Signal Representations

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

https://cims.nyu.edu/~cfgranda/pages/MTDS_spring19/index.html

Carlos Fernandez-Granda

Motivation

Limitation of frequency representation: no time resolution

Limitation of Wiener filtering: cannot adapt to noisy signal

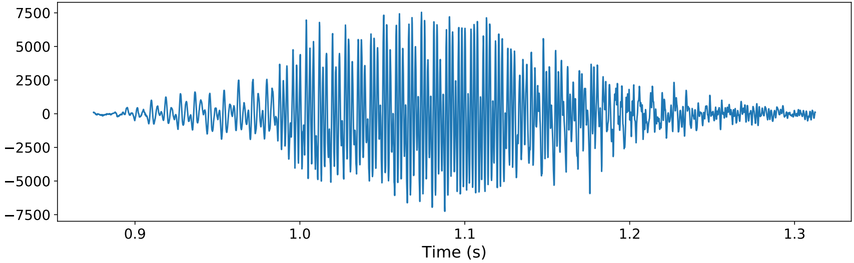
Windowing

Short-time Fourier transform

Multiresolution analysis

Denoising via thresholding

Speech signal



Beyond Fourier

Problem: How to capture local fluctuations

Beyond Fourier

Problem: How to capture local fluctuations

First segment signal, then compute DFT

Beyond Fourier

Problem: How to capture local fluctuations

First segment signal, then compute DFT

Naive segmentation: multiplication by rectangular window

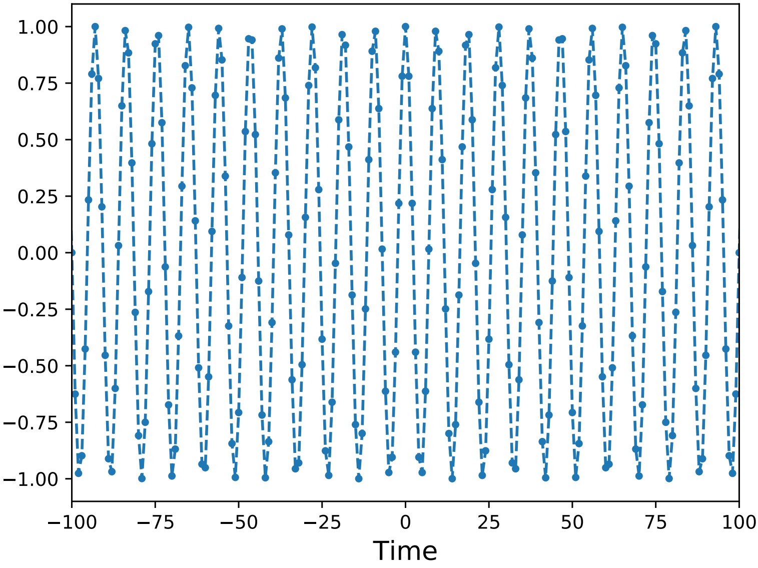
Rectangular window

Rectangular window $\vec{\pi} \in \mathbb{C}^N$ with width $2w$:

$$\vec{\pi}[j] := \begin{cases} 1 & \text{if } |j| \leq w, \\ 0 & \text{otherwise.} \end{cases}$$

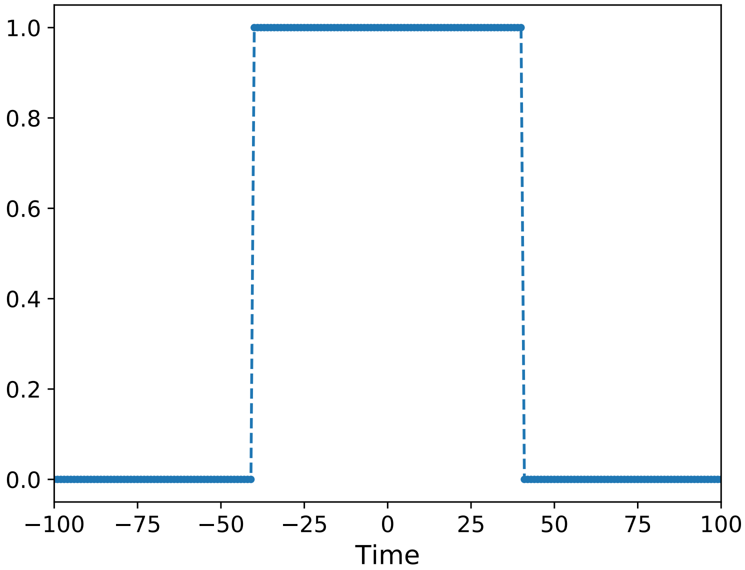
Is this a good choice?

Signal

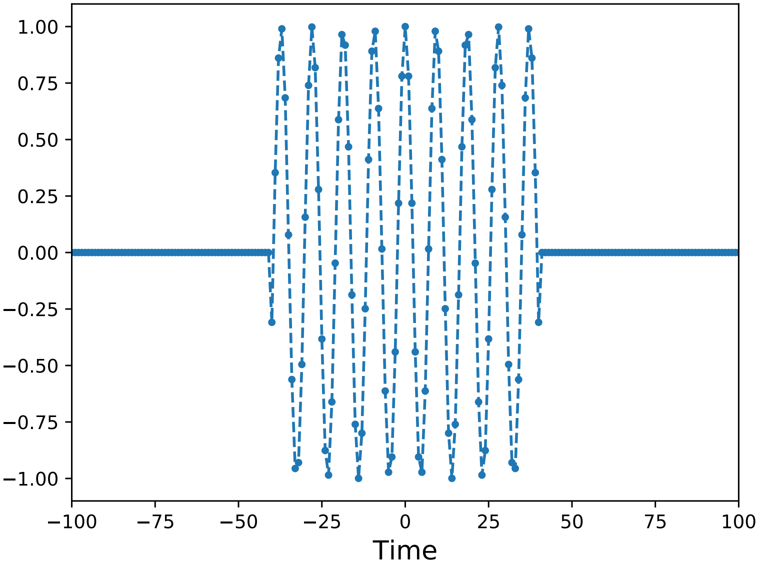


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−100
−75
−50
−25
0
25
50
75
100
Time

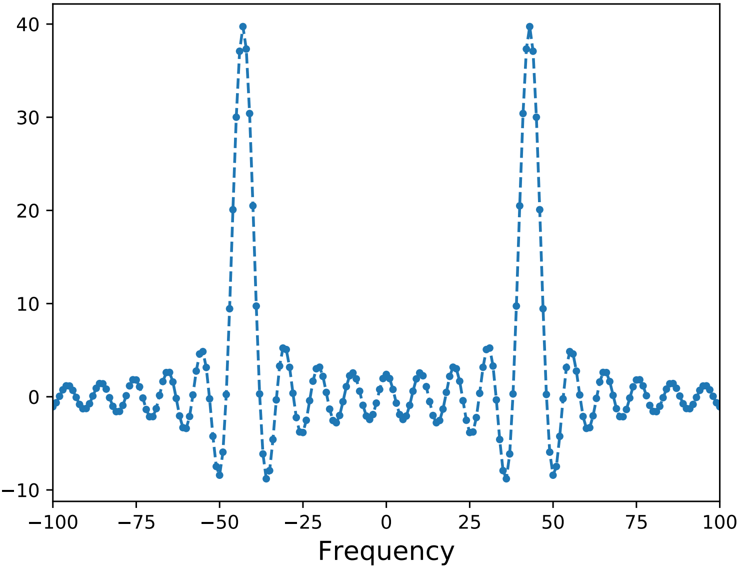
Window



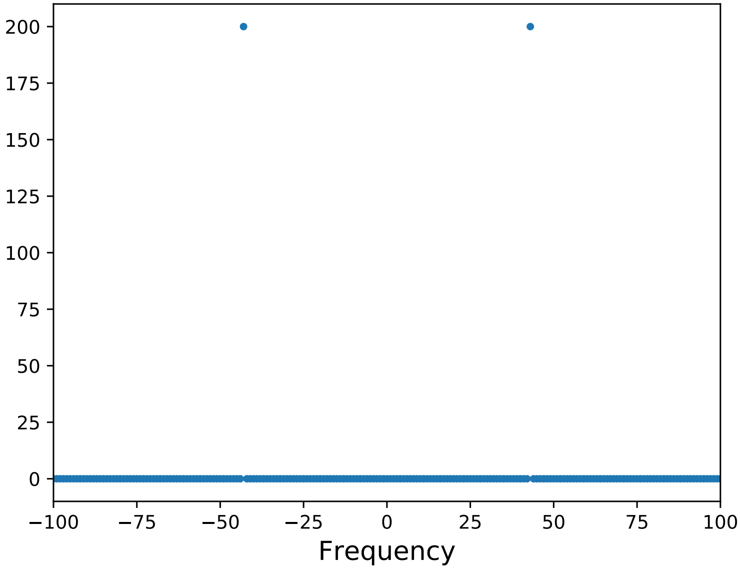
Windowed signal



DFT of windowed signal



DFT of signal



Multiplication in time is convolution in frequency

Let $\vec{y} := \vec{x}_1 \circ \vec{x}_2$ for $\vec{x}_1, \vec{x}_2 \in \mathbb{C}^N$

The DFT of $\vec{y}$ equals

$$\hat{y} = \frac{1}{N} \hat{x}_1 * \hat{x}_2,$$

$\hat{x}_1$ and $\hat{x}_2$ are the DFTs of $\vec{x}_1$ and $\vec{x}_2$ respectively

Proof

$$\hat{y}\left[k\right] := \sum_{j=1}^{N} \vec{x}_1(j)\vec{x}_2(j) \exp\left(-\frac{i2\pi kj}{N}\right)$$

Proof

$$
\begin{aligned}
\hat{y}\left[k\right] &:= \sum_{j=1}^{N} \vec{x}_1(j)\vec{x}_2(j) \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=1}^{N} \frac{1}{N} \sum_{l=1}^{n} \hat{x}_1(l) \exp\left(\frac{i2\pi lj}{N}\right) \vec{x}_2(j) \exp\left(-\frac{i2\pi kj}{N}\right)
\end{aligned}
$$

Proof

$$
\begin{aligned}
\hat{y}[k] &:= \sum_{j=1}^{N} \vec{x}_1(j)\vec{x}_2(j) \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=1}^{N} \frac{1}{N} \sum_{l=1}^{n} \hat{x}_1(l) \exp\left(\frac{i2\pi lj}{N}\right) \vec{x}_2(j) \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \frac{1}{N} \sum_{l=1}^{N} \hat{x}_1(l) \sum_{j=1}^{N} \vec{x}_2(j) \exp\left(-\frac{i2\pi(k-l)j}{N}\right)
\end{aligned}
$$

Proof

$$\begin{aligned}
\hat{y}\,[k] &:= \sum_{j=1}^{N} \vec{x}_1(j)\vec{x}_2(j) \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=1}^{N} \frac{1}{N} \sum_{l=1}^{n} \hat{x}_1(l) \exp\left(\frac{i2\pi lj}{N}\right) \vec{x}_2(j) \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \frac{1}{N} \sum_{l=1}^{N} \hat{x}_1(l) \sum_{j=1}^{N} \vec{x}_2(j) \exp\left(-\frac{i2\pi(k-l)j}{N}\right) \\
&= \frac{1}{N} \sum_{l=1}^{N} \hat{x}_1(l) \hat{x}_2^{\downarrow l}[k]
\end{aligned}$$

DFT of rectangular window

$$\vec{\pi}[j] := \begin{cases} 1 & \text{if } |j| \leq w, \\ 0 & \text{otherwise,} \end{cases}$$

DFT of rectangular window

$$\hat{\pi}(0) = \sum_{j=-N/2+1}^{N/2} \vec{\pi}[j]$$
$$= \sum_{j=-w}^{w} 1 = 2w + 1$$

DFT of rectangular window

$$\hat{\pi}\left(k\right) = \sum_{j=-N/2+1}^{N/2} \vec{x}\left[j\right] \exp\left(-\frac{i2\pi kj}{N}\right)$$

DFT of rectangular window

$$
\begin{aligned}
\hat{\pi}(k) &= \sum_{j=-N/2+1}^{N/2} \vec{x}[j] \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=-w}^{w} \exp\left(-\frac{i2\pi k}{N}\right)^{j}
\end{aligned}
$$

DFT of rectangular window

$$
\begin{aligned}
\hat{\pi}(k) &= \sum_{j=-N/2+1}^{N/2} \vec{x}[j] \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=-w}^{w} \exp\left(-\frac{i2\pi k}{N}\right)^{j} \\
&= \frac{\exp\left(\frac{i2\pi kw}{N}\right) - \exp\left(-\frac{i2\pi k(w+1)}{N}\right)}{1 - \exp\left(-\frac{i2\pi k}{N}\right)}
\end{aligned}
$$

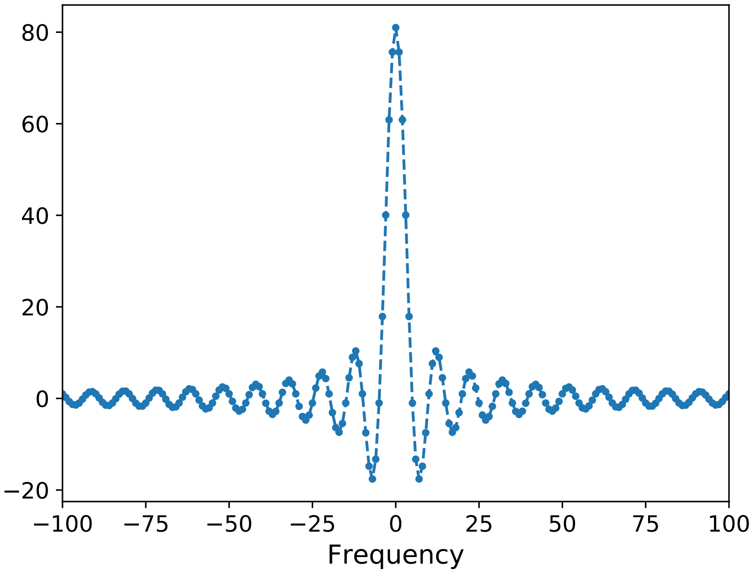
DFT of rectangular window

$$
\begin{aligned}
\hat{\pi}(k) &= \sum_{j=-N/2+1}^{N/2} \vec{x}[j] \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=-w}^{w} \exp\left(-\frac{i2\pi k}{N}\right)^{j} \\
&= \frac{\exp\left(\frac{i2\pi kw}{N}\right) - \exp\left(-\frac{i2\pi k(w+1)}{N}\right)}{1 - \exp\left(-\frac{i2\pi k}{N}\right)} \\
&= \frac{\exp\left(-\frac{i2\pi k}{2N}\right) 2i \sin\left(\frac{2\pi k(w+1/2)}{N}\right)}{\exp\left(-\frac{i2\pi k}{2N}\right) 2i \sin\left(\frac{\pi k}{N}\right)}
\end{aligned}
$$

