Thresholding

DS-GA 1013 / MATH-GA 2824 Mathematical Tools for Data Science

Carlos Fernandez-Granda

Prerequisites

Short-time Fourier transform

Wavelets

Denoising

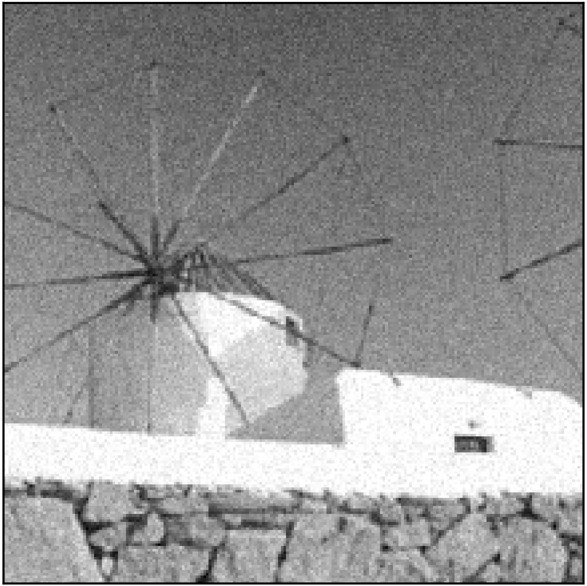
In real applications measurements are usually corrupted by noise

Simple, yet useful, model:

$$\text{data} = \text{signal} + \text{noise}$$

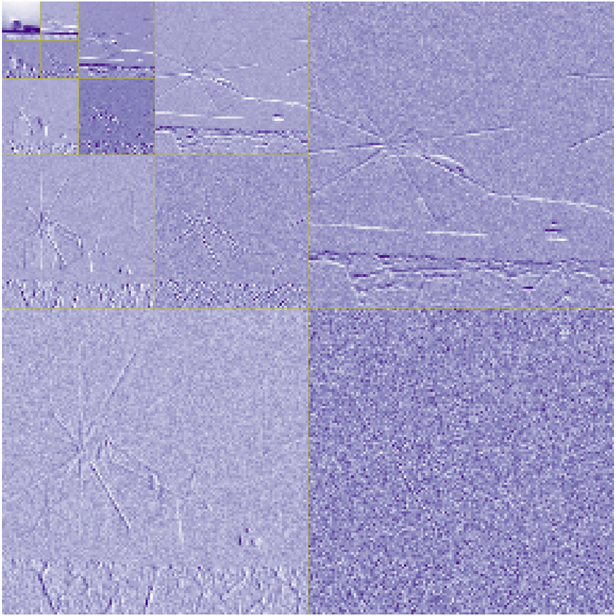
Denoising is the problem of estimating the signal from the noisy data

Noisy image

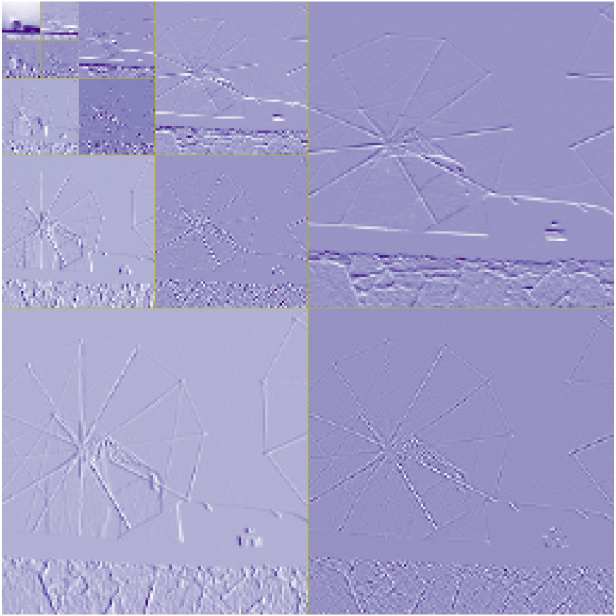


Image

Wavelet coefficients of noisy image



Wavelet coefficients of clean image



What do you notice?

Wavelet coefficients of signal are sparse

Wavelet coefficients of noise are dense

Why are wavelet coefficients of noisy image dense?

If $\tilde{z}$ is Gaussian with mean μ and covariance matrix Σ, then for any A, $A\tilde{z}$ is Gaussian with mean $A\mu$ and covariance matrix $A\Sigma A^T$

For wavelet transform, A is orthogonal so if $\Sigma = I$, then

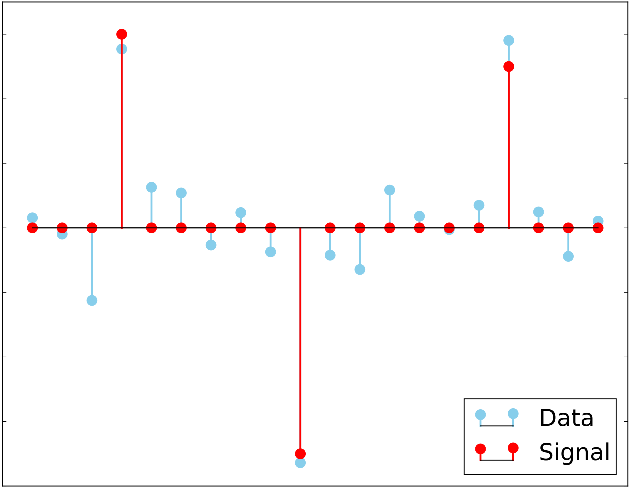
$$\Sigma_{A\tilde{z}} = AA^T = I$$

How can we exploit this to denoise?

