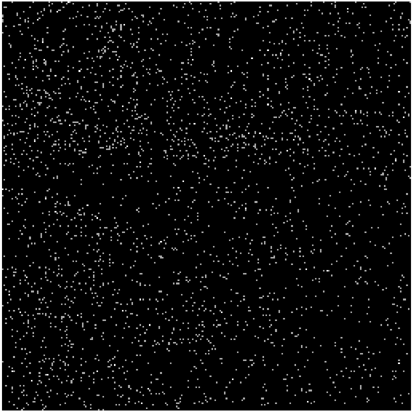
Efficient Denoising and Deblurring of Extremely Low-Energy Images Using Off-the-Shelf Gaussian Filters

Noisy Poisson image - peak 1



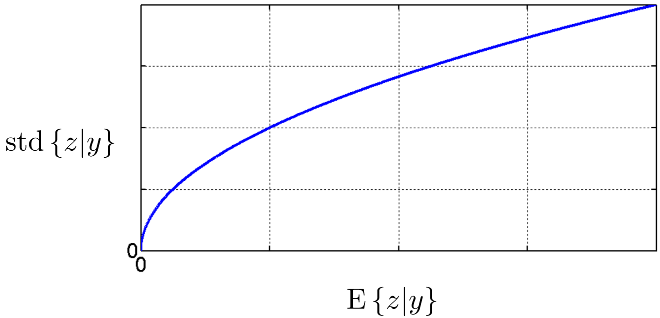
z when peak of y is 1

Noisy Poisson image - peak 0.1



z when peak of y is 0.1

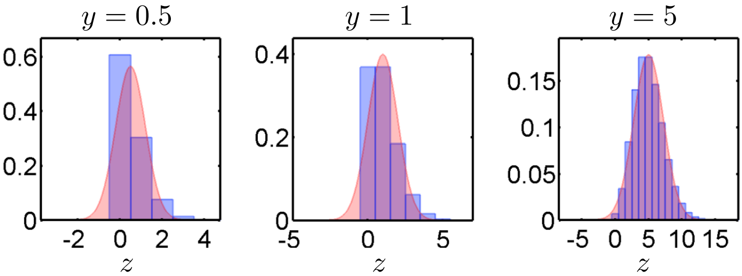
Signal-to-noise ratio



$$\text{noise st.dev} = \sqrt{\text{noise-free signal}}$$

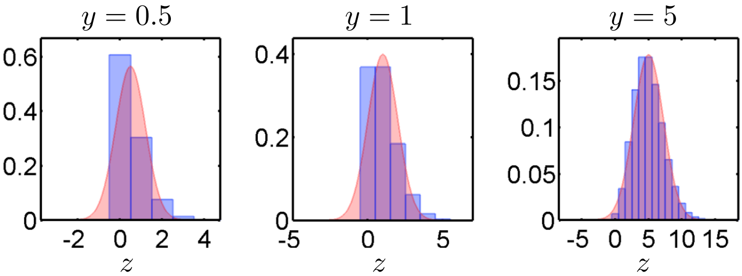
- Noise is relatively stronger at lower intensities
- SNR $\to 0$ as the intensity decreases.

Photon-Limited Imaging



Discrete Poisson $\mathcal{P}(y)$ vs continuous normal $\mathcal{N}(y, y)$

Photon-Limited Imaging

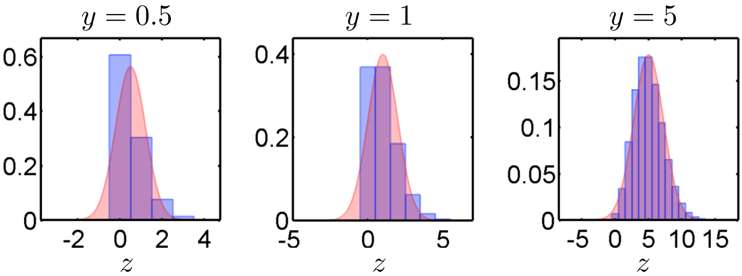


Discrete Poisson $\mathcal{P}(y)$ vs continuous normal $\mathcal{N}(y, y)$

- We are interested in cases where

$$\text{peak intensity} = [0.1, 4], \text{ i.e. SNR} < [0.3, 2]$$

Photon-Limited Imaging



Discrete Poisson $\mathcal{P}(y)$ vs continuous normal $\mathcal{N}(y, y)$

- We are interested in cases where
  $$\text{peak intensity} = [0.1, 4], \text{ i.e. SNR} < [0.3, 2]$$
- Only a couple of counts per pixel: *photon-limited imaging*

Variance stabilization

- Nonlinear 1-D mapping to make the noise variance invariant with respect to the noise-free signal: Variance-Stabilizing Transformation (VST).

Variance stabilization

- Nonlinear 1-D mapping to make the noise variance invariant with respect to the noise-free signal: Variance-Stabilizing Transformation (VST).
  - conditional variance no longer depends on unknown distribution parameter

Variance stabilization

- Nonlinear 1-D mapping to make the noise variance invariant with respect to the noise-free signal: Variance-Stabilizing Transformation (VST).
  - conditional variance no longer depends on unknown distribution parameter
  - heteroskedastic data becomes homoskedastic

Variance stabilization

- Nonlinear 1-D mapping to make the noise variance invariant with respect to the noise-free signal: Variance-Stabilizing Transformation (VST).
  - conditional variance no longer depends on unknown distribution parameter
  - heteroskedastic data becomes homoskedastic
- Anscombe VST for Poisson data (1948):

$$a(z) = 2\sqrt{z + {}^{3}\!/_{8}}.$$

Variance stabilization

- Nonlinear 1-D mapping to make the noise variance invariant with respect to the noise-free signal: Variance-Stabilizing Transformation (VST).
  - conditional variance no longer depends on unknown distribution parameter
  - heteroskedastic data becomes homoskedastic
- Anscombe VST for Poisson data (1948):

$$a(z) = 2\sqrt{z + {}^3\!/_8}.$$

- Noise becomes asymptotically standard normal $\mathcal{N}(0, 1)$.

Variance stabilization

- Nonlinear 1-D mapping to make the noise variance invariant with respect to the noise-free signal: Variance-Stabilizing Transformation (VST).
  - conditional variance no longer depends on unknown distribution parameter
  - heteroskedastic data becomes homoskedastic
- Anscombe VST for Poisson data (1948):

$$a(z) = 2\sqrt{z + {}^{3}\!/_{8}}.$$

- Noise becomes asymptotically standard normal $\mathcal{N}(0, 1)$.
- Constant noise variance $\rightarrow$ additive noise filters.

