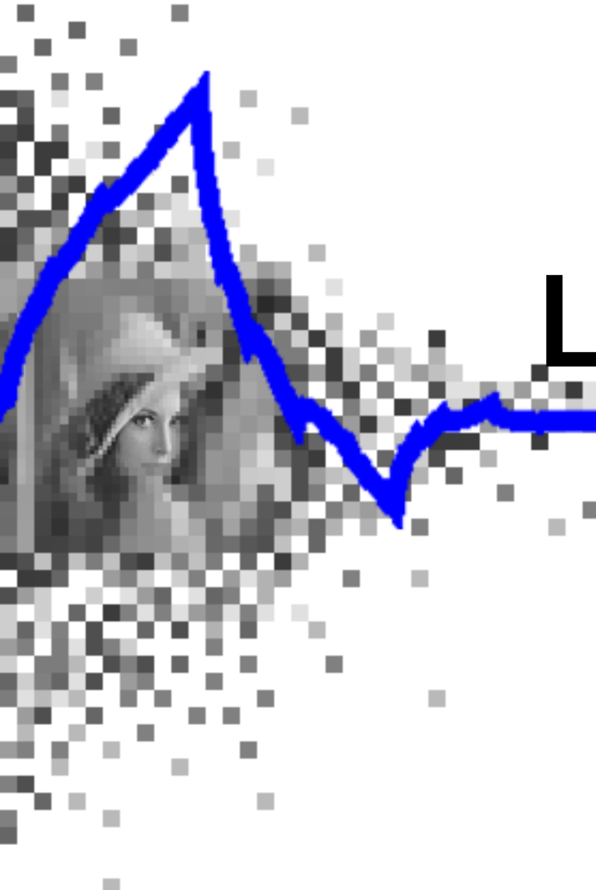


Linear and Non Linear Denoising

Gabriel Peyré

http://www.ceremade.dauphine.fr/~peyre/numerical-tour/

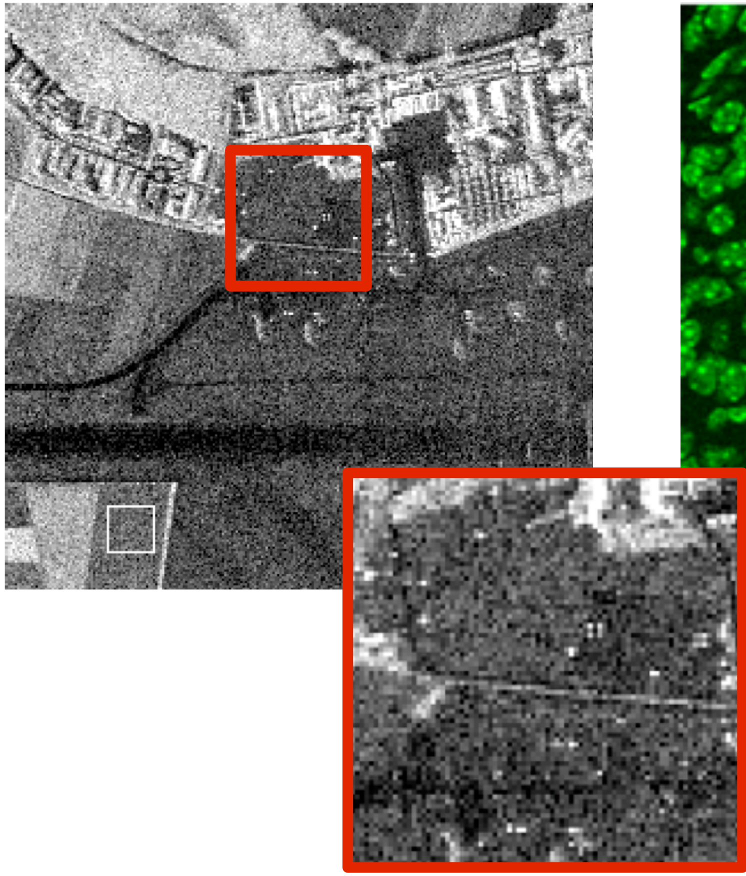
CNRS
CEREMADE
Dauphine Université Paris

Overview

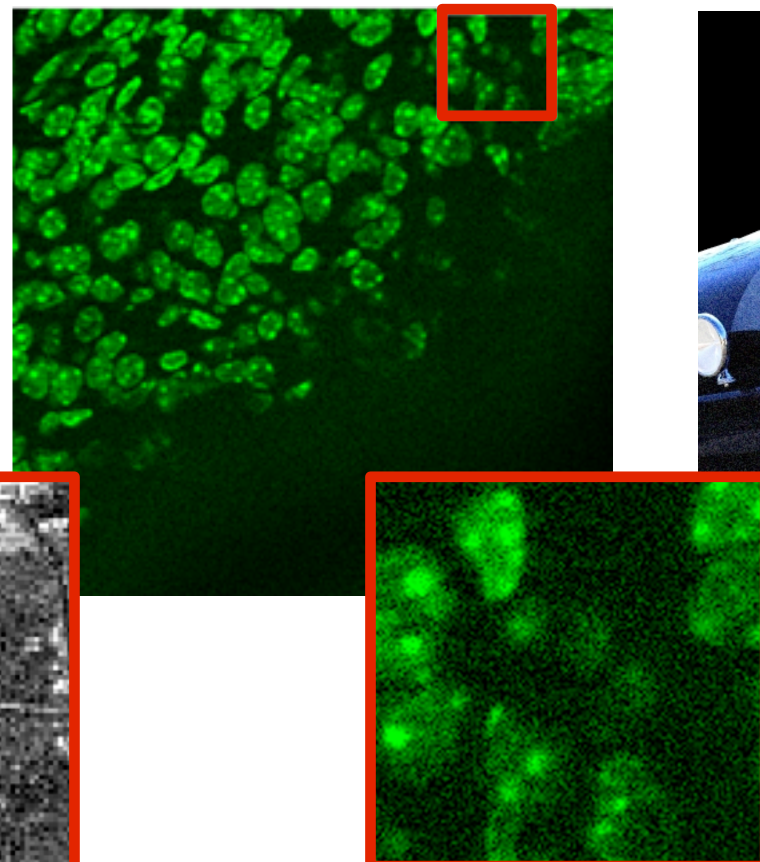
- Noise in Signals and Images
- Linear Denoising by Blurring
- Non-linear Wavelet Denoising
- Translation Invariant Thresholding
- Other Diagonal Thresholders
- Non-diagonal Block Thresholding
- Data-dependent Noise

Noise in Images

SAR imaging



Confocal microscopy



Digital camera

Data modeling (regularity, geometry, . . .)

Noise modeling (statistics, parameter estimation, . . .)

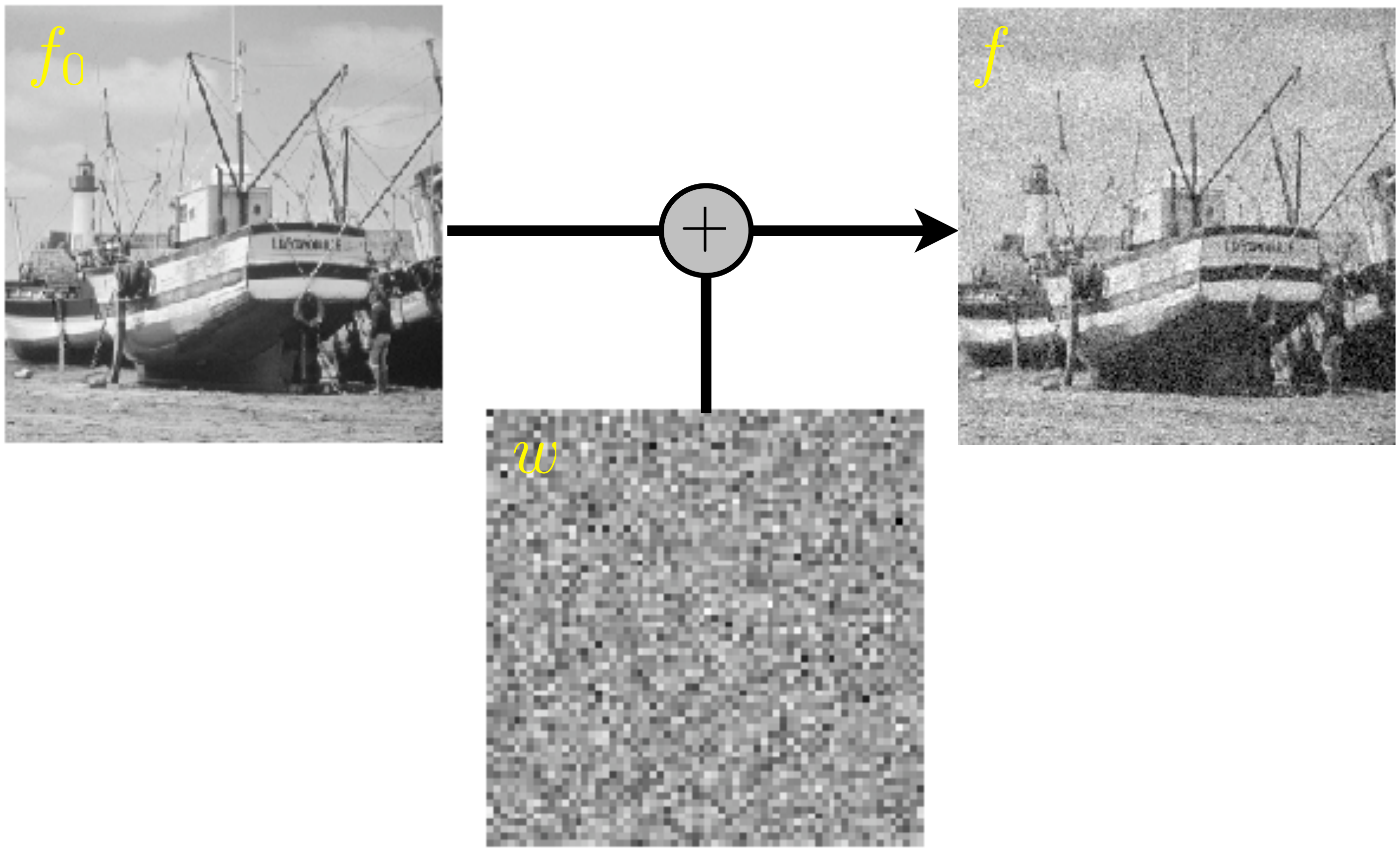
Image formation process (additive, multiplicative noise, . . .)

Denoising Problem

Idealization: $f = f_0 \oplus w$

$\oplus$: mixing operator (addition, multiplication, ...)

w: (realization of) a random vector of known distribution.



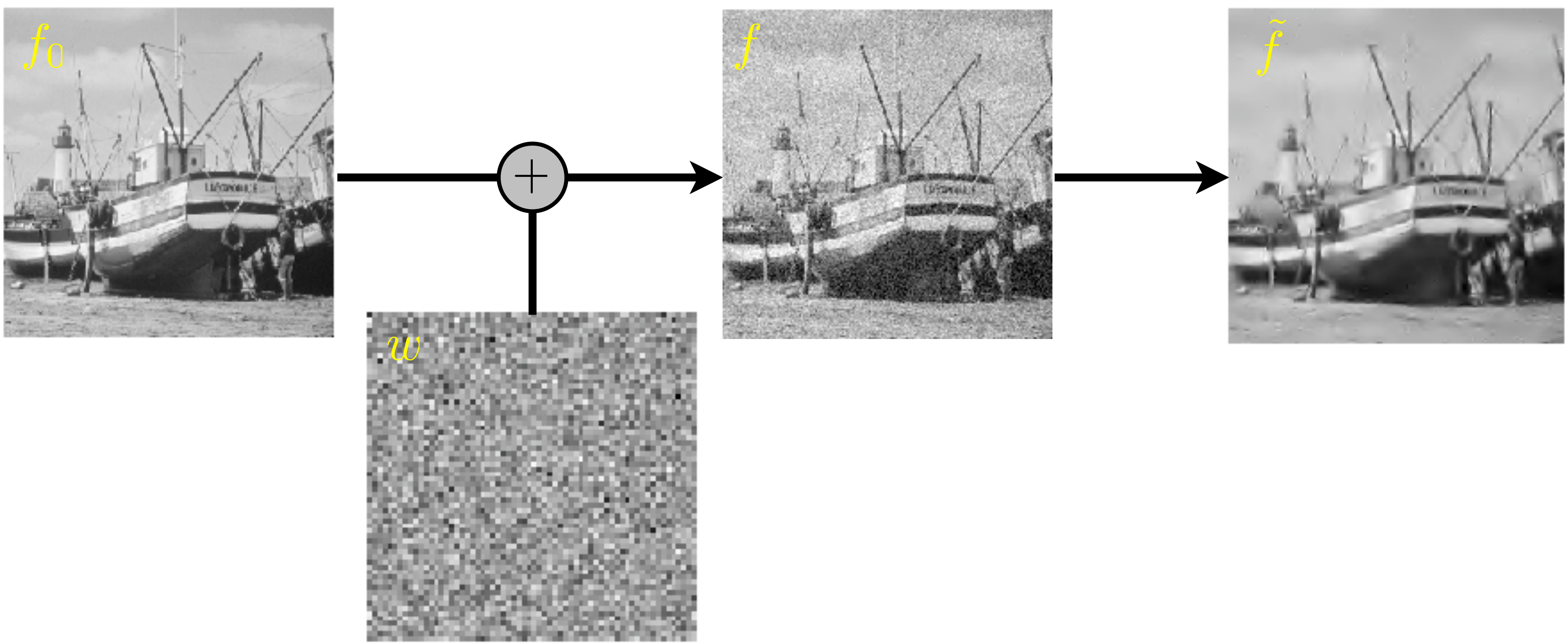
Denoising Problem

Idealization: $f = f_0 \oplus w$

$\oplus$: mixing operator (addition, multiplication, ...)

w: (realization of) a random vector of known distribution.

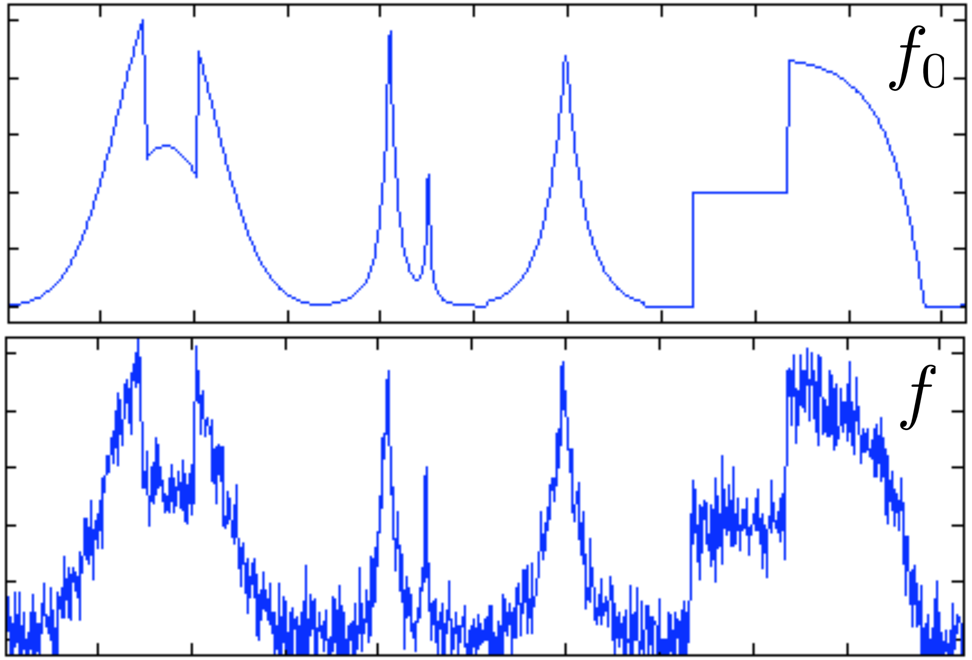
Denoising/estimation: $f \longmapsto \tilde{f} \in \mathbb{R}^N$ $\quad\quad$ $\tilde{f}$ depends only on f.



Additive Noise Model

Noisy signal $f = f_0 + w$, where $w[i] \sim W$ i.i.d, zero mean.

$\longrightarrow$ f is a realization of a random variable of mean f_0.



Additive Noise Model

Noisy signal $f = f_0 + w$, where $w[i] \sim W$ i.i.d, zero mean.

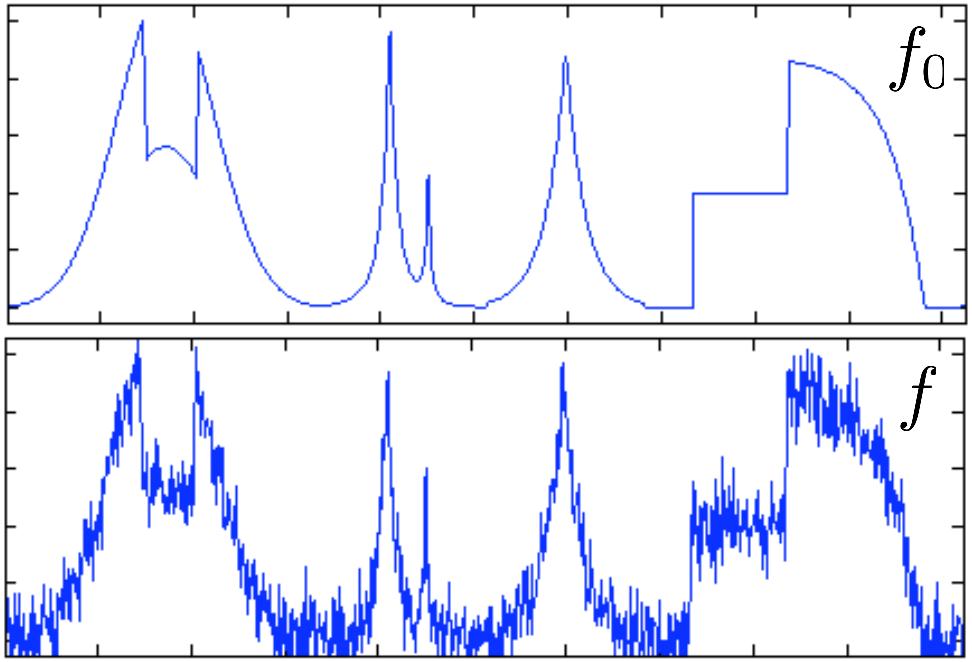
$\longrightarrow$ f is a realization of a random variable of mean f_0.

Denoising $\Longleftrightarrow$ estimating the mean from a single realization.

$$f \longmapsto \tilde{f} \in \mathbb{R}^N \qquad \tilde{f} \text{ depends only on } f.$$

$\longrightarrow$ exploit the redundancies/regularities in f_0.

Probabilistic framework: minimize the risk $E_W(\|\tilde{f} - f_0\|^2)$ (average wrt. $w \sim W$)

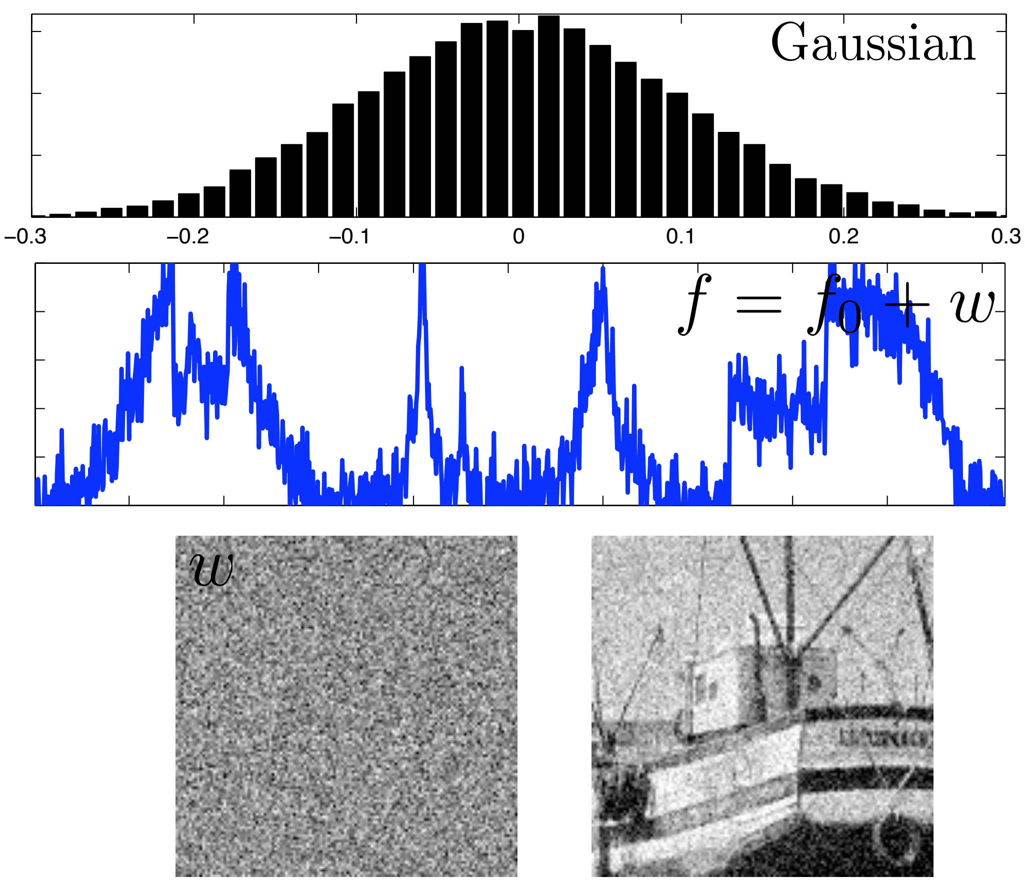


Noise Distributions

White noise: $w[i] \sim \mathcal{N}(0, \sigma)$.