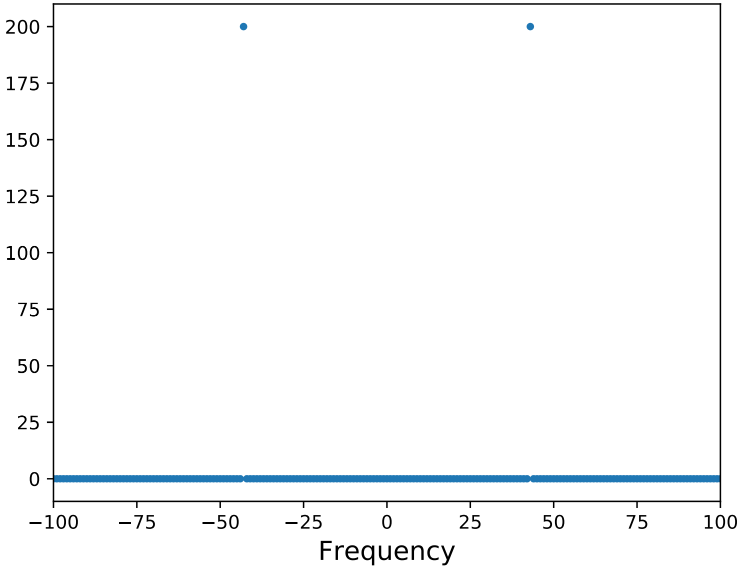
DFT of rectangular window

$$
\begin{aligned}
\hat{\pi}(k) &= \sum_{j=-N/2+1}^{N/2} \vec{x}[j] \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=-w}^{w} \exp\left(-\frac{i2\pi k}{N}\right)^{j} \\
&= \frac{\exp\left(\frac{i2\pi kw}{N}\right) - \exp\left(-\frac{i2\pi k(w+1)}{N}\right)}{1 - \exp\left(-\frac{i2\pi k}{N}\right)} \\
&= \frac{\exp\left(-\frac{i2\pi k}{2N}\right) 2i \sin\left(\frac{2\pi k(w+1/2)}{N}\right)}{\exp\left(-\frac{i2\pi k}{2N}\right) 2i \sin\left(\frac{\pi k}{N}\right)} \\
&= \frac{\sin\left(\frac{2\pi k(w+1/2)}{N}\right)}{\sin\left(\frac{\pi k}{N}\right)}
\end{aligned}
$$

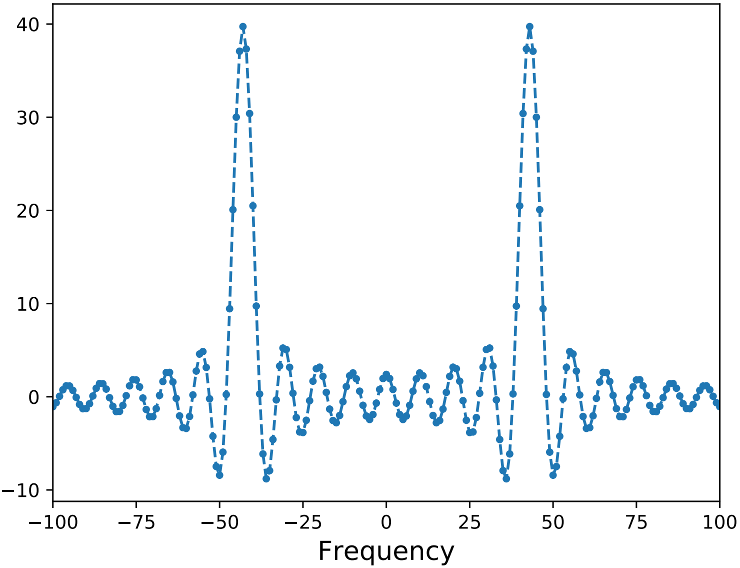
DFT of rectangular window



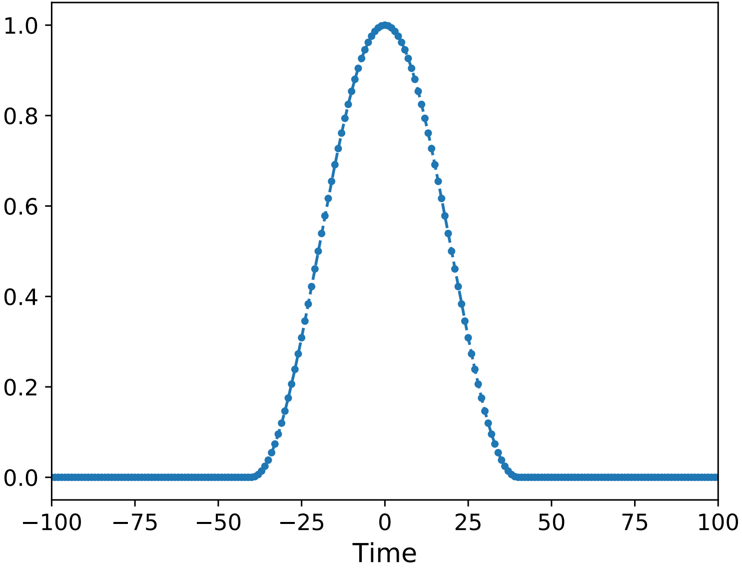
DFT of signal



DFT of windowed signal



Hann window



Hann window

The Hann window $h \in \mathbb{C}^N$ of width $2w$ equals

$$\vec{h}[j] := \begin{cases} \frac{1}{2}\left(1 + \cos\left(\frac{\pi j}{w}\right)\right) & \text{if } |j| \leq w, \\ 0 & \text{otherwise} \end{cases}$$

Shifting in the frequency domain

For any vector $\vec{x} \in \mathbb{C}^N$, if $\vec{y} \in \mathbb{C}^N$ is defined as

$$\vec{y}\,[j] := \vec{x}[j] \exp\left(\frac{i2\pi mj}{N}\right),$$

then the DFT of $\vec{y}$ equals $\hat{y} = \hat{x}^{\downarrow m}$, where $\hat{x}$ is the DFT of $\vec{x}$

Proof

$$\hat{y}\left[k\right] = \sum_{j=1}^{N} \vec{y}\left[j\right] \exp\left(-\frac{i2\pi kj}{N}\right)$$

Proof

$$
\begin{aligned}
\hat{y}\left[k\right] &= \sum_{j=1}^{N} \vec{y}\left[j\right] \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=1}^{N} \vec{x}[j] \exp\left(\frac{i2\pi mj}{N}\right) \exp\left(-\frac{i2\pi kj}{N}\right)
\end{aligned}
$$

Proof

$$
\begin{aligned}
\hat{y}\,[k] &= \sum_{j=1}^{N} \vec{y}\,[j] \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=1}^{N} \vec{x}[j] \exp\left(\frac{i2\pi mj}{N}\right) \exp\left(-\frac{i2\pi kj}{N}\right) \\
&= \sum_{j=1}^{N} \vec{x}[j] \exp\left(-\frac{i2\pi(k-m)j}{N}\right)
\end{aligned}
$$

DFT of the Hann window

$$\vec{h}[j] = \frac{1}{2}\left(1 + \cos\left(\frac{\pi j}{w}\right)\right)\vec{\pi}[j]$$

DFT of the Hann window

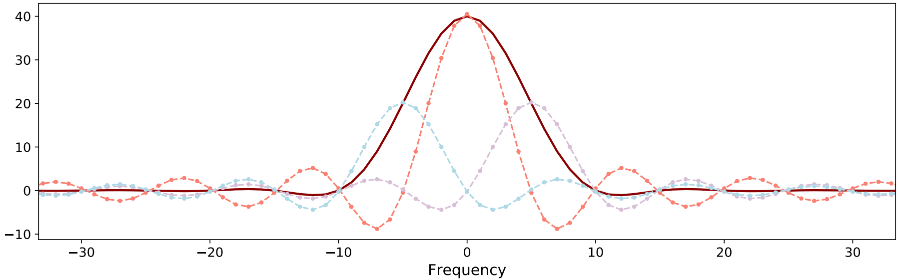
$$
\begin{aligned}
\vec{h}[j] &= \frac{1}{2}\left(1+\cos\left(\frac{\pi j}{w}\right)\right)\vec{\pi}[j] \\
&= \frac{1}{2}\left(1+\frac{1}{2}\exp\left(\frac{i2\pi Nj}{2wN}\right)+\frac{1}{2}\exp\left(-\frac{i2\pi Nj}{2wN}\right)\right)\vec{\pi}[j]
\end{aligned}
$$

DFT of the Hann window

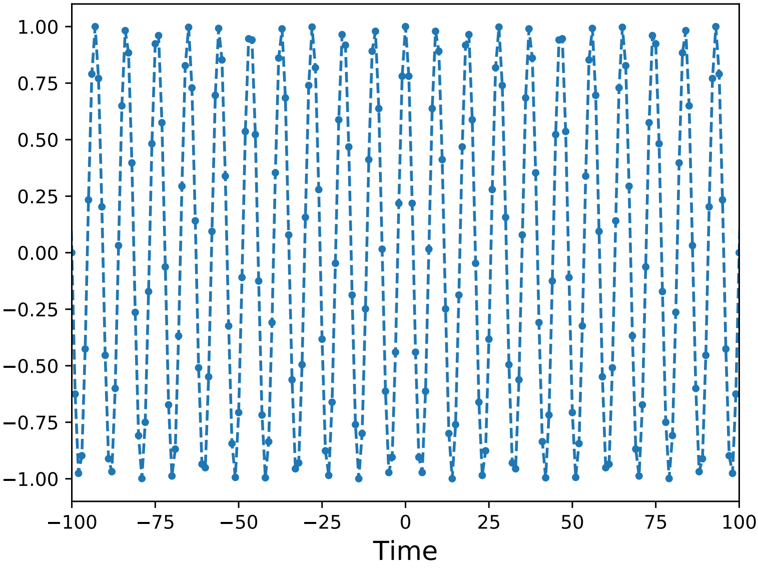
$$\begin{aligned}\vec{h}[j] &= \frac{1}{2}\left(1+\cos\left(\frac{\pi j}{w}\right)\right)\vec{\pi}[j] \\ &= \frac{1}{2}\left(1+\frac{1}{2}\exp\left(\frac{i2\pi Nj}{2wN}\right)+\frac{1}{2}\exp\left(-\frac{i2\pi Nj}{2wN}\right)\right)\vec{\pi}[j]\end{aligned}$$

$$\hat{h} = \frac{1}{2}\hat{\pi} + \frac{1}{4}\hat{\pi}^{\downarrow -N/2w} + \frac{1}{4}\hat{\pi}^{\downarrow N/2w}$$

DFT of the Hann window

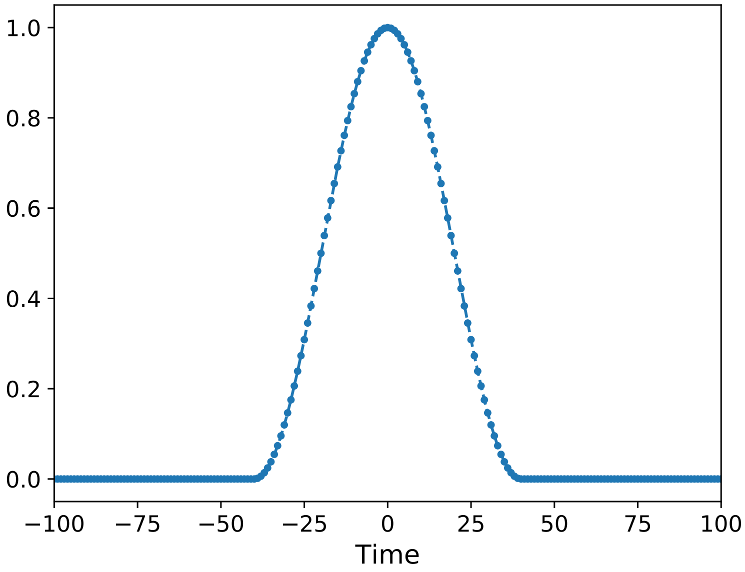


Signal

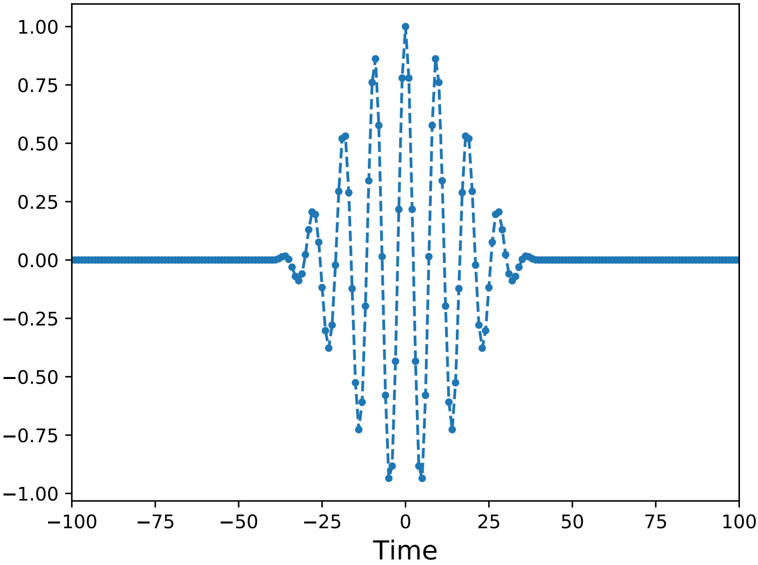


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−100
−75
−50
−25
0
25
50
75
100
Time

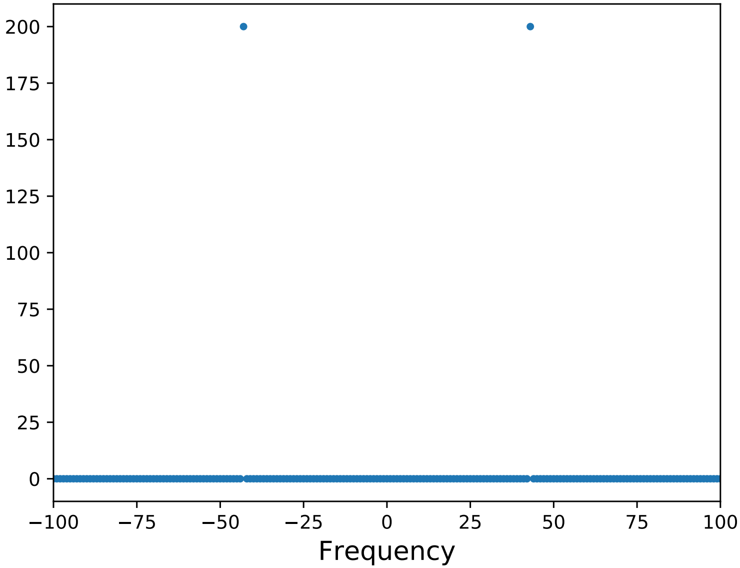
Hann window



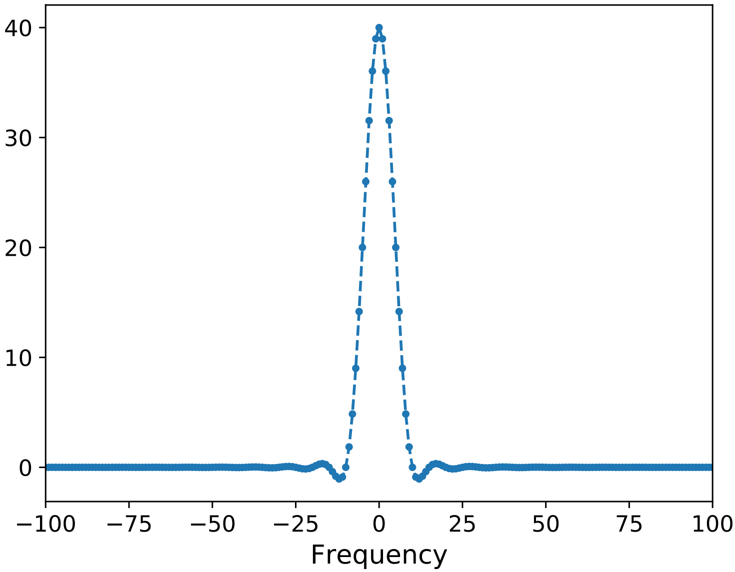
Windowed signal



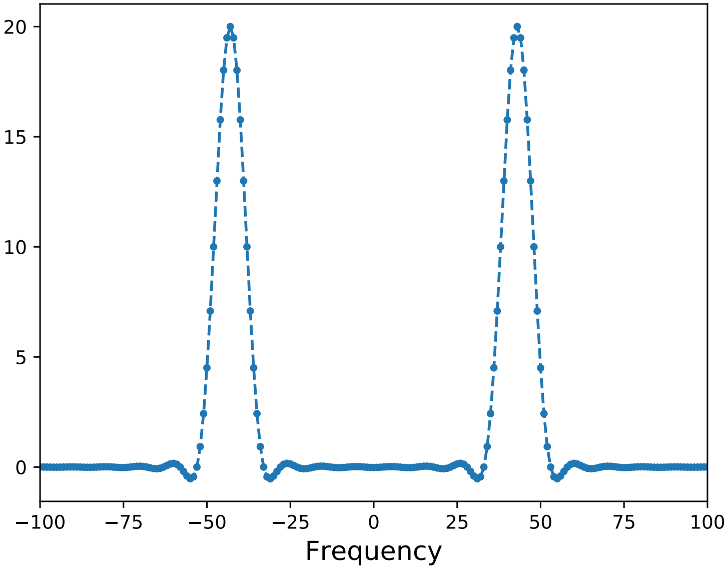
DFT of signal



DFT of Hann window



DFT of windowed signal



Time-frequency resolution

Time resolution governed by width of window

Can we just make the window arbitrarily narrow?