Wavelet coefficients of signal are sparse

Wavelet coefficients of noise are dense

How would you denoise this signal?

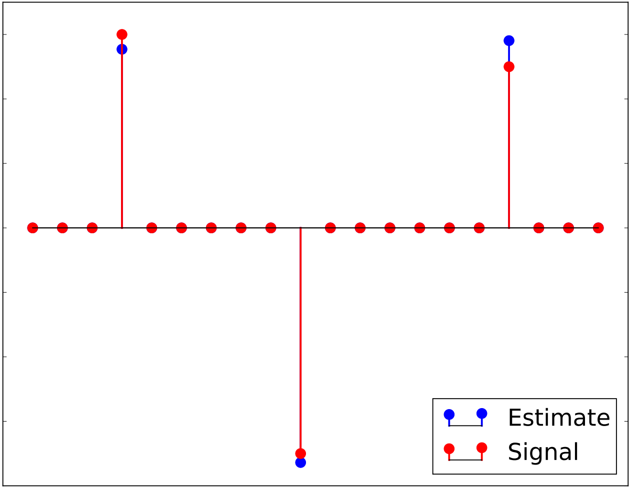


Thresholding

Hard-thresholding operator

$$\mathcal{H}_{\eta}(v)[j] := \begin{cases} v[j] & \text{if } |v[j]| > \eta \\ 0 & \text{otherwise} \end{cases}$$

Denoising via hard thresholding



Denoising via hard thresholding

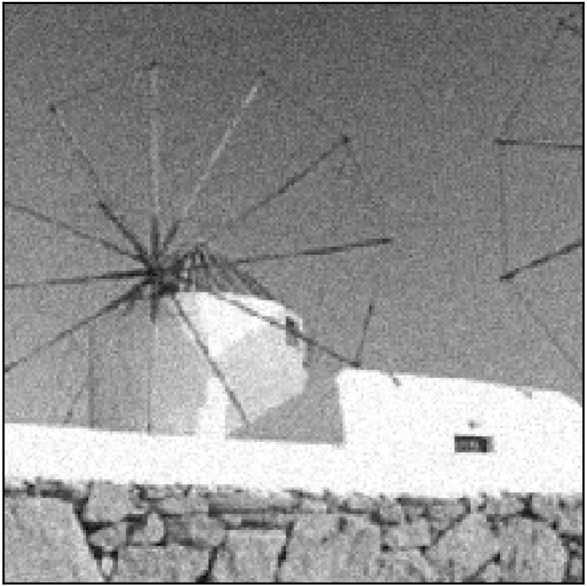
Given data y and a sparsifying linear transform A

1. Compute coefficients Ay
2. Apply the hard-thresholding operator $\mathcal{H}_\eta : \mathbb{C}^N \to \mathbb{C}^N$ to Ay
3. Invert the transform

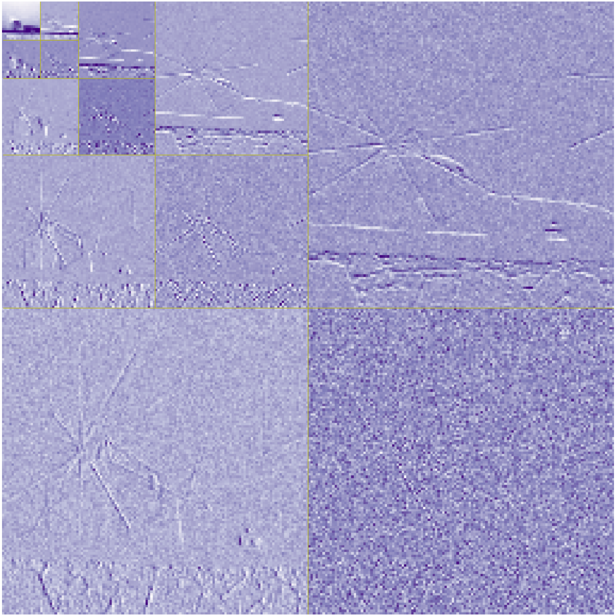
$$x_{\text{est}} := L\,\mathcal{H}_\eta\,(Ay)\,,$$

