Lecture 27: Wavelets to Modern Signal Processing

Foundations of Digital Signal Processing

Outline

- Review of Filter Banks and Wavelets
- Applications of Wavelets
- Applications of Filter Banks and Time-Frequency Representations
- Modern Signal Processing: Vectors and Matrices
- Modern Signal Processing: Compressive Sensing
- Modern Signal Processing: Diagonalization
- Modern Signal Processing: Graph Signal Processing

News

- **Schedule / Plan**
  - ~~*Tomorrow:, Nov. 16 Homework #10*~~
  - ~~*Tuesday, Nov. 19: Coding Assignment #6*~~
  - ~~*Next Week: No Due Dates (except Tuesday)*~~
  - *Thursday, Nov. 29th:* *Homework #11*
  - *Tuesday, Dec. 4th:* *Exam #3*
  - *Wednesday, Dec. 5th:* *Coding Assignment #7 (short)*
  - *Wednesday, Dec. 12th:* *Final Exam*
  - *Friday, Dec. 14th:* *EEE5502 Reports Due*

Lecture 27: Wavelets to Modern Signal Processing

Foundations of Digital Signal Processing

Outline

- **Review of Filter Banks and Wavelets**
- Applications of Wavelets
- Applications of Filter Banks and Time-Frequency Representations
- Modern Signal Processing: Vectors and Matrices
- Modern Signal Processing: Compressive Sensing
- Modern Signal Processing: Diagonalization
- Modern Signal Processing: Graph Signal Processing

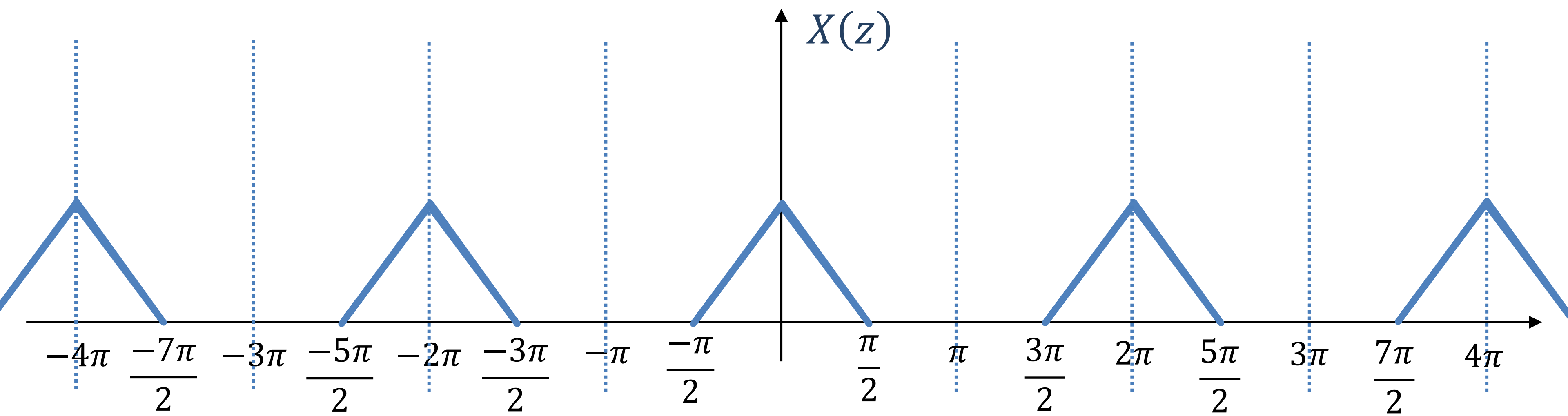
Filter Banks

- **Question: Can we generalize perfect reconstruction?**

$$X(z) \rightarrow \boxed{\downarrow 2} \rightarrow Y(z)$$

DTFT: $Y(\omega) = X(\omega/2) + X\big((\omega - 2\pi)/2\big)$

Z-Domain: $Y(\omega) = X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)$



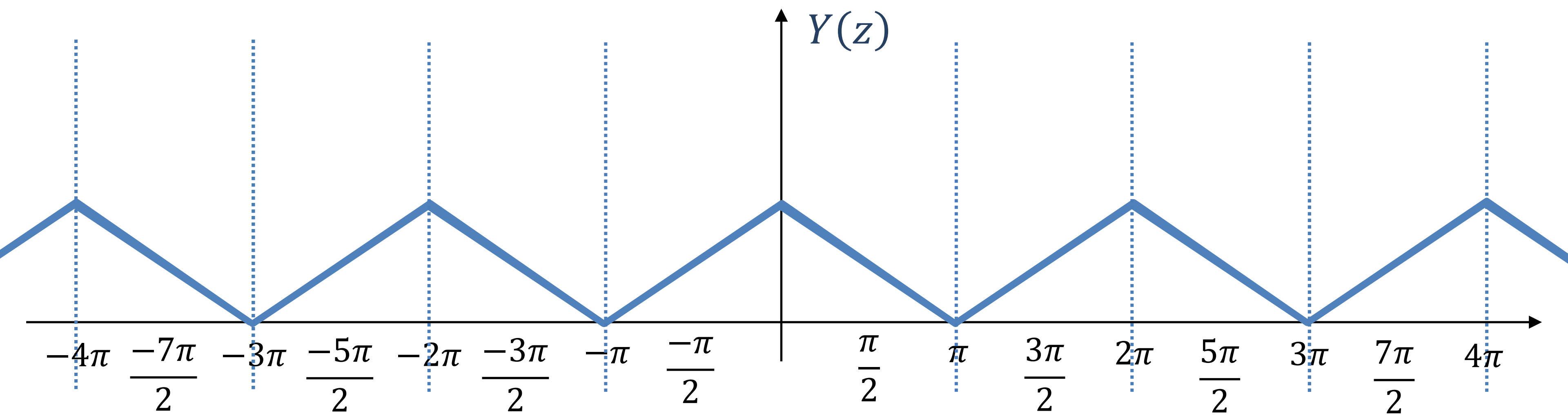
Filter Banks

- **Question: Can we generalize perfect reconstruction?**

$$X(z) \rightarrow \boxed{\downarrow 2} \rightarrow Y(z)$$

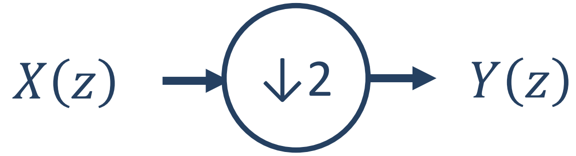
DTFT: $Y(\omega) = X(\omega/2) + X\big((\omega - 2\pi)/2\big)$

Z-Domain: $Y(z) = X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)$



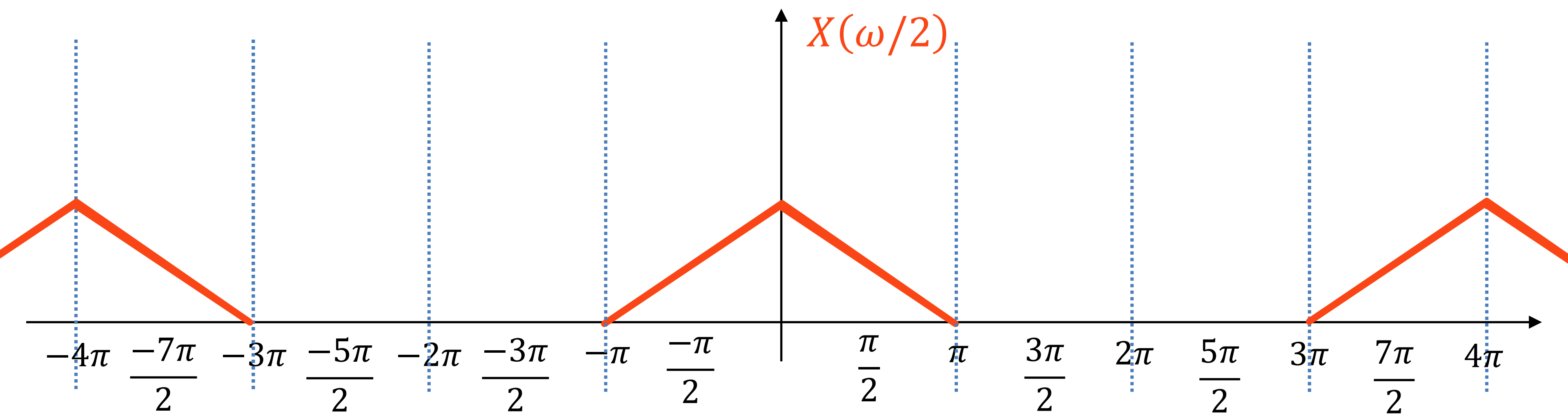
Filter Banks

- **Question: Can we generalize perfect reconstruction?**



DTFT: $Y(\omega) = X(\omega/2) + X\big((\omega - 2\pi)/2\big)$

Z-Domain: $Y(z) = X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)$



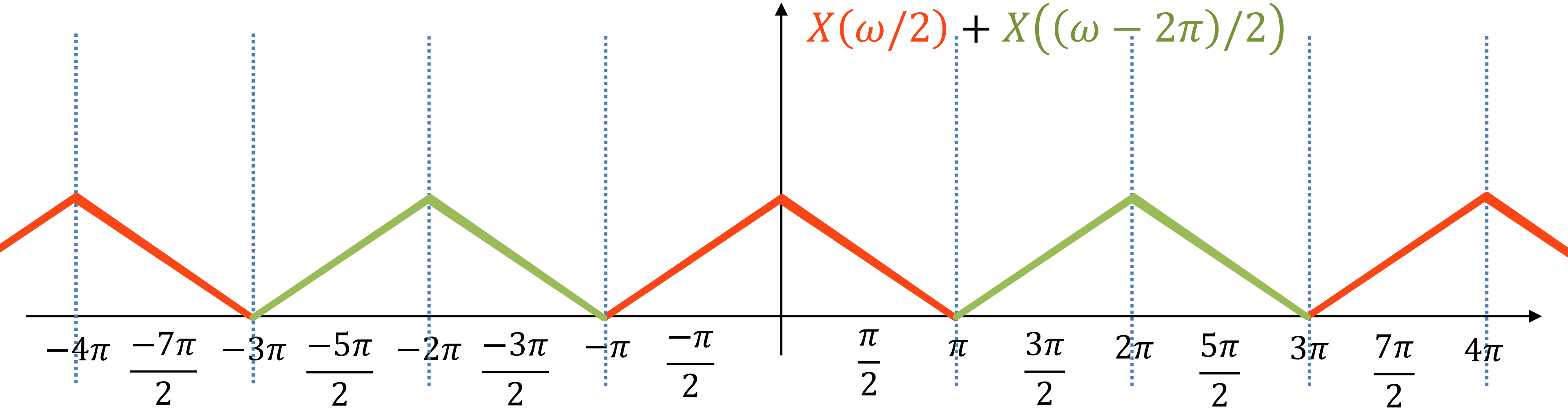
Filter Banks

- **Question: Can we generalize perfect reconstruction?**

$$X(z) \rightarrow \boxed{\downarrow 2} \rightarrow Y(z)$$

DTFT: $Y(\omega) = X(\omega/2) + X\big((\omega - 2\pi)/2\big)$

Z-Domain: $Y(z) = X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)$

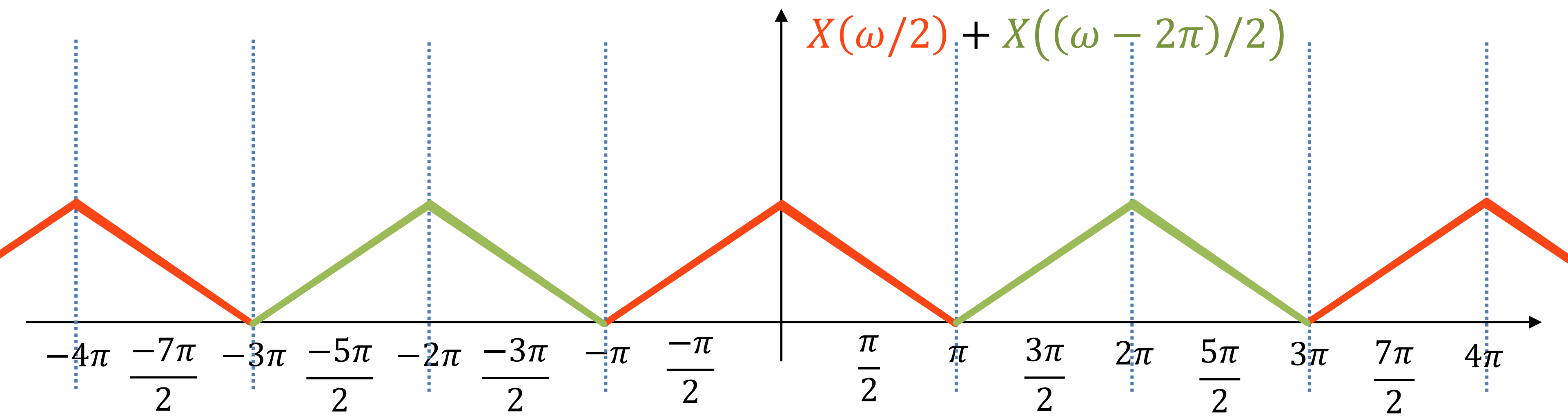


- **Question: Can we generalize perfect reconstruction?**

$$X(z) \rightarrow \downarrow M \rightarrow Y(z)$$

DTFT: $$Y(\omega) = \sum_{m=0}^{M} X\big((\omega - 2\pi m)/M\big)$$

Z-Domain: $$Y(z) = \sum_{m=0}^{M} X\left(z^{1/M} e^{-\frac{2\pi}{M}m}\right)$$



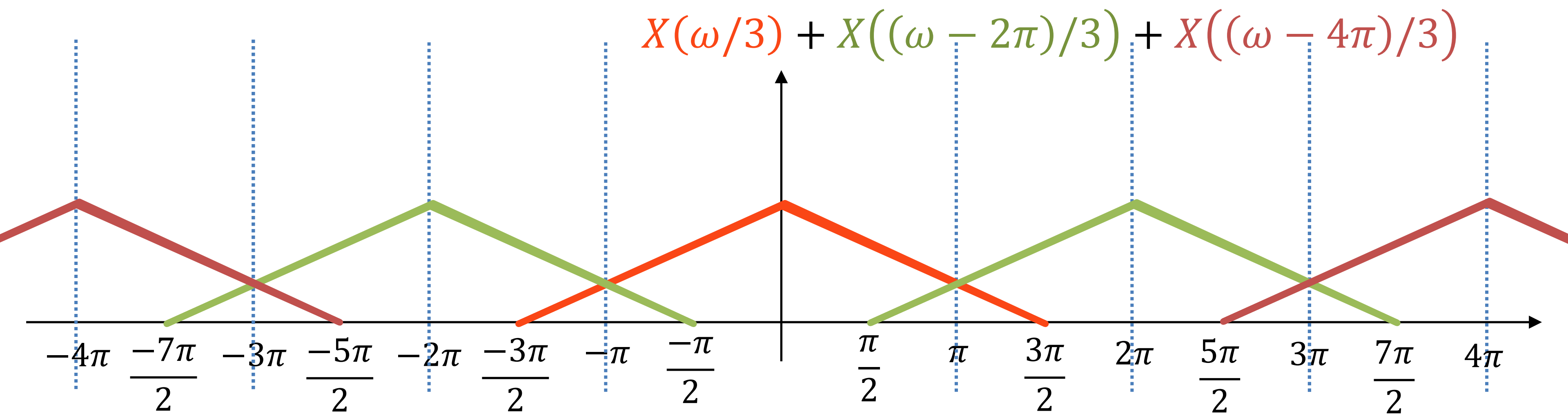
Filter Banks

- **Question: Can we generalize perfect reconstruction?**

$$X(z) \rightarrow \boxed{\downarrow M} \rightarrow Y(z)$$

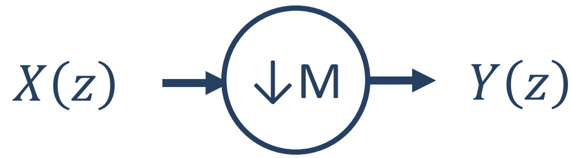
DTFT: $$Y(\omega) = \sum_{m=0}^{M} X\big((\omega - 2\pi m)/M\big)$$

Z-Domain: $$Y(z) = \sum_{m=0}^{M} X\left(z^{1/M} e^{-\frac{2\pi}{M} m}\right)$$



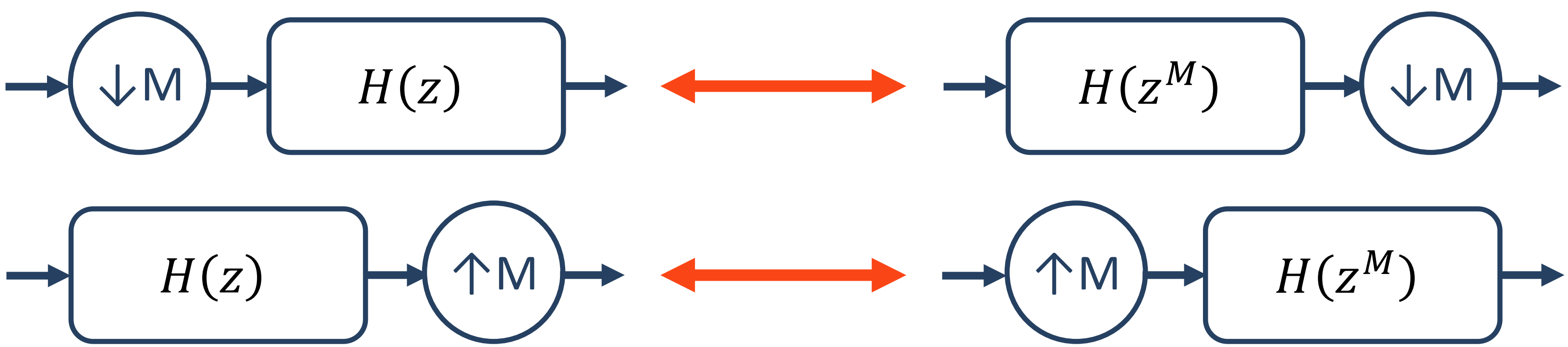
Noble Properties

Noble Properties

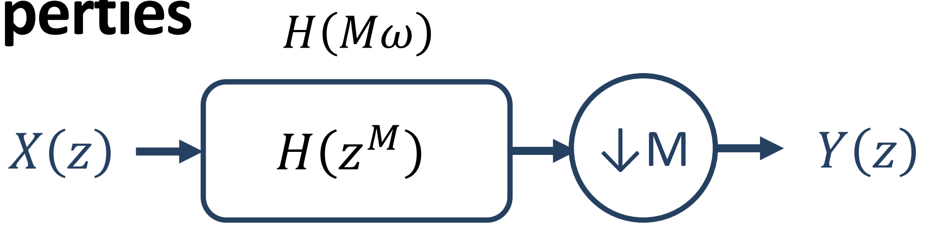


DTFT: $$Y(\omega) = \sum_{m=0}^{M} X\big((\omega - 2\pi m)/M\big)$$

Z-Domain: $$Y(z) = \sum_{m=0}^{M} X\left(z^{1/M} e^{-\frac{2\pi}{M}m}\right)$$

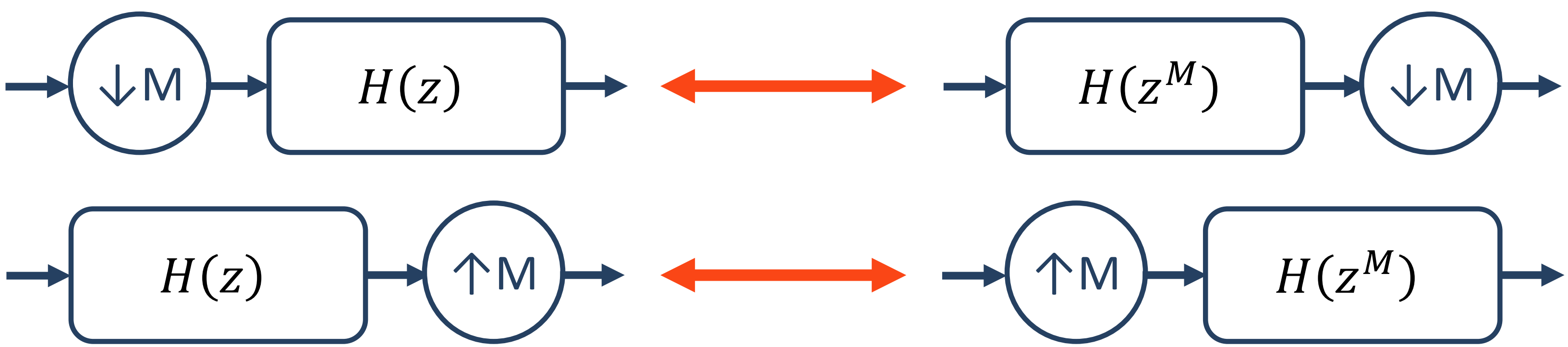


Noble Properties



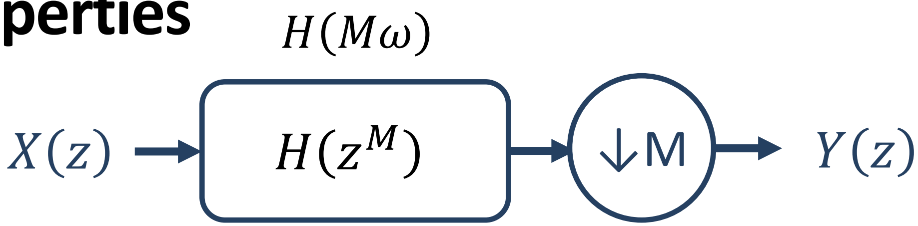
DTFT: $$Y(\omega) = \sum_{m=0}^{M} H(M(\omega - 2\pi m)/M)X\big((\omega - 2\pi m)/M\big)$$

Z-Domain: $$Y(z) = \sum_{m=0}^{M} H\left(z^{M/M}e^{-\frac{M2\pi}{M}m}\right)X\left(z^{M}e^{-\frac{2\pi}{M}m}\right)$$



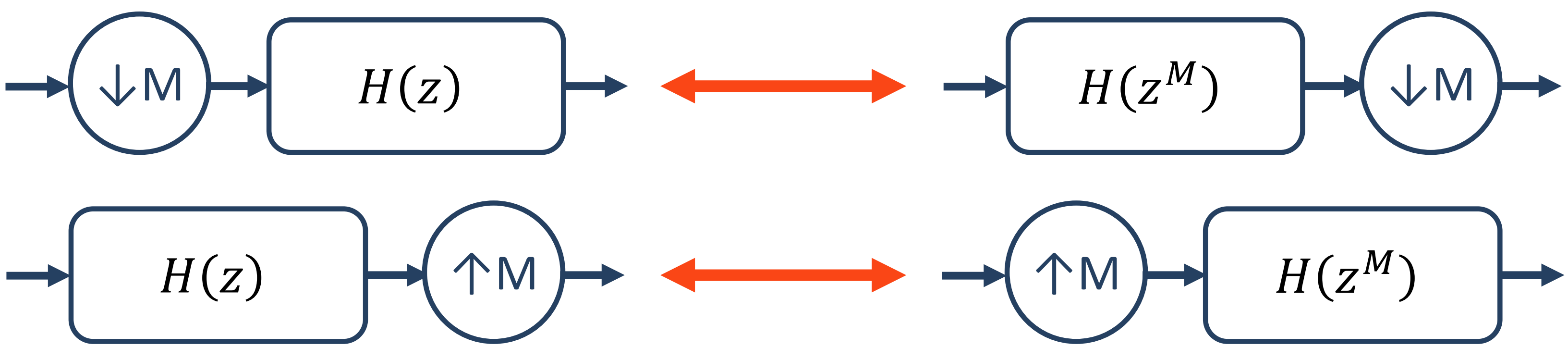
Noble Properties

Noble Properties



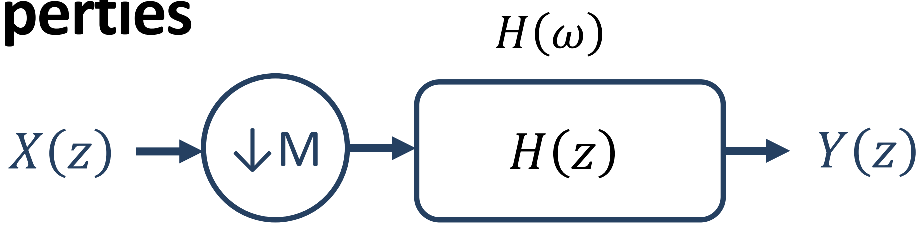
DTFT: $$Y(\omega) = \sum_{m=0}^{M} H(\omega) X\big((\omega - 2\pi m)/M\big)$$

Z-Domain: $$Y(z) = \sum_{m=0}^{M} H(z) X\left(z^M e^{-\frac{2\pi}{M}m}\right)$$



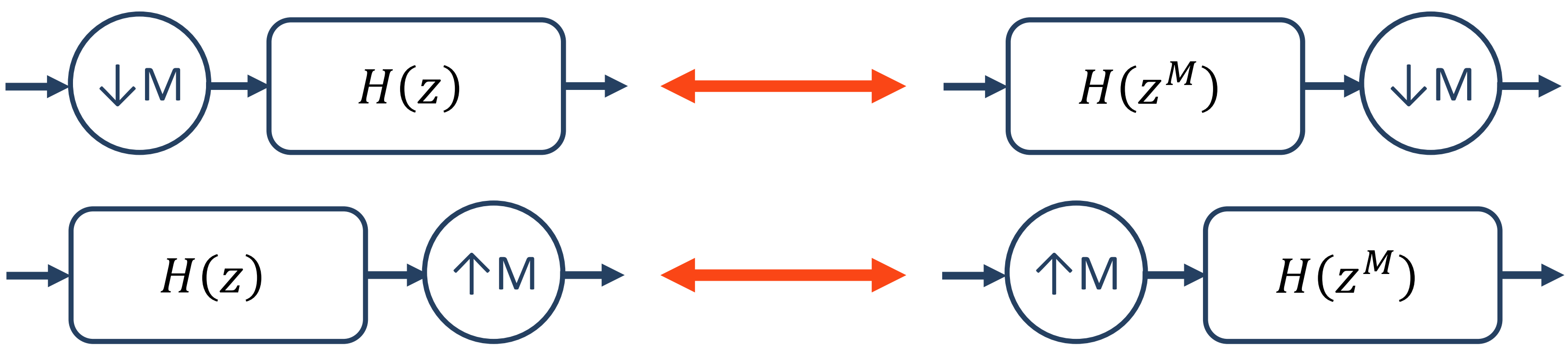
Noble Properties

- **Noble Properties**



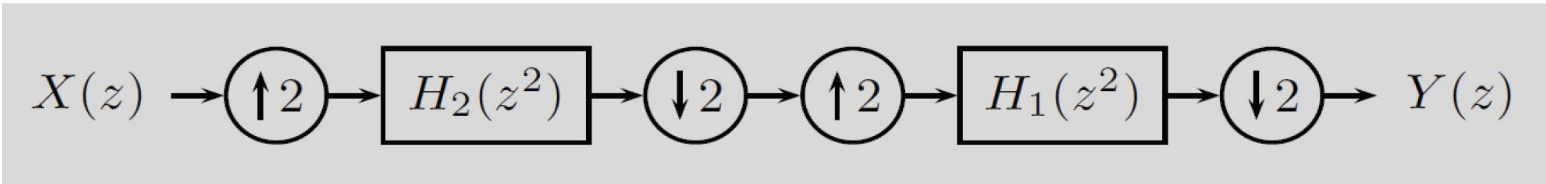
DTFT: $$Y(\omega) = H(\omega)\sum_{m=0}^{M} X\big((\omega - 2\pi m)/M\big)$$

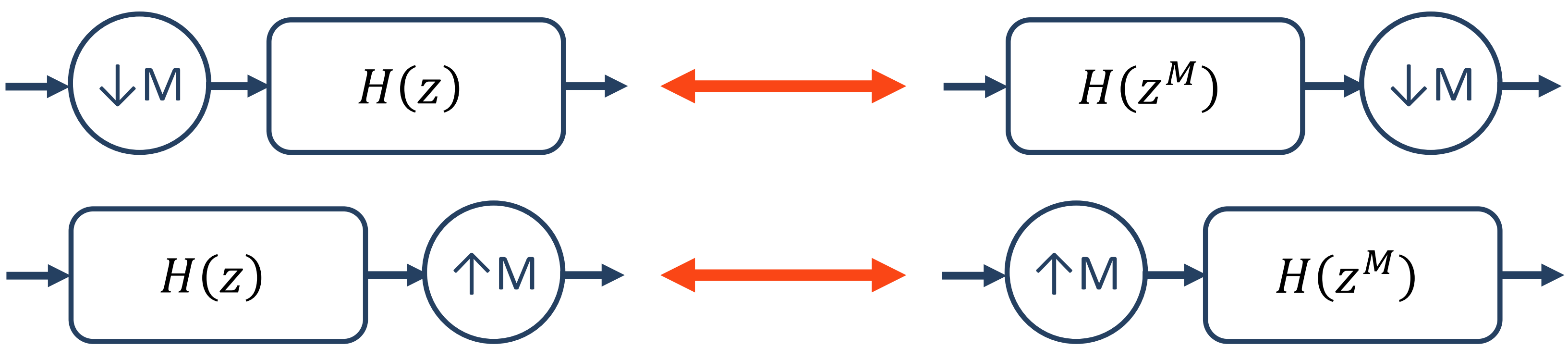
Z-Domain: $$Y(z) = H(z)\sum_{m=0}^{M} X\left(z^{M} e^{-\frac{2\pi}{M}m}\right)$$



Noble Properties

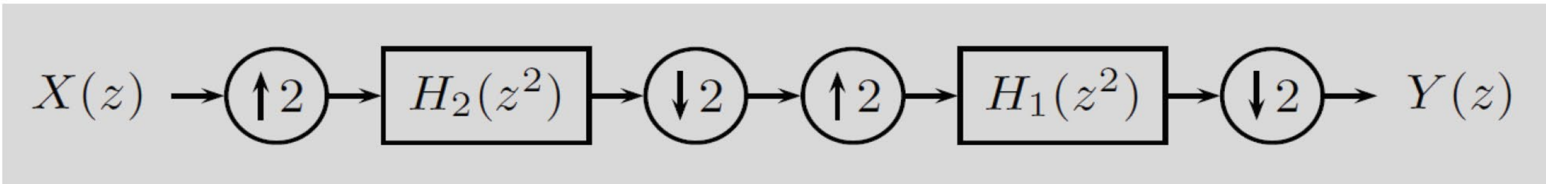
- **Example: Simplify the following**

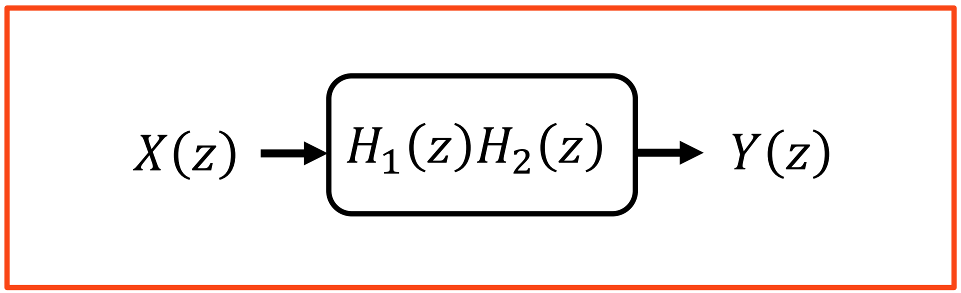


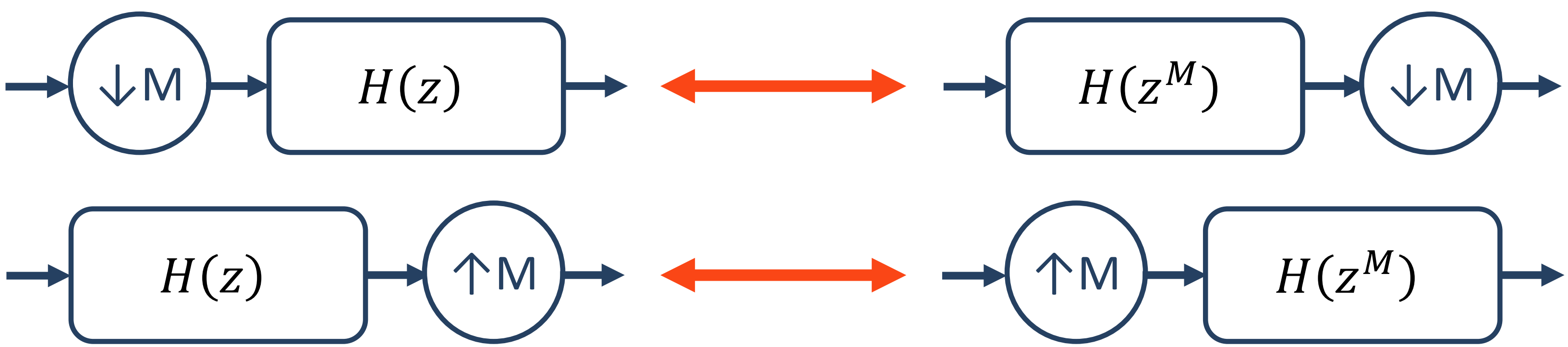


Noble Properties

- **Example: Simplify the following**





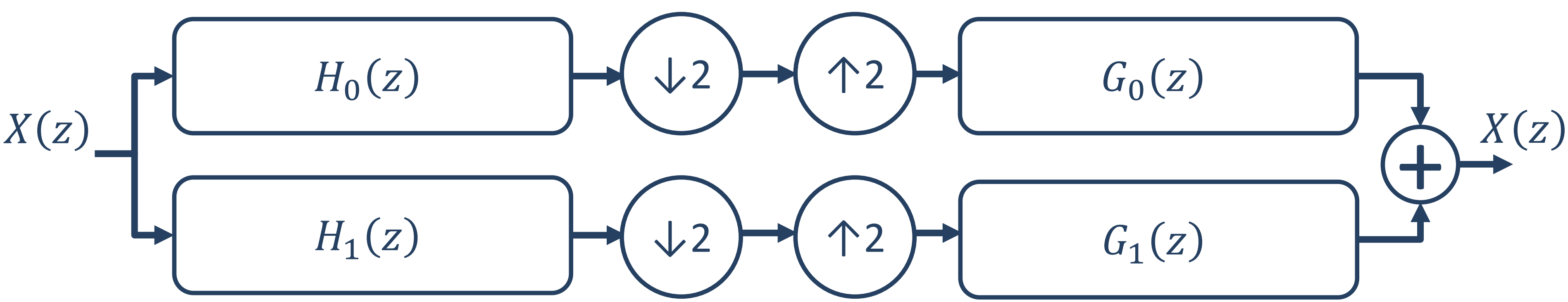


Filter Banks

- **Perfect Reconstruction**

$$X(z) = \frac{1}{2}X(z)\ [H_0(z)G_0(z) + H_1(z)G_1(z)] + \frac{1}{2}X(-z)[H_0(-z)G_0(z) + H_1(-z)G_1(z)]$$

  - **Option 1:** Alias canceling
    - ◊ $H_0(z)G_0(z) + H_1(z)G_1(z) = 2$
    - ◊ $H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0$

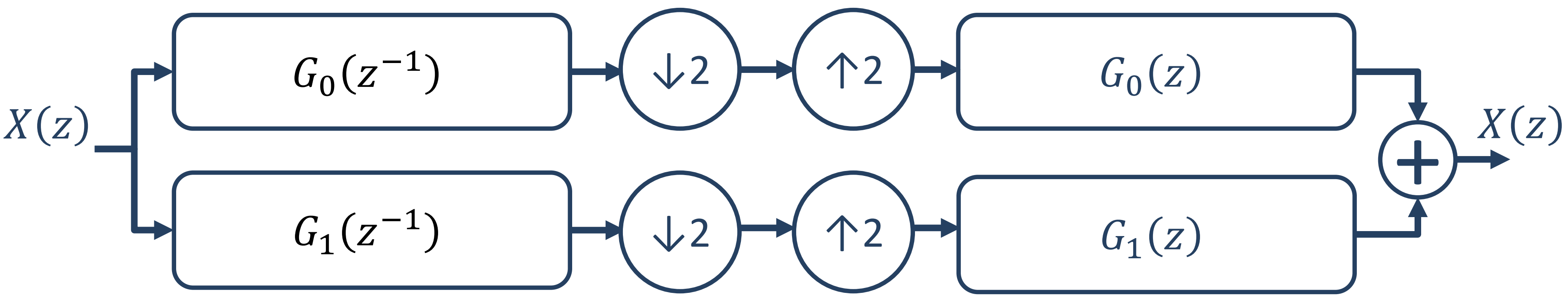


Filter Banks

Perfect Reconstruction

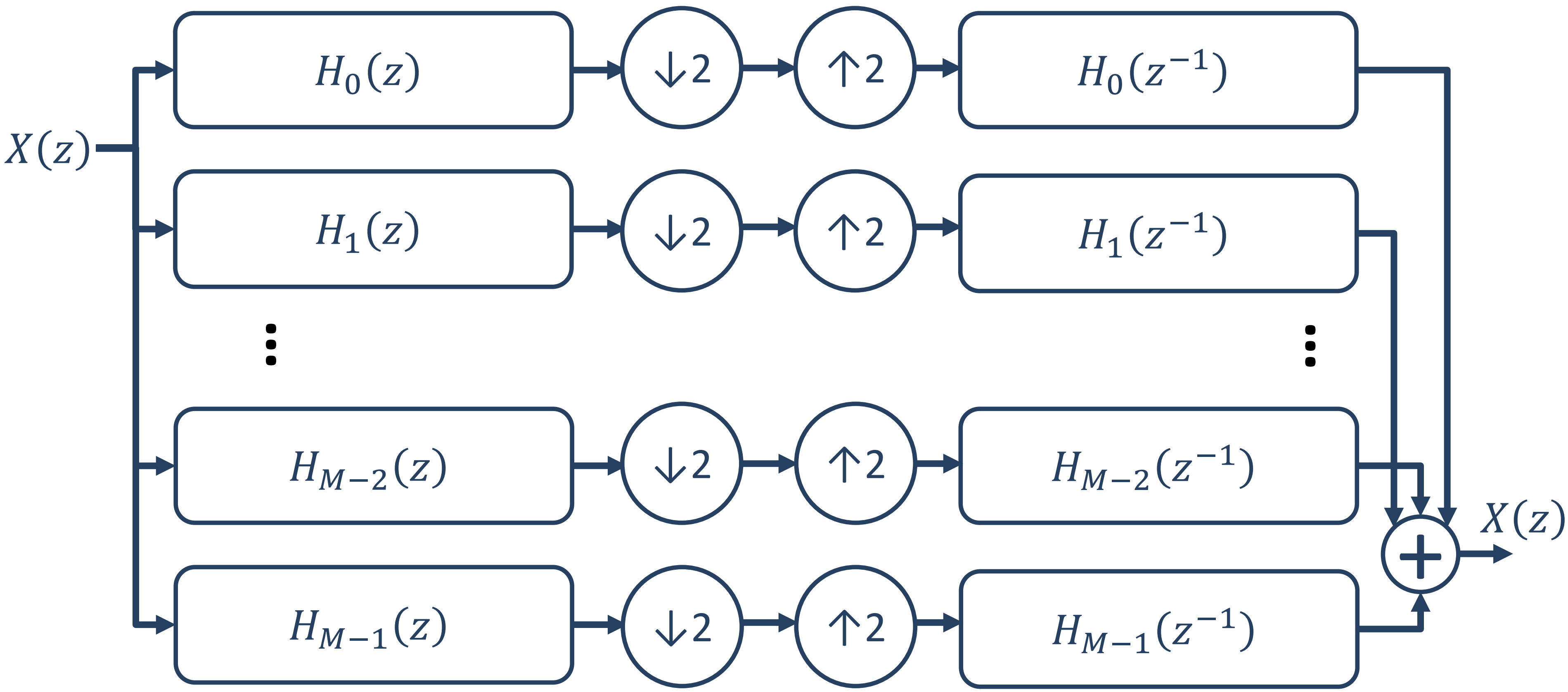
$$X(z) = \frac{1}{2}X(z)\ \ [G_0(z^{-1})G_0(z) \quad + G_1(z^{-1})G_1(z)]$$
$$+\frac{1}{2}X(-z)[G_0(-z^{-1})G_0(z) + G_1(-z^{-1})G_1(z)]$$

- **Option 2:** Orthogonal Filter Bank
  - ◊ $G_0(z)G_0(z^{-1}) + G_0(-z)G_0(-z^{-1}) = 2$
  - ◊ $G_1(z)G_1(z^{-1}) + G_1(-z)G_1(-z^{-1}) = 2$
  - ◊ $G_0(z)G_1(z^{-1}) + G_0(-z)G_1(-z^{-1}) = 0$



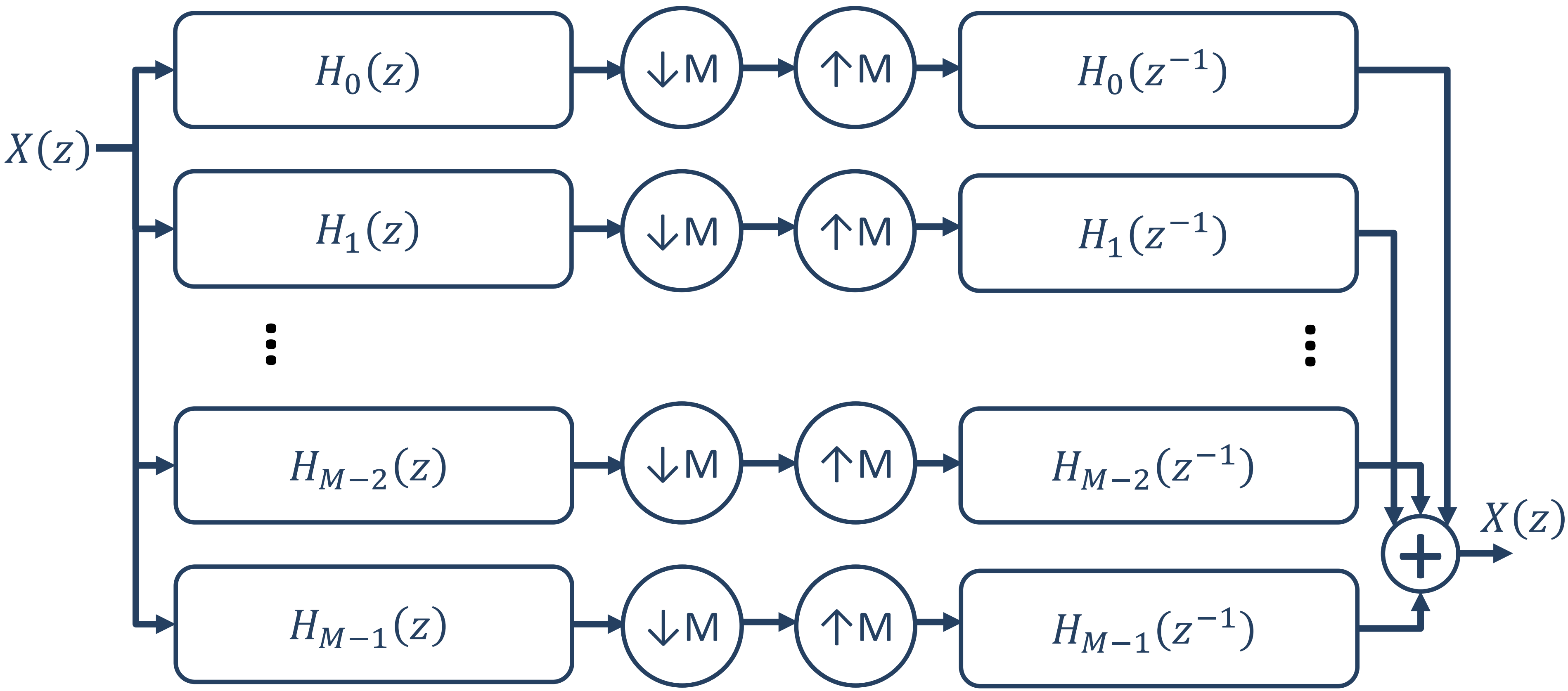
Multi-Channel Filter Banks

- **Question: Can we generalize perfect reconstruction?**
  - $G_m(z)G_m(z^{-1}) + G_m(-z)G_m(-z^{-1}) = 2$ for all m
  - $G_m(z)G_k(z^{-1}) + G_m(-z)G_k(-z^{-1}) = 0$ for all m, k



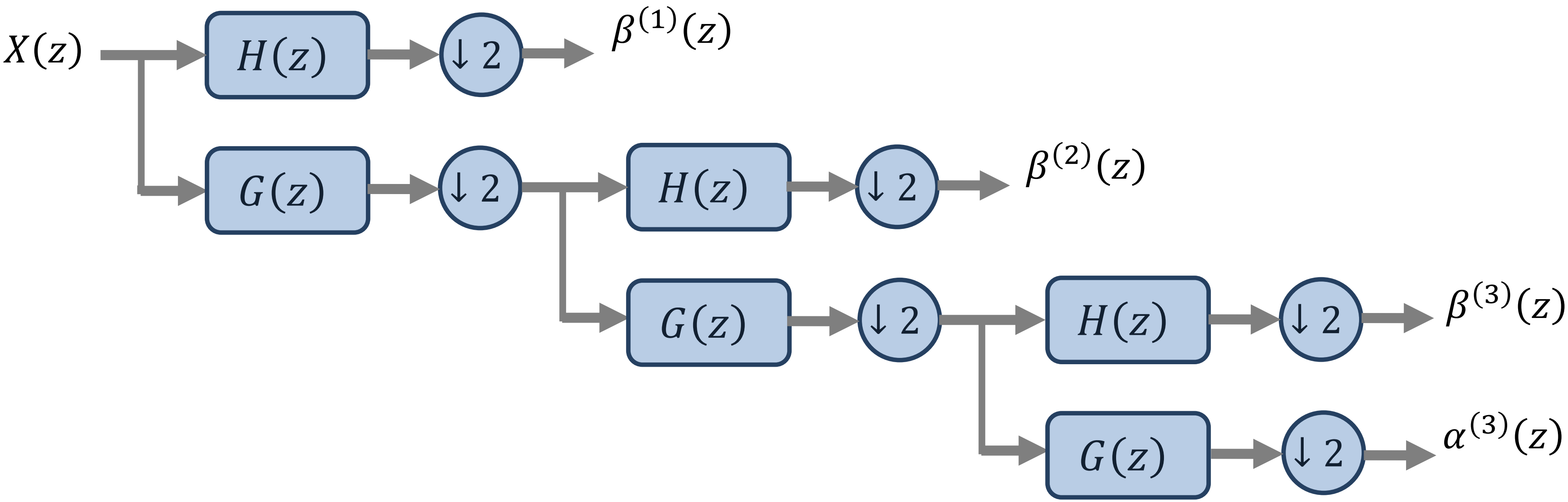
Multi-Channel Filter Banks

- **Question: Can we generalize perfect reconstruction?**
  - $\sum_{k=0}^{M-1} G_m\left(e^{j\frac{2\pi}{M}k}\, z\right) G_m\left(e^{j\frac{2\pi}{M}k}\, z^{-1}\right) = M$ for all m
  - $\sum_{k=0}^{M-1} G_m\left(e^{j\frac{2\pi}{M}k}\, z\right) G_k\left(e^{j\frac{2\pi}{M}k}\, z^{-1}\right) = 0$ for all m, k



Wavelets / Sub-band coding

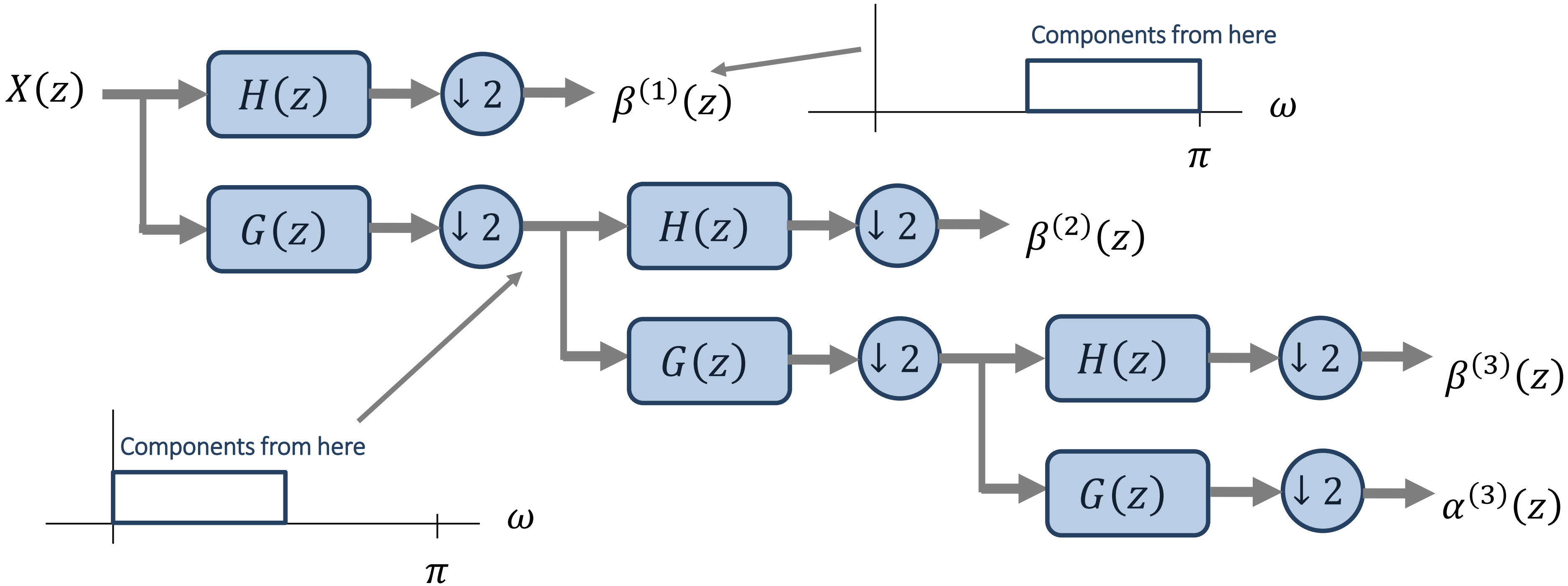
- **Consider this new [analysis] filter bank:** What is going on here?