Compressing in time dilates in frequency and vice versa

$x \in \mathcal{L}_2[-T/2, T/2]$ is nonzero in a band of width $2w$ around zero

Let y be such that

$$y(t) = x(\alpha t), \quad \text{for all } t \in [-T/2, T/2],$$

for some positive real number α such that $w/\alpha < T$

The Fourier series coefficients of y equal

$$\hat{y}[k] = \frac{1}{\alpha} \left\langle x, \phi_{k/\alpha} \right\rangle$$

Proof

$$\hat{y}[k] = \int_{t=-T/2}^{T/2} y(t) \exp\left(-\frac{i2\pi kt}{T}\right) \, \mathrm{d}t$$

Proof

$$
\begin{aligned}
\hat{y}\left[k\right] &= \int_{t=-T/2}^{T/2} y(t) \exp\left(-\frac{i2\pi kt}{T}\right) \, \mathrm{d}t \\
&= \int_{t=-w/\alpha}^{w/\alpha} x(\alpha t) \exp\left(-\frac{i2\pi kt}{T}\right) \, \mathrm{d}t
\end{aligned}
$$

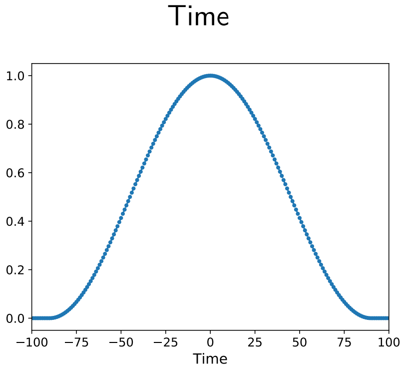
Proof

$$
\begin{aligned}
\hat{y}[k] &= \int_{t=-T/2}^{T/2} y(t) \exp\left(-\frac{i2\pi k t}{T}\right) \, \mathrm{d}t \\
&= \int_{t=-w/\alpha}^{w/\alpha} x(\alpha t) \exp\left(-\frac{i2\pi k t}{T}\right) \, \mathrm{d}t \\
&= \frac{1}{\alpha} \int_{\tau=-w}^{w} x(\tau) \exp\left(-\frac{i2\pi k \tau}{\alpha T}\right) \, \mathrm{d}\tau
\end{aligned}
$$

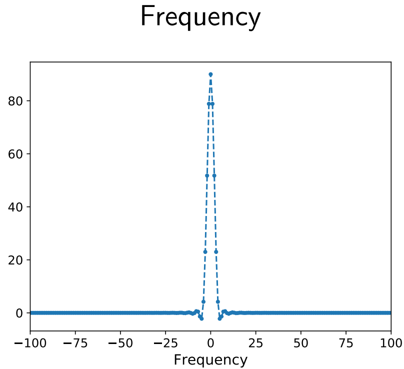
Proof

$$
\begin{aligned}
\hat{y}[k] &= \int_{t=-T/2}^{T/2} y(t) \exp\left(-\frac{i2\pi kt}{T}\right) \, dt \\
&= \int_{t=-w/\alpha}^{w/\alpha} x(\alpha t) \exp\left(-\frac{i2\pi kt}{T}\right) \, dt \\
&= \frac{1}{\alpha} \int_{\tau=-w}^{w} x(\tau) \exp\left(-\frac{i2\pi k\tau}{\alpha T}\right) \, d\tau \\
&= \frac{1}{\alpha} \int_{\tau=-T/2}^{T/2} x(\tau) \exp\left(-\frac{i2\pi k\tau}{\alpha T}\right) \, d\tau
\end{aligned}
$$

$w = 90$

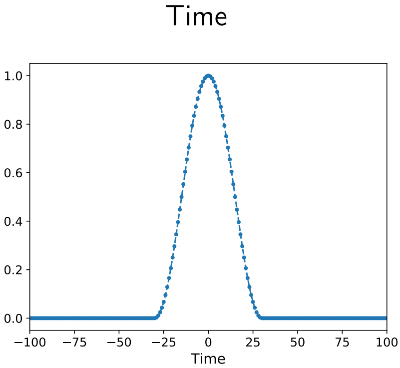


Time
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
Time

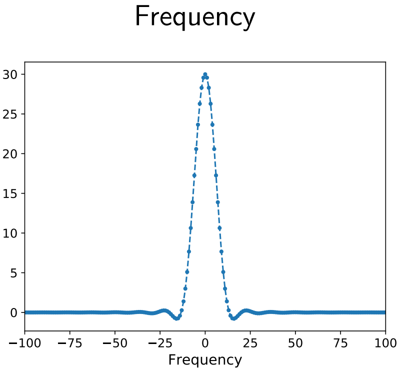


Frequency
80
60
40
20
0
−100
−75
−50
−25
0
25
50
75
100
Frequency

$w = 30$

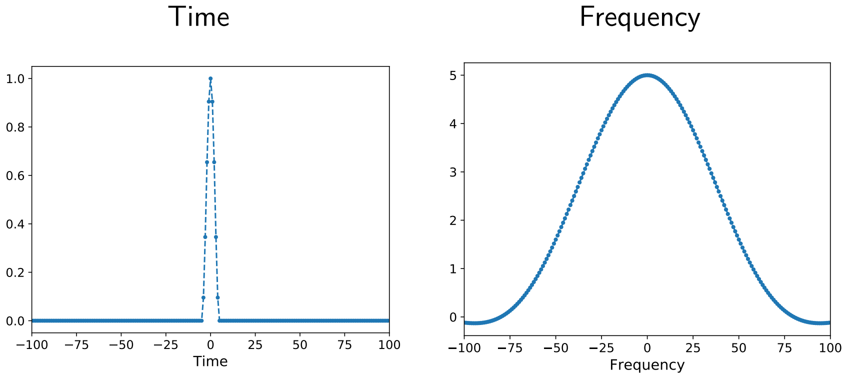


Time
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
Time



Frequency
30
25
20
15
10
5
0
−100
−75
−50
−25
0
25
50
75
100
Frequency

$w = 5$



Time
Frequency
1.0
0.8
0.6
0.4
0.2
0.0
−100
−75
−50
−25
0
25
50
75
100
Time
5
4
3
2
1
0
−100
−75
−50
−25
0
25
50
75
100
Frequency

Time-frequency resolution

Fundamental trade-off

Uncertainty principle: cannot resolve in time and frequency simultaneously

Windowing

Short-time Fourier transform

Multiresolution analysis

Denoising via thresholding

Short-time Fourier transform

1. Segment in overlapping intervals of length ℓ
2. Multiply by window vector
3. Compute DFT of length ℓ

Short-time Fourier transform

The short-time Fourier transform of $\vec{x} \in \mathbb{C}^N$ is

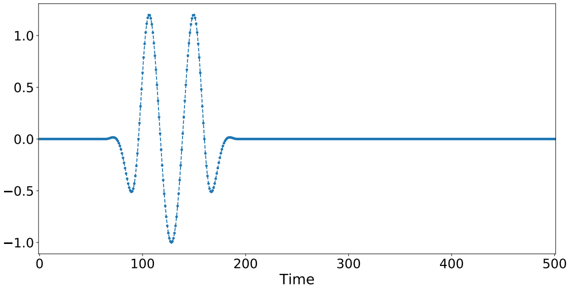
$$\text{STFT}_{[\ell]}(\vec{x})[k,s] := \left\langle \vec{x}, \vec{\psi}_k^{\downarrow s(1-\alpha_{\text{ov}})\ell} \right\rangle, \quad 0 \leq k \leq \ell - 1,\ 0 \leq s \leq \frac{N}{(1-\alpha_{\text{ov}})\ell},$$

$$\vec{\psi}_k[j] := \begin{cases} w_{[\ell]}(j) \exp\left(\frac{i 2\pi k j}{\ell}\right) & \text{if } 1 \leq j \leq \ell \\ 0 & \text{otherwise} \end{cases}$$

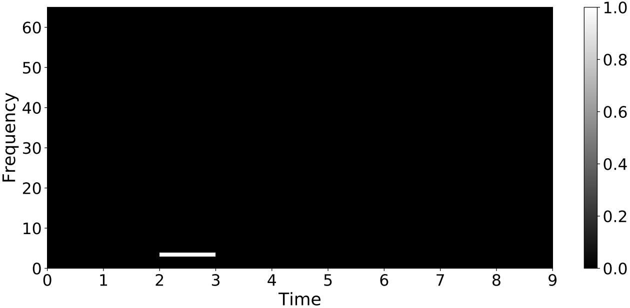
Overlap between adjacent segments equals $\alpha_{\text{ov}}\ell$

$k = 3$ $s = 2$ ($N := 500$, $\ell := 128$, $\alpha_{\mathrm{ov}} := 0.5$)

Basis vector

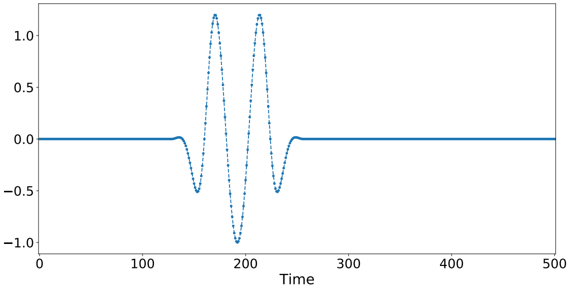


STFT
coefficient

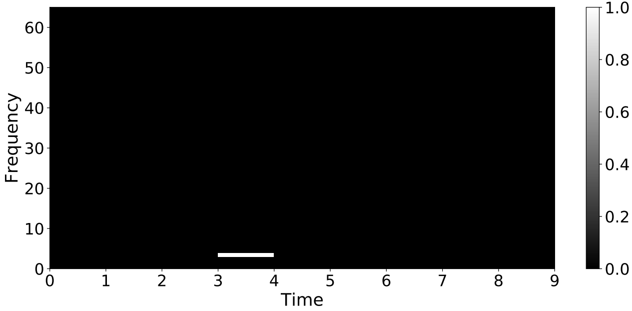


$k = 3$ $s = 3$ ($N := 500$, $\ell := 128$, $\alpha_{\mathrm{ov}} := 0.5$)

Basis vector

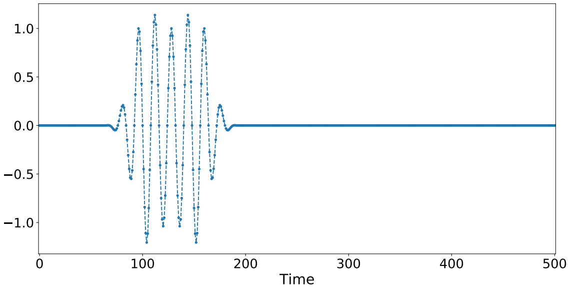


STFT
coefficient

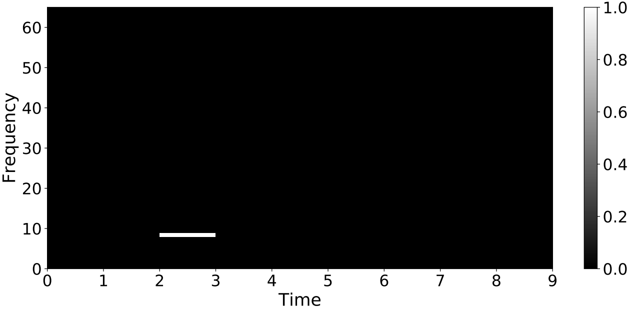


$k = 8$ $s = 2$ ($N := 500$, $\ell := 128$, $\alpha_{\mathrm{ov}} := 0.5$)

Basis vector

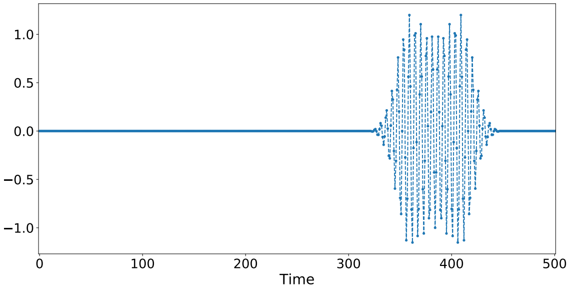


STFT
coefficient

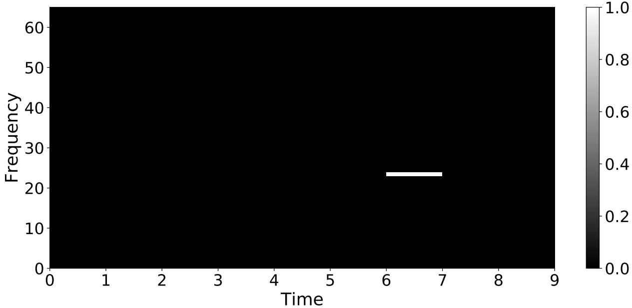


$k = 23$ $s = 6$ ($N := 500$, $\ell := 128$, $\alpha_{\text{ov}} := 0.5$)