Invariant under $U \in \mathcal{O}(n)$: $(Uw)[i] \sim \mathcal{N}(0, \sigma)$.

Uniform noise: $w[i] \in [-a, a]$.



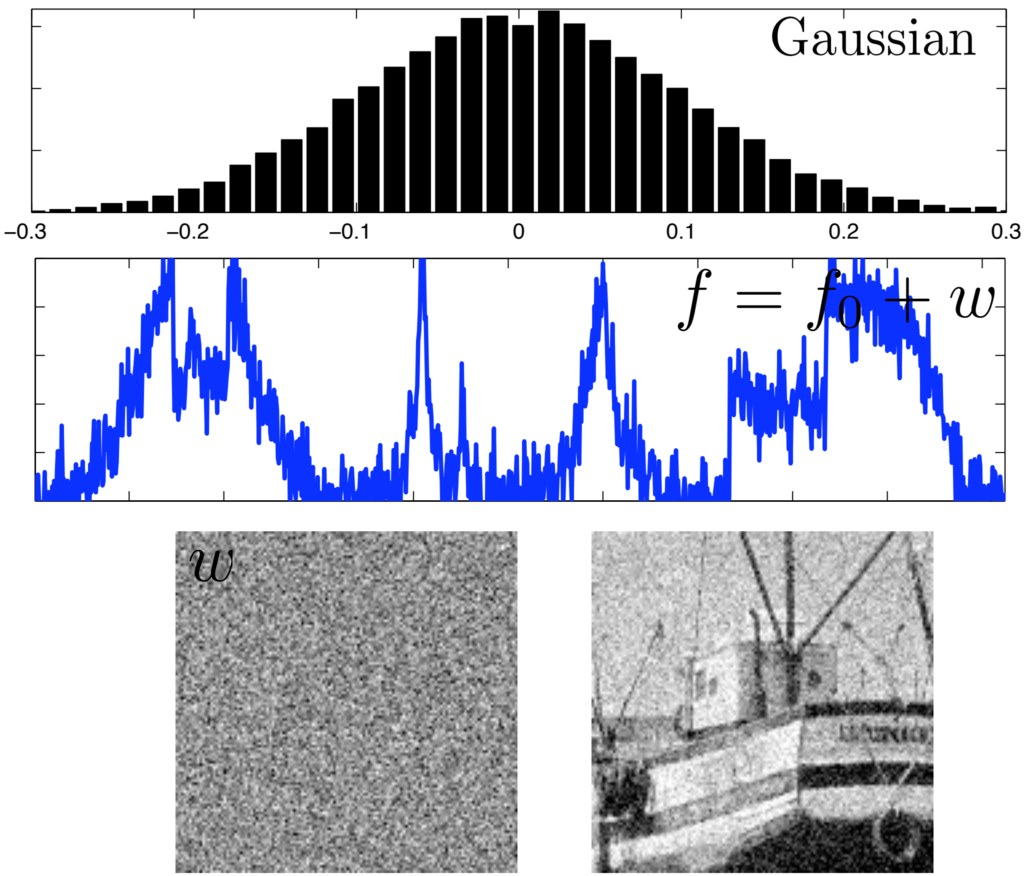
Noise Distributions

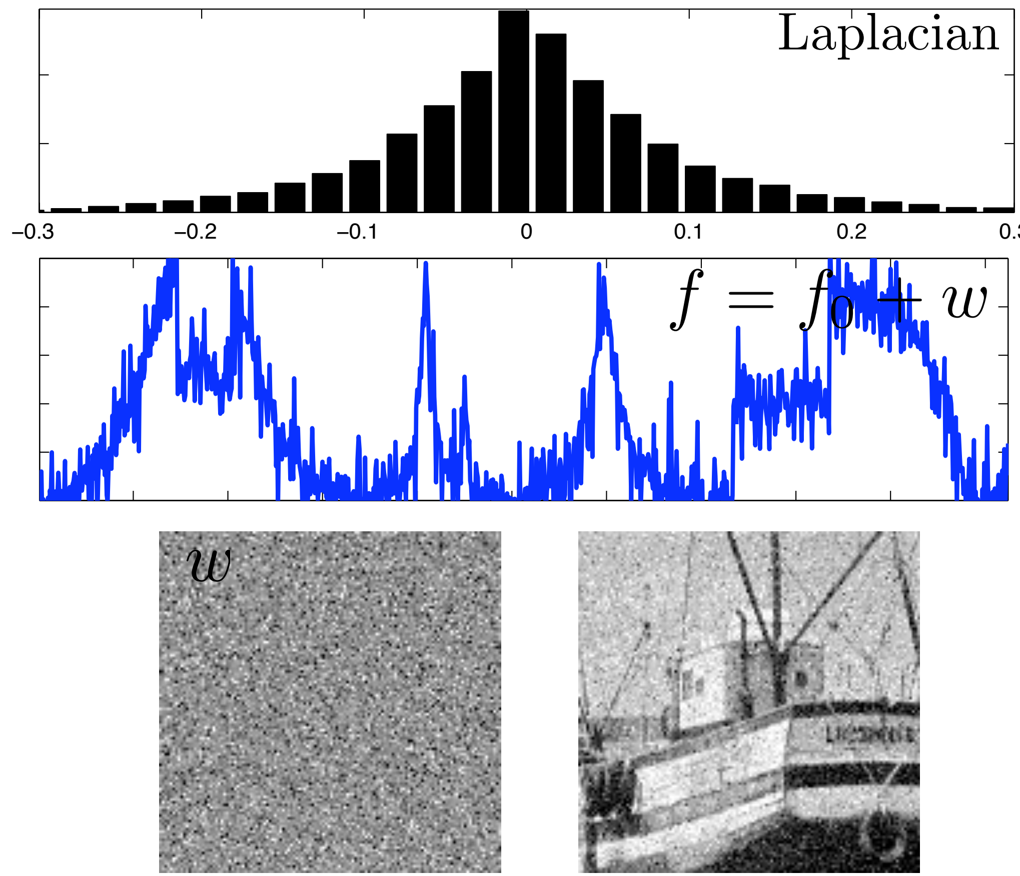
White noise: $w[i] \sim \mathcal{N}(0, \sigma)$.

Invariant under $U \in \mathcal{O}(n)$: $(Uw)[i] \sim \mathcal{N}(0, \sigma)$.

Uniform noise: $w[i] \in [-a, a]$.

Impulse noise: $p(w[i] = x) \propto e^{-|x/\sigma|^{\alpha}}$, $\alpha < 2$.





Noise Distributions

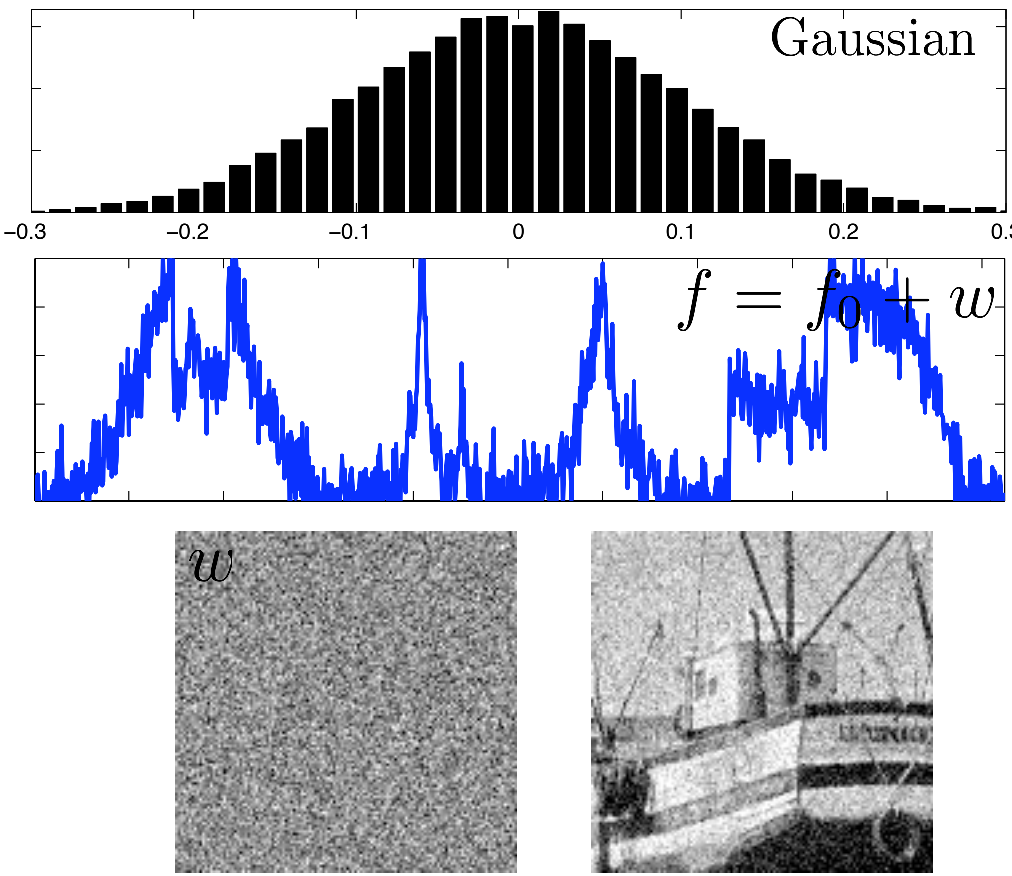
White noise: $w[i] \sim \mathcal{N}(0, \sigma)$.

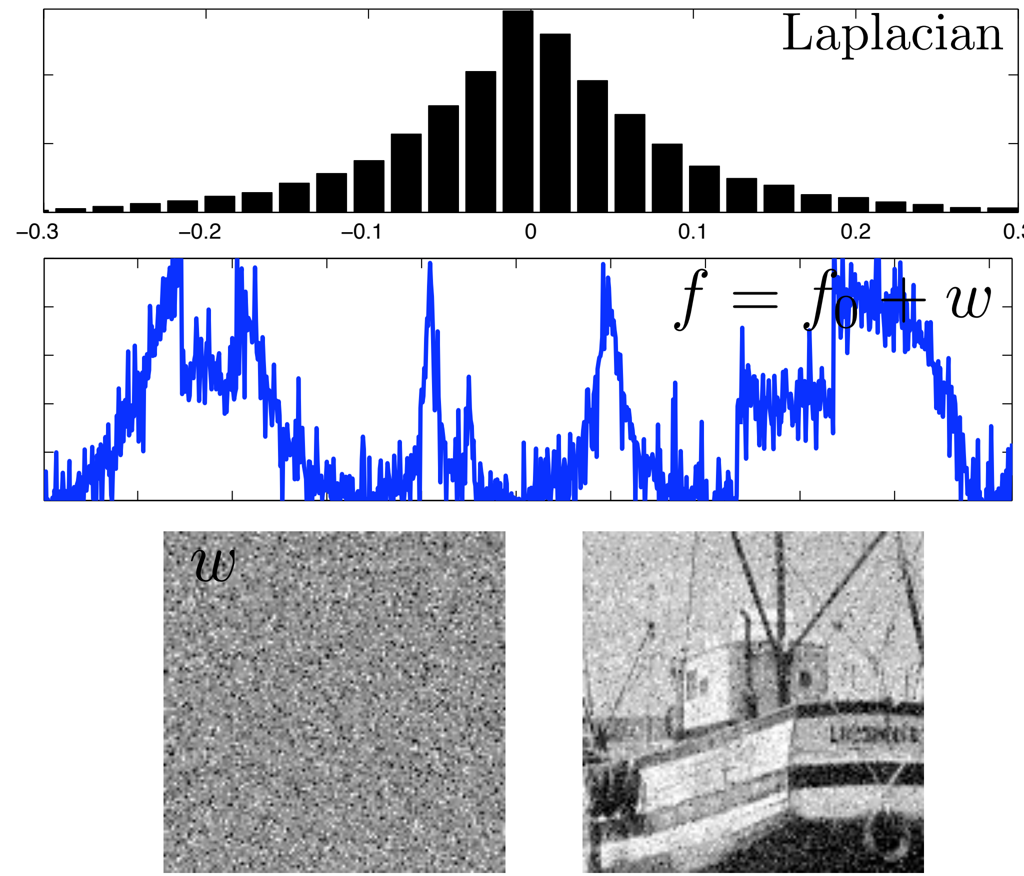
Invariant under $U \in \mathcal{O}(n)$: $(Uw)[i] \sim \mathcal{N}(0, \sigma)$.

Uniform noise: $w[i] \in [-a, a]$.

Impulse noise: $p(w[i] = x) \propto e^{-|x/\sigma|^\alpha}$, $\alpha < 2$.

Noise parameter estimation: from the image, from calibration.





Data-dependent Noise

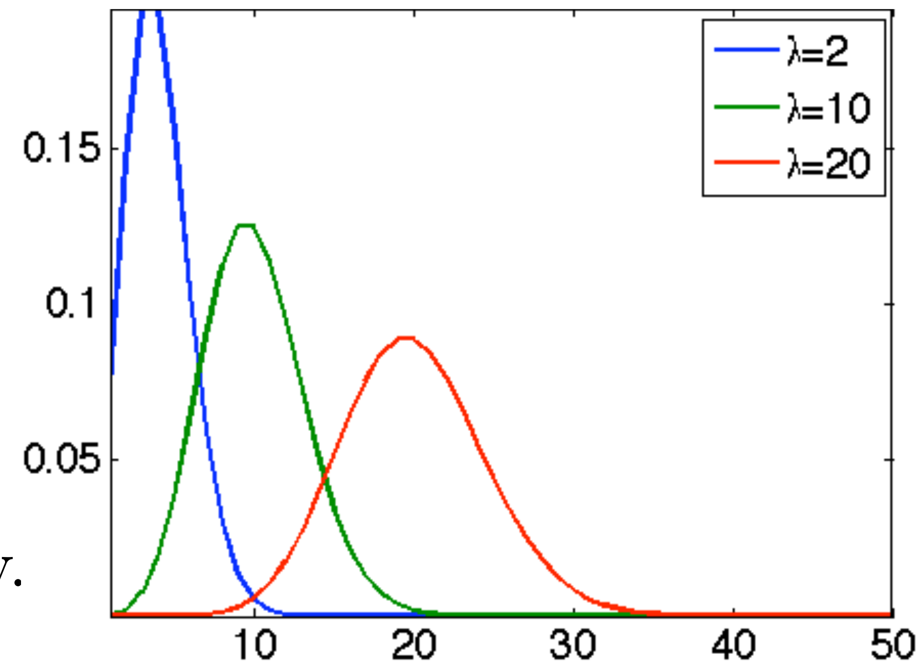
Poisson noise: $f[i] \sim \mathcal{P}(\lambda)$

where $\lambda = \lambda_i = f_0[i]$.

$$\mathbb{P}(f[i] = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

$$\mathrm{E}(f[i]) = \lambda \qquad \mathrm{Var}(f[i]) = \lambda$$

$\rightarrow$ counting photons, $f_0[i]$ =intensity.



$\lambda_{\max} = 5$

$\lambda_{\max} = 10$

$\lambda_{\max} = 50$

$\lambda_{\max} = 100$

Data-dependent Noise

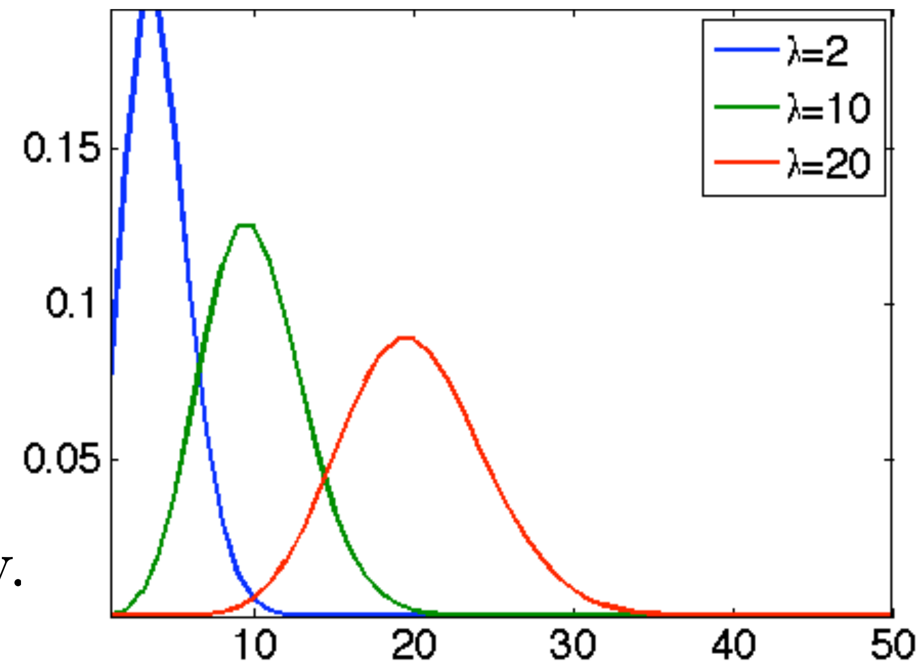
Poisson noise: $f[i] \sim \mathcal{P}(\lambda)$

where $\lambda = \lambda_i = f_0[i]$.

$$\mathbb{P}(f[i] = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

$$\mathrm{E}(f[i]) = \lambda \qquad \mathrm{Var}(f[i]) = \lambda$$

$\rightarrow$ counting photons, $f_0[i] =$intensity.




$\lambda_{\max} = 5$


$\lambda_{\max} = 10$


$\lambda_{\max} = 50$


$\lambda_{\max} = 100$

Variance stabilization: non-linear mapping $\varphi : \mathbb{R} \rightarrow \mathbb{R}$

such that $\varphi(f[i]) \simeq \mathcal{N}(\varphi(a[i]), \sigma)$, σ fixed.

Overview

- Noise in Signals and Images
- Linear Denoising by Blurring
- Non-linear Wavelet Denoising
- Translation Invariant Thresholding
- Other Diagonal Thresholders
- Non-diagonal Block Thresholding
- Data-dependent Noise

Linear Denoising Estimator

Noisy observations $f = f_0 + w$, $w[i] \sim \mathcal{N}(0, \sigma)$.

Linear estimator $\tilde{f}$ of f_0: depends *linearly* on f.

Linear Denoising Estimator

Noisy observations $f = f_0 + w$, $w[i] \sim \mathcal{N}(0, \sigma)$.

Linear estimator $\tilde{f}$ of f_0: depends *linearly* on f.

Translation invariant estimator:

$\tilde{f} = f \star h$ where $h \in \mathbb{R}^N$ is a (low pass) filter.

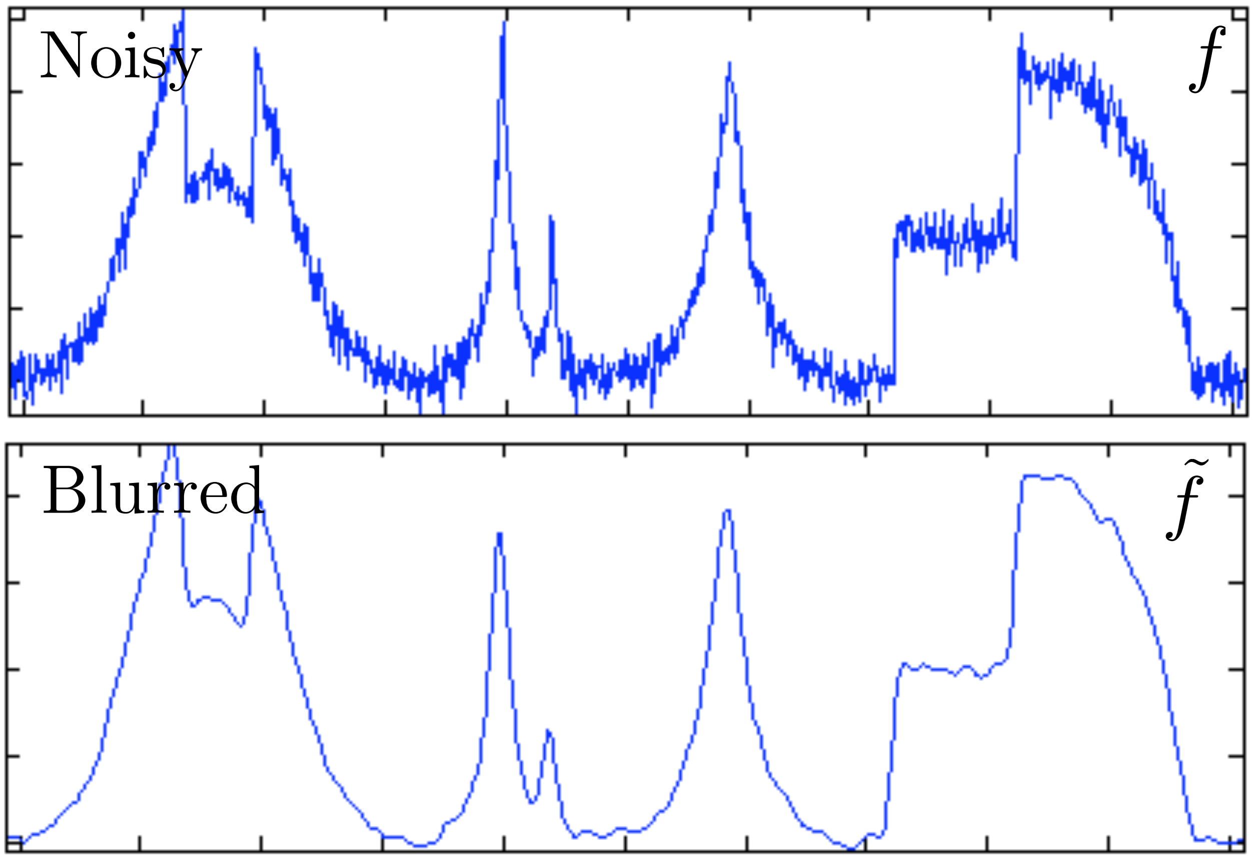
Linear Denoising Estimator

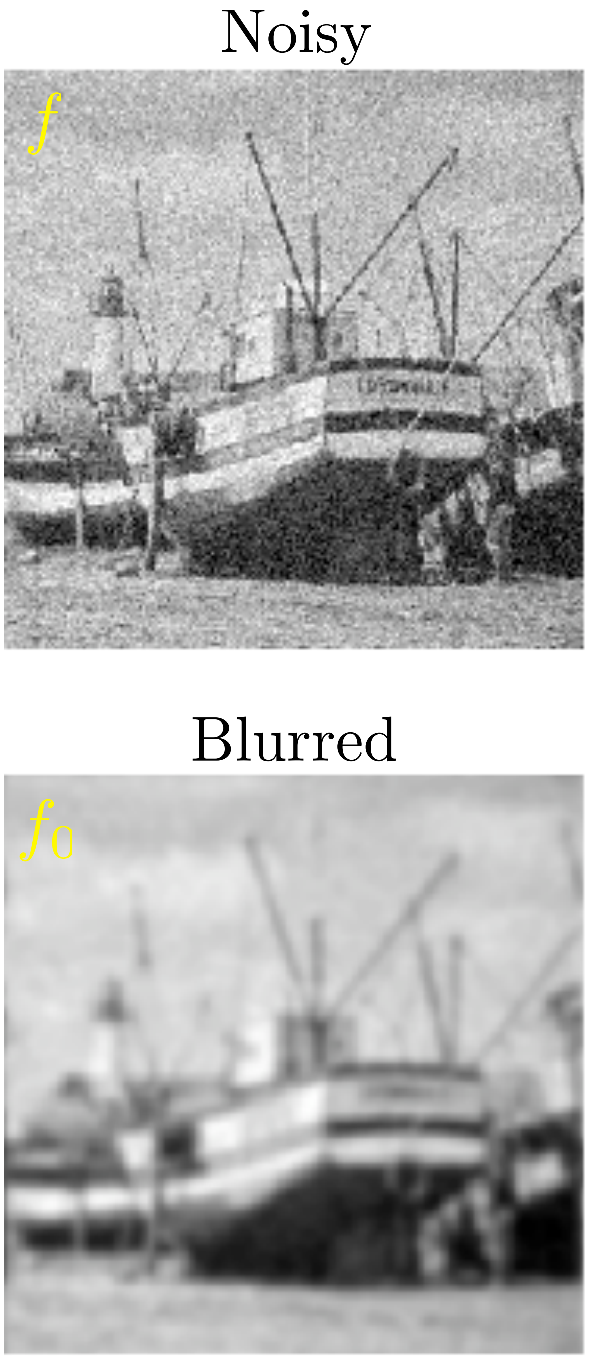
Noisy observations $f = f_0 + w$, $w[i] \sim \mathcal{N}(0, \sigma)$.

