Technical University of Denmark

Computational Imaging and Spectroscopy: Sparse Recovery

Thierry SOREZE
DTU July 2024

DTU Fotonik
Department of Photonics Engineering

Denoising and sparsity

Problem formulation

We consider the signal X consisting of N equally spaced samples, sampled on a regular grid of dimension d, from its noisy measurements Y:

$$Y[k] = X[k] \odot \varepsilon[k]$$

Then denoising can be approached by the following operation, given that X is sparse in a specific dictionary:

$$\tilde{X} = \mathbf{R}\mathcal{D}(\mathbf{T}Y)$$

With $\mathcal{D}$ a nonlinear estimation rule for the coefficients, $\mathbf{T}$ a transform by which the signal is known to be sparse

Denoising and sparsity

Term by term nonlinear denoising

Let $\boldsymbol{\Phi}$ be a dictionary whose columns are $\left(\varphi_{j,\ell,\mathbf{k}}\right)_{j,\ell,\mathbf{k}}$ a collection of atoms. j and $\mathbf{k} = (k_1, \ldots, k_d)$ the parameters for scale and position, ℓ is an integer indexing the orientations.

We set $\alpha_{j,\ell,\mathbf{k}} = \langle X, \varphi_{j,\ell,\mathbf{k}} \rangle$ as the unknown frame coefficients of the true data, and $\beta_{j,\ell,\mathbf{k}} = \langle Y, \varphi_{j,\ell,\mathbf{k}} \rangle$ the observed coefficients, and $\eta_{j,\ell,\mathbf{k}}$ is the noise sequence in the transform domain.

Denoising and sparsity

Term by term nonlinear denoising

The coefficients $\beta_{j,\ell,\mathbf{k}}$ are thresholded with threshold $\tau_{\sigma_{j,\ell}}$ by assuming the following decision rule:

$$
\begin{array}{llll}
\text{if} & |\beta_{j,\ell,\mathbf{k}}| \geq \tau_{\sigma_{j,\ell}} & \text{then } \beta_{j,\ell,\mathbf{k}} & \text{is significant} \\
\text{if} & |\beta_{j,\ell,\mathbf{k}}| \leq \tau_{\sigma_{j,\ell}} & \text{then } \beta_{j,\ell,\mathbf{k}} & \text{is not significant}
\end{array}
$$

In most applications $\tau = 3.$ The noise level can be estimated in many case by $\tilde{\sigma} = \mathbf{MAD}(w_1)/0.6745.$ $\sigma_{j,\ell}$ is estimated by $\sigma_{j,\ell} = \tilde{\sigma}\|\varphi_{j,\ell,\mathbf{k}}\|$

Denoising and sparsity

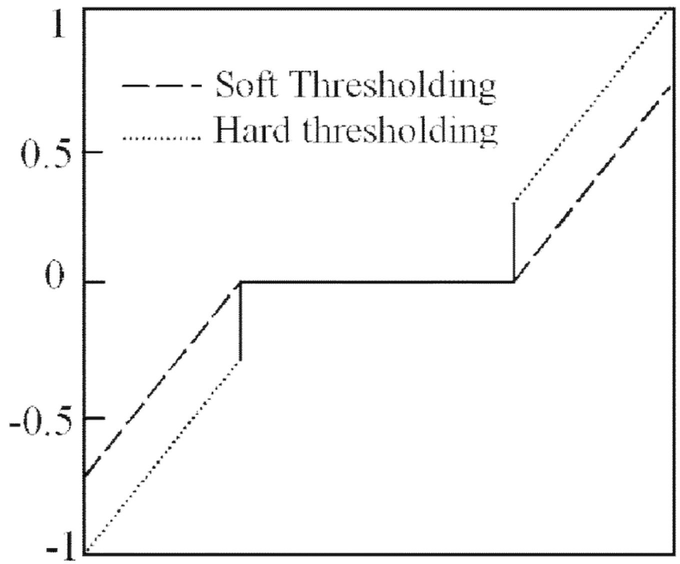
Hard and Soft thresholding

$$\tilde{\beta}_{j,\ell,\mathbf{k}} = \mathrm{HardThresh}_{t_{j,\ell}}(\beta_{j,\ell,\mathbf{k}}) = \begin{cases} \beta_{j,\ell,\mathbf{k}} & \text{if} \quad |\beta_{j,\ell,\mathbf{k}}| \geq t_{j,\ell} \\ 0 & \text{otherwise} \end{cases}$$