Variance stabilization

- Nonlinear 1-D mapping to make the noise variance invariant with respect to the noise-free signal: Variance-Stabilizing Transformation (VST).
  - conditional variance no longer depends on unknown distribution parameter
  - heteroskedastic data becomes homoskedastic
- Anscombe VST for Poisson data (1948):

$$a(z) = 2\sqrt{z + 3/8}.$$

- Noise becomes asymptotically standard normal $\mathcal{N}(0, 1)$.
- Constant noise variance $\rightarrow$ additive noise filters.
- Fast and simple.

Poisson Denoising Evolution from a VST perspective

Poisson Denoising via Anscombe VST (1948)

1: Apply VST $\rightarrow$ Anscombe
2: Denoising with AWGN filter
3: Asymptotically unbiased Inverse VST

Poisson Denoising Evolution from a VST perspective

Poisson Denoising via Anscombe VST (1948)

1: Apply VST $\rightarrow$ Anscombe
2: Denoising with AWGN filter
3: Asymptotically unbiased Inverse VST

- Vast literature and applications based on it: the workhorse of Poisson data processing

Poisson Denoising Evolution from a VST perspective

Poisson Denoising via Anscombe VST (1948)

1: Apply VST $\rightarrow$ Anscombe
2: Denoising with AWGN filter
3: Asymptotically unbiased Inverse VST

- Vast literature and applications based on it: the workhorse of Poisson data processing

Problem: $\mathrm{E}\{a(z) \mid y\} = a(y) + \mathrm{bias}(y)$

Poisson Denoising Evolution from a VST perspective

Poisson Denoising via Anscombe VST (1948)

1: Apply VST $\rightarrow$ Anscombe
2: Denoising with AWGN filter
3: Asymptotically unbiased Inverse VST

- Vast literature and applications based on it: the workhorse of Poisson data processing

Problem: $\mathrm{E}\{a(z) \mid y\} = a(y) + \mathrm{bias}(y)$
Asymptotically unbiased inverse accurate only for $y \gtrsim 5$.

Poisson Denoising Evolution from a VST perspective

Poisson Denoising via Anscombe VST (1948)
1: Apply VST $\rightarrow$ Anscombe
2: Denoising with AWGN filter
3: Asymptotically unbiased Inverse VST

- Vast literature and applications based on it: the workhorse of Poisson data processing

Problem: $\mathrm{E}\{a(z) \mid y\} = a(y) + \mathrm{bias}(y)$

Asymptotically unbiased inverse accurate only for $y \gtrsim 5$.

$\Longrightarrow$ Need for ad-hoc filtering solutions for low-count Poisson data.

Poisson Denoising Evolution from a VST perspective

Mäkitalo & Foi (2009)

1: Apply VST
2: Denoising with AWGN filter
3: Exact Unbiased Inverse VST

- Introduces an exact inverse for the whole input range

$$\mathrm{E}\{a(z) \mid y\} \longmapsto \mathrm{E}\{z \mid y\} = y$$

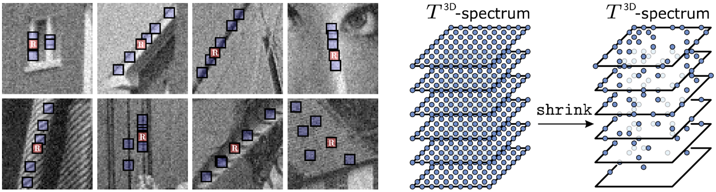
Poisson Denoising Evolution from a VST perspective

Mäkitalo & Foi (2009)

1: Apply VST → Anscombe
2: Denoising with AWGN filter → BM3D
3: Exact Unbiased Inverse VST

- Introduces an exact inverse for the whole input range
- Outperformed all earlier approaches.

BM3D :



Poisson Denoising Evolution from a VST perspective

Mäkitalo & Foi (2009)
1: Apply VST → Anscombe
2: Denoising with AWGN filter → BM3D
3: Exact Unbiased Inverse VST

- Introduces an exact inverse for the whole input range
- Outperformed all earlier approaches.

Problem: For low counts (*e.g.*, peak$\ll$1, or SNR$\ll$0dB), Poisson VST are invariably inaccurate.
$\Longrightarrow$ Further need for ad-hoc filtering solutions for Poisson data at extremely low counts.

Poisson Denoising Evolution from a VST perspective

e.g., Salmon *et al.* (2014), Giryes et al. (2014), and many others

1: *Binning*
2: Apply VST
3: Denoising with AWGN filter
4: Exact Unbiased Inverse VST
5: *Debinning*

- Binning: replace $h \times h$ blocks of pixels with their sum.
- Binned data stays Poisson $\Longrightarrow$ does not interfere with VST.

Poisson Denoising Evolution from a VST perspective

e.g., Salmon *et al.* (2014), Giryes et al. (2014), and many others

1: *Binning*
2: Apply VST
3: Denoising with AWGN filter
4: Exact Unbiased Inverse VST
5: *Debinning*

- Binning: replace $h \times h$ blocks of pixels with their sum.
- Binned data stays Poisson $\Longrightarrow$ does not interfere with VST.

Problem Binning corresponds to a non-adaptive smoothing:
$\Longrightarrow$ Binning+VST at extremely low counts inferior to SoA.

Proposed Algorithm

Iterative Poisson Image Denoising via VST — Azzari & Foi (2016)

for K times **do**
  Combination of z with previous estimate (initialize as z)
  Binning - decreasing bin size
  Apply VST
  Denoising with AWGN filter
  Exact Unbiased Inverse VST
  Debinning
end for
return the last estimate

Azzari & Foi, *IEEE Signal Processing Letters* (8) 2016

Proposed Algorithm

Iterative Poisson Image Denoising via VST Azzari & Foi (2016)

```
for K times do
    Combination of z with previous estimate (initialize as z)
    Binning - decreasing bin size
    Apply VST → Anscombe
    Denoising with AWGN filter → BM3D
    Exact Unbiased Inverse VST
    Debinning
end for
return the last estimate
```

Super-fast and state-of-the-art quality at low and even extremely low counts.

Increase of SNR

- We define the convex combination

$$\bar{z}_i = \lambda_i z + (1 - \lambda_i)\, \hat{y}_{i-1} \qquad 0 < \lambda_i \leq 1$$

where $\hat{y}_{i-1}$ is the estimate of y at the $(i-1)$-th iteration.

Increase of SNR

- We define the convex combination

$$\bar{z}_i = \lambda_i z + (1 - \lambda_i)\,\hat{y}_{i-1} \qquad 0 < \lambda_i \leq 1$$

where $\hat{y}_{i-1}$ is the estimate of y at the $(i-1)$-th iteration.