Basis vector



STFT
coefficient



Matrix representation of STFT

$$
\begin{bmatrix}
\multicolumn{2}{c}{\multirow{2}{*}{$F_{[\ell]}$}} & 0 & 0 & 0 & 0 & \cdots \\
 & & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$F_{[\ell]}$}} & 0 & 0 & \cdots \\
0 & 0 & & & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$F_{[\ell]}$}} & \cdots \\
0 & 0 & 0 & 0 & & & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\begin{bmatrix}
\multicolumn{1}{c}{\multirow{2}{*}{$\text{diag}\left(w_{[\ell]}\right)$}} & 0 & 0 & \cdots \\
 & 0 & 0 & \cdots \\
0 & \multirow{2}{*}{$\text{diag}\left(w_{[\ell]}\right)$} & 0 & \cdots \\
0 & & 0 & \cdots \\
0 & 0 & \multirow{2}{*}{$\text{diag}\left(w_{[\ell]}\right)$} & \cdots \\
0 & 0 & & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\vec{x},
$$

Computing the STFT

Number of segments: $n_{\mathsf{seg}} := N/(1-\alpha_{\mathsf{ov}})\ell$

Complexity of multiplication by window:

Complexity of applying DFT:

Total complexity:

Computing the STFT

Number of segments: $n_{\text{seg}} := N/(1-\alpha_{\text{ov}})\ell$

Complexity of multiplication by window: $n_{\text{seg}}\ell$

Complexity of applying DFT:

Total complexity:

Computing the STFT

Number of segments: $n_{\text{seg}} := N/(1-\alpha_{\text{ov}})\ell$

Complexity of multiplication by window: $n_{\text{seg}}\ell$

Complexity of applying DFT: $n_{\text{seg}}\ell \log \ell$ (FFT)

Total complexity:

Computing the STFT

Number of segments: $n_{\text{seg}} := N/(1-\alpha_{\text{ov}})\ell$

Complexity of multiplication by window: $n_{\text{seg}}\ell$

Complexity of applying DFT: $n_{\text{seg}}\ell \log \ell$ (FFT)

Total complexity: $O(N \log \ell)$ (overlap is a fixed fraction)

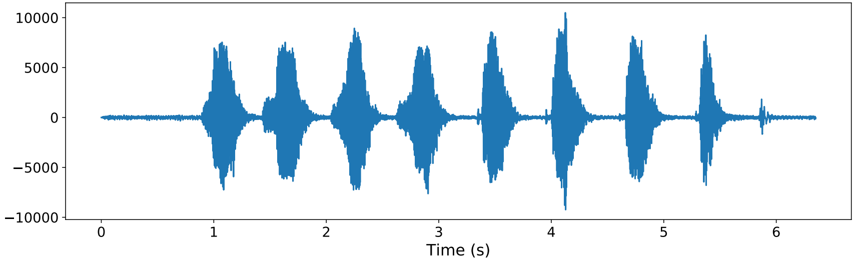
Inverting the STFT

Apply inverse DFT to each segment

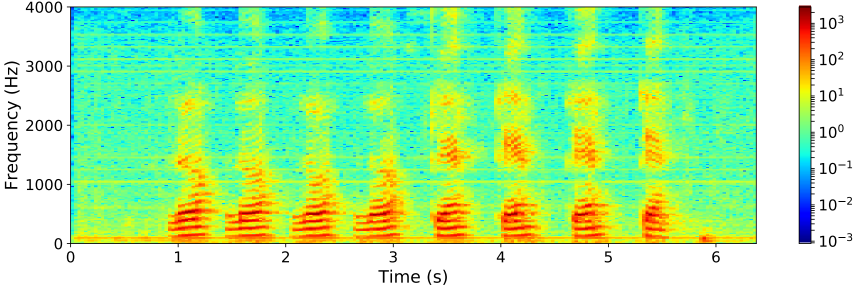
Combine segments

Same complexity

Speech signal (window length = 62.5 ms)



Speech signal (window length = 62.5 ms)



Windowing

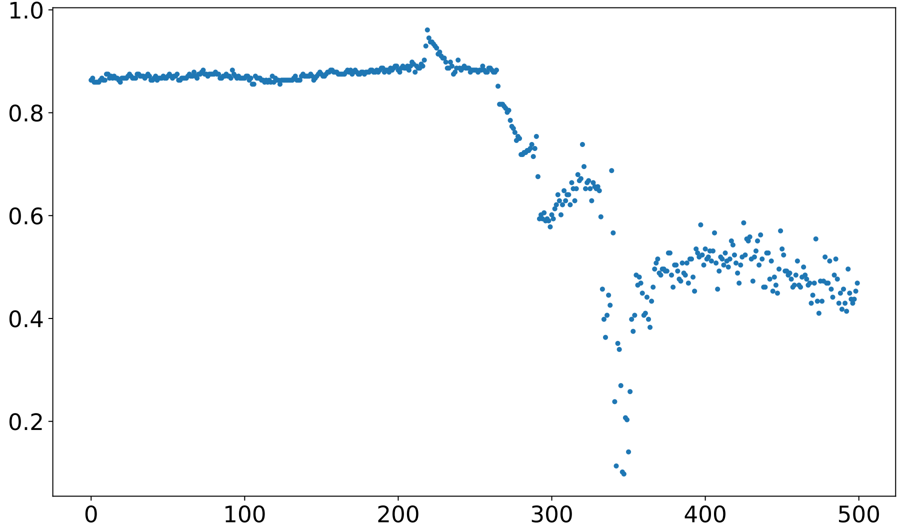
Short-time Fourier transform

Multiresolution analysis

Denoising via thresholding

Image

Vertical line (column 135)



Multiresolution analysis

Scale / resolution at which information is encoded is not uniform

Goal: Decompose signals into components at different resolutions

Multiresolution decomposition

Let $N := 2^K$ for some K, a multiresolution decomposition of $\mathbb{C}^N$ is a sequence of nested subspaces $\mathcal{V}_K \subset \mathcal{V}_{K-1} \subset \ldots \subset \mathcal{V}_0$ satisfying:

- $\mathcal{V}_0 = \mathbb{C}^N$
- $\mathcal{V}_k$ is invariant to translations of scale 2^k for $0 \leq k \leq K$. If $\vec{x} \in \mathcal{V}_k$ then

$$\vec{x}^{\downarrow 2^k} \in \mathcal{V}_k \qquad \text{for all } l \in \mathbb{Z}$$

- For any $\vec{x} \in \mathcal{V}_j$ that is nonzero only between 1 and $N/2$, the dilated vector $\vec{x}_{\leftrightarrow 2}$ belongs to $\mathcal{V}_{j+1}$

Dilation

Let $\vec{x} \in \mathbb{C}^N$ be such that $\vec{x}[j] = 0$ for all $j \geq N/M$, where M is a positive integer

The dilation of $\vec{x}$ by a factor of M is

$$\vec{x}_{\leftrightarrow M}[j] = \vec{x}\left[\left\lceil \frac{j}{M} \right\rceil\right]$$

How to build a multiresolution decomposition

- Set the coarsest subspace to be spanned by a low-frequency vector $\vec{\varphi}$, called a scaling vector or father wavelet

$$\mathcal{V}_K := \operatorname{span}\left(\vec{\varphi}\right).$$

How to build a multiresolution decomposition

- Decompose the finer subspaces into the direct sum

$$\mathcal{V}_k := \mathcal{W}_k \oplus \mathcal{V}_{k+1}, \qquad 0 \leq k \leq K-1,$$

  where $\mathcal{W}_k$ captures the finest resolution available at level k

- Set $\mathcal{W}_k$ to be spanned by shifts of a vector $\vec{\mu}$ dilated to have the appropriate resolution:

$$\mathcal{V}_k := \mathcal{W}_k \oplus \mathcal{V}_{k+1}, \qquad 0 \leq k \leq K-1,$$

$$\mathcal{W}_{K-1} := \bigoplus_{m=0}^{\frac{N-1}{2^{k+1}}} \operatorname{span}\left(\vec{\mu}_{\leftrightarrow 2^k}^{\downarrow m2^{k+1}}\right).$$

  The vector $\vec{\mu}$ is called a mother wavelet

Challenge

How to choose mother and father wavelets?

If chosen appropriately, basis vectors can be orthonormal

Haar wavelet basis

The Haar father wavelet $\vec{\varphi}$ is a constant vector, such that

$$\vec{\varphi}[j] := \frac{1}{\sqrt{N}}, \qquad 1 \leq j \leq N$$

The mother wavelet $\vec{\mu}$ satisfies

$$\vec{\mu}[j] := \begin{cases} -\frac{1}{\sqrt{2}}, & j = 1, \\ \frac{1}{\sqrt{2}}, & j = 2, \\ 0, & j > 2 \end{cases}$$

Other options: Meyer, Daubechies, coiflets, symmlets, etc.

Haar wavelets

Scale

2^0

Basis functions

Haar wavelets

Scale

Basis functions

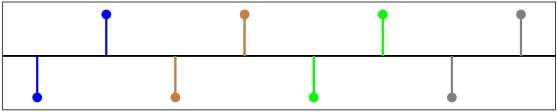
2^0

Haar wavelets

Scale

Basis functions

2^0

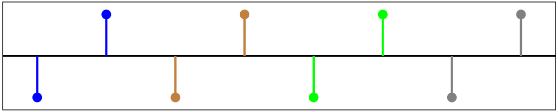


Haar wavelets

Scale

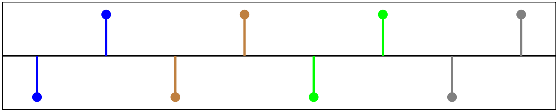
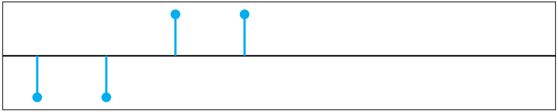
Basis functions