$$\tilde{\beta}_{j,\ell,\mathbf{k}} = \mathrm{SoftThresh}_{t_{j,\ell}}(\beta_{j,\ell,\mathbf{k}}) = \begin{cases} sign(\beta_{j,\ell,\mathbf{k}})(|\beta_{j,\ell,\mathbf{k}}| - t_{j,\ell}) & \text{if} \quad |\beta_{j,\ell,\mathbf{k}}| \geq t_{j,\ell} \\ 0 & \text{otherwise} \end{cases}$$

$$t = \sigma\sqrt{2\log N}$$

$$t_{j,\ell} = \sigma\sqrt{2\log N_{j,\ell}}$$



Denoising and sparsity

Multiplicative noise

In various imaging system the data representing the unknown signal X is corrupted with multiplicative noise, which can be modeled as:

$$Y[k] = X[k]\varepsilon[k]$$

The multiplicative noise follows a Gamma distribution with parameter K:

$$\mathrm{pdf}(\varepsilon[k]) = \frac{K^K \varepsilon[k]^{K-1} e^{(-K)\varepsilon[k]}}{(K-1)!}$$

We can expressed the noisy signal as corrupted by additive noise:

$$Y_s = \log Y[k] = \log X[k] + \log \varepsilon[k] = \log X[k] + \log \epsilon[k]$$

Denoising and sparsity

Multiplicative noise

The pdf of $\epsilon[k]$ is given by

$$\mathrm{pdf}(\epsilon[k]) = \frac{K^K \epsilon[k]^{K-1} e^{\left(K\epsilon[k] - e^{\epsilon[k]}\right)}}{(K-1)!}$$

The mean and the variance of $\epsilon[k]$ are given by

$$\psi_0(K) - \log K, \qquad \psi_1(K)$$

With $\psi_0(Z)$ the polygamma function.

Denoising and sparsity

Poisson noise

In many acquisition devices, i.e. photodetectors in cameras, CT, etc.) the noise comes from fluctuations of a counting process, which be modeled by a Poisson noise model. Each observation $Y[k]$ follows a Poisson distribution, $\mathcal{P}(X[k]), \forall k. X[k]$ is the underlying signal. The mean and variance of $\mathcal{P}(X[k])$ are equivalent and equal to $X[k]$.

Anscombe variance stabilization transform (VST) $\mathcal{A}$:

$$Y_s = \mathcal{A}(Y[k]) = 2\sqrt{Y[k] + \frac{3}{8}}$$

Then if $X[k]$ is large we can consider that $Y_s[k] \sim \mathcal{N}\left(2\sqrt{X[k]}, 1\right)$

Denoising and sparsity

Mixed Gaussian and Poisson noise

In the case of CCD for instance, the read noise is modeled by a Gaussian model, so $Y[k] = \varepsilon[k] + g_0\xi[k]$, with $\varepsilon[k] \sim \mathcal{N}(\mu, \sigma^2)$ and $\xi[k] \sim \mathcal{P}(X[k])$, g_0 is the CCD detector gain.

$$Y_s = \mathcal{A}(Y[k]) = 2\sqrt{g_0 Y[k] + \frac{3}{8}{g_0}^2 + \sigma^2 - g_0\mu}$$

For $X[k] \geq 20$ $Y_s \sim \mathcal{N}\left(2\sqrt{X[k]/g_0}, 1\right)$

Sparse Denoising

Applications: denoising

One of the main interest of redundant transforms is image restoration, and in particular denoising

$$\tilde{w}_j[k,l] = \mathrm{HardThresh}_{t_j}\left(w_j[k,l]\right) = \begin{cases} w_j[k,l] & if\ \left|w_j[k,l]\right| \geq t_j \\ 0 & \text{otherwise} \end{cases}$$

$w_j[k,l]$ is the wavelet coefficient at scale j and at spatial position (k,l)

$t_j = \tau\sigma_j$, where σ_j is the noise standard deviation at scale j, and τ is a constant generally chosen between 3 and 5.