Linear estimator $\tilde{f}$ of f_0: depends *linearly* on f.

Translation invariant estimator:

$\tilde{f} = f \star h$ where $h \in \mathbb{R}^N$ is a (low pass) filter.

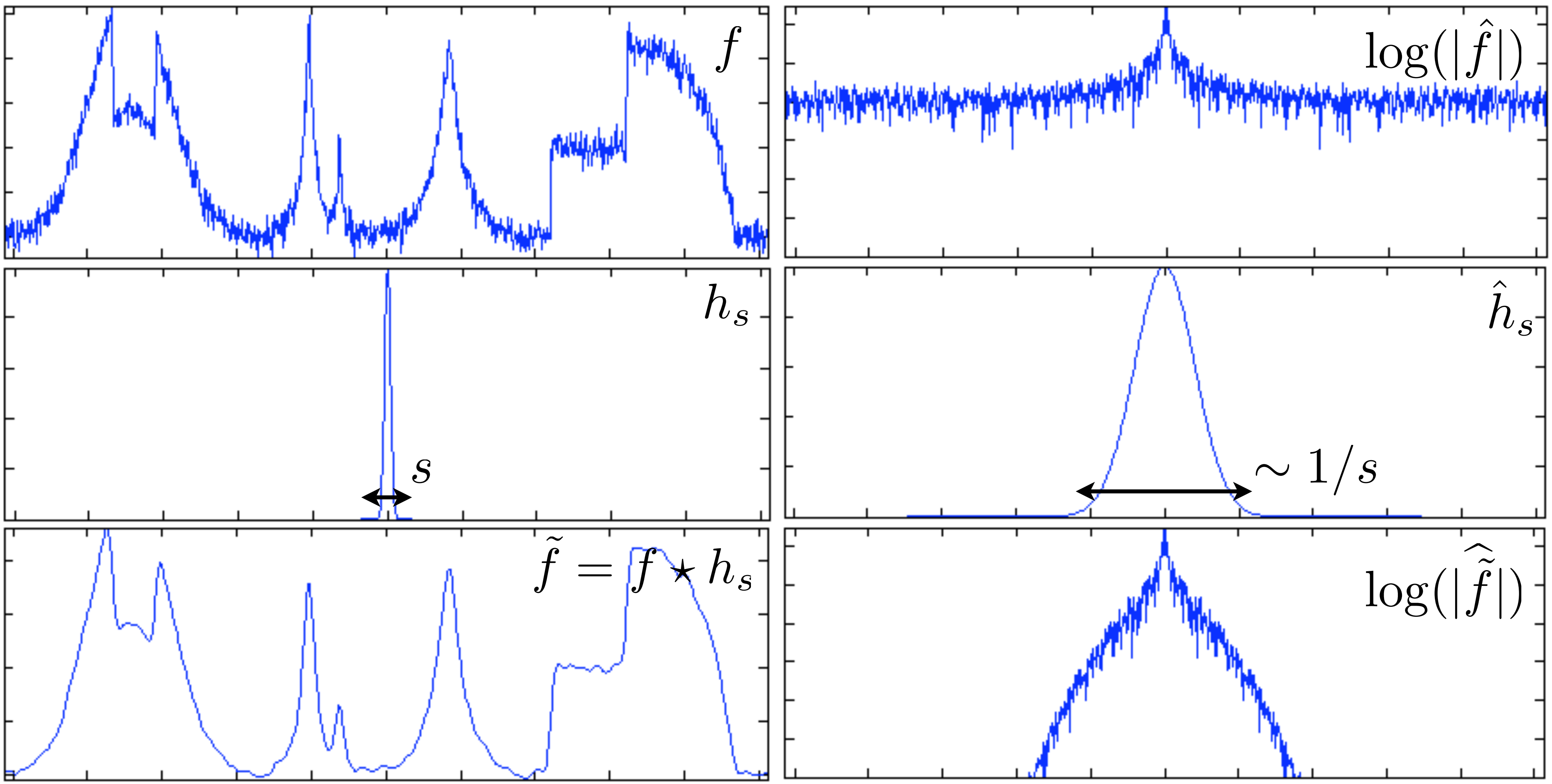




Fourier and Denoising

Gaussian filter: for $-N/2 < i \leqslant N/2$, $\quad h_s[i] = \dfrac{1}{Z_s} \exp\left(-\dfrac{i^2}{2s^2}\right)$

Z_s ensures that $\sum_i h_s[i] = 1$ (low pass). $\quad s =$ width of the filter.



Optimal Filter Choice

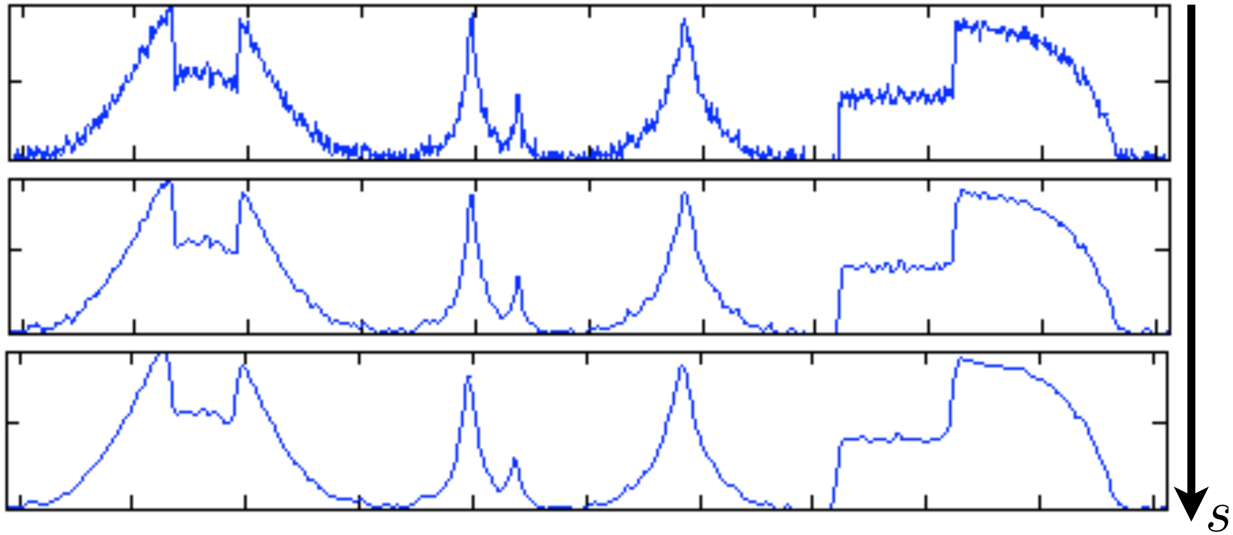
$$\|\tilde{f} - f_0\| \leqslant \|h_s \star f_0 - f_0\| + \|h_s \star w\|$$

Filter width s should be optimized to match:

– *Noise level* σ*:* $\|h_s \star w\|$ decreases with s.

– *Regularity of* f_0*:* (discontinuitites, edges):

$\|h_s \star f_0 - f_0\|$ increases with s.



Optimal Filter Choice

$$\|\tilde{f} - f_0\| \leqslant \|h_s \star f_0 - f_0\| + \|h_s \star w\|$$

Filter width s should be optimized to match:

– *Noise level* σ*:* $\|h_s \star w\|$ decreases with s.

– *Regularity of* f_0*:* (discontinuitites, edges):

$\|h_s \star f_0 - f_0\|$ increases with s.

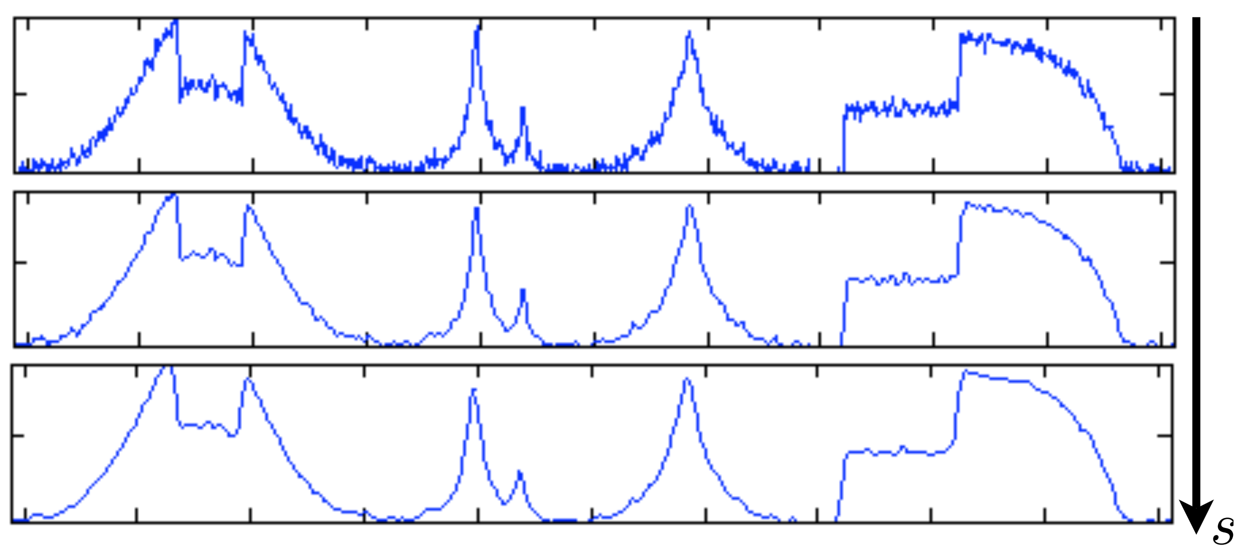
Wiener filter: random model for $f_0 \sim F$, indep. of $w \sim W$

F stationary random vector, known spectra $|\hat{F}[\omega]|^2$

Optimal h, minimizes

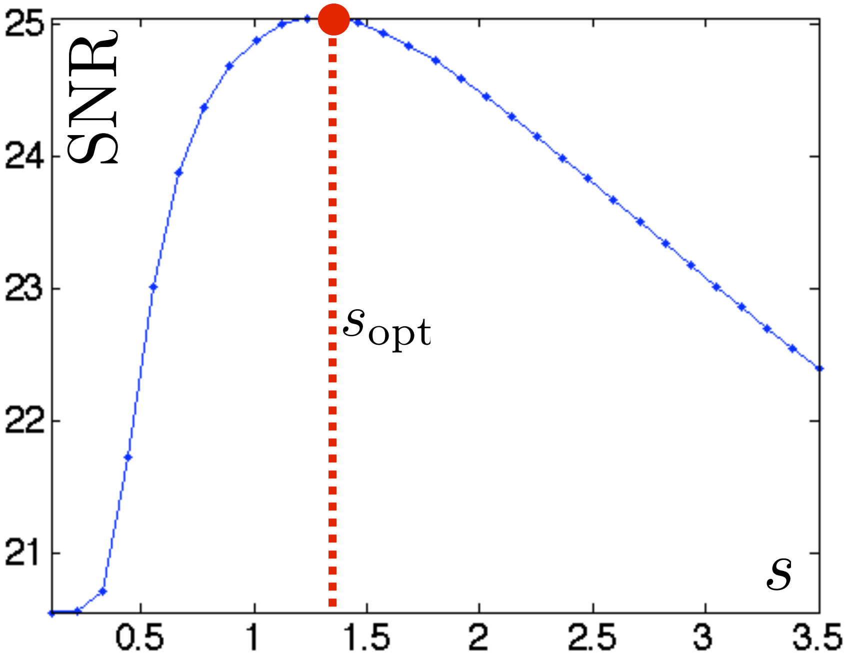
$$\mathrm{E}_{W,F}(\|h \star (F + W) - F\|^2)$$

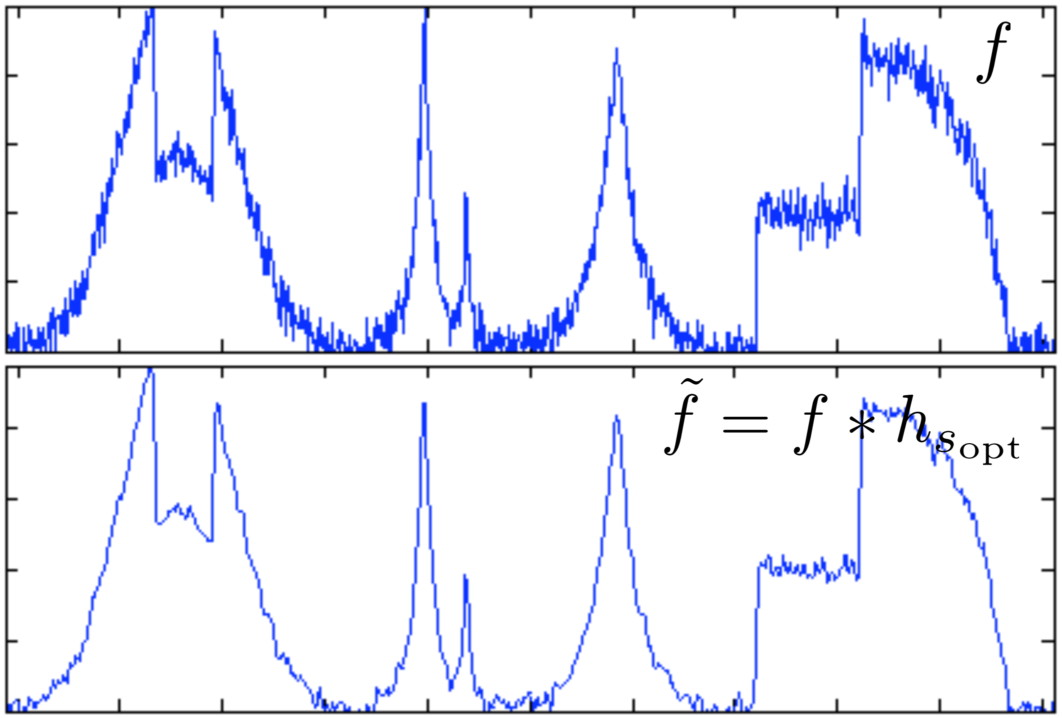
$$\Longrightarrow \quad \hat{h}[\omega] = \frac{|\hat{F}[\omega]|^2}{|\hat{F}[\omega]|^2 + \sigma^2}$$



Oracle Estimation of Optimal Filter

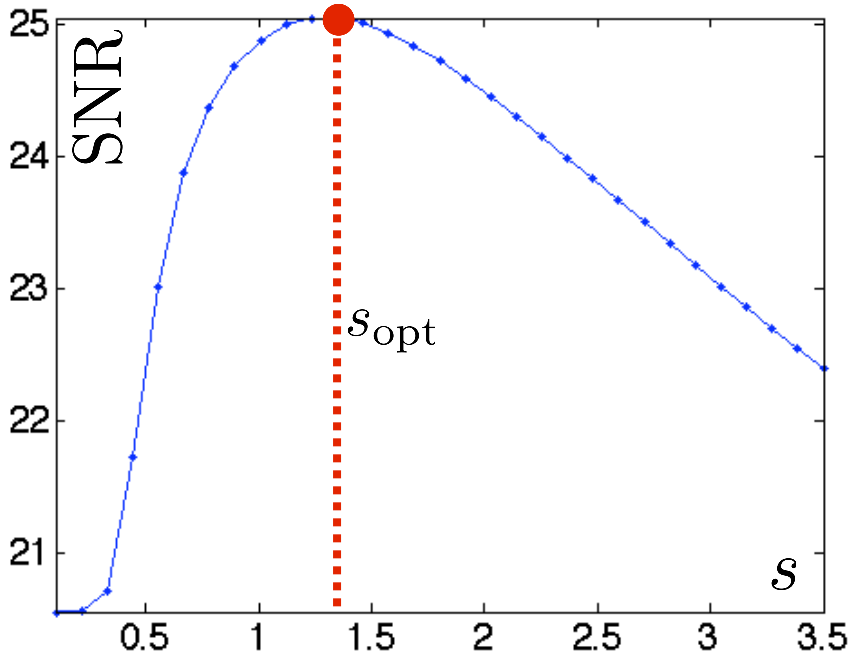
$$\mathrm{SNR}(f_0, \tilde{f}) = -10 \log_{10}\left(\frac{\|f_0 - \tilde{f}\|}{\|f_0\|}\right)$$

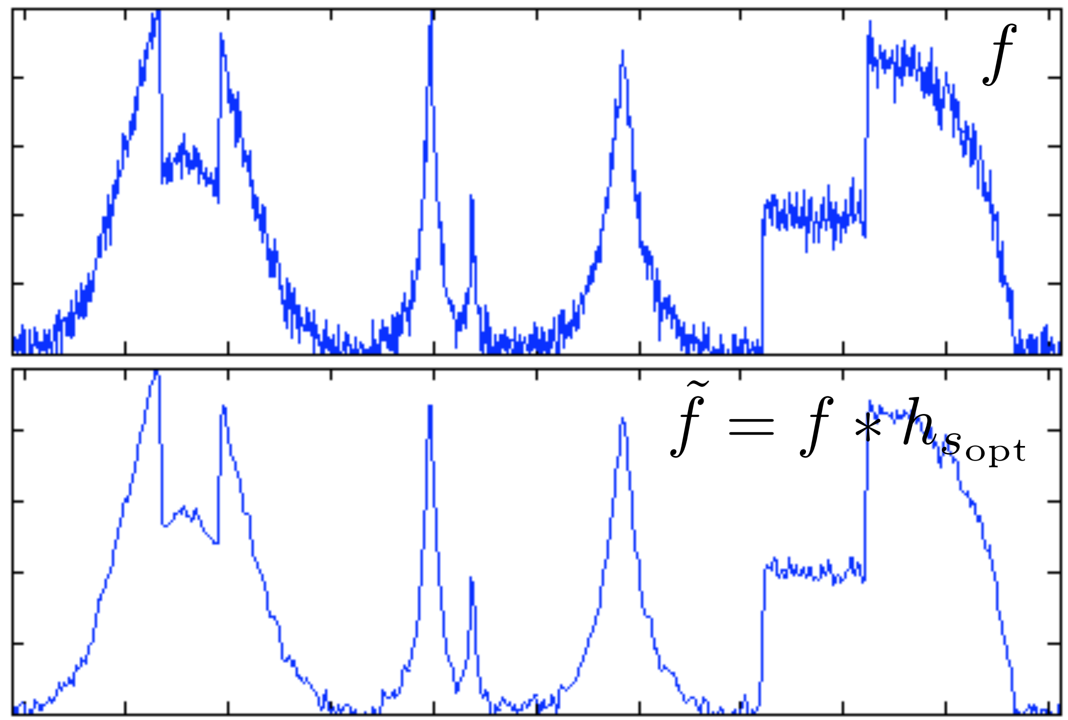


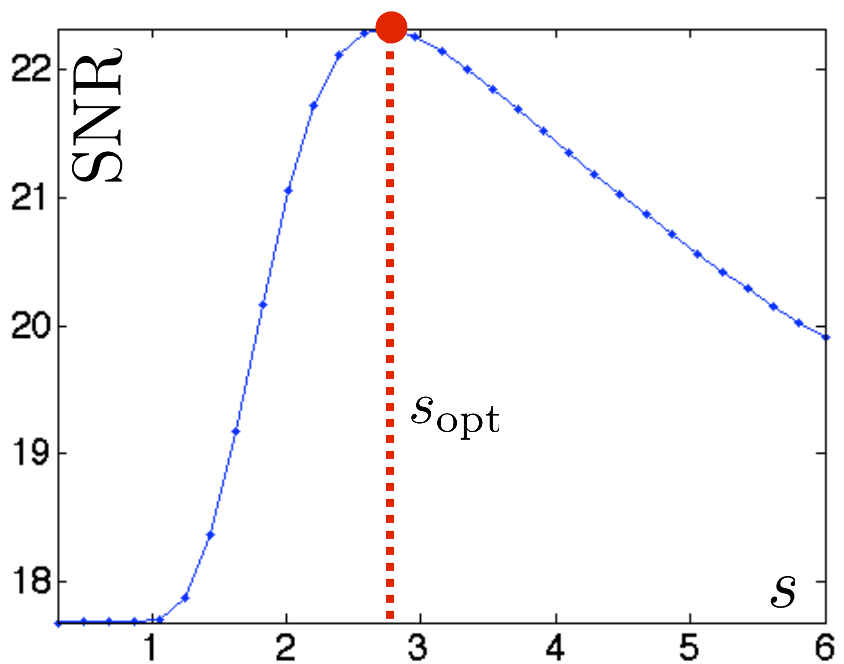


Oracle Estimation of Optimal Filter

$$\mathrm{SNR}(f_0, \tilde{f}) = -10 \log_{10}\left(\frac{\|f_0 - \tilde{f}\|}{\|f_0\|}\right)$$