2^0



2^1

Haar wavelets

Scale	Basis functions
2^0	
2^1	

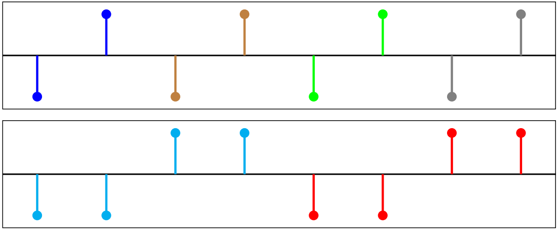
Haar wavelets

Scale

Basis functions

2^0

2^1



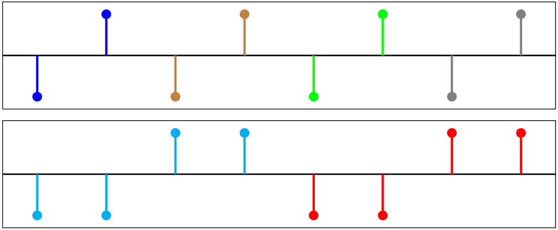
Haar wavelets

Scale | Basis functions

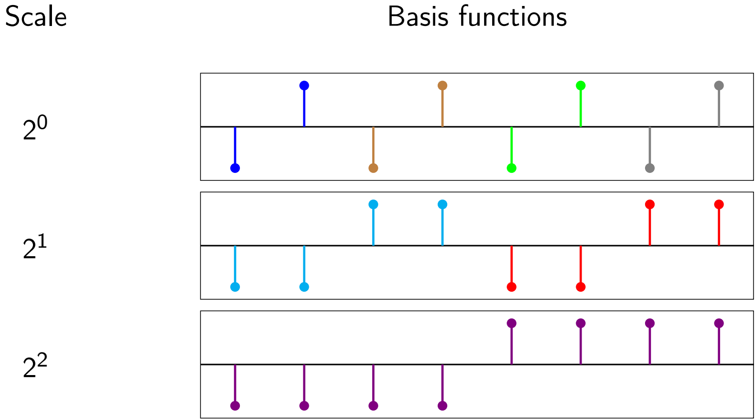
2^0

2^1

2^2



Haar wavelets



Haar wavelets

Scale

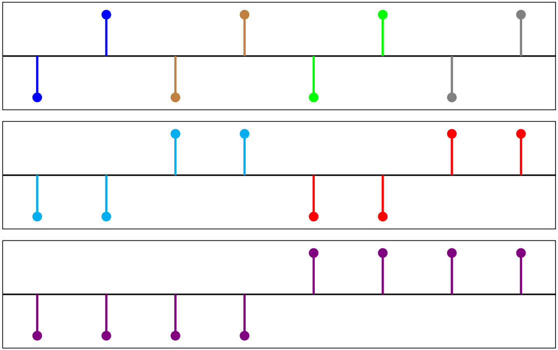
Basis functions

2^0

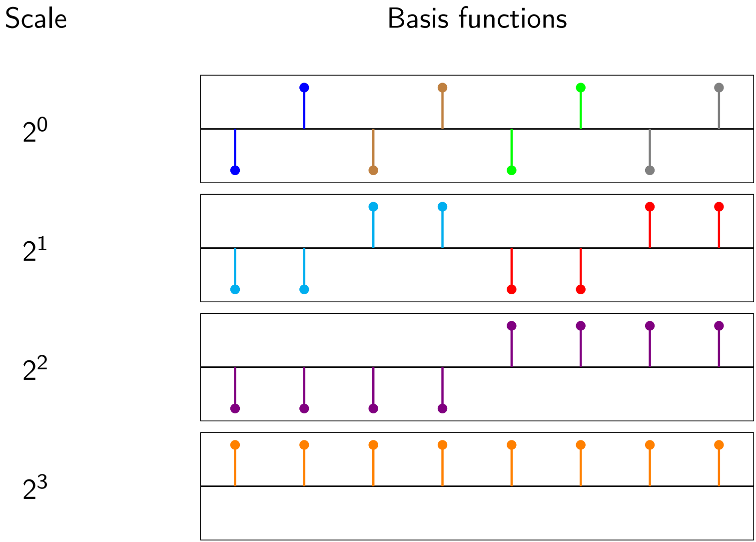
2^1

2^2

2^3

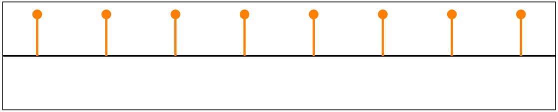


Haar wavelets

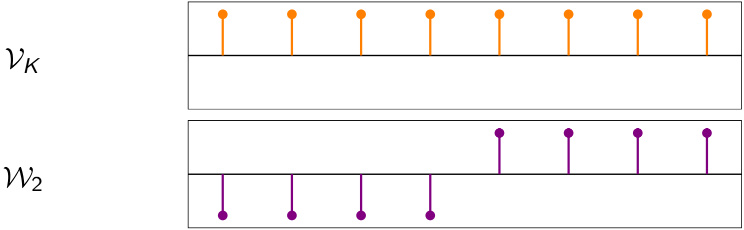


Multiresolution decomposition

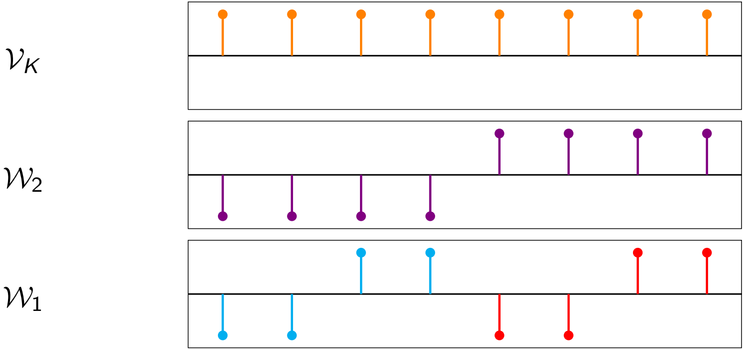
$\mathcal{V}_K$



Multiresolution decomposition



Multiresolution decomposition



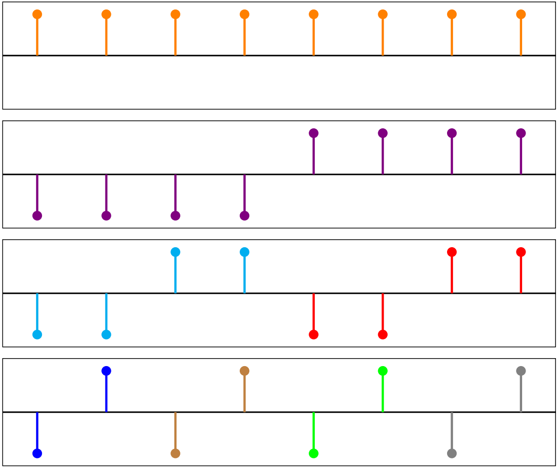
Multiresolution decomposition

$\mathcal{V}_K$

$\mathcal{W}_2$

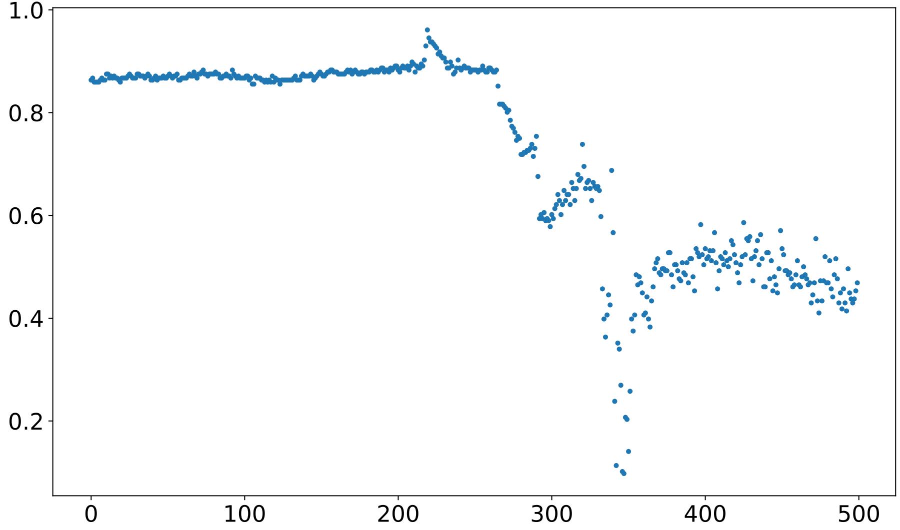
$\mathcal{W}_1$

$\mathcal{W}_0$

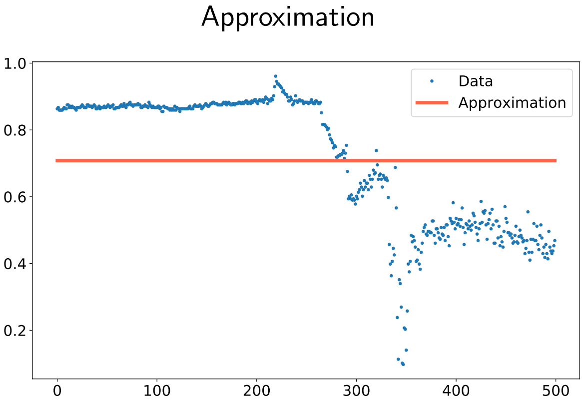


$\mathcal{P}_{\mathcal{V}_k}\,\vec{x}$ is an approximation of $\vec{x}$ at scale 2^k

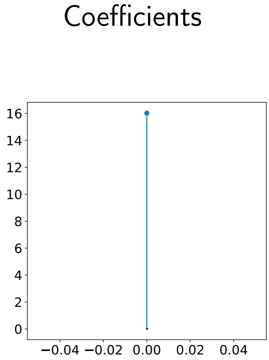
Vertical line (column 135)



Scale 2^9

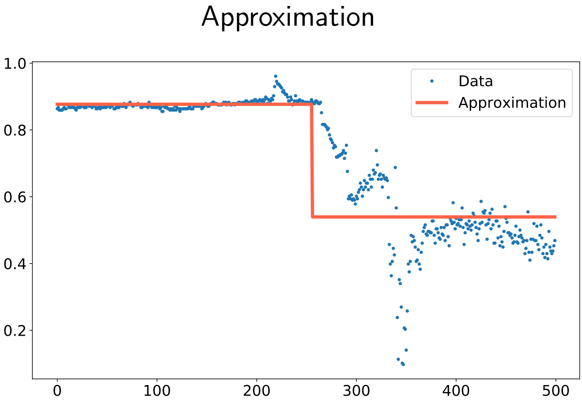


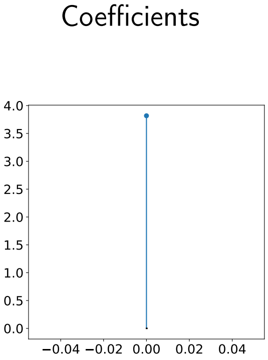
Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500
Data
Approximation



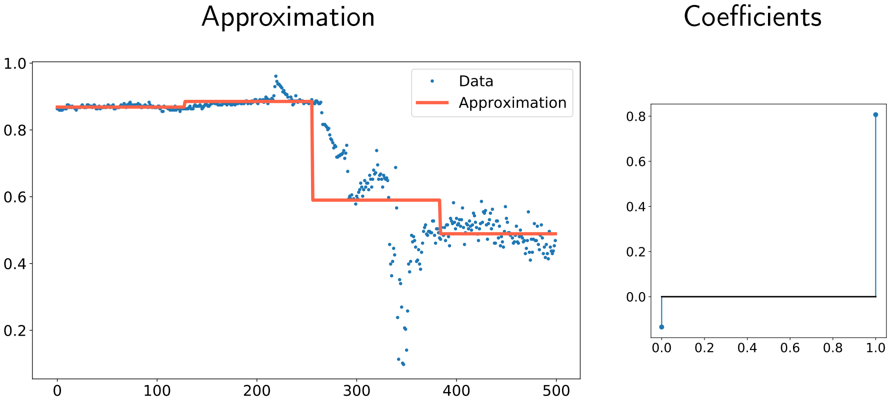
Coefficients
16
14
12
10
8
6
4
2
0
−0.04
−0.02
0.00
0.02
0.04

Scale 2^8

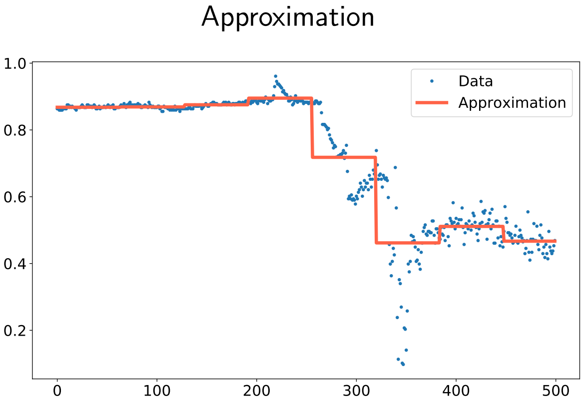

Approximation
Data
Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500


Coefficients
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.04
−0.02
0.00
0.02
0.04

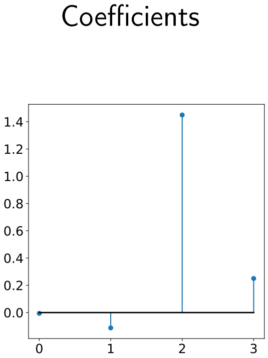
Scale 2^7


Approximation
Coefficients
Data
Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Scale 2^6

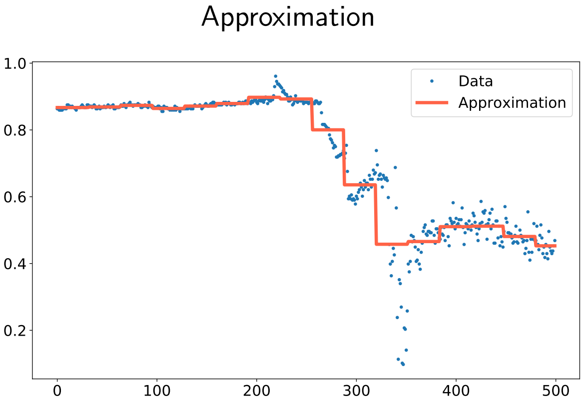


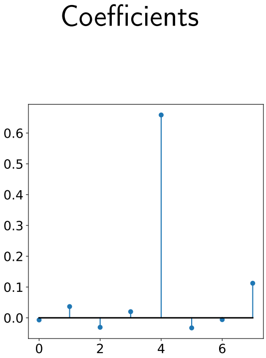
Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500
Data
Approximation



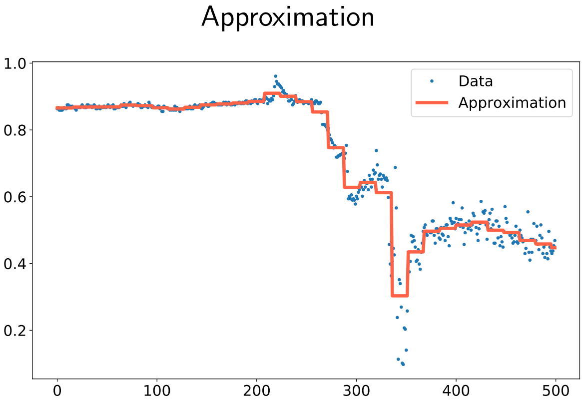
Coefficients
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3

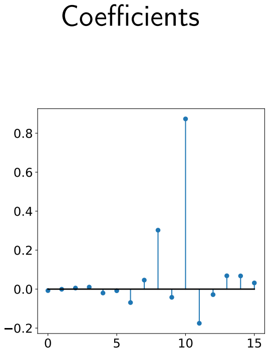
Scale 2^5


Approximation
Data
Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500

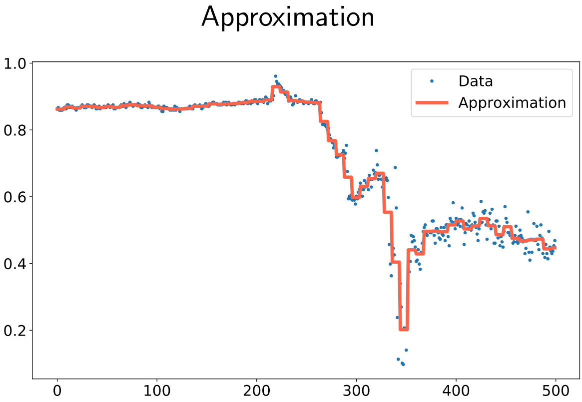

Coefficients
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6

Scale 2^4

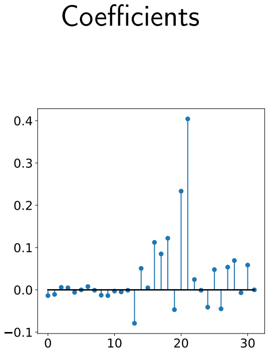




Scale 2^3

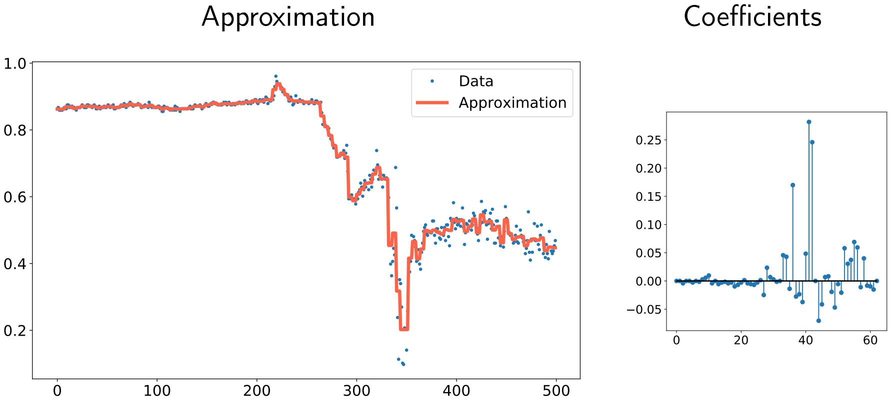


Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500
Data
Approximation

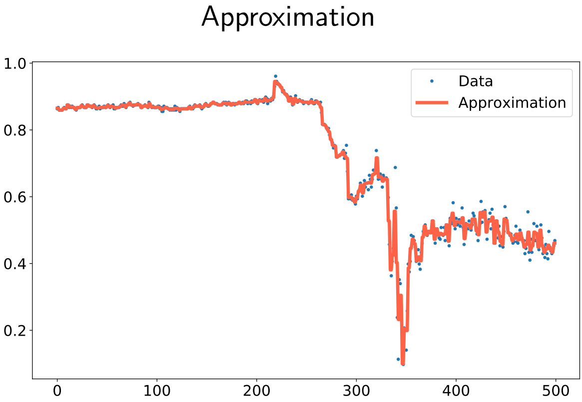


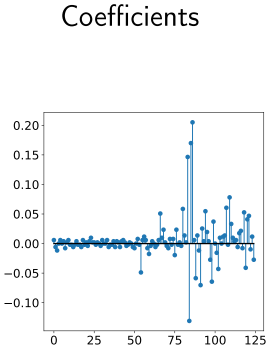
Coefficients
0.4
0.3
0.2
0.1
0.0
−0.1
0
10
20
30

Scale 2^2

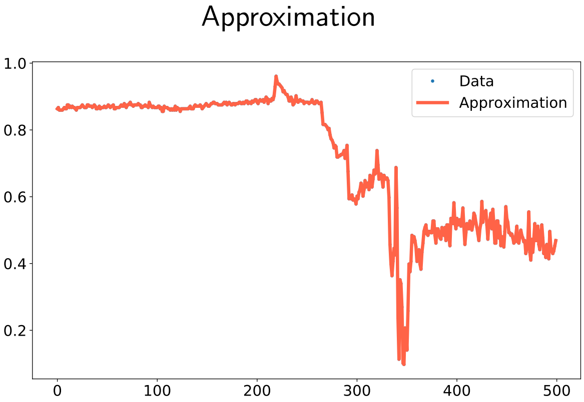


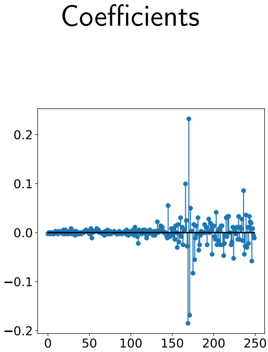
Scale 2^1


Approximation
Data
Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500

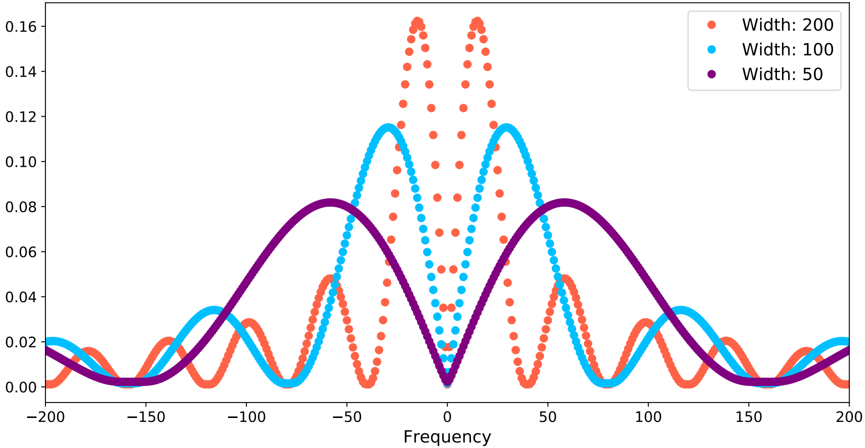

Coefficients
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
0
25
50
75
100
125