- **Assume H^* is a half-band high pass filter**
- **Assume G^* is a half-band low pass filter**

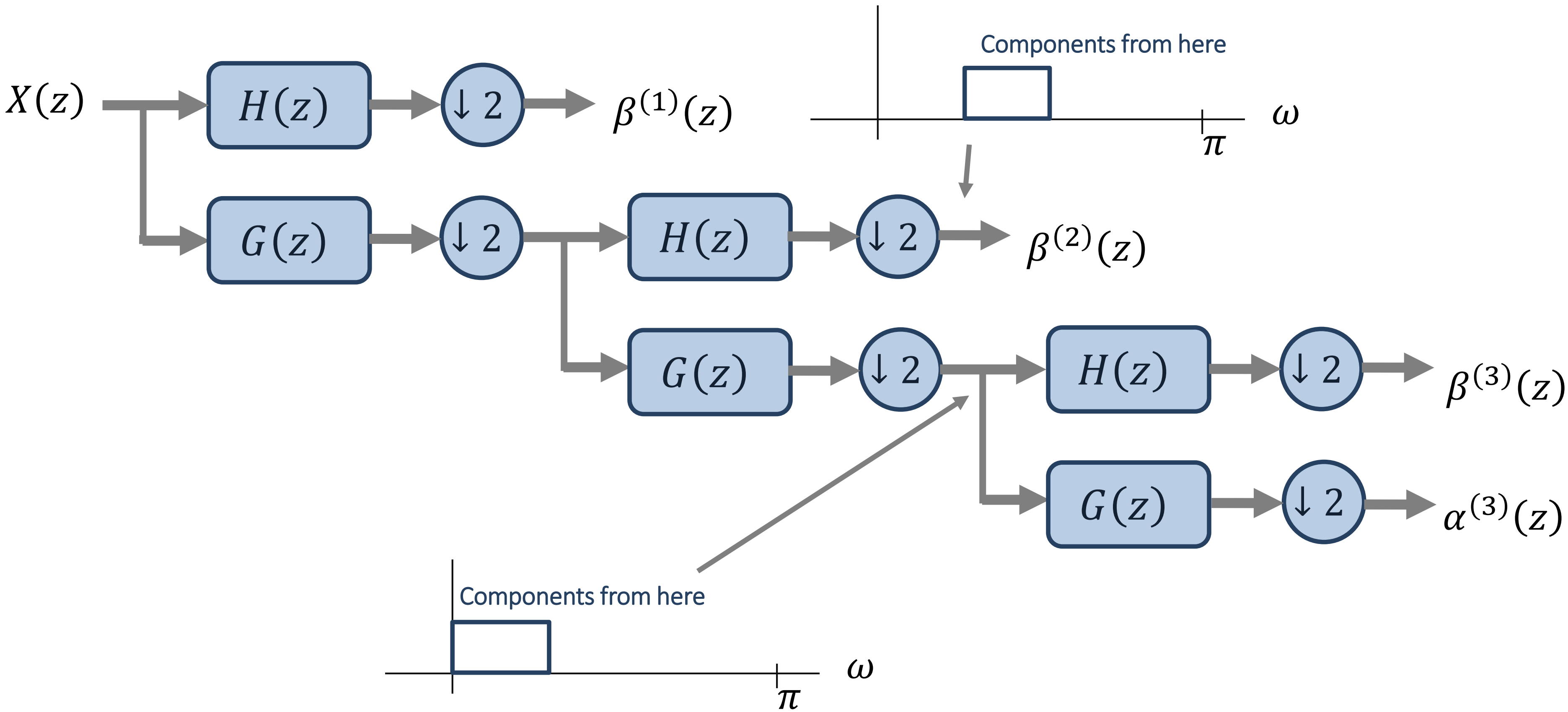
Wavelets / Sub-band coding

- **Consider this new [analysis] filter bank: What is going on here?**



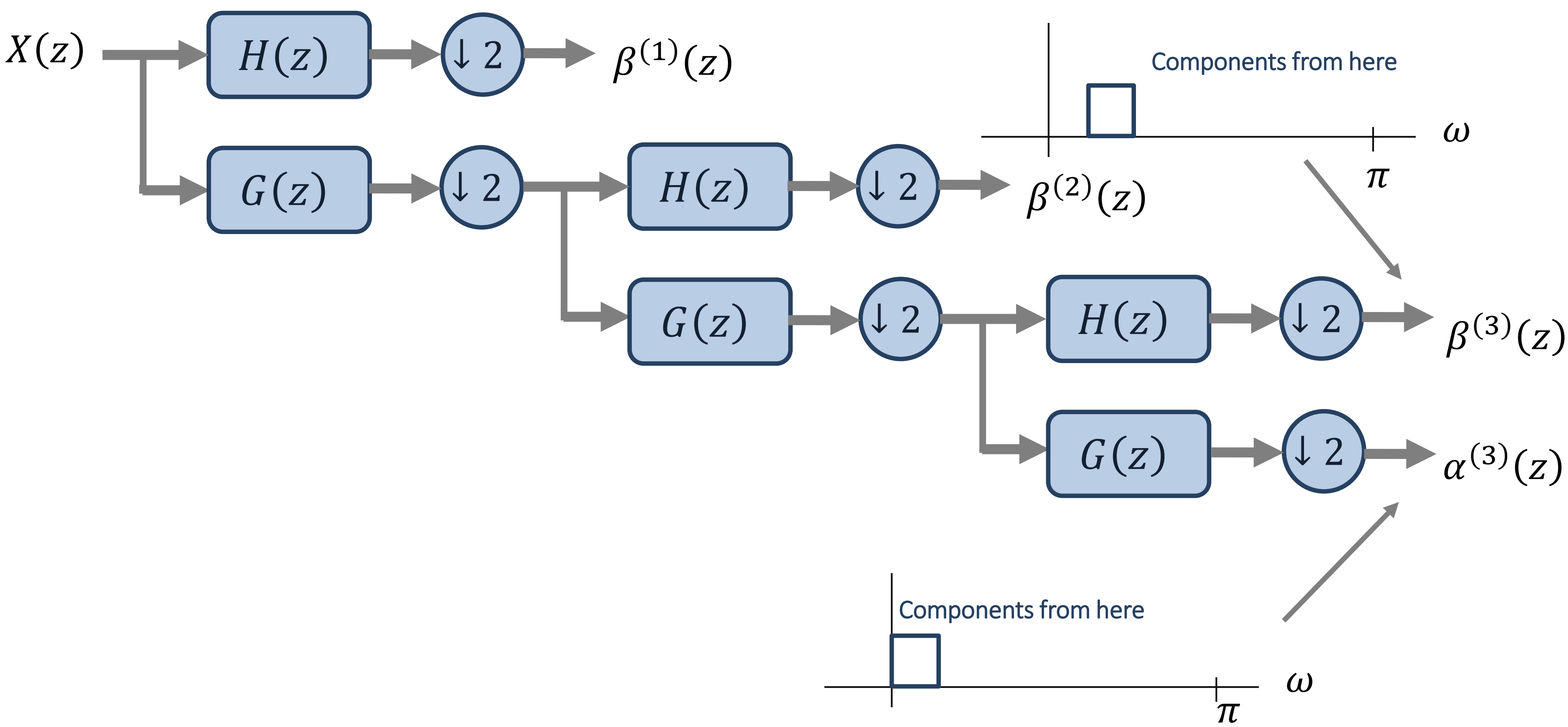
Wavelets / Sub-band coding

- **Consider this new [analysis] filter bank: What is going on here?**



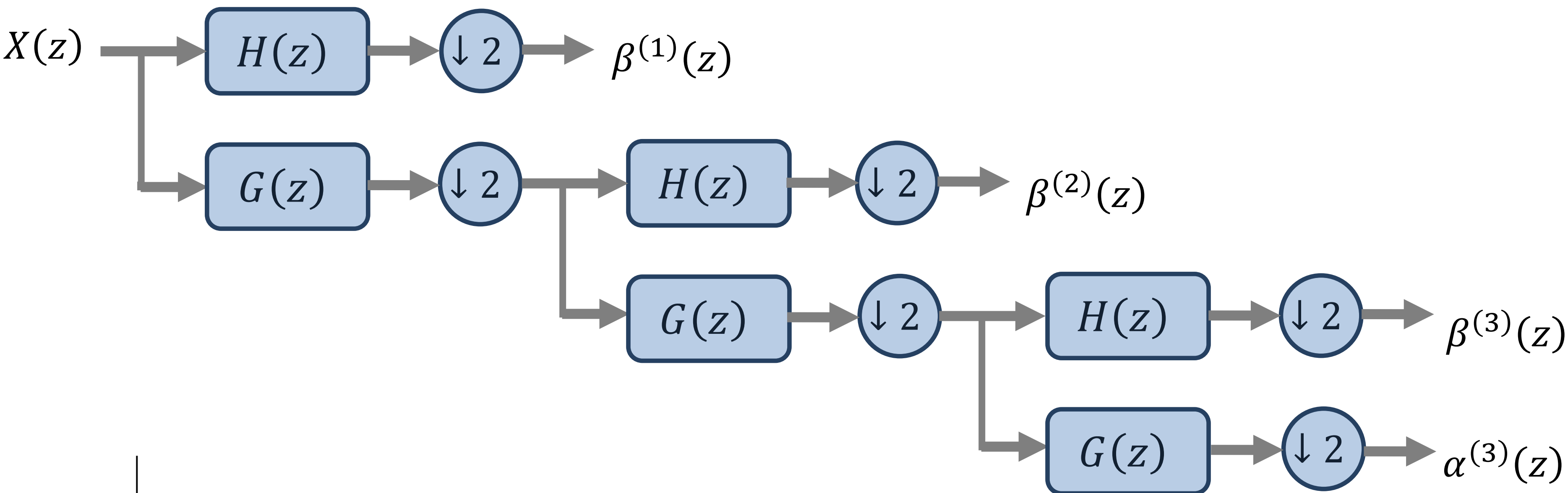
Wavelets / Sub-band coding

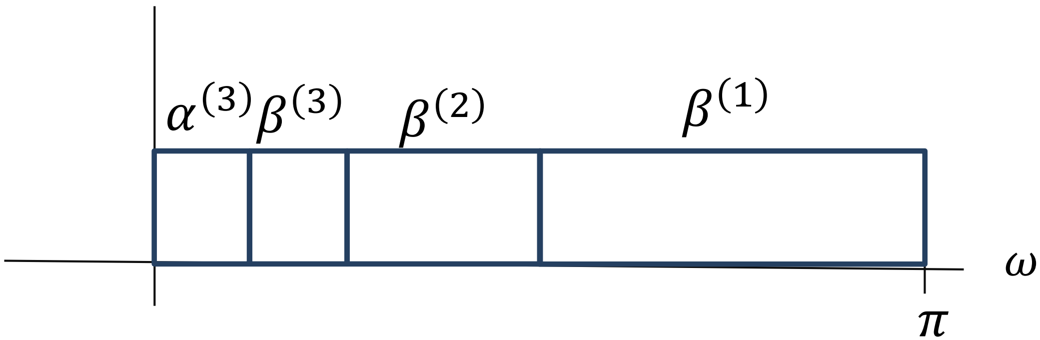
- **Consider this new [analysis] filter bank: What is going on here?**



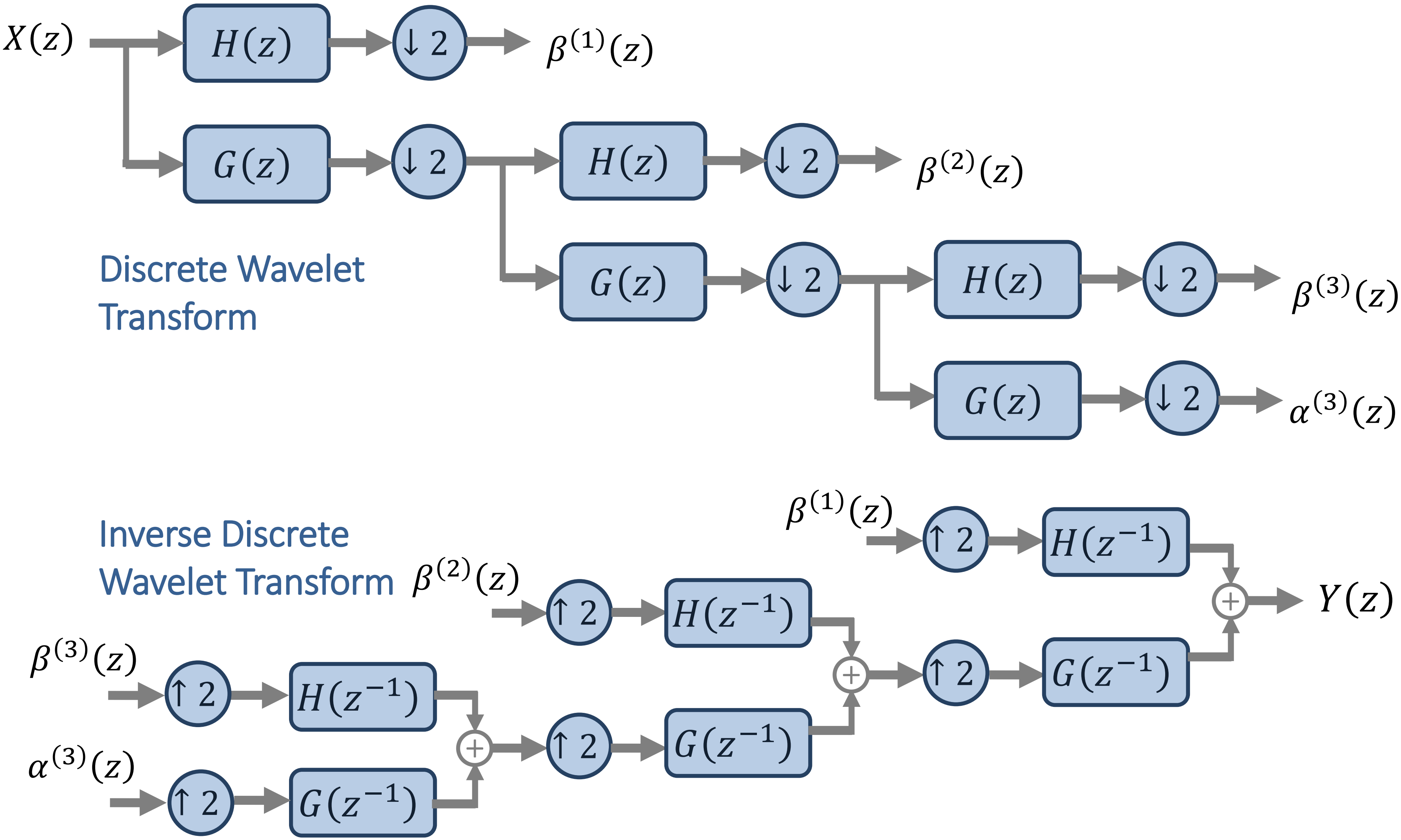
Wavelets / Sub-band coding

- **Consider this new [analysis] filter bank: What is going on here?**



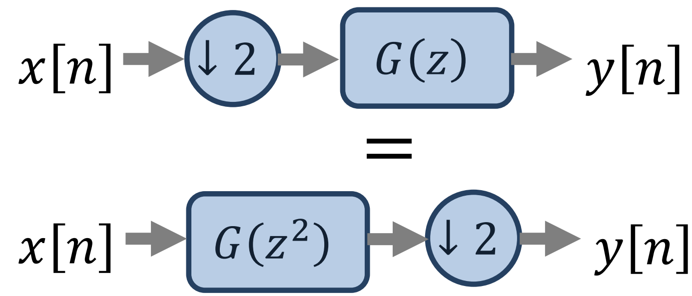


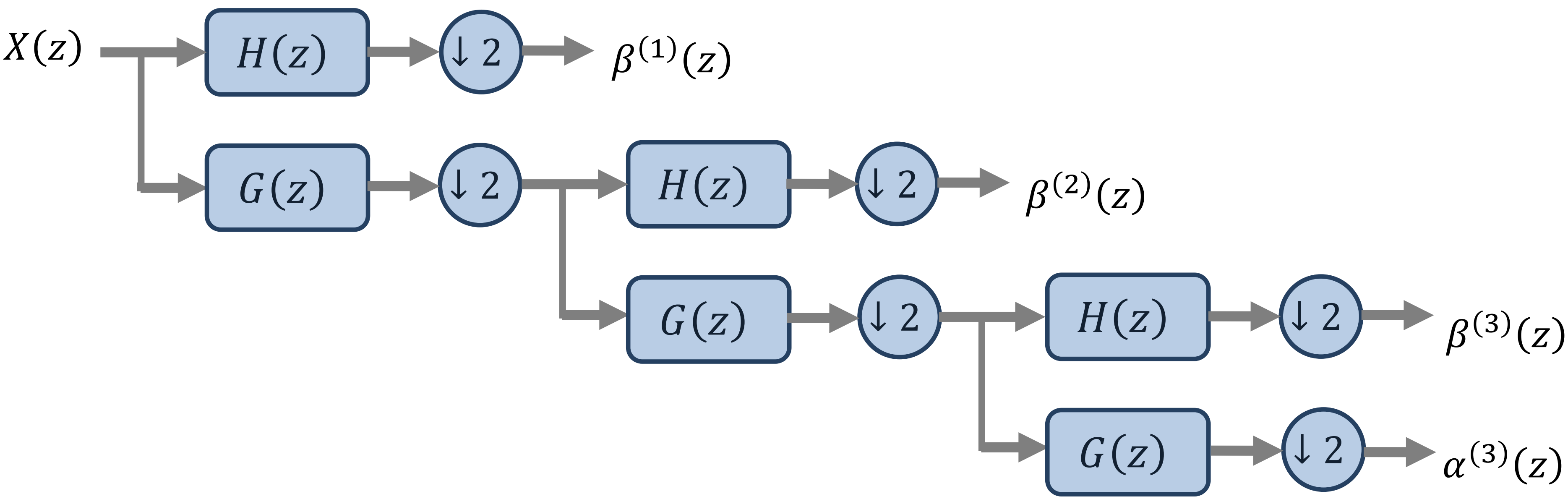
Wavelets / Sub-band coding



Wavelets / Sub-band coding

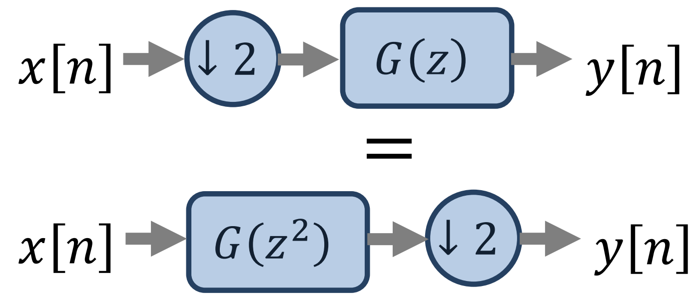
- **How do you analyze this?**

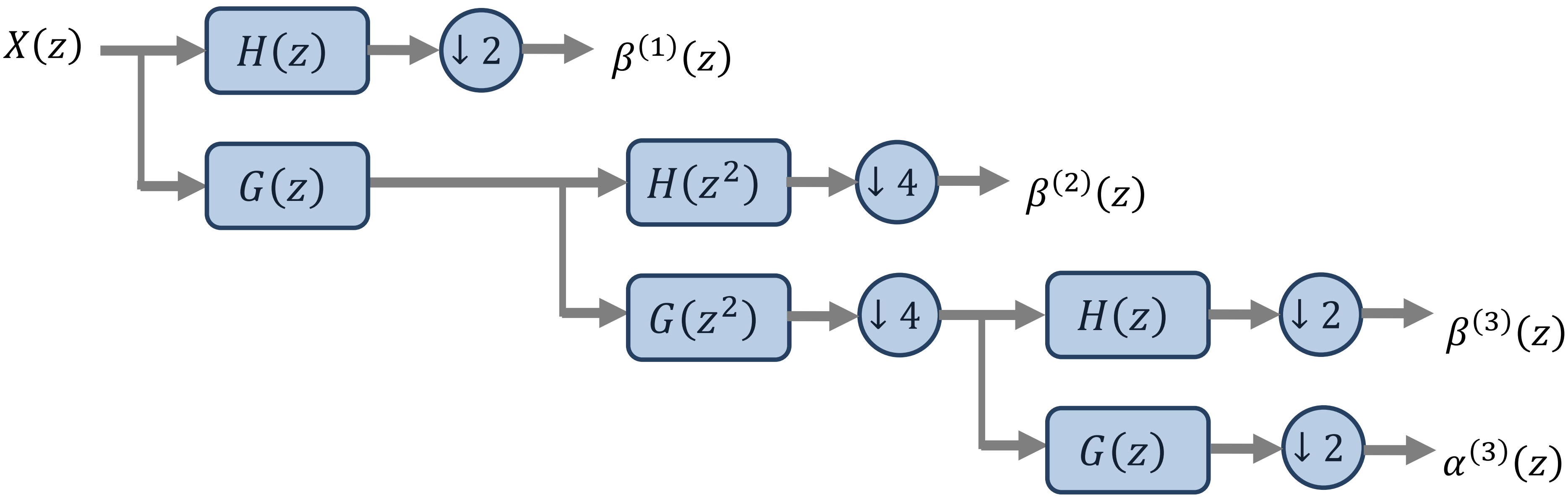




Wavelets / Sub-band coding

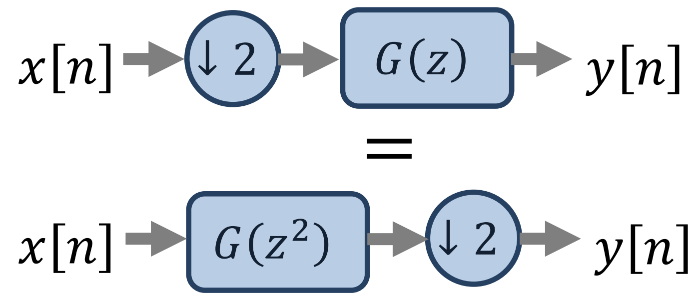
- **How do you analyze this?**

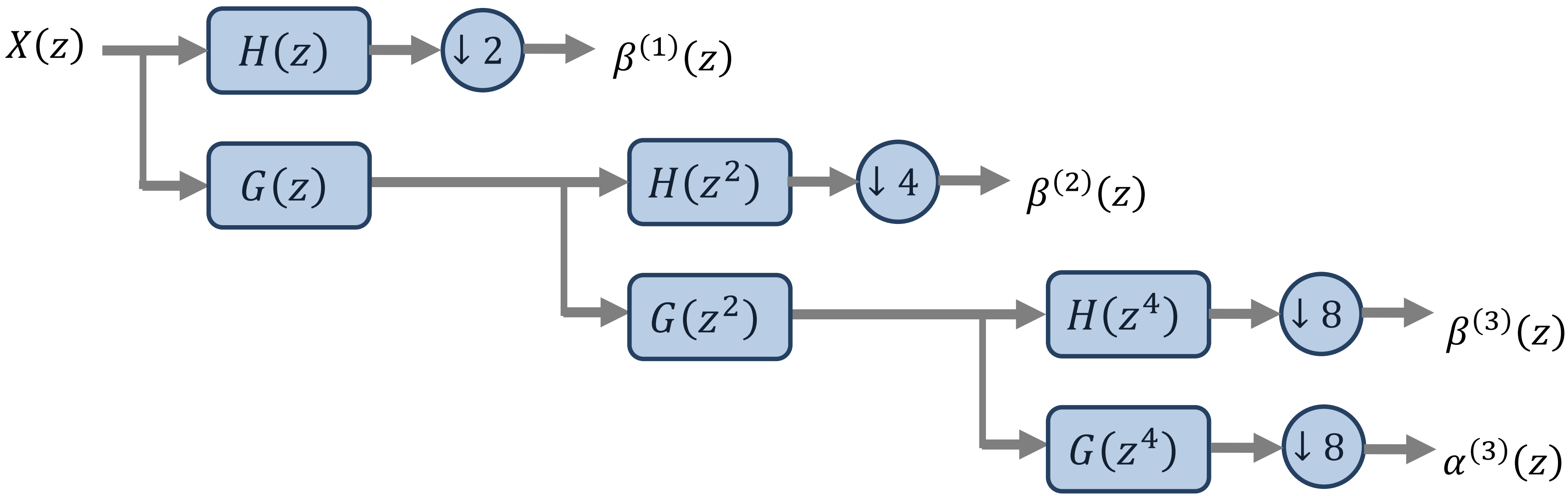




Wavelets / Sub-band coding

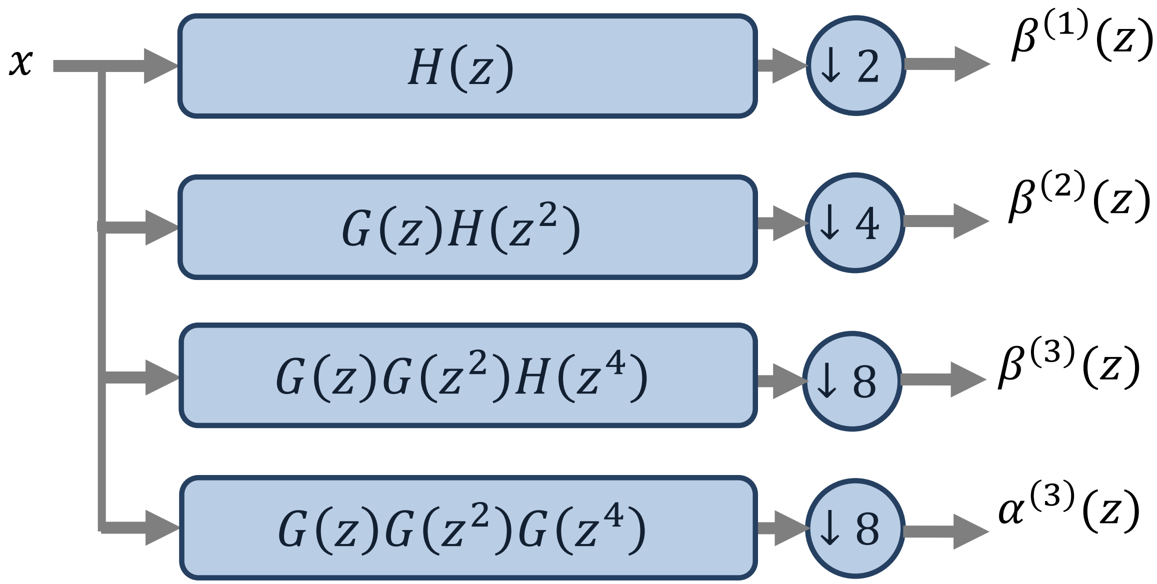
- **How do you analyze this?**





Wavelets / Sub-band coding

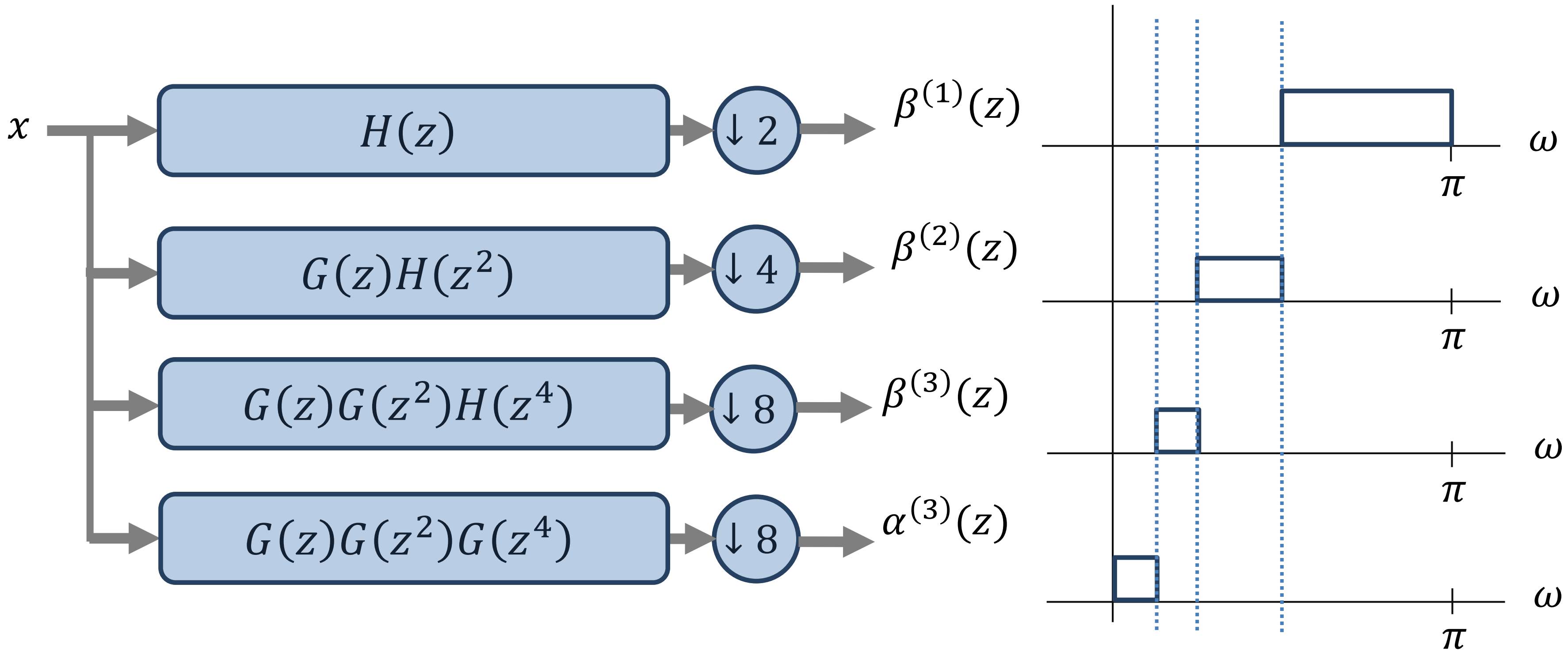
- **How do you analyze this?**



It's a filter bank!

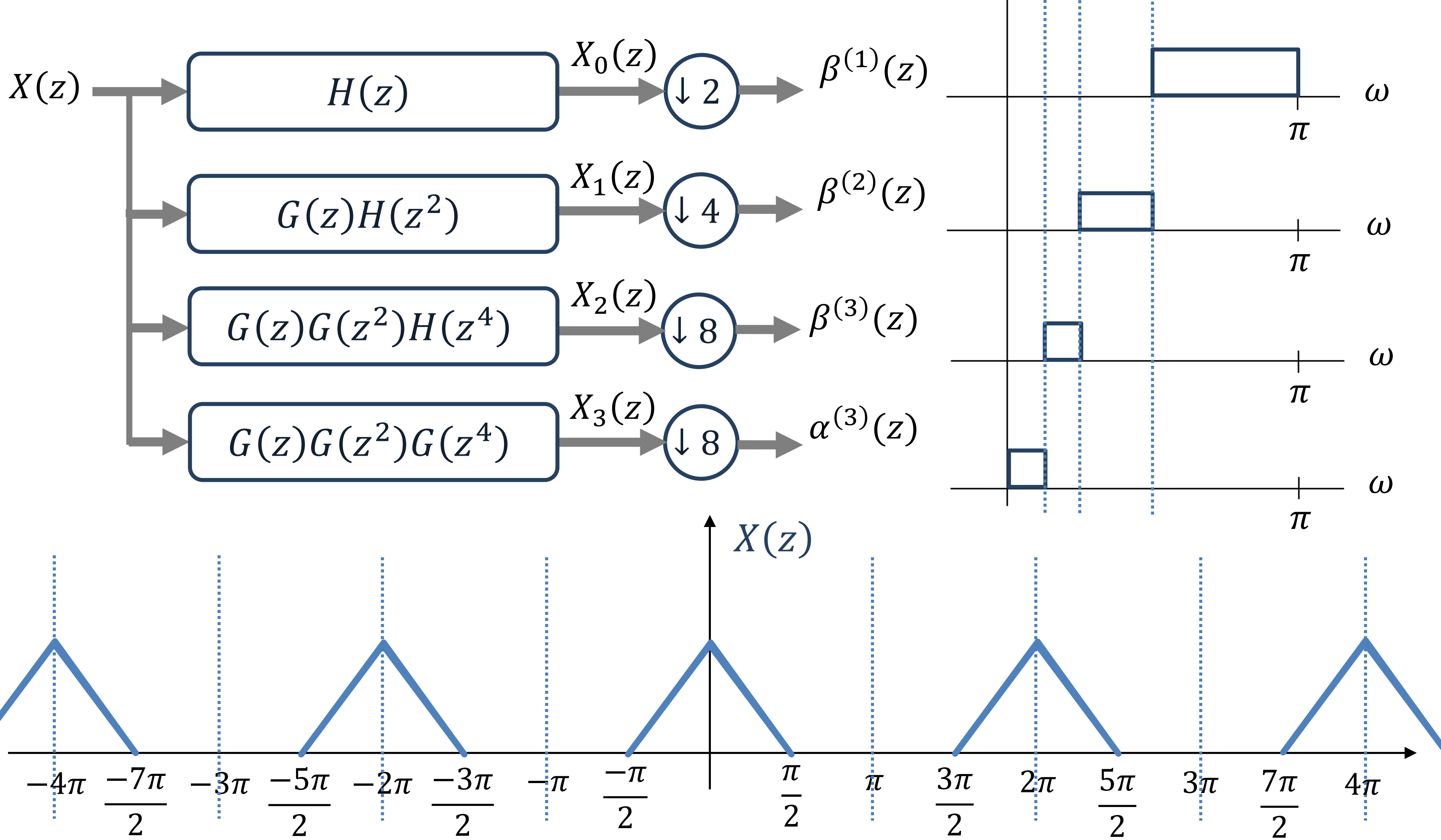
Wavelets / Sub-band coding

- **How do you analyze this?**



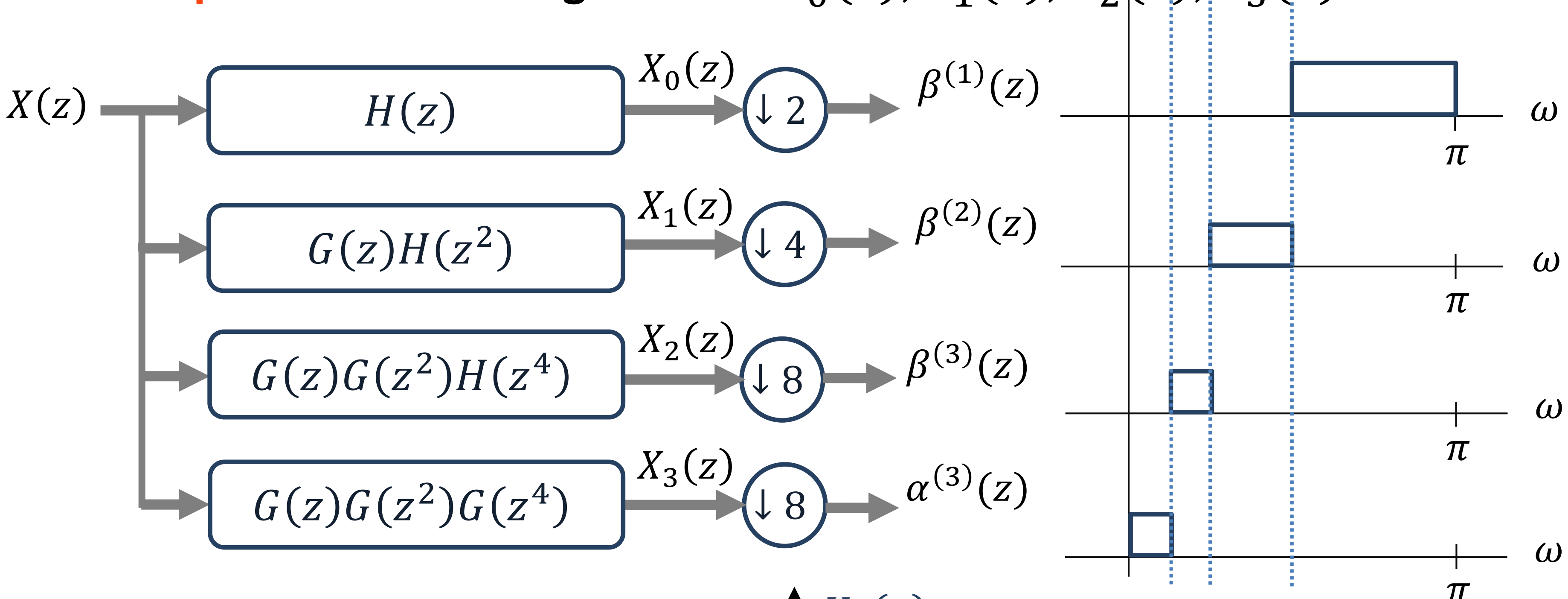
Wavelets / Sub-band coding

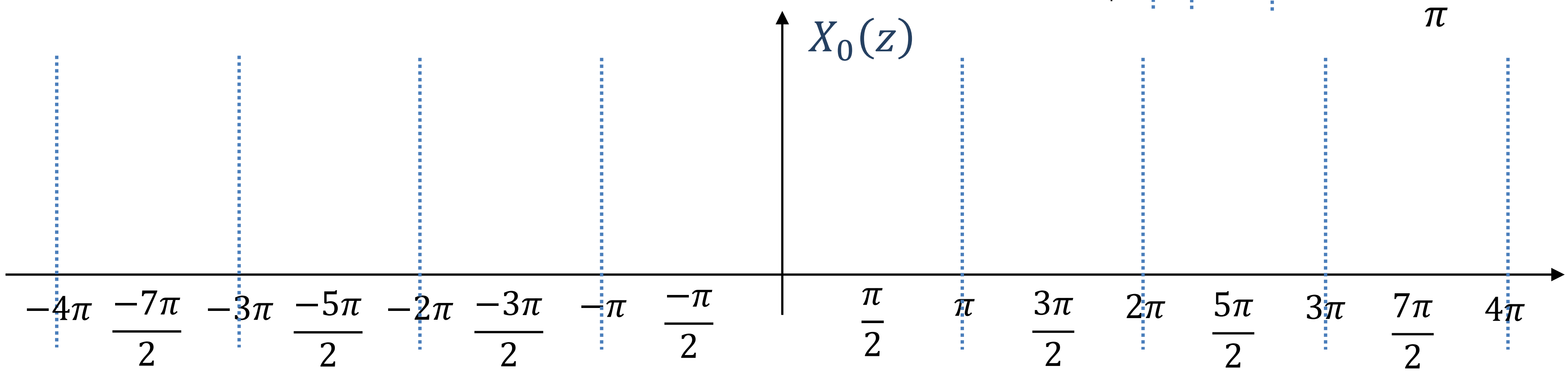
- **Example: Determine magnitude of $X_0(z), X_1(z), X_2(z), X_3(z)$**



Wavelets / Sub-band coding

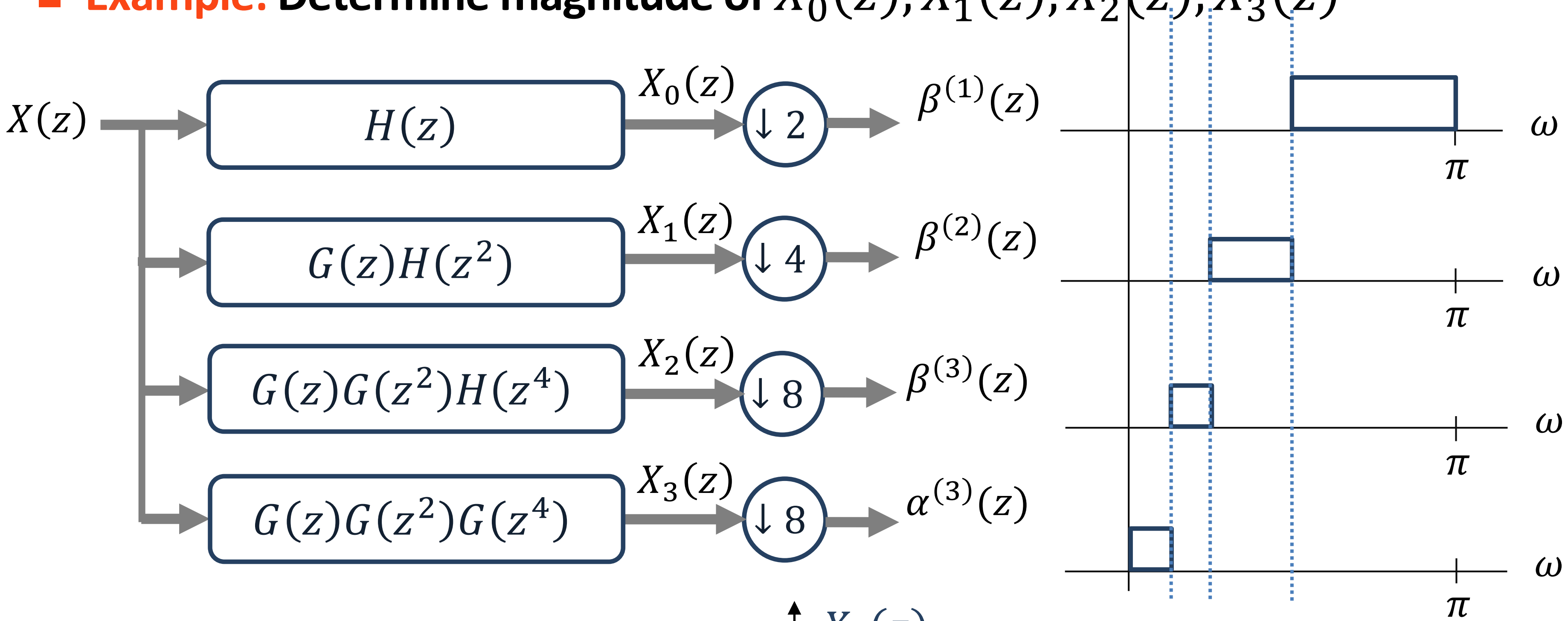
- **Example: Determine magnitude of $X_0(z), X_1(z), X_2(z), X_3(z)$**

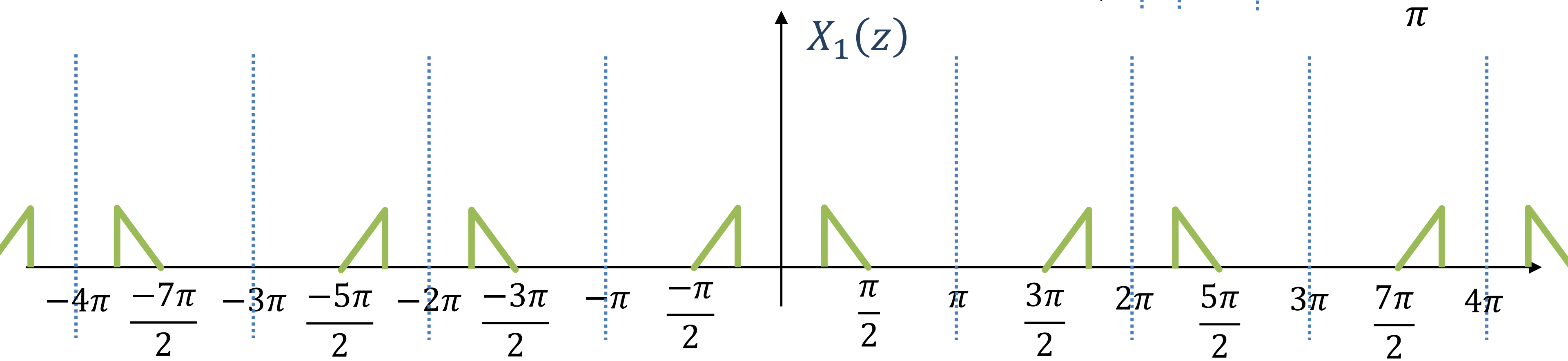




Wavelets / Sub-band coding

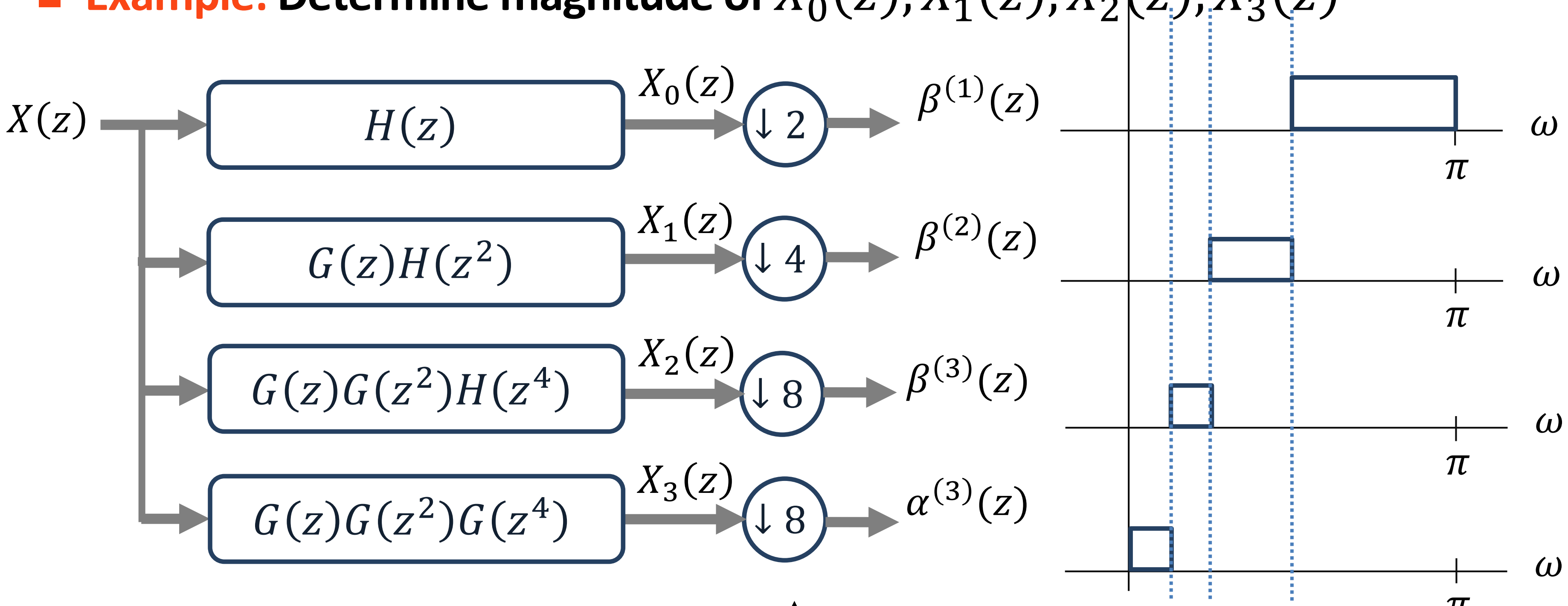
- **Example:** **Determine magnitude of** $X_0(z), X_1(z), X_2(z), X_3(z)$

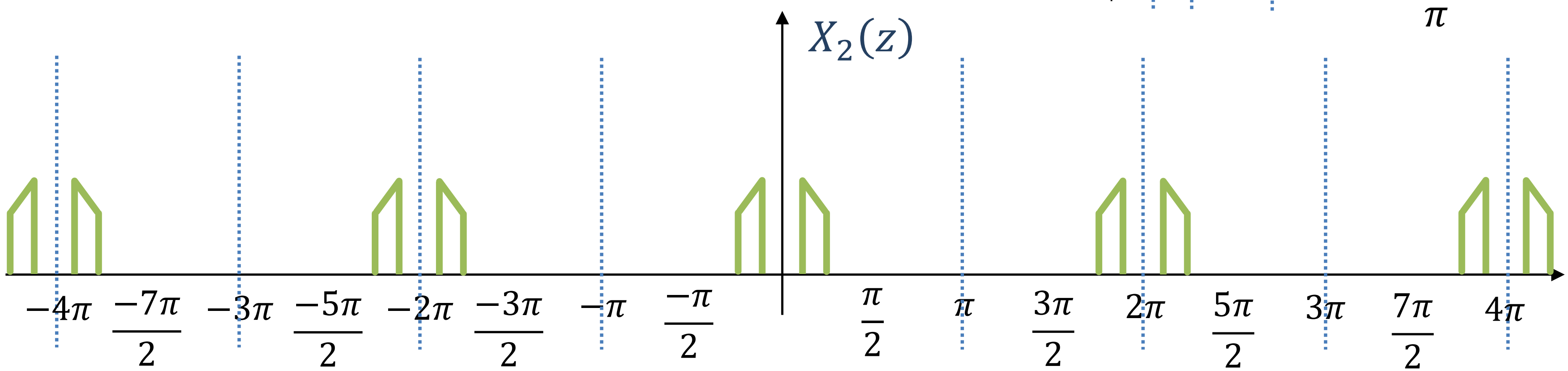




Wavelets / Sub-band coding

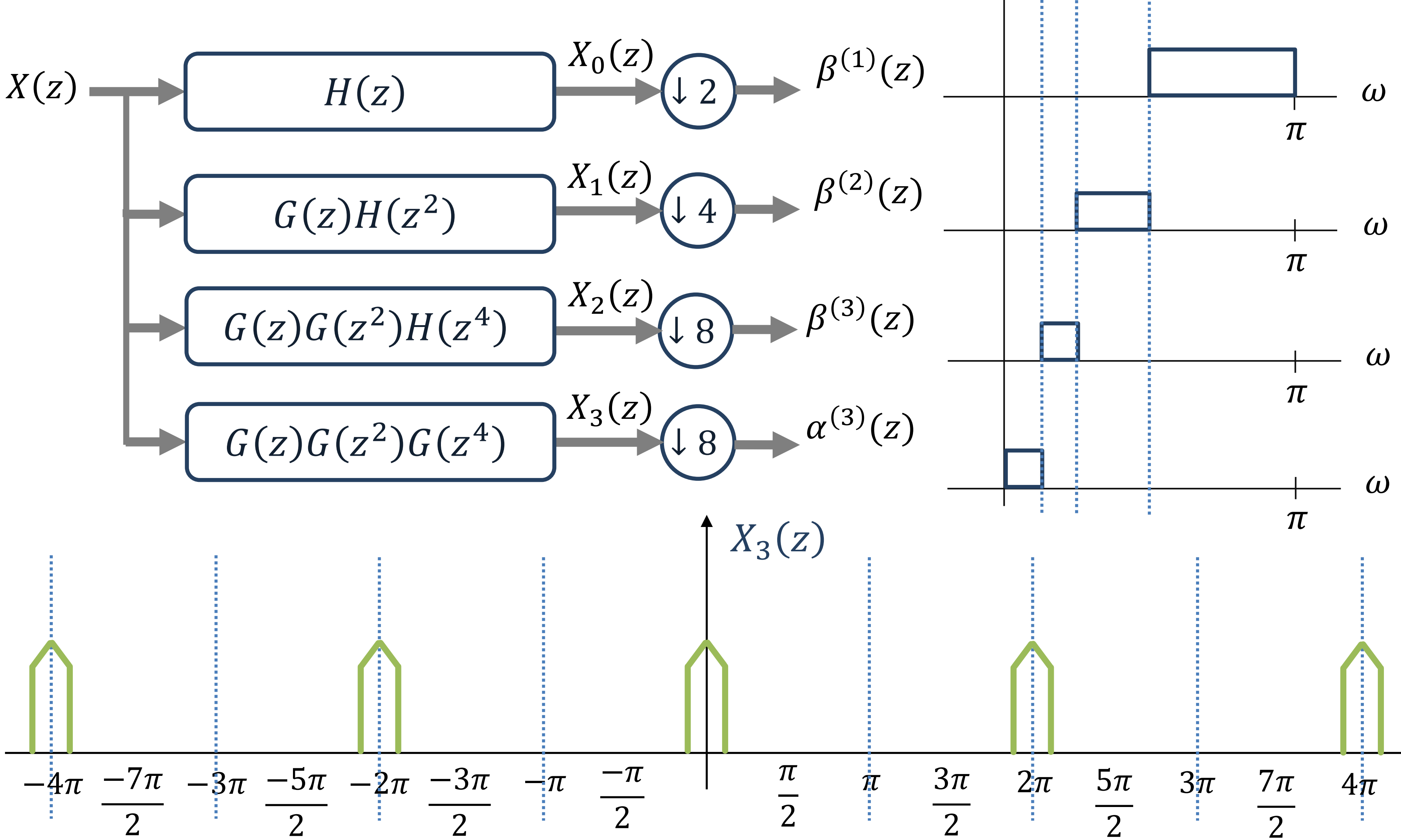
- **Example:** **Determine magnitude of** $X_0(z), X_1(z), X_2(z), X_3(z)$



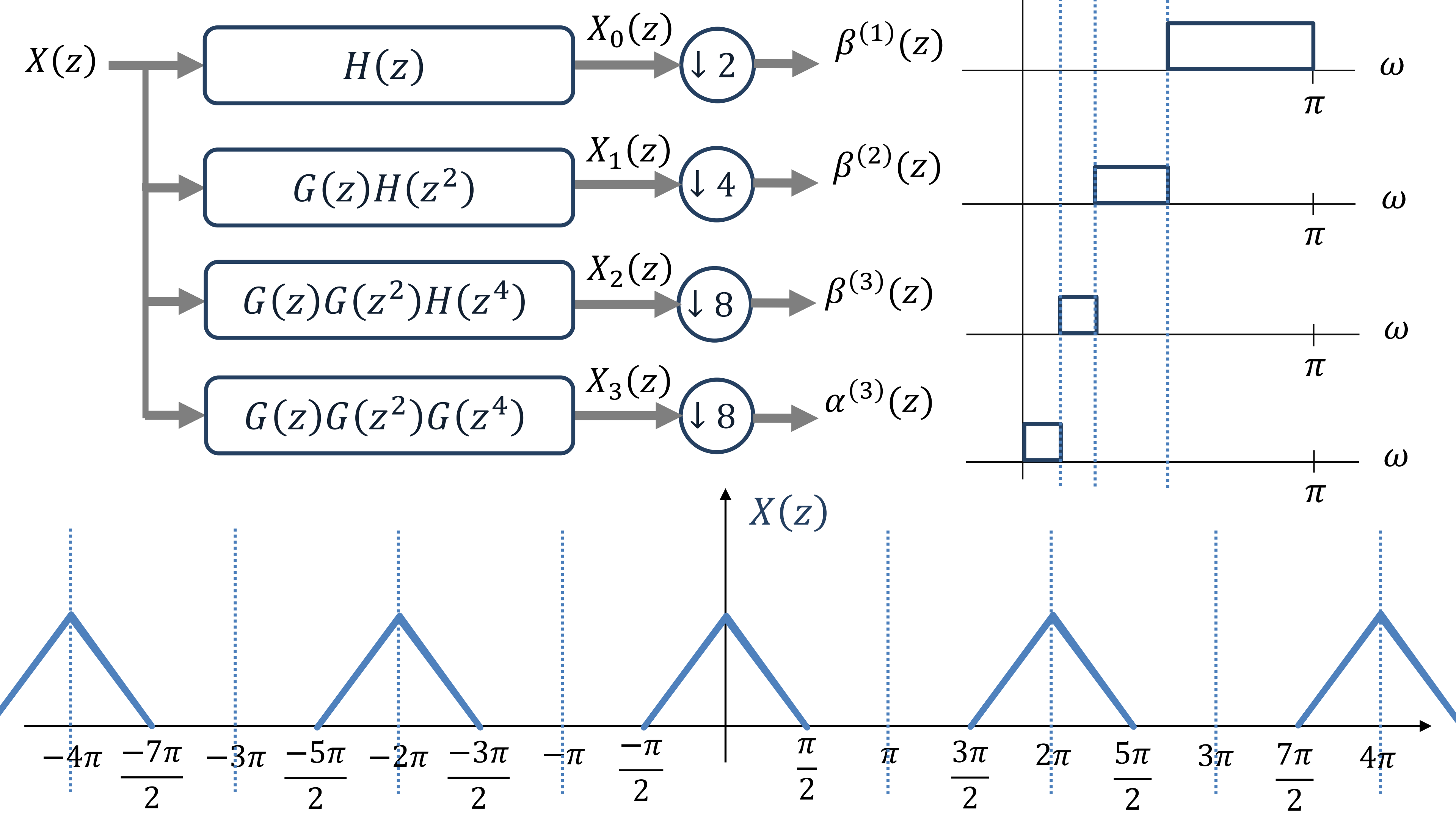


Wavelets / Sub-band coding

- **Example: Determine magnitude of** $X_0(z), X_1(z), X_2(z), X_3(z)$

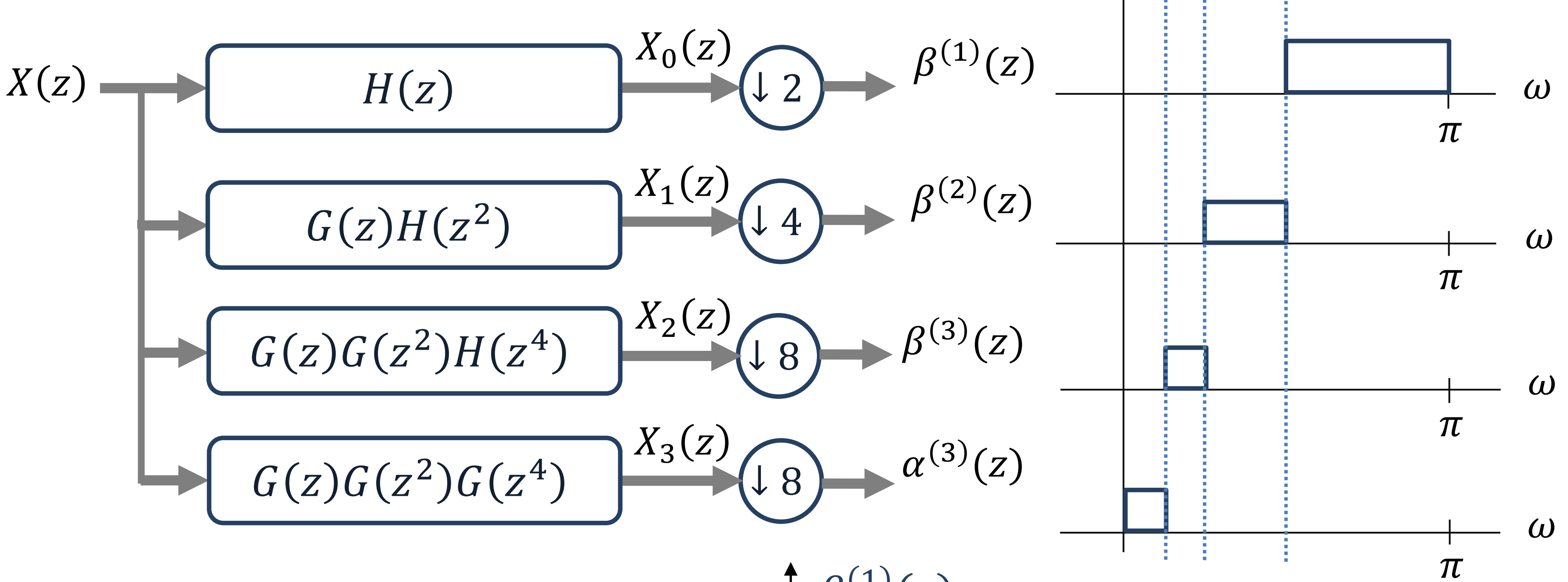


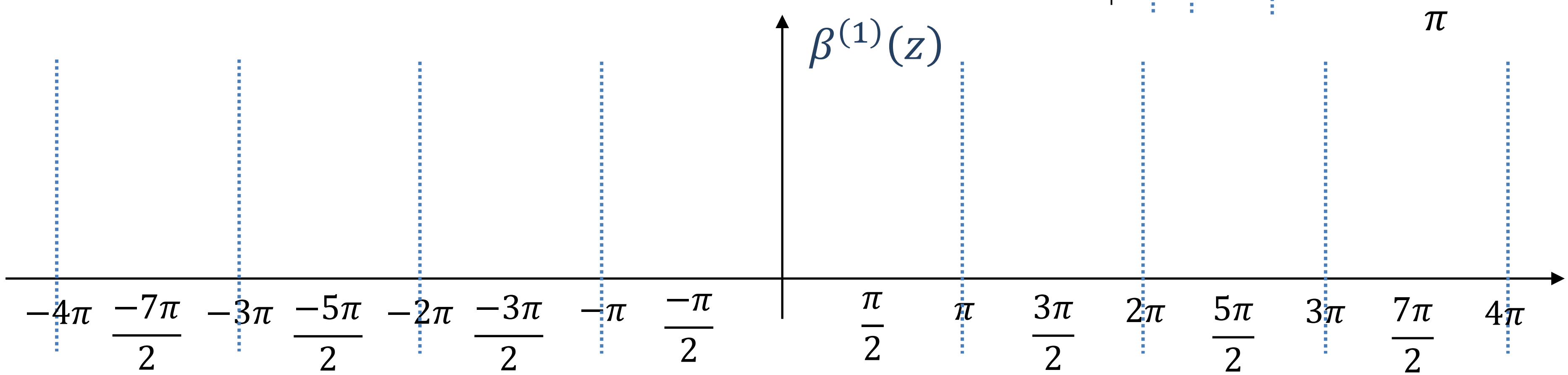
- **Example:** **Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$