- The mean and variance of $\lambda_i^{-2}\bar{z}_i$ are

$$\mathrm{E}\{\lambda_i^{-2}\bar{z}_i|y\} = \mathrm{var}\{\lambda_i^{-2}\bar{z}_i|y\} = \lambda_i^{-2}y.$$

Increase of SNR

- We define the convex combination

$$\bar{z}_i = \lambda_i z + (1 - \lambda_i)\, \hat{y}_{i-1} \qquad 0 < \lambda_i \leq 1$$

  where $\hat{y}_{i-1}$ is the estimate of y at the $(i-1)$-th iteration.
- The mean and variance of $\lambda_i^{-2}\bar{z}_i$ are

$$\mathrm{E}\{\lambda_i^{-2}\bar{z}_i | y\} = \mathrm{var}\,\{\lambda_i^{-2}\bar{z}_i | y\} = \lambda_i^{-2} y.$$

- $\lambda_i^{-2}\bar{z}_i$ is not Poisson, but is nonetheless stabilized by Anscombe, due to a classical result (Bar-Lev&Enis, 1990).

Increase of SNR

- We define the convex combination

$$\bar{z}_i = \lambda_i z + (1 - \lambda_i)\,\hat{y}_{i-1} \qquad 0 < \lambda_i \leq 1$$

  where $\hat{y}_{i-1}$ is the estimate of y at the $(i-1)$-th iteration.
- The mean and variance of $\lambda_i^{-2}\bar{z}_i$ are

$$\mathrm{E}\{\lambda_i^{-2}\bar{z}_i|y\} = \mathrm{var}\,\{\lambda_i^{-2}\bar{z}_i|y\} = \lambda_i^{-2}y.$$

- $\lambda_i^{-2}\bar{z}_i$ is not Poisson, but is nonetheless stabilized by Anscombe, due to a classical result (Bar-Lev&Enis, 1990).
- We develop Exact Unbiased Inverse $\mathrm{E}\{a(\lambda_i^{-2}\bar{z}_i)|y\} \mapsto y$

Increase of SNR

- We define the convex combination

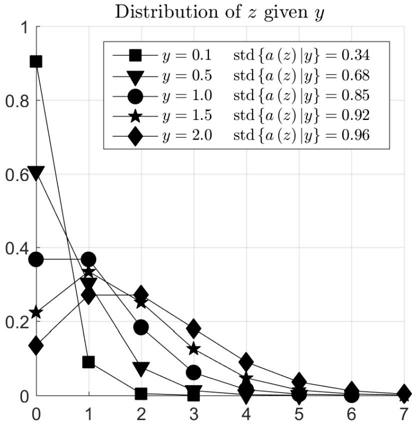
$$\bar{z}_i = \lambda_i z + (1-\lambda_i)\,\hat{y}_{i-1} \qquad 0 < \lambda_i \leq 1$$

  where $\hat{y}_{i-1}$ is the estimate of y at the $(i-1)$-th iteration.
- The mean and variance of $\lambda_i^{-2}\bar{z}_i$ are

$$\mathrm{E}\{\lambda_i^{-2}\bar{z}_i|y\} = \mathrm{var}\{\lambda_i^{-2}\bar{z}_i|y\} = \lambda_i^{-2}y.$$

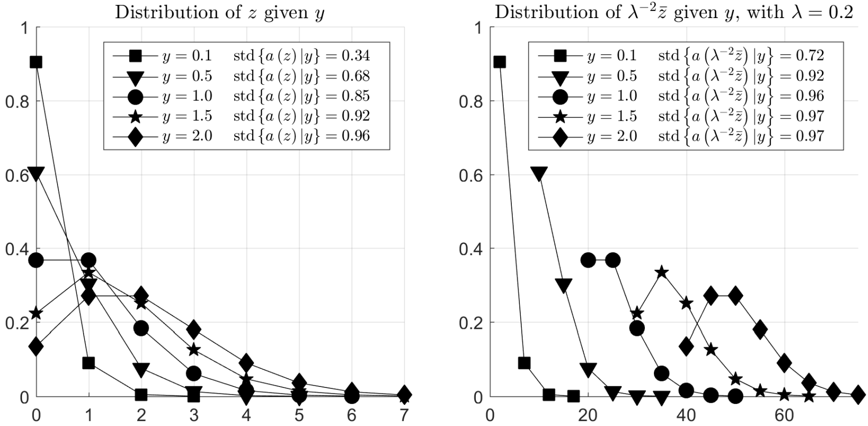
- $\lambda_i^{-2}\bar{z}_i$ is not Poisson, but is nonetheless stabilized by Anscombe, due to a classical result (Bar-Lev&Enis, 1990).
- We develop Exact Unbiased Inverse $\mathrm{E}\{a(\lambda_i^{-2}\bar{z}_i)|y\} \mapsto y$
- Can be interpreted as a form of boosting/twicing through VST.

Effect of convex combination on the data distributions



Poisson distributions have significant overlap, stabilization is poor

Effect of convex combination on the data distributions



After combination distributions are more disjoint

$\implies$ VST works better

Experiments and Results

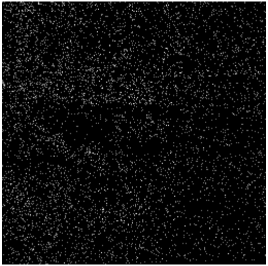
- Algorithm compared to state-of-the-art methods on a dataset of natural images.
- Superior overall performance in terms of PSNR and SSIM
- Proposed VST algorithm with BM3D is significantly less expensive than any of the other competitive methods.
    - At most 4 iterations.
- Very competitive results also when using other (including simpler) AWGN filters.

Some results

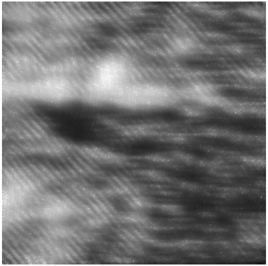
Method	Peak	Cam_{256^2}	Man_{512^2}	Bridge_{256^2}	Peppers_{256^2}	Time_{256^2}
NLSPCA	0.2	17.87	19.18	17.56	17.21	90s/12s
SPDA		17.80	19.73	17.81	17.25	5h/27min
P^4IP		**18.58**	–	17.54	17.44	few mins/∼30s
VST+BM3D		18.69	19.82	17.70	17.19	0.69s/0.12s
Proposed		18.40	**19.94**	**18.13**	**17.54**	0.83s
NLSPCA	1	20.25	21.46	19.02	19.50	86s/16s
SPDA		20.15	21.15	19.30	19.97	5h/25min
P^4IP		20.54	–	19.31	20.07	few mins
VST+BM3D		20.69	22.07	19.59	20.22	0.78s/0.10s
Proposed		**21.07**	**22.30**	**19.86**	**20.44**	0.82s