Scale 2^0


Approximation
Data
Approximation
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500

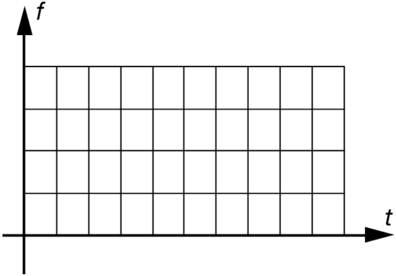

Coefficients
0.2
0.1
0.0
−0.1
−0.2
0
50
100
150
200
250

Haar wavelets in the frequency domain

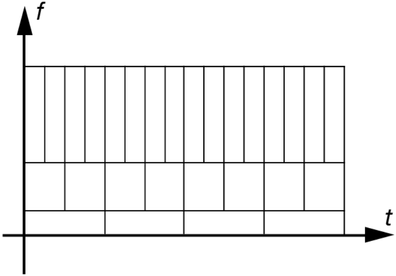


Time-frequency support of basis vectors

STFT



Wavelets



2D Wavelets

Extension to 2D by using outer products of 1D basis vectors

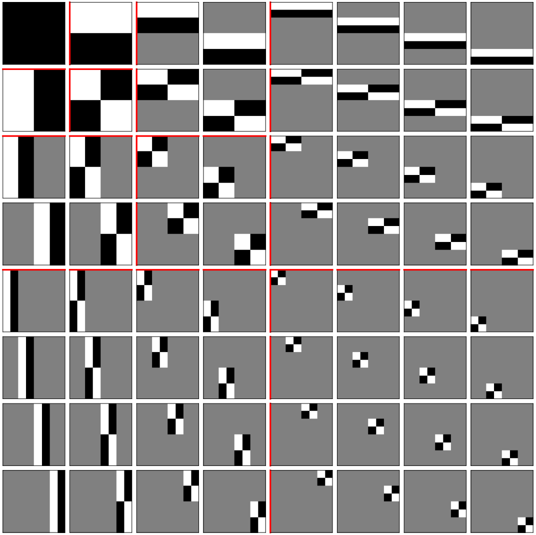
To build a 2D basis vector at scale (m_1, m_2) and shift (s_1, s_2) we set

$$\xi^{\text{2D}}_{[s_1,s_2,m_1,m_2]} := \xi^{\text{1D}}_{[s_1,m_1]} \left(\xi^{\text{1D}}_{[s_2,m_2]}\right)^T,$$

where ξ^{1D} can refer to 1D father or mother wavelets

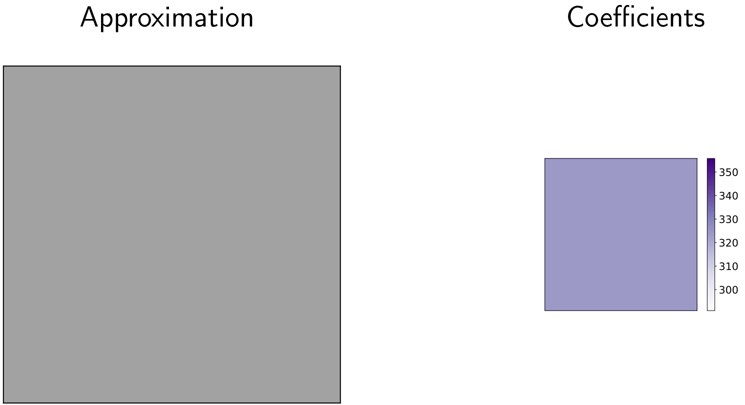
Nonseparable designs: steerable pyramid, curvelets, bandlets...

2D Haar wavelet basis vectors



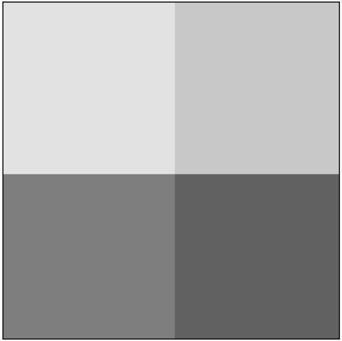
Image

2D Haar wavelet decomposition

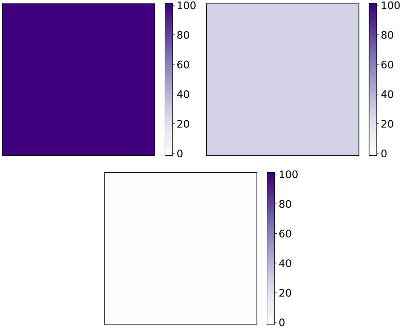


2D Haar wavelet decomposition

Approximation

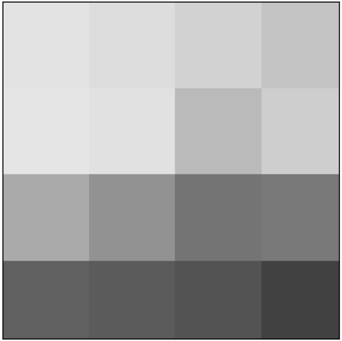


Coefficients

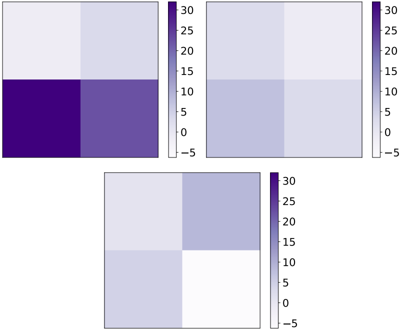


2D Haar wavelet decomposition

Approximation

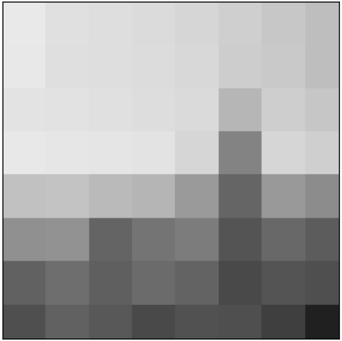


Coefficients

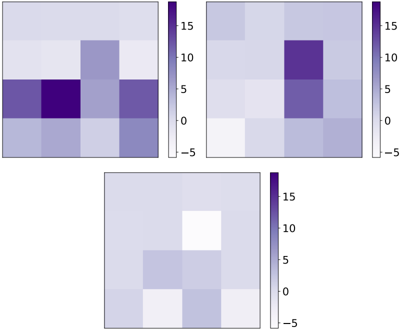


2D Haar wavelet decomposition

Approximation

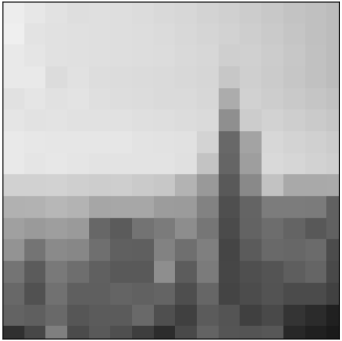


Coefficients

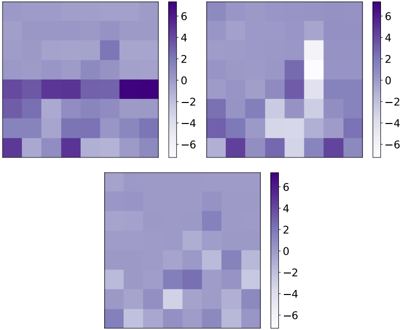


2D Haar wavelet decomposition

Approximation



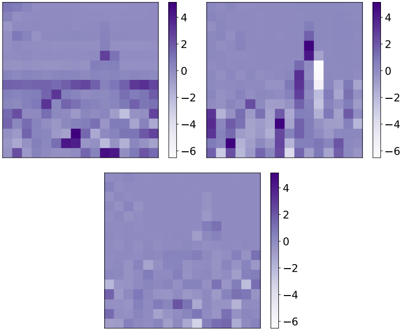
Coefficients



2D Haar wavelet decomposition

Approximation

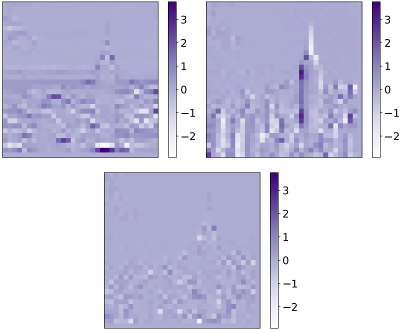
Coefficients



2D Haar wavelet decomposition

Approximation

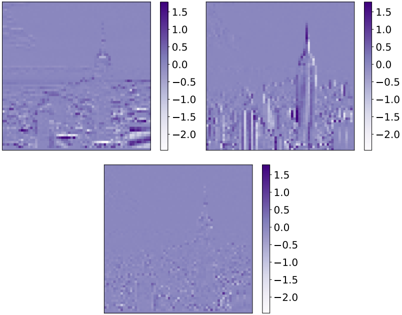
Coefficients



2D Haar wavelet decomposition

Approximation

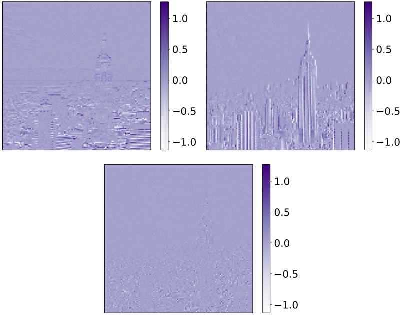
Coefficients



2D Haar wavelet decomposition

Approximation

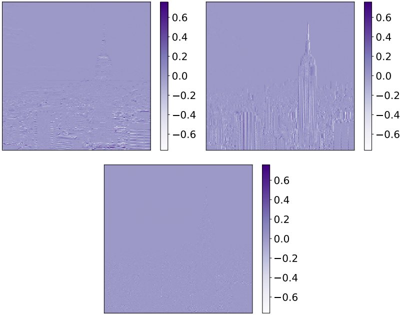
Coefficients



2D Haar wavelet decomposition

Approximation

Coefficients



Denoising

Aim: Estimate signal $\vec{x}$ from data of the form

$$\vec{y} = \vec{x} + \vec{z}$$

Motivation

STFT coefficients of audio and wavelet coefficients of images are sparse

Coefficients of noise are dense

Idea: Get rid of small entries in

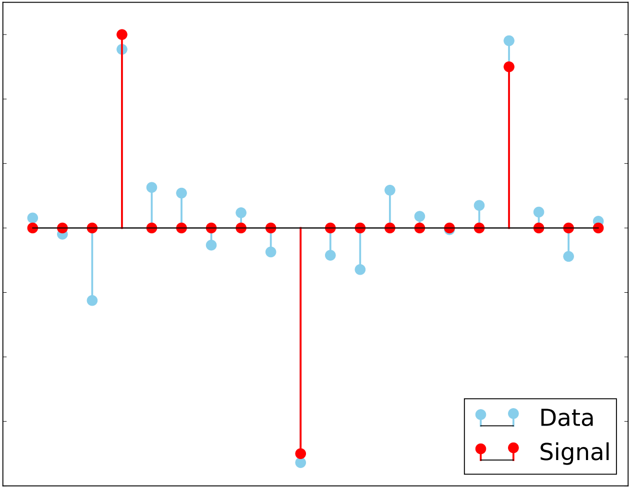
$$A\vec{y} = A\vec{x} + A\vec{z}$$

Why are noise coefficients dense?

If $\vec{\mathbf{z}}$ is Gaussian with mean $\vec{\mu}$ and covariance matrix Σ, then for any A, $A\vec{\mathbf{z}}$ is Gaussian with mean $A\vec{\mu}$ and covariance matrix $A\Sigma A^*$

If A is orthogonal, iid zero mean noise is mapped to iid zero mean noise

Example

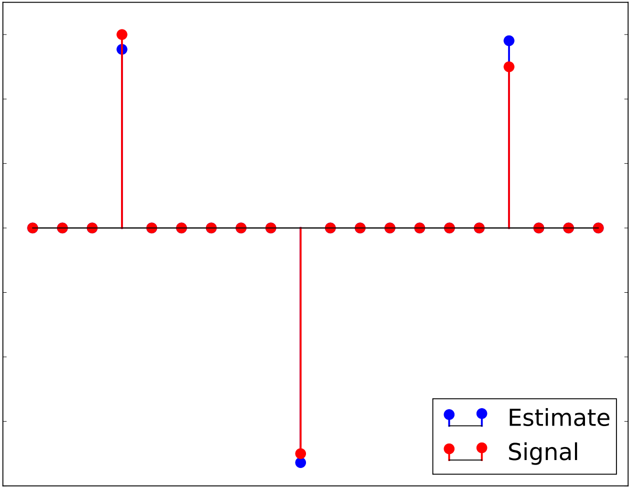


Thresholding

Hard-thresholding operator

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

Denoising via hard thresholding



Denoising via hard thresholding

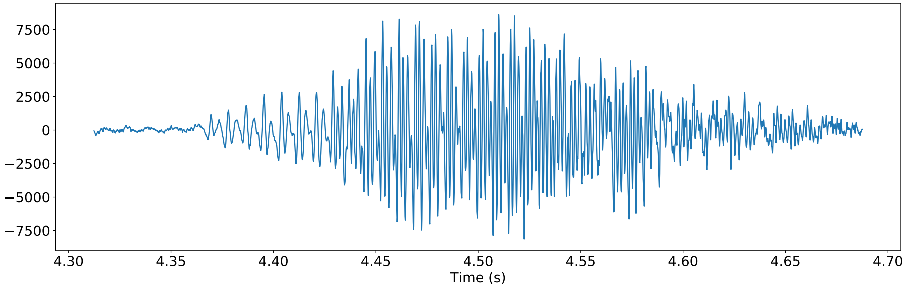
Given data $\vec{y}$ and a sparsifying linear transform A