where L is a left inverse of A

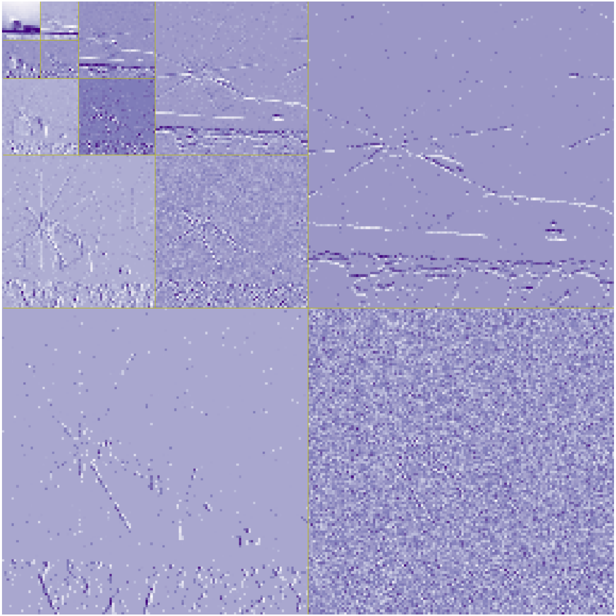
Noisy image



Noisy wavelet coefficients



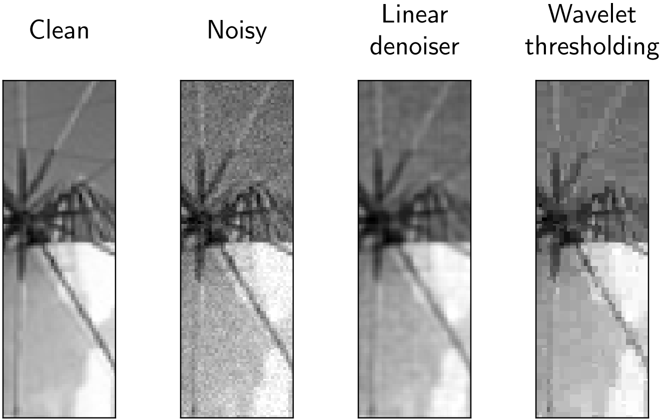
Thresholded wavelet coefficients



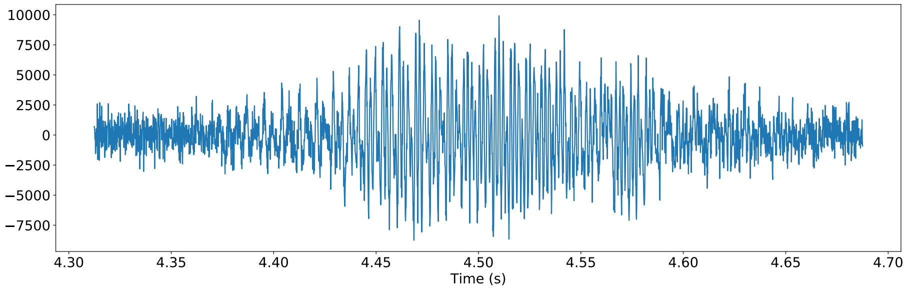
Denoised image

Clean image

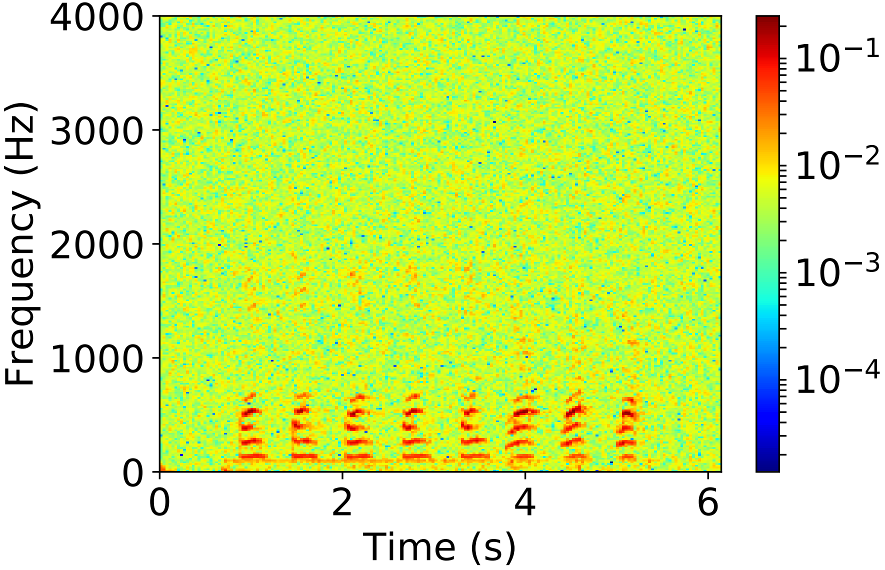
Comparison



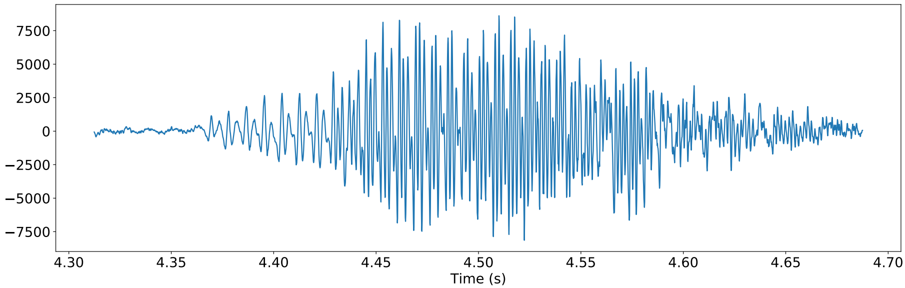