RED = Methods using fixed 3×3 binning

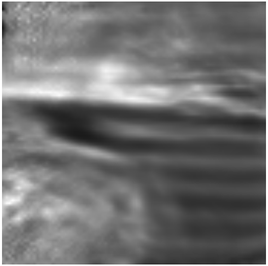
Image y



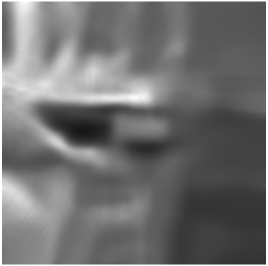
Noisy z (-3.43 0.01)



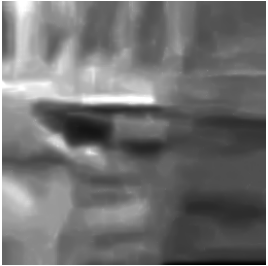
NLSPCA (17.55 0.24)



SPDA (17.68 0.25)



VST+BM3D (17.72 0.24)

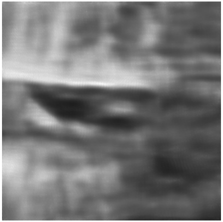


Proposed (**18.00 0.26**)

Denoising of *Bridge* at peak 0.2 (PSNR (dB) SSIM)

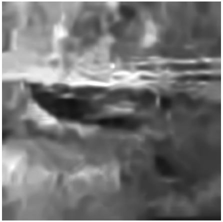
Image y

Noisy z (3.49dB)

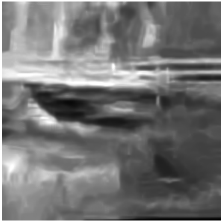


NLSPCA (19.18dB)

SPDA (19.36dB)

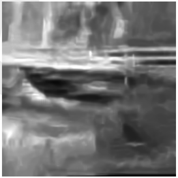


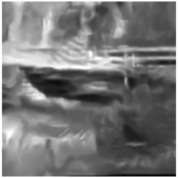
VST+BM3D (19.43dB)



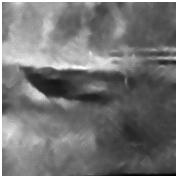
Proposed (**19.81**dB)

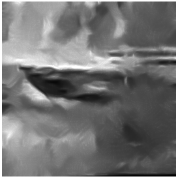
Denoising of *Bridge* at peak 1 (PSNR (dB) SSIM)

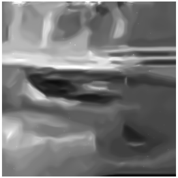

BM3D (19.81dB 0.362)


SAPCA (19.83dB 0.364)

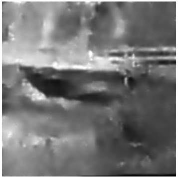

SADCT (19.81dB 0.351)

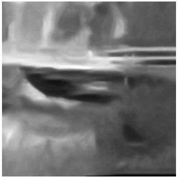

NLM (19.44dB 0.317)


FOVNLM (19.59dB 0.334)


SAFIR (19.60dB 0.338)


BLSGSM (19.57dB 0.347)


KSVD (19.72dB 0.341)


NLMPO (19.66dB 0.339)

Adopting different AWGN filters (PSNR (dB) SSIM)

Poisson Deblurring: Problem Formulation

Let us consider a Poisson image $z(x)$ as independent realizations of a Poisson random variable with mean and variance $g(x) \geq 0$, where $g = y \circledast v$:

$$z(x) \sim \mathcal{P}(g(x)), \quad \mathrm{P}(z(x) \,|g(x)) = \begin{cases} \frac{g(x)^{z(x)} e^{-g(x)}}{z(x)!} & z \in \mathbb{N} \cup \{0\} \\ 0 & \text{elsewhere.} \end{cases}$$

$$\mathrm{E}\{z|y\} = \mathrm{var}\{z|y\} = y \circledast v = g.$$

Goal: Estimate y from the observed z and PSF v (Poisson deblurring)

Extending the Iterative VST Poisson Denoising Approach to Deblurring

- Use of direct denoising for deconvolution is well explored for AWGN case: ForWaRD (Neelamani et al. 2004), BM3D (Dabov et al. 2008).

Extending the Iterative VST Poisson Denoising Approach to Deblurring

- Use of direct denoising for deconvolution is well explored for AWGN case: ForWaRD (Neelamani et al. 2004), BM3D (Dabov et al. 2008).
  - adopt linear regularized deconvolution

$$z^{\mathsf{RI}} = \mathcal{F}^{-1}(T^{\mathsf{RI}} Z) = t^{\mathsf{RI}} \circledast z\,, \qquad T^{\mathsf{RI}} = \frac{V^*}{|V|^2 + \mathcal{E}^2}\,,$$

where $\mathcal{F}$ Fourier transform, $Z = \mathcal{F}(z)$, $V = \mathcal{F}(v)$, V^* complex conjugate, $\mathcal{E}^2 \geq 0$ regularization term.

Extending the Iterative VST Poisson Denoising Approach to Deblurring

- Use of direct denoising for deconvolution is well explored for AWGN case: ForWaRD (Neelamani et al. 2004), BM3D (Dabov et al. 2008).
  - adopt linear regularized deconvolution

    $$z^{\mathsf{RI}} = \mathcal{F}^{-1}(T^{\mathsf{RI}} Z) = t^{\mathsf{RI}} \circledast z\,, \qquad T^{\mathsf{RI}} = \frac{V^*}{|V|^2 + \mathcal{E}^2}\,,$$

    where $\mathcal{F}$ Fourier transform, $Z = \mathcal{F}(z)$, $V = \mathcal{F}(v)$, V^* complex conjugate, $\mathcal{E}^2 \geq 0$ regularization term.
  - convolution with t^{RI} modifies independent noise into spatially correlated noise (colored noise)

Extending the Iterative VST Poisson Denoising Approach to Deblurring

- Use of direct denoising for deconvolution is well explored for AWGN case: ForWaRD (Neelamani et al. 2004), BM3D (Dabov et al. 2008).
  - adopt linear regularized deconvolution

$$z^{\mathsf{RI}} = \mathcal{F}^{-1}(T^{\mathsf{RI}} Z) = t^{\mathsf{RI}} \circledast z\,, \qquad T^{\mathsf{RI}} = \frac{V^*}{|V|^2 + \mathcal{E}^2}\,,$$

    where $\mathcal{F}$ Fourier transform, $Z = \mathcal{F}(z)$, $V = \mathcal{F}(v)$, V^* complex conjugate, $\mathcal{E}^2 \geq 0$ regularization term.
  - convolution with t^{RI} modifies independent noise into spatially correlated noise (colored noise)
  - denoise z^{RI} under colored Gaussian noise model