Wavelets / Sub-band coding

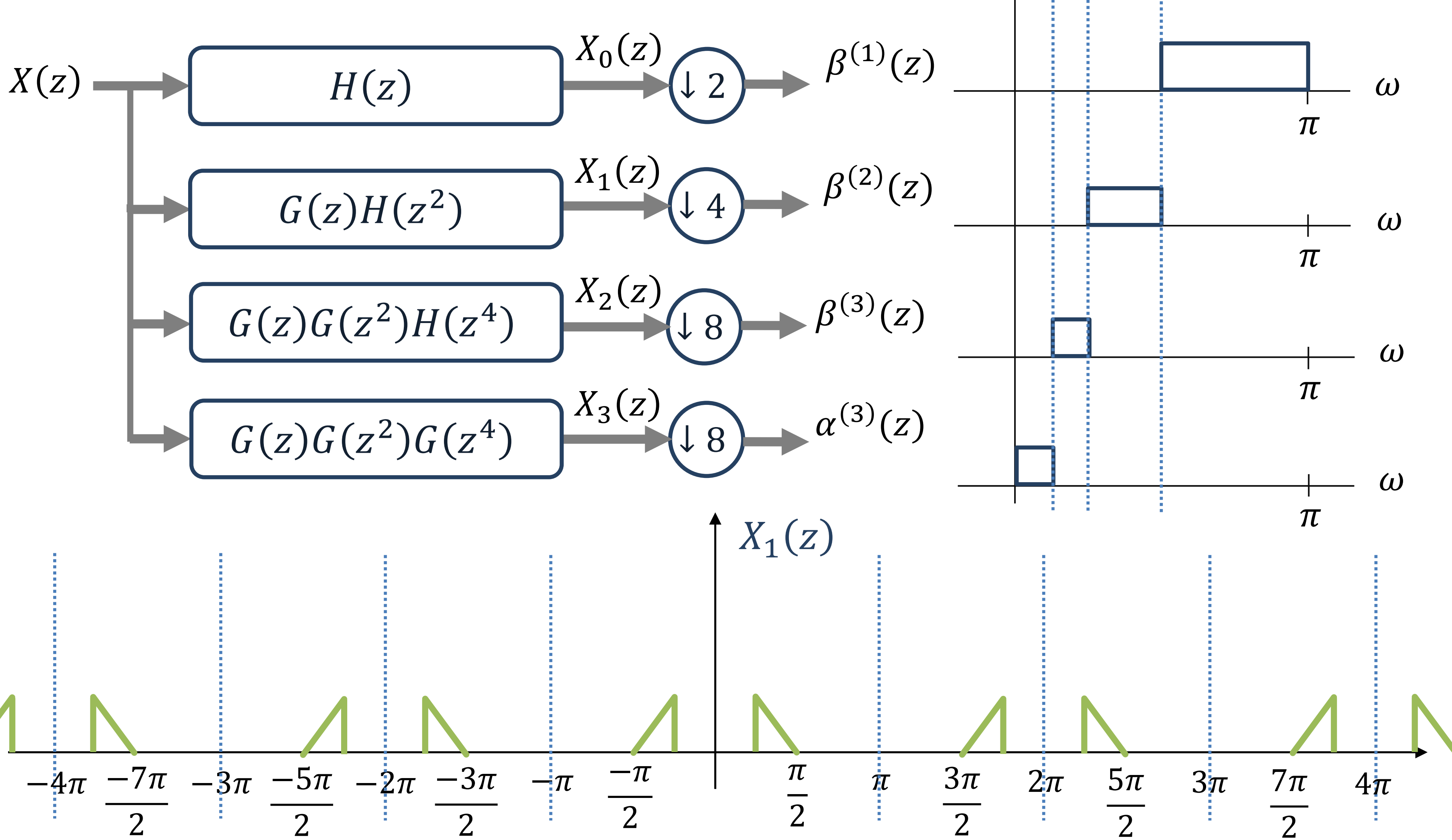
- **Example: Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$





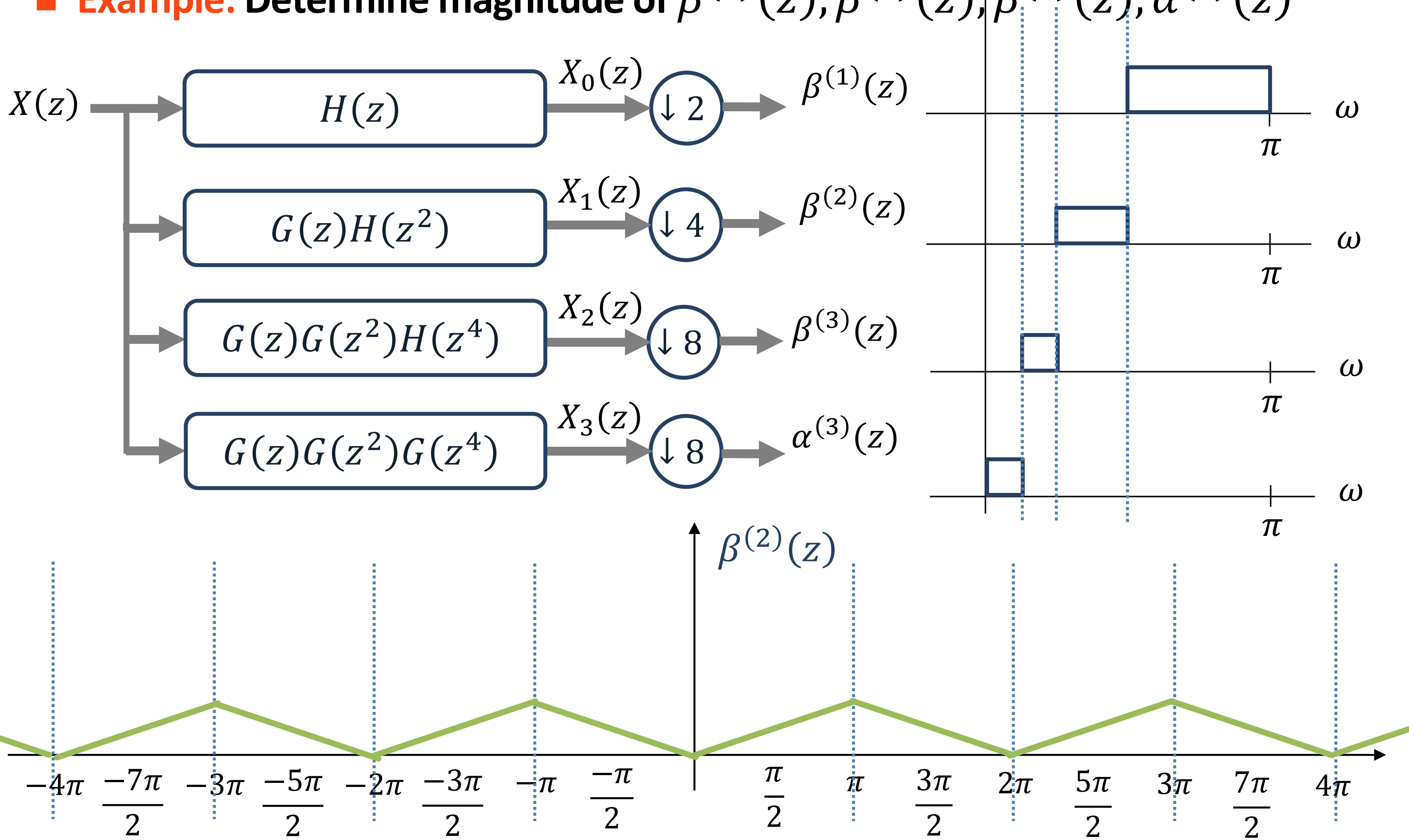
Wavelets / Sub-band coding

- **Example: Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$



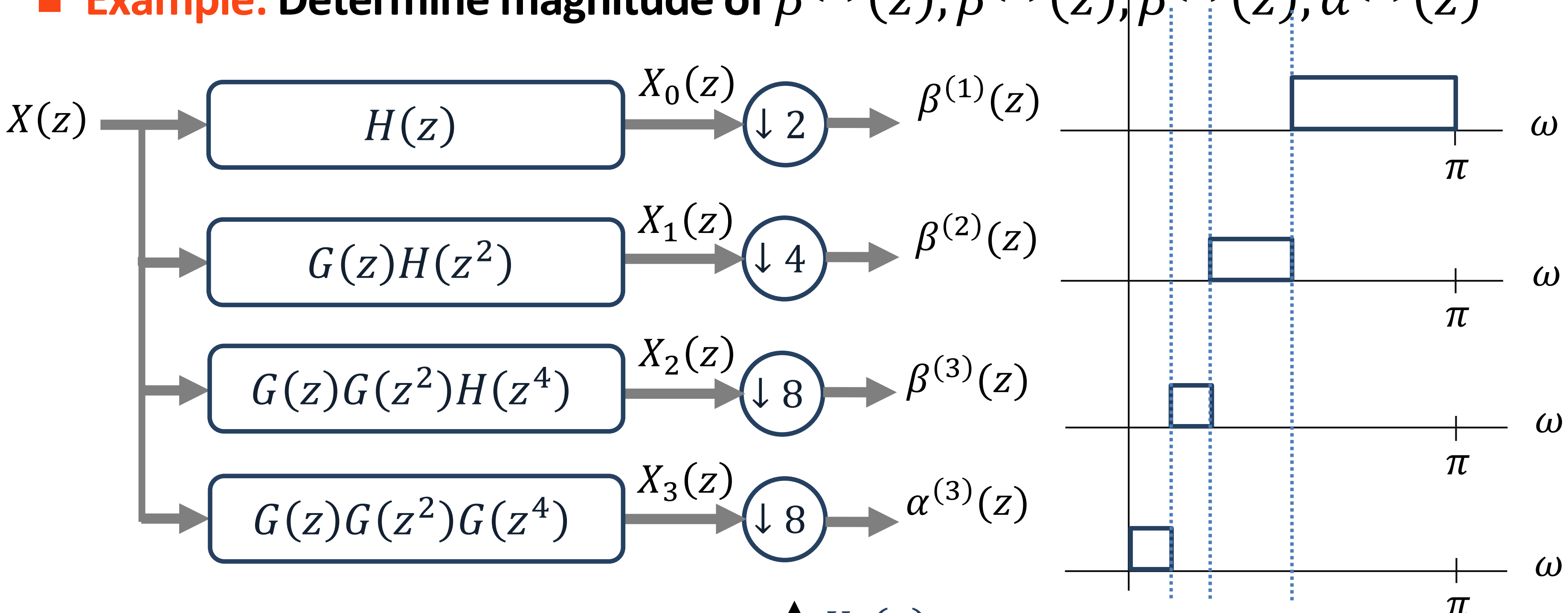
Wavelets / Sub-band coding

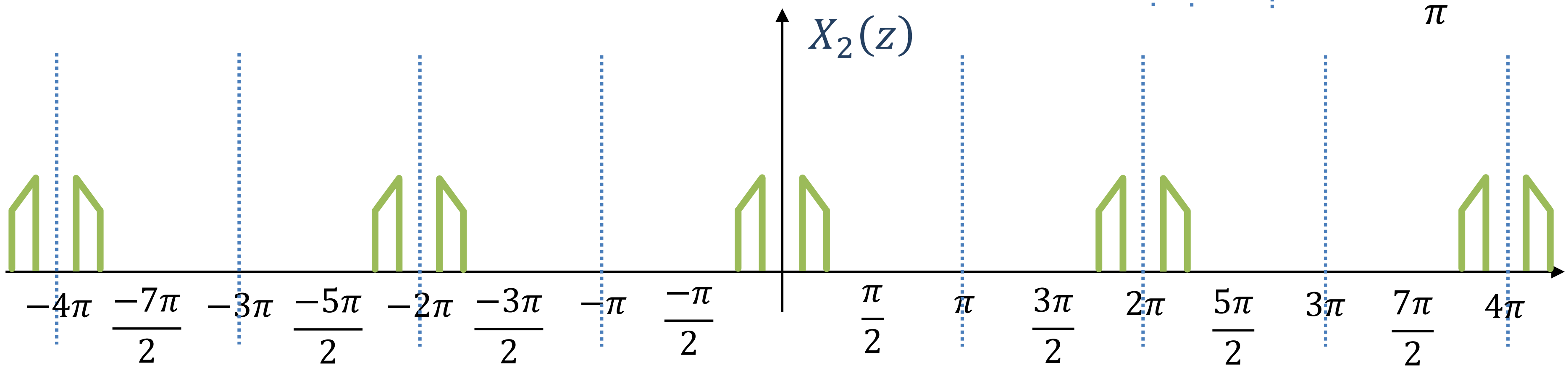
- **Example:** **Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$



Wavelets / Sub-band coding

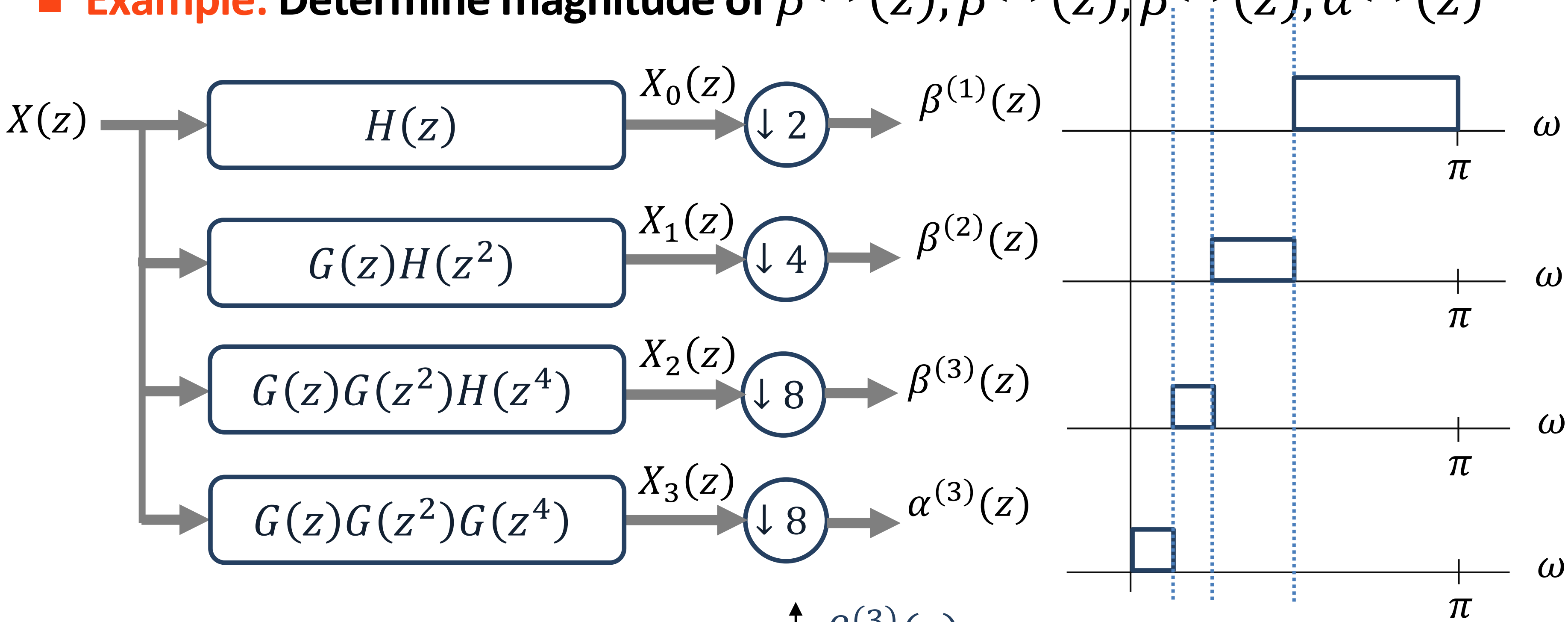
- **Example:** **Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$

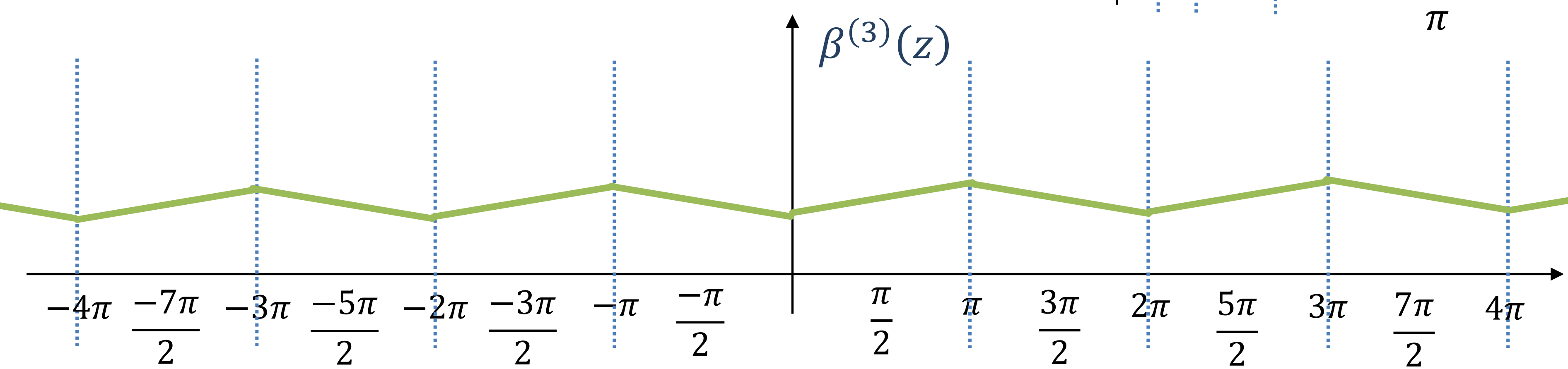




Wavelets / Sub-band coding

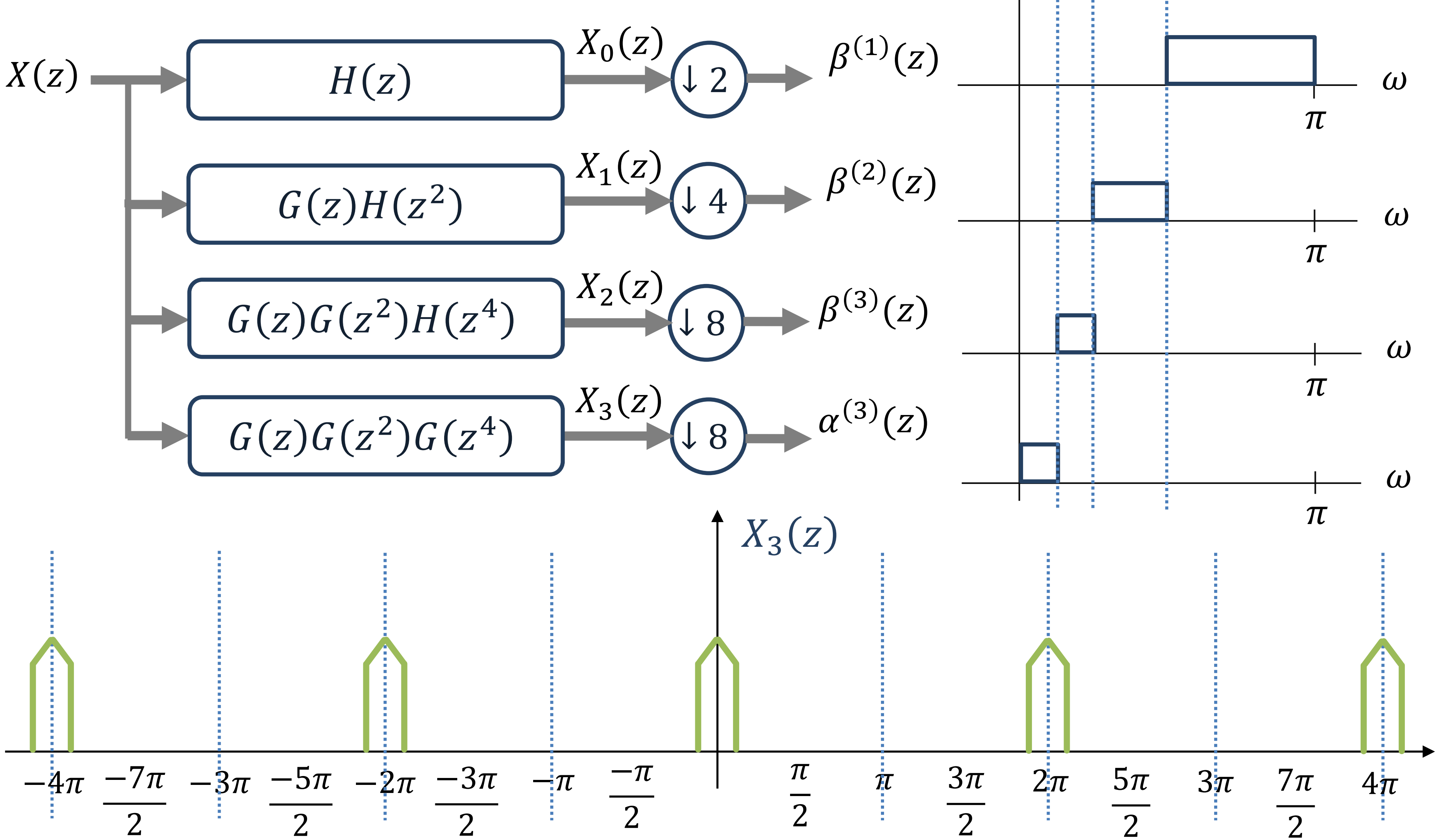
- **Example: Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$





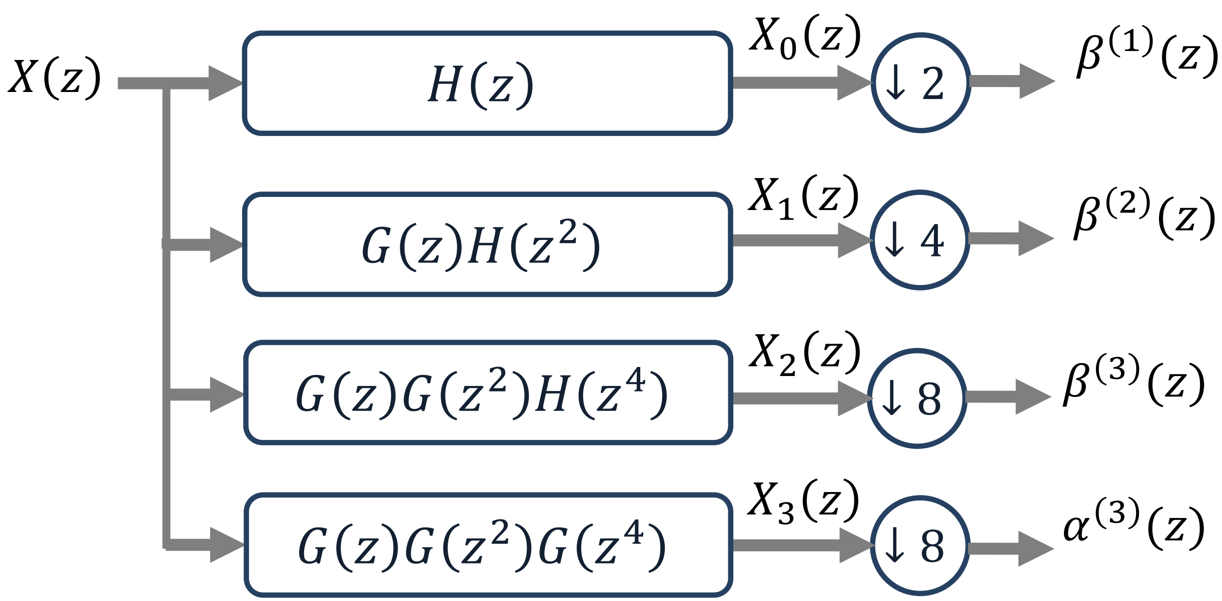
Wavelets / Sub-band coding

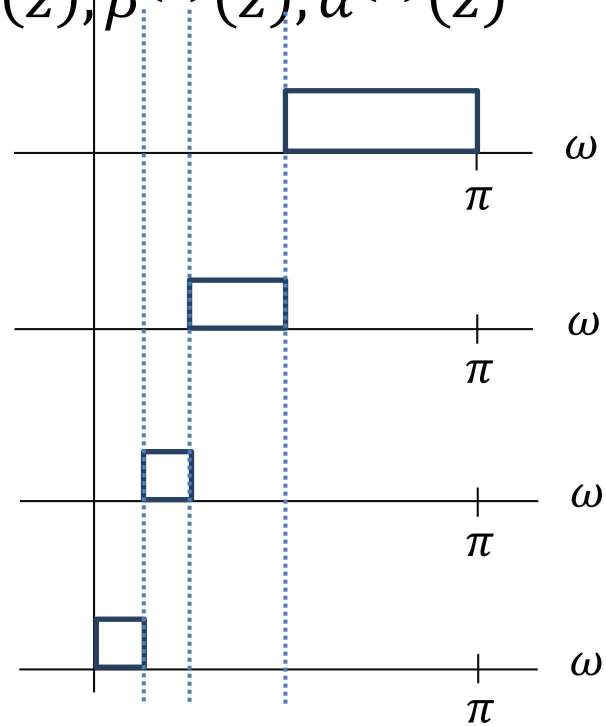
- **Example: Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$

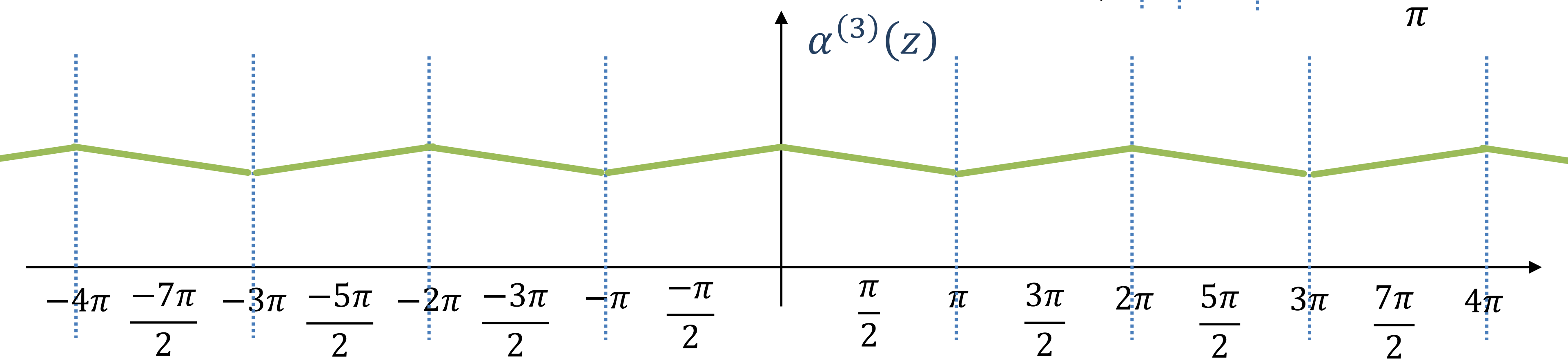


Wavelets / Sub-band coding

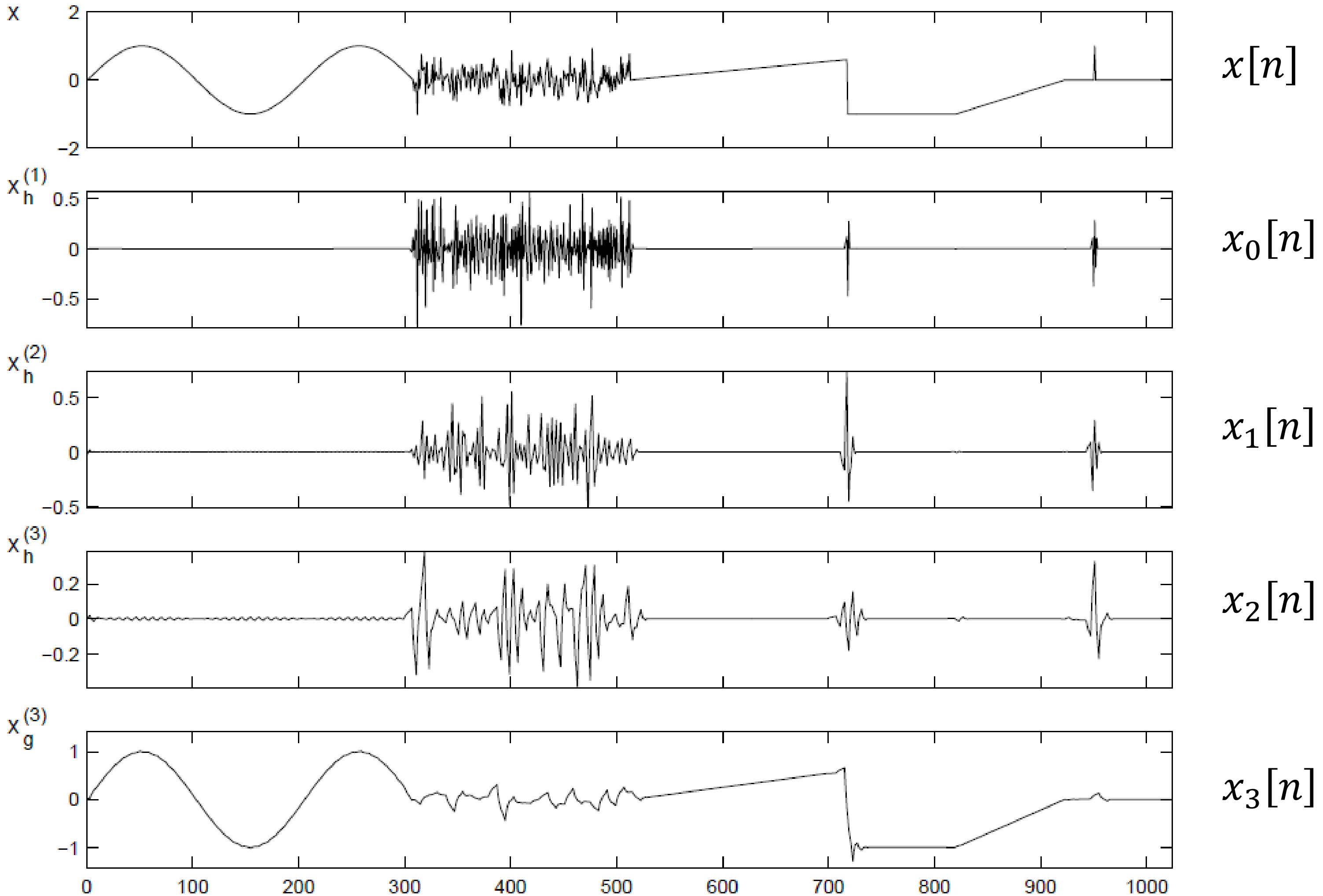
- **Example: Determine magnitude of** $\beta^{(1)}(z), \beta^{(2)}(z), \beta^{(3)}(z), \alpha^{(3)}(z)$







Wavelets / Sub-band coding



From: "Fourier and Wavelet Signal Processing (alpha 3.2 release)" by Jelena Kovacevic, Vivek K Goyal, and Martin Vetterli

Lecture 27: Wavelets to Modern Signal Processing

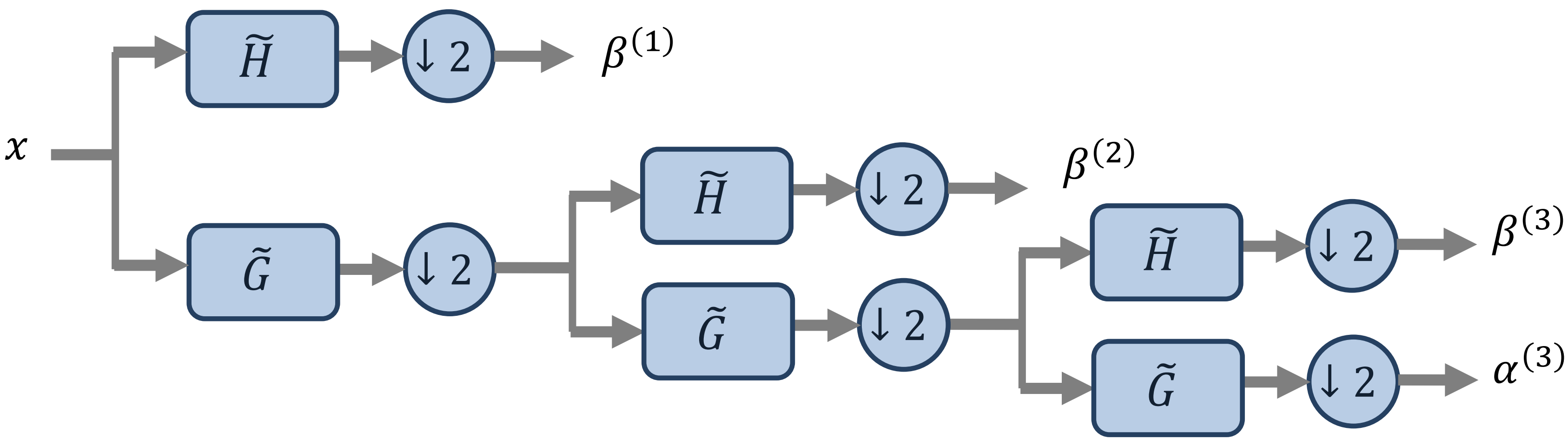
Foundations of Digital Signal Processing

Outline

- Review of Filter Banks and Wavelets
- **Applications of Wavelets**
- Applications of Filter Banks and Time-Frequency Representations
- Modern Signal Processing: Vectors and Matrices
- Modern Signal Processing: Diagonalization
- Modern Signal Processing: Compressive Sensing
- Modern Signal Processing: Compressive Sensing

Application of Wavelets

- **Computational Complexity of the Discrete Wavelet Transform**

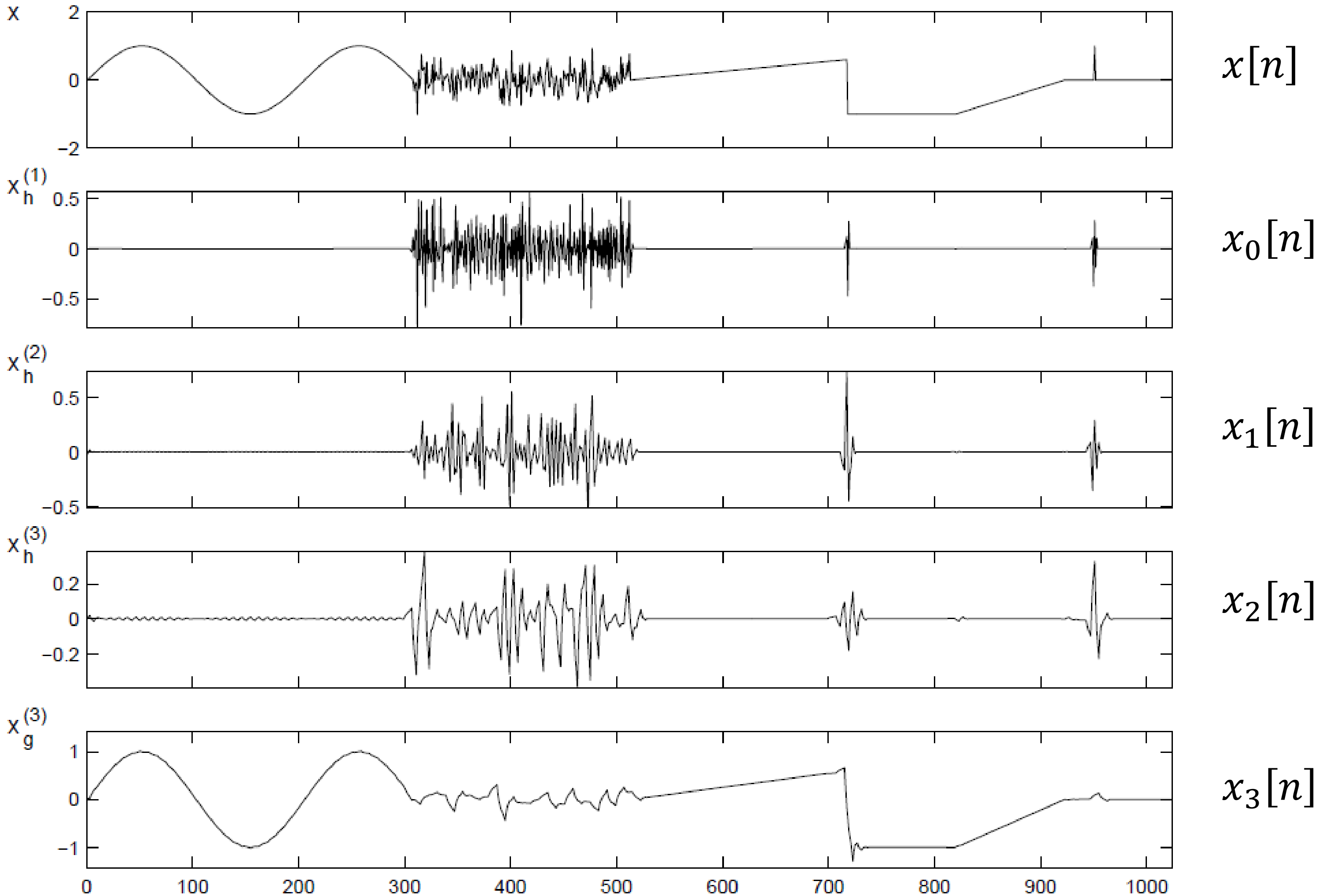


$\mathcal{O}(NM)$ $\quad$ $\mathcal{O}\left(\frac{NM}{2}\right)$ $\quad$ $\mathcal{O}\left(\frac{NM}{4}\right)$

Filter size $\quad$ Signal length

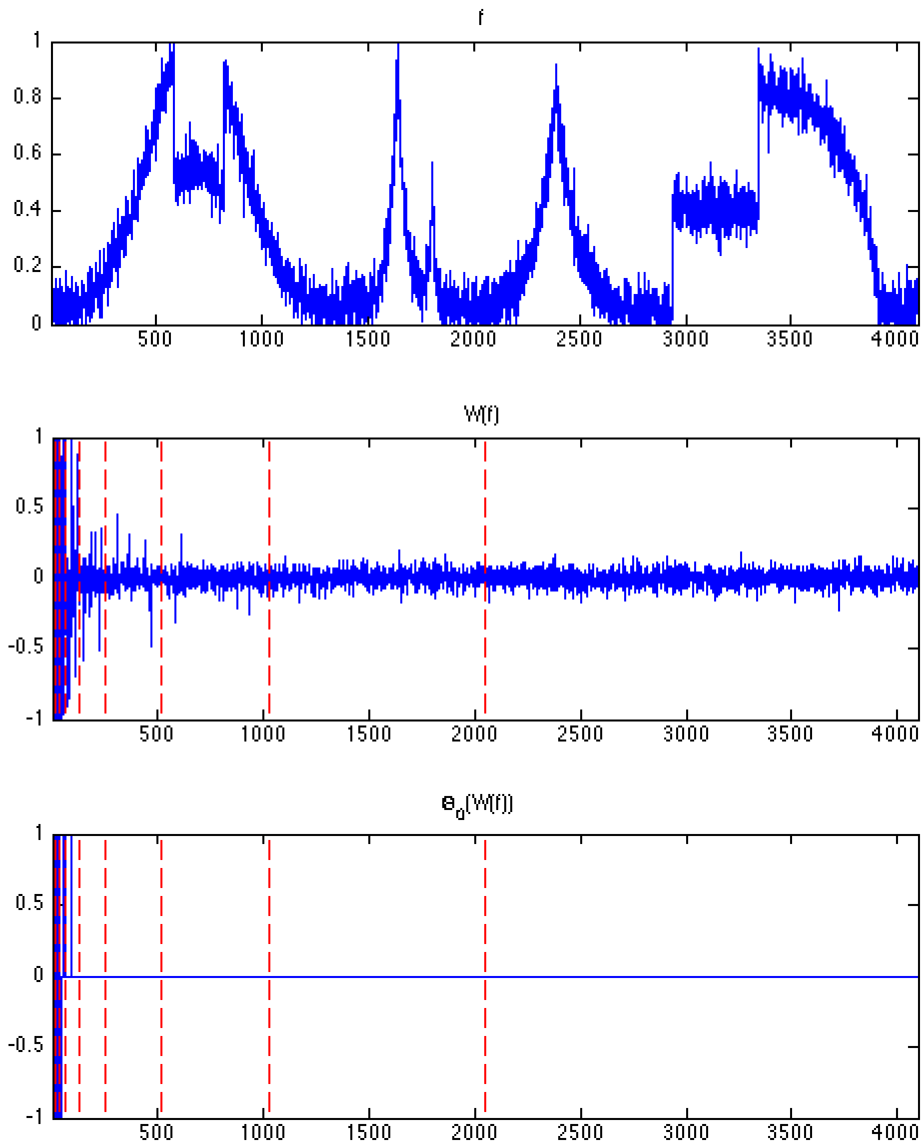
Overall Complexity: $\mathcal{O}(NM) \sim \mathcal{O}(M)$

Application of Wavelets



From: “Fourier and Wavelet Signal Processing (alpha 3.2 release)” by Jelena Kovacevic, Vivek K Goyal, and Martin Vetterli

Application of Wavelets



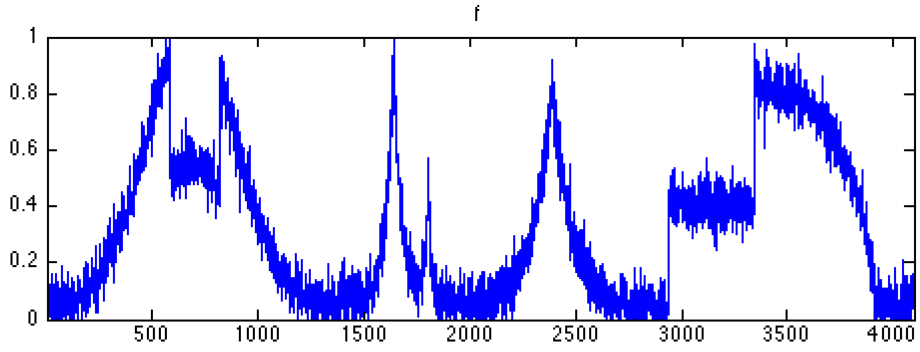
Signal x[n]

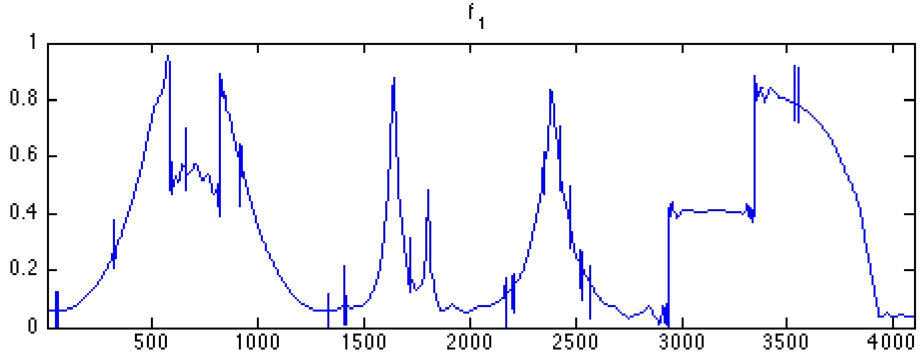
Wavelet Coefficients
(after discrete wavelet transform)

Thresholded Wavelet Coefficients

From: http://www.numerical-tours.com/matlab/denoisingwav_1_wavelet_1d/

Application of Wavelets





Signal x[n]

Denoised signal (from inverse discrete wavelet transform of thresholded coefficients)

From: http://www.numerical-tours.com/matlab/denoisingwav_1_wavelet_1d/

Application of Wavelets

- **Example**

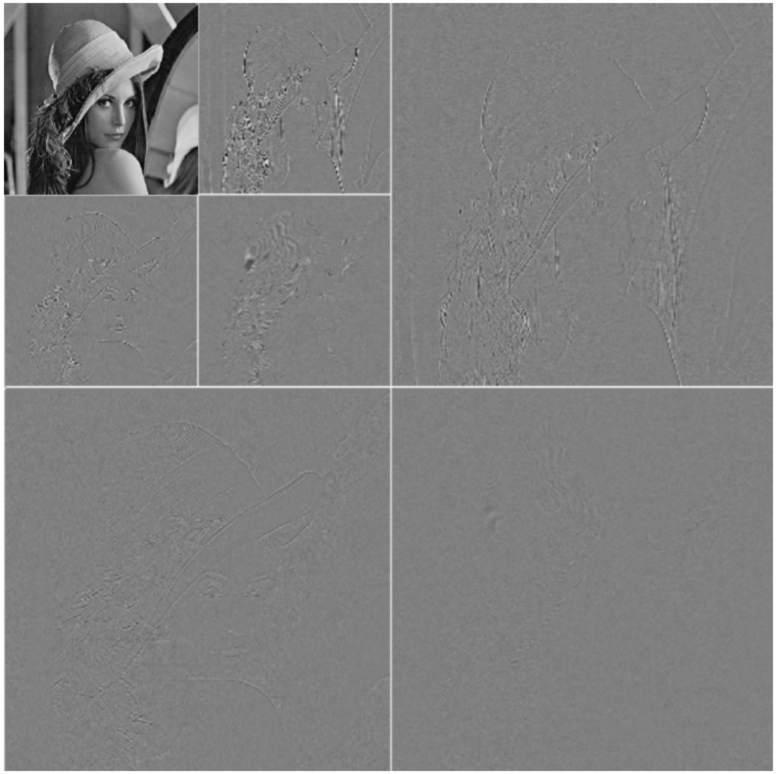
1 level DWT

From:
http://web.stanford.edu/class/ee398a/handouts/lectures/09-SubbandCoding.pdf

Application of Wavelets

- **Example**

2 level
DWT

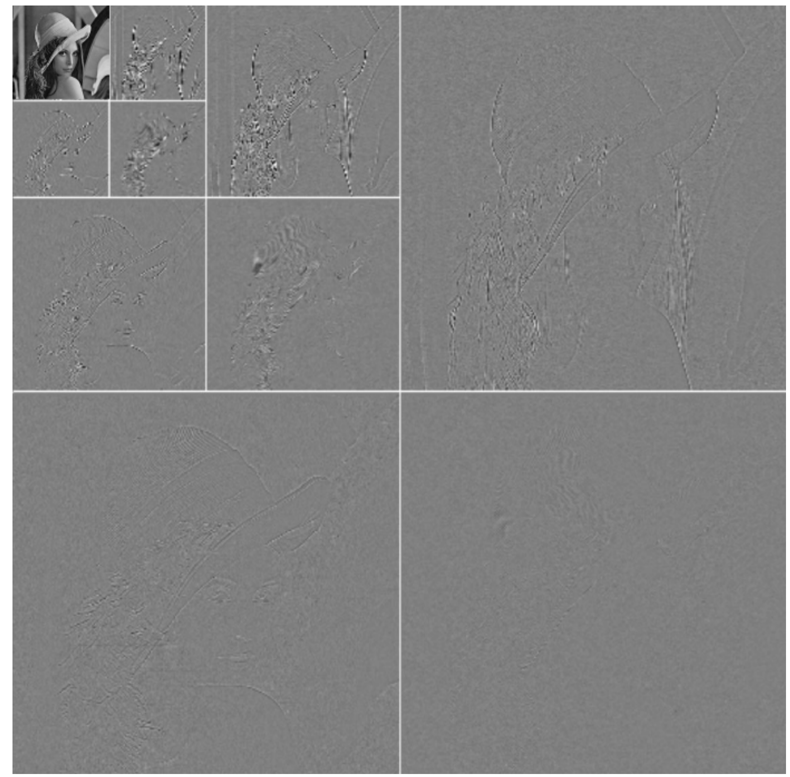


From:
http://web.stanford.edu/class/ee398a/handouts/lectures/09-SubbandCoding.pdf

Application of Wavelets

- **Example**

3 level
DWT

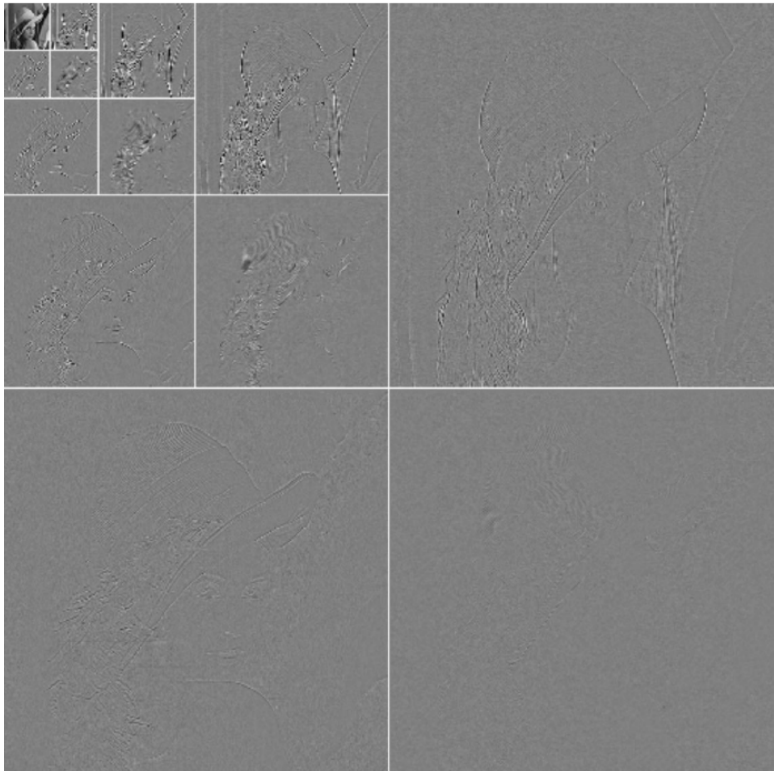


From:
http://web.stanford.edu/class/ee398a/handouts/lectures/09-SubbandCoding.pdf