If the analysis filter is normalized to a unit ℓ_2 norm we have $\sigma_j = \sigma$ for all j.

Sparse Denoising

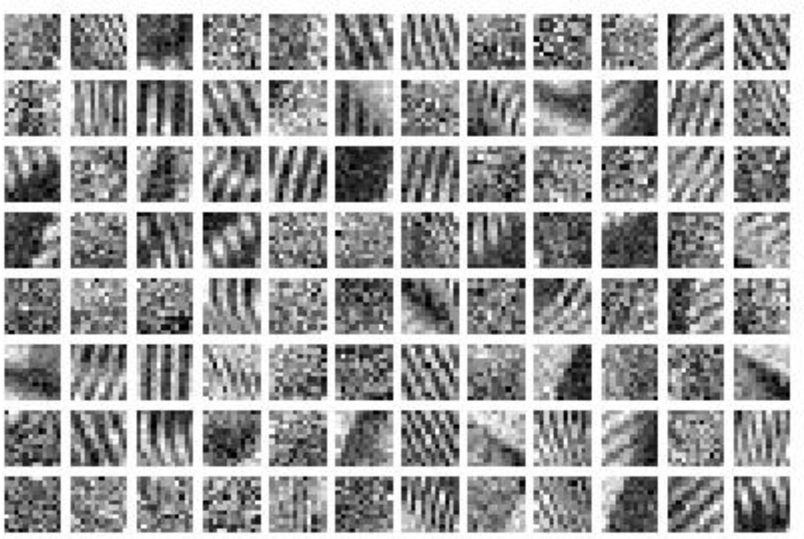
Applications: denoising

If we denote the wavelet transform and reconstruction operators respectively by $\mathbf{T}_W$ and $\mathbf{R}_W$, with the relation $\mathbf{R}_W = \mathbf{T}_W^{-1}$ for an orthogonal transform, the denoising procedure of an image X by thresholding, with a threshold parameter τ can be expressed as follow:

$$\tilde{X} = \mathbf{R}_W \mathrm{HardThresh}_\tau(\mathbf{T}_W X)$$

Therefore in this context a wavelet nonlinear denoising consists in taking the wavelet transform of the image, hard thresholding and reconstructing the image with the remaining coefficients.

Dictionary Learning



Dictionary Learning

Problem formulation

Let $x_i \in \mathbb{R}^N, i = 1, \cdots, P$ be a set of exemplar signals. Denote $\mathbf{X} \in \mathbb{R}^{N\times P}$storing the vectors x_i as its columns. The aim of Dictionary learning (DL) is to solve the following problem: Find $\mathbf{\Phi}$ and $\boldsymbol{\alpha}$ such that $\boldsymbol{X} \approx \mathbf{\Phi}\boldsymbol{\alpha}$

Where $\mathbf{\Phi} \in \mathbb{R}^{N\times T}$ is the dictionary of T atoms, and $\boldsymbol{\alpha} \in \mathbb{R}^{T\times P}$ is the matrix whose i-th column is the synthesis coefficients vectors α_i of the exemplar x_i in $\mathbf{\Phi}$

Our main prior is that $\boldsymbol{\alpha}$ is sparse. We can cast the DL problem as the following optimization problem

$$\min_{\mathbf{\Phi},\boldsymbol{\alpha}} \frac{1}{2} \|\mathbf{X} - \mathbf{\Phi}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \sum_{i=1}^{T} \lambda_i \|\alpha_i\|_p^p \qquad \mathrm{s.t} \qquad \mathbf{\Phi} \in \mathcal{D}$$

Dictionary Learning

Alternating minimization

This problem is non convex, even for p=1 and $\mathcal{D}$ convex. We can adopt an alternating minimization strategy to solve it:

$$\boldsymbol{\alpha}^{(t+1)} \in \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \left\| \mathbf{X} - \boldsymbol{\Phi}^{(t)} \boldsymbol{\alpha} \right\|_F^2 + \lambda_i \|\alpha_i\|_p^p$$

$$\boldsymbol{\Phi}^{(t+1)} \in \arg\min_{\boldsymbol{\Phi} \in \mathcal{D}} \frac{1}{2} \left\| \mathbf{X} - \boldsymbol{\Phi} \boldsymbol{\alpha}^{(t+1)} \right\|$$