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

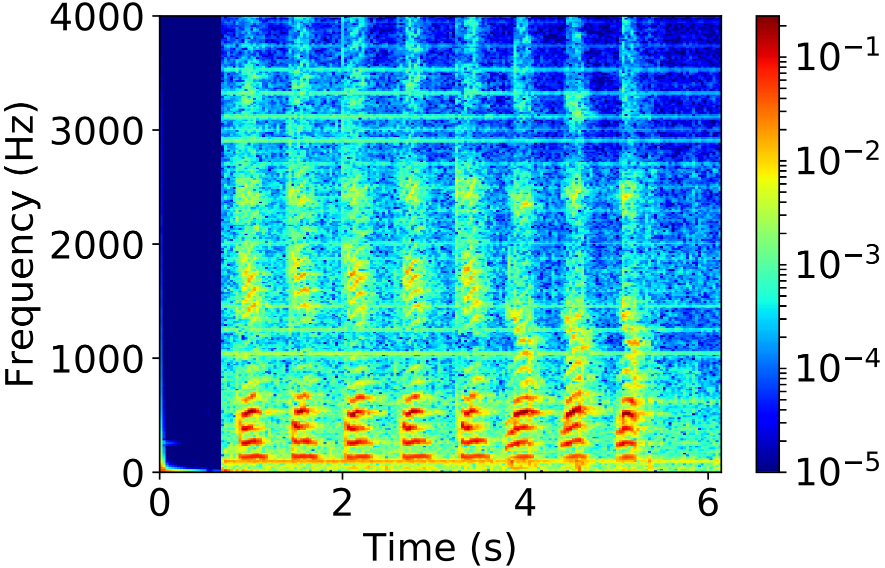
$$\vec{x}_{\text{est}} := L\,\mathcal{H}_\eta\,(A\vec{y})\,,$$

   where L is a left inverse of A

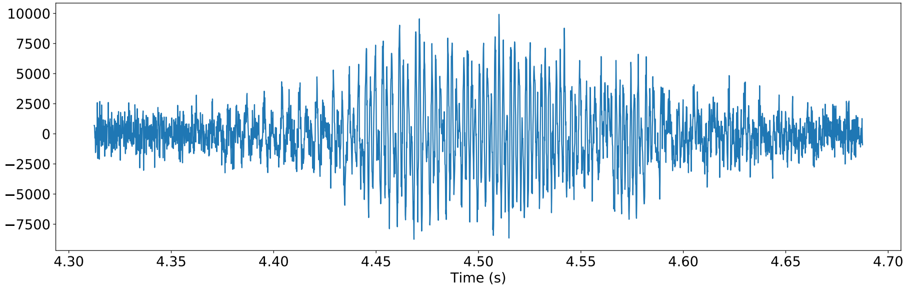
Speech signal



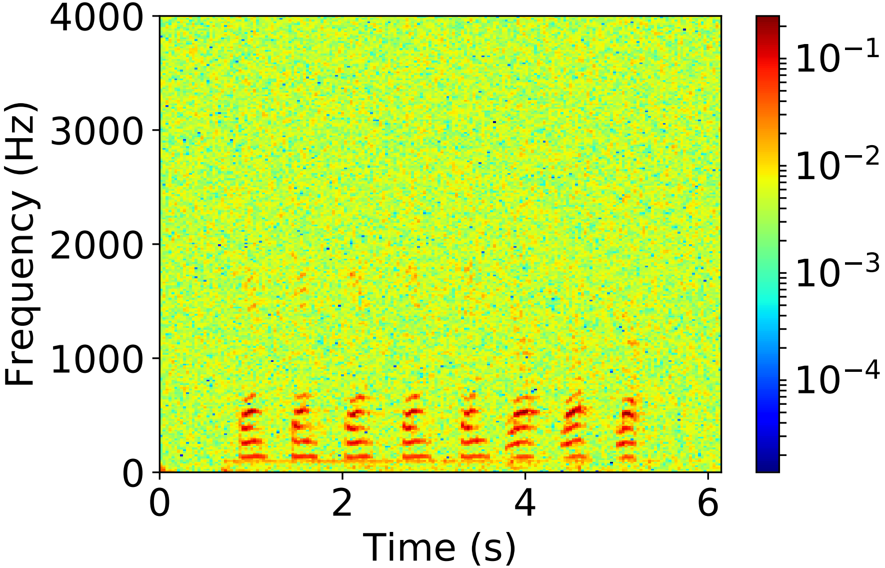
STFT coefficients



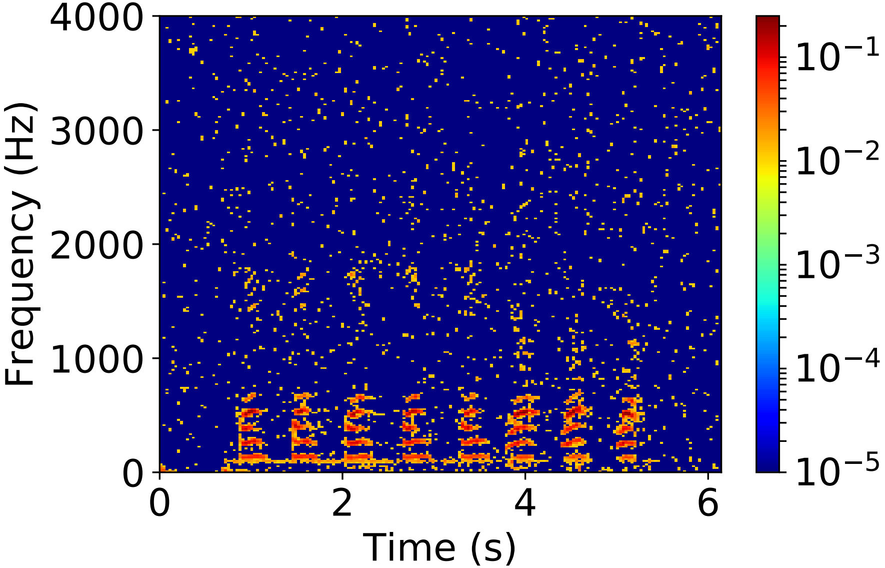
Noisy signal



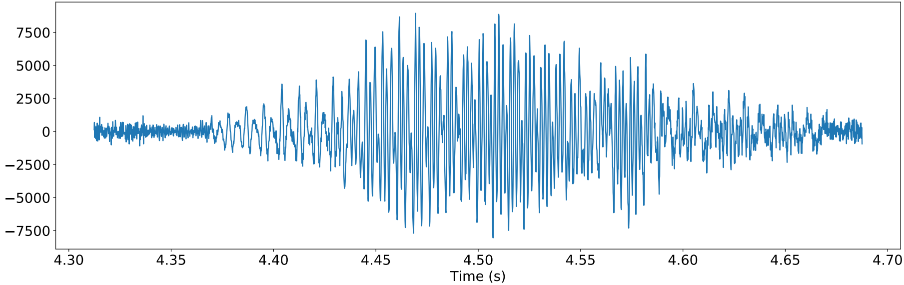
STFT coefficients



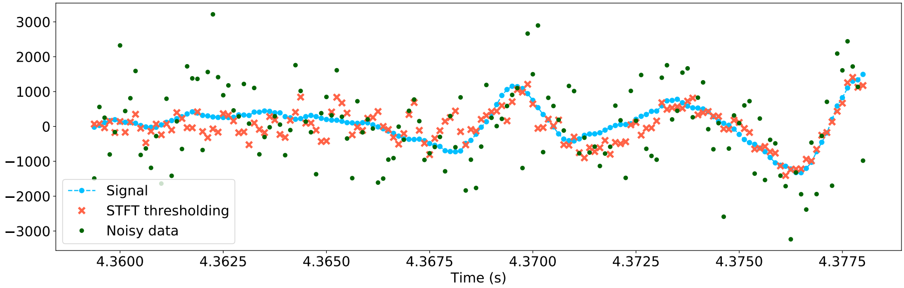
Thresholded STFT coefficients



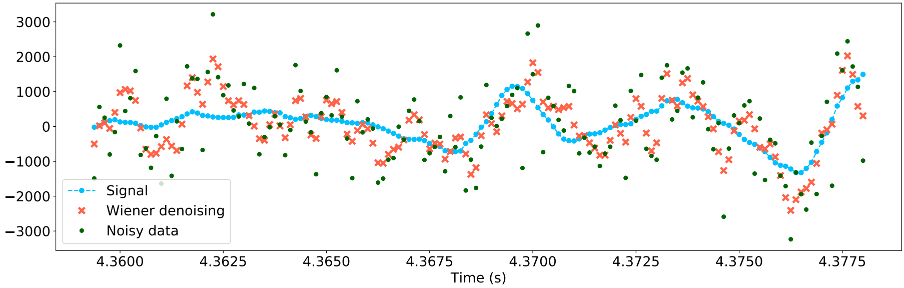
Denoised signal



Denoised signal

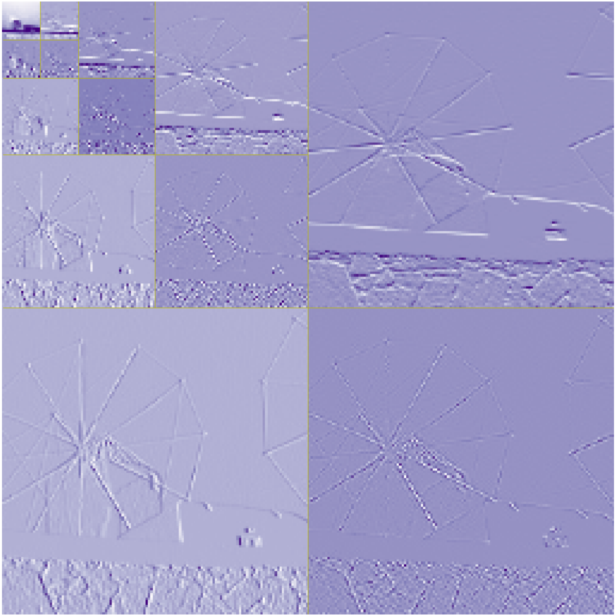


Denoised signal (Wiener filtering)

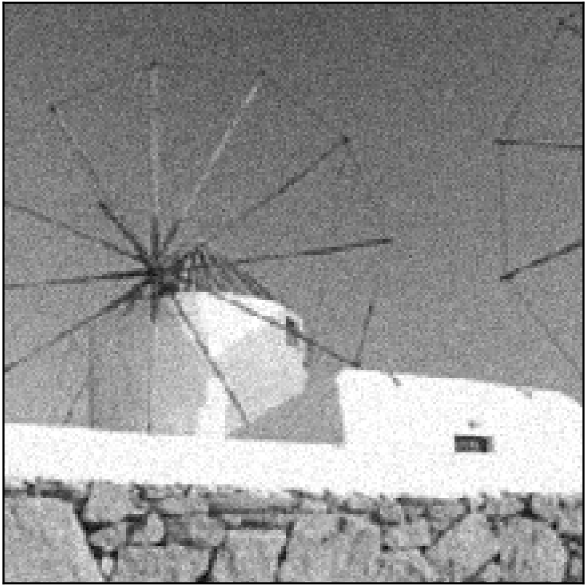


Image

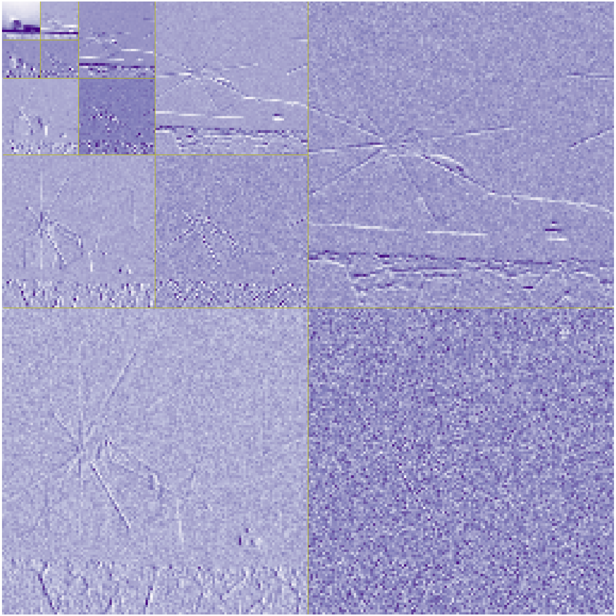
Wavelet coefficients



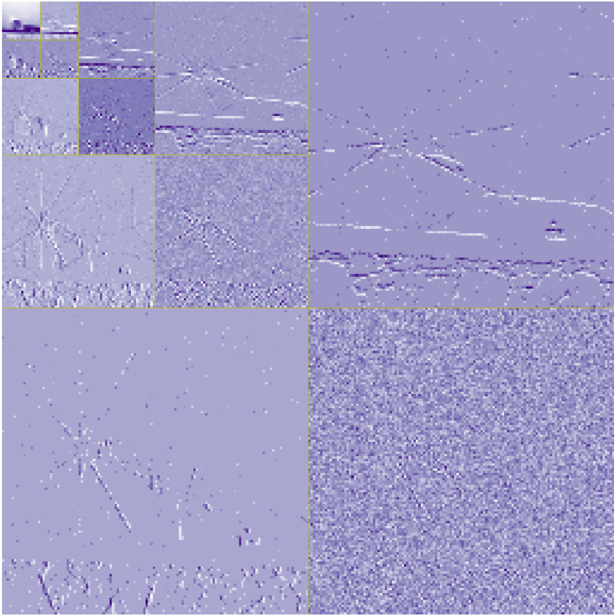
Noisy signal



Wavelet coefficients

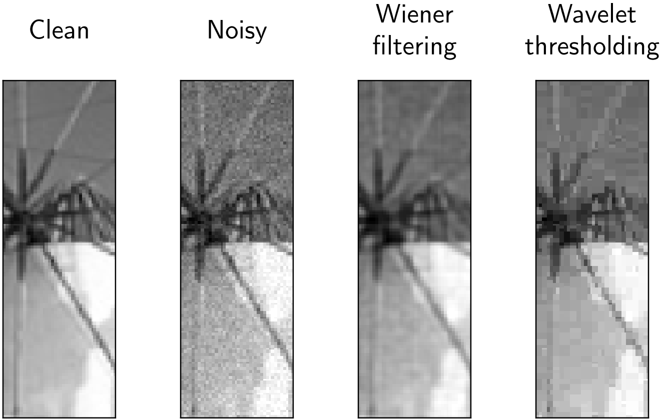


Thresholded wavelet coefficients

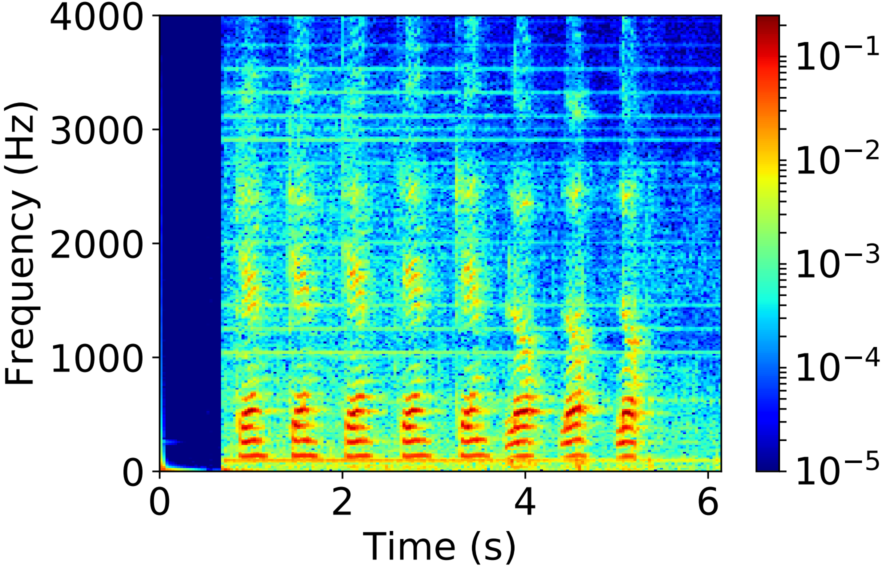


Denoised signal

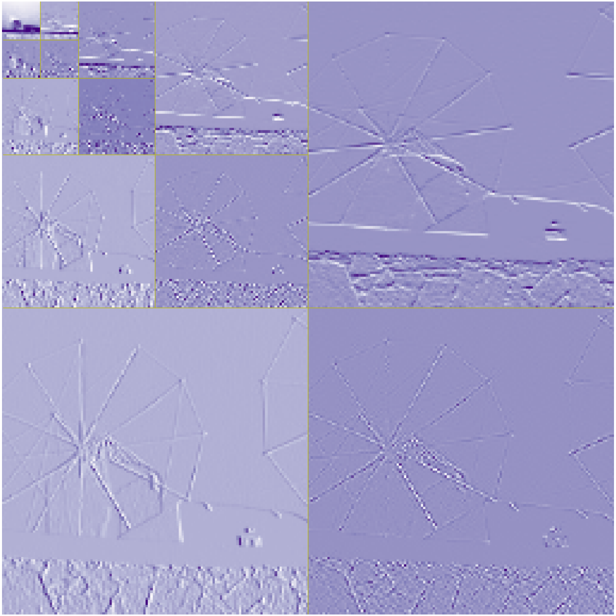
Comparison



Coefficients are structured



Coefficients are structured



Block thresholding

Assumption: Coefficients are *group sparse*, nonzero coefficients *cluster* together