Noisy speech signal



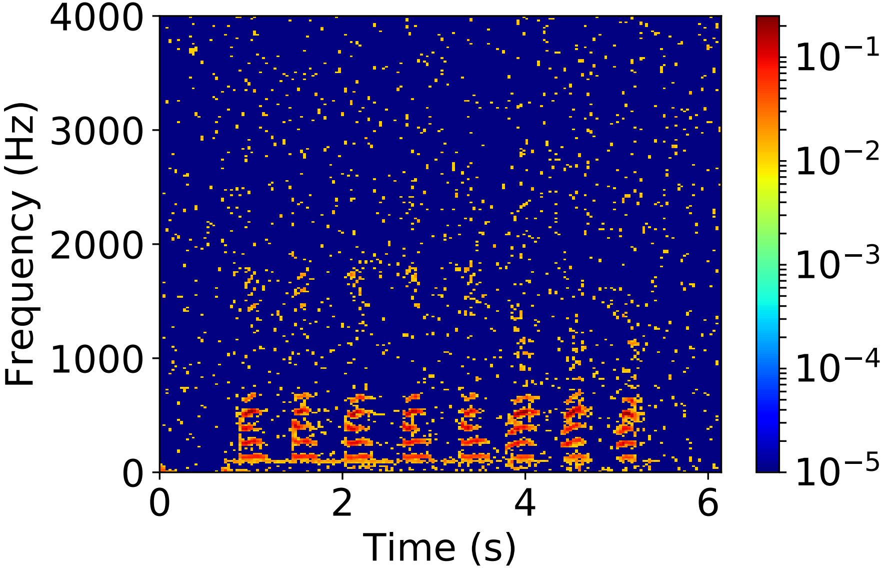
STFT coefficients of noisy data



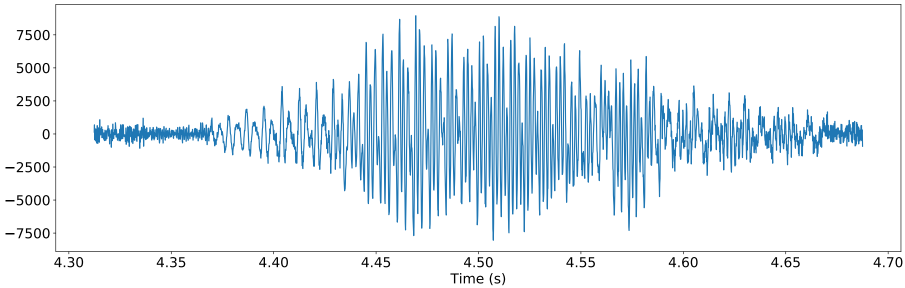
Speech signal



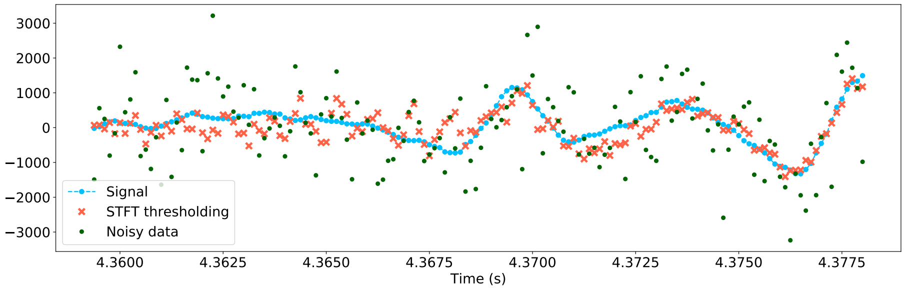
Thresholded STFT coefficients



Denoised signal



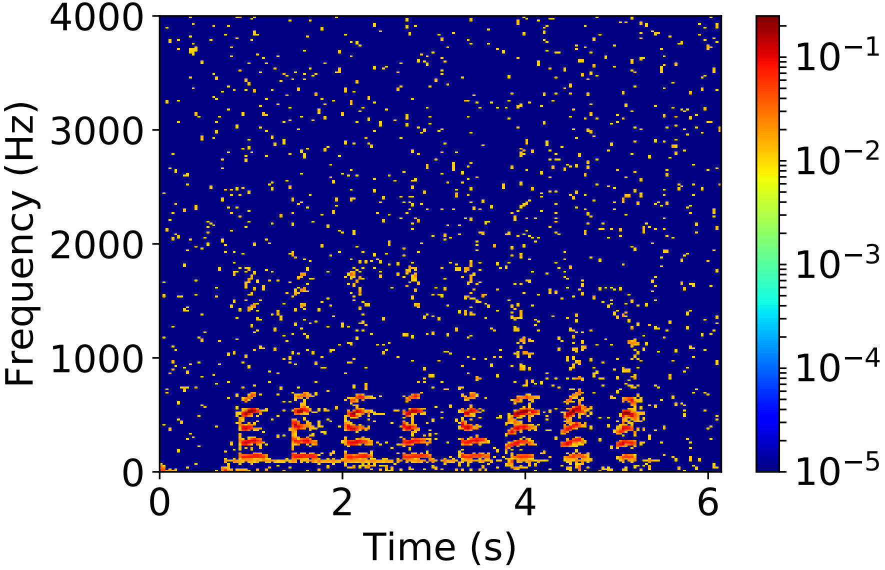
Denoised signal