Application of Wavelets

- **Example**

4 level DWT

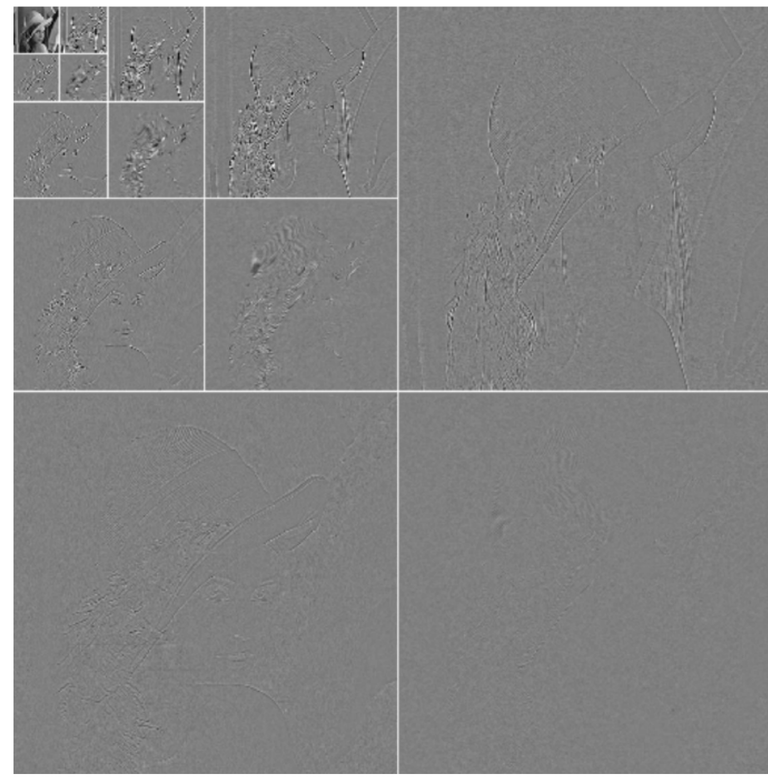


From:
http://web.stanford.edu/class/ee398a/handouts/lectures/09-SubbandCoding.pdf

Application of Wavelets

- **Question:** **How can we use this??**

4 level DWT



From:
http://web.stanford.edu/class/ee398a/handouts/lectures/09-SubbandCoding.pdf

Application of Wavelets

■ Denoising

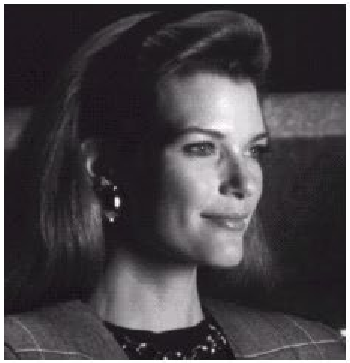
(a) original image (b) noisy image (c) orthogonal wavelets

(d) undecimated wavelets (e) steerable pyramid (f) log-Gabor wavelets

From: Sylvain Fischer, Filip Sroubek, Laurent U Perrinet, Gabriel Cristobal, "Self-Invertible 2D Log-Gabor Wavelets," International Journal of Computer Vision 75(2):231-246

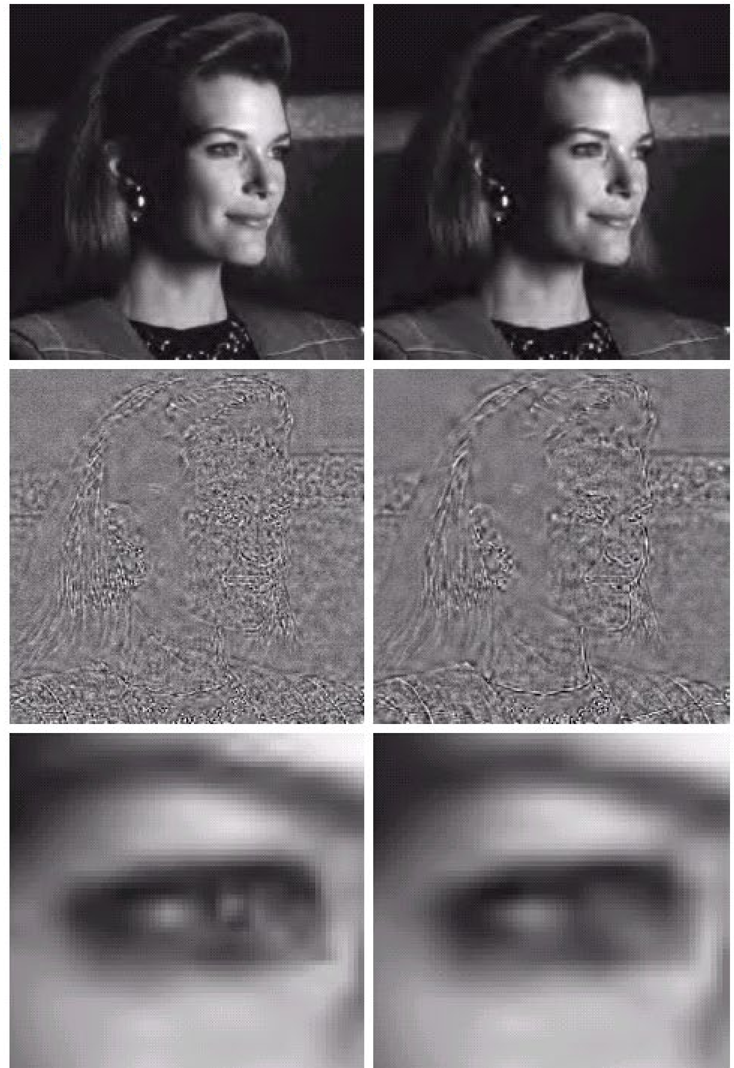
Application of Wavelets

■ Compression



Original
512x512
8bpp

0.074 bpp



0.048 bpp

Error
images

enlarged

From:
http://web.stanford.edu/class/ee398a/handouts/lectures/09-SubbandCoding.pdf

[Gonzalez, Woods, 2001]

Lecture 27: Wavelets to Modern Signal Processing

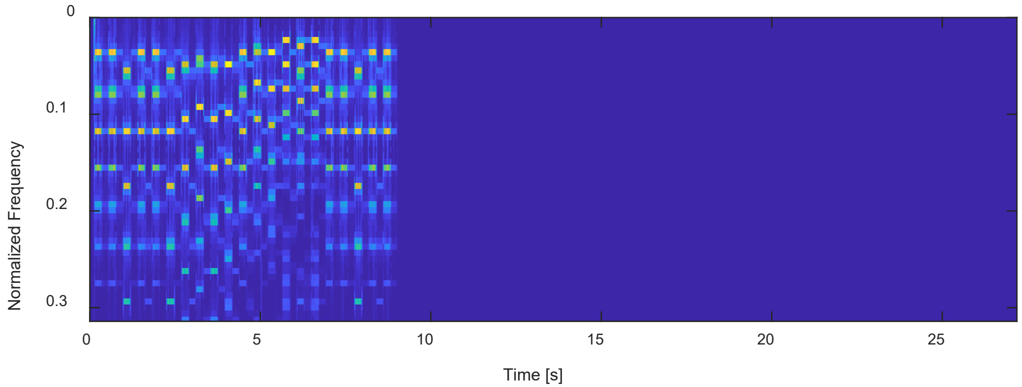
Foundations of Digital Signal Processing

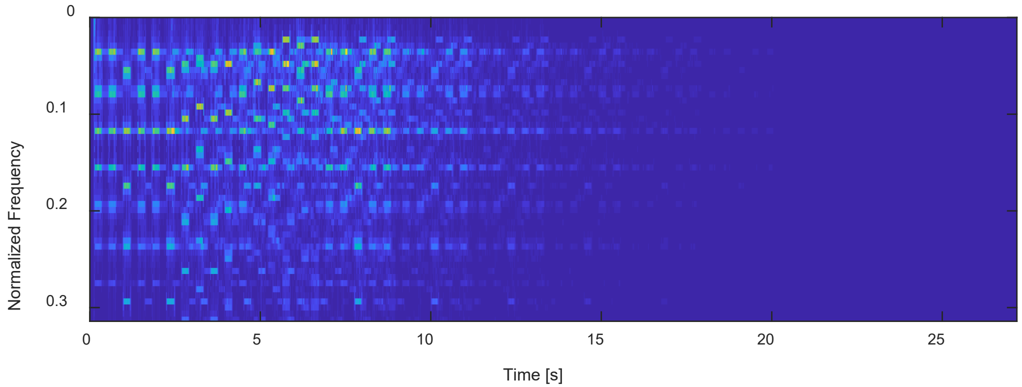
Outline

- Review of Filter Banks and Wavelets
- Applications of Wavelets
- **Applications of Filter Banks and Time-Frequency Representations**
- Modern Signal Processing: Vectors and Matrices
- Modern Signal Processing: Compressive Sensing
- Modern Signal Processing: Diagonalization
- Modern Signal Processing: Graph Signal Processing

Applications of Time-Frequency Rep.

- **Example:** **Example from our coding assignment**

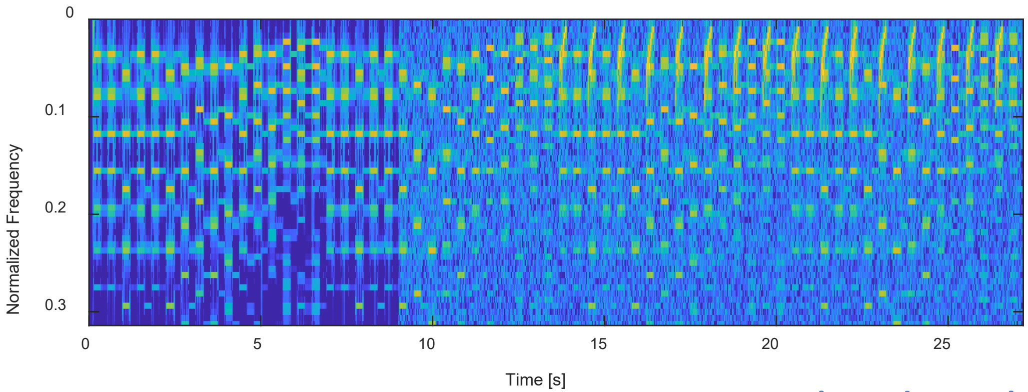




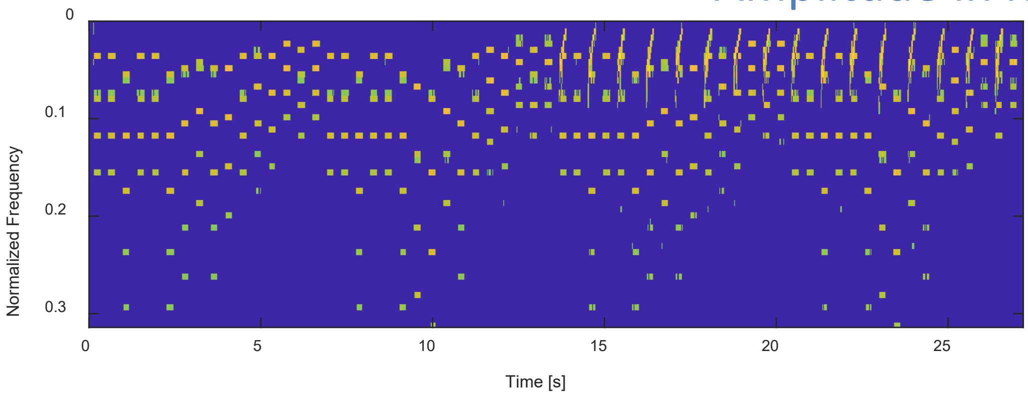
Echo / Reverb

Applications of Time-Frequency Rep.

- **Example:** **Example from our coding assignment**



Amplitude in log scale

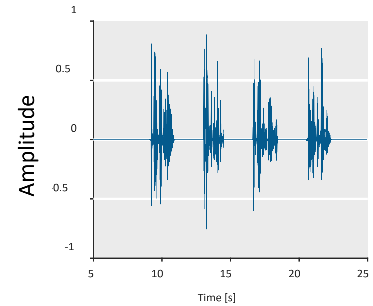


Threshold denoising

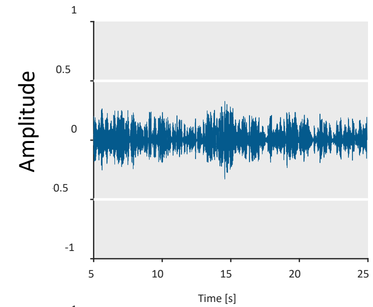
Low Latency Audio Denoising

Setup

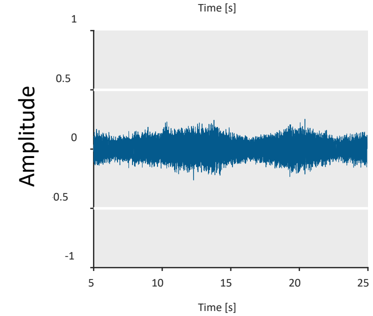
- **Testing Setup:**
  - Speech Signal – HINT
    (Hearing in Noise Testing)
  - Noise types (Input SNR: 12 dB, 6 dB)
    - ◊ Babble (background speech)
      - Frequency magnitude shaped as speech signal
    - ◊ Car noise
      - Frequency magnitude shaped as speech signal
    - ◊ White noise
      - Frequency magnitude shaped as speech signal



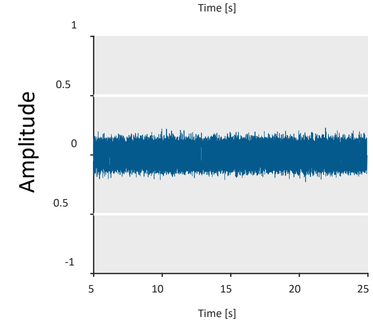
HINT speech



Babble noise



Car noise



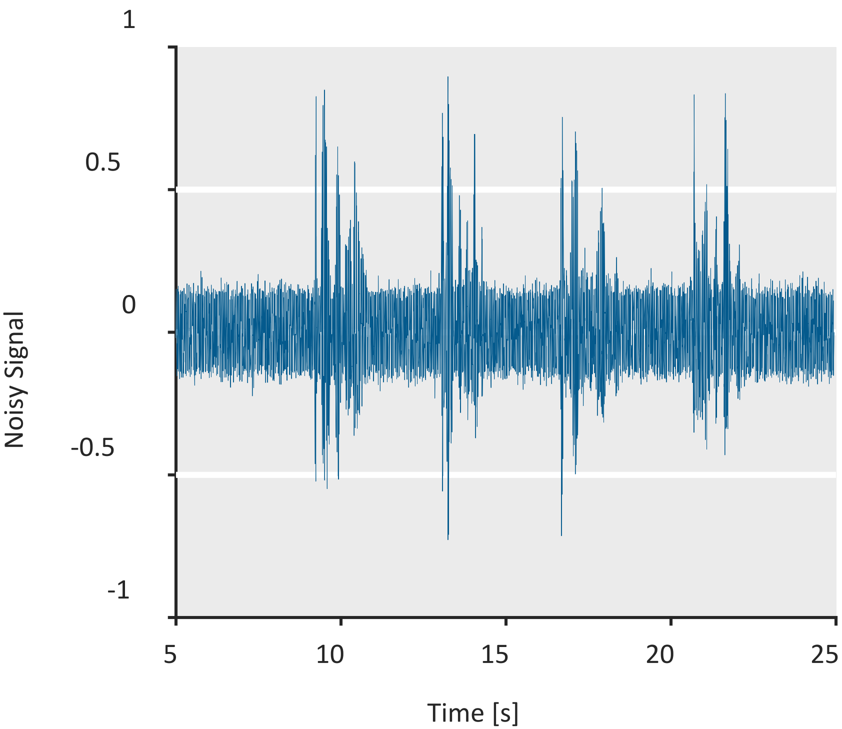
White noise

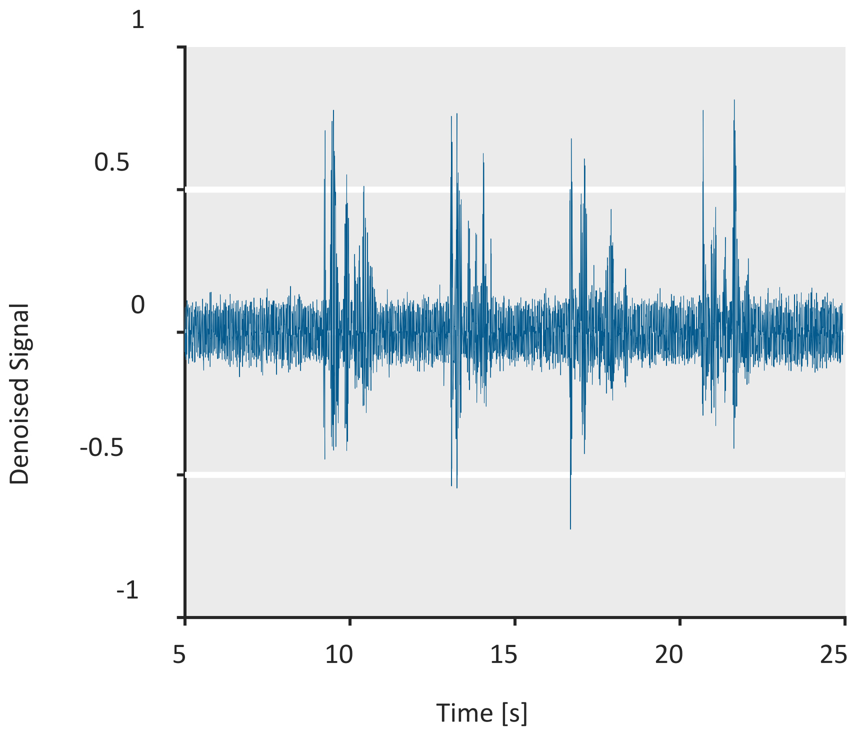
Results:

- **HINT speech**
- **White noise**
  - Shaped
- **Noisy speech vs Denoised speech**
  - Frequency domain vs our approach

Frequency Domain Approach

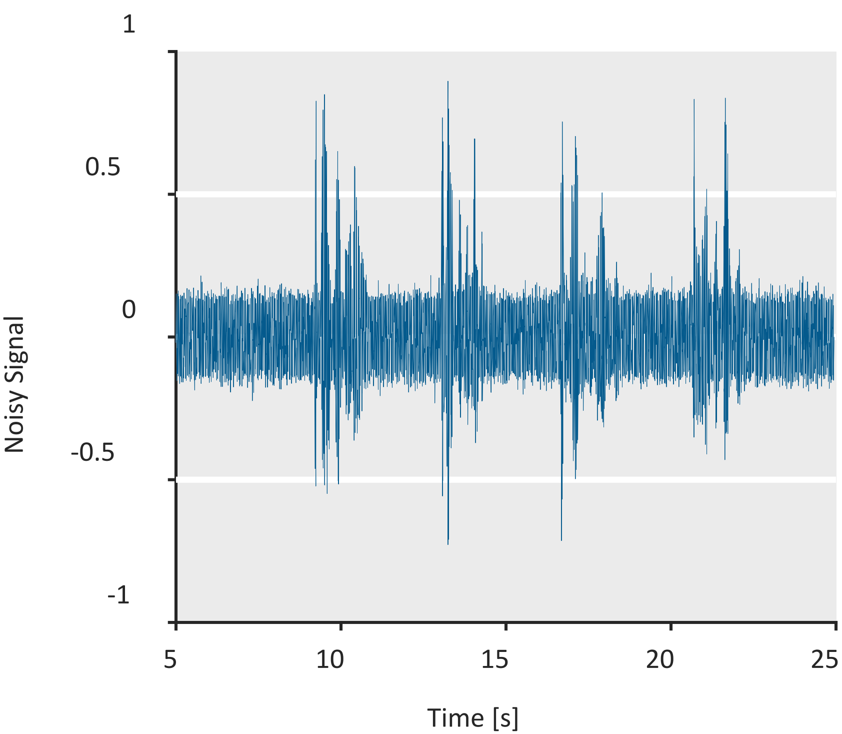
- **Input SNR 6 dB, Frame size = 16 samples**

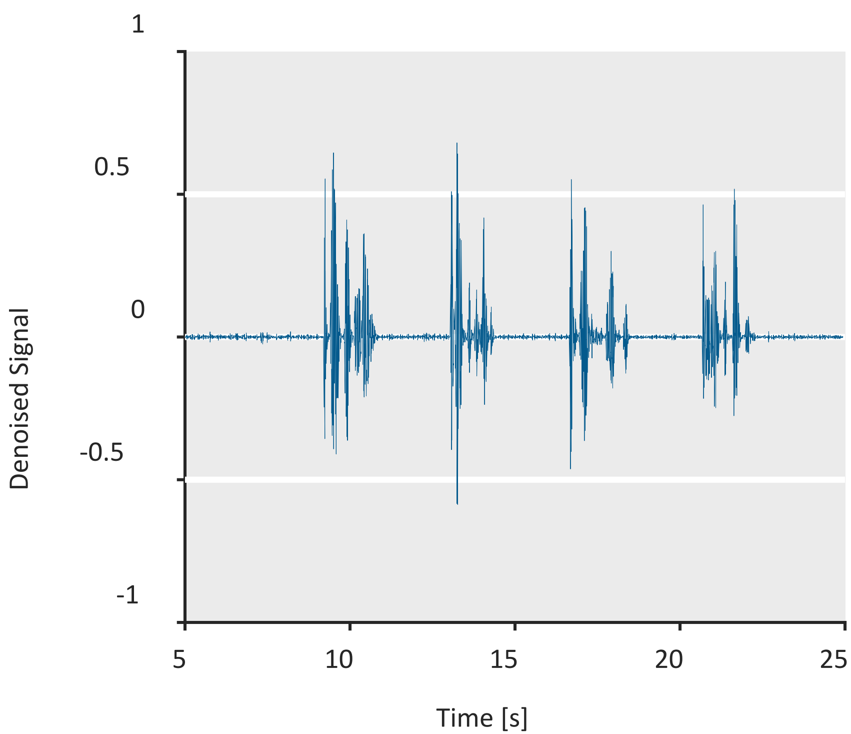




Our Approach

- **Input SNR 6 dB, Filter bank with 16 filters**



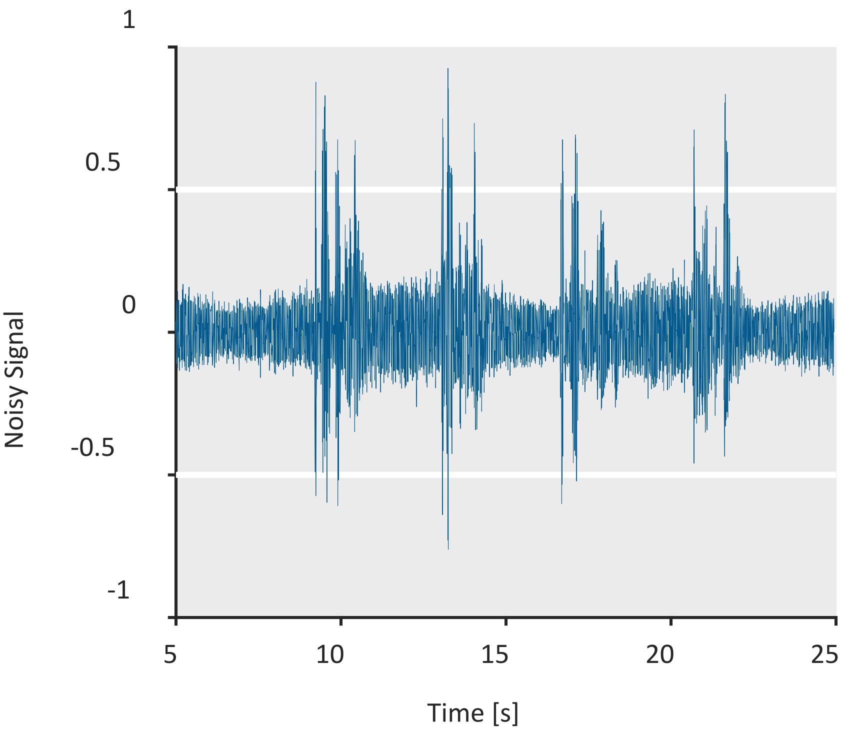


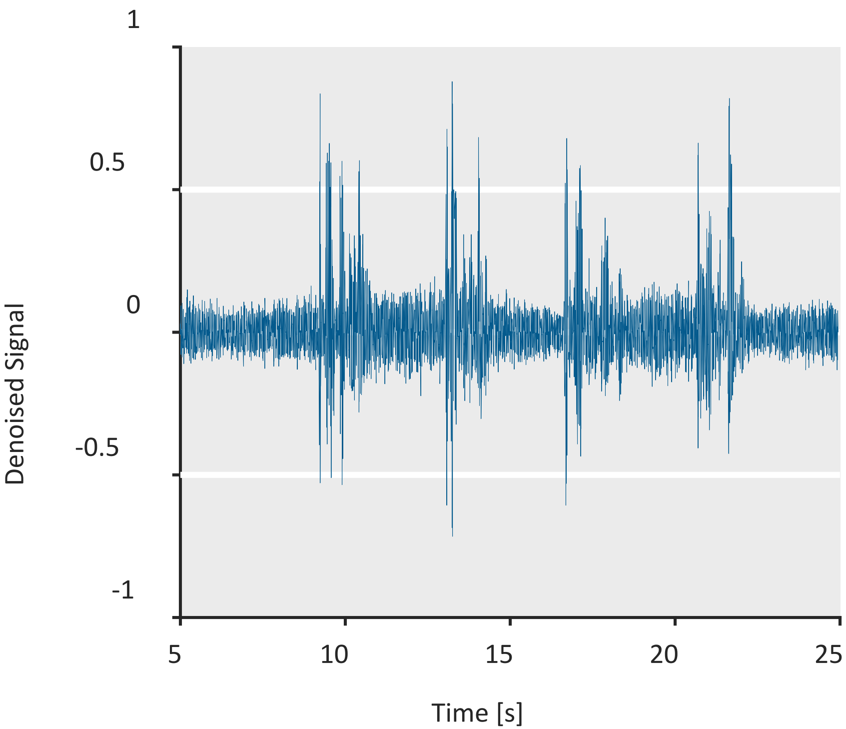
Results:

- **HINT speech**
- **Car noise**
  - Shaped
- **Noisy speech vs Denoised speech**
  - Frequency domain vs our approach

Frequency Domain Approach

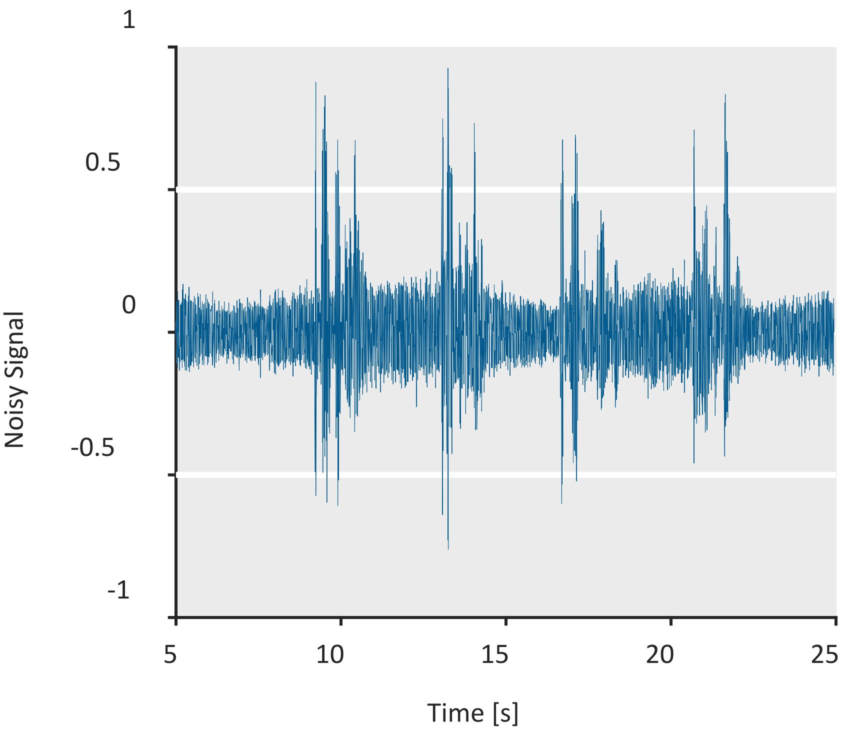
- **Input SNR 6 dB, Frame size = 16 samples**

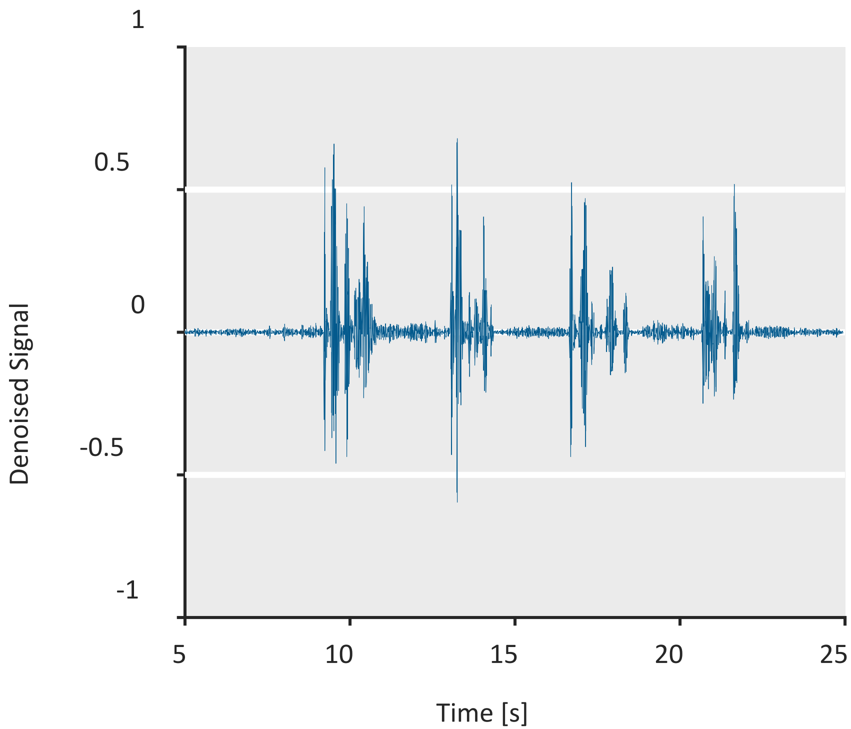




Our Approach

- **Input SNR 6 dB, Filter bank with 16 filters**



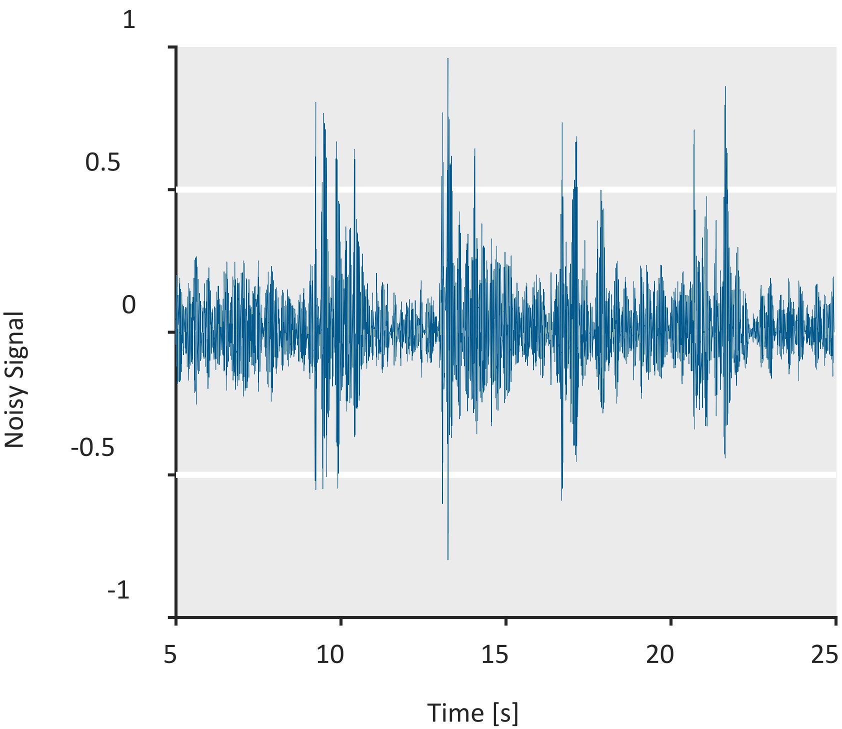


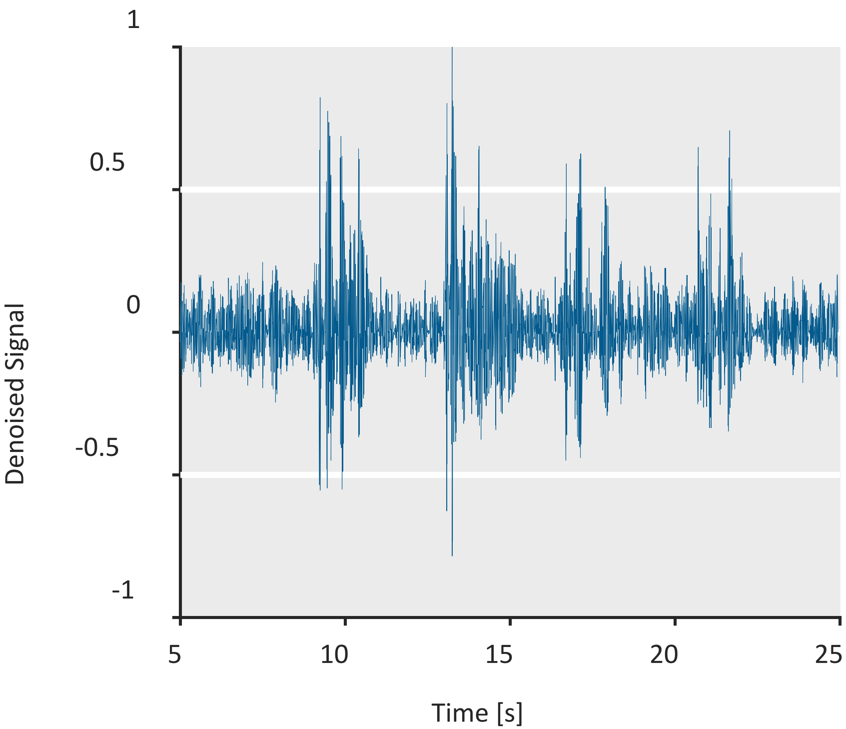
Results

- **HINT speech**
- **Babble noise**
  - Shaped
- **Noisy speech vs Denoised speech**
  - Frequency domain vs our approach

Frequency Domain Approach

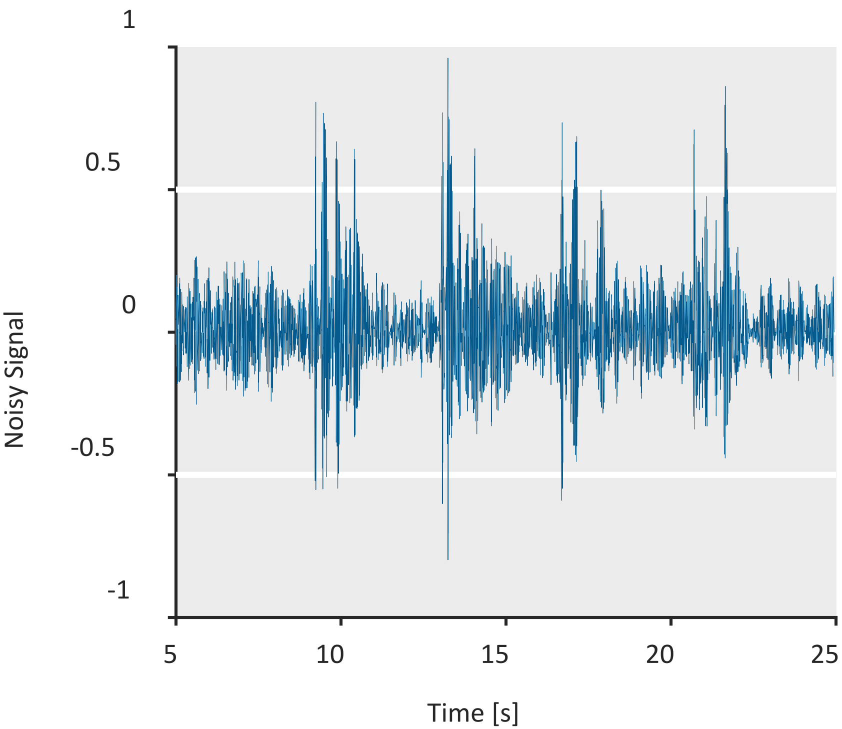
- **Input SNR 6 dB, Frame size = 16 samples**

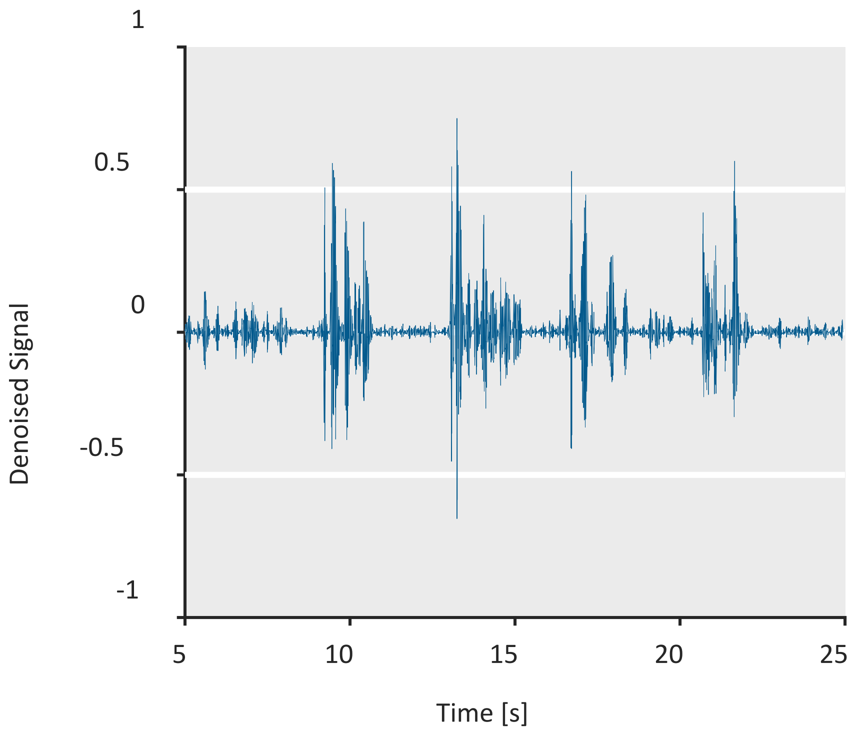




Our Approach

- **Input SNR 6 dB, Filter bank with 16 filters**

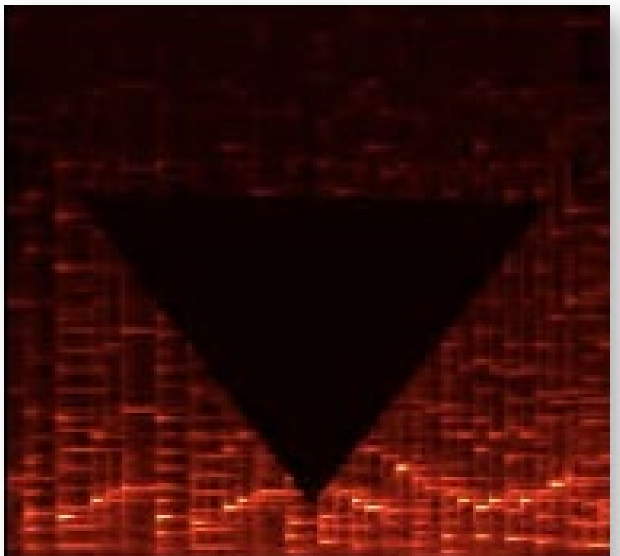




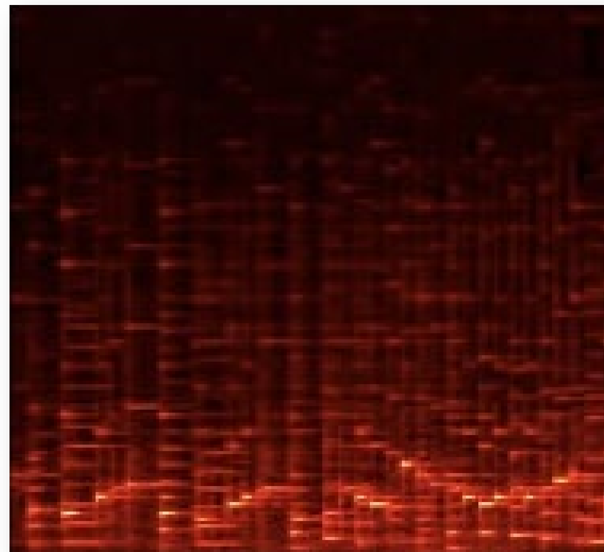
Learning from Audio

Applications of Time-Frequency Rep.

- **Example:** **From Paris Smaragdis at the University of Illinois**
  - https://paris.cs.illinois.edu/demos/index.html



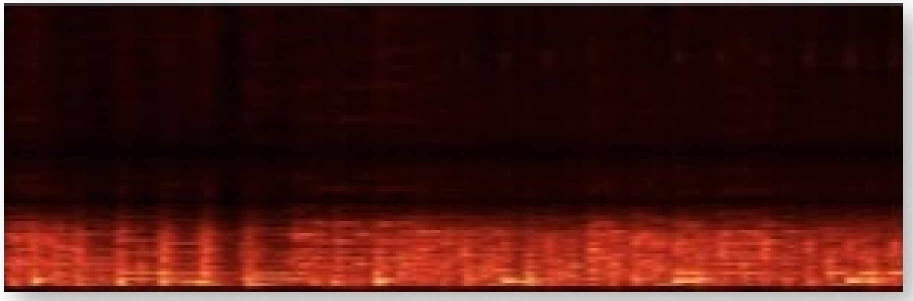
Missing Data



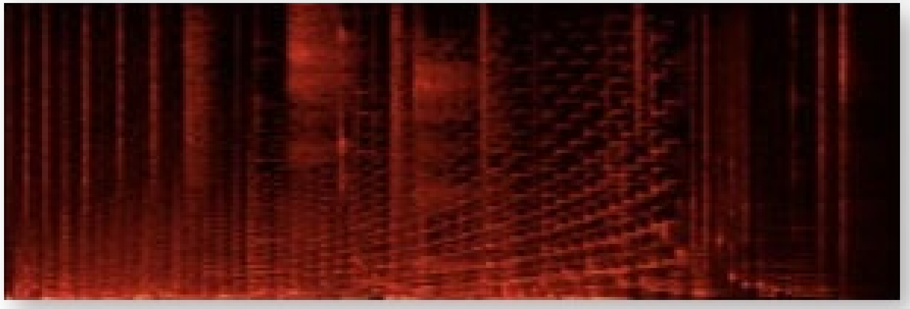
Reconstructed Data

Applications of Time-Frequency Rep.

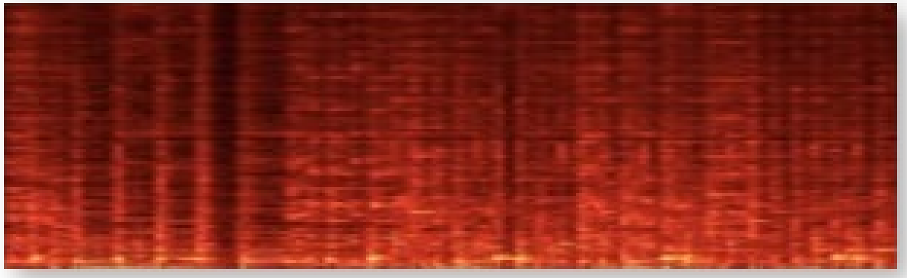
- **Example:** **From Paris Smaragdis at the University of Illinois**
  - https://paris.cs.illinois.edu/demos/index.html



Missing Data



Training Data

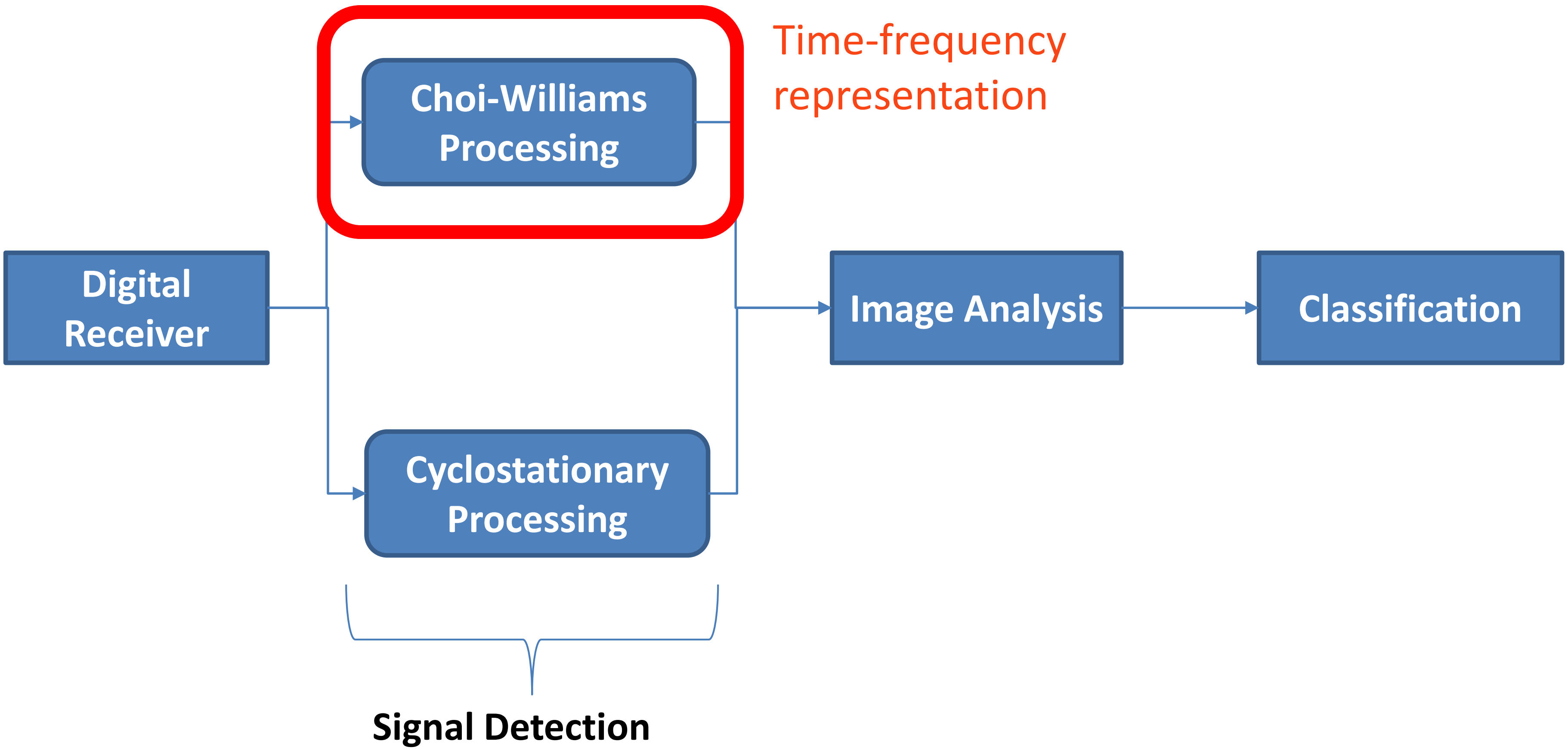


Reconstructed Music

Radar Signal Processing

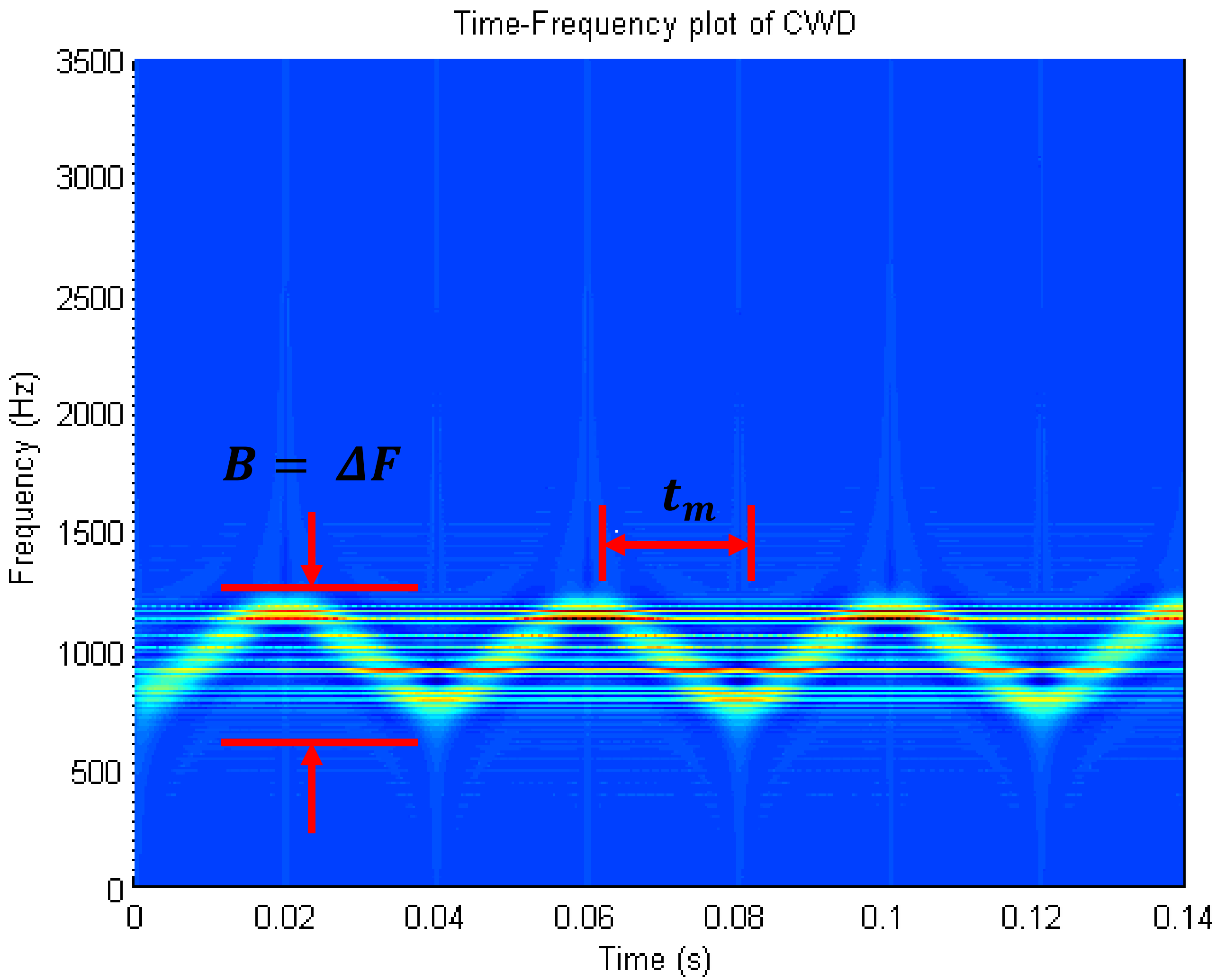
Applications of Time-Frequency Rep.

- **Goal: To detect low probability of intercept radar signals**



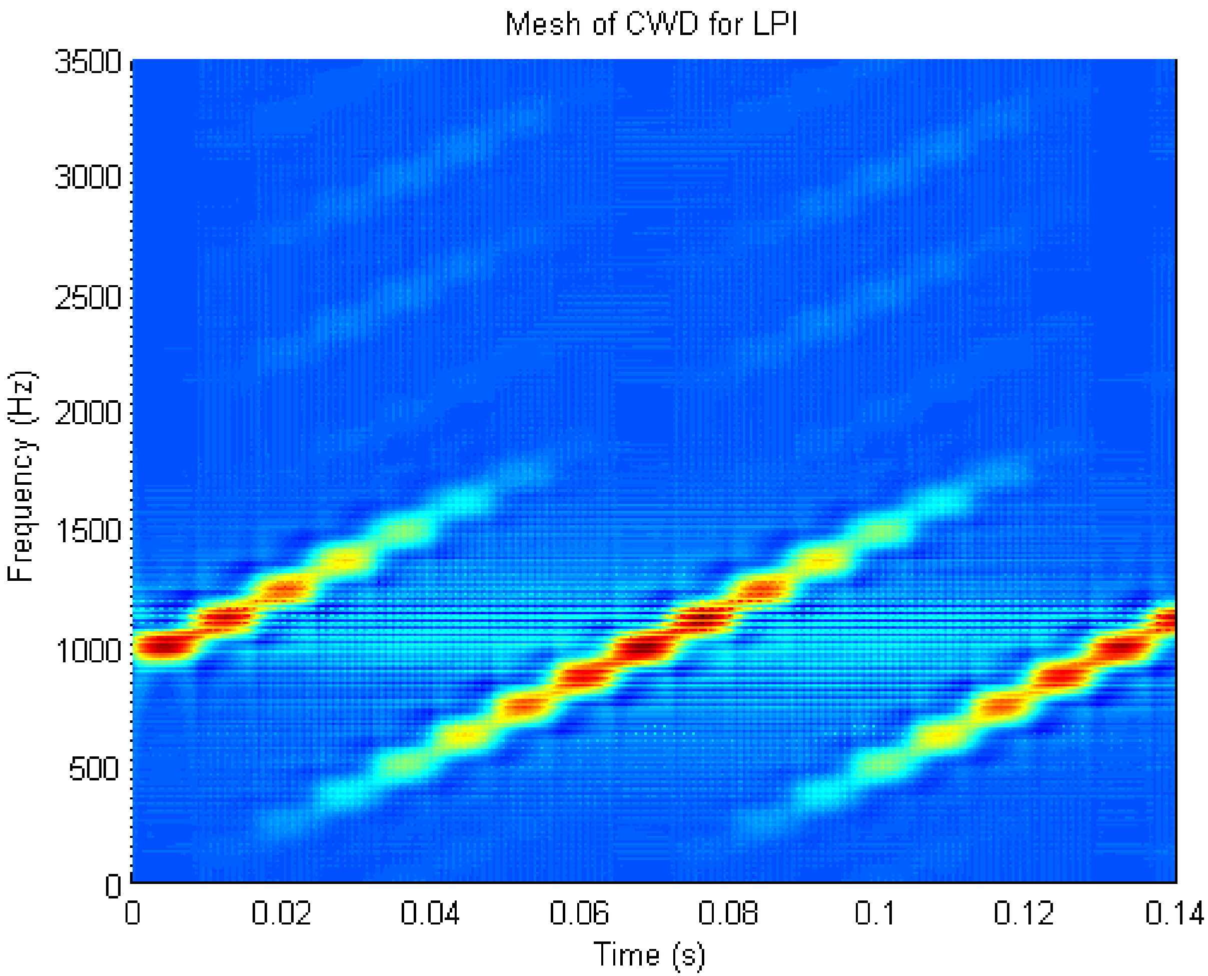
Applications of Time-Frequency Rep.