Dictionary Learning

Alternating minimization

DL via Alternating minimization

Input: exemplars or patches **X**, regularization parameter or target sparsity level

Initialization: initial dictionary $\boldsymbol{\Phi}^{(0)}$

Main iteration:

For t=0 to Niter-1 **do**

  Sparse coding: update $\boldsymbol{\alpha}^{(t+1)}$ for fixed $\boldsymbol{\Phi}^{(t)}$

  Dictionary update: Update $\boldsymbol{\Phi}^{(t+1)}$ for fixed $\boldsymbol{\alpha}^{(t+1)}$

Dictionary Learning

Alternating minimization

Selected algorithms for Dictionary update:

- ❑Projected Gradient Descent
- ❑Method of Optimal Directions (MOD)
- ❑K-SVD

Dictionary Learning

Dictionary Learning and Linear Inverse Problem

Problem formulation

$$y = \mathbf{H}x_0 + \varepsilon$$

We can tackle this problem by learning the dictionary and solving it at the same time. This is done by formulating the problem as:

$$\min_{x,\mathbf{\Phi}\in\mathcal{D},(\alpha)_{1\leq k\leq p}} \frac{1}{2}\|y - \mathbf{H}x\|^2 + \frac{\mu}{P}\left(\sum_{k=1}^{P}\frac{1}{2}\|R_k(x) - \mathbf{\Phi}\alpha\|^2 + \lambda\|\alpha_k\|_1\right)$$

Dictionary Learning

Dictionary Learning and Linear Inverse Problem

Problem solution via Alternating minimization

Updating x for a fixed dictionary $\mathbf{\Phi}^{(t+1)}$ and sparse code $\alpha^{(t+1)}$ leads to the following problem

$$\min_{x\in\mathbb{R}^N} \frac{1}{2}\|y-\mathbf{H}x\|^2+\frac{\mu}{P}\sum_{k=1}^{P}\frac{1}{2}\left\|R_k(x)-\mathbf{\Phi}^{(t+1)}\alpha_k^{(t+1)}\right\|^2$$

Whose minimizer has the closed form

$$x^{(t+1)}=\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}+\mu\mathbf{I}\right)^{-1}\left(\mathbf{H}^{\mathrm{T}}y+\frac{\mu}{P}\sum_{k=1}^{P}R_k^*\left(\mathbf{\Phi}^{(t+1)}\alpha_k^{(t+1)}\right)\right)$$

Dictionary Learning

Alternating minimization

DL via Alternating minimization

Input: observation y, operator $\mathbf{H}$, parameters μ and λ
Initialization: $x^{(t+1)} = 0$, initial dictionary $\mathbf{\Phi}^{(0)}$
Main iteration:

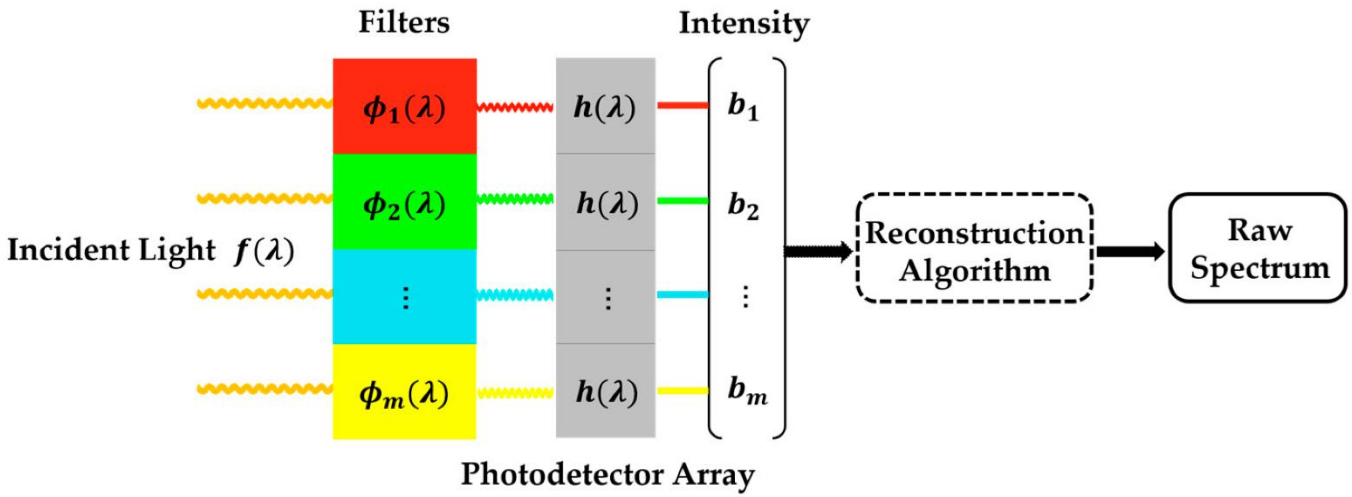
For t=0 to Niter-1 **do**

Update $x^{(t+1)}$

Sparse coding: update $\boldsymbol{\alpha}^{(t+1)}$ for fixed $\mathbf{\Phi}^{(t)}$

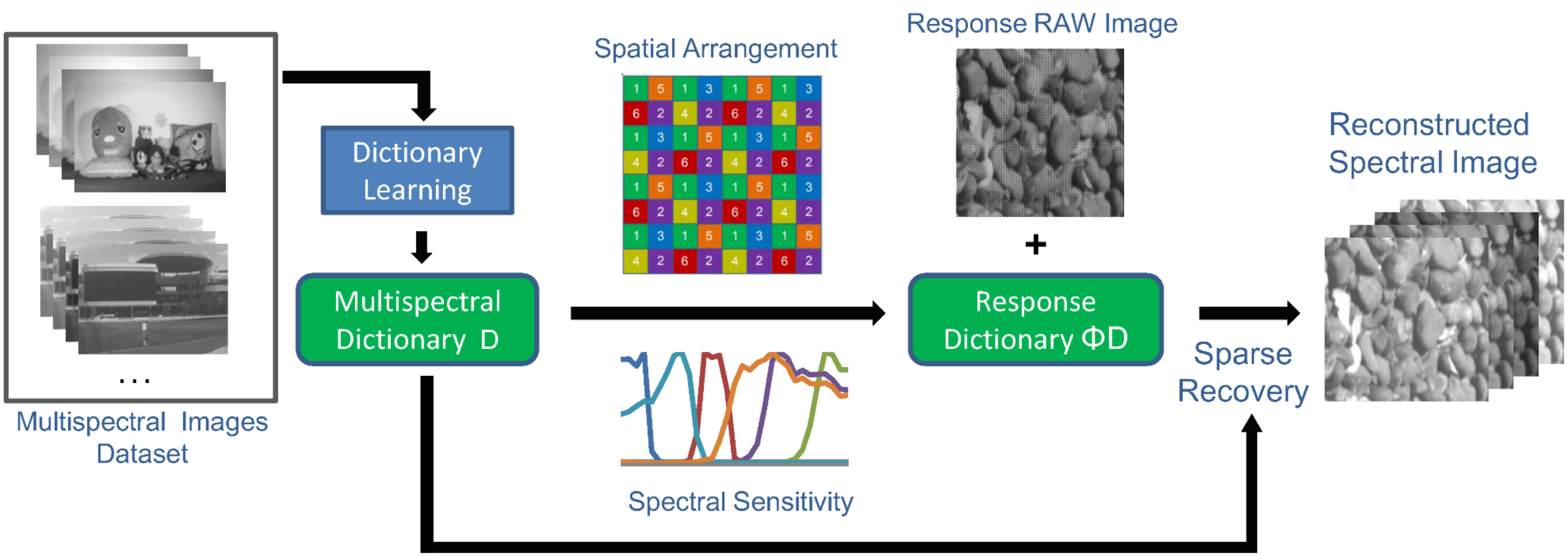
Dictionary update: Update $\mathbf{\Phi}^{(t+1)}$ for fixed $\boldsymbol{\alpha}^{(t+1)}$

Dictionary Learning



Zhang S, Dong Y, Fu H, Huang S-L, Zhang L. A Spectral Reconstruction Algorithm of Miniature Spectrometer Based on Sparse Optimization and Dictionary Learning. *Sensors*. 2018; 18(2):644.