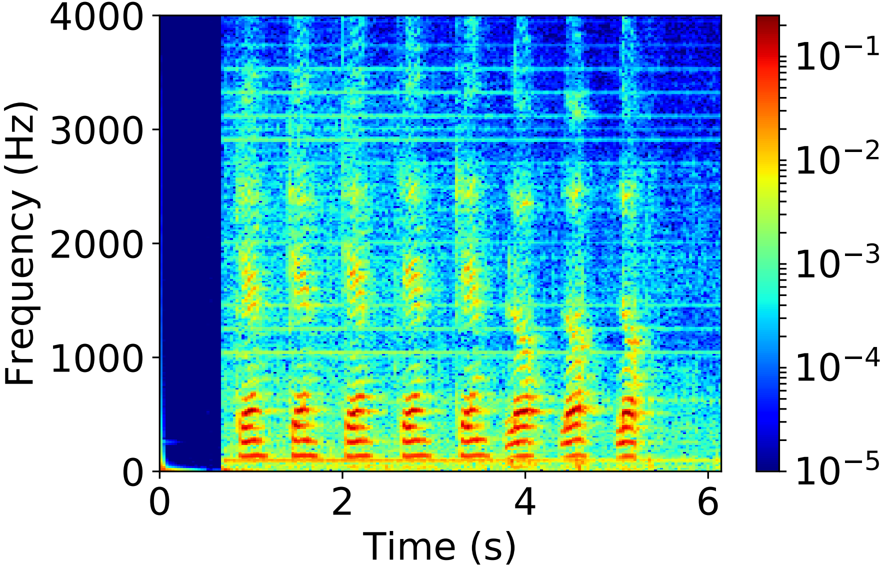
Can we do better?

The denoised signal sounds quite strange...

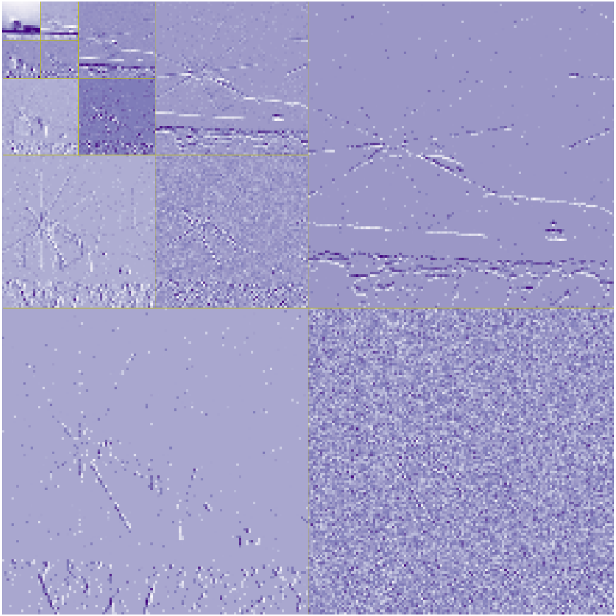
Thresholded STFT coefficients



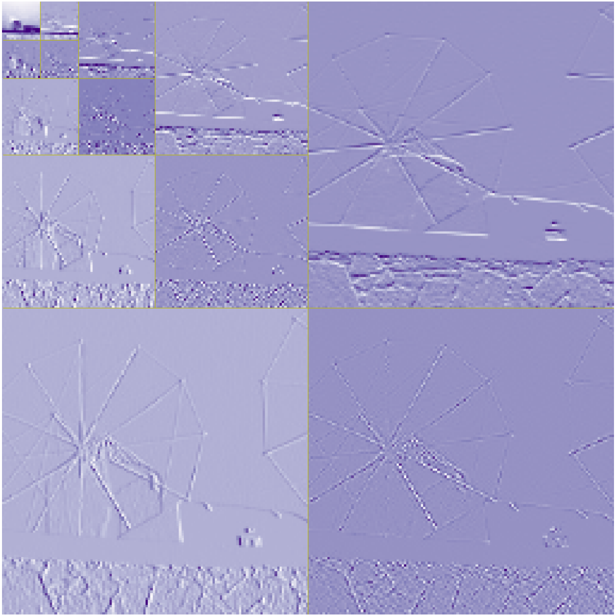
Clean STFT coefficients



Thresholded wavelet coefficients



Clean wavelet coefficients



How can we exploit structure in the coefficients?