- **Radar Signals: Frequency Modulation Continuous Waveform**



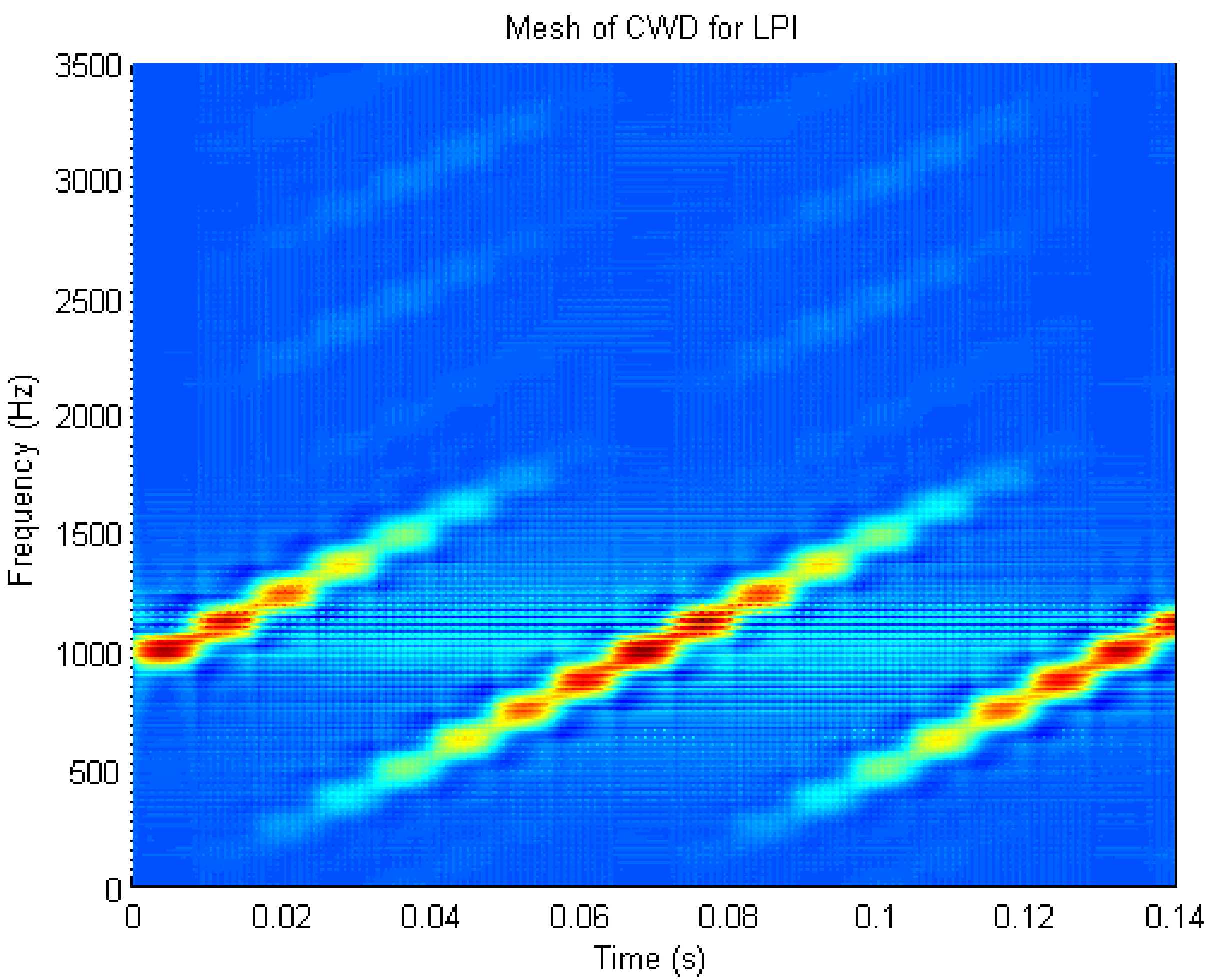
Applications of Time-Frequency Rep.

- **Radar Signals: Frank Codes**



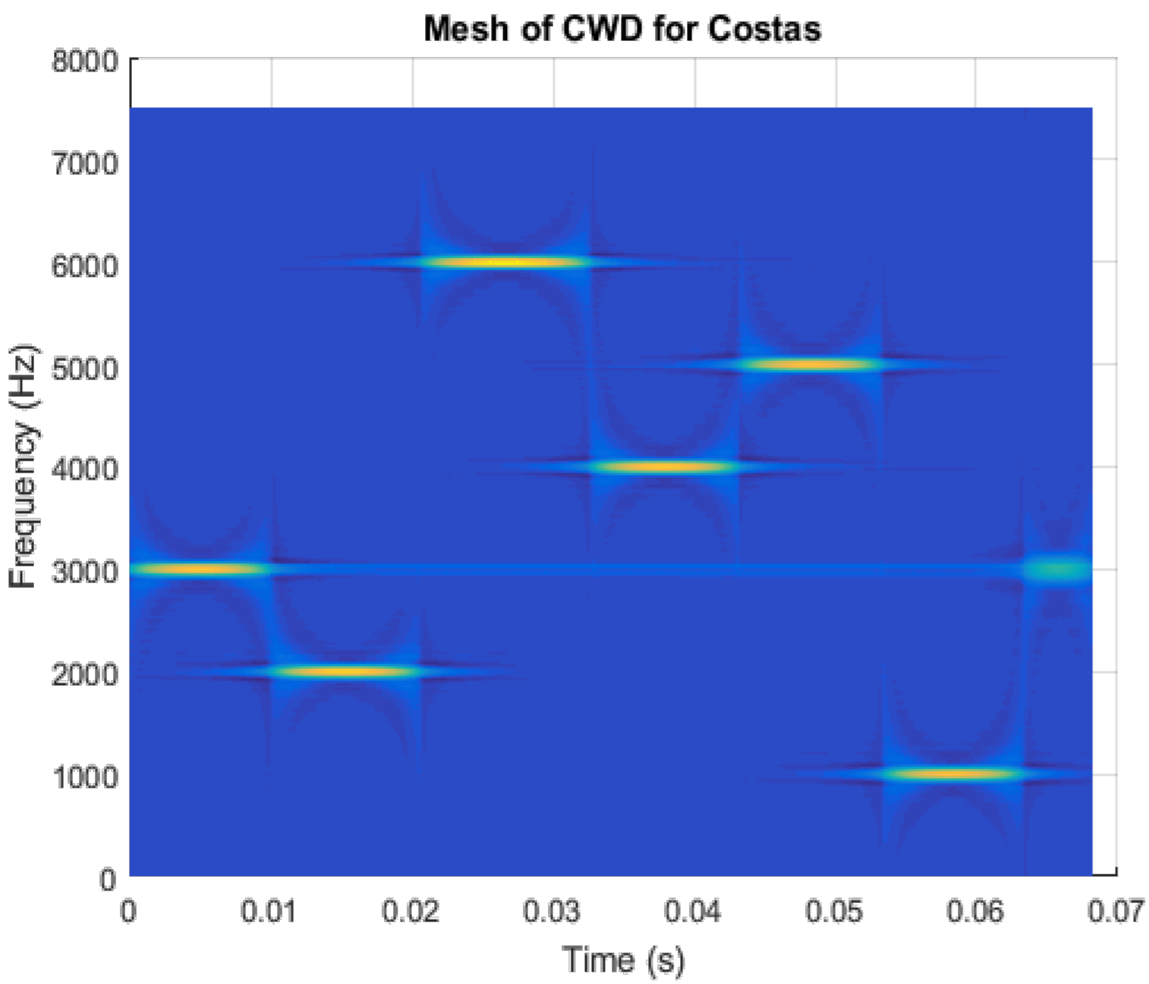
Applications of Time-Frequency Rep.

- **Radar Signals: Frank Codes**



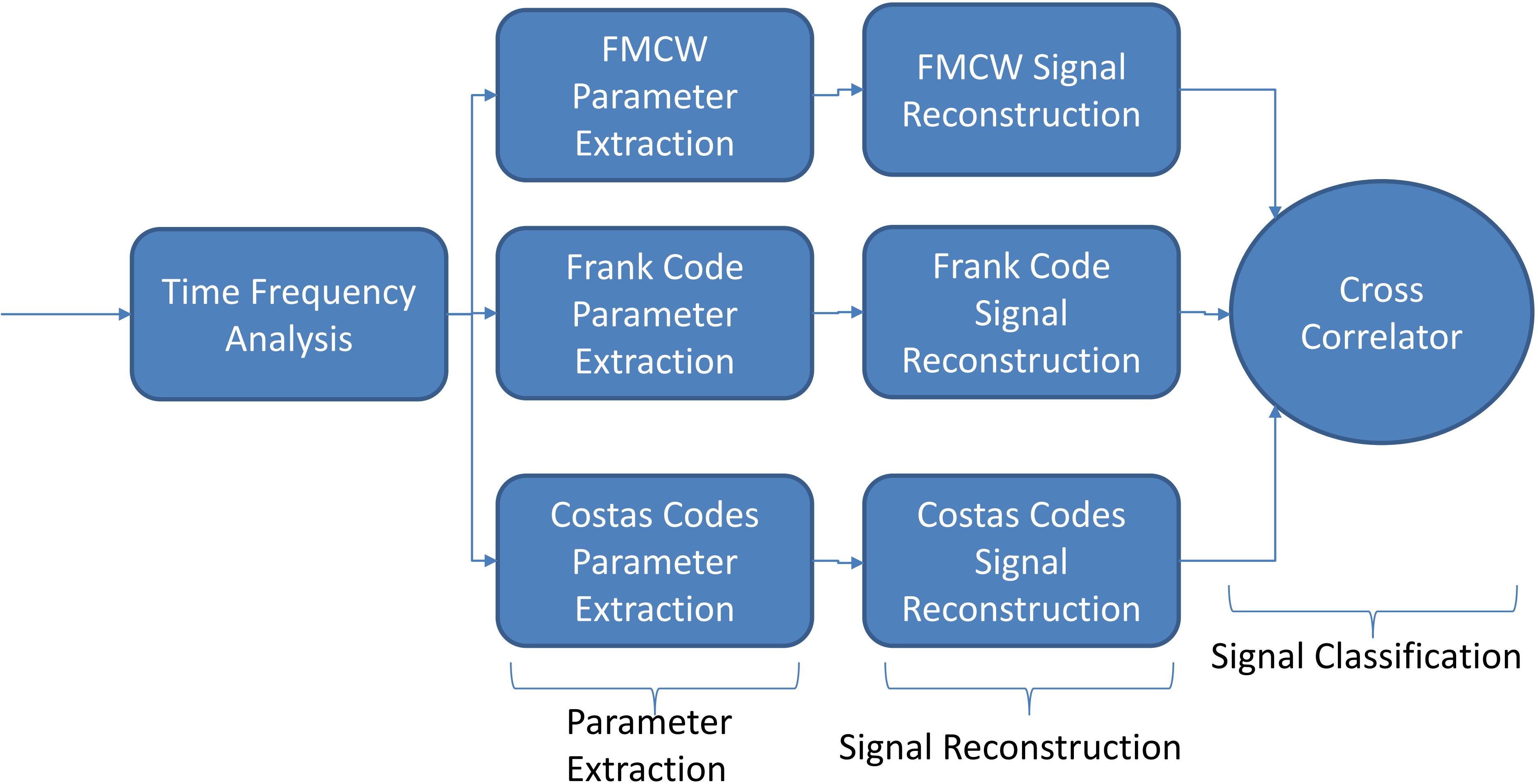
Applications of Time-Frequency Rep.

- **Radar Signals: Costas Codes**



Applications of Time-Frequency Rep.

- **Goal:** **To detect low probability of intercept radar signals**

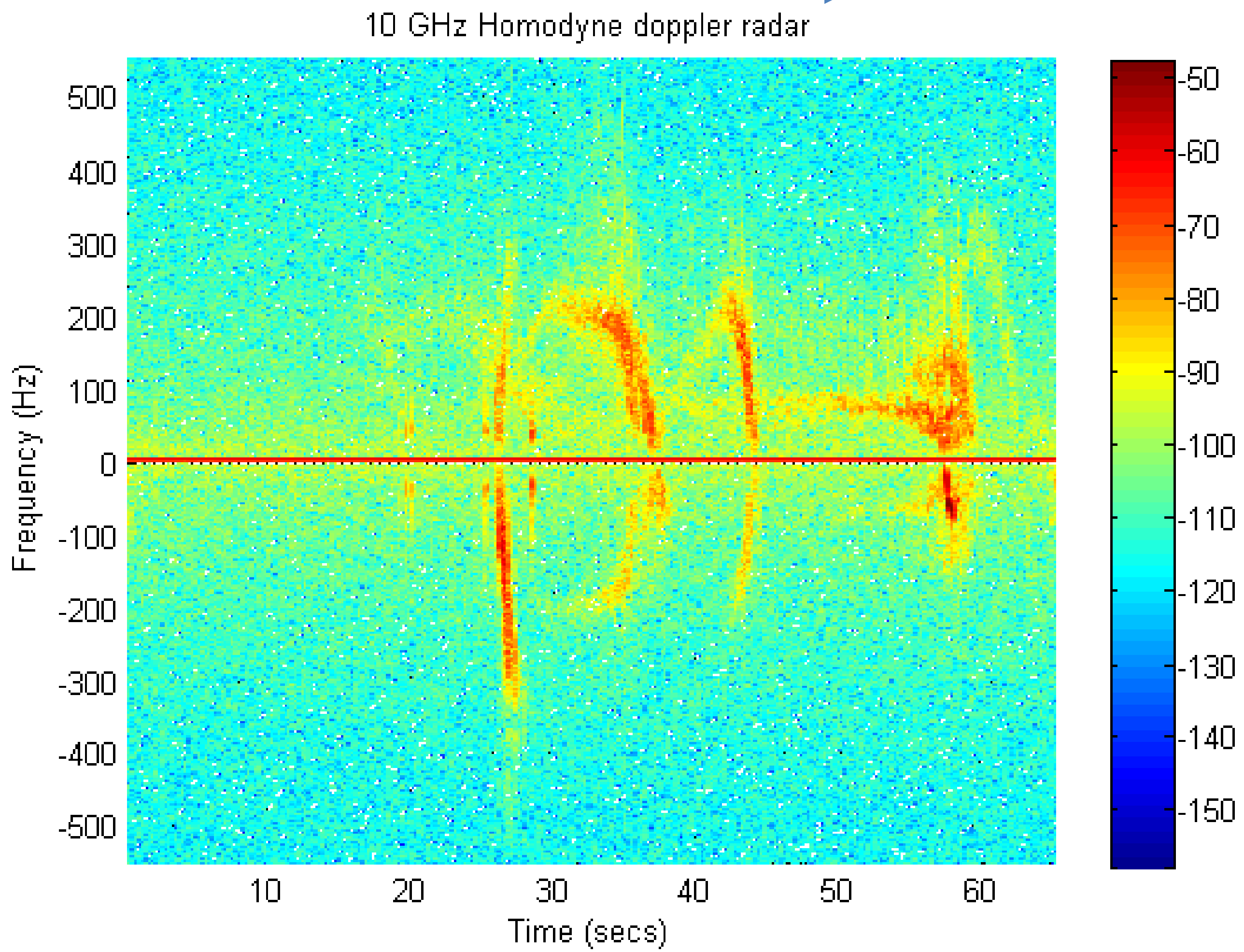


Applications of Time-Frequency Rep.

- **Doppler Radar**

Distance from radar

Object traveling toward radar



Object traveling away from radar

From: http://home.earthlink.net/~w6rmk/radar10g4.htm

Applications of Time-Frequency Rep.

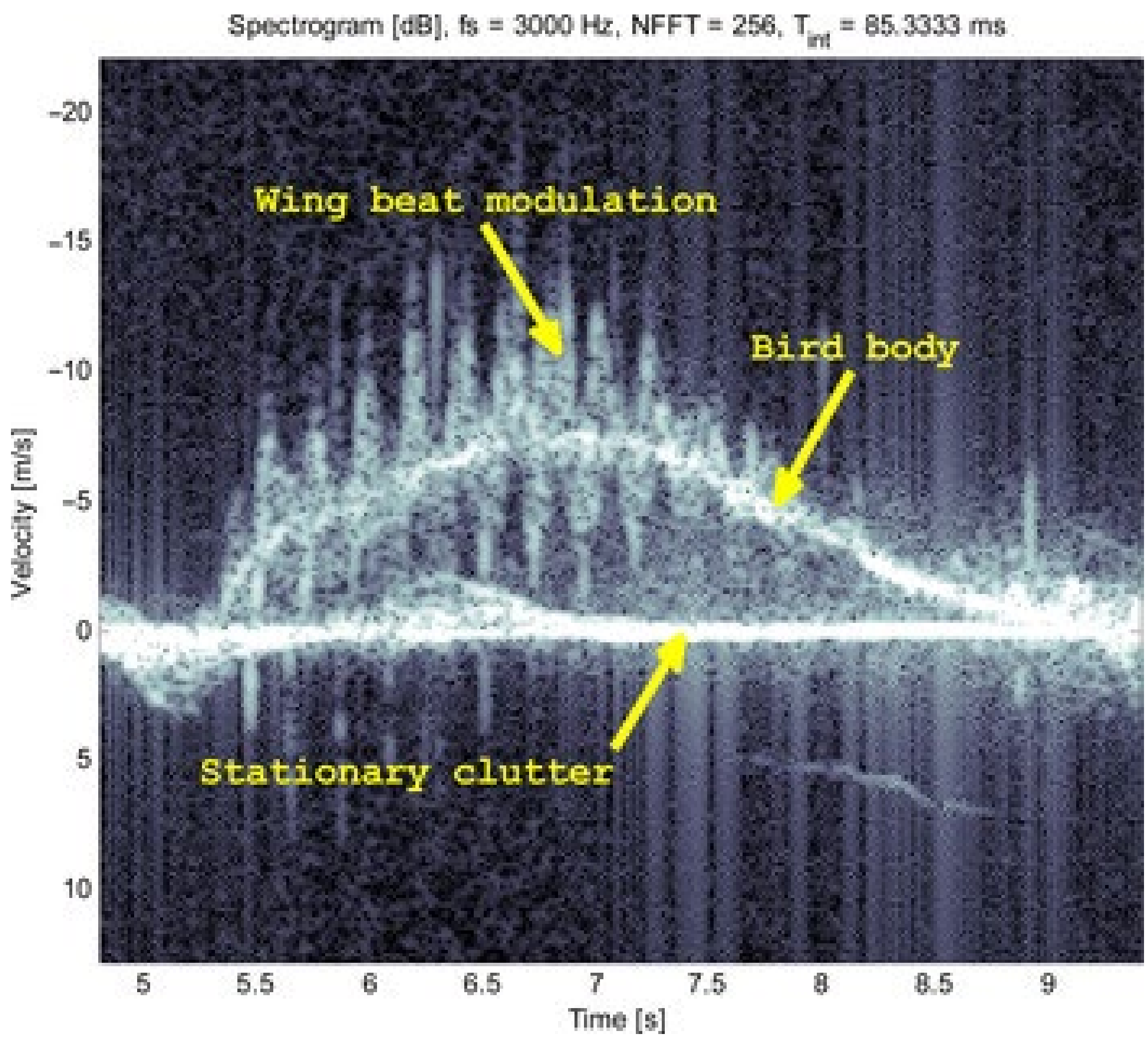
- **Doppler Radar**

From: Ronny I.A. Harmanny, Jacco J.M. de Wit and Gilles Premel-Cabic,"Radar micro-Doppler mini-UAV classification using spectrograms and cepstrograms,"International Journal of Microwave and Wireless TechnologiesVolume 7 special Issue 3-4, 2014.

Distance from radar

Object traveling away from radar [negative frequencies]

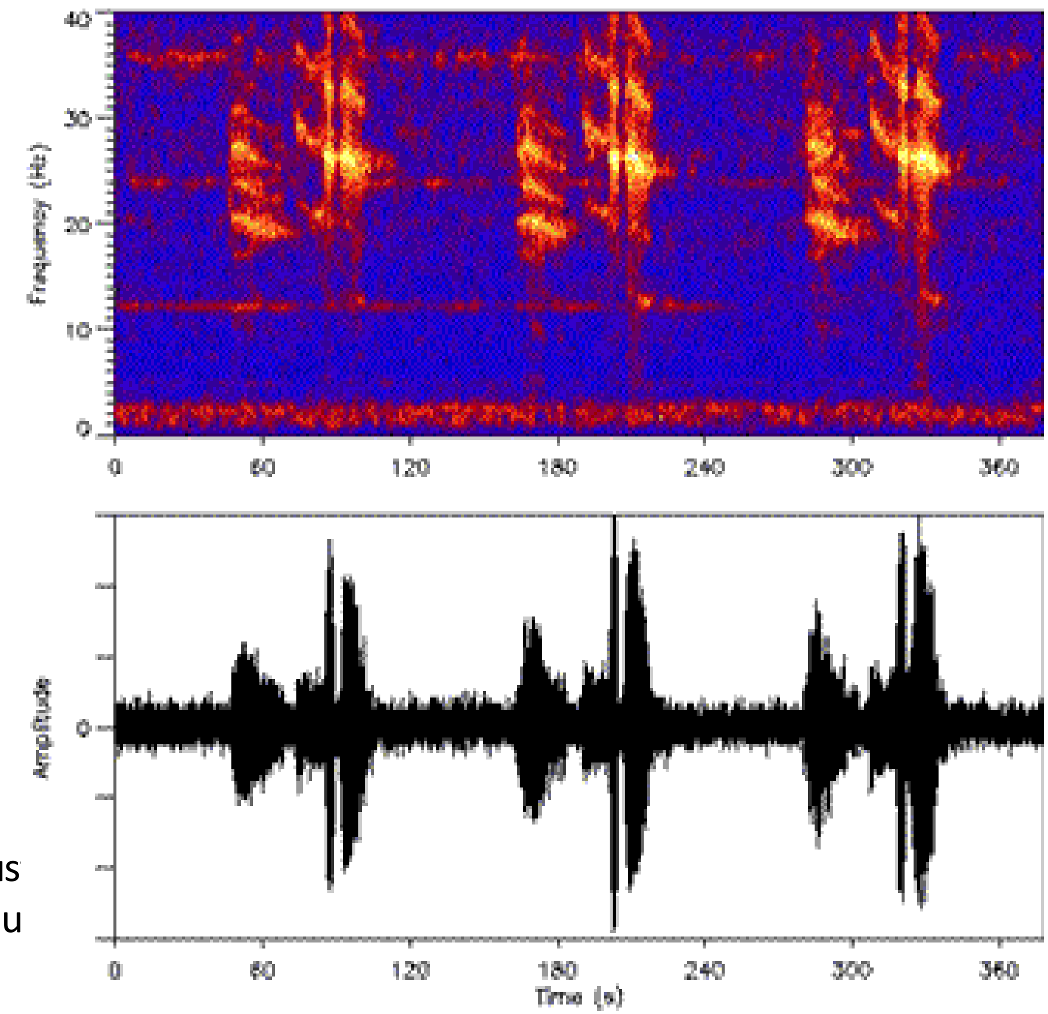
Object traveling toward radar [positive frequencies]



Analyzing time-domain data

Applications of Time-Frequency Rep.

- **Analyzing Whale Calls**
  - South Pacific blue whale call

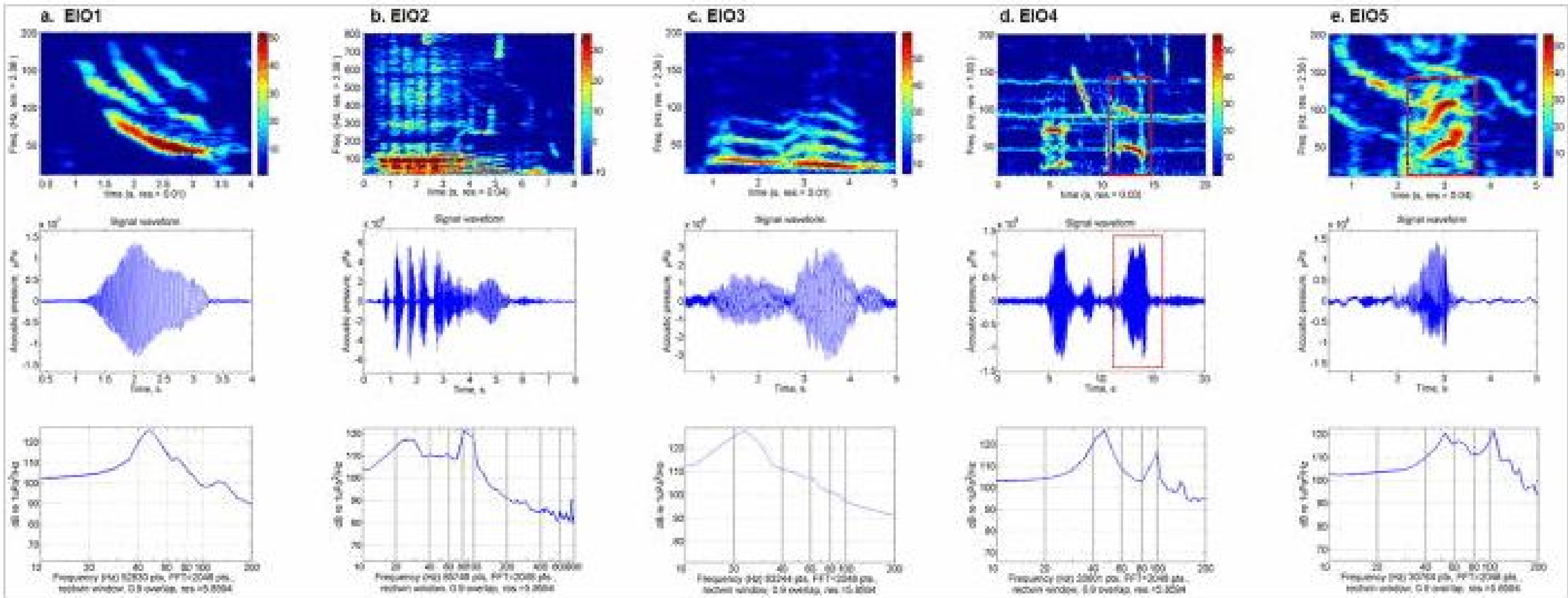


  - **From:** https://www.pmel.noaa.gov/acoustics/whales/sounds/sounds_spacblue.html

Applications of Time-Frequency Rep.

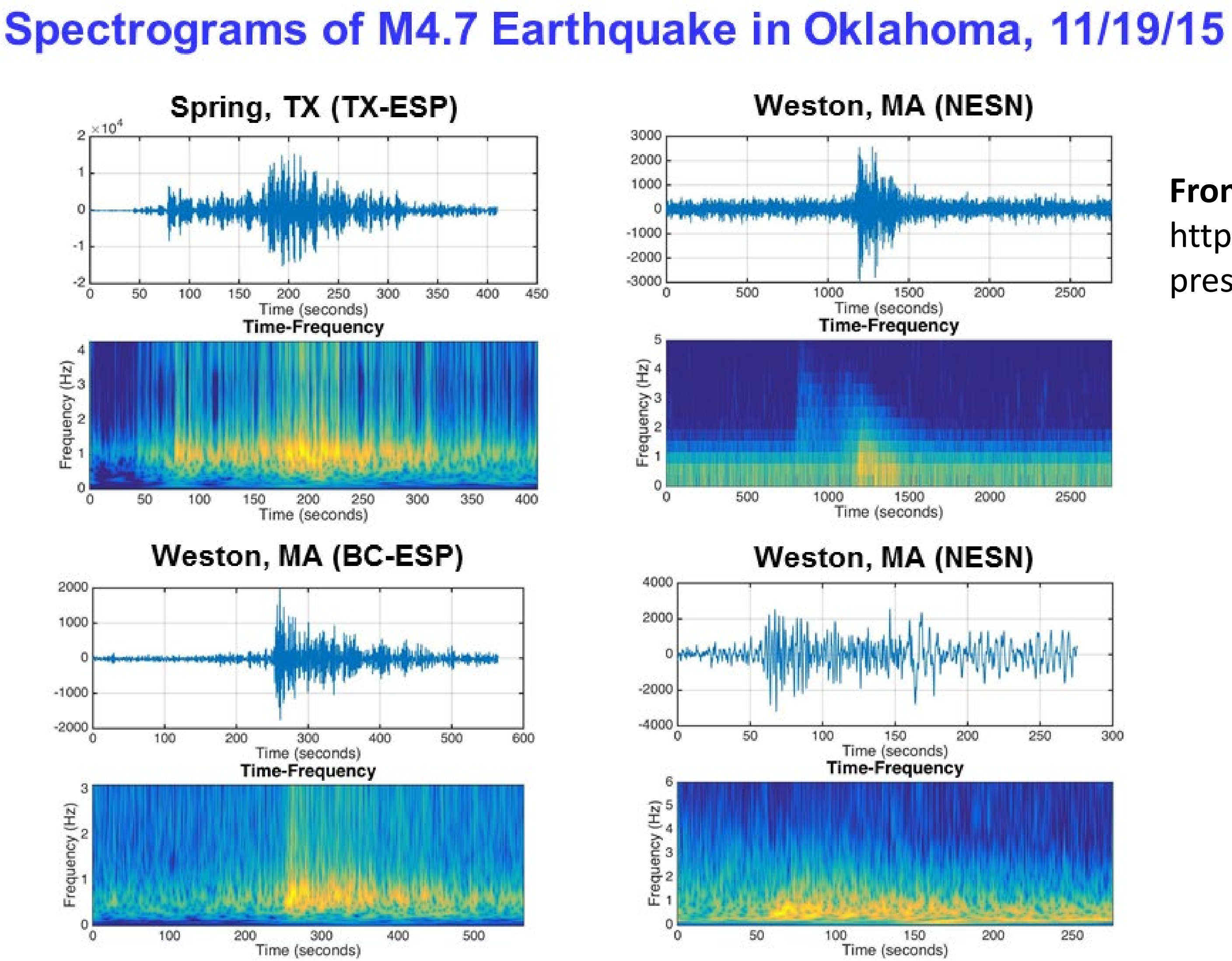
Analyzing Whale Calls

- Five Different Vocalizations of Pygmy Blue Whales



- **From:** https://souwest.org/2014/06/16/what-did-you-say-new-paper-on-pymy-blue-whale-acoustics/

Applications of Time-Frequency Rep.



From:
https://akafka.wordpress.com/

Lecture 27: Wavelets to Modern Signal Processing

Foundations of Digital Signal Processing

Outline

- Review of Filter Banks and Wavelets
- Applications of Wavelets
- Applications of Filter Banks and Time-Frequency Representations
- **Modern Signal Processing: Vectors and Matrices**
- Modern Signal Processing: Diagonalization
- Modern Signal Processing: Compressive Sensing
- Modern Signal Processing: Compressive Sensing

Filters

- **Linear operators are linear systems!**
  - Any linear DSP tool can be expressed as a linear operator or matrix.
  - $\mathbf{y} = \mathbf{H}\,\mathbf{x}$, $\mathbf{x}, \mathbf{y}$ are vectors and $\mathbf{H}$ is a matrix

- **Filter Properties**
  - Causal
  - Anti-causal
  - Two-sided
  - Finite Impulse Response
  - Infinite Impulse Response
  - BIBO Stablity

Filters

- **Linear operators are linear systems!**
  - Any linear DSP tool can be expressed as a linear operator or matrix.
  - $\mathbf{y} = \mathbf{H}\,\mathbf{x}$, $\mathbf{x}, \mathbf{y}$ are vectors and $\mathbf{H}$ is a matrix

Input signal

System / Filter

Output signal

- **Filter Properties**
  - Causal
  - Anti-causal
  - Two-sided
  - Finite Impulse Response
  - Infinite Impulse Response
  - BIBO Stablity

Filters

- **Memoryless System**
  - A filter is memoryless when the output is only dependent on the present time input.

$$\mathbf{y} = \underbrace{\begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & h[0] & 0 & 0 & 0 & \ddots \\ \ddots & 0 & h[0] & 0 & 0 & \ddots \\ \ddots & 0 & 0 & h[0] & 0 & \ddots \\ \ddots & 0 & 0 & 0 & h[0] & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}}_{\text{Diagonal Matrix (of infinite size)}} \begin{bmatrix} \vdots \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \mathbf{Hx}$$

Filters

- **Causal**
  - A filter is causal when the output is only dependent on the present and past time inputs.

$$\mathbf{y} = \underbrace{\begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & h[0] & 0 & 0 & 0 & \ddots \\ \ddots & h[1] & h[0] & 0 & 0 & \ddots \\ \ddots & h[2] & h[1] & h[0] & 0 & \ddots \\ \ddots & h[3] & h[2] & h[1] & h[0] & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}}_{\text{Left Triangular Matrix (of infinite size)}} \begin{bmatrix} \vdots \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \mathbf{Hx}$$

Filters

- **Shift invariant**
  - A filter is shift (or time) invariant when shifting the input will only shift the output.

$$\mathbf{y} = \underbrace{\begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & h[0] & h[-1] & h[-2] & h[-3] & \ddots \\ \ddots & h[1] & h[0] & h[-1] & h[-2] & \ddots \\ \ddots & h[2] & h[1] & h[0] & h[-1] & \ddots \\ \ddots & h[3] & h[2] & h[1] & h[0] & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}}_{\text{Toeplitz Matrix (of infinite size)}} \begin{bmatrix} \vdots \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \mathbf{Hx}$$

Filters

- **BIBO Stability**
  - A filter is bounded input, bounded output (BIBO) stable when a bounded input produces a bounded outp

Each row is absolutely summable

$$\mathbf{y} = \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & h[0] & h[-1] & h[-2] & h[-3] & \ddots \\ \ddots & h[1] & h[0] & h[-1] & h[-2] & \ddots \\ \ddots & h[2] & h[1] & h[0] & h[-1] & \ddots \\ \ddots & h[3] & h[2] & h[1] & h[0] & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = \mathbf{Hx}$$

Representations

- **This also allows us to talk about representations for signals**
  - $\mathbf{y} = \mathbf{F}\,\mathbf{x}$, $\mathbf{x}, \mathbf{y}$ are vectors and $\mathbf{F}$ is a matrix

Time signal

DFT matrix

Frequency signal

Representations

- **This also allows us to talk about representations for signals**
  - $\mathbf{y} = \mathbf{\Phi}\,\mathbf{x}$, $\mathbf{x}, \mathbf{y}$ are vectors and $\mathbf{\Phi}$ is a matrix

    Time signal

    Basis matrix

    Frequency signal

  - Much of modern signal focuses on how to best represent signals / data and how we can exploit that in our processing

Lecture 27: Wavelets to Modern Signal Processing

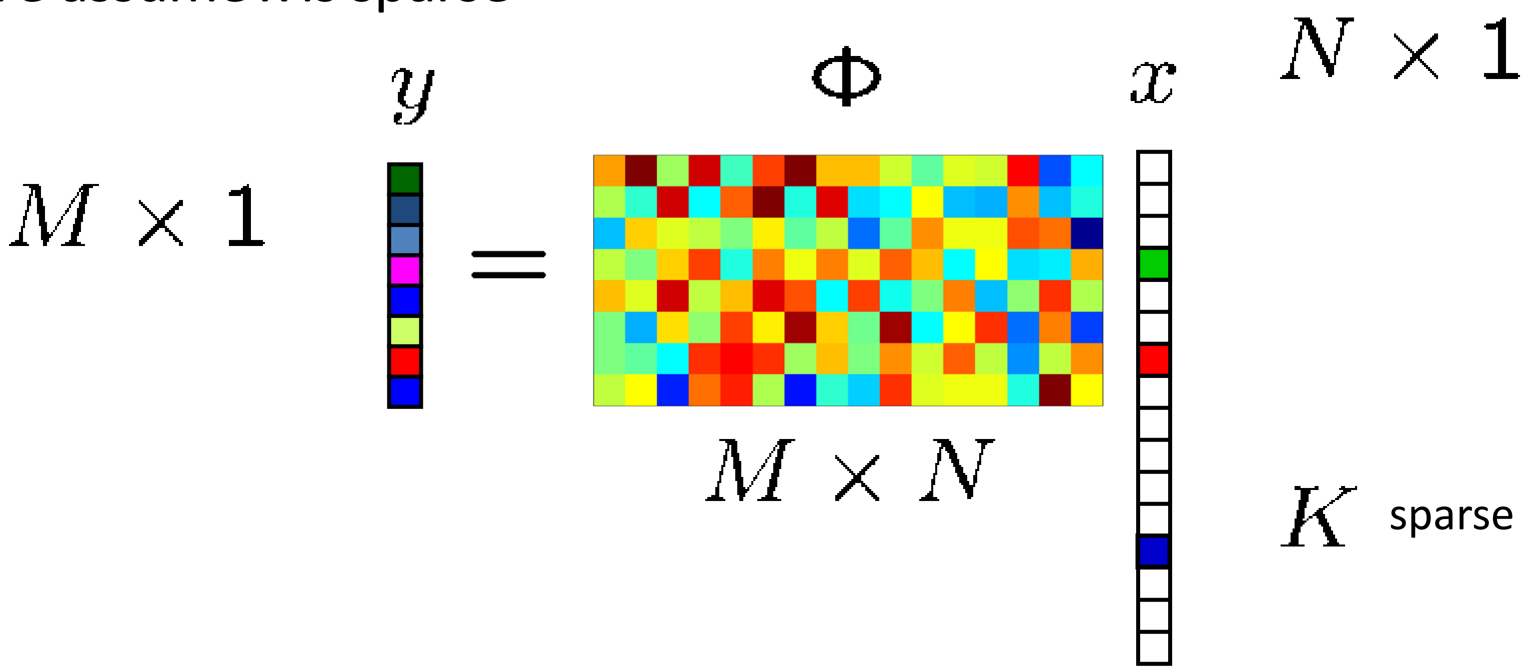
Foundations of Digital Signal Processing

Outline

- Review of Filter Banks and Wavelets
- Applications of Wavelets
- Applications of Filter Banks and Time-Frequency Representations
- Modern Signal Processing: Vectors and Matrices
- **Modern Signal Processing: Compressive Sensing**
- Modern Signal Processing: Diagonalization
- Modern Signal Processing: Graph Signal Processing

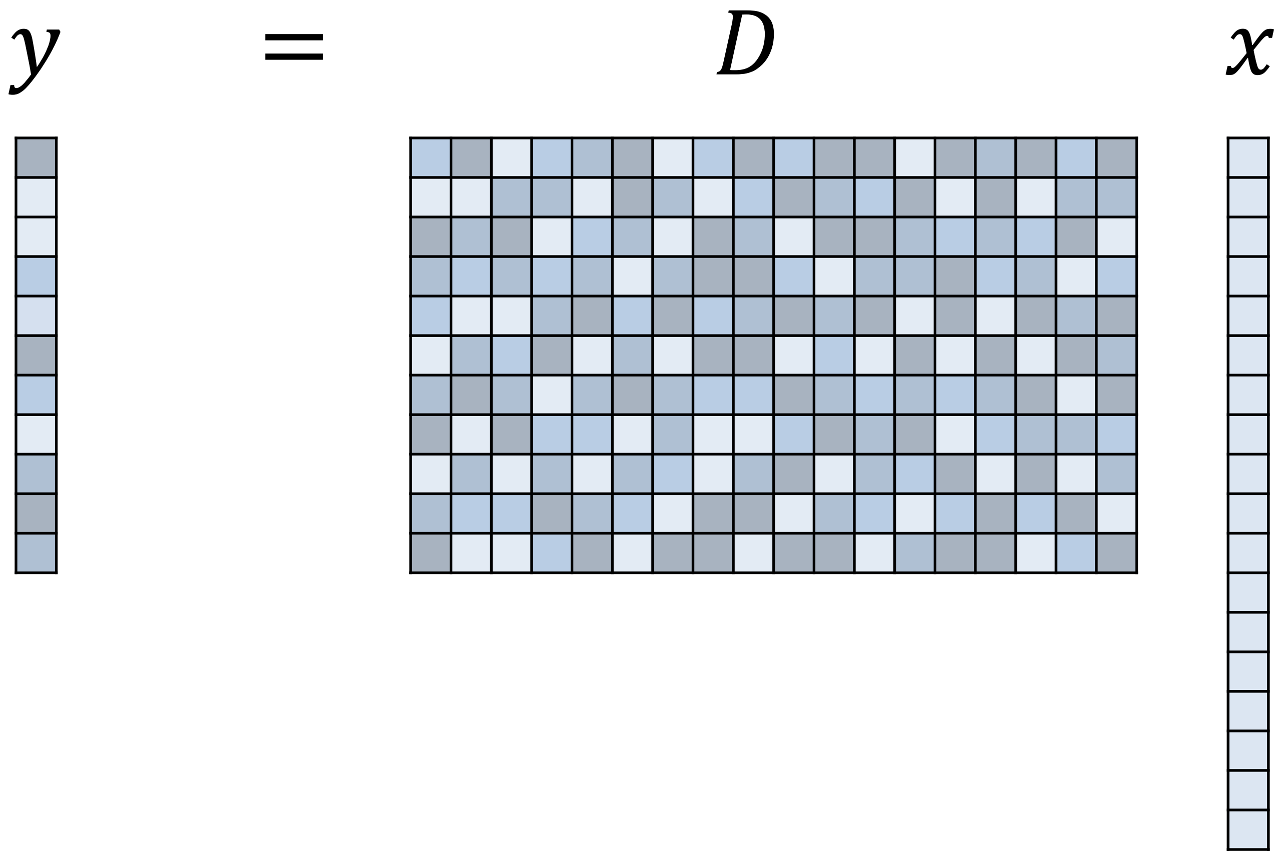
Sparse Recovery Conditions

- **An assumption about how we can represent *y***
  - We assume x is sparse



- **Question:** **Why is this a common setup?**

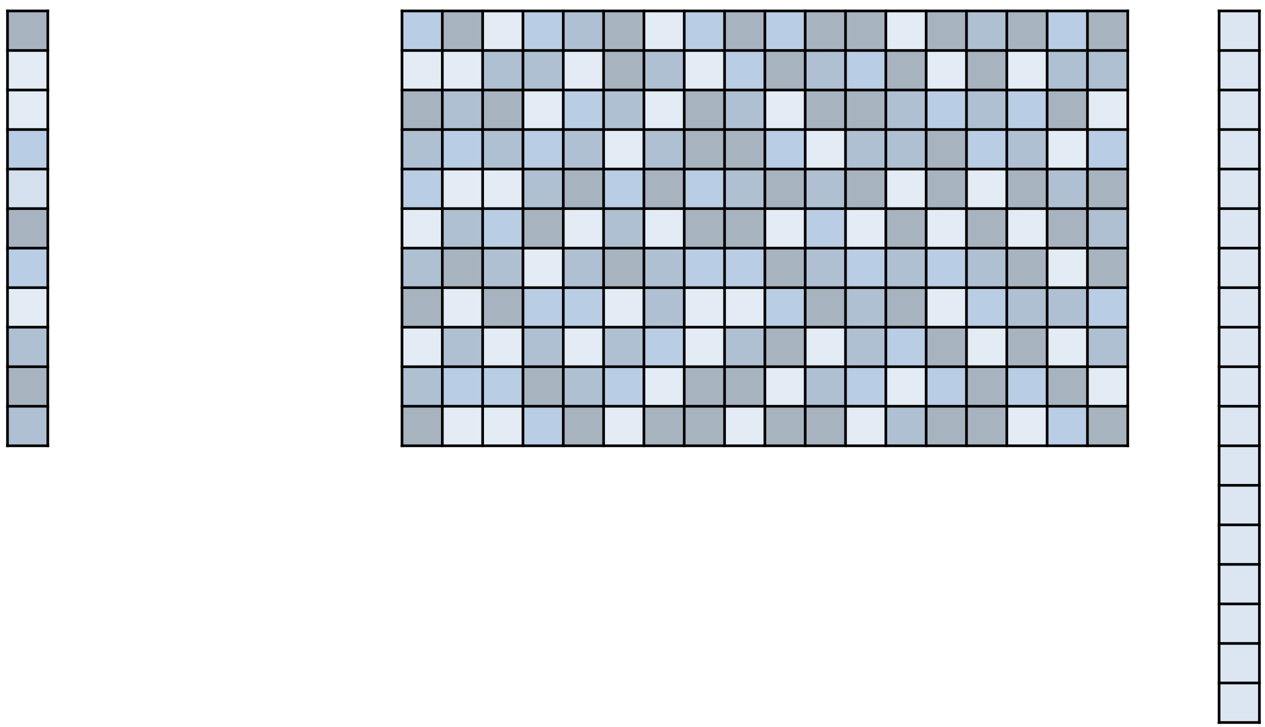
- **Orthogonal Matching Pursuit**
  - **Problem setup:** $y = Dx$



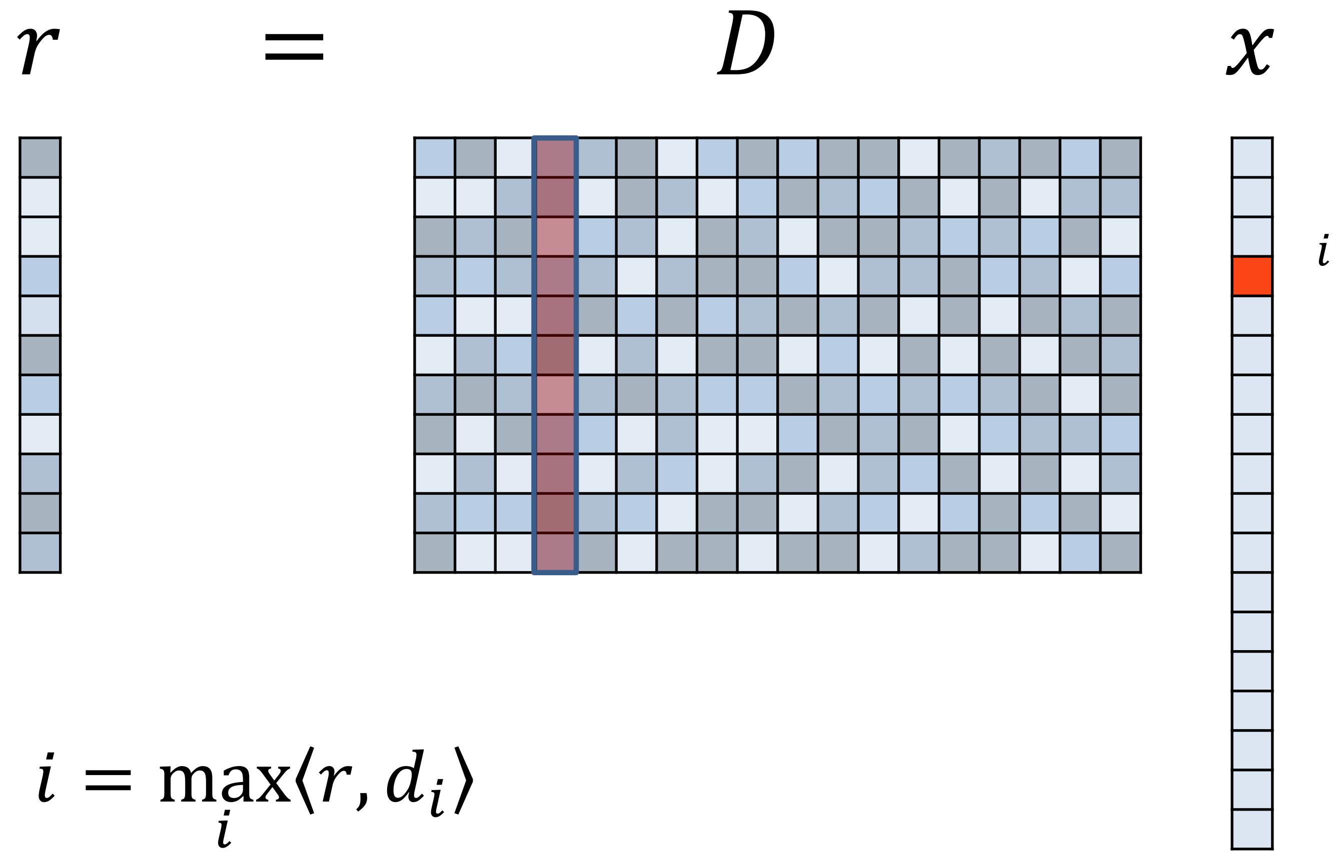
Sparse Learning

- **Orthogonal Matching Pursuit**
  - **Step 0:** Set $r = y$

$$r = D \quad x$$

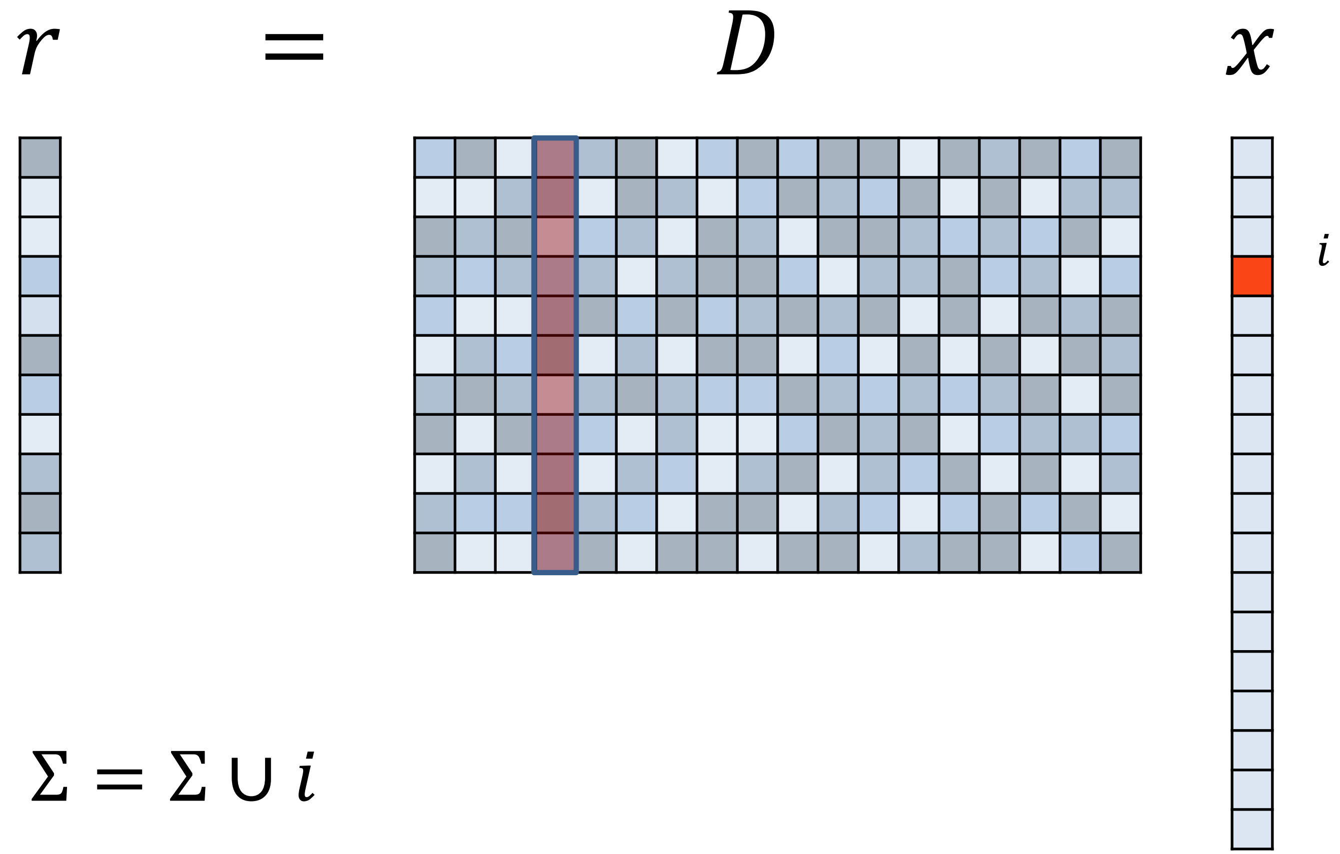


- **Orthogonal Matching Pursuit**
  - **Step 1:** Find column of *D* that best matches *r*



$$i = \max_i \langle r, d_i \rangle$$

- **Orthogonal Matching Pursuit**
  - **Step 2:** Add index in a set (initialized as empty)