Dictionary Learning

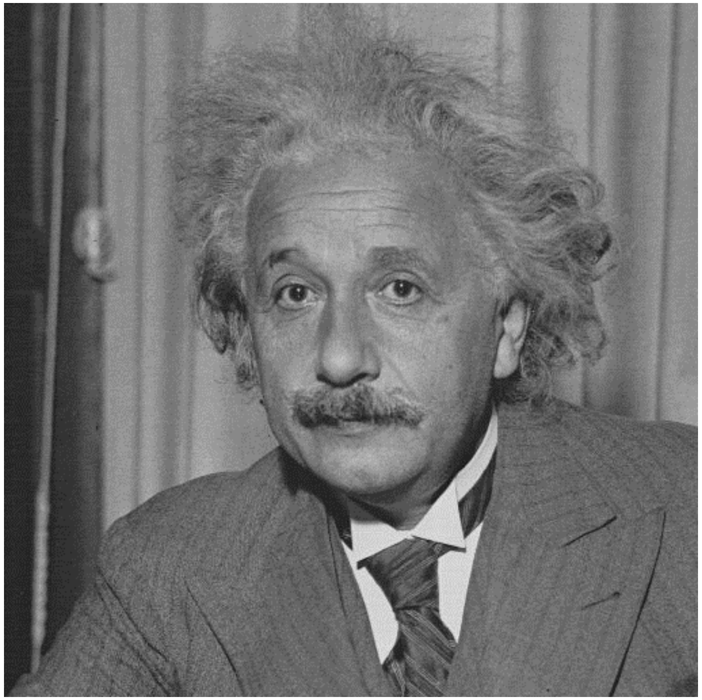


Wu R, Li Y, Xie X, Lin Z. Optimized Multi-Spectral Filter Arrays for Spectral Reconstruction. *Sensors*. 2019; 19(13):2905.

Sparse Denoising

Applications

Denoising by SWT



Load the Einstein image

1. Apply a Gaussian noise to it
2. Apply a 2D-DWT, Threshold and reconstruct
3. Apply a 2D-UWT, Threshold and reconstruct
4. Compare the perfomance of DWT and UWT

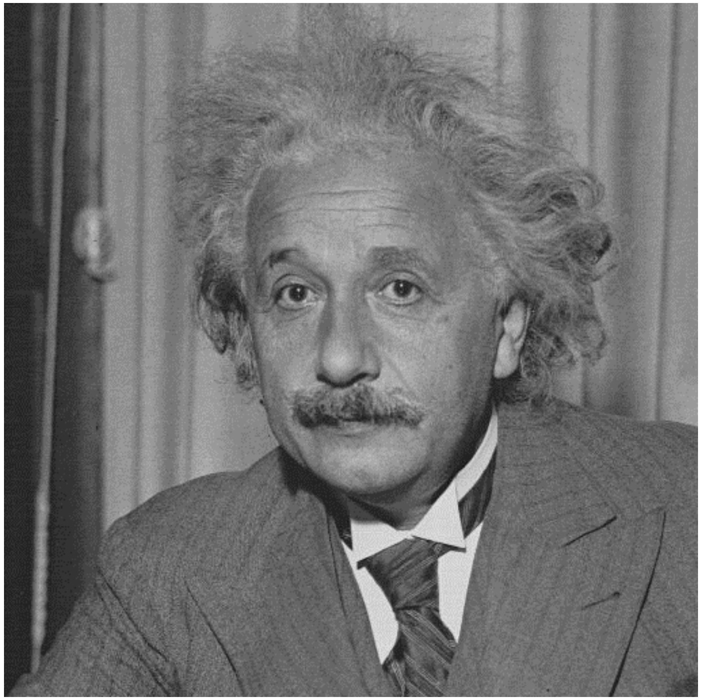
Wavelet Symmlet 4
Level 2

$$\mathbf{PSNR} = 20\log_{10}\frac{N\left|\max_{k,l}X[k,l]-\min_{k,l}X[k,l]\right|}{\sqrt{\sum_{k,l}\left(\tilde{X}[k,l]-X[k,l]\right)^2}}\ \text{dB}$$