In clean signal coefficients are clustered

Idea: Threshold them in blocks

Block thresholding

Threshold according to block of surrounding coefficients $\mathcal{I}_j$

$$\mathcal{B}_\eta(v)[j] := \begin{cases} v[j] & \text{if } j \in \mathcal{I}_j \quad \text{such that } \left|\left|v_{\mathcal{I}_j}\right|\right|_2 > \eta,, \\ 0 & \text{otherwise} \end{cases}$$

Denoising via block thresholding

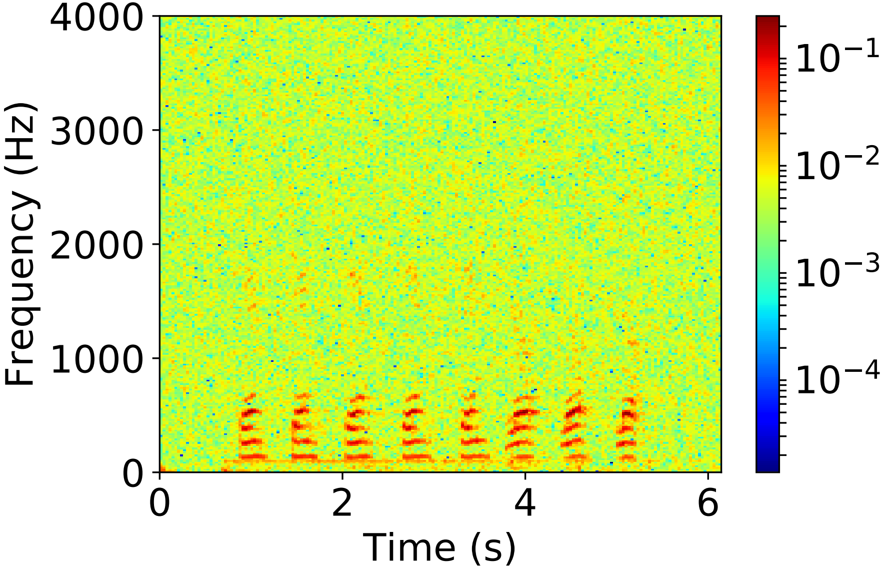
Given data y and a sparsifying linear transform A

1. Compute coefficients Ay
2. Apply the block-thresholding operator $\mathcal{H}_\eta : \mathbb{C}^N \to \mathbb{C}^N$ to Ay
3. Invert the transform

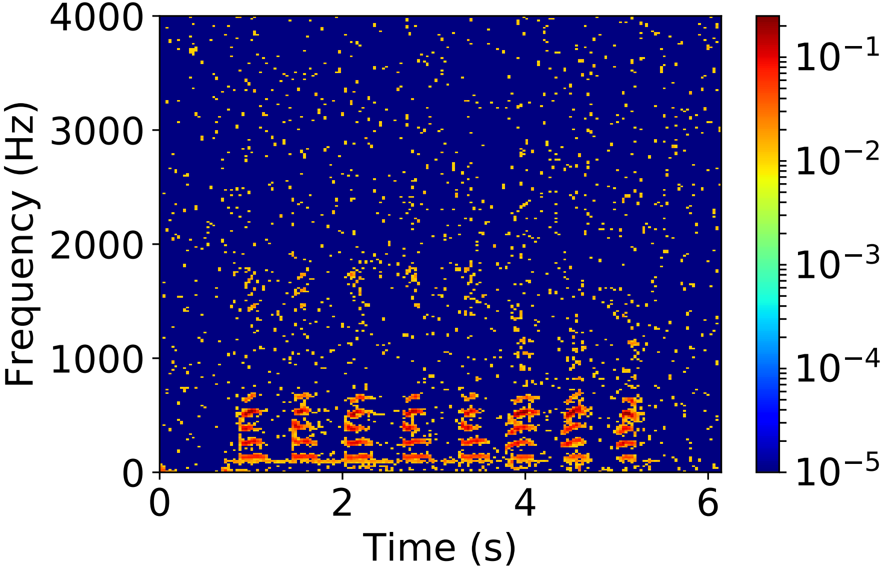
$$x_{\text{est}} := L\,\mathcal{B}_\eta\,(Ay)\,,$$