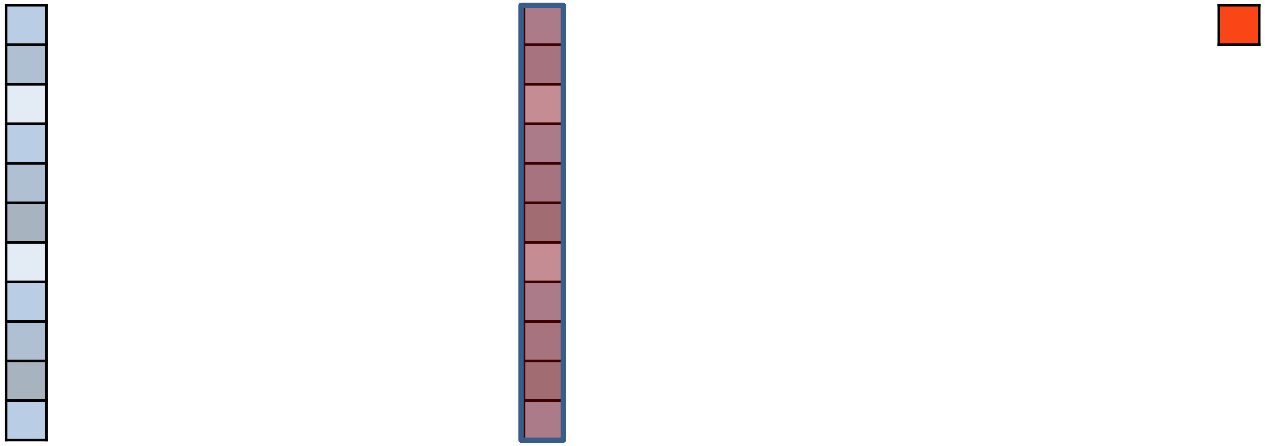


$$\Sigma = \Sigma \cup i$$

Sparse Learning

- **Orthogonal Matching Pursuit**
  - **Step 3:** Compute least-squares estimate of relevant indices of *x*

$$\hat{y} \qquad = \qquad D \qquad x$$

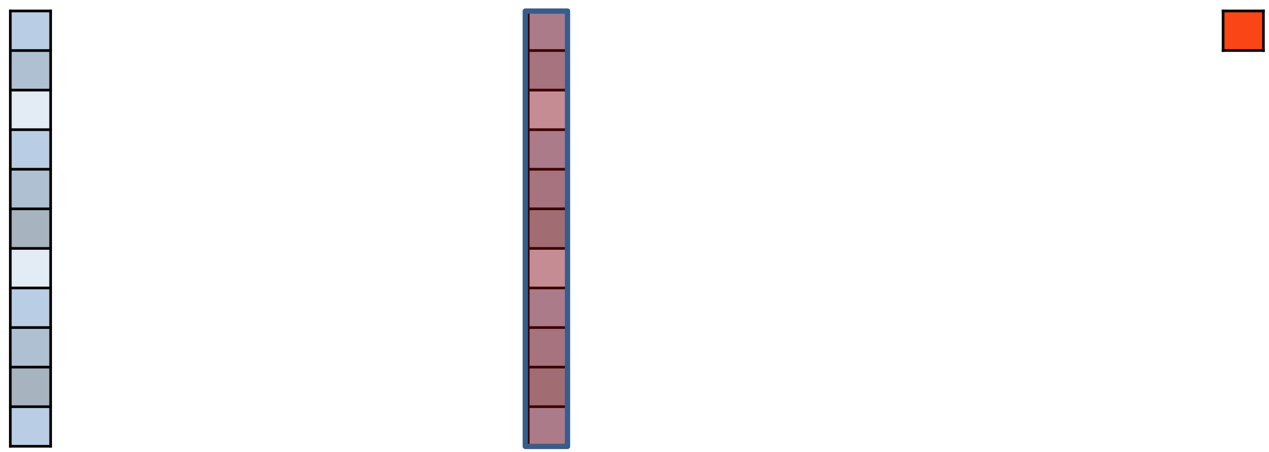


$$x_\Sigma = D^\dagger r$$

Sparse Learning

- **Orthogonal Matching Pursuit**
  - **Step 4:** Compute new residual *r*

$$\hat{y} \quad = \quad D \quad x$$

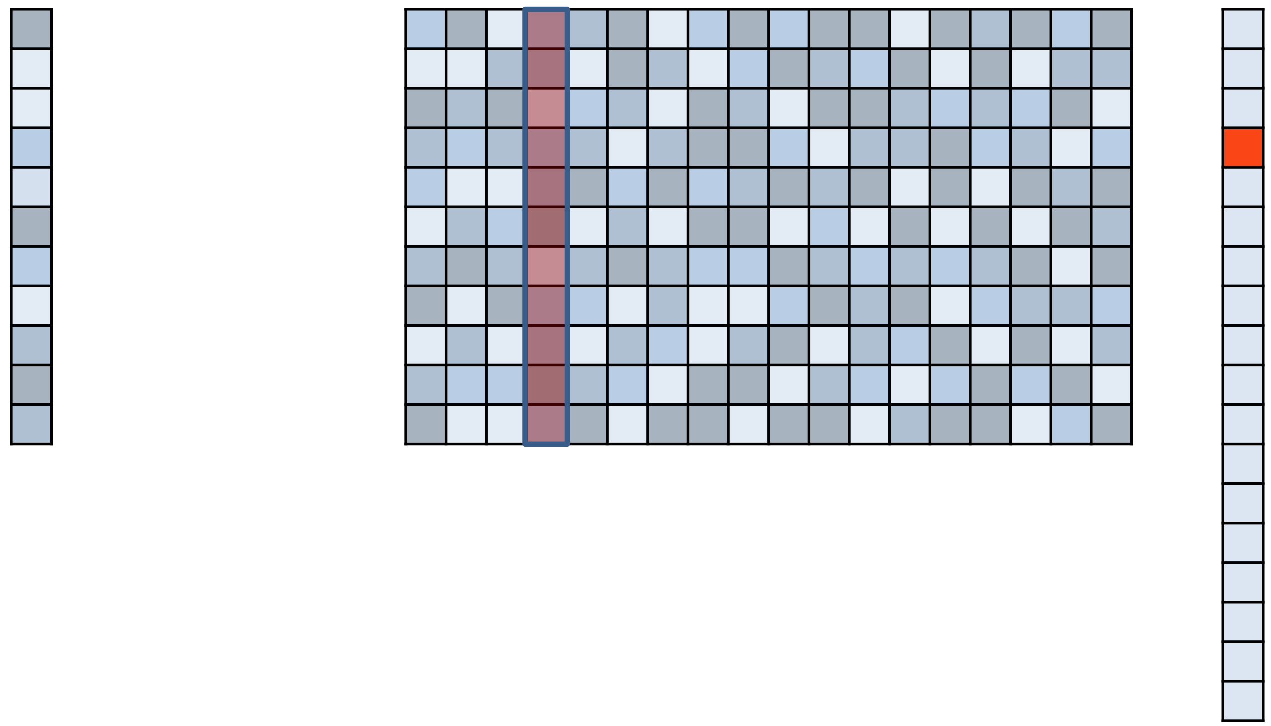


$$r = \hat{y} - Dx$$

Sparse Learning

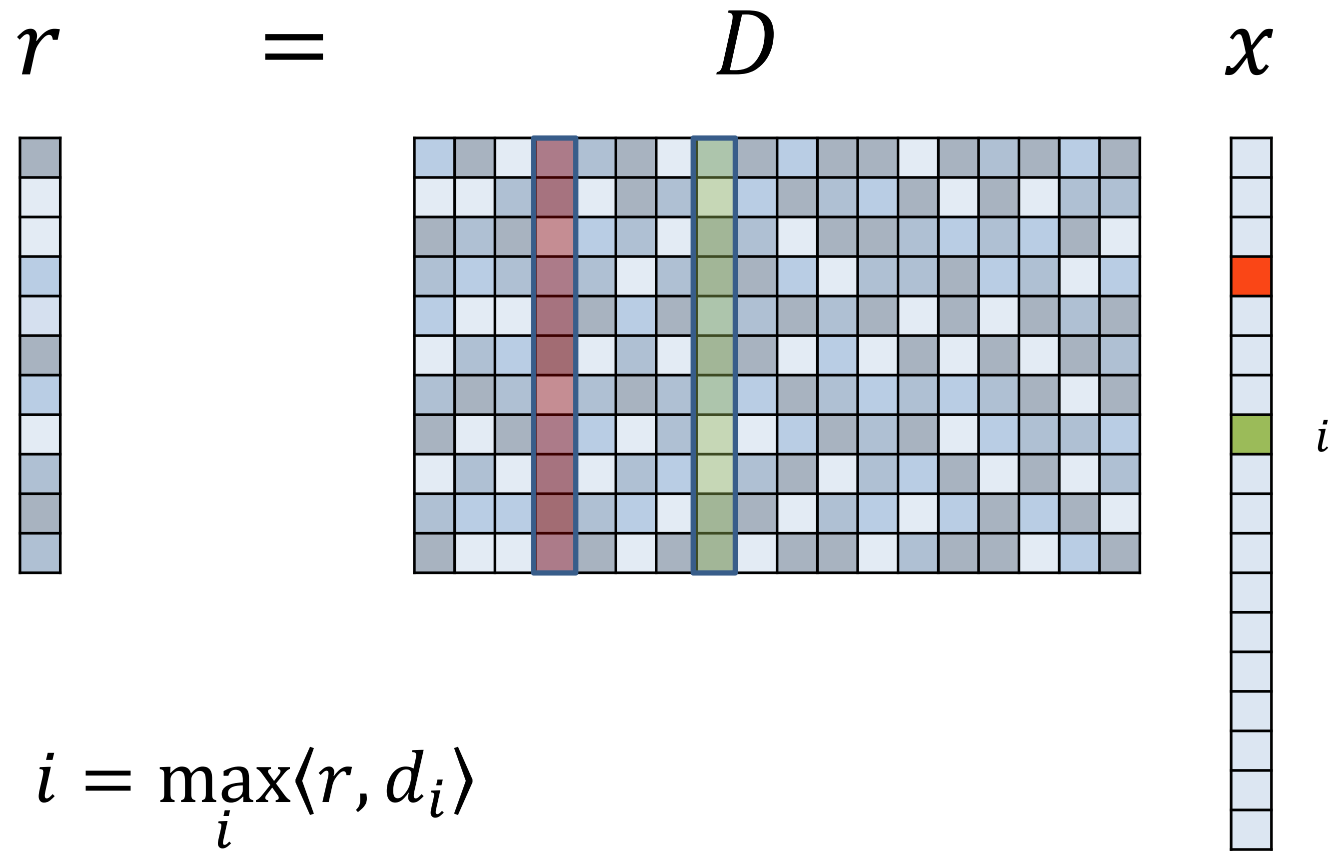
- **Orthogonal Matching Pursuit**
  - **Step 5:** Repeat steps 1-4 until convergence

$$y \quad = \quad D \quad x$$



Sparse Learning

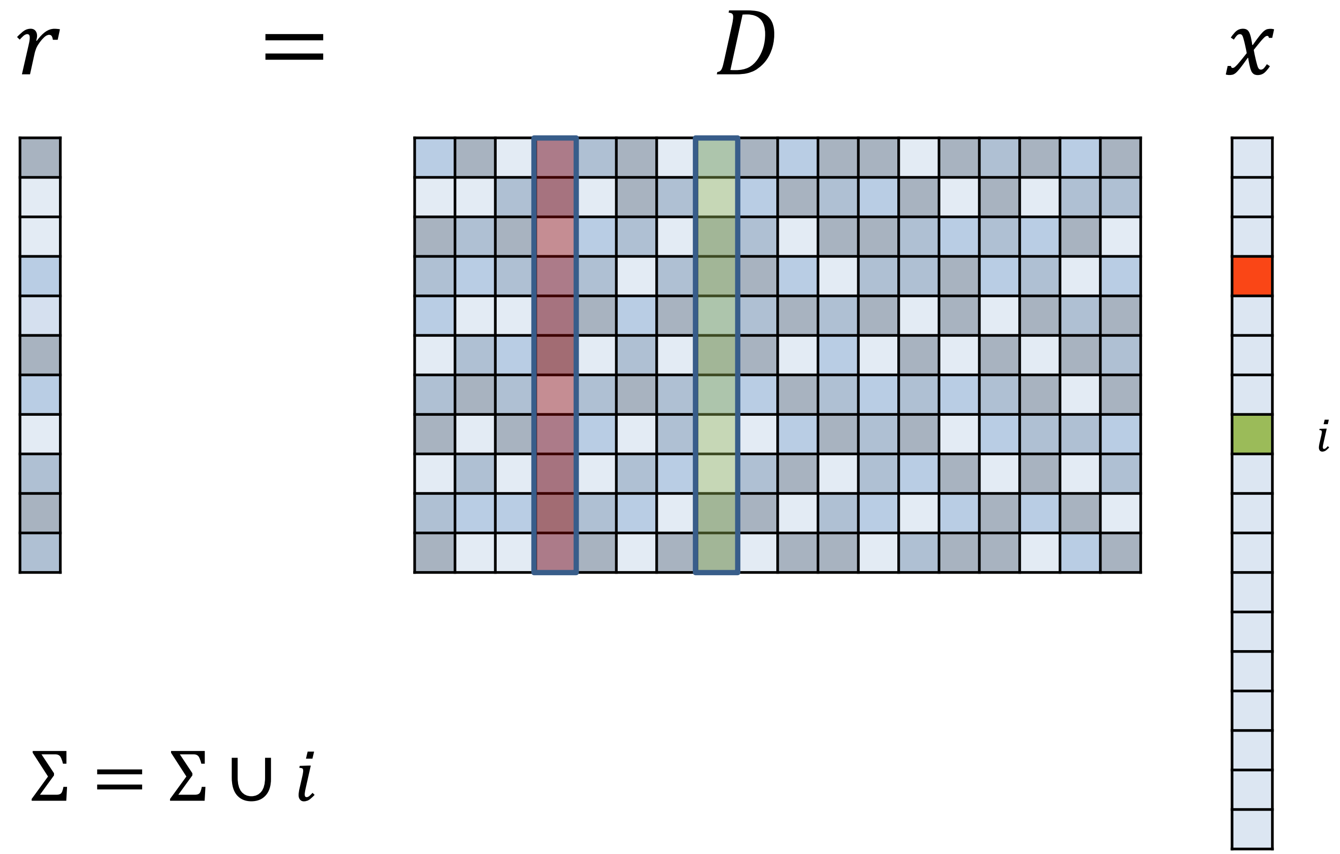
- **Orthogonal Matching Pursuit**
  - **Step 1:** Find column of *D* that best matches *y*



$$i = \max_{i} \langle r, d_i \rangle$$

Sparse Learning

- **Orthogonal Matching Pursuit**
  - **Step 2:** Add index in a set (initialized as empty)

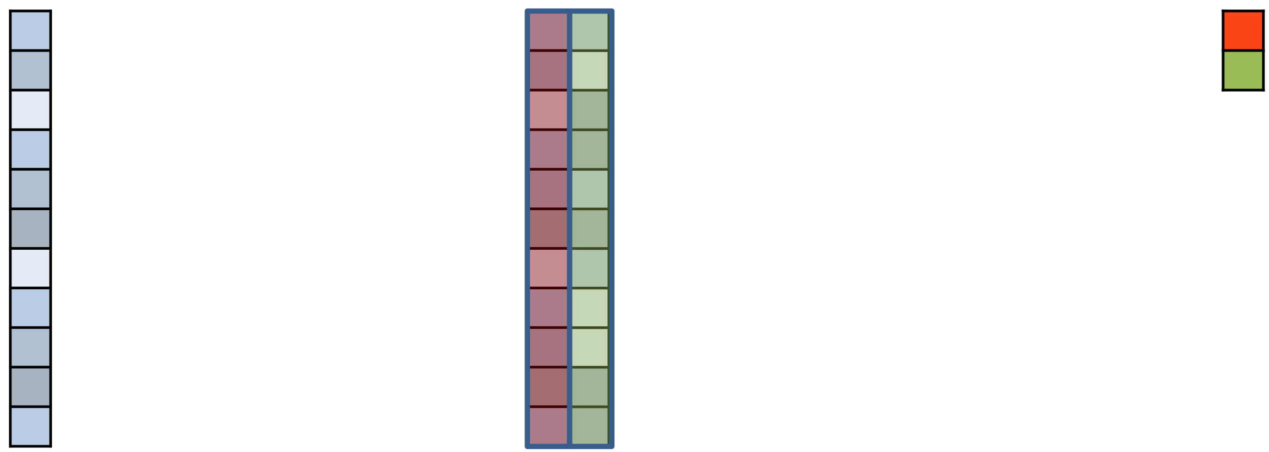


$$\Sigma = \Sigma \cup i$$

Sparse Learning

- **Orthogonal Matching Pursuit**
  - **Step 3:** Compute least-squares estimate of relevant indices of *x*

$$\hat{y} \quad = \quad D \quad x$$

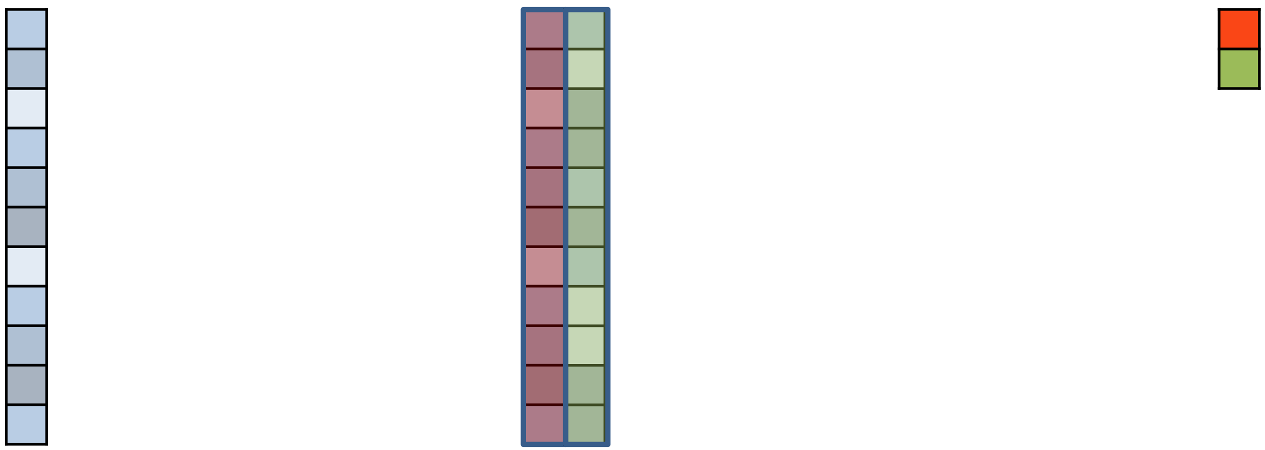


$$x_\Sigma = D^\dagger r$$

Sparse Learning

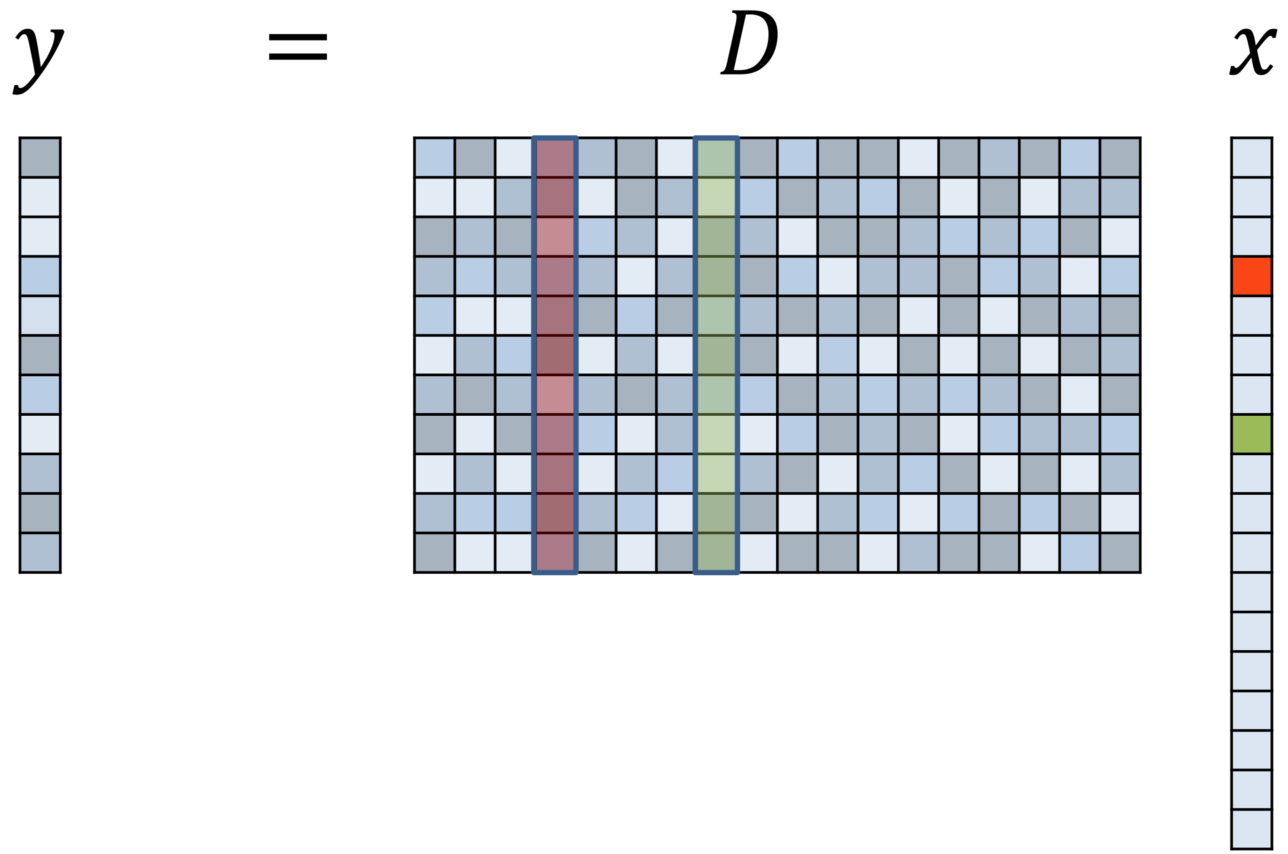
- **Orthogonal Matching Pursuit**
  - **Step 4:** Compute new residual *r*

$$\hat{y} \quad = \quad D \quad x$$



$$r = \hat{y} - Dx$$

- **Orthogonal Matching Pursuit**
  - **Step 5:** Repeat steps 1-4 until convergence

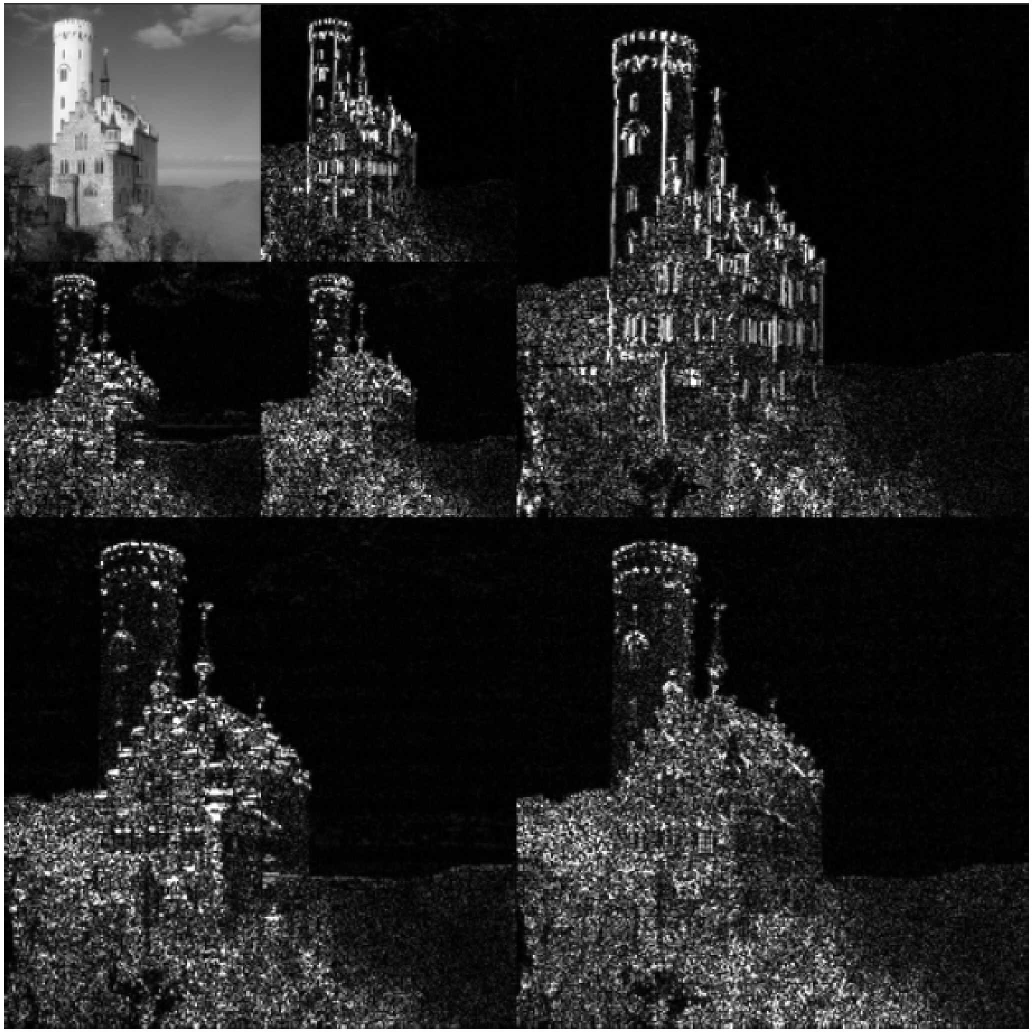


Applications

- **Question: Why is sparsity useful / important?**

Applications

- **Most natural data is sparse in some basis**

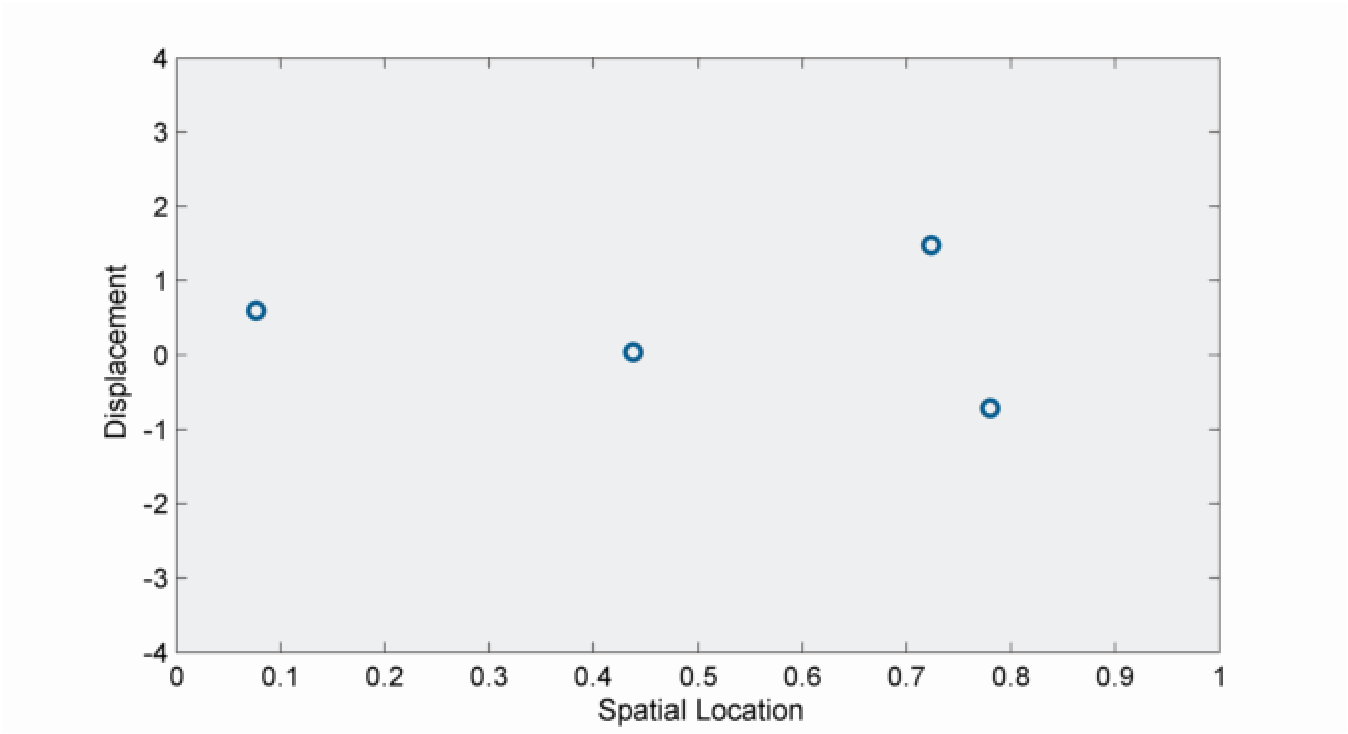


Applications

- **Most natural data is sparse in some basis**

Applications

- **Most natural data is sparse in some basis**



Applications

- **Most natural data is sparse in some basis**
  - For a standing wave, we can represent the data at four points on the string by

$$X = \Phi Y F^*$$

Measured Time Data

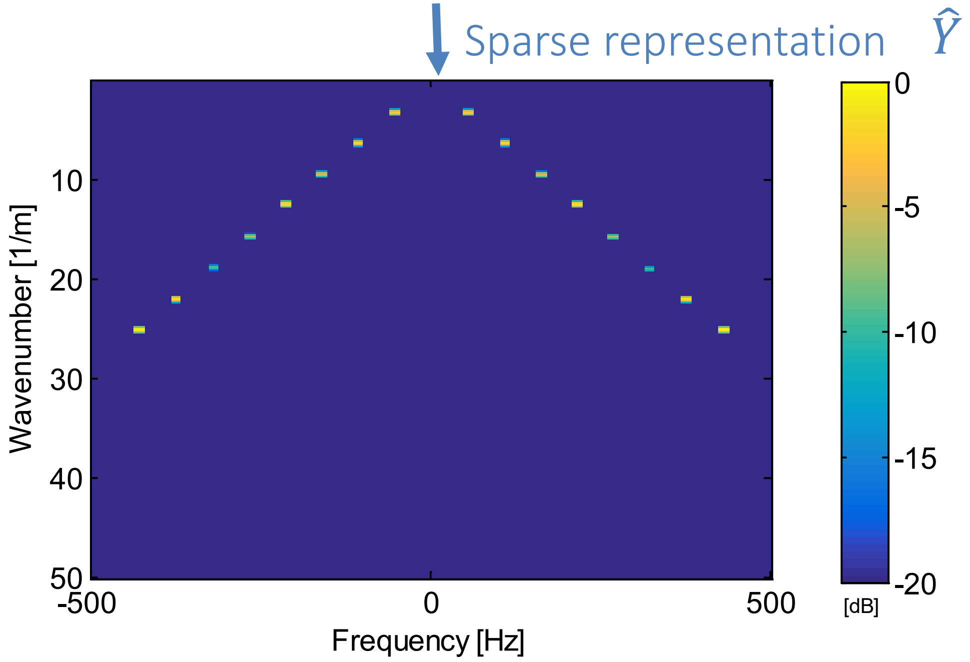
Sampled DFT matrix in space

Sparse Frequency-Wavenumber Representation

DFT matrix in time

Applications

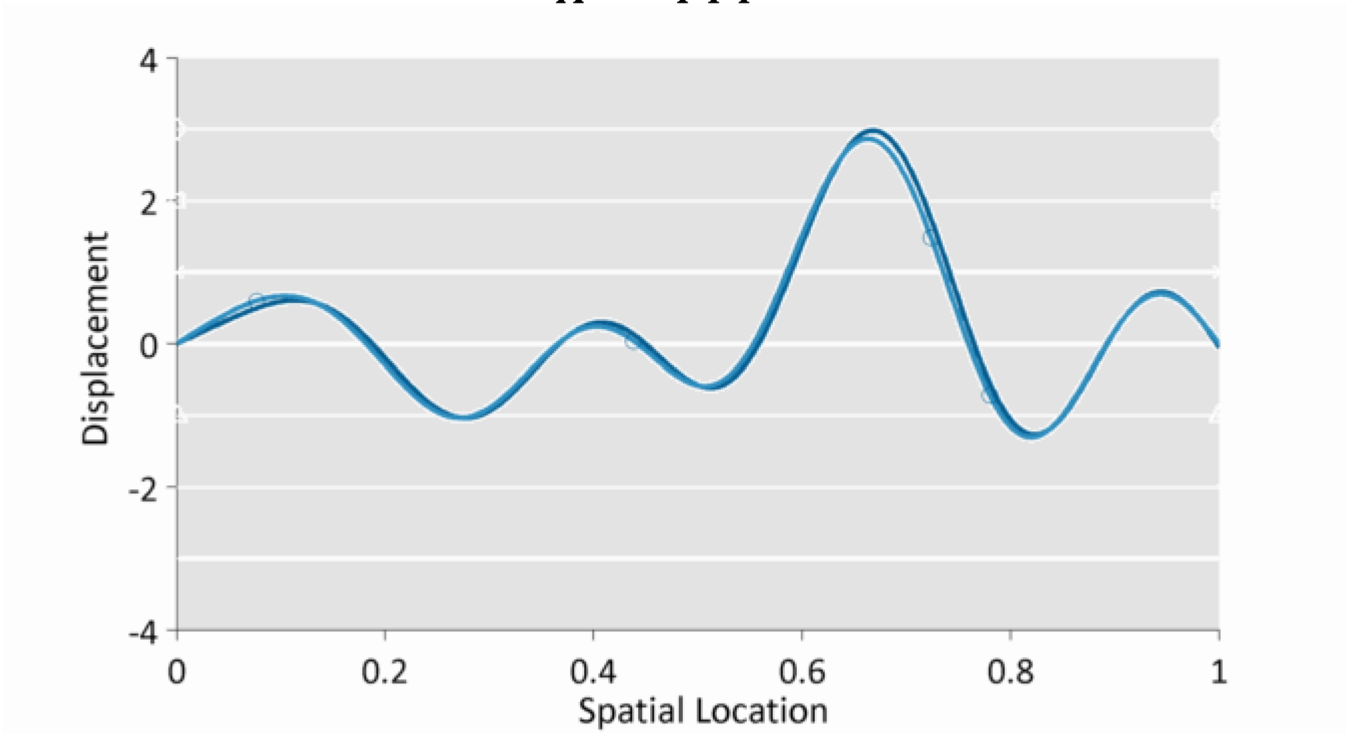
- **Most natural data is sparse in some basis**
  - For a standing wave, we can represent the data at four points on the string by



Applications

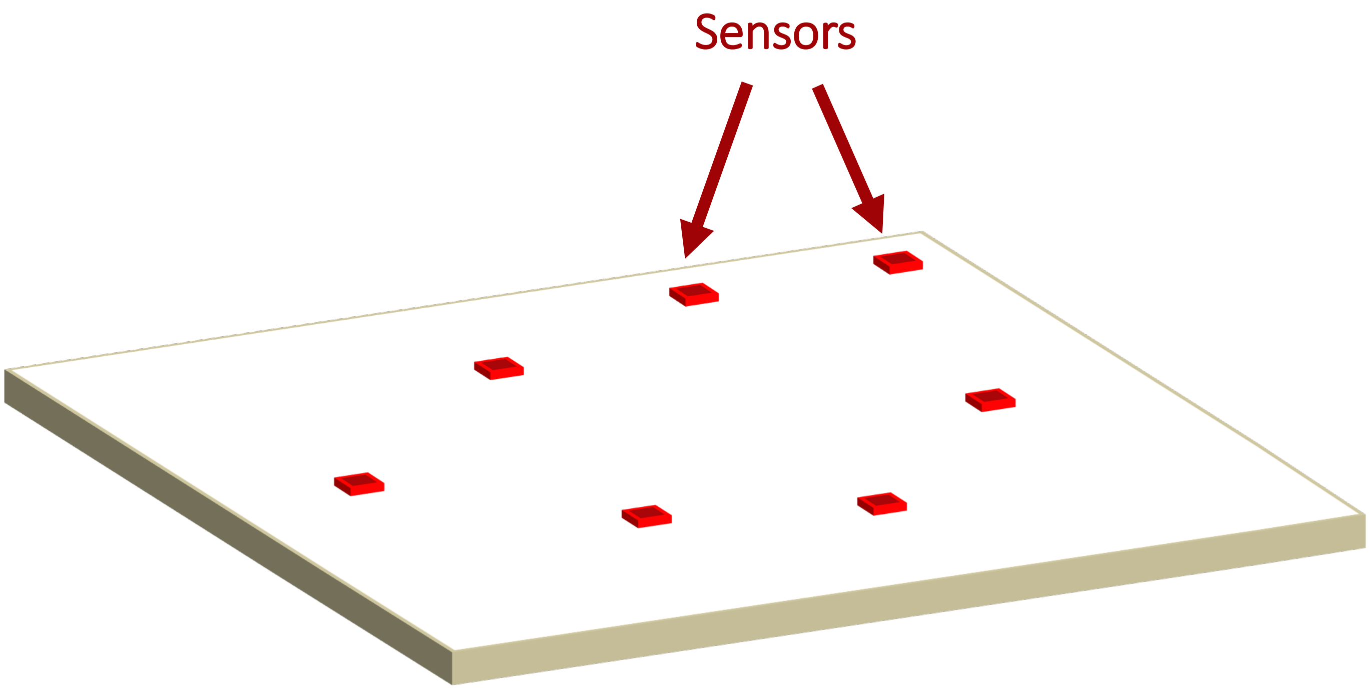
- **Most natural data is sparse in some basis**

$$\hat{X} = \Phi \hat{Y} F^*$$



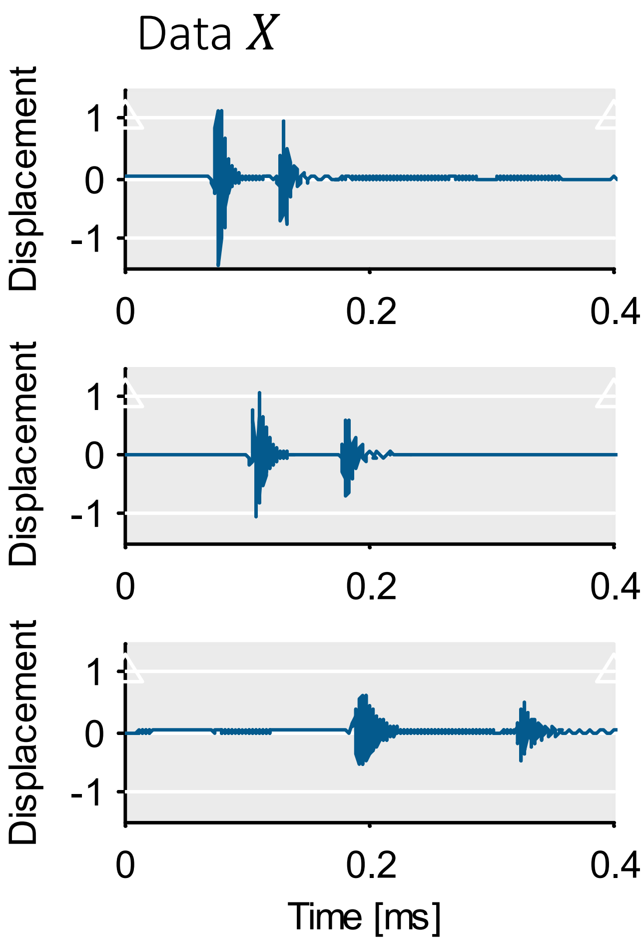
Applications

- **Question: Why Compressive sensing useful / important?**



Applications

- **Most natural data is sparse in some basis**

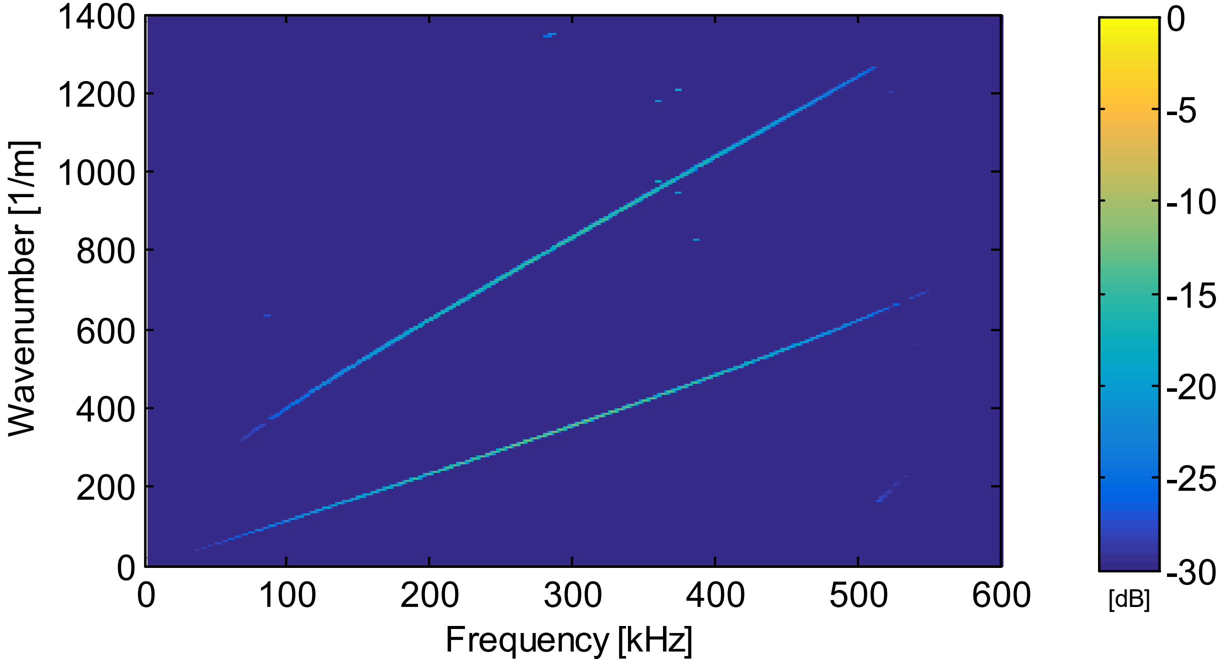


Applications

- **Most natural data is sparse in some basis**

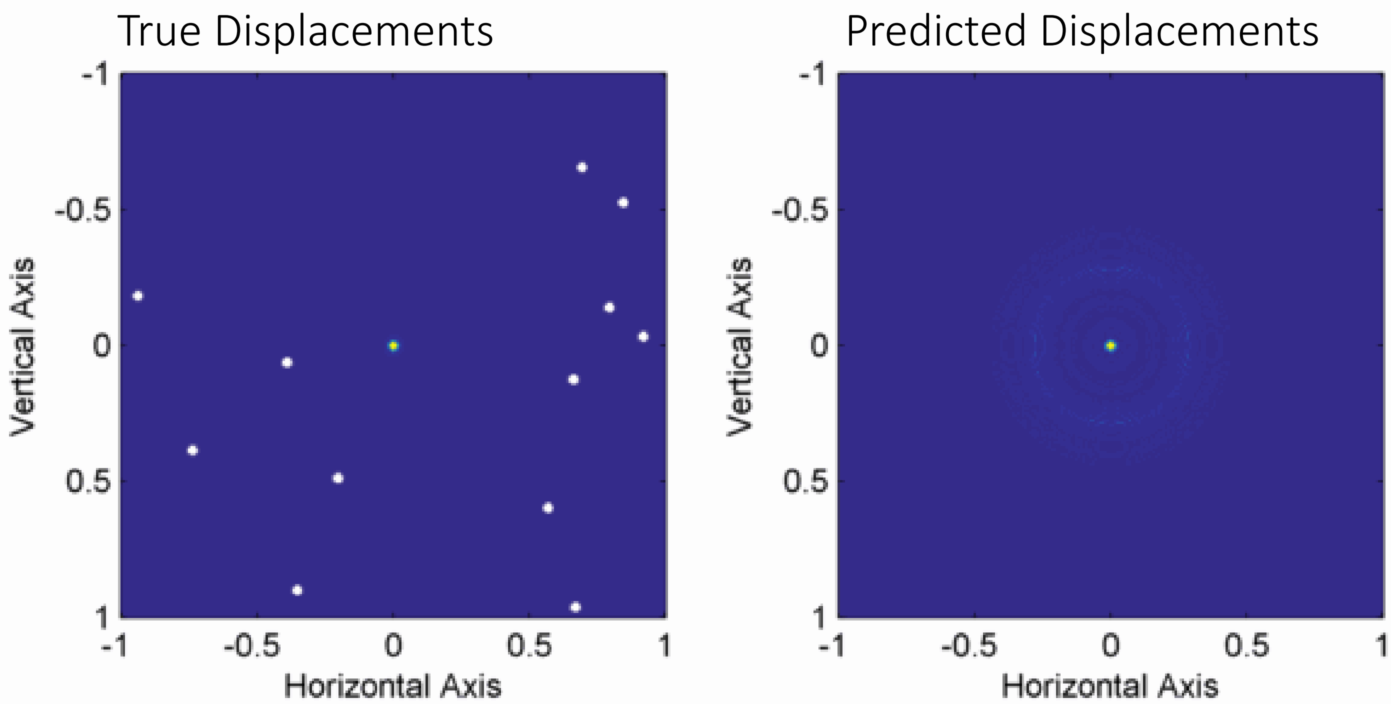
$$X = \Phi Y$$

Sparse representation $\hat{Y}$



Applications

- **Most natural data is sparse in some basis**



Lecture 27: Wavelets to Modern Signal Processing

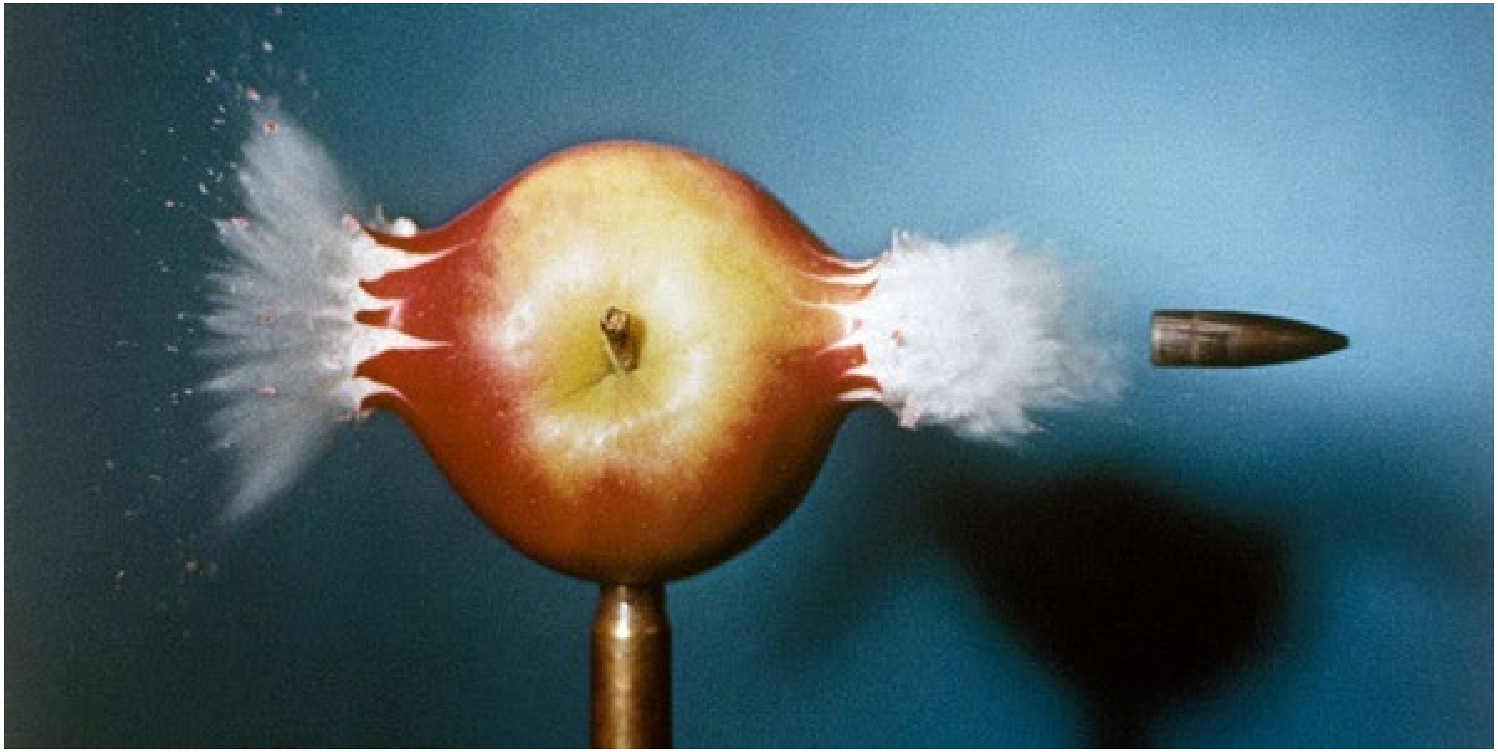
Foundations of Digital Signal Processing

Outline

- Review of Filter Banks and Wavelets
- Applications of Wavelets
- Applications of Filter Banks and Time-Frequency Representations
- Modern Signal Processing: Vectors and Matrices
- Modern Signal Processing: Compressive Sensing
- **Modern Signal Processing: Diagonalization**
- Modern Signal Processing: Graph Signal Processing

Applications

- **Sensing is expensive**



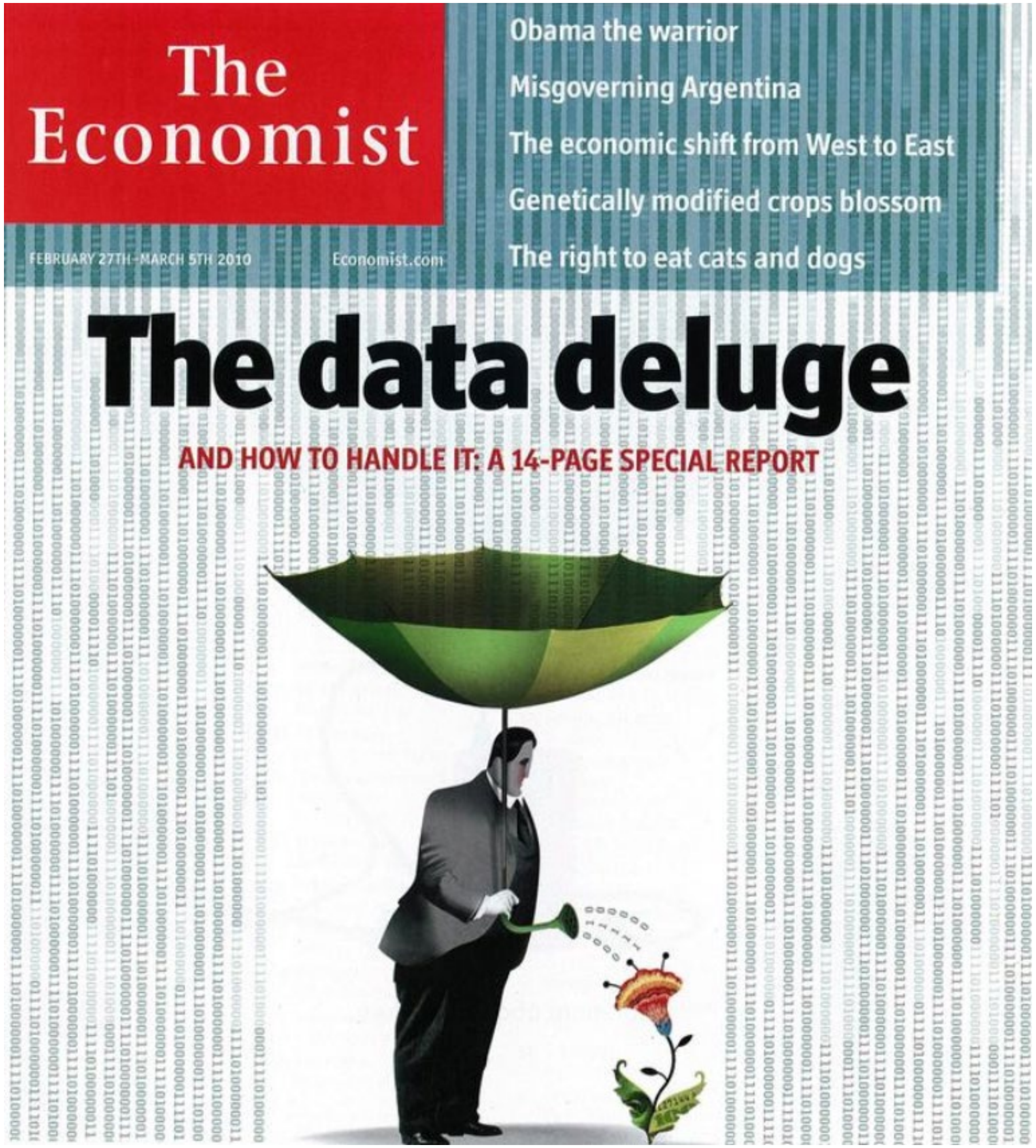
Applications

- **Sensing is expensive**
  - **Cost of high-speed cameras**

Product Name	Cost for Demo Unit	Cost for New Unit
SA5 775K M1 (MONO 8 GIGS)	$68,500	$90,000
SA5 775K M2 w/ mech. Shutter	$77,000	$103,120
SA5 1000K C2 RV COLOR -16 GIGS	$80,000	$113,120
BC2 HD with Keypad	$90,000	$132,400
SA2 M2 (MONO 16 GIG HIGH DEF)	$55,500	$100,000