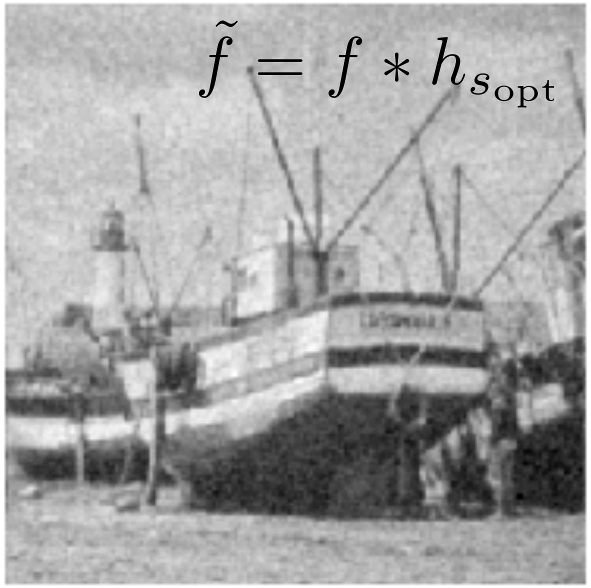






Noisy, SNR=17.6dB

Denoised, SNR=22.3dB



Overview

- Noise in Signals and Images
- Linear Denoising by Blurring
- Non-linear Wavelet Denoising
- Translation Invariant Thresholding
- Other Diagonal Thresholders
- Non-diagonal Block Thresholding
- Data-dependent Noise

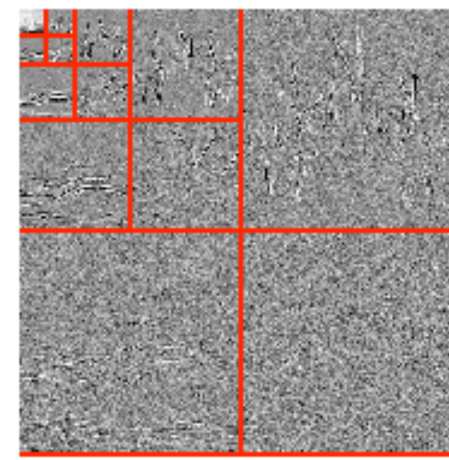
Diagonal Thresholding

Orthogonal basis $\{\psi_m\}_m$ of $\mathbb{R}^N$ (e.g. wavelets).

$$\langle f,\, \psi_m \rangle = \langle f_0,\, \psi_m \rangle + \langle w,\, \psi_m \rangle$$

$$\langle w,\, \psi_m \rangle \sim \mathcal{N}(0, \sigma) \quad \Longrightarrow \quad \text{white noise.}$$

$\langle f,\, \psi_m \rangle$



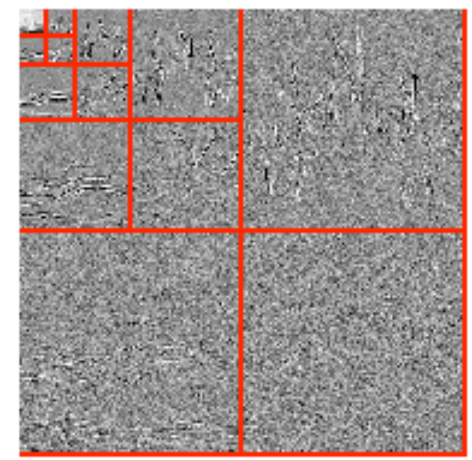
Diagonal Thresholding

Orthogonal basis $\{\psi_m\}_m$ of $\mathbb{R}^N$ (e.g. wavelets).

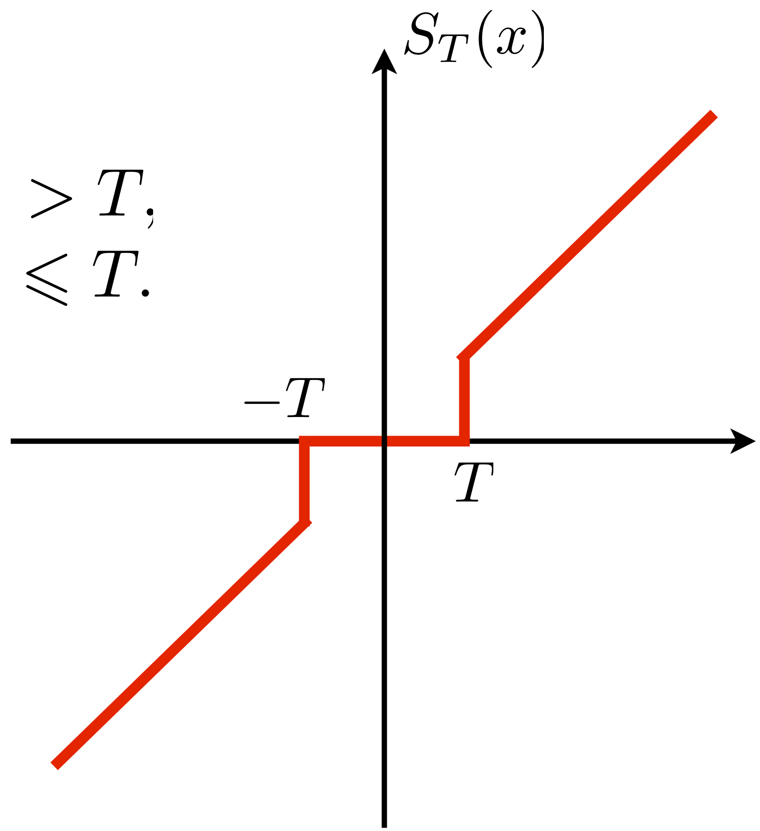
$$\langle f,\, \psi_m \rangle = \langle f_0,\, \psi_m \rangle + \langle w,\, \psi_m \rangle$$

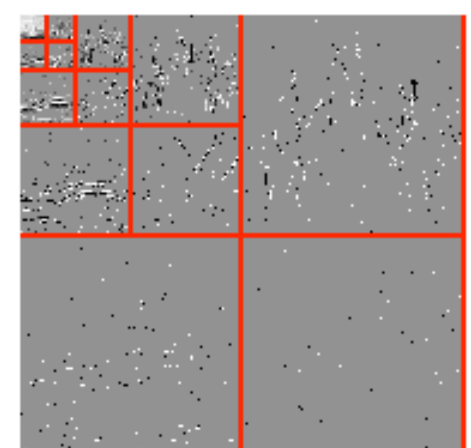
$$\langle w,\, \psi_m \rangle \sim \mathcal{N}(0, \sigma) \quad \Longrightarrow \quad \text{white noise.}$$

$$\tilde{f} = \sum_{|\langle f,\, \psi_m \rangle| > T} \langle f,\, \psi_m \rangle \psi_m = \sum_m S_T(\langle f,\, \psi_m \rangle) \psi_m$$

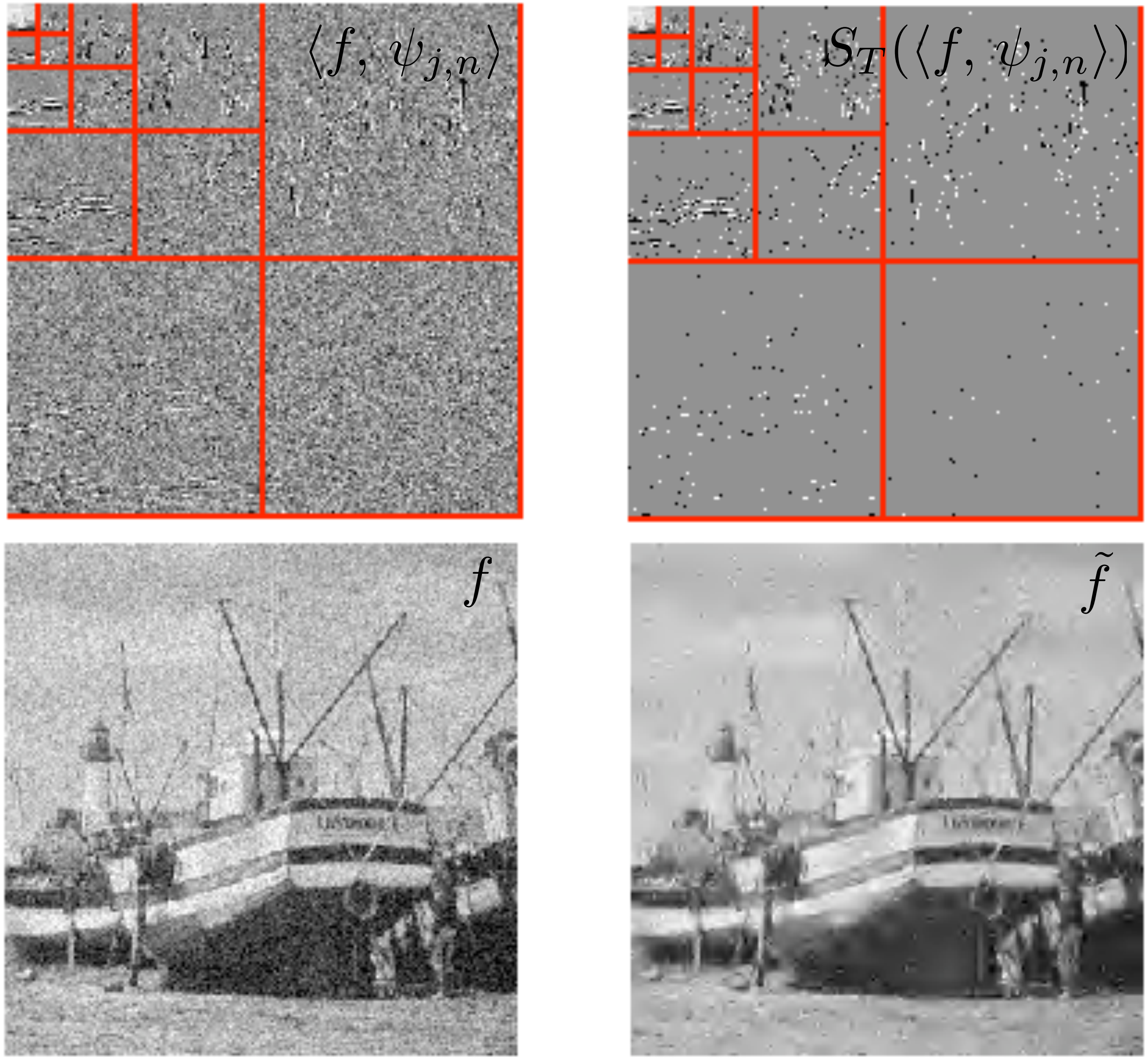


$$S_T(x) = \begin{cases} x & \text{if} \quad |x| > T, \\ 0 & \text{if} \quad |x| \leqslant T. \end{cases}$$





Wavelet Diagonal Hard Thresholding



Hard thresholding
SNR=20.9dB

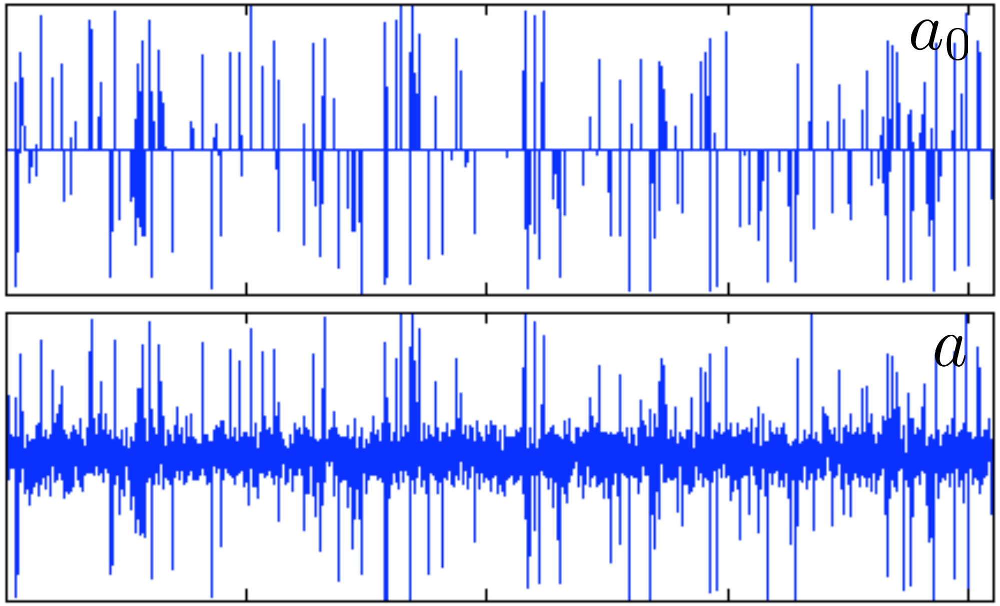
Sparse Signal Estimation

$a_0[m] = \langle f_0, \psi_m \rangle \in \mathbb{R}^N \approx$ sparse signal:

$\|a_0\|_0 = \# \{m \setminus a_0[m] \neq 0\}$ is small.

$f = f_0 + w$, with $w[i] \sim \mathcal{N}(0, \sigma)$

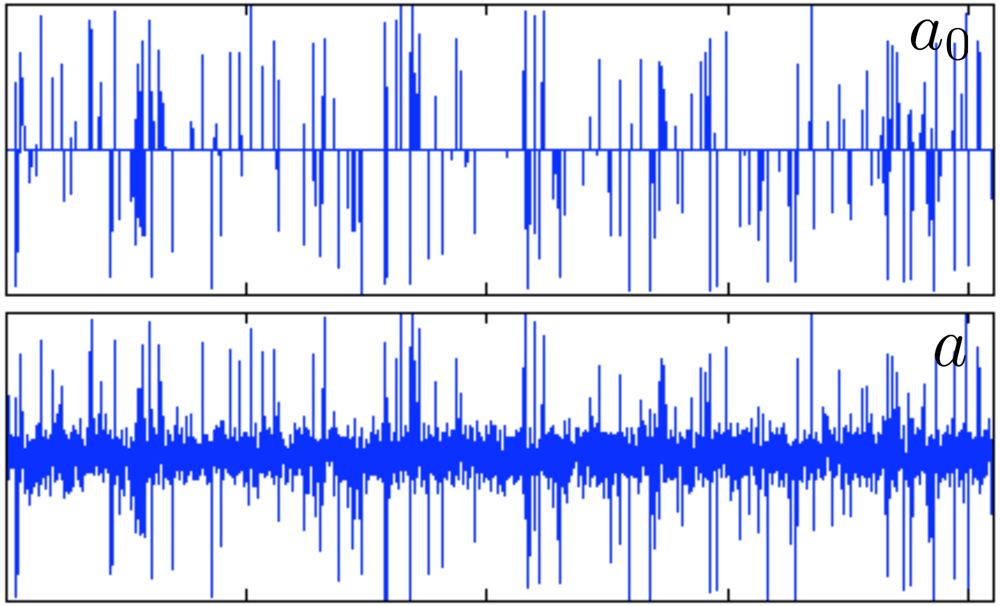
$a = a_0 + z$, with $z[m] \sim \mathcal{N}(0, \sigma)$



Sparse Signal Estimation

$a_0[m] = \langle f_0, \psi_m \rangle \in \mathbb{R}^N \approx$ sparse signal:

$\|a_0\|_0 = \#\{m \setminus a_0[m] \neq 0\}$ is small.



$f = f_0 + w$, with $w[i] \sim \mathcal{N}(0, \sigma)$

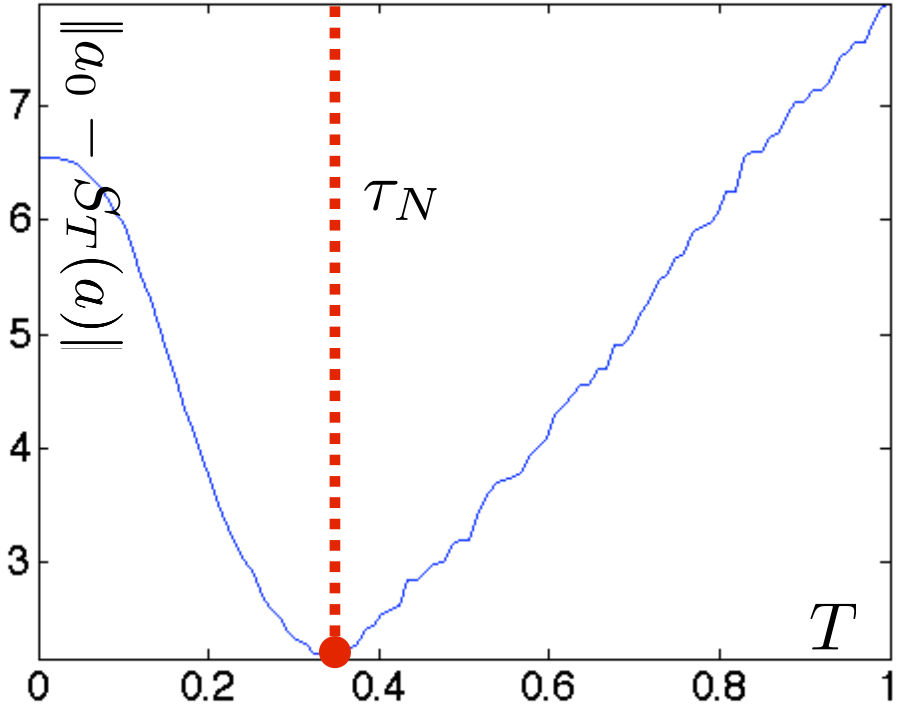
$a = a_0 + z$, with $z[m] \sim \mathcal{N}(0, \sigma)$

If $\displaystyle\min_{m : a_0[m] \neq 0} |a_0[m]|$ is large enough,

$\|f_0 - \tilde{f}\| = \|a_0 - S_T(a)\|$ minimum for

$$T \approx \tau_N = \max_{0 \leqslant m < N} |z[m]|$$

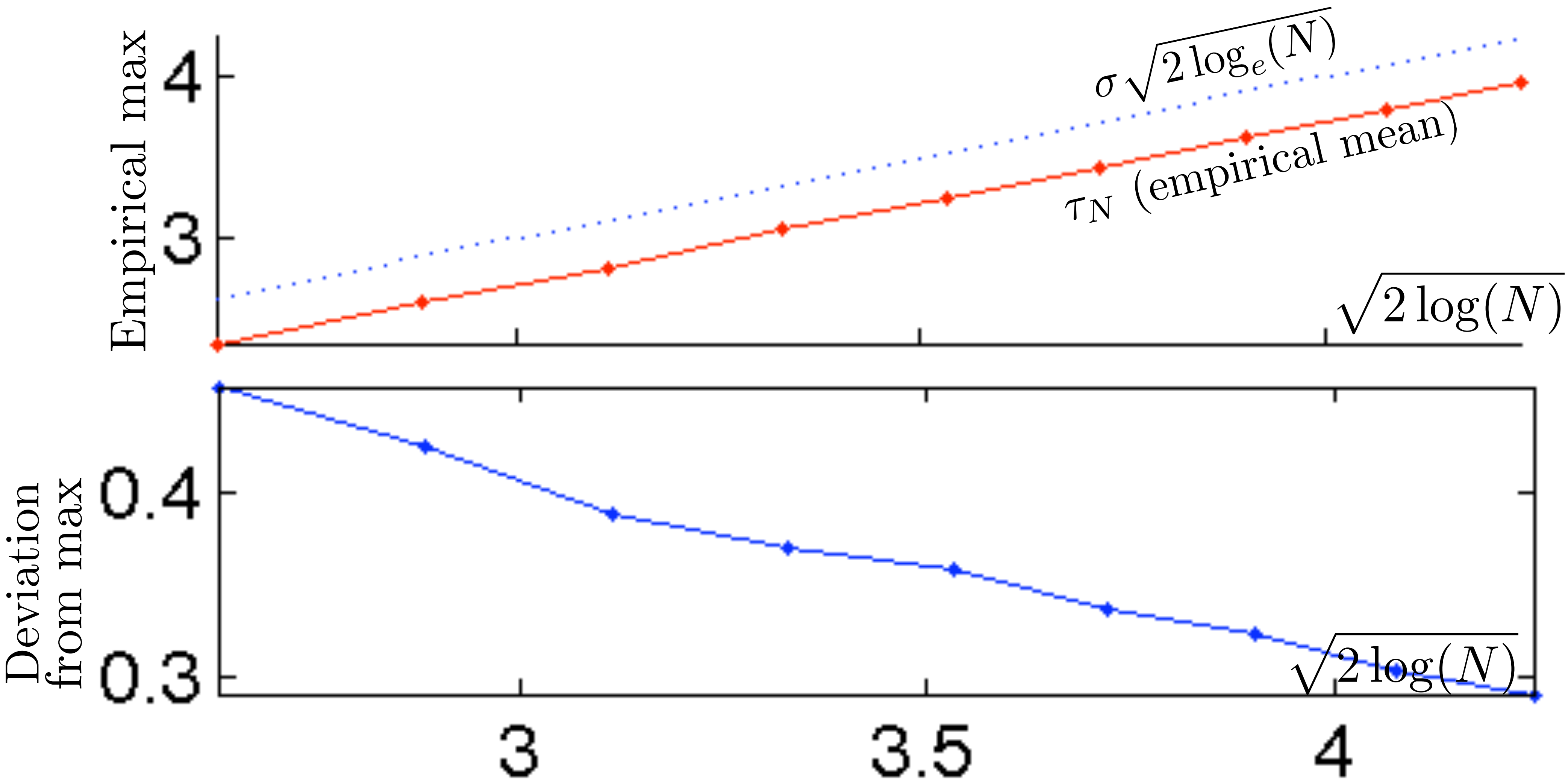
Maximum of N dimensional white noise.