   where L is a left inverse of A

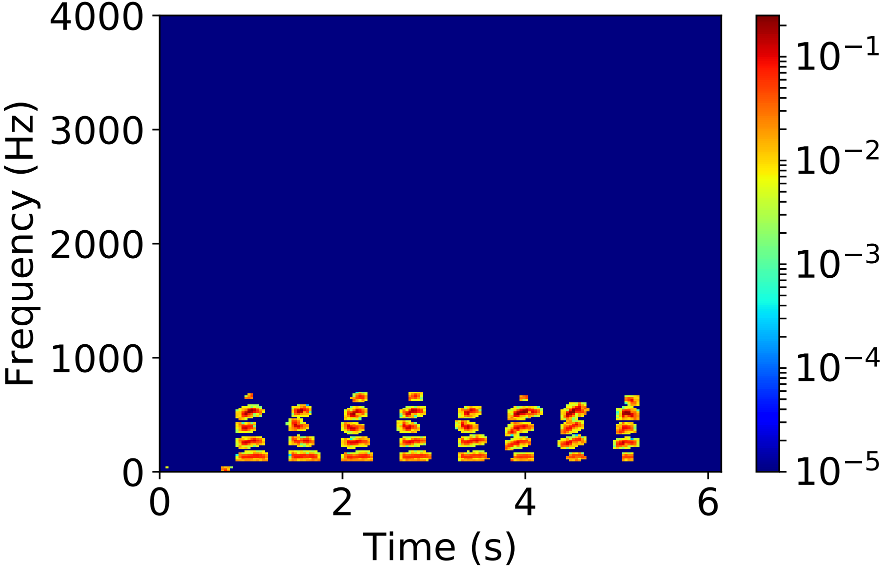
Noisy STFT coefficients



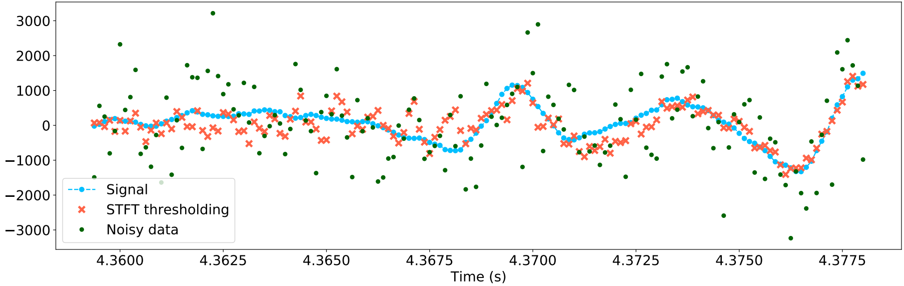
Thresholded STFT coefficients



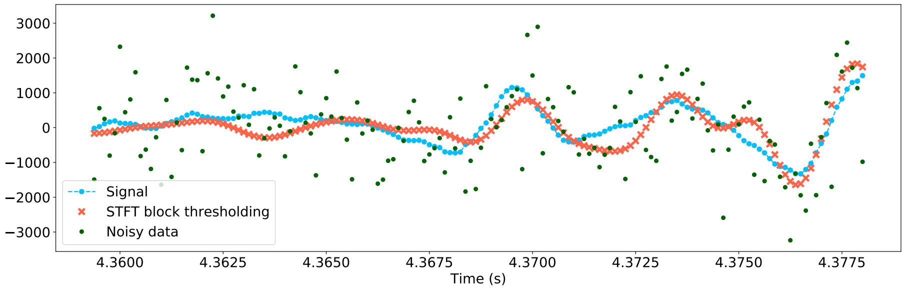
Block-thresholded STFT coefficients (block of length 5)