Applications

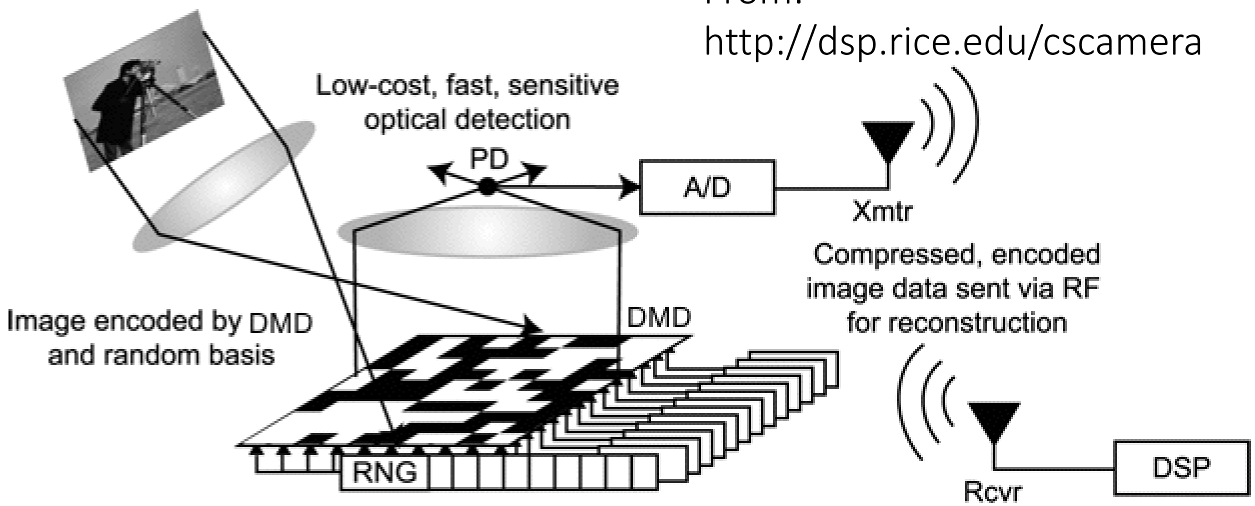
- **Sensing is expensive**

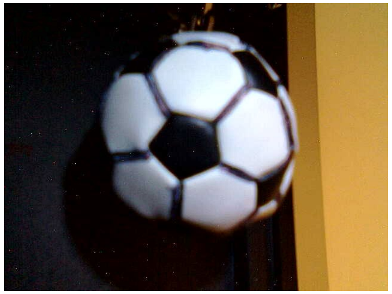


Applications

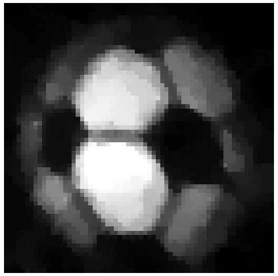
- **Sensing is expensive – single pixel camera**

From:
http://dsp.rice.edu/cscamera

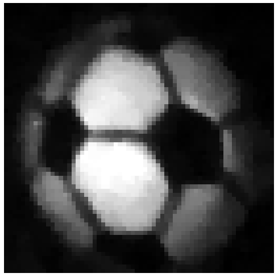




Original Object



4096 pixels from 800 measurements (20%)

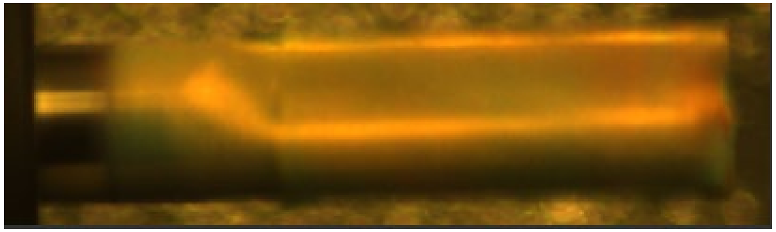


4096 pixels from 1600 measurements (40%)

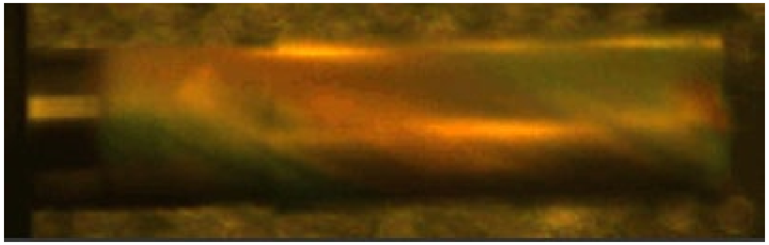
Applications

- **Sensing is expensive**

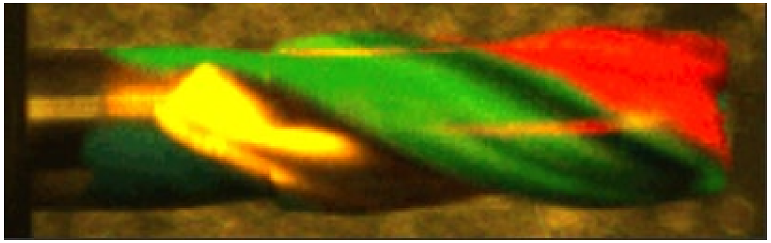
Normal video at 25fps



Compressively sampled video at 25fps

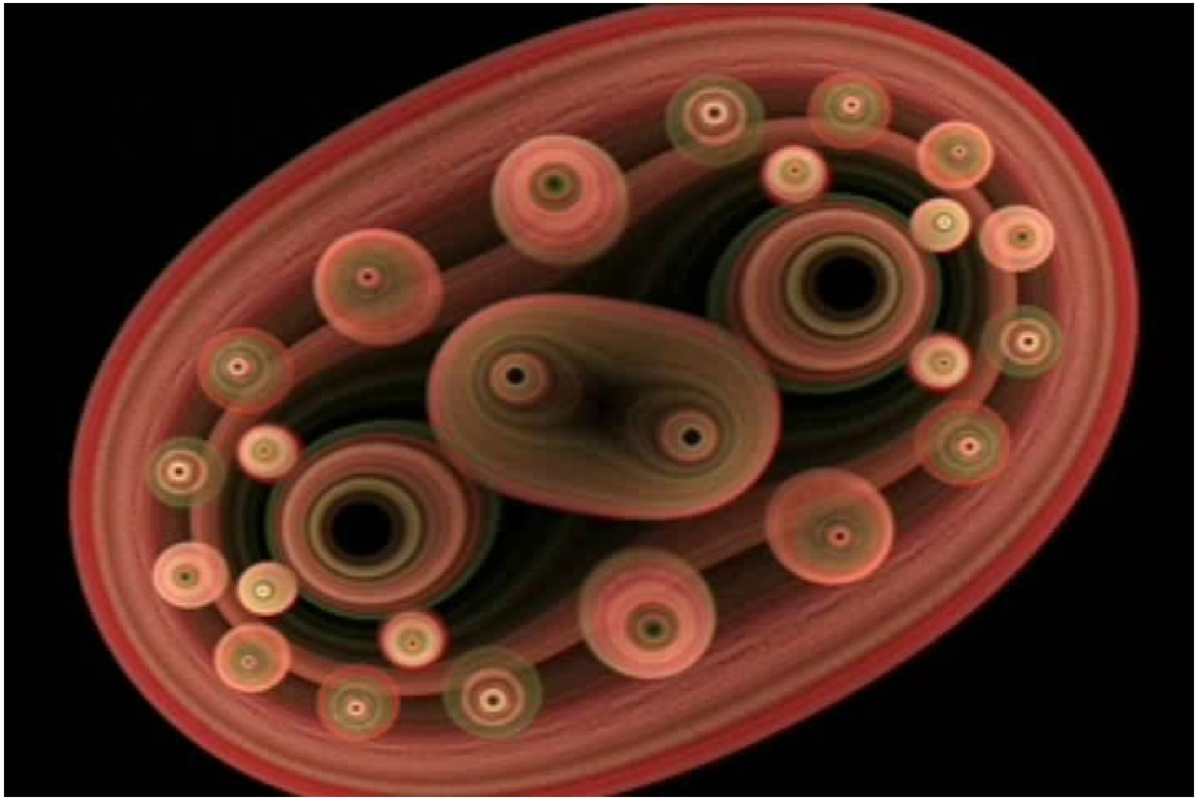


Recovered video at 2000fps



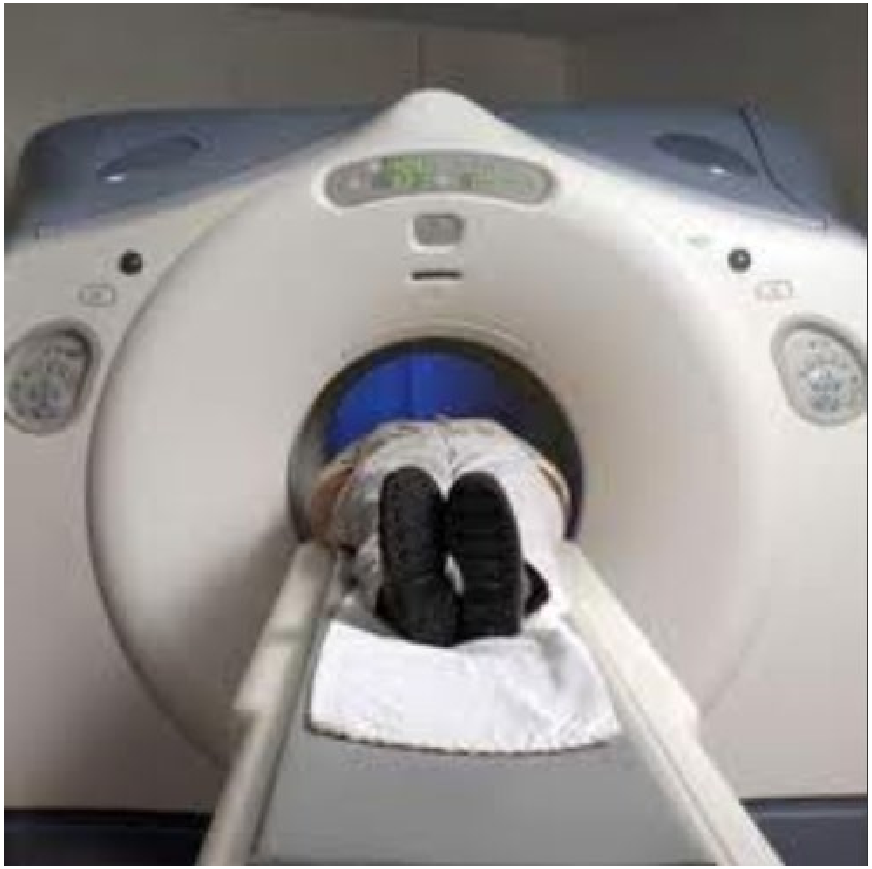
Applications

- **Sensing is expensive**



Applications

- **Sensing is expensive**
  - **Magnetic Resonance Imaging**



Diagonalization

Diagonalization

- **Eigenvalue decomposition**
  - A number λ and a nonzero vector x are called an eigenvalue and an eigenvector of a **square** matrix A (they are also known as an eigenpair) when

$$Hx = \lambda x$$

Diagonalization

Eigenvalue decomposition

- A number λ and a nonzero vector x are called an eigenvalue and an eigenvector of a **square** matrix A (they are also known as an eigenpair) when

$$Hx = \lambda x$$

- If we concatenate every eigenvector $x_1, x_2, x_3, \dots, x_N$, we can form:

$$H\begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \lambda_N \end{bmatrix}$$

$$HX = X\Lambda$$

Diagonalization of our linear operator

$$H = X\Lambda X^{-1}$$

Diagonalization

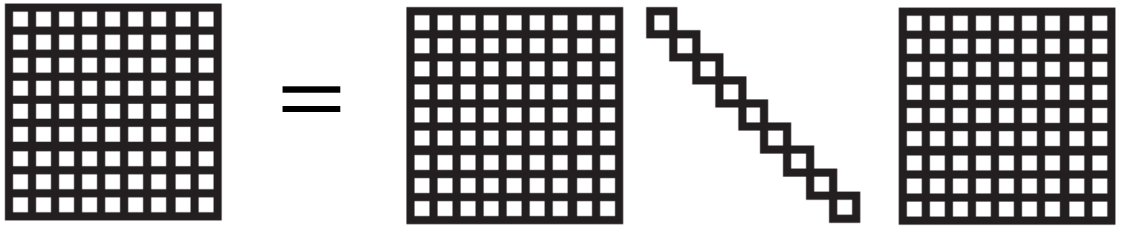
Eigenvalue decomposition

- A number λ and a nonzero vector x are called an eigenvalue and an eigenvector of a **square** matrix A (they are also known as an eigenpair) when

$$Hx = \lambda x$$

- We have two useful representations for the eigen-decomposition:

(1) $H = X\Lambda X^{-1} = X\Lambda\tilde{X}^*$



Diagonalization

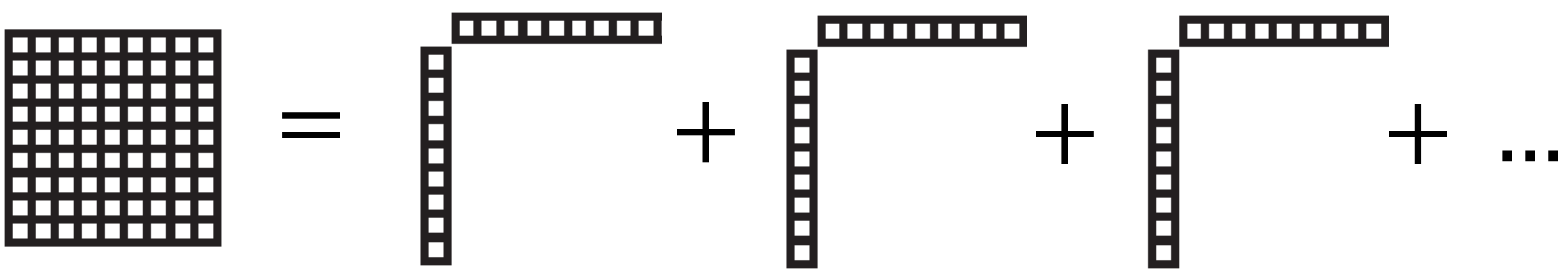
Eigenvalue decomposition

- A number λ and a nonzero vector x are called an eigenvalue and an eigenvector of a **square** matrix A (they are also known as an eigenpair) when

$$Hx = \lambda x$$

- We have two useful representations for the eigen-decomposition:

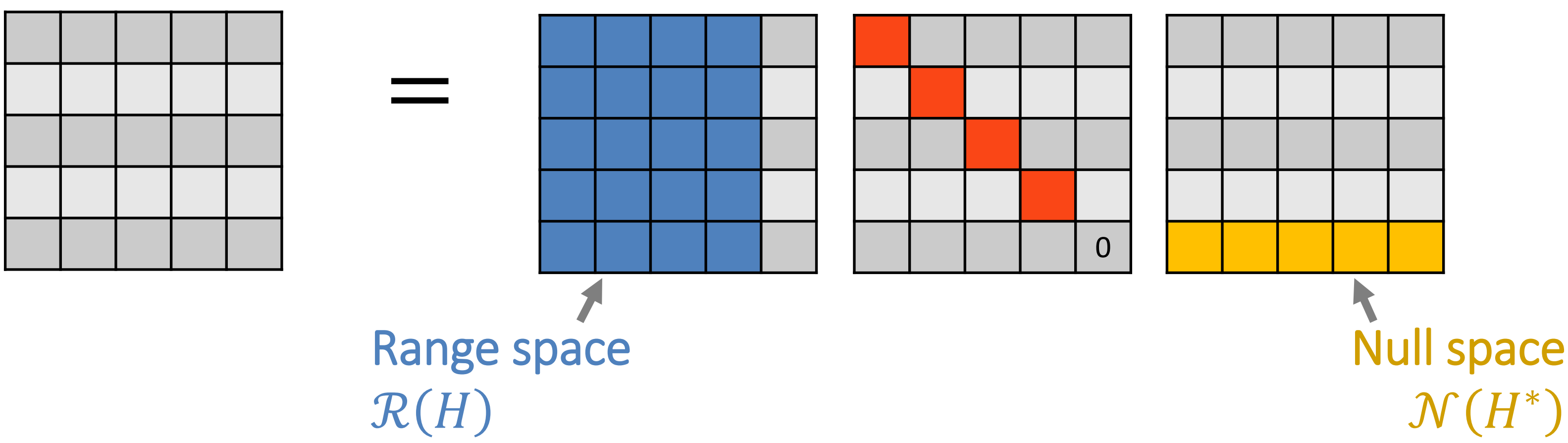
(2) $H = \lambda_1 x_1 \tilde{x}_1^* + \lambda_2 x_2 \tilde{x}_2^* + \cdots + \lambda_N x_N \tilde{x}_N^*$



Diagonalization

- **Eigenvalue decomposition**
  - Properties of the eigenvectors

$$H = X\Lambda X^{-1} = X\Lambda\tilde{X}^{*}$$

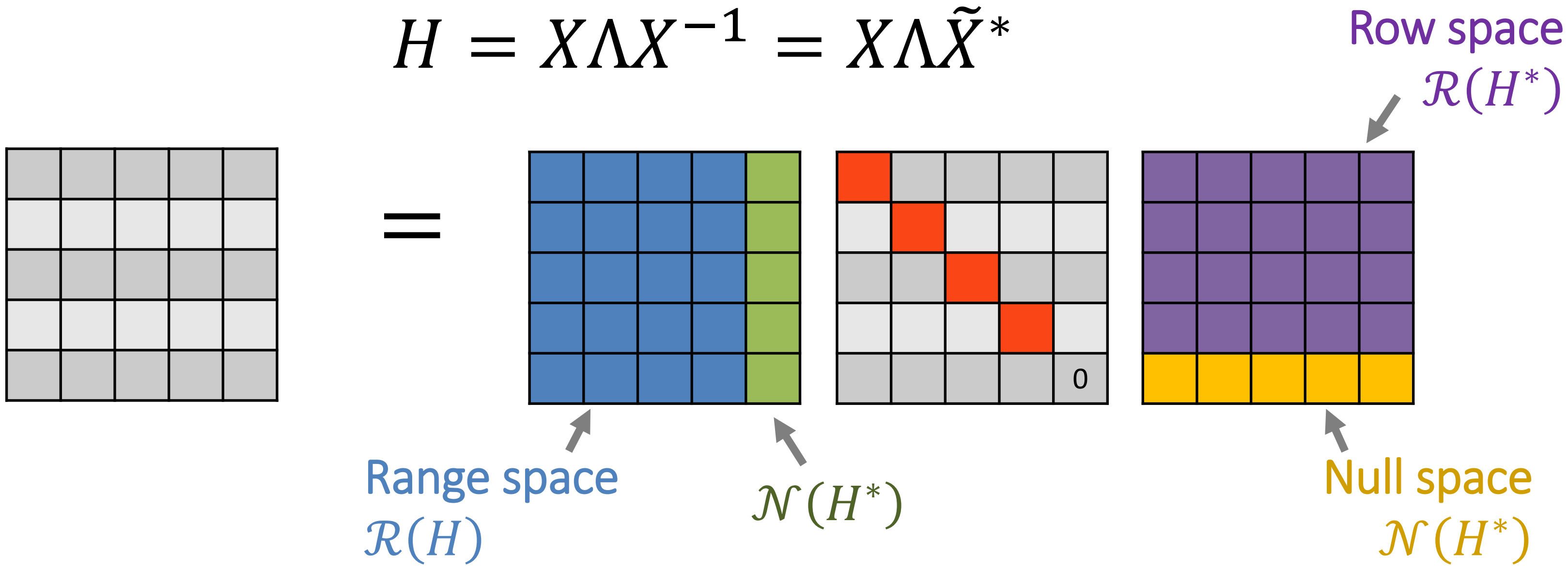


Diagonalization

Eigenvalue decomposition

- Properties of the eigenvectors

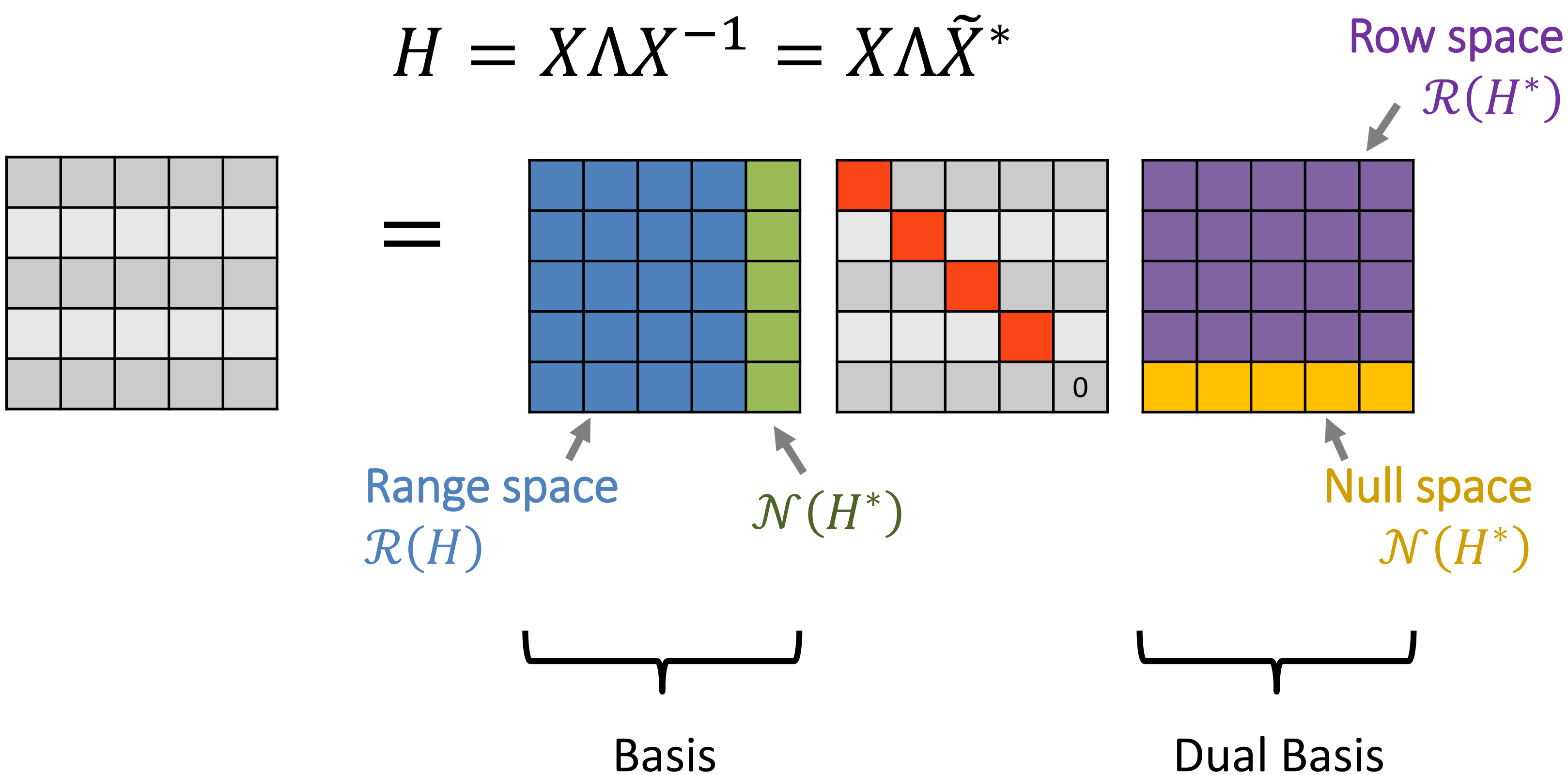
$$H = X\Lambda X^{-1} = X\Lambda\tilde{X}^*$$



Eigenvalue decomposition

- Properties of the eigenvectors

$$H = X\Lambda X^{-1} = X\Lambda\tilde{X}^*$$



Diagonalization

- **Question:** How is eigenvalue decomposition useful?

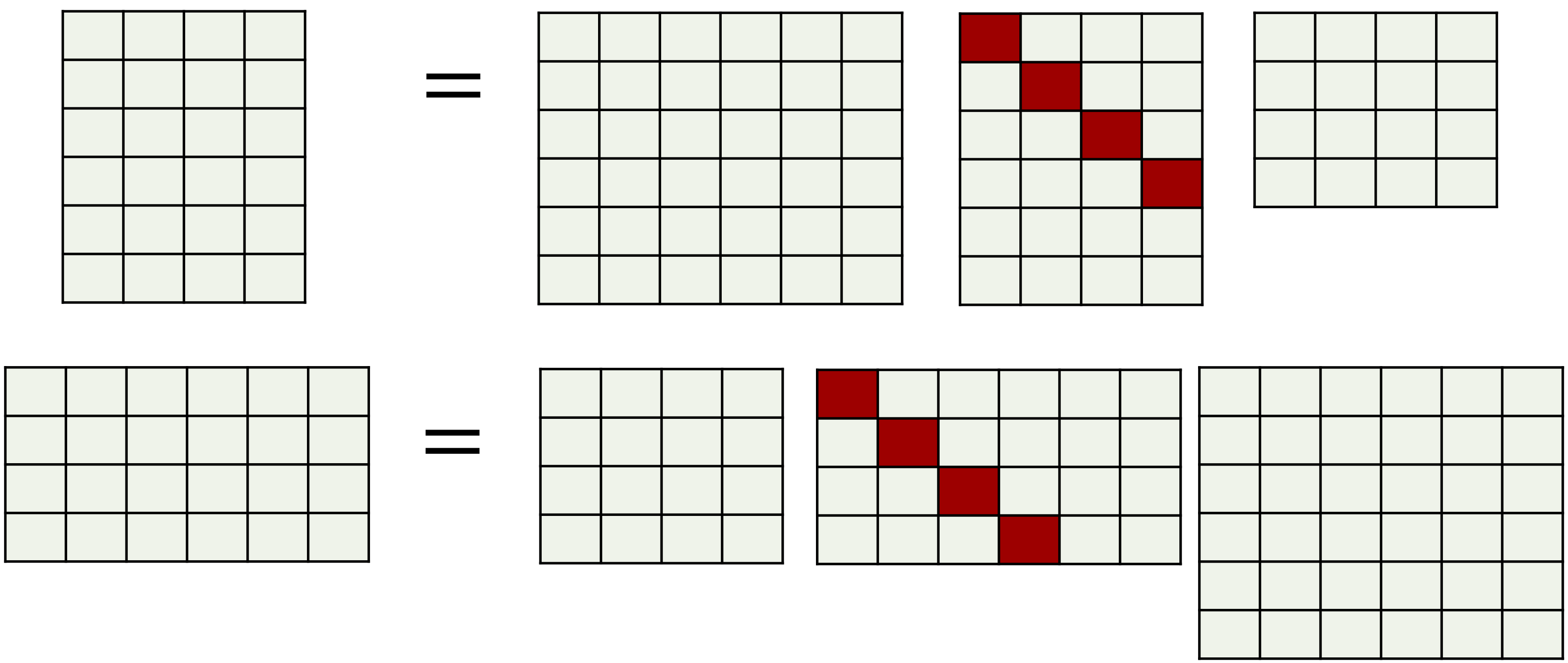
Diagonalization

- **Properties of Singular Value Decomposition**
  - **Singular value decomposition:** $H = U\Sigma V^*$
  - **Left Singular vectors:** U
  - **Right Singular vectors:** V

- **Relationship with eigendecomposition**
  - **Left Singular vector:** $HH^* = U\Sigma^2U^*$ (eigendecomposition of HH^*)
  - **Right Singular vectors:** $H^*H = V\Sigma^2V^*$ (eigendecomposition of H^*H)

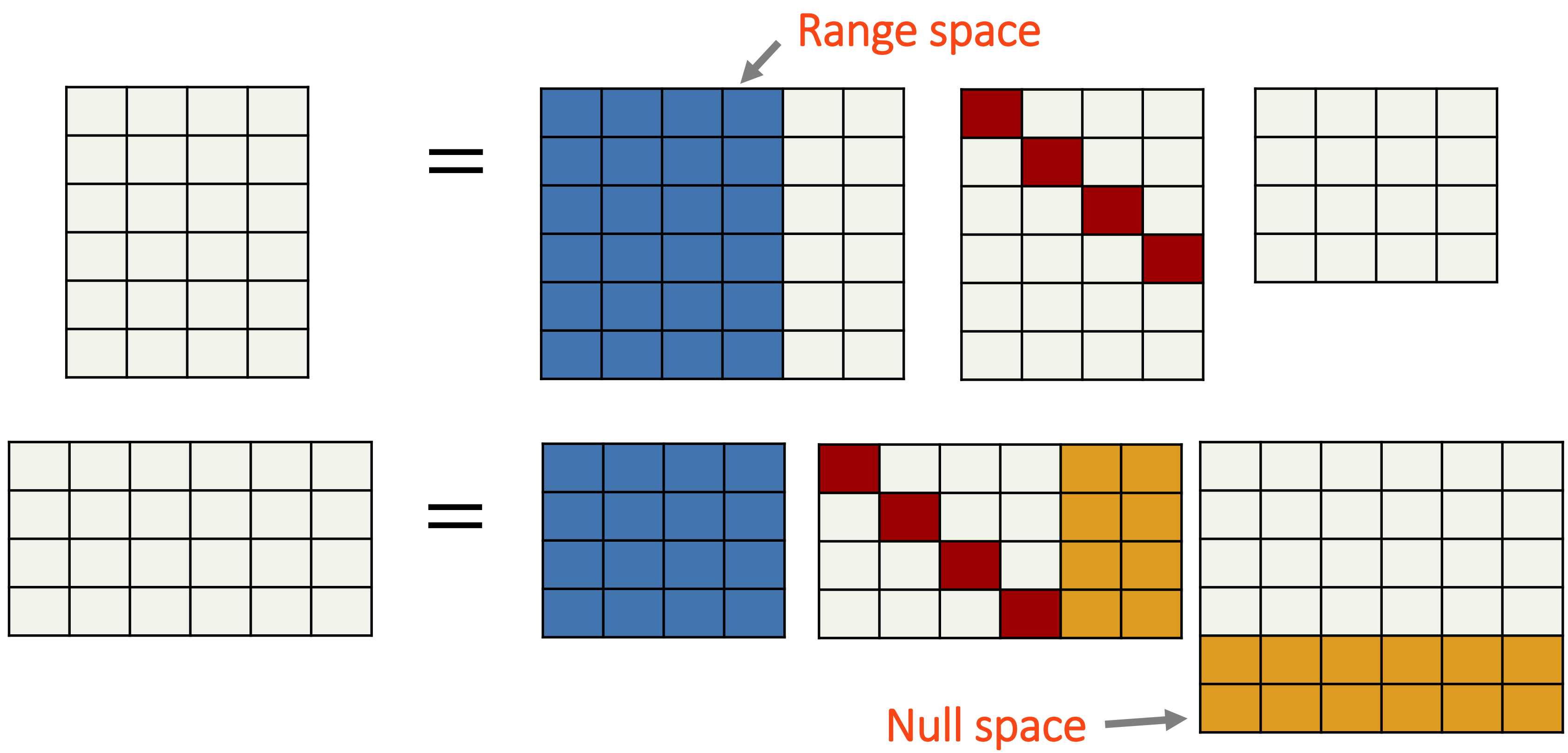
Diagonalization

- **Singular Value Decomposition**
  - **Singular value decomposition:** $H = U\Sigma V^*$
  - **Left Singular vectors:** $HH^* = U\Sigma U^*$
  - **Right Singular vectors:** $H^*H = V\Sigma V^*$



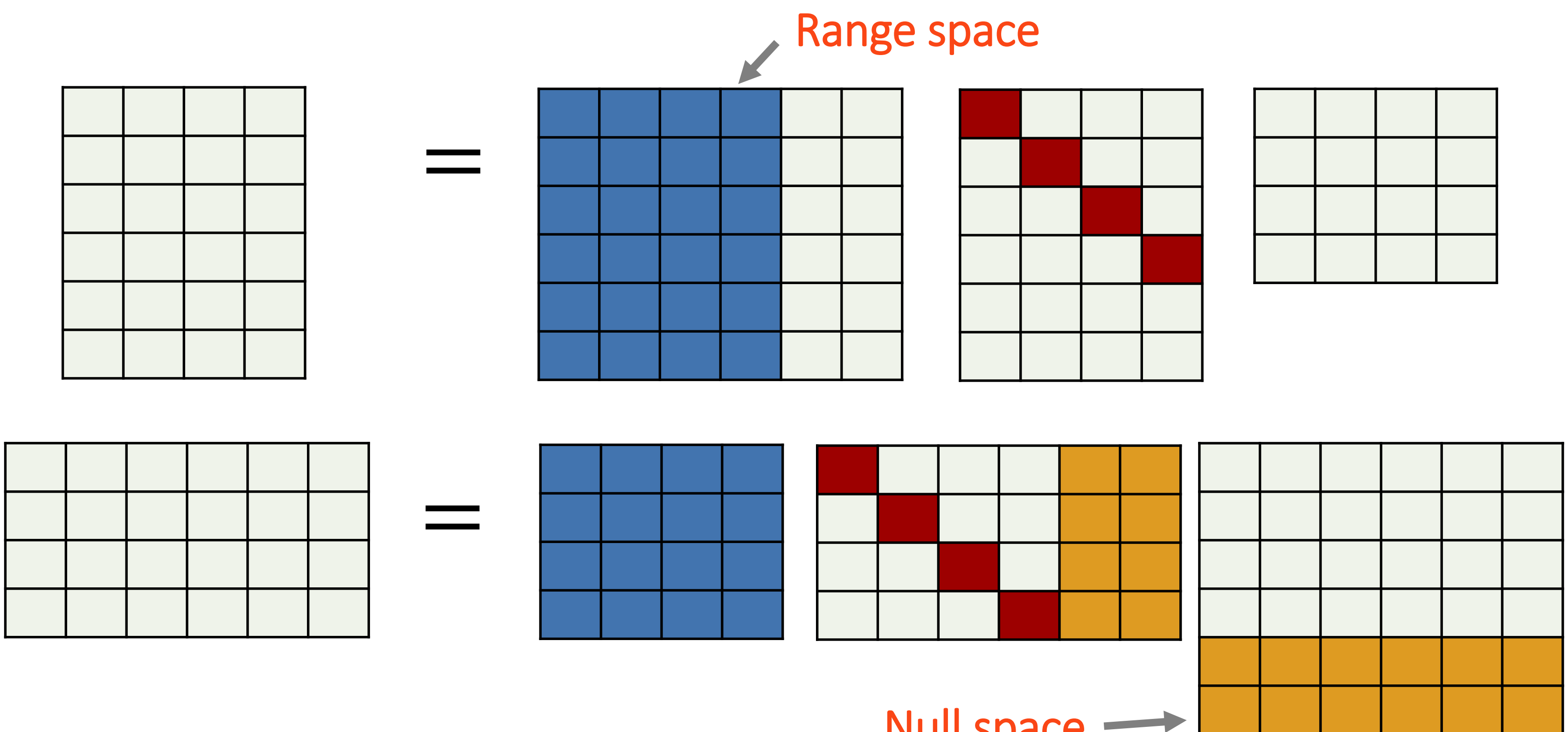
Diagonalization

- **Singular Value Decomposition Properties**
  - **Null space:** Columns of V corresponding to 0 singular values
  - **Range Space:** Columns of U corresponding to non-zero singular values



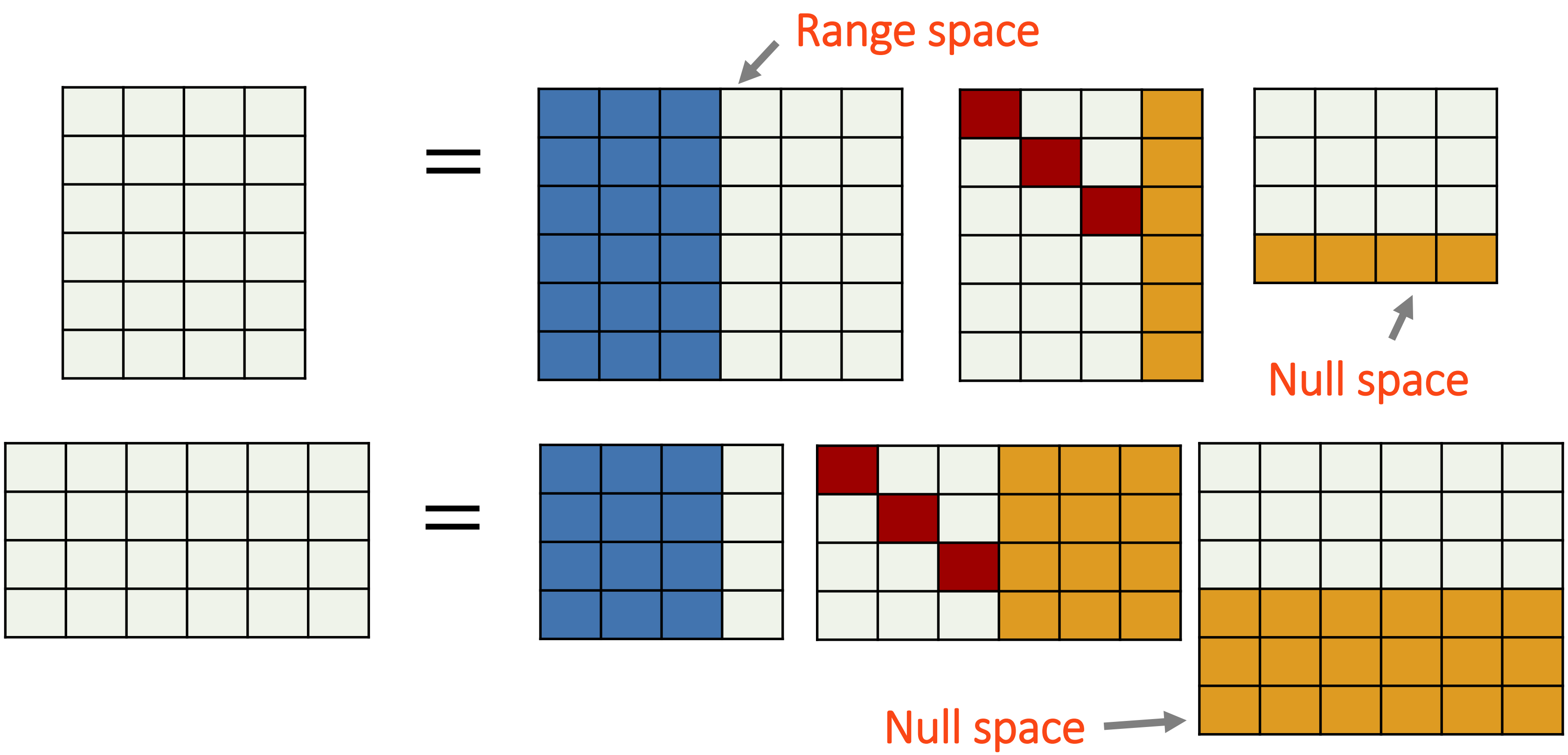
Diagonalization

- **Singular Value Decomposition Properties**
  - **Null space:** Columns of V corresponding to 0 singular values
  - **Range Space:** Columns of U corresponding to non-zero singular values



Diagonalization

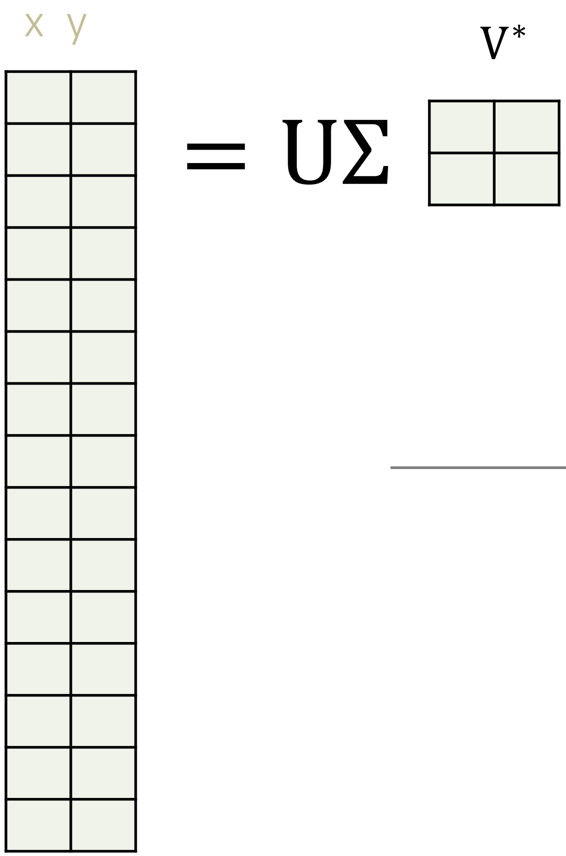
- **Singular Value Decomposition Properties**
  - **Null space:** Columns of V corresponding to 0 singular values
  - **Range Space:** Columns of U corresponding to non-zero singular values

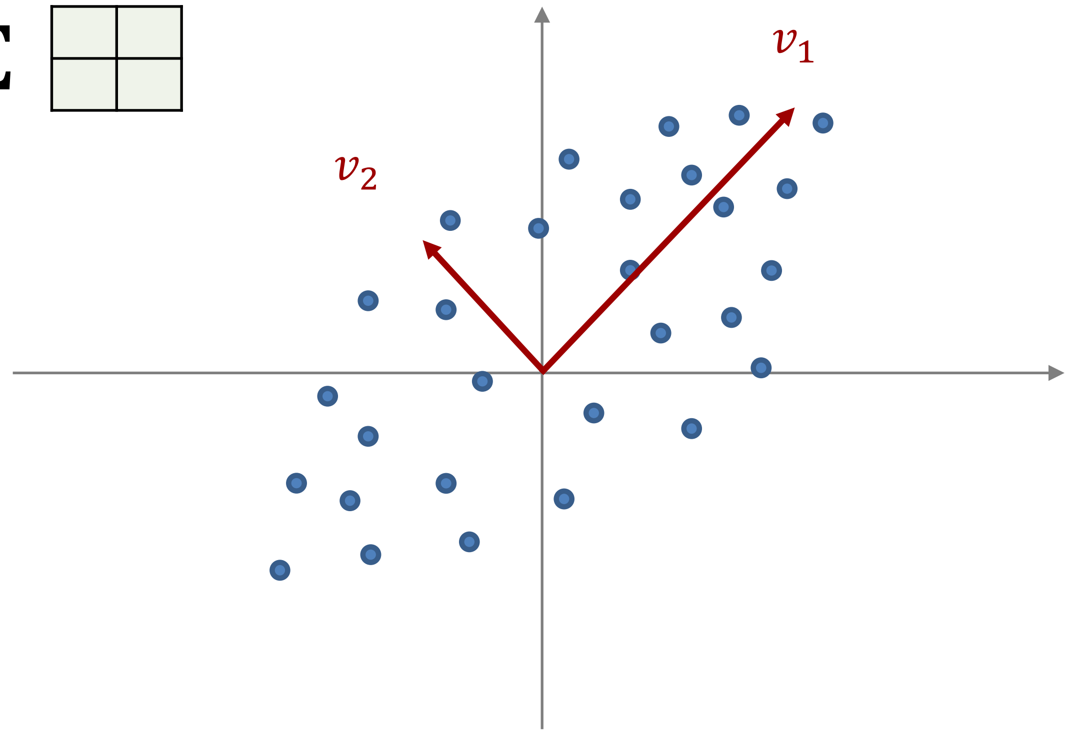


Diagonalization

Singular Value Decomposition Properties

- Principal Components

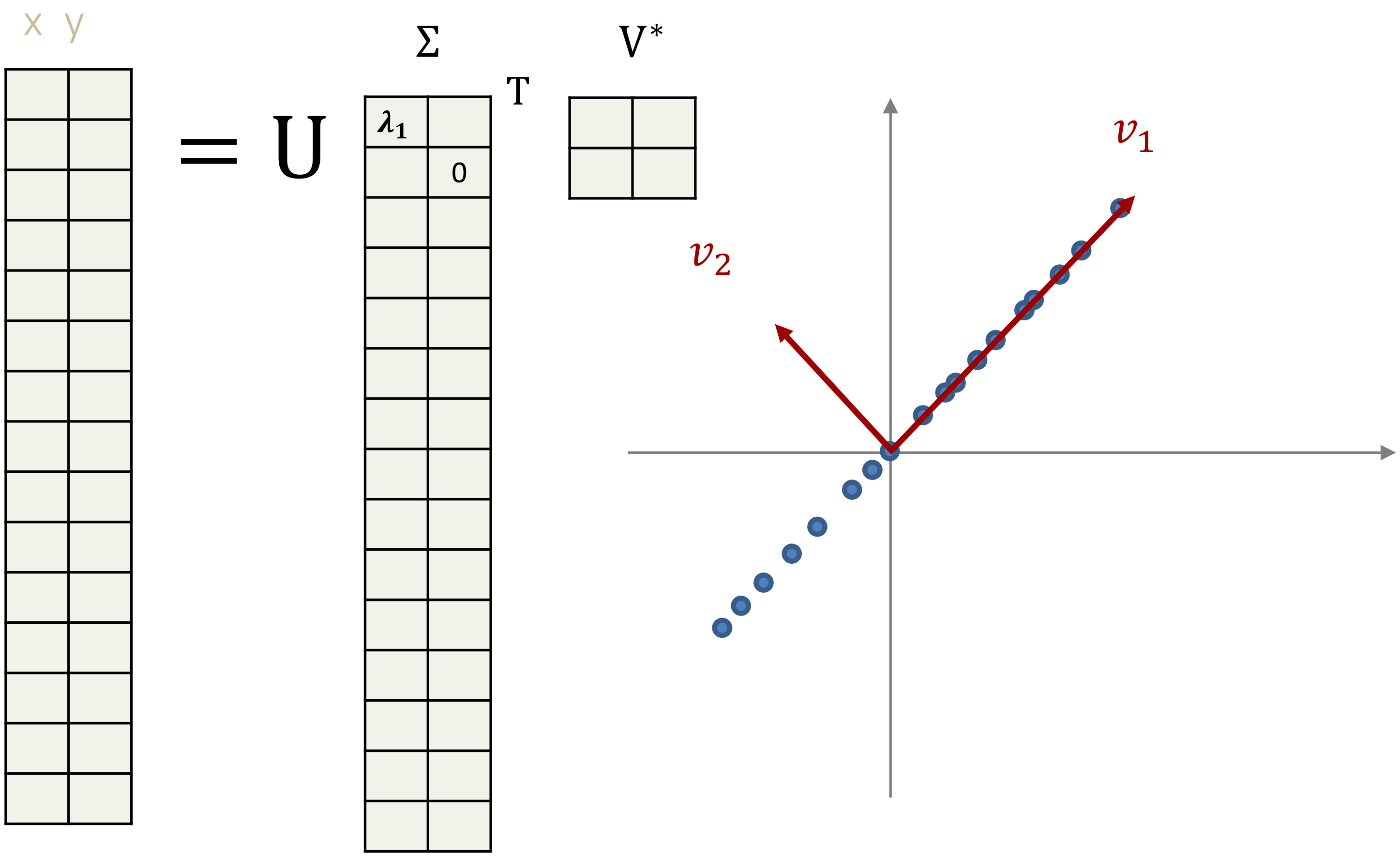




Diagonalization

Singular Value Decomposition Properties

- Principal Components



Diagonalization

- **Question:** When is singular value decomposition useful?

Diagonalization

- **Question:** When is singular value decomposition useful?
  - **List below is from Wikipedia:**
  - Pseudoinverse
  - Solving homogeneous linear equation (i.e., finding the null space)
  - Total Least Squares
  - Range Space, Null Space, and Rank
  - Low-rank matrix approximation
  - Separable models
  - Nearest Orthogonal Matrix
  - Find optimal rotations (The Kabsch algorithm)
  - Genomic Signal Processing
  - Big Data
  - [Blind signal separation] (added by me)

Example Application: Recommendation Engines

Diagonalization

- **Recommendation Engines (collaborative filtering)**
  - Recommendations based on rating

Products

Customers

	2	5			2
			1	2	3
2	3				2
2	1		4	1	5

Source: Badrul M. Sarwar et al., “Application of Dimensionality Reduction in Recommender System -- A Case Study,” ACM WedKDD Workshop, 2000.

Diagonalization

- **Recommendation Engines (collaborative filtering)**
  - Recommendations based on rating (step 1): Fill ratings matrix

Products

Customers

	2	5			2
			1	2	3
2	3				2
2	1		4	1	5

→

2	2	5	2.5	1.5	2
2	2	5	1	2	3
2	3	5	2.5	1.5	2
2	1	5	4	1	5

↑ ↑ ↑ ↑ ↑ ↑

Fill in average rate for each product

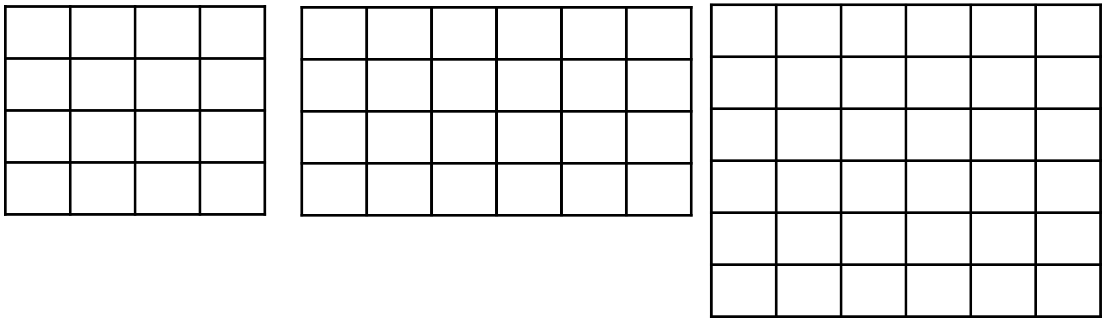
Source: Badrul M. Sarwar et al., "Application of Dimensionality Reduction in Recommender System -- A Case Study," ACM WedKDD Workshop, 2000.

- **Recommendation Engines (collaborative filtering)**
  - Recommendations based on rating (step 2): Compute the SVD

Products

Customers						
	2	2	5	2.5	1.5	2
	2	2	5	1	2	3
	2	3	5	2.5	1.5	2
	2	1	5	4	1	5

=



Source: Badrul M. Sarwar et al., “Application of Dimensionality Reduction in Recommender System -- A Case Study,” ACM WedKDD Workshop, 2000.

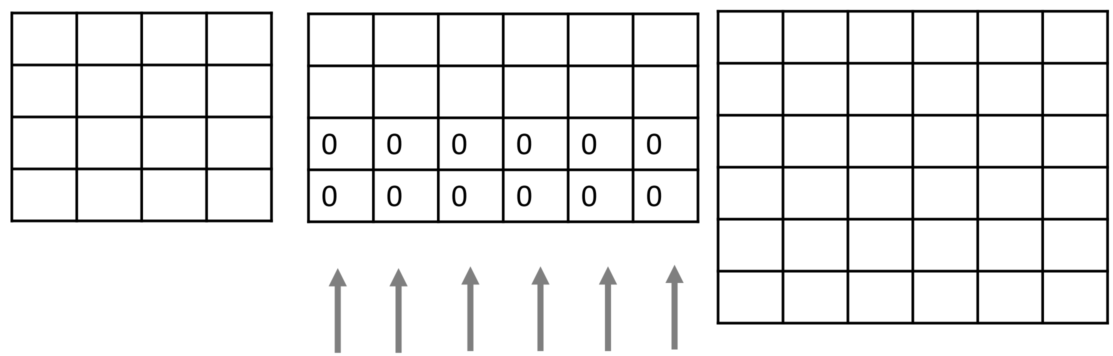
Diagonalization

- **Recommendation Engines (collaborative filtering)**
  - Recommendations based on rating (step 3): Compute low-rank approx.

Products

Customers

2	2	5	2.5	1.5	2
2	2	5	1	2	3
2	3	5	2.5	1.5	2
2	1	5	4	1	5



Force S to be lower rank

Products

$=$

1.9	2.2	4.9	2.1	1.6	2.4
2.0	2.3	4.9	2.0	1.6	2.3
2.1	2.6	5.1	2.0	1.8	2.2
2.0	1.0	5.0	4.0	1.0	5.0

Customers

Source: Badrul M. Sarwar et al., "Application of Dimensionality Reduction in Recommender System -- A Case Study," ACM WedKDD Workshop, 2000.