Optimal Threshold Selection

If $z[m] \sim \mathcal{N}(0,\sigma)$ white noise:

Universal threshold: $\tau_N \sim \sigma\sqrt{2\log_e(N)}$.



Non-linear Approximation and Estimation

Random vector: $F = f_0 + W$.

W unit variance white noise.

Realization: $f = f_0 + w$, $w \sim W$.

Estimator (random vector): $$\tilde{F} = \sum_{|\langle F,\, \psi_m \rangle| > T} \langle F,\, \psi_m \rangle \psi_m$$

Performance: decay of the average risk $E_W(\|f_0 - \tilde{F}\|^2)$.

Non-linear Approximation and Estimation

Random vector: $F = f_0 + W$.
W unit variance white noise.

Realization: $f = f_0 + w$, $w \sim W$.

Estimator (random vector): $$\tilde{F} = \sum_{|\langle F,\, \psi_m \rangle| > T} \langle F,\, \psi_m \rangle \psi_m$$

Performance: decay of the average risk $E_W(\|f_0 - \tilde{F}\|^2)$.

Choice of the threshold: universal $T = \sigma\sqrt{2\log(N)}$.

Theorem: [Donoho, Johnstone]

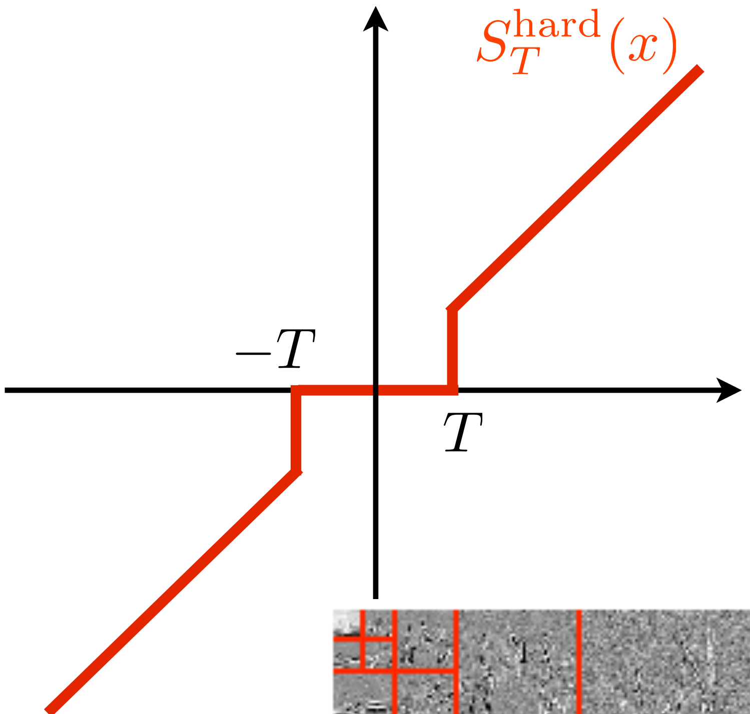
If $\|f_0 - f_{0,M}\|^2 = O(M^{-\alpha})$ then $E(\|f_0 - \tilde{F}\|^2) = O(\sigma^{\frac{2\alpha}{\alpha+1}})$.

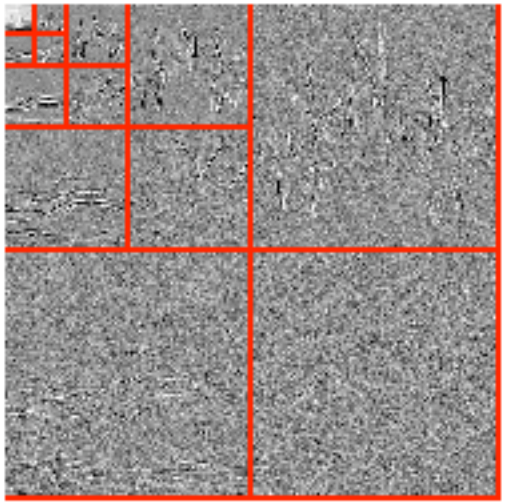
Remark: If $\|f_0 - f_{0,M}\|^2 \sim M^{-\alpha}$, then $\|f_0 - f_{0,M}\|^2 \sim MT^2 \sim T^{\frac{2\alpha}{\alpha+1}}$.

Hard vs. Soft Thresholding

Hard thresholding: $$S_T^{\text{hard}}(x) = \begin{cases} x & \text{if} \quad |x| > T, \\ 0 & \text{if} \quad |x| \leqslant T. \end{cases}$$

$$\tilde{f}^{\text{h}} = \sum_m S_T^{\text{hard}}(\langle f,\, \psi_m \rangle)\psi_m$$





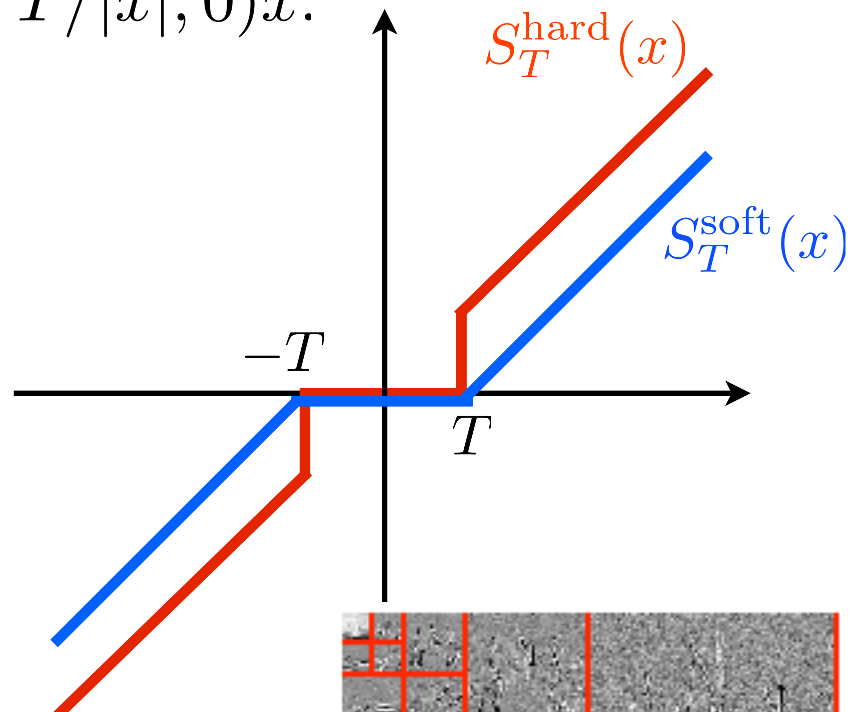
Hard vs. Soft Thresholding

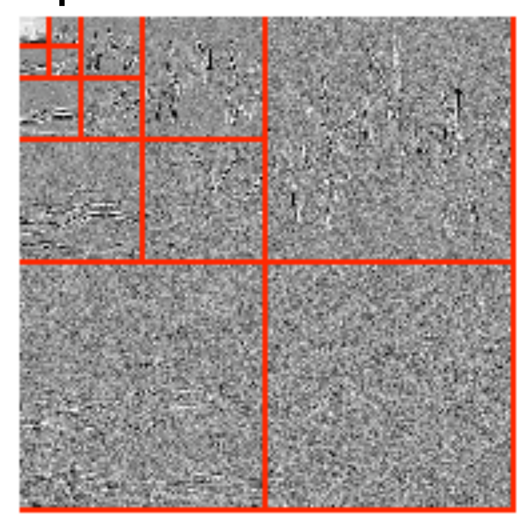
Hard thresholding: $$S_T^{\text{hard}}(x) = \begin{cases} x & \text{if} \quad |x| > T, \\ 0 & \text{if} \quad |x| \leqslant T. \end{cases}$$

Soft thresholding: $$S_T^{\text{soft}}(x) = \max(1 - T/|x|, 0)x.$$

$$\tilde{f}^{\text{h}} = \sum_m S_T^{\text{hard}}(\langle f,\, \psi_m \rangle)\psi_m$$

$$\tilde{f}^{\text{s}} = \sum_m S_T^{\text{soft}}(\langle f,\, \psi_m \rangle)\psi_m$$

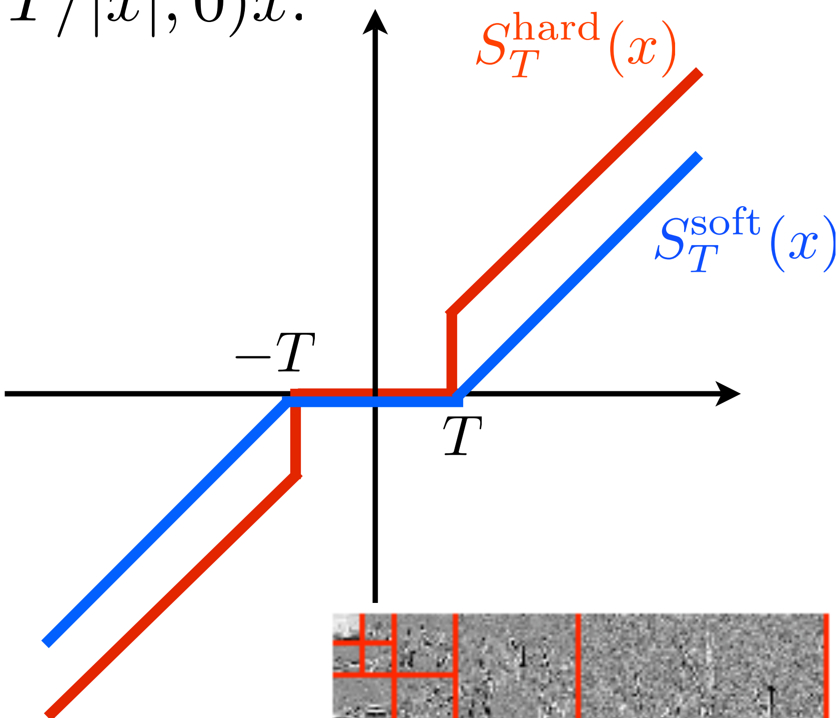




Hard vs. Soft Thresholding

Hard thresholding: $$S_T^{\text{hard}}(x) = \begin{cases} x & \text{if} \quad |x| > T, \\ 0 & \text{if} \quad |x| \leqslant T. \end{cases}$$

Soft thresholding: $$S_T^{\text{soft}}(x) = \max(1 - T/|x|, 0)x.$$

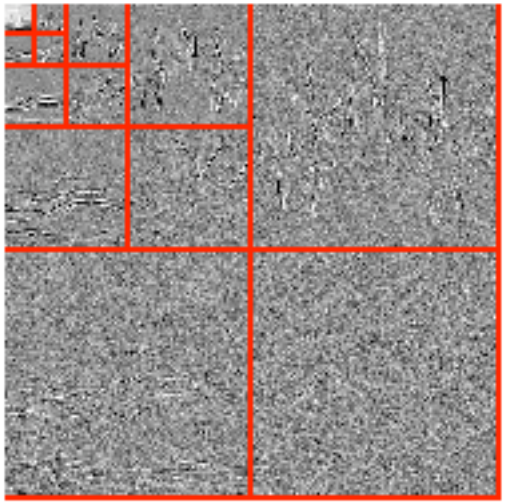


$$\tilde{f}^{\text{h}} = \sum_m S_T^{\text{hard}}(\langle f,\, \psi_m \rangle)\psi_m$$

$$\tilde{f}^{\text{s}} = \sum_m S_T^{\text{soft}}(\langle f,\, \psi_m \rangle)\psi_m$$

Soft thresholding produces smoother denoising, but introduces a bias.

$\Longrightarrow$ do not soft threshold low pass residual.



Optimal Threshold

"Oracle" threshold optimization: find $T = T_{\text{opt}}$ that minimizes

$$\mathrm{T}^{\text{hard}}_{\text{opt}} = \underset{T}{\operatorname{argmin}} \|f_0 - \tilde{f}^{\text{h}}\|^2$$

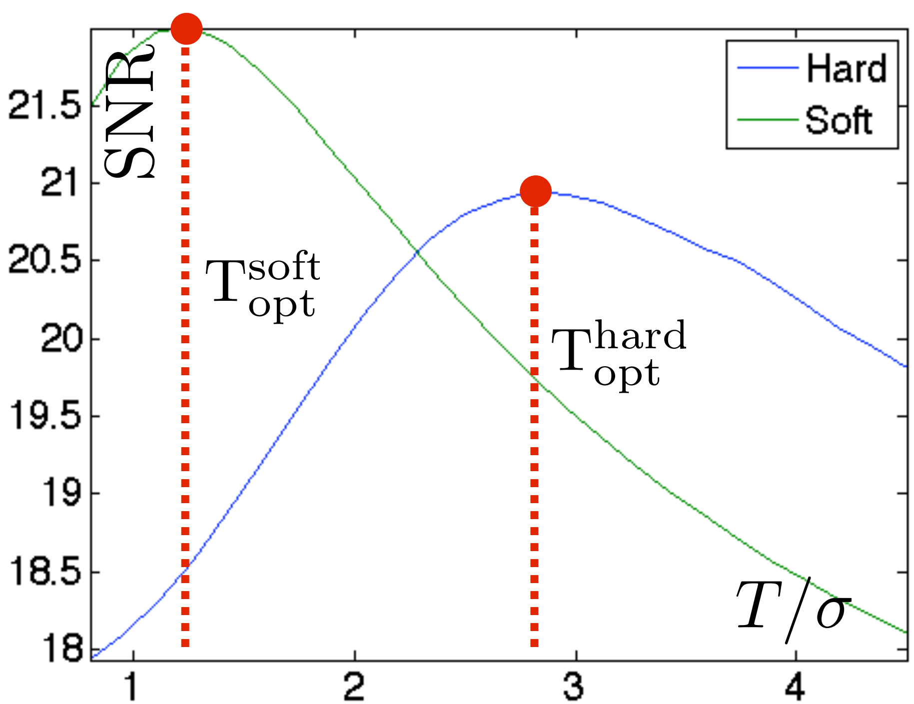
$$\mathrm{T}^{\text{soft}}_{\text{opt}} = \underset{T}{\operatorname{argmin}} \|f_0 - \tilde{f}^{\text{s}}\|^2$$

$$\mathrm{SNR}(f_0, \tilde{f}) = -10\log_{10}\left(\frac{\|f_0 - \tilde{f}\|}{\|f_0\|}\right)$$

(expressed in dB)

Empirical finding: $T^{\text{hard}}_{\text{opt}} \approx 3\sigma, \quad T^{\text{soft}}_{\text{opt}} \approx \frac{3}{2}\sigma,$

For natural images, orthogonal wavelets: "Hard < Soft".



Hard, SNR=20.9dB

Soft, SNR=21.8dB

Overview

- Noise in Signals and Images
- Linear Denoising by Blurring
- Non-linear Wavelet Denoising
- Translation Invariant Thresholding
- Other Diagonal Thresholders
- Non-diagonal Block Thresholding
- Data-dependent Noise

Translation Invariant Denoising

$$\begin{tikzcd} f \arrow[r, "\mathcal{D}"] \arrow[d, "\tau"'] & \mathcal{D}(f) \\ f_\tau \arrow[r, "\mathcal{D}"'] & \mathcal{D}(f_\tau) \arrow[u, "-\tau"'] \end{tikzcd}$$

Denoising: $\mathcal{D}(f) = \sum_m S_T(\langle f,\, \psi_m \rangle)\psi_m$.

Translation: $f_\tau(x) = f(x - \tau)$.

Translation invariant denoising: $\mathcal{D}(f) = \mathcal{D}(f_\tau)_{-\tau}$.

Translation Invariant Denoising

$$\begin{tikzcd} f \arrow[r, "\mathcal{D}"] \arrow[d, "\tau"'] & \mathcal{D}(f) \\ f_\tau \arrow[r, "\mathcal{D}"'] & \mathcal{D}(f_\tau) \arrow[u, "-\tau"'] \end{tikzcd}$$

Denoising: $\mathcal{D}(f) = \sum_m S_T(\langle f,\, \psi_m \rangle)\psi_m$.

Translation: $f_\tau(x) = f(x - \tau)$.

Translation invariant denoising: $\mathcal{D}(f) = \mathcal{D}(f_\tau)_{-\tau}$.

Orthogonal wavelet frame $\{\psi_m = \psi_{j,n}\}_{j,n}$ of $\mathbb{R}^N$:

$\longrightarrow$ not translation invariant denoising.

Translation Invariant Denoising

$$\begin{tikzcd} f \arrow[r, "\mathcal{D}"] \arrow[d, "\tau"'] & \mathcal{D}(f) \\ f_\tau \arrow[r, "\mathcal{D}"'] & \mathcal{D}(f_\tau) \arrow[u, "-\tau"'] \end{tikzcd}$$

Denoising: $\mathcal{D}(f) = \sum_m S_T(\langle f, \psi_m \rangle)\psi_m$.

Translation: $f_\tau(x) = f(x - \tau)$.

Translation invariant denoising: $\mathcal{D}(f) = \mathcal{D}(f_\tau)_{-\tau}$.

Orthogonal wavelet frame $\{\psi_m = \psi_{j,n}\}_{j,n}$ of $\mathbb{R}^N$:

$\longrightarrow$ not translation invariant denoising.

Cycle spining: "averaging to make invariant".

$$\mathcal{D}_{\text{inv}}(f) = \frac{1}{N} \sum_{\tau=0}^{N-1} \mathcal{D}(f_{\frac{\tau}{N}})_{-\frac{\tau}{N}}$$

$\longrightarrow$ invariant by translations k/N.

$\longrightarrow$ $O(N^2)$ operations.

Translation Invariant Wavelets

Translation invariant redundant frame $\{\psi_m\}_m$ of $\mathbb{R}^N$:

$$\forall\, k,\ \forall\, \tau,\ (\psi_k)_\tau \in \{\psi_m\}_m.$$

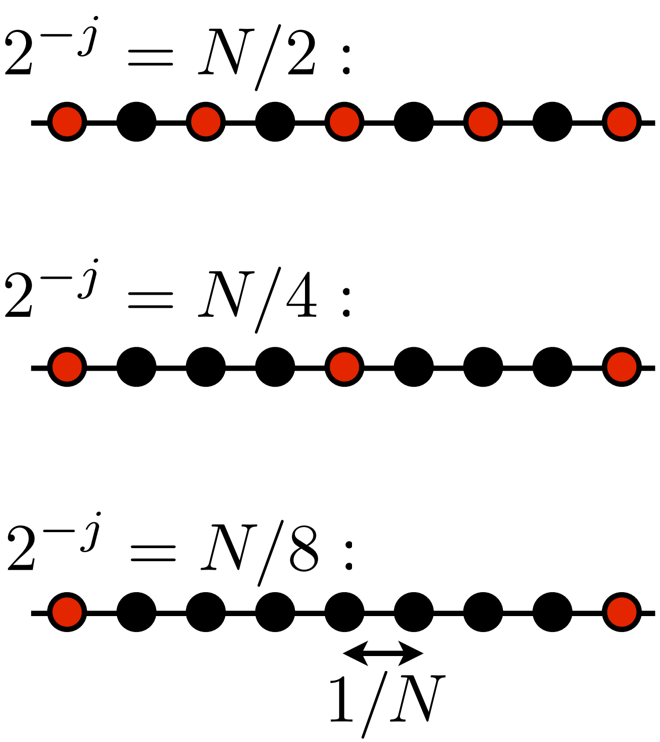
Dyadic translation:

$$\left\{ \psi_{j,n}(x) = \frac{1}{2^{j/2}} \psi\left(\frac{x}{2^j} - \frac{n}{2^j}\right) \right\}_{j_0 \leqslant j < 0}^{0 \leqslant n < 2^{-j}}$$

$\longrightarrow$ orthogonal basis of a space $V_{j_0} \simeq \mathbb{R}^N$.

$$(\psi_{j,n})_\tau \notin \{\psi_{j,n}\} \quad \text{for} \quad \tau = 2^j/2$$

$\longrightarrow$ no translation invariance.