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

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

Denoising via block thresholding

Given data $\vec{y}$ and a sparsifying linear transform A

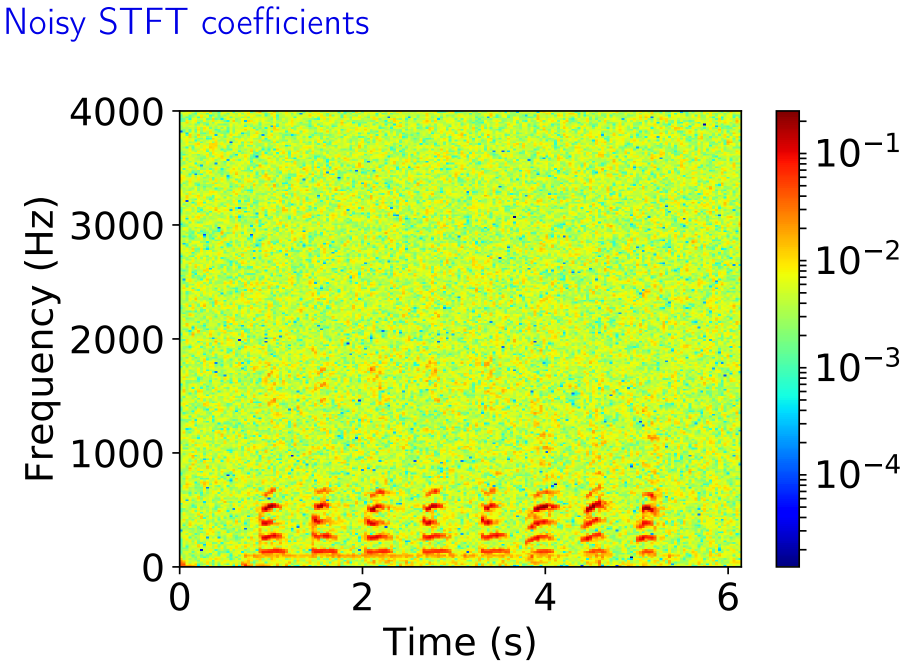
1. Compute coefficients $A\vec{y}$

2. Apply the block-thresholding operator $\mathcal{H}_\eta : \mathbb{C}^n \to \mathbb{C}^n$ to $A\vec{y}$

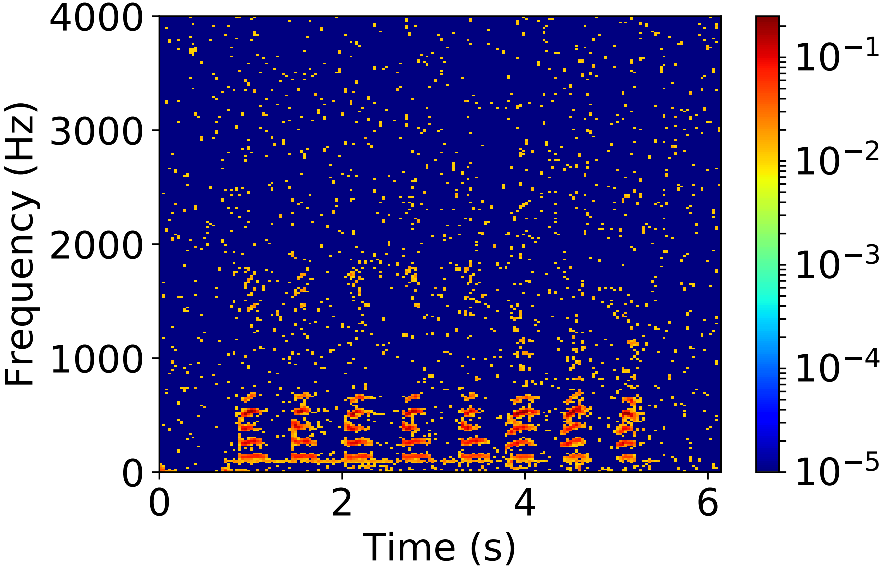
3. Inverting the transform

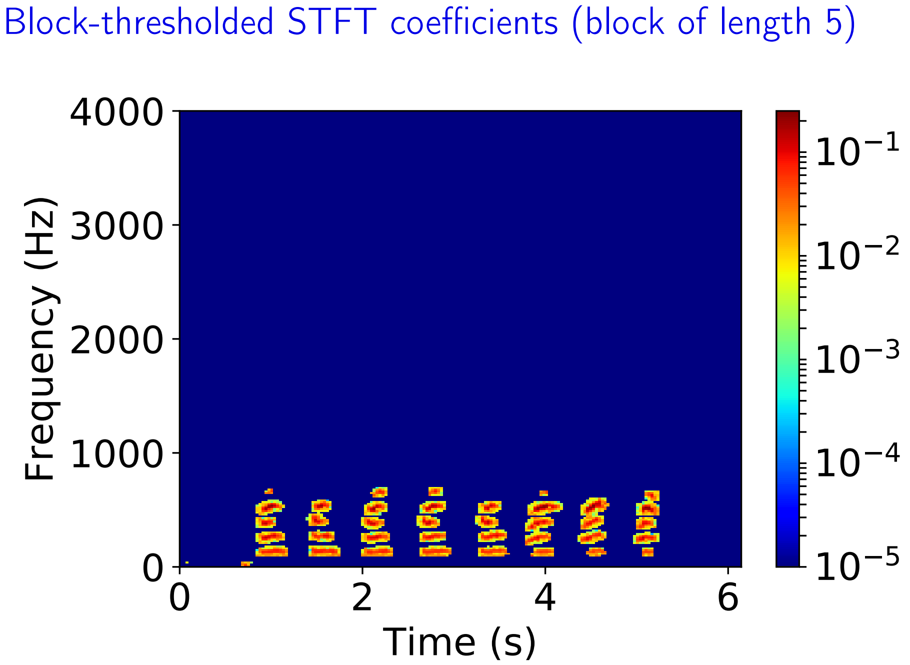
$$\vec{x}_{\text{est}} := L\,\mathcal{B}_\eta\left(A\vec{y}\right),$$

   where L is a left inverse of A

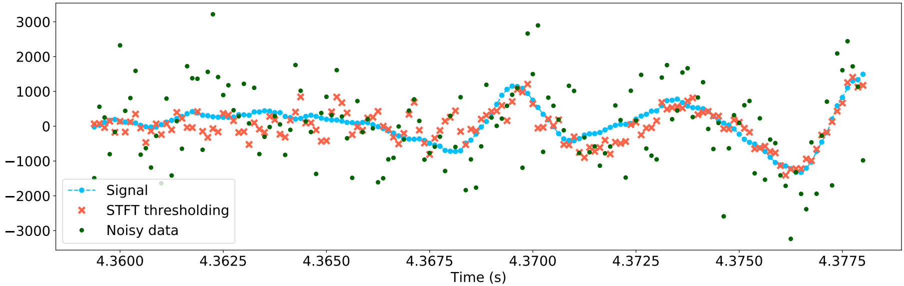

Noisy STFT coefficients
4000
3000
2000
1000
0
Frequency (Hz)
0
2
4
6
Time (s)
10^{-1}
10^{-2}
10^{-3}
10^{-4}

Thresholded STFT coefficients

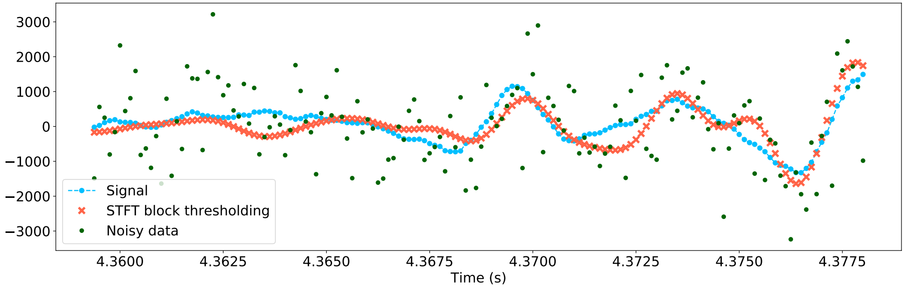



Block-thresholded STFT coefficients (block of length 5)
Frequency (Hz)
4000
3000
2000
1000
0
0
2
4
6
Time (s)
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}

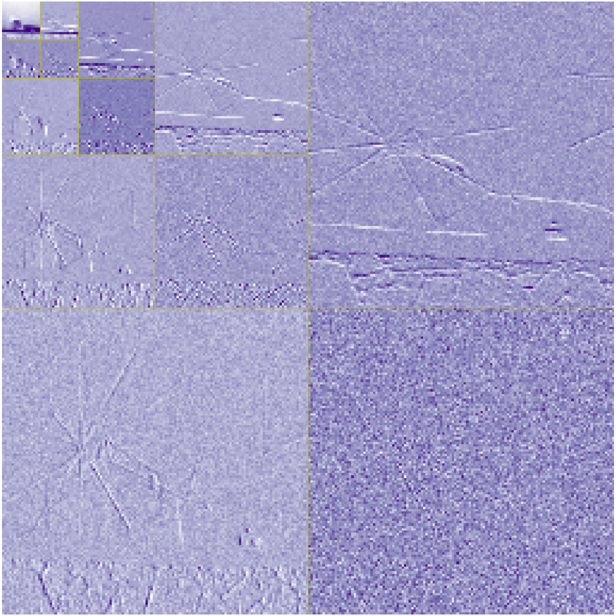
Thresholding



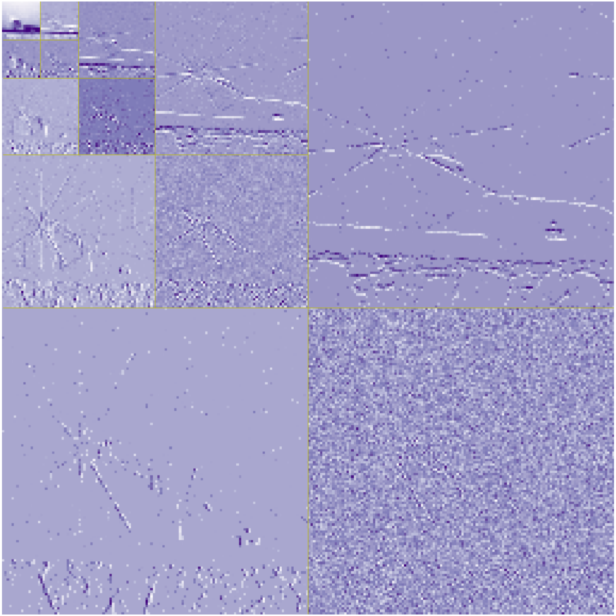
Block thresholding



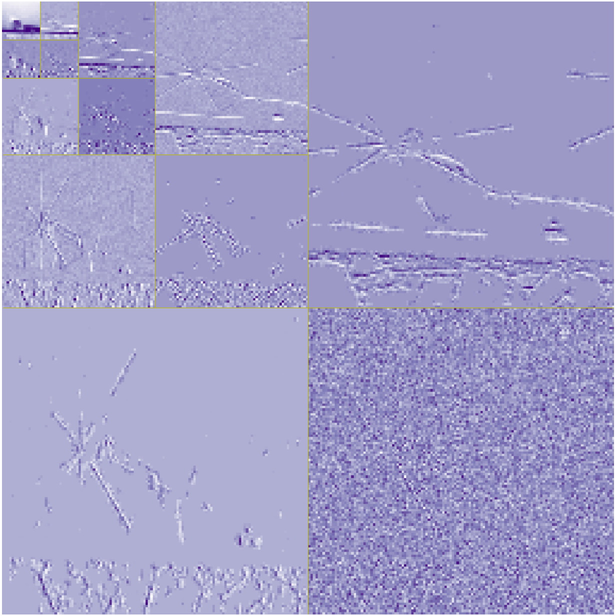
Noisy wavelet coefficients



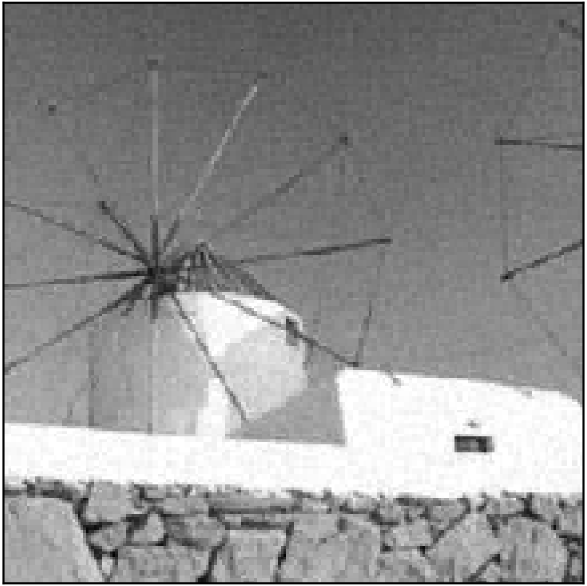
Thresholded wavelet coefficients



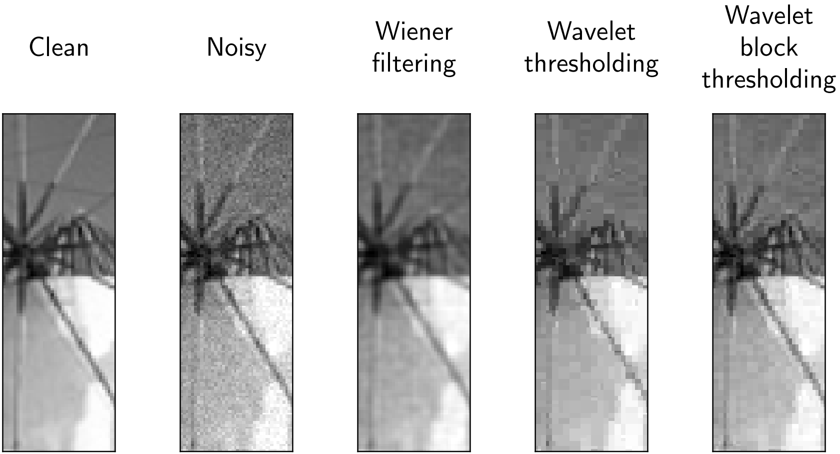
Block thresholded wavelet coefficients



Denoised signal



Comparison


Clean
Noisy
Wiener filtering
Wavelet thresholding
Wavelet block thresholding