Sparse Denoising

Applications

Denoising by SWT



The median absolute deviation (MAD)

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.median_abs_deviation.html

Sparse Denoising

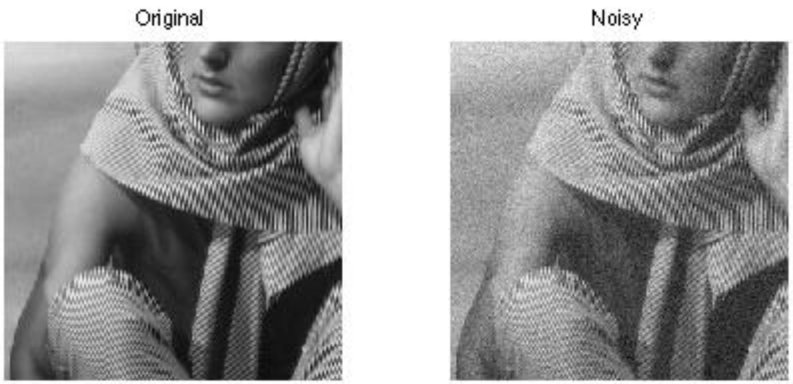
Application: Dictionary Learning

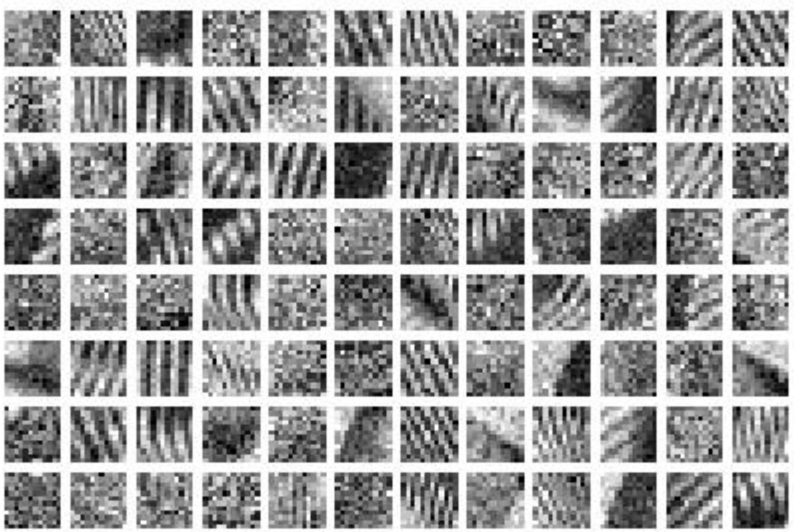
Patch extraction

- ❑ Load Barbara and apply a Gaussian noise
- ❑ Extract reference patches
- ❑ Display few random patches

Sparse coding

- ❑ Learn the dictionary from extracted patches
- ❑ Obtain the denoised image





Sparse Denoising

Application: Dictionary Learning

1. Load the noisy image.
2. Extract small patches from the image.
3. Reshape the patches for dictionary learning.
4. Perform dictionary learning to learn a set of basis functions (atoms).
5. Denoise the patches using sparse coding and the learned dictionary.
6. Reconstruct the denoised image from the denoised patches.

Step by step examples:

https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto_examples/decomposition/plot_image_denoising.html

https://scikit-learn.org/stable/auto_examples/decomposition/plot_image_denoising.html

Sparse Denoising

Application 3

Denoising with Shearlets (OPTIONAL!)

Load the Barbara image

Apply a Gaussian noise to it

Apply a 2D-UWT, Threshold and reconstruct

Apply a 2D shearlet transform, Threshold and reconstruct

Compare the perfomance of Shearlet and UWT