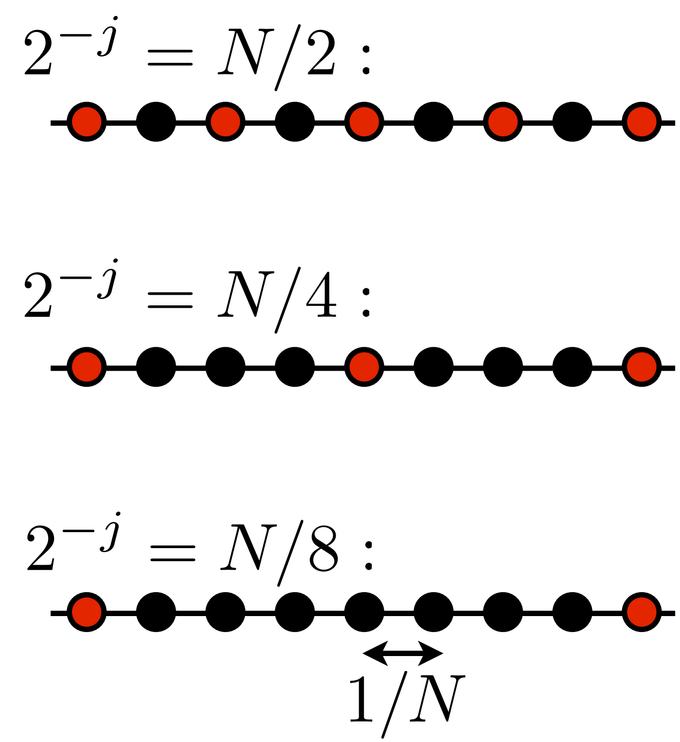
Translation Invariant Wavelets

Translation invariant redundant frame $\{\psi_m\}_m$ of $\mathbb{R}^N$:

$$\forall\, k,\ \forall\, \tau,\ (\psi_k)_\tau \in \{\psi_m\}_m.$$

Dyadic translation:

$$\left\{ \psi_{j,n}(x) = \frac{1}{2^{j/2}} \psi\left(\frac{x}{2^j} - \frac{n}{2^j}\right) \right\}_{j_0 \leqslant j < 0}^{0 \leqslant n < 2^{-j}}$$



$\longrightarrow$ orthogonal basis of a space $V_{j_0} \simeq \mathbb{R}^N$.

$$(\psi_{j,n})_\tau \notin \{\psi_{j,n}\} \quad \text{for} \quad \tau = 2^j/2$$

$\longrightarrow$ no translation invariance.

Uniform translation:

$$\left\{ \bar{\psi}_{j,n}(x) = \frac{1}{2^{j/2}} \psi\left(\frac{x}{2^j} - \frac{n}{N}\right) \right\}_{j_0 \leqslant j < 0}^{0 \leqslant n < N}$$

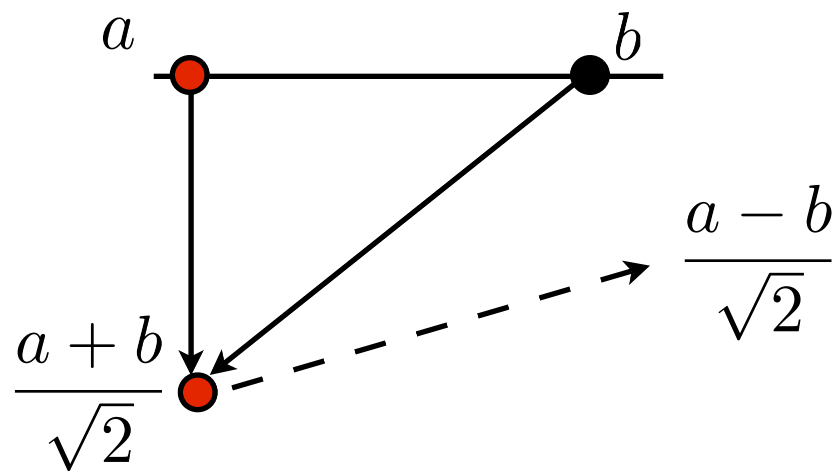
$\longrightarrow$ redundant familly of a space $V_{j_0} \simeq \mathbb{R}^N$ with $N \log_2(N)$ vectors.

Translation Invariant Haar (1D)

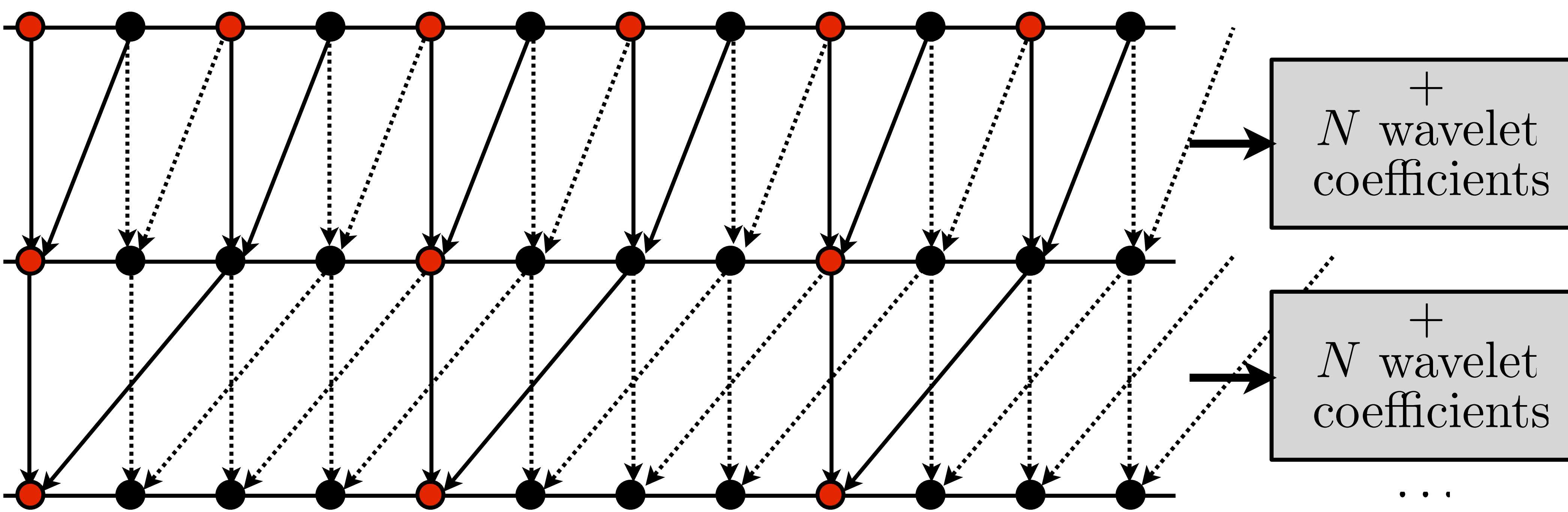
Fast algorithm: no sub-sampling, $O(N \log_2(N))$ operations.

Computes the coefficients

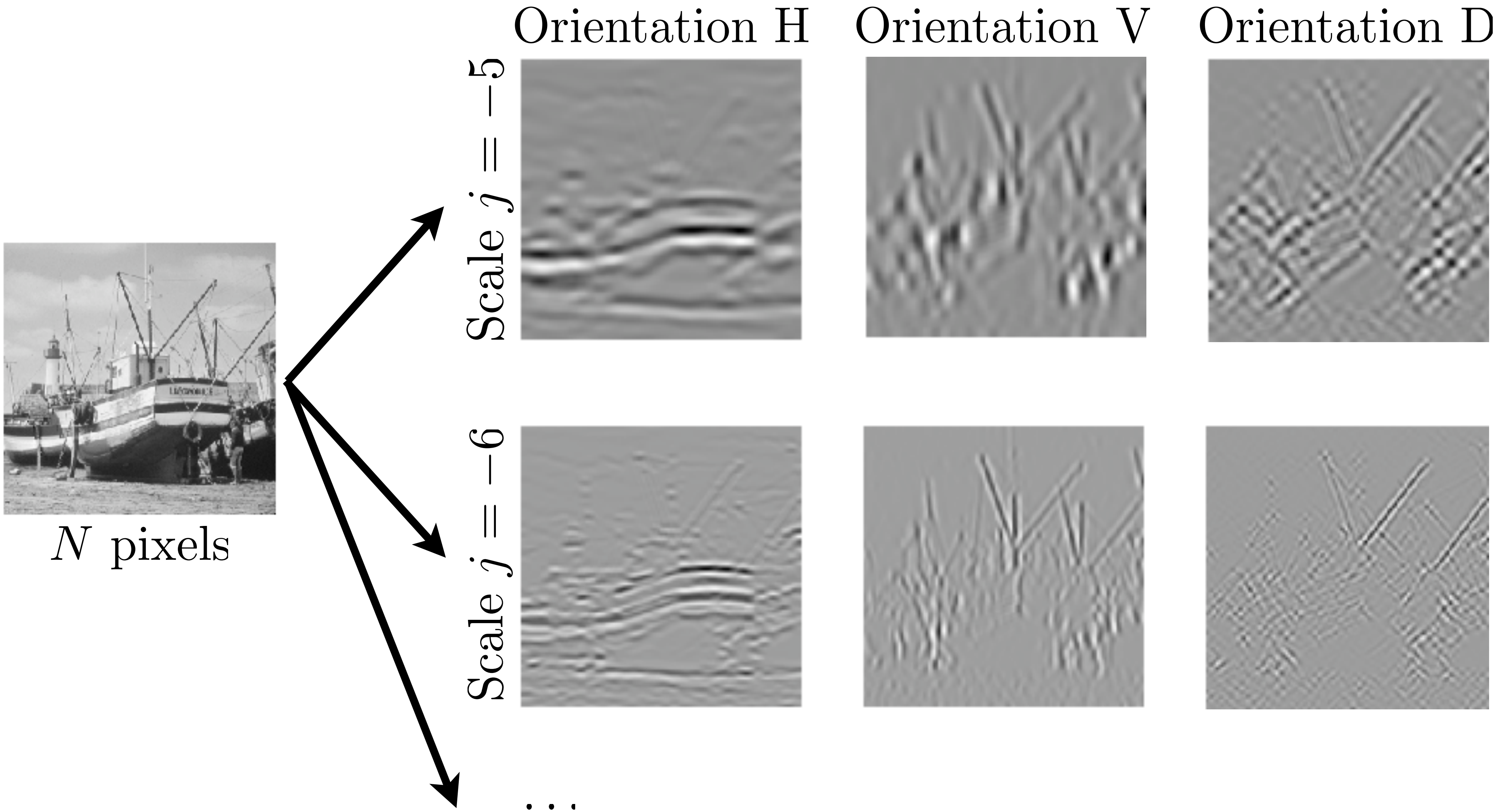
$$\langle f, \psi_{j,n} \rangle \quad \text{for} \quad \begin{cases} -\log_2(N) < j \leqslant 0 \\ 0 \leqslant n < N \end{cases}$$



Basic step: average / difference.

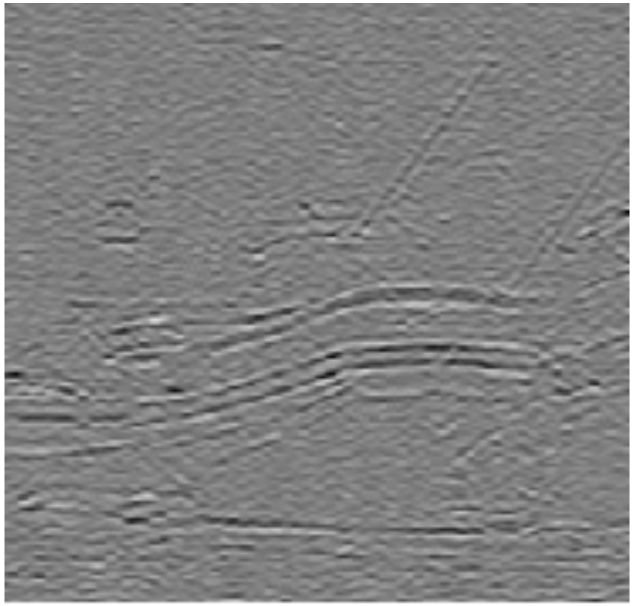


Translation Invariant Transform (2D)

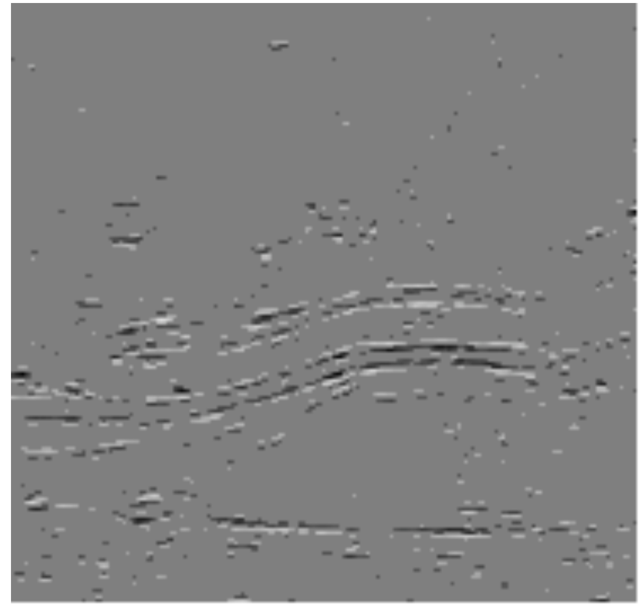


$(3J+1)N$ coefficients, $J = \#\text{scales} \leqslant \log_2(N)/2$

Translation Invariant Thresholding



Noisy coefficients



Thresholded coefficients

Soft, orth. SNR=21.8dB

Hard, TI, SNR=22.7dB

Optimal Invariant Threshold

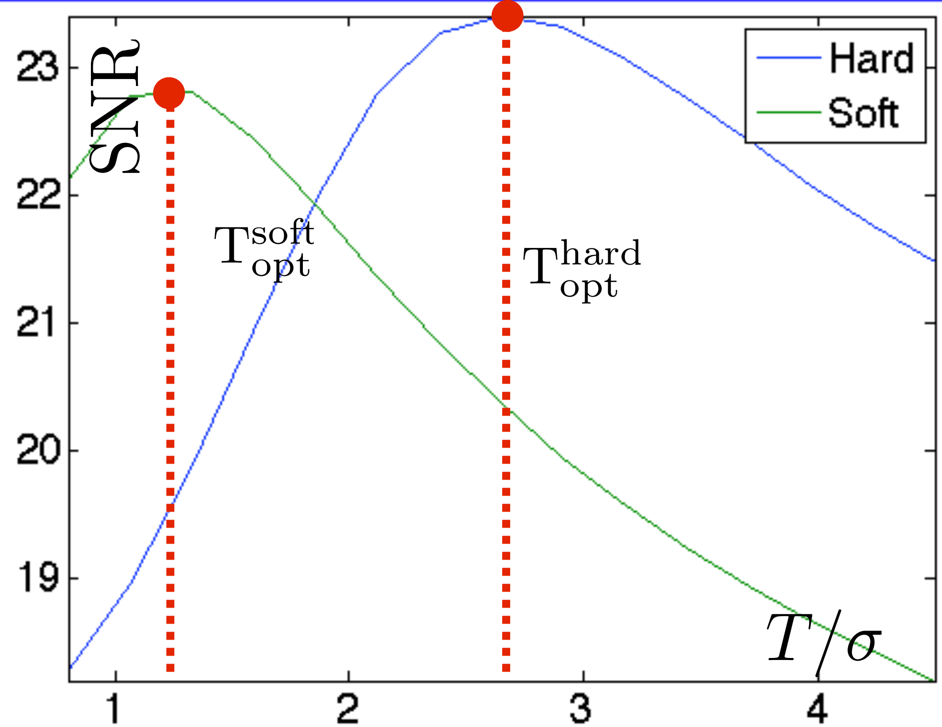
Empirical finding:

$$T^{\mathrm{hard}}_{\mathrm{opt}} \approx 3\sigma, \quad T^{\mathrm{soft}}_{\mathrm{opt}} \approx \tfrac{3}{2}\sigma,$$

For natural images,

translation invariant wavelets:

“Hard > Soft”.



Soft, orth. SNR=21.8dB

Hard, TI, SNR=22.7dB

Overview

- Noise in Signals and Images
- Linear Denoising by Blurring
- Non-linear Wavelet Denoising
- Translation Invariant Thresholding
- Other Diagonal Thresholders
- Non-diagonal Block Thresholding
- Data-dependent Noise

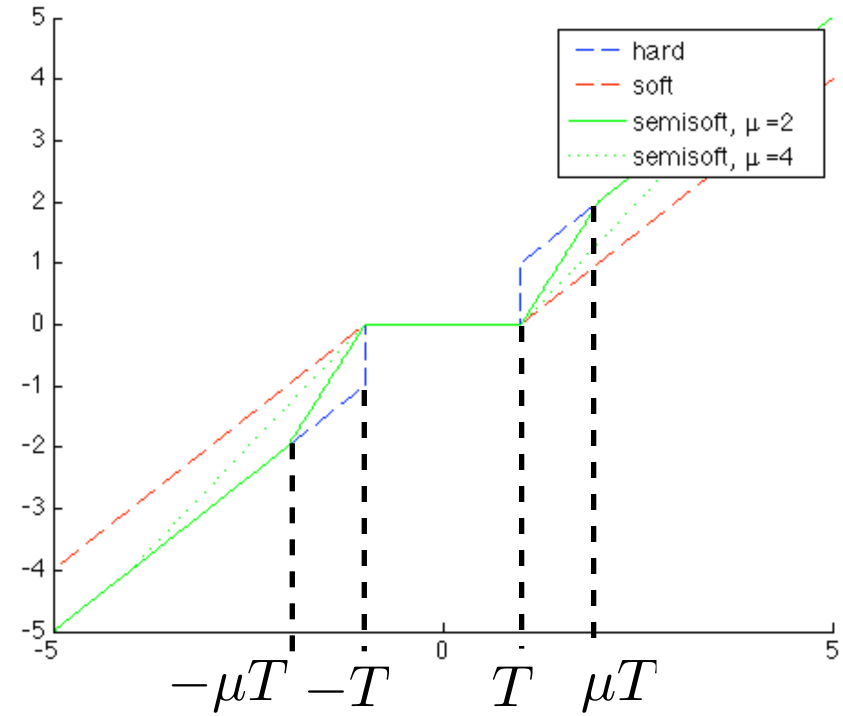
Between Hard and Soft Thresholding

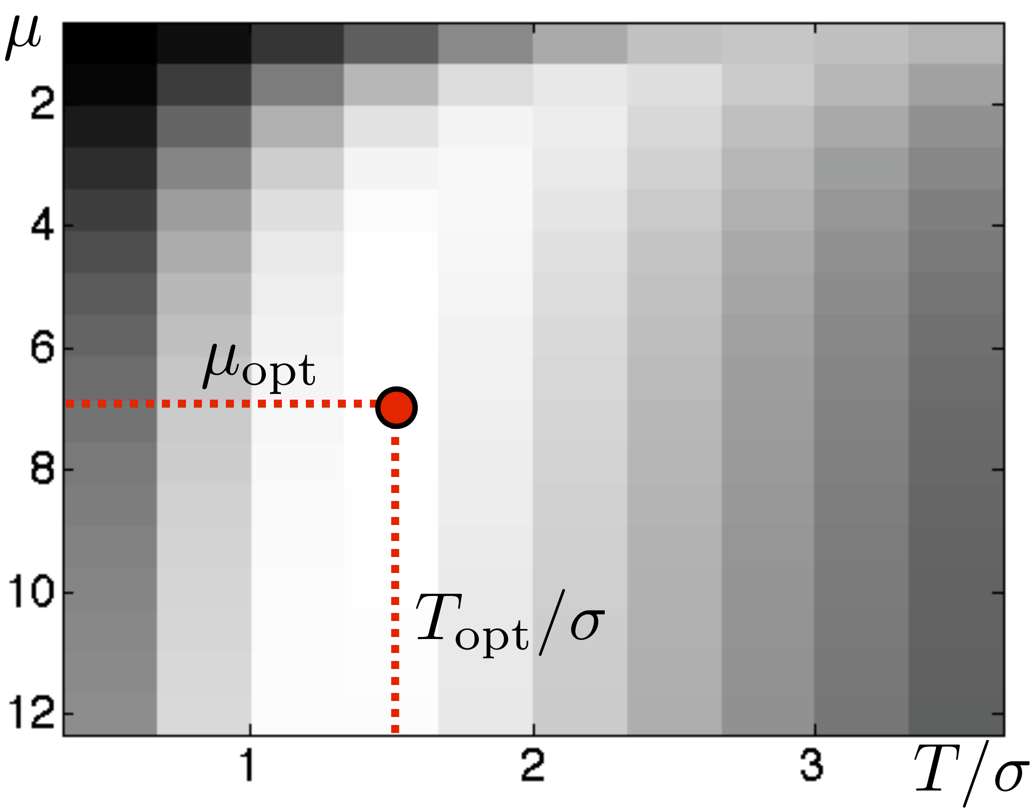
Soft-hard thresholding:

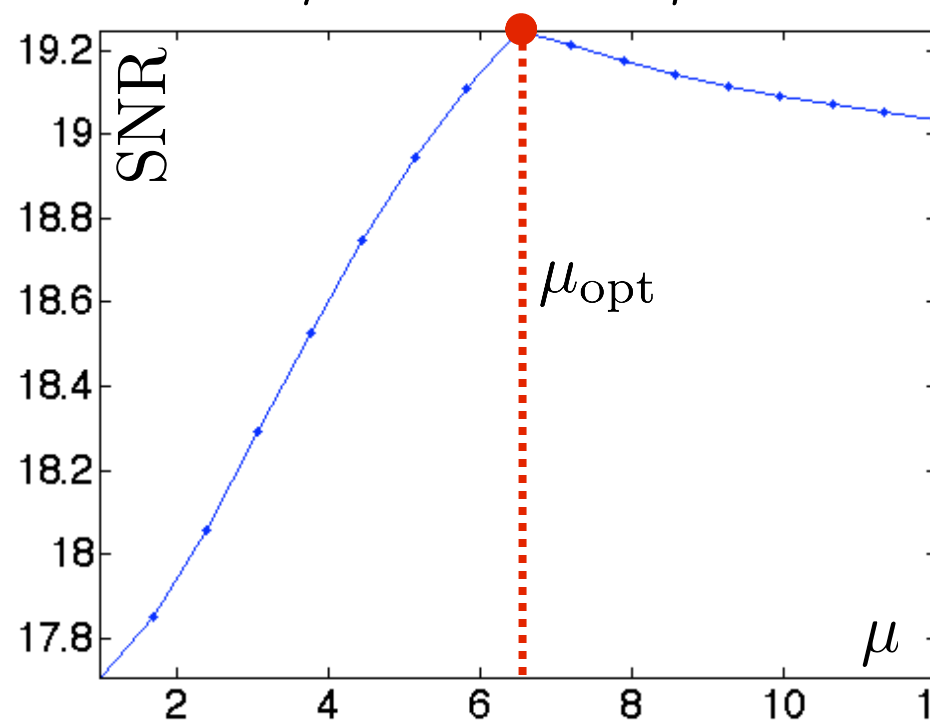
S_T^{μ} parameterized by $\mu > 1$.