Extending the Iterative VST Poisson Denoising Approach to Deblurring

- Use of direct denoising for deconvolution is well explored for AWGN case: ForWaRD (Neelamani et al. 2004), BM3D (Dabov et al. 2008).
  - adopt linear regularized deconvolution

$$z^{\mathsf{RI}} = \mathcal{F}^{-1}(T^{\mathsf{RI}} Z) = t^{\mathsf{RI}} \circledast z\,, \qquad T^{\mathsf{RI}} = \frac{V^*}{|V|^2 + \mathcal{E}^2}\,,$$

    where $\mathcal{F}$ Fourier transform, $Z = \mathcal{F}(z)$, $V = \mathcal{F}(v)$, V^* complex conjugate, $\mathcal{E}^2 \geq 0$ regularization term.
  - convolution with t^{RI} modifies independent noise into spatially correlated noise (colored noise)
  - denoise z^{RI} under colored Gaussian noise model
- Extension of Poisson VST denoising requires:

Extending the Iterative VST Poisson Denoising Approach to Deblurring

- Use of direct denoising for deconvolution is well explored for AWGN case: ForWaRD (Neelamani et al. 2004), BM3D (Dabov et al. 2008).
  - adopt linear regularized deconvolution

$$z^{\mathsf{RI}} = \mathcal{F}^{-1}(T^{\mathsf{RI}} Z) = t^{\mathsf{RI}} \circledast z\,, \qquad T^{\mathsf{RI}} = \frac{V^*}{|V|^2 + \mathcal{E}^2}\,,$$

    where $\mathcal{F}$ Fourier transform, $Z = \mathcal{F}(z)$, $V = \mathcal{F}(v)$, V^* complex conjugate, $\mathcal{E}^2 \geq 0$ regularization term.
  - convolution with t^{RI} modifies independent noise into spatially correlated noise (colored noise)
  - denoise z^{RI} under colored Gaussian noise model
- Extension of Poisson VST denoising requires:
  - specific ℓ_2 normalization of linear regularized inverse filters

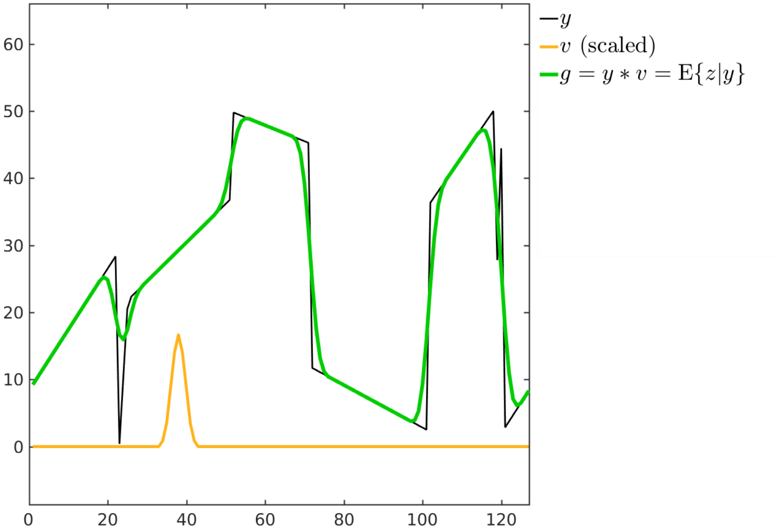
Extending the Iterative VST Poisson Denoising Approach to Deblurring

- Use of direct denoising for deconvolution is well explored for AWGN case: ForWaRD (Neelamani et al. 2004), BM3D (Dabov et al. 2008).
  - adopt linear regularized deconvolution

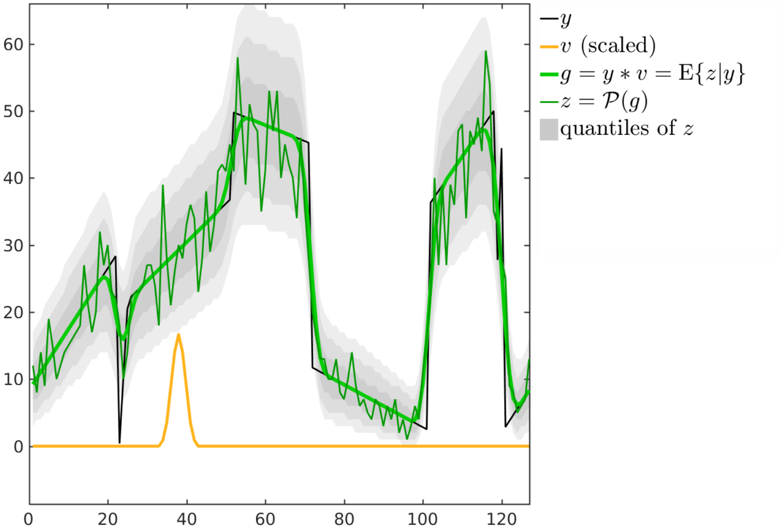
    $$z^{\mathsf{RI}} = \mathcal{F}^{-1}(T^{\mathsf{RI}} Z) = t^{\mathsf{RI}} \circledast z\,, \qquad T^{\mathsf{RI}} = \frac{V^*}{|V|^2 + \mathcal{E}^2}\,,$$

    where $\mathcal{F}$ Fourier transform, $Z = \mathcal{F}(z)$, $V = \mathcal{F}(v)$, V^* complex conjugate, $\mathcal{E}^2 \geq 0$ regularization term.
  - convolution with t^{RI} modifies independent noise into spatially correlated noise (colored noise)
  - denoise z^{RI} under colored Gaussian noise model
- Extension of Poisson VST denoising requires:
  - specific ℓ_2 normalization of linear regularized inverse filters
  - model noise power spectrum under VST