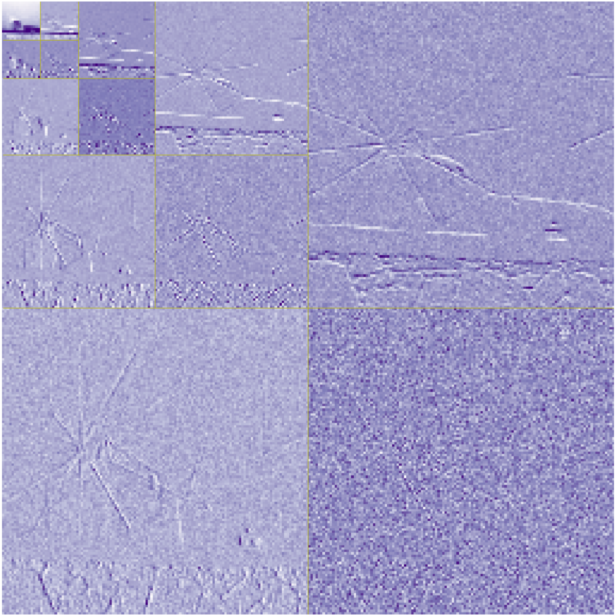
Thresholding



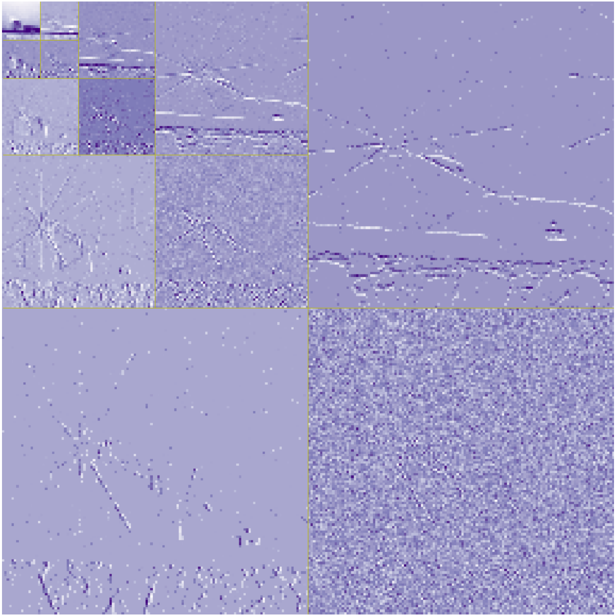
Block thresholding



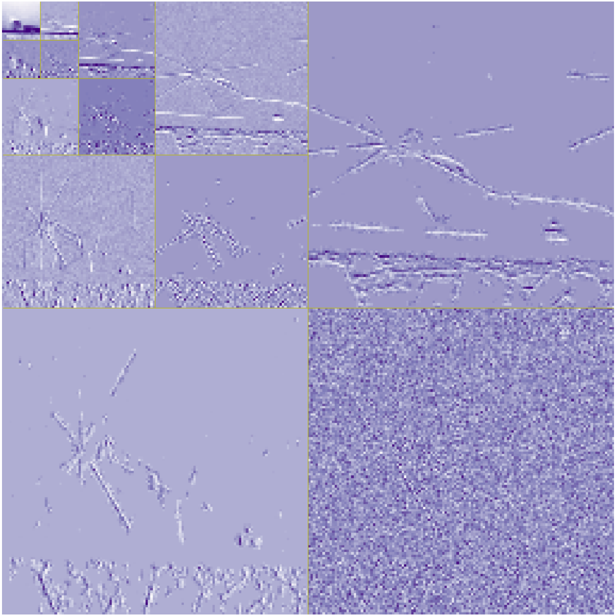
Noisy wavelet coefficients



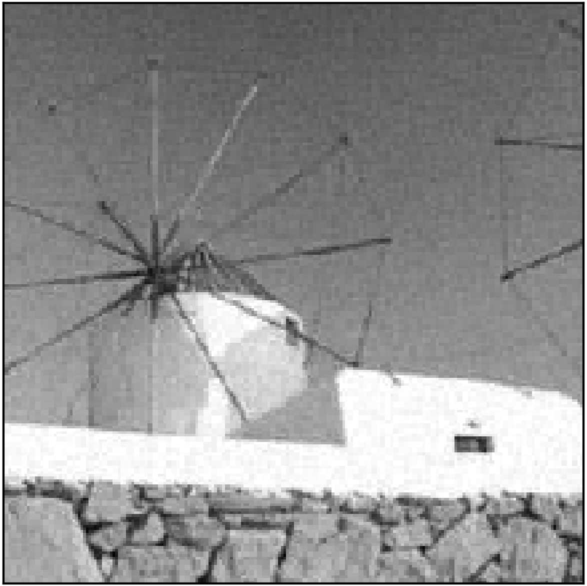
Thresholded wavelet coefficients



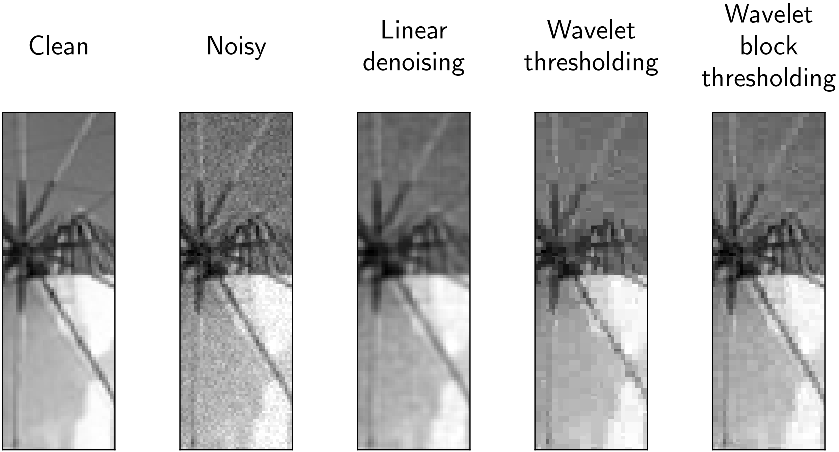
Block thresholded wavelet coefficients



Denoised signal



Comparison


Clean
Noisy
Linear denoising
Wavelet thresholding
Wavelet block thresholding

What have we learned

How to denoise using sparsifying transforms such as wavelets and the STFT

How to adapt thresholding to signal structure