$\mu = 1$: hard thresholding.

$\mu = +\infty$: soft thresholding.



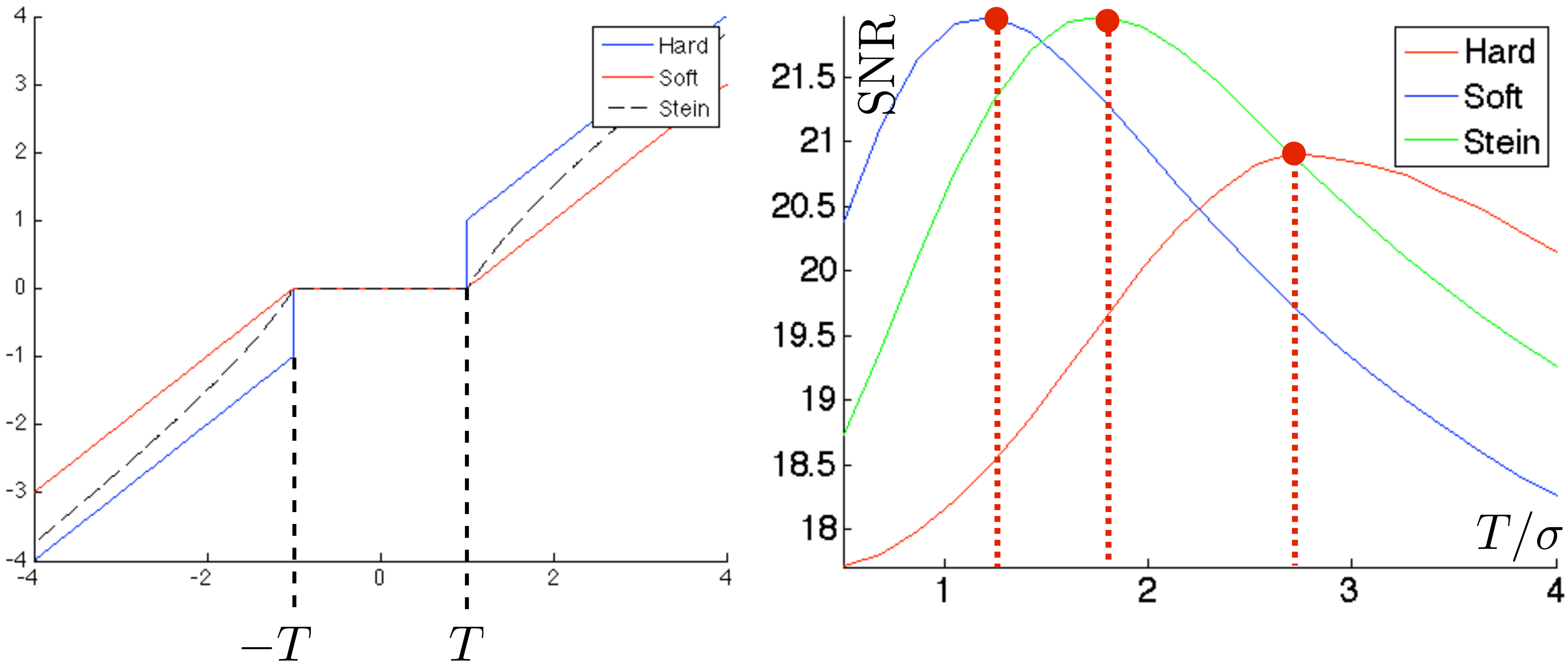




Stein Quadratic-Soft Thresholder

Soft thresholding: $$S_T^{\text{soft}}(x) = \max\left(1 - \frac{T}{|x|}, 0\right) x$$

Stein thresholding: $$S_T^{\text{Stein}}(x) = \max\left(1 - \frac{T^2}{|x|^2}, 0\right) x$$



Overview

- Noise in Signals and Images
- Linear Denoising by Blurring
- Non-linear Wavelet Denoising
- Translation Invariant Thresholding
- Other Diagonal Thresholders
- Non-diagonal Block Thresholding
- Data-dependent Noise

Block Thresholding

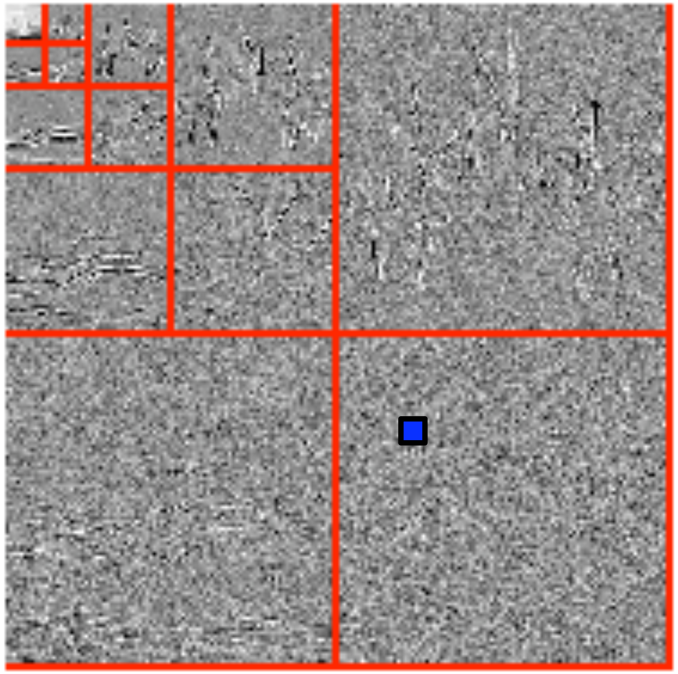
Block of $s \times s$ coefficients: $\{(j,n)\}_{j,n} = \bigcup_k B_k$

Block energy: $$E_k = \frac{1}{s^2} \sum_{m \in B_k} |\langle f,\, \psi_m \rangle|^2$$

For $m = (j,n) \in B_k$, block thresholding:

$$S_T^{\text{block}}(\langle f,\, \psi_m \rangle) = A_T(E_k) \langle f,\, \psi_m \rangle$$

where A_T is an attenuation function



Block Thresholding

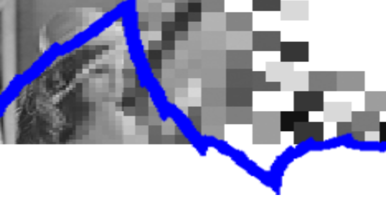
Block of $s \times s$ coefficients: $\{(j,n)\}_{j,n} = \bigcup_k B_k$

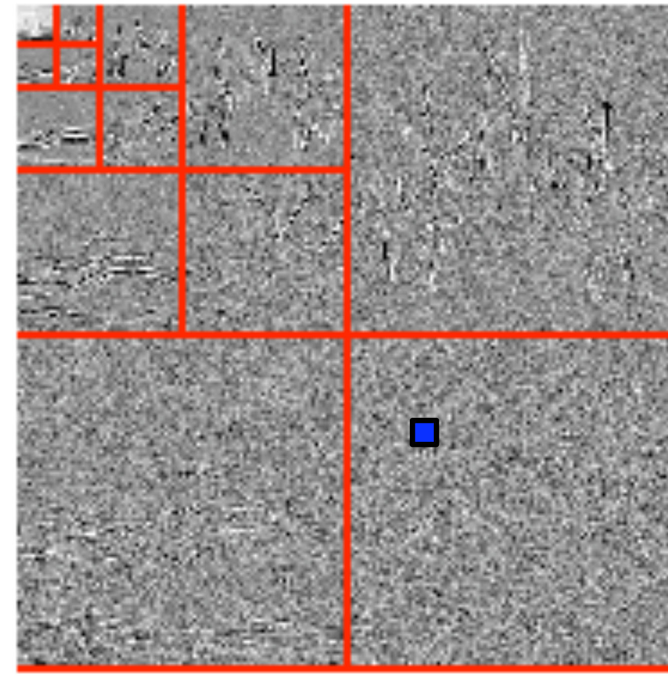
Block energy: $$E_k = \frac{1}{s^2} \sum_{m \in B_k} |\langle f,\, \psi_m \rangle|^2$$

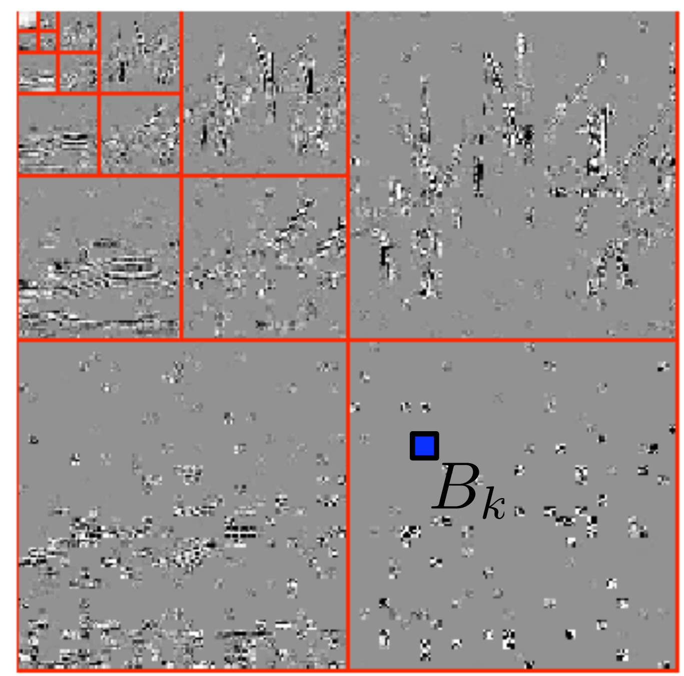
For $m = (j,n) \in B_k$, block thresholding:

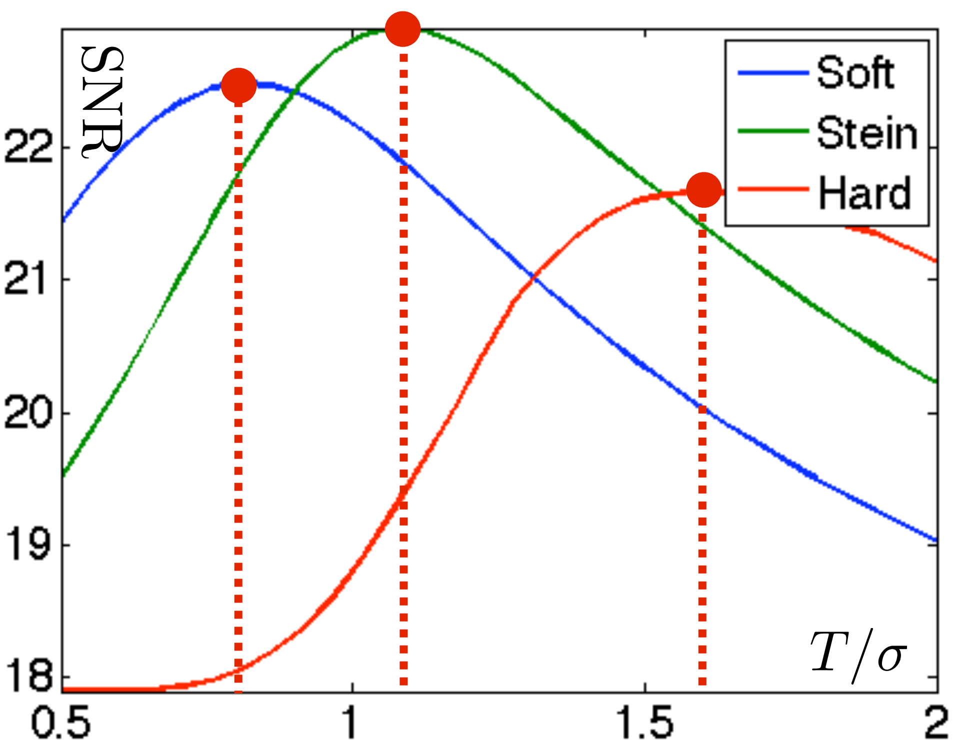
$$S_T^{\text{block}}(\langle f,\, \psi_m \rangle) = A_T(E_k) \langle f,\, \psi_m \rangle$$

where A_T is an attenuation function









$A_T(x) =$

$\max(1 - \frac{x^2}{T^2}, 0)$

$\max(1 - \frac{|x|}{T}, 0)$

$1_{|x|>T}(x)$

Optimal Block Choice

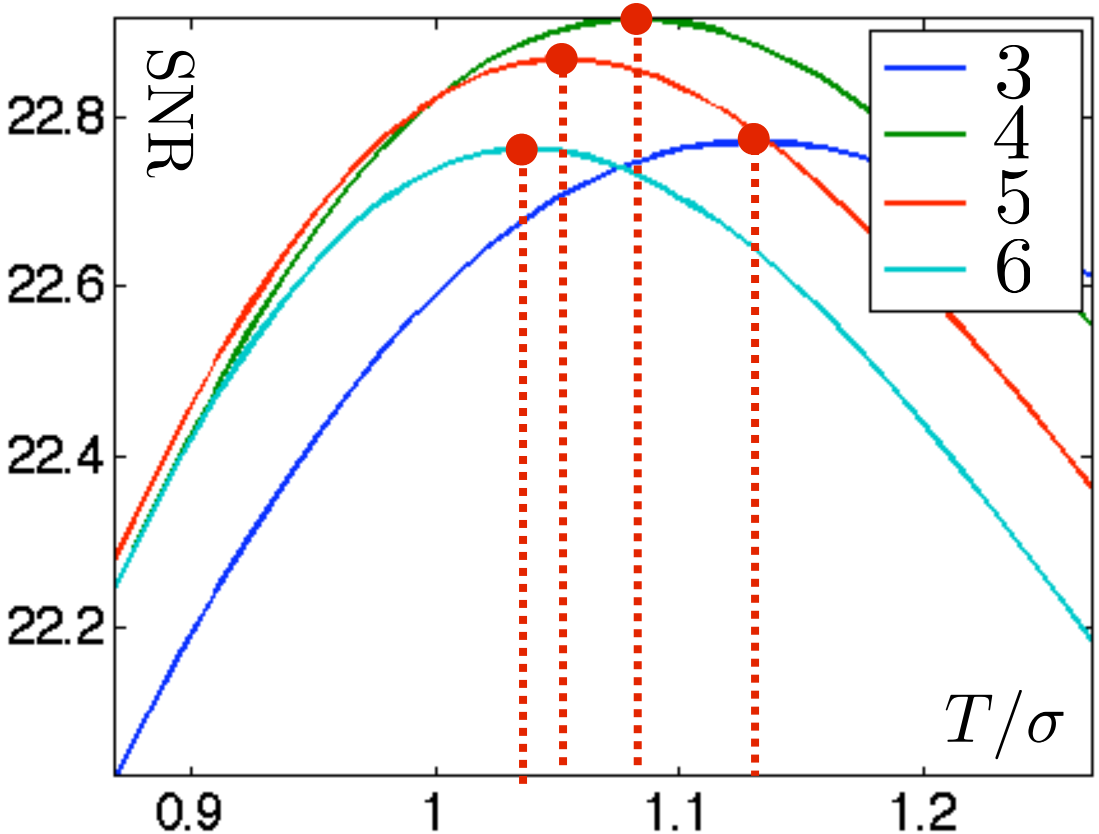
Non-diagonal (block) estimator.

Same thresholding decision for all $(j, n) \in B_k$.

Uses the structure (redundancies) of coefficients of natural images.

$\longrightarrow$ edges generate clusters of large coefficients.

Size s of the blocks



Block denoising, SNR=22.8dB

Comparison

Hard orthogonal

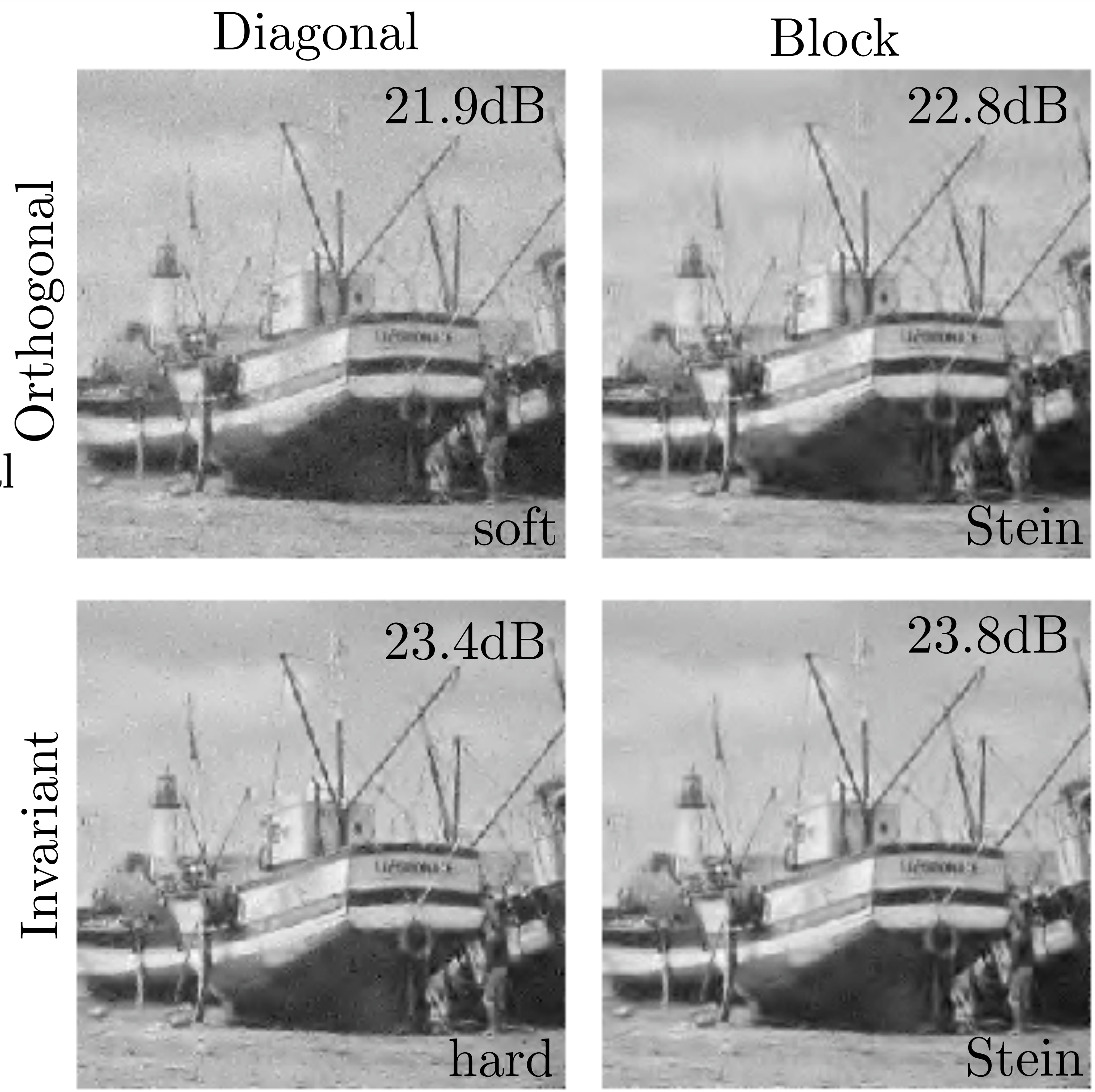
Soft orthogonal

Stein block orthogonal

Hard invariant

Stein block invariant

SNR



Overview

- Noise in Signals and Images
- Linear Denoising by Blurring
- Non-linear Wavelet Denoising
- Translation Invariant Thresholding
- Other Diagonal Thresholders
- Non-diagonal Block Thresholding
- Data-dependent Noise

Poisson Noise

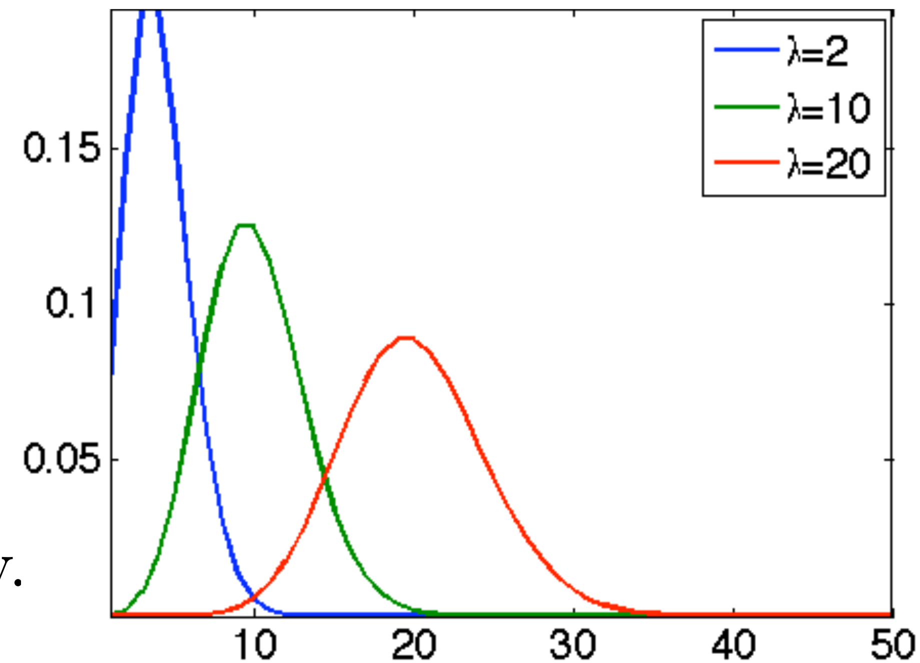
Poisson noise: $f[i] \sim \mathcal{P}(\lambda)$

where $\lambda = \lambda_i = f_0[i]$.

$$\mathbb{P}(f[i] = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

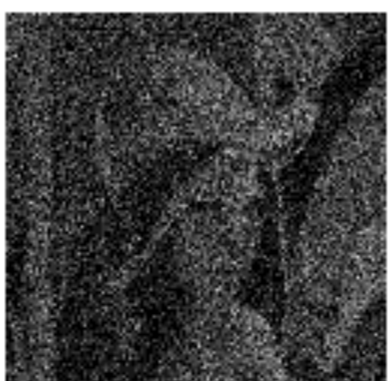
$$\mathrm{E}(f[i]) = \lambda \qquad \mathrm{Var}(f[i]) = \lambda$$

$\rightarrow$ counting photons, $f_0[i]$ =intensity.