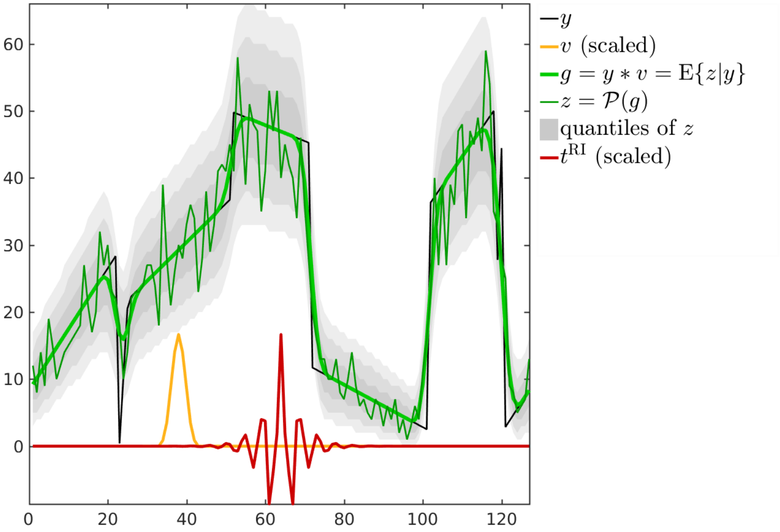
Poisson blurred observations



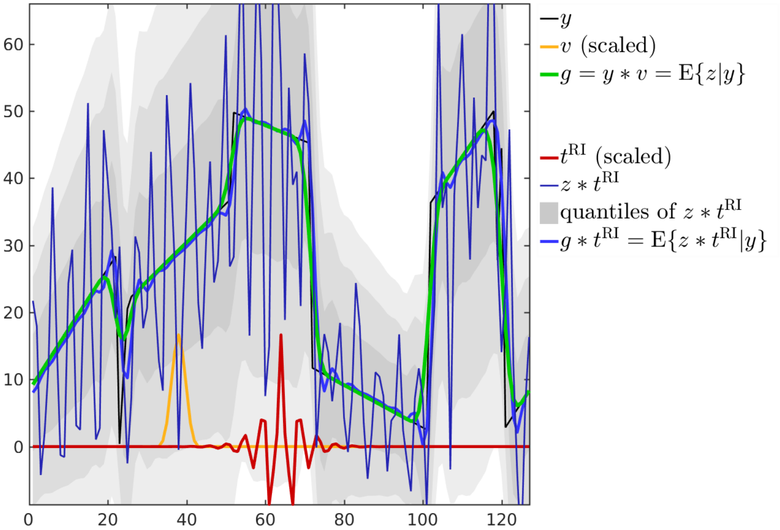
Poisson blurred observations



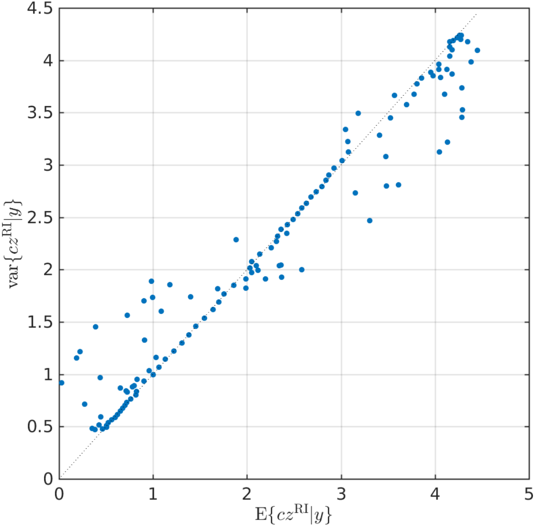
Regularized inverse kernel



Regularized deconvolution: noise amplification and correlation



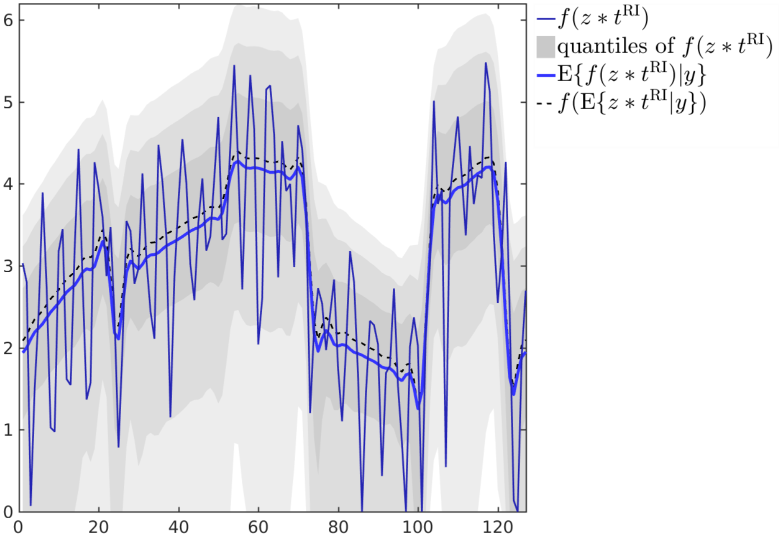
Approximately linear signal-dependent noise variance



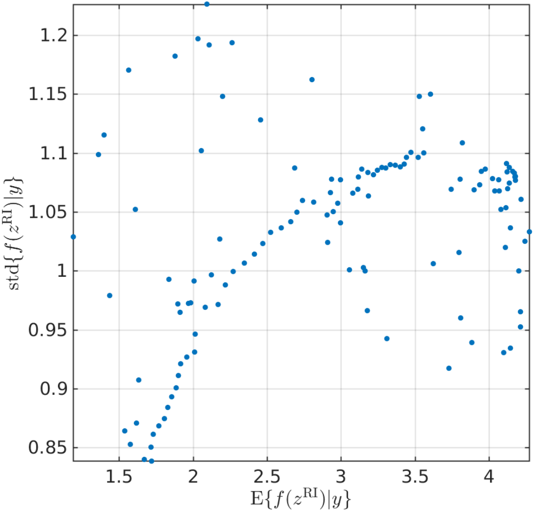
Approximately linear signal-dependent noise variance

- For even symmetric PSF, discrepancies between mean and variance are due only to even terms of order 2 or larger in the Taylor expansion of $y \circledast v$.
- Effective stabilization of variance, particularly where $y \circledast v$ is smooth and for symmetric PSFs.

Stabilization of variance and bias



Stabilized variance is approximately unitary



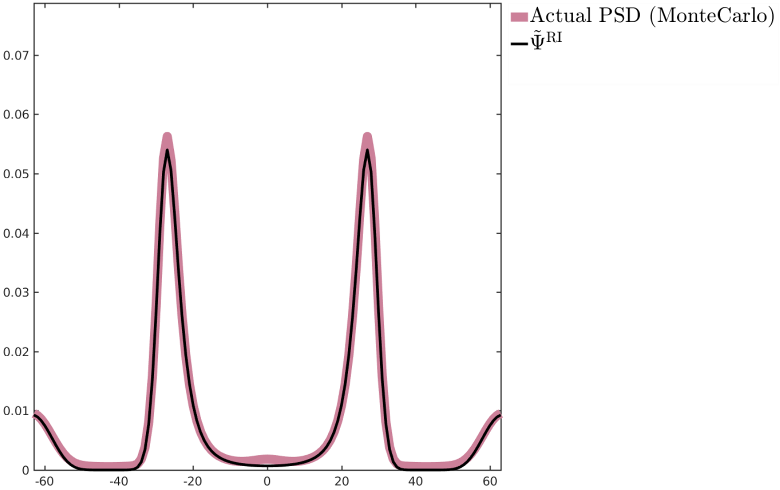
Colored Noise Power Spectrum

We model the noise power spectrum after stabilization as

$$\tilde{\Psi}^{\mathsf{RI}} = \Psi^{\mathsf{RI}} \| \Psi^{\mathsf{RI}} \|_1^{-1} |\Omega|^2 \,.$$

$\tilde{\Psi}^{\mathsf{RI}}$ corresponds to unitary spatial domain variance.