Example: digital camera, confocal microscopy,

TEP and SPECT tomography, ...



$\lambda_{\max} = 5$

$\lambda_{\max} = 10$

$\lambda_{\max} = 50$

$\lambda_{\max} = 100$

Poisson Noise Variance Stabilization

Variance stabilization: non-linear mapping $\varphi : \mathbb{R} \to \mathbb{R}$

such that $\varphi(f[i]) \simeq \mathcal{N}(\varphi(a[i]), \sigma)$, σ fixed.

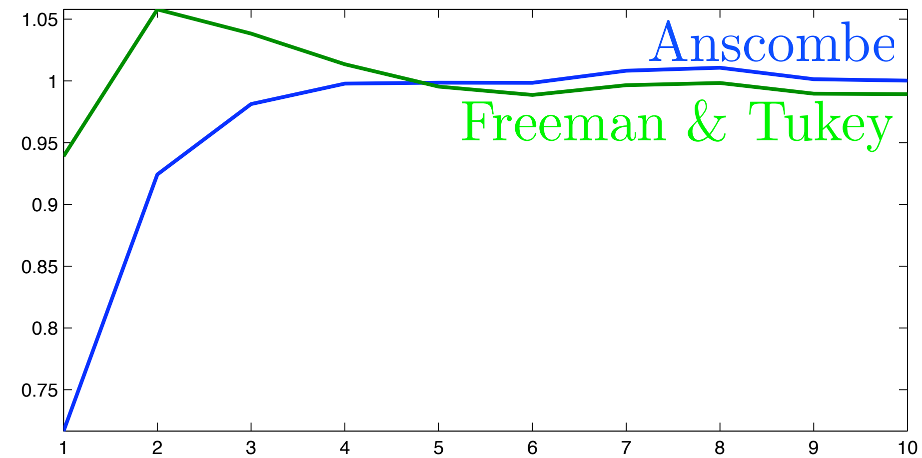
Poisson Noise Variance Stabilization

Variance stabilization: non-linear mapping $\varphi : \mathbb{R} \rightarrow \mathbb{R}$

such that $\varphi(f[i]) \simeq \mathcal{N}(\varphi(a[i]), \sigma)$, σ fixed.

Anscombes v.s.: $\varphi(x) = 2\sqrt{x + 3/8}$

Freeman & Tukey v.s.: $\varphi(x) = \sqrt{x+1} + \sqrt{x}$



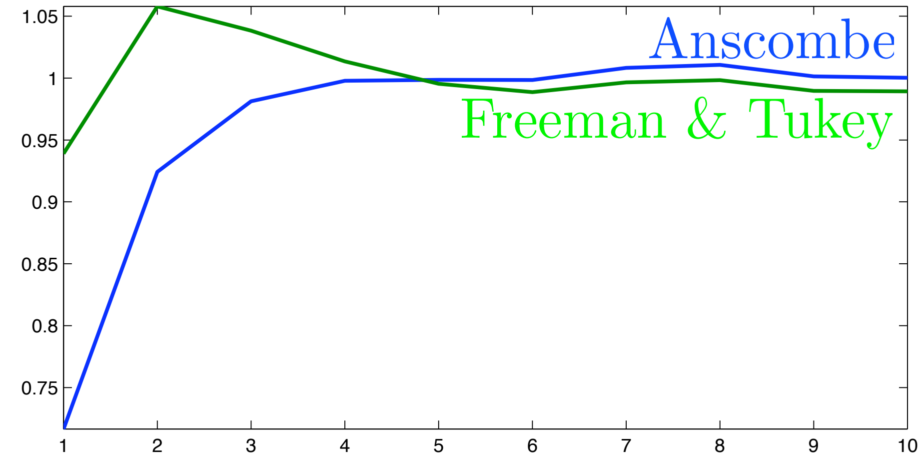
Poisson Noise Variance Stabilization

Variance stabilization: non-linear mapping $\varphi : \mathbb{R} \to \mathbb{R}$

such that $\varphi(f[i]) \simeq \mathcal{N}(\varphi(a[i]), \sigma)$, σ fixed.

Anscombes v.s.: $\varphi(x) = 2\sqrt{x + 3/8}$

Freeman & Tukey v.s.: $\varphi(x) = \sqrt{x+1} + \sqrt{x}$



Stabilized denoising:

$$\tilde{f} = \varphi^{-1}\Big(\mathrm{Denoise}(\varphi(f))\Big)$$

Un-stabilized
SNR=20.5dB

Stabilized
SNR=21.4dB

Multiplicative Noise

Multiplicative noise: $f = f_0 \cdot w$ where $w \sim W$, $\mathrm{E}(W) = 1$.

$$\mathrm{E}(f[i]) = f_0[i], \qquad \mathrm{Var}(f[i]) = f_0[i]^2\sigma^2 \quad \text{where} \quad \sigma^2 = \mathrm{Var}(W).$$

Multiplicative Noise

Multiplicative noise: $f = f_0 \cdot w$ where $w \sim W$, $\mathrm{E}(W) = 1$.

$$\mathrm{E}(f[i]) = f_0[i], \qquad \mathrm{Var}(f[i]) = f_0[i]^2\sigma^2 \quad \text{where} \quad \sigma^2 = \mathrm{Var}(W).$$

Example: SAR images.

Each image: $f^{(s)} = f_0 \cdot w^{(s)} + n^{(s)}$

$n^{(s)}$: white additive noise

$w^{(s)}$: one-sided exponential

$p(w^{(s)}[i] = x) \propto e^{-x}\, \mathbb{I}_{x>0}$

Multiplicative Noise

Multiplicative noise: $f = f_0 \cdot w$ where $w \sim W$, $\mathrm{E}(W) = 1$.

$$\mathrm{E}(f[i]) = f_0[i], \qquad \mathrm{Var}(f[i]) = f_0[i]^2 \sigma^2 \quad \text{where} \quad \sigma^2 = \mathrm{Var}(W).$$

Example: SAR images.

Each image: $f^{(s)} = f_0 \cdot w^{(s)} + n^{(s)}$

$n^{(s)}$: white additive noise

$w^{(s)}$: one-sided exponential

$p(w^{(s)}[i] = x) \propto e^{-x}\, \mathbb{I}_{x>0}$

$$\text{Averaging:} \quad f = \frac{1}{S} \sum_{s=1}^{S} f^{(s)} \approx f_0 \cdot w$$

$$w \sim \Gamma(\sigma = K^{-\frac{1}{2}}, \mu = 1) \qquad p(w = x) \propto x^{K-1} e^{-Kx}$$

$\sigma = 0.1$

$\sigma = 0.2$

$\sigma = 0.3$

$\sigma = 0.6$

Multiplicative Noise Stabilization

Γ-multiplicative noise model: $\quad f = f_0 \cdot w, \quad p(w = x) \propto x^{K-1} e^{-Kx}$

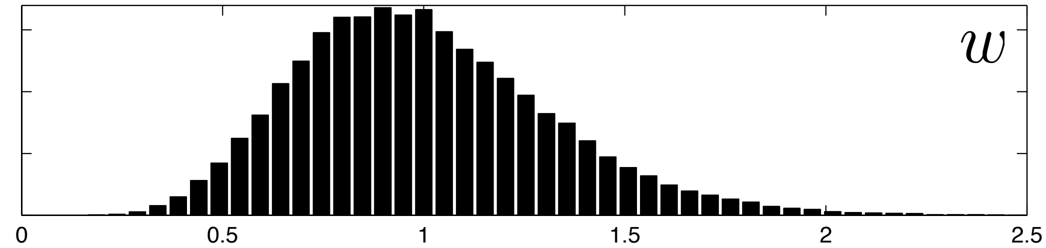
Variance stabilization:

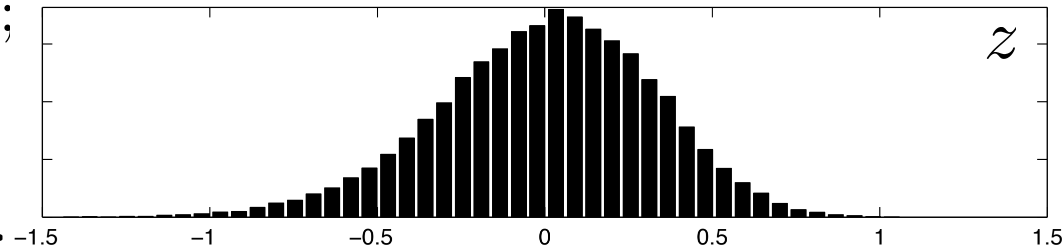
$$\varphi(x) = \log(x) - c$$

where $c = \mathrm{E}(\log(w)) = \psi(K) - \log(K)$;

$$\psi(x) = \Gamma'(x)/\Gamma(x)$$

$$f = f_0 \cdot w \quad \Longrightarrow \quad \varphi(f) = \varphi(f_0) + z.$$





Multiplicative Noise Stabilization

Γ-multiplicative noise model: $\quad f = f_0 \cdot w, \quad p(w = x) \propto x^{K-1} e^{-Kx}$

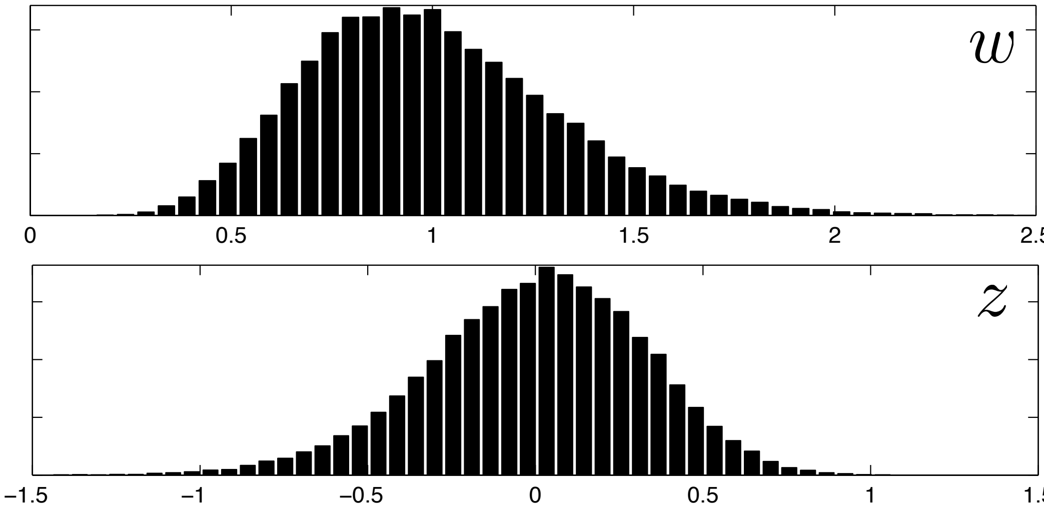
Variance stabilization:

$$\varphi(x) = \log(x) - c$$

where $c = \mathrm{E}(\log(w)) = \psi(K) - \log(K)$;

$$\psi(x) = \Gamma'(x)/\Gamma(x)$$

$$f = f_0 \cdot w \quad \Longrightarrow \quad \varphi(f) = \varphi(f_0) + z.$$



Un-stabilized
SNR=18.4dB

Stabilized
SNR=19.6dB

Conclusion

Noise modeling:

→ additive, multiplicative, etc.

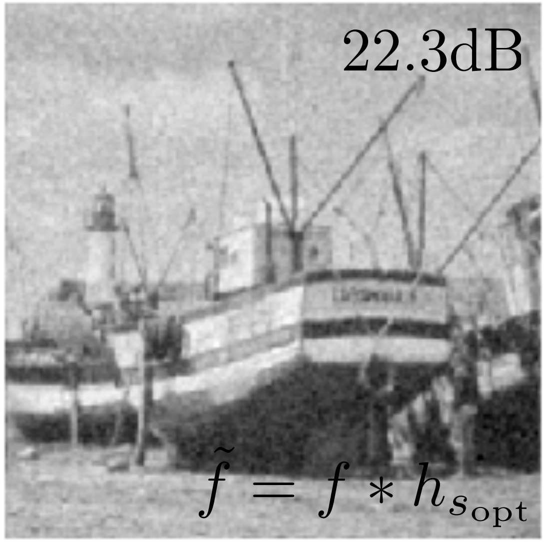
Conclusion

Noise modeling:

→ additive, multiplicative, etc.

Linear denoising:

→ low-pass filtering (blurring).



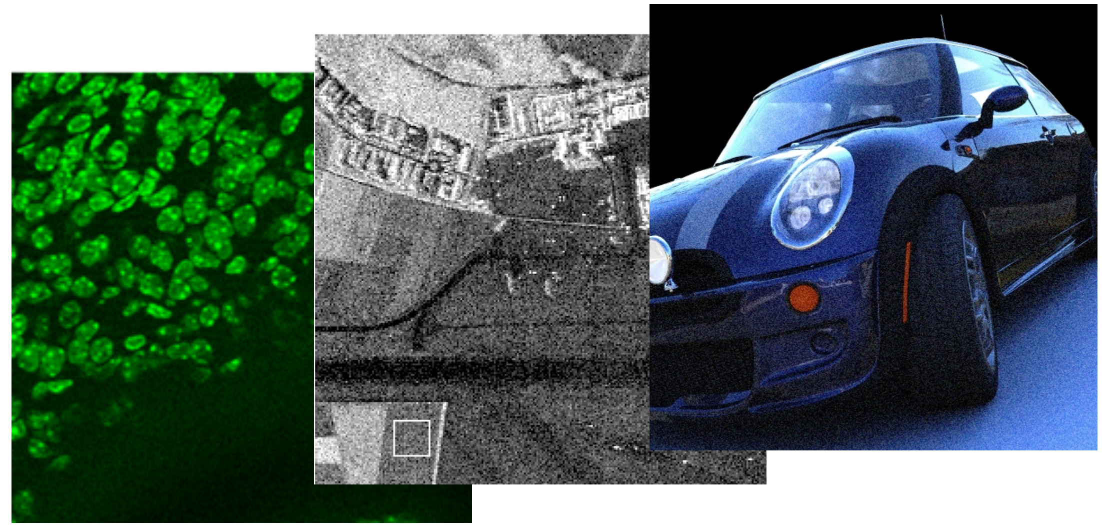
Conclusion

Noise modeling:

$\rightarrow$ additive, multiplicative, etc.

Linear denoising:

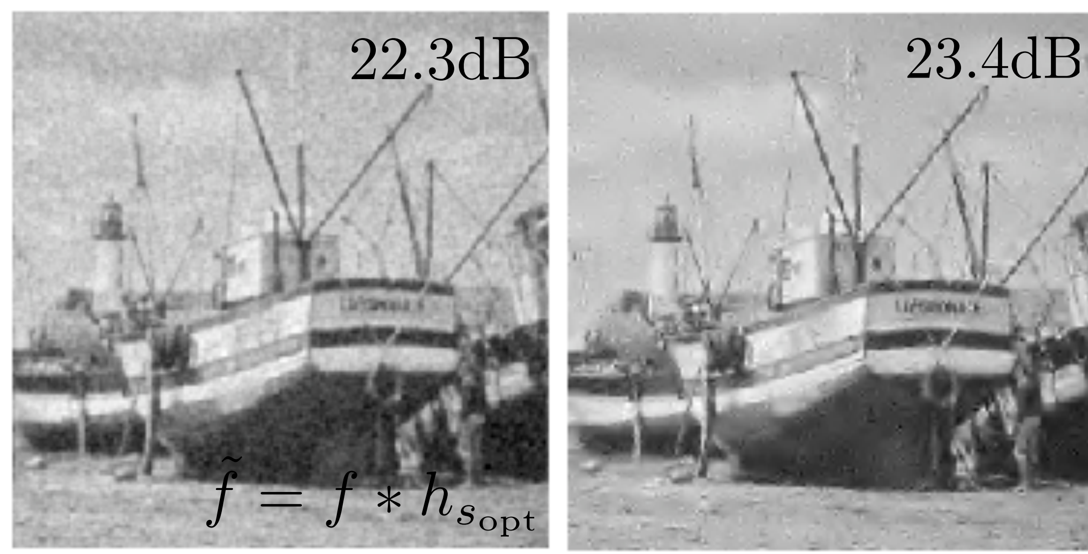
$\rightarrow$ low-pass filtering (blurring).



Non-linear denoising:

$\rightarrow$ thresholding small coefficients.

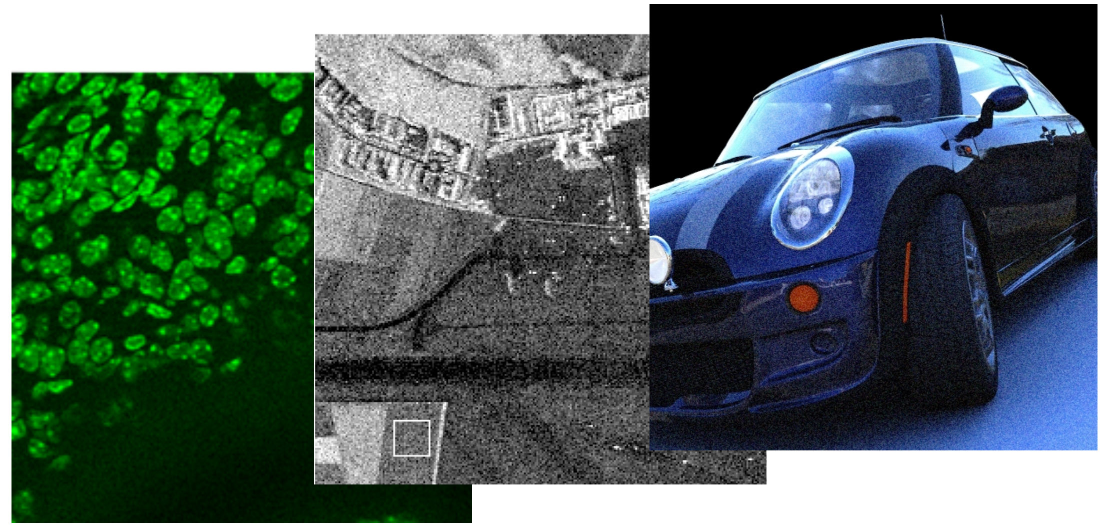
$\rightarrow$ translation invartiance.



Conclusion

Noise modeling:

$\rightarrow$ additive, multiplicative, etc.



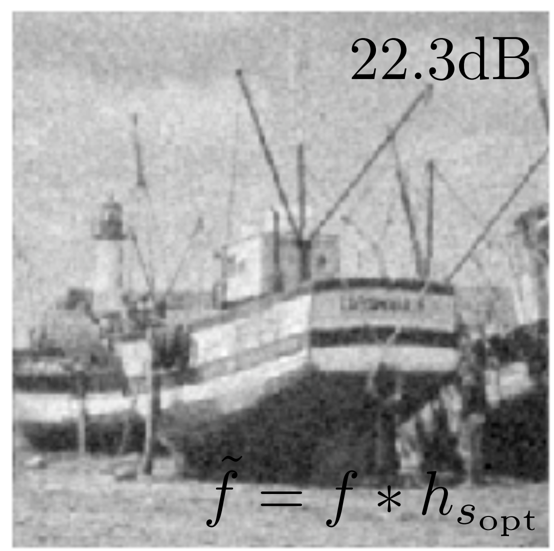
Linear denoising:

$\rightarrow$ low-pass filtering (blurring).

Non-linear denoising:

$\rightarrow$ thresholding small coefficients.

$\rightarrow$ translation invariance.



Exploiting wavelet structure:

$\rightarrow$ block-thresholding.