Colored Noise Power Spectrum



Colored Noise Power Spectrum and Denoising

We denoise the stabilized regularized inverse data with a filter Φ for colored noise:

$$D_i = \Phi\left[\bar{\bar{z}}_i^{\mathsf{RI}}, \tilde{\Psi}^{\mathsf{RI}}\right].$$

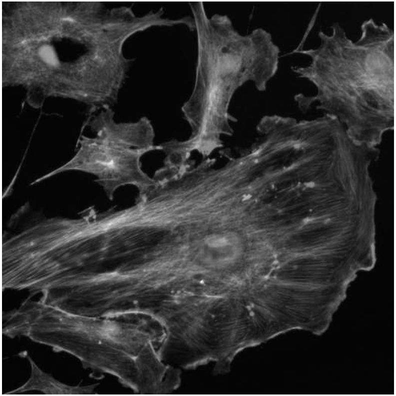
For *transform-domain filters* such as BM3D, $\tilde{\Psi}^{\mathsf{RI}}$ determines the internal shrinkage thresholds for each transform coefficient.

Poisson deblurring results

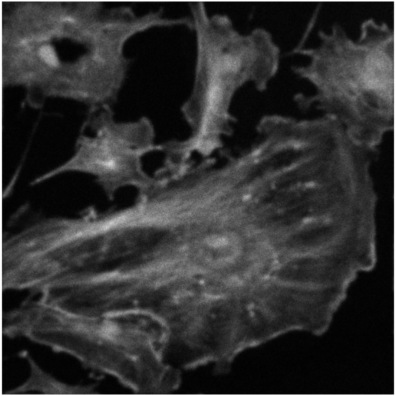
(PSNR, dB, average over 10 noise realizations)

Method	Peak	*Cameraman*	*Moon*	*Fluocells*
Proposed	255	**24.54**	**27.75**	**31.66**
PURE-LET		24.46	27.66	31.42
PoissonHessReg		23.04	27.00	30.59
SPIRAL-TAP-TI		24.06	25.61	30.46
PoissonDeconv		22.78	25.03	30.96
Proposed	25.5	**23.03**	**26.06**	**29.03**
PURE-LET		22.85	25.69	28.88
PoissonHessReg		21.38	25.15	27.60
SPIRAL-TAP-TI		22.22	24.93	28.05
PoissonDeconv		21.57	24.62	27.19
Proposed	2.55	**21.15**	**24.24**	**26.11**
PURE-LET		20.65	23.91	25.81
PoissonHessReg		18.70	23.27	24.06
SPIRAL-TAP-TI		20.30	21.68	25.17
PoissonDeconv		15.03	15.28	18.51

Fluocells at peak 255
Gaussian PSF with variance 3

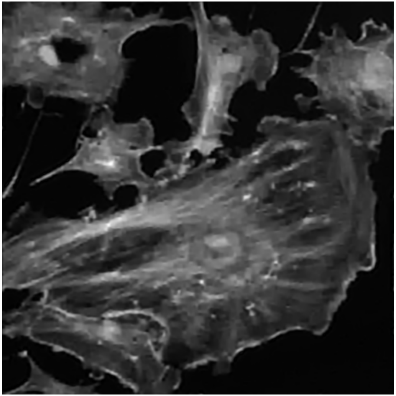


Original

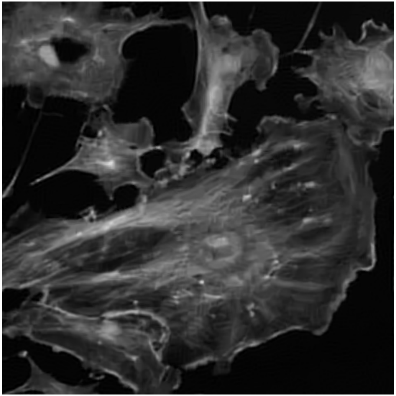


Observed (PSNR 28.01 dB)

Deblurring results



PURE-LET (31.42 dB)



Proposed (31.67 dB)

References

Anscombe, F.J., “The transformation of Poisson, binomial and negative-binomial data”, *Biometrika*, vol. 35, no. 3/4, pp. 246–254, Dec. 1948.

Arsenault, H. H., and M. Denis, “Integral expression for transforming signal-dependent noise into signal-independent noise”, *Opt. Lett.*, vol.6, no. 5, pp. 210-212, May 1981.

Azzari, L., and A. Foi, “Gaussian-Cauchy mixture modeling for robust signal-dependent noise estimation”, *Proc. 2014 IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP 2014)*, pp. 5357-5361, Florence, Italy, May 2014.

L. Azzari and A. Foi, "Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising", IEEE Signal Processing Letters, vol. 23, no. 8, pp. 1086-1090, August 2016.

L. Azzari and A. Foi, "Variance Stabilization in Poisson Image Deblurring", Proc. 2017 IEEE Int. Sym. Biomedical Imaging, pp. 728–731, Melbourne, Australia, April 18-21, 2017.

L. Azzari, L. R. Borges, and A. Foi, "Modeling and Estimation of Signal-Dependent and Correlated Noise", in Denoising of Photographic Images and Video: Fundamentals, Open Challenges and New Trends, M. Bertalmío (Ed.), Springer, 2018.

Bar-Lev, S. K., and P. Enis, “On the construction of classes of variance stabilizing transformations”, *Statistics & Probability Letters*, vol. 10, pp. 95–100, July 1990.

Bartlett, M. S., “The Square Root Transformation in Analysis of Variance,” *J. R. Statist. Soc. Suppl.*, vol.3, no. 1, pp. 68-78, 1936.

L. R. Borges, L. Azzari, P. R. Bakic, A. D. A. Maidment, M. A. C. Vieira, and A. Foi, "Restoration of low-dose digital breast tomosynthesis", Measurement Science and Technology, vol. 29, no. 6 (Special Feature on Advanced X-Ray Tomography), April 2018.

L. R. Borges, I. Guerrero, P. R. Bakic, A. Foi, A. D. A. Maidment, and M. A. C. Vieira, "Method for Simulating Dose Reduction in Digital Breast Tomosynthesis", IEEE Trans. Medical Imaging, vol. 36, no. 11, pp. 2331-2342, November 2017.

Cohen, A. C., *Truncated and Censored Samples*, CRC Press, 1991.

Conn, A. R., K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, MPS-SIAM Series on Optimization, vol. 8, 2009.

Coupé, P., P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images," *IEEE Transactions on Medical Imaging*, vol. 27, no. 4, pp. 425–441, April 2008.

Curtiss, J.H., "On transformations used in the analysis of variance", *The Annals of Mathematical Statistics*, vol. 14, no. 2, pp. 107–122, June 1943.

Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080-2095, August 2007.

Efron, B., "Transformation theory: How normal is a family of distributions?", *The Annals of Statistics*, vol. 10, no. 2, pp. 323–339, 1982.

Foi, A., "Clipped noisy images: heteroskedastic modeling and practical denoising", *Signal Processing*, vol. 89, no. 12, pp. 2609-2629, December 2009.

Foi, A., "Noise estimation and removal in MR imaging: the variance-stabilization approach," *Proc. 2011 IEEE Int. Symp. Biomedical Imaging (ISBI 2011)*, Chicago, IL, USA, 30 March - 2 April 2011.

Foi, A., S. Alenius, V. Katkovnik, and K. Egiazarian, "Noise measurement for raw-data of digital imaging sensors by automatic segmentation of non-uniform targets", *IEEE Sensors Journal*, vol. 7, no. 10, pp. 1456-1461, October 2007.

Foi, A., M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single image raw-data", *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737-1754, October 2008.

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

Foi, A., "Optimization of variance-stabilizing transformations", preprint, 2009.

Foi, A. "Removal of signal-dependent noise: the BM3D filter and optimized variance-stabilizing transformations," presented at *"Patch-based Image Representation, Manifolds and Sparsity" Minisymposium*, INRIA Centre de Rennes Bretagne Atlantique, IRISA, Rennes, France, April 2009. http://videos.rennes.inria.fr/seminaire_Irisa/Vista/

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

Fryzlewicz, P. and G. P. Nason, "A Haar-Fisz algorithm for Poisson Intensity Estimation," *J. Comp. Graph. Stat.*, vol.13, pp. 621-638, 2004.

Fryzlewicz, P., "Data-driven wavelet-Fisz methodology for nonparametric function estimation," *Electronic Journal of Statistics*, vol.2, pp. 863-896, 2008.

Guan, Y., "Variance stabilizing transformations of Poisson, binomial and negative binomial distributions," *Statistics and Probability Letters*, 2009.

Z. Harmany, R. Marcia, and R. Willett, "This is SPIRAL-TAP: Sparse Poisson intensity reconstruction algorithms?theory and prac- tice," IEEE Trans. Image Process., vol. 21, no. 3, pp. 1084?1096, 201

Hirakawa, K. and T. W. Parks, “Image denoising using total least squares,” *IEEE Trans. Image Process.*, vol.15, no. 9, pp. 2730-2742, September 2006.

Jin, X., Z. Xu, and K. Hirakawa, “Noise Parameter Estimation for Poisson Corrupted Images Using Variance Stabilization Transforms”, *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1329–1339, March 2014.

Kasturi, R. and J. F. Walkup and T. F. Krile, “Image restoration by transformation of signal-dependent noise to signal-independent noise,” *Applied Optics*, vol.22, no. 22, pp. 3537-3542, November 1983.

Kervrann, C., and J. Boulanger, “Local adaptivity to variable smoothness for exemplar-based image denoising and representation”, *International Journal of Computer Vision*, vol. 79, no. 1, pp. 45–69, Aug. 2008.

S. Lefkimmiatis and M. Unser, "Poisson image reconstruction with Hessian Schatten-norm regularization," IEEE Trans. Image Process., vol. 22, no. 11, pp. 4314?4327, 2013.

J. Li, F. Luisier, and T. Blu, "Deconvolution of Poissonian images with the PURE-LET approach," in 2016 IEEE Int. Conf. Image Process., Sept. 2016, pp. 2708?2712.

Luisier, F., T. Blu, and M. Unser, "Image Denoising in Mixed Poisson-Gaussian Noise", *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 696–708, March 2011.

Maggioni, M., V. Katkovnik, K. Egiazarian, and A. Foi, "A Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction", *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119-133, January 2013.

Manjón, J. V., P. Coupé, A. Buades, D. L. Collins, and M. Robles, "New methods for MRI denoising based on sparseness and self-similarity," *Medical Image Analysis*, vol. 16, no. 1, pp. 18–27, 2012.

Murtagh, F., J.-L. Starck and F. Murtagh, “Image restoration with noise suppression using a multiresolution support”, *Astronomy and Astrophysics*, vol. 112, no. 179, 1995.
Mäkitalo, M., and A. Foi, “Optimal inversion of the Anscombe transformation in low-count Poisson image denoising”, *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 99–109, January 2011.
Mäkitalo, M., and A. Foi, “A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation”, *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2697-2698, September 2011.
Mäkitalo, M., and A. Foi, “Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise”, *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 91–103, January 2013.
Mäkitalo, M., and A. Foi, “Noise parameter mismatch in variance stabilization, with an application to Poisson-Gaussian noise estimation”, *IEEE Trans. Image Process.*, 2014.

Mäkitalo, M., A. Foi, D. Fevralev, and V. Lukin, “Denoising of single-look SAR images based on variance stabilization and nonlocal filters”, *Proc. Int. Conf. Math. Meth. Electromagn. Th., MMET 2010*, Kiev, Ukraine, September 2010.

Nason, G. P., *Wavelet Methods in Statistics with R*, Springer, 2008.

R. Neelamani, H. Choi, and R. Baraniuk, "ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems," IEEE Trans. Signal Process., vol. 52, no. 2, pp. 418?433, 2004.

Nelder, J. A. and R. Mead, “A simplex method for function minimization,” *The Computer Journal*, vol.7, pp. 308-313, 1965.

Portilla, J., V. Strela, M.J. Wainwright, and E.P. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain”, *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

Prucnal, P. R. and B. E. A. Saleh, “Transformation of image-signal-dependent noise into image signal-independent noise,” *Optics Letters*, vol.6, no. 7, July 1981.

Pyatykh, S., and J. Hesser, “Image Sensor Noise Parameter Estimation by Variance Stabilization and Normality Assessment,” *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3990–3998, September 2014.

Starck, J. L., F. Murtagh, and A. Bijaoui, *Image Processing and Data Analysis*, Cambridge University Press, Cambridge, 1998.

Tippet, L. H. C., “Statistical methods in textile research. Part 2, Uses of the binomial and Poisson distributions,” *Shirley Inst. Mem.*, vol.13, pp. 35-72, 1934.

Tibshirani, R., “Estimating Transformations for Regression Via Additivity and Variance Stabilization,” *Journal of the American Statistical Association*, vol.83, no. 402, pp. 394-405, June 1988.

Uss M., Vozel B., Lukin, V., Chehdi K., "NoiseNet: Signal-Dependent Noise Variance Estimation with Convolutional Neural Network", International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS 2018, September 2018.

Zhang, B., and M. J. Fadili and J. L. Starck, “Wavelets, Ridgelets and Curvelets for Poisson Noise Removal,” *IEEE Trans. Image Process.*, vol.17, no. 7, pp. 1093-1108, July 2008.