Example Application: Facial Recognition

Diagonalization

- **Eigenfaces** (images from http://www.quantumblah.org/?m=20120116)
  - Start with collection of faces

Diagonalization

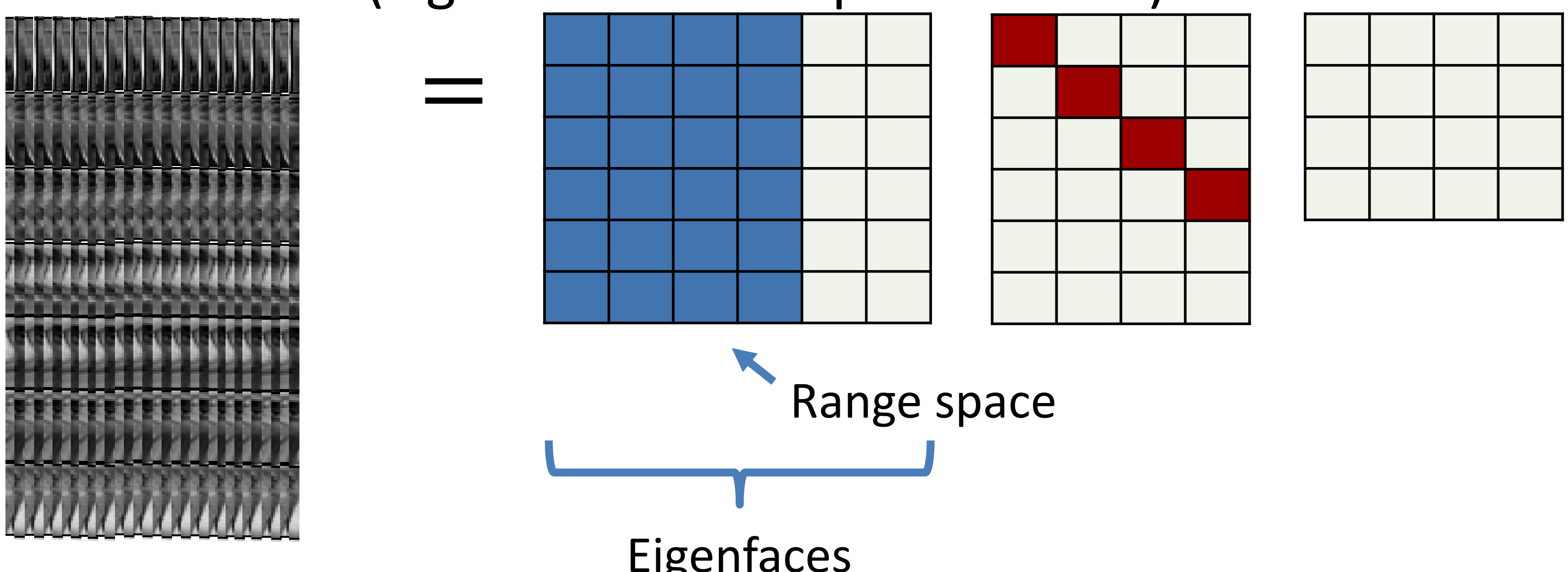
- **Eigenfaces** (images from http://www.quantumblah.org/?m=20120116)
  - Start with collection of faces
  - Vectorize each face

$N \times N$ matrix

N^2 vector

Diagonalization

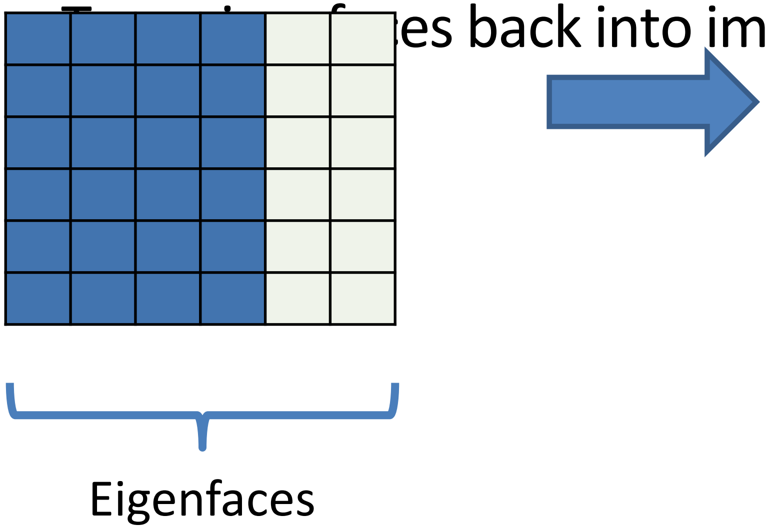
- **Eigenfaces** (images from http://www.quantumblah.org/?m=20120116)
  - Start with collection of faces
  - Vectorize each face
  - Concatenate vectors
  - Perform SVD (Eigenvalue decomp. if $N^2 = M$)

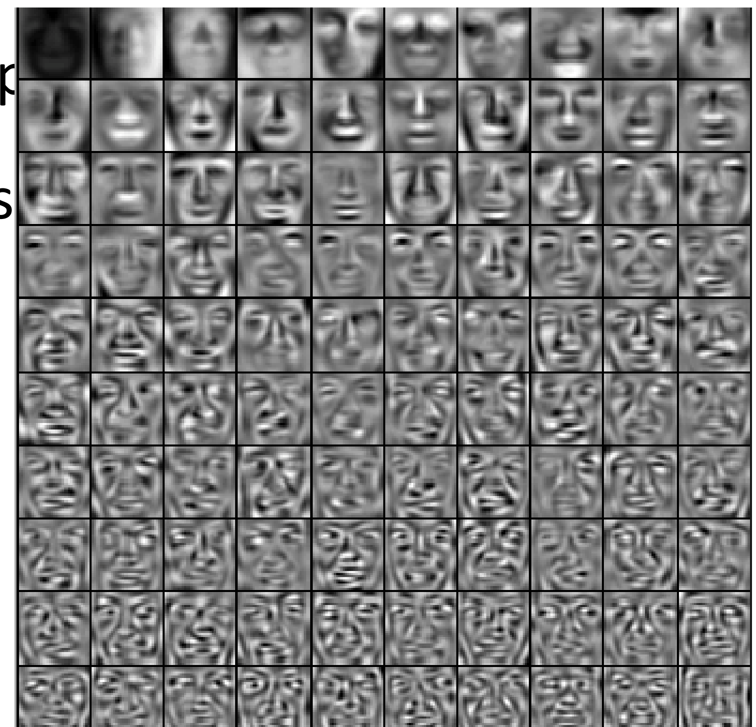


$N^2 \times M$ matrix

Diagonalization

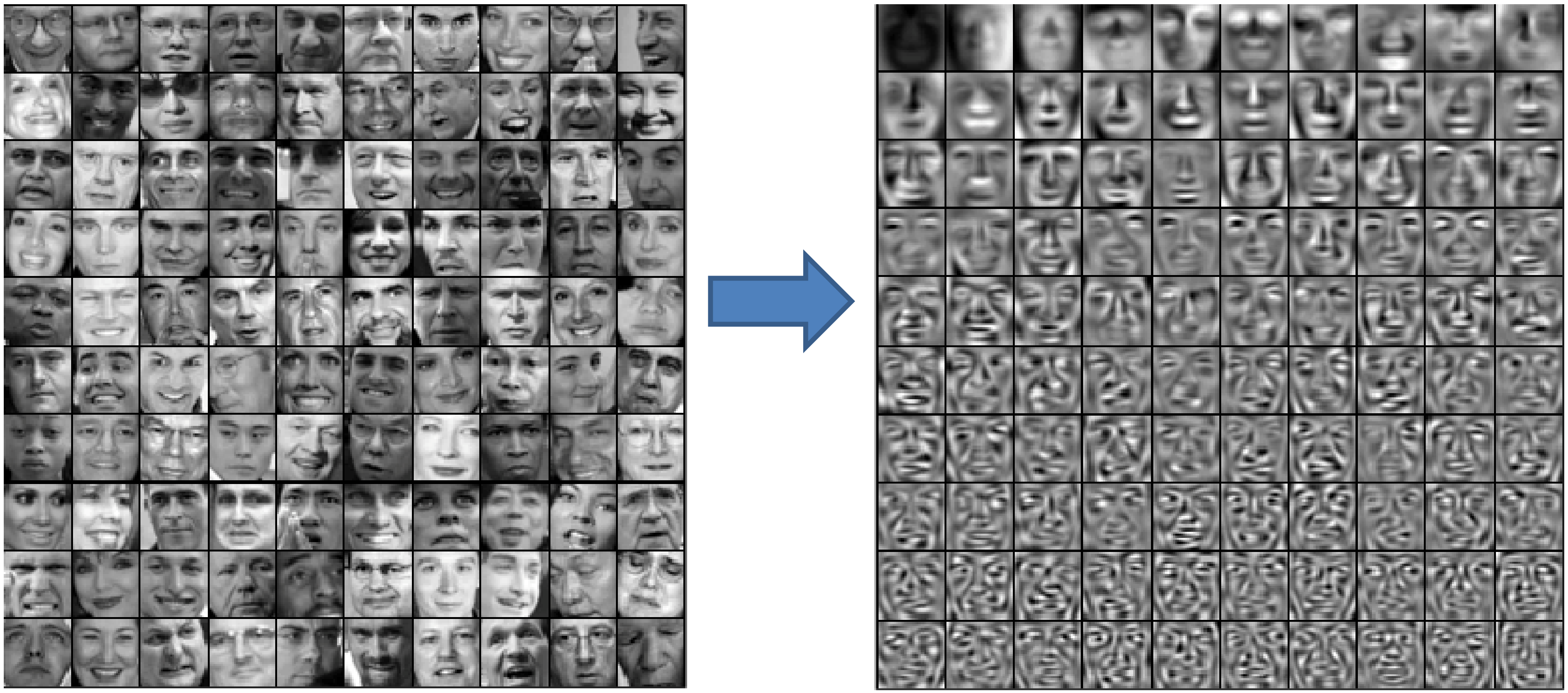
- **Eigenfaces** (images from http://www.quantumblah.org/?m=20120116)
  - Start with collection of faces
  - Vectorize each face
  - Concatenate vectors
  - Perform SVD (Eigenvalue decomp
  - es back into images





Diagonalization

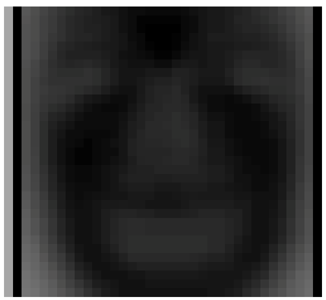
- **Eigenfaces** (images from http://www.quantumblah.org/?m=20120116)



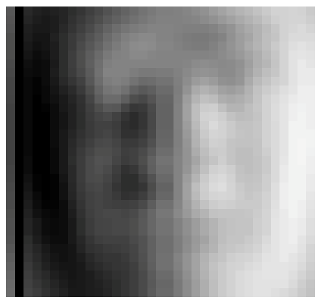
Diagonalization

- **Eigenfaces Eigenfaces** (images from http://www.quantumblah.org/?m=20120116)
  - Eigenfaces represent general features of faces

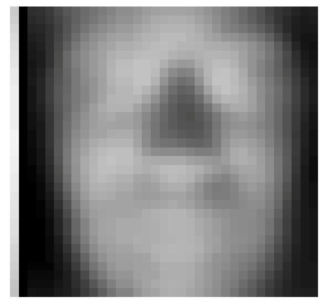
1st Eigenface



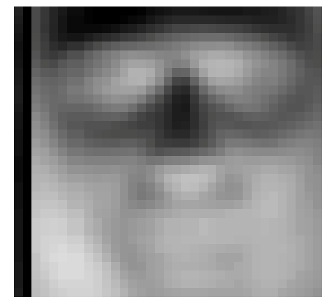
2nd Eigenface



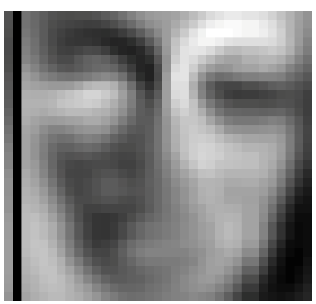
3rd Eigenface



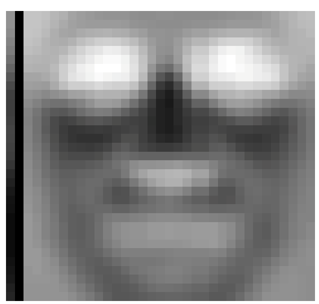
4th Eigenface



5th Eigenface



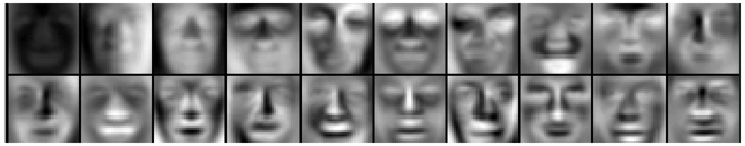
6th Eigenface



Diagonalization

- **Eigenfaces** (images from http://www.quantumblah.org/?m=20120116)

Reduce dimensions



Inner product between each eigenface and test image

Test images

High values represent face,
low values represent no face

High value | Low value

Lecture 27: Wavelets to Modern Signal Processing

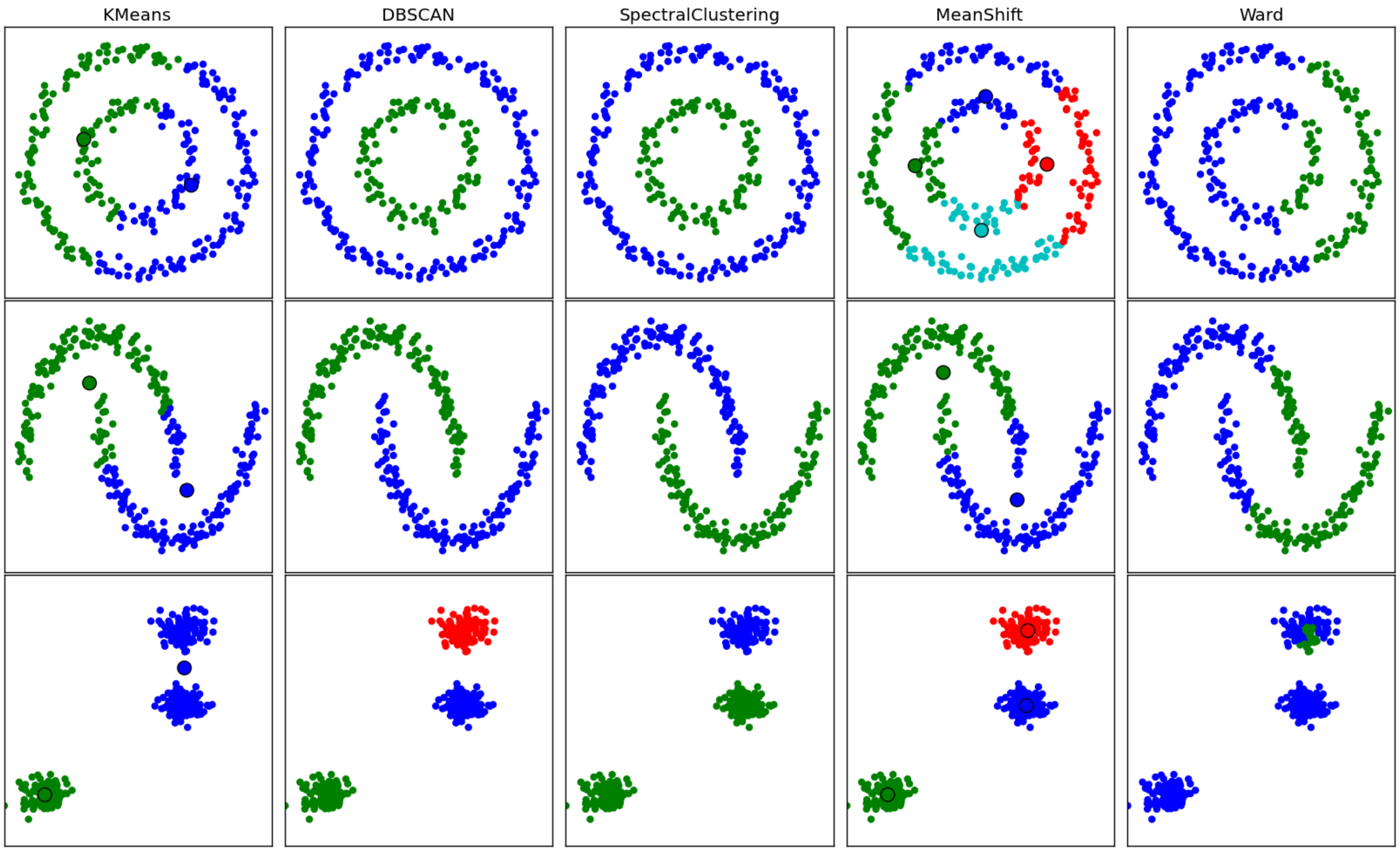
Foundations of Digital Signal Processing

Outline

- Review of Filter Banks and Wavelets
- Applications of Wavelets
- Applications of Filter Banks and Time-Frequency Representations
- Modern Signal Processing: Vectors and Matrices
- Modern Signal Processing: Compressive Sensing
- Modern Signal Processing: Diagonalization
- **Modern Signal Processing: Graph Signal Processing**

Spectral Clustering

Spectral Clustering Examples

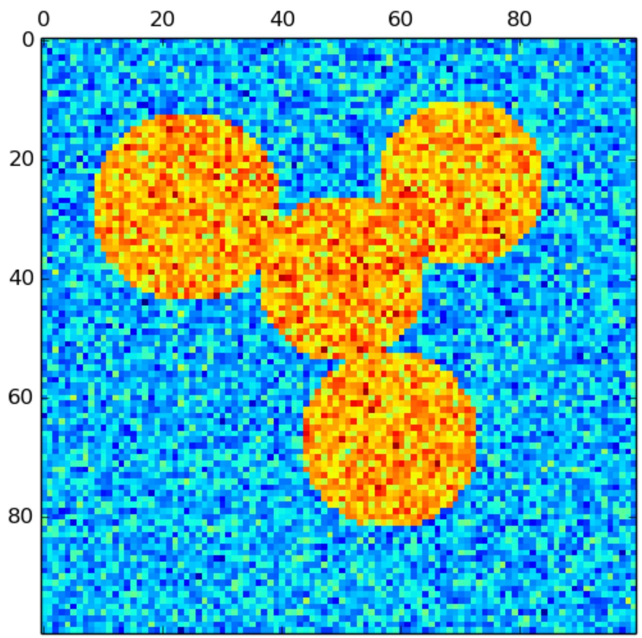


From: http://jaquesgrobler.github.io/Online-Scikit-Learn-stat-tut/modules/clustering.html

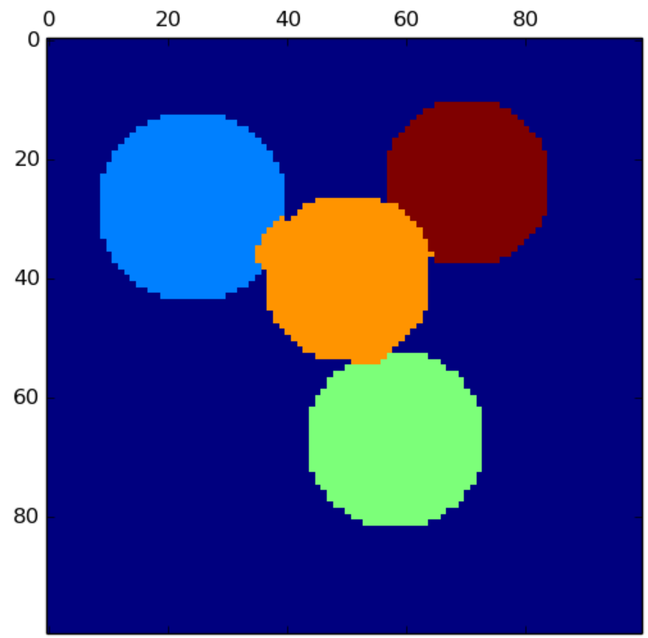
Spectral Clustering

Spectral Clustering Examples

Original



Spectral Clustering



From: http://scikit-learn.org/stable/auto_examples/cluster/plot_segmentation_toy.html#example-cluster-plot-segmentation-toy-py

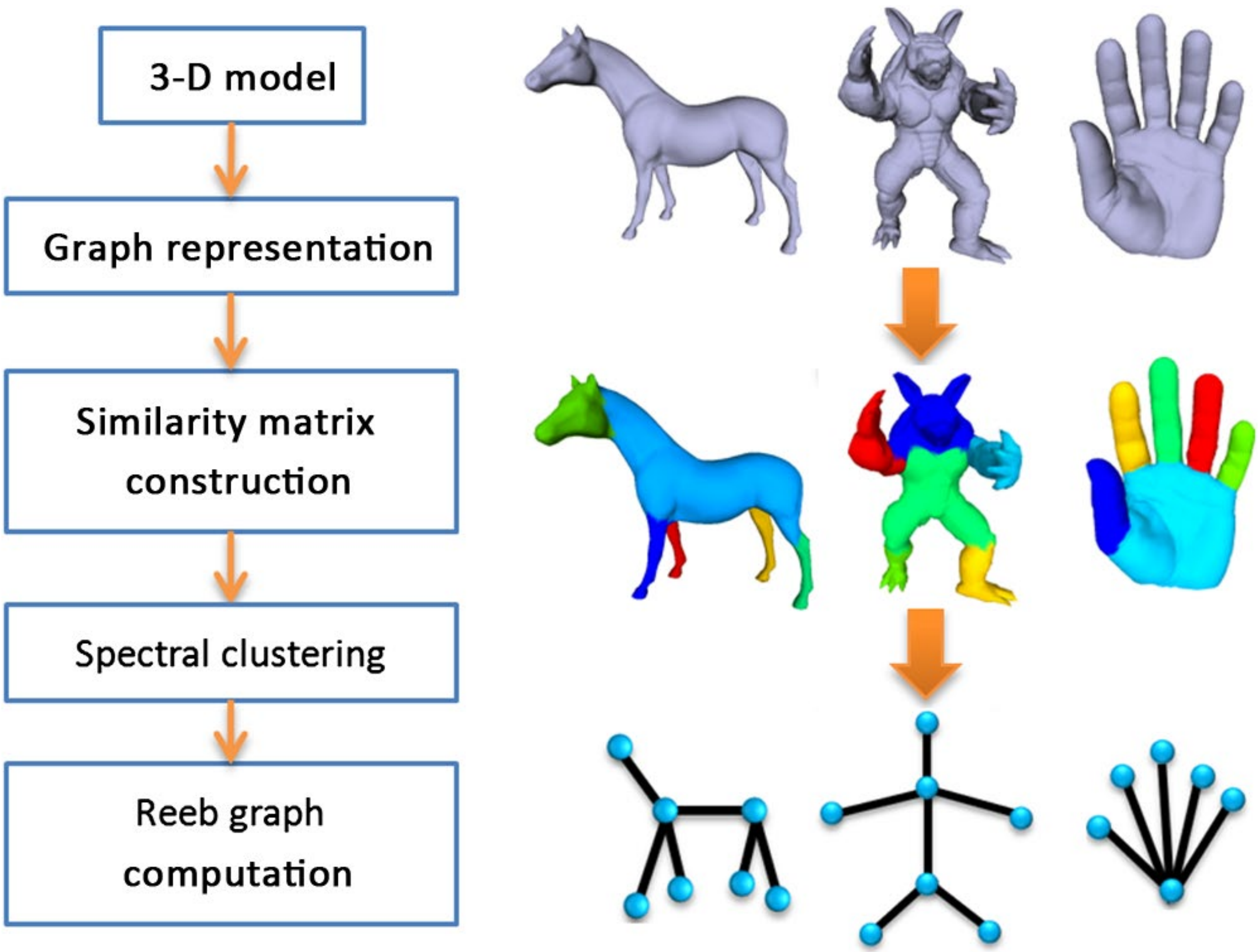
Spectral Clustering Examples

Two different clustering approaches after the eigen-decomposition

From: http://scikit-learn.org/stable/auto_examples/cluster/plot_lena_segmentation.html#example-cluster-plot-lena-segmentation-py

Spectral Clustering

Spectral Clustering Examples



From: http://opticalengineering.spiedigitallibrary.org/article.aspx?articleid=1183272%20

Signals with Structure

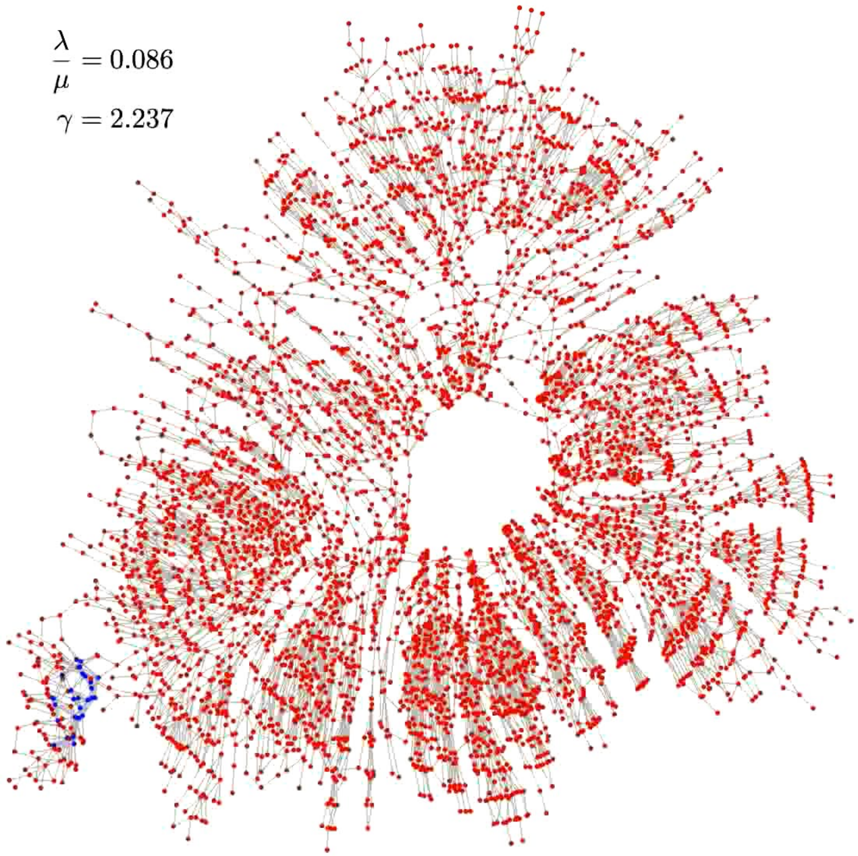
- **So, what have we learned?**
  - Signals have structure
  - Structure can be imposed with graphs
  - Graphs can improve results and quicken analysis

- **Question: How else could we use graphs?**

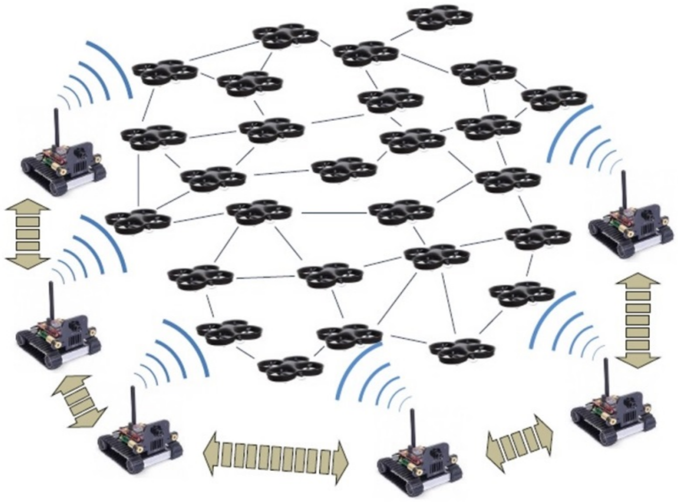
Signals with Structure

- **Some other examples**

Analyzing the spread of infection

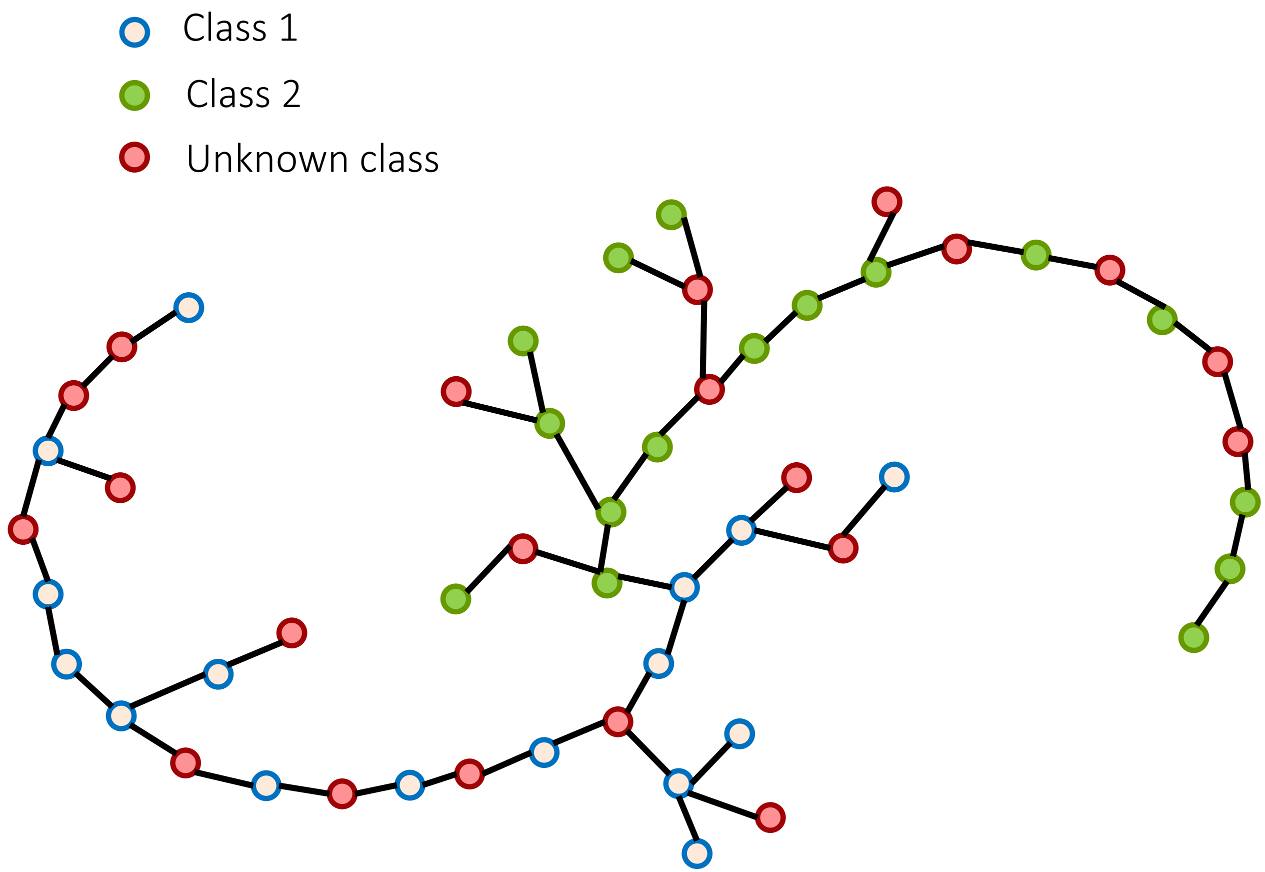


Collaboration between independent agents



Spectral Clustering

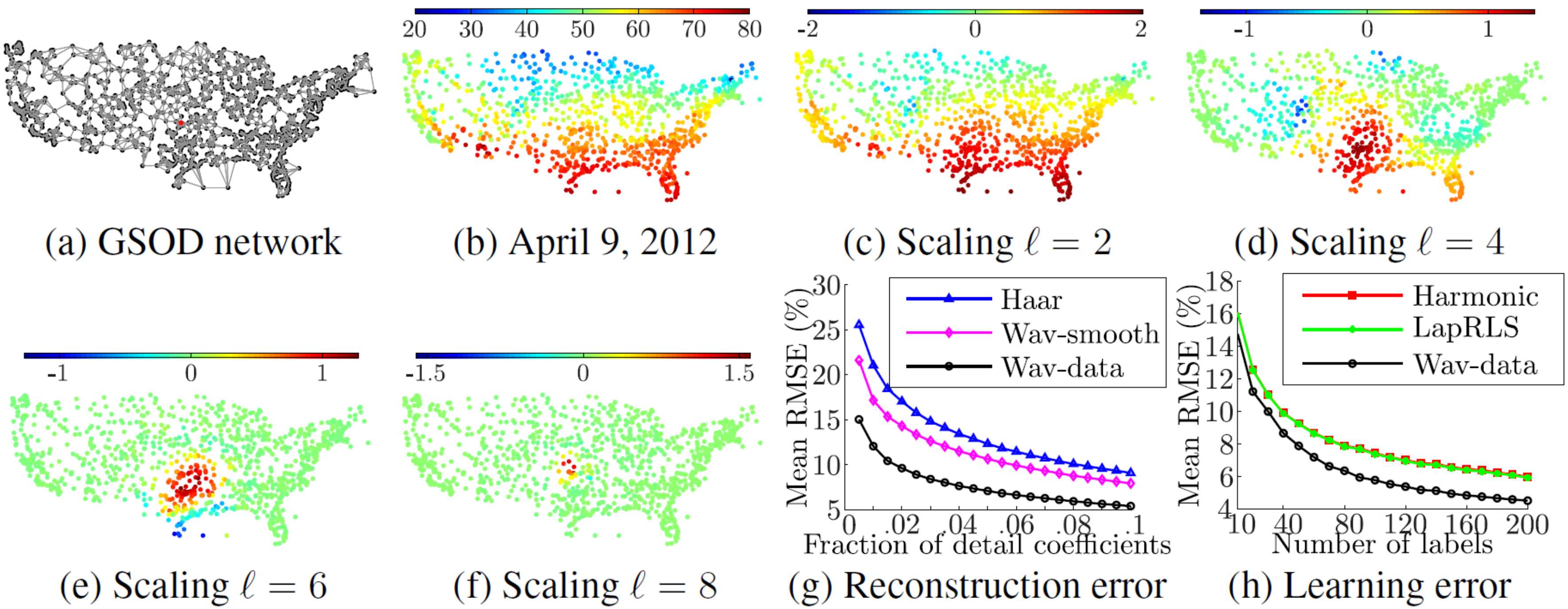
Examples [semi-supervised learning and classification]



Easy classification problem

Signals with Structure

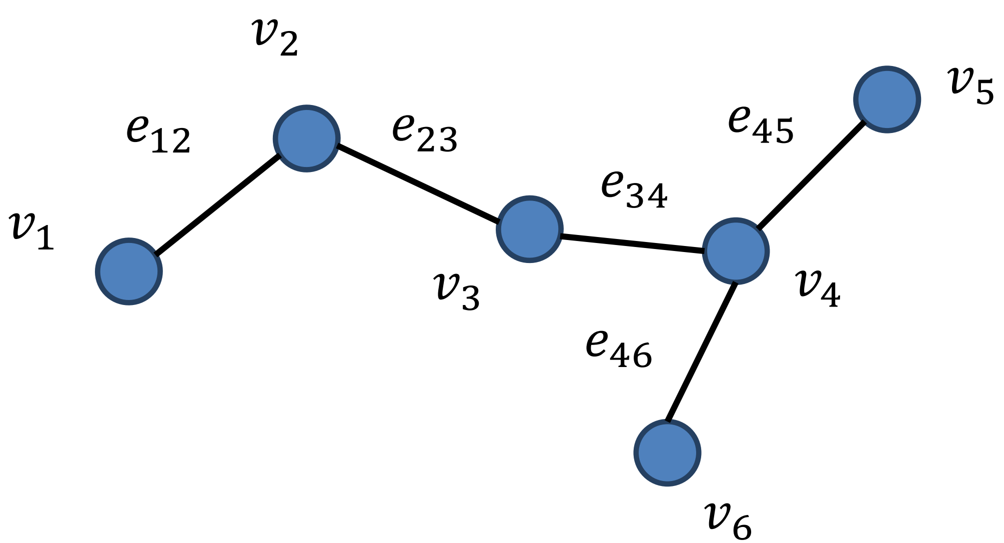
■ Examples [graph filtering and graph wavelets]



From: http://papers.nips.cc/paper/5046-wavelets-on-graphs-via-deep-learning.pdf

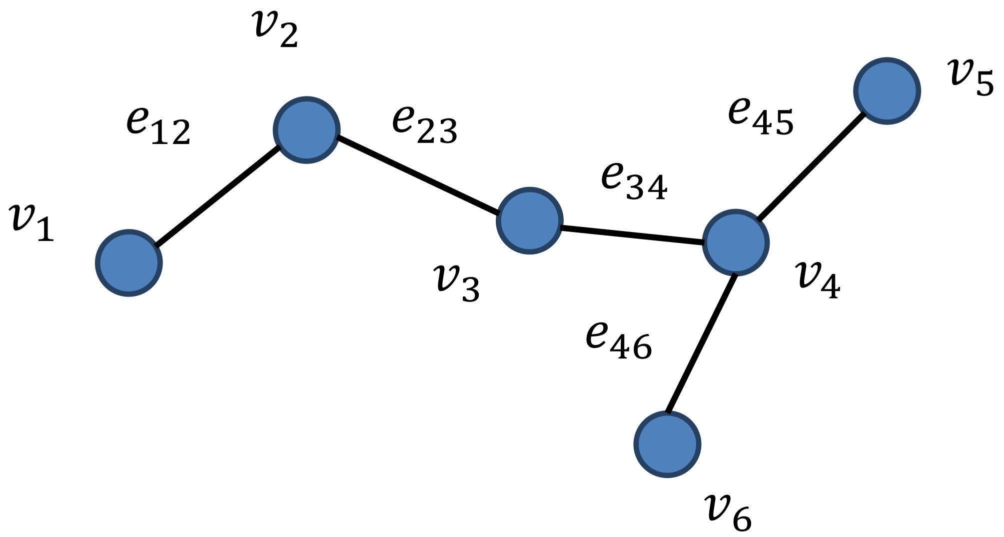
Graph Processing

- **We define a graph by nodes and edges $G = \{V, \mathcal{E}\}$**
  - Let $v_1, v_2, v_3, v_4, \ldots, v_N \in V$ be vertices in the graph
  - Let $e_{ij} \in \mathcal{E}$ be an edge in the graph that connects node i to node j



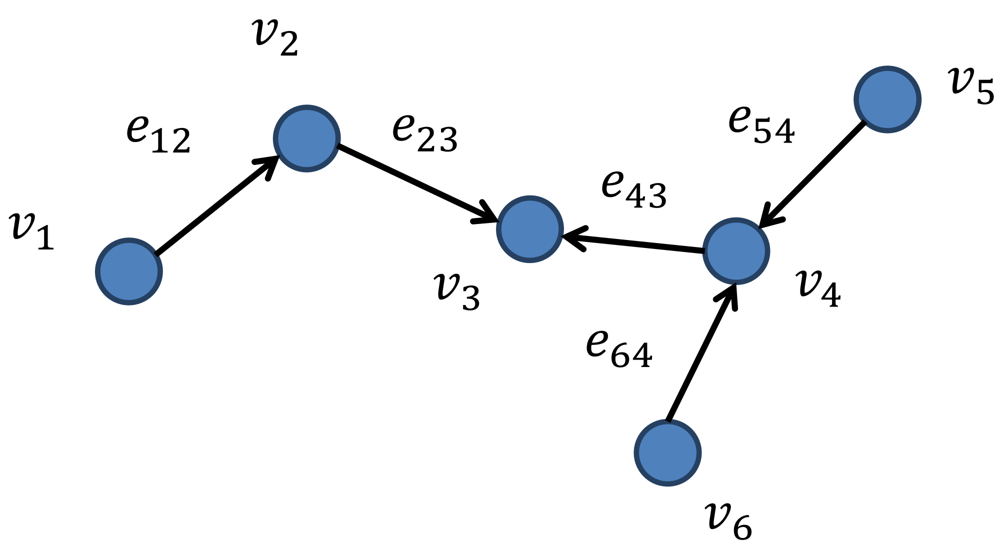
Graph Processing

- **We define a graph by nodes and edges $G = \{V, \mathcal{E}\}$**
  - Let $v_1, v_2, v_3, v_4, \ldots, v_N \in V$ be vertices in the graph
  - Let $e_{ij} \in \mathcal{E}$ be an edge in the graph that connects node i to node j
- **Undirected graph**
  - $e_{ij} = e_{ji}$



Graph Processing

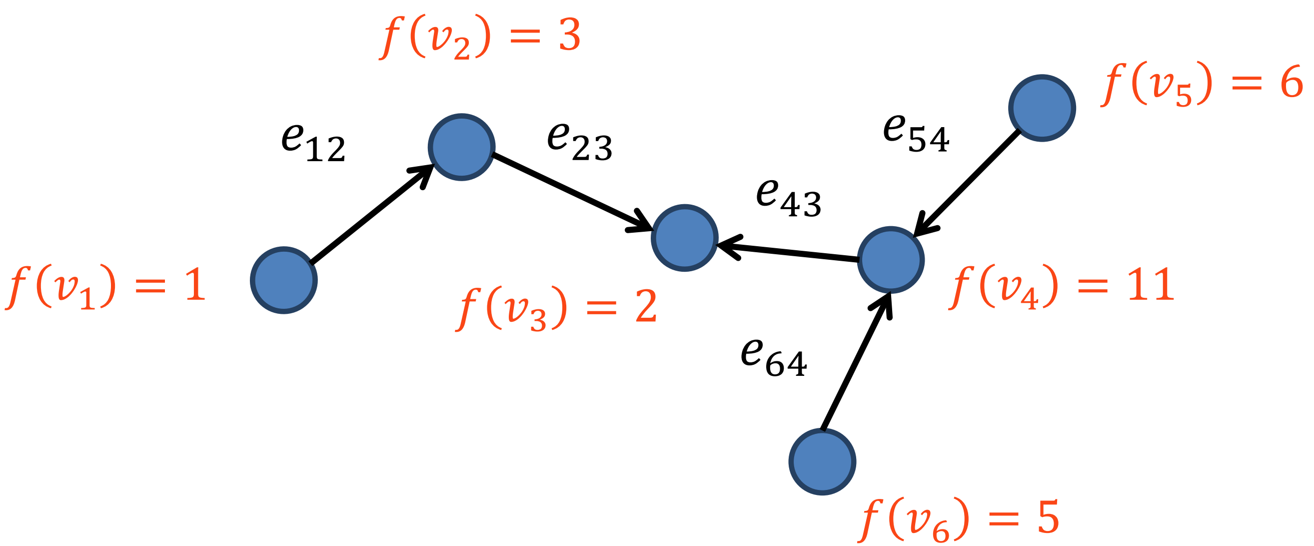
- **We define a graph by nodes and edges $G = \{V, \mathcal{E}\}$**
  - Let $v_1, v_2, v_3, v_4, \dots, v_N \in V$ be vertices in the graph
  - Let $e_{ij} \in \mathcal{E}$ be an edge in the graph that connects node i to node j
- **Directed graph**
  - $e_{ij} \neq e_{ji}$



Graph Processing

- **We define a graph by nodes and edges $G = \{V, \mathcal{E}\}$**
  - Let $v_1, v_2, v_3, v_4, \dots, v_N \in V$ be vertices in the graph
  - Let $e_{ij} \in \mathcal{E}$ be an edge in the graph that connects node i to node j

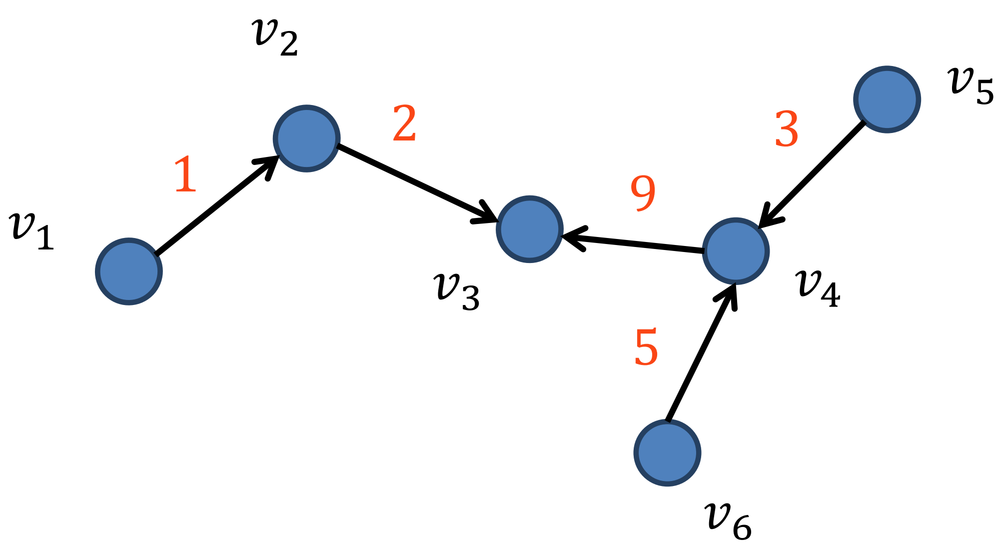
- **Vertices**
  - Vertices can possess signals / functions: $f: V \to \mathbb{R}^N$



Graph Processing

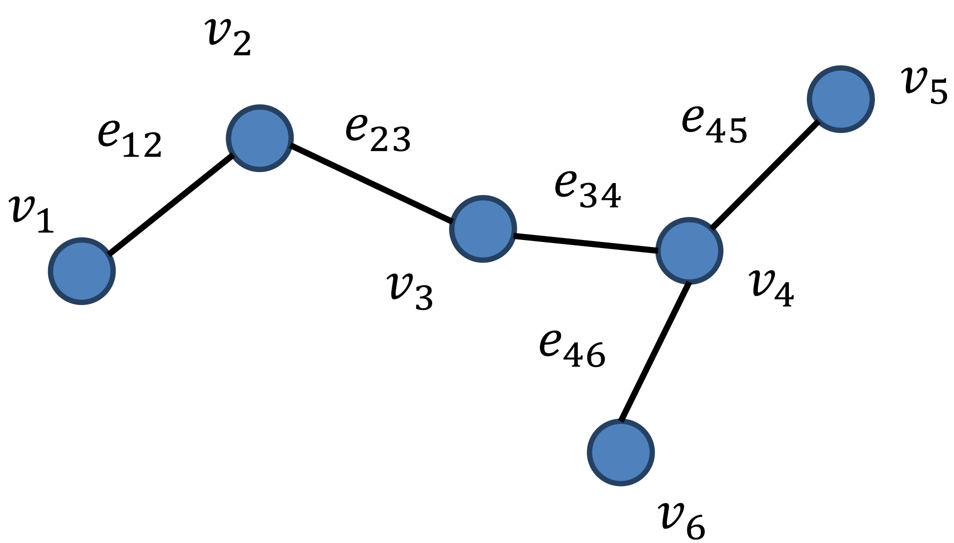
- **We define a graph by nodes and edges $G = \{V, \mathcal{E}, W\}$**
  - Let $v_1, v_2, v_3, v_4, \ldots, v_N \in V$ be vertices in the graph
  - Let $e_{ij} \in \mathcal{E}$ be an edge in the graph that connects node i to node j

- **Edges**
  - Edges can have weights W



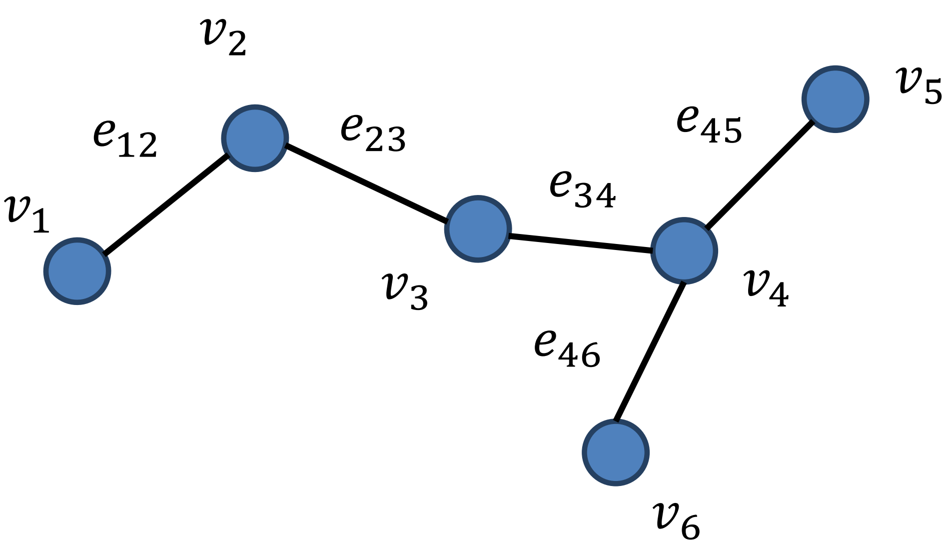
Analyzing Graphs

- **Adjacency Matrix** A
  - $a_{i,j} = \begin{cases} w_{i,j} & \text{if} \quad e_{i,j} \in \mathcal{E} \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$
- **What is the adjacency matrix of the graph below?**
  - Assume all weights are 1



Analyzing Graphs

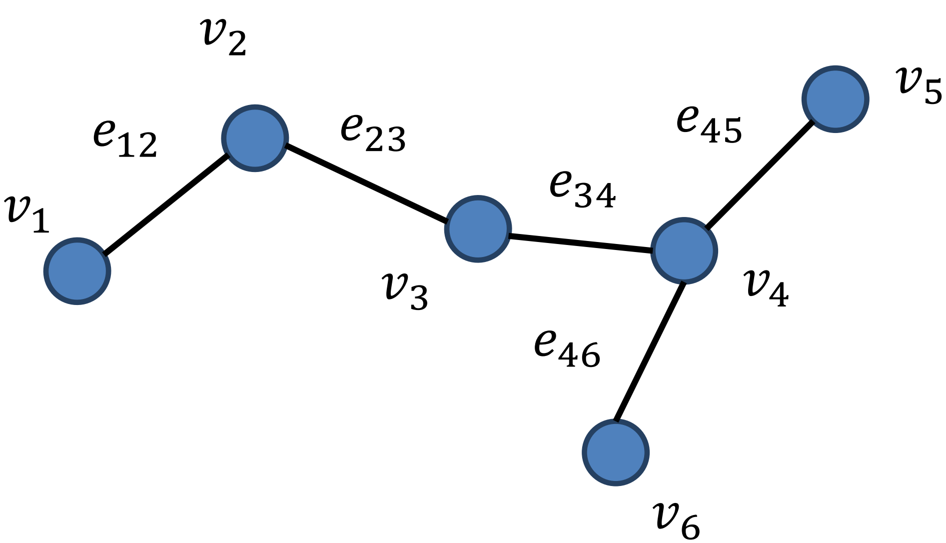
- **Adjacency Matrix A**
  - $$a_{i,j} = \begin{cases} w_{i,j} & \text{if} \quad e_{i,j} \in \mathcal{E} \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$
- **What is the adjacency matrix of the graph below?**
  - Assume all weights are 1



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Analyzing Graphs

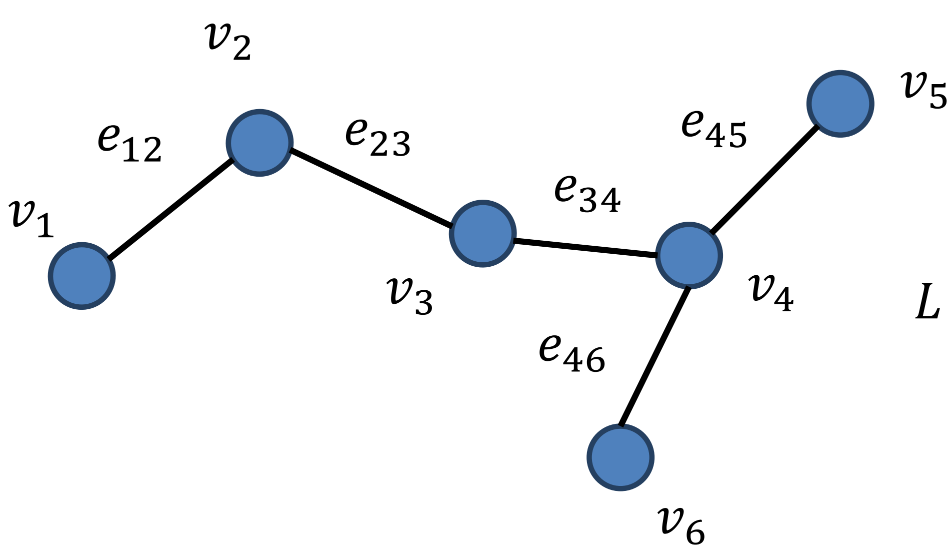
- **(Un-normalized) Laplacian Matrix** L
  - $L = D - A$, $D =$ sum of weights connected to each vertex
- **What is the Laplacian matrix of the graph below?**
  - Assume all weights are 1



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Analyzing Graphs

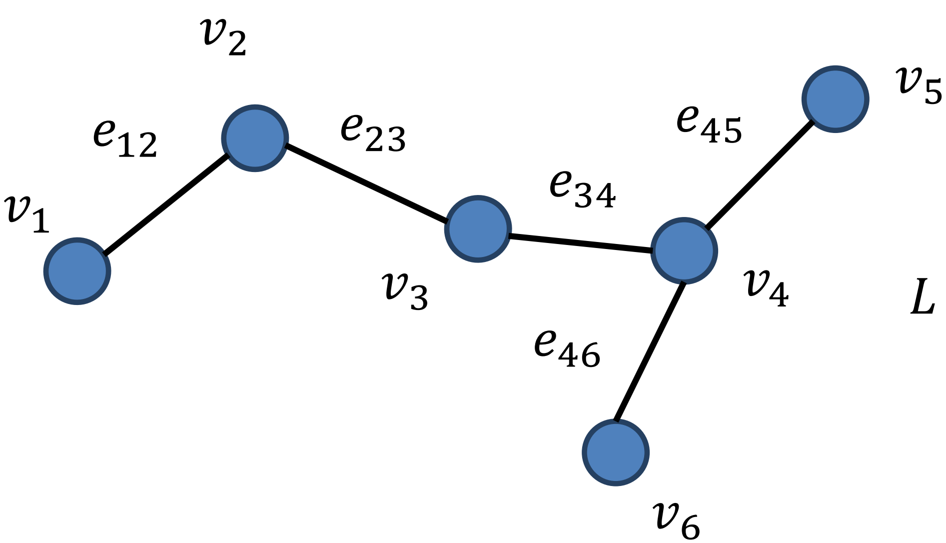
- **(Un-normalized) Laplacian Matrix L**
  - $L = D - A$, $D =$ sum of weights connected to each vertex
- **What is the Laplacian matrix of the graph below?**
  - Assume all weights are 1



$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Analyzing Graphs

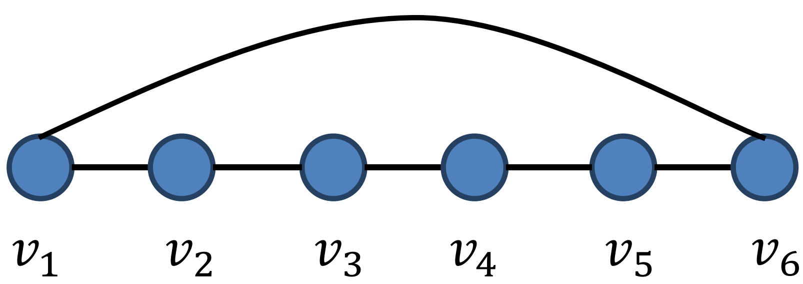
- **Properties of the Laplacian Matrix L**
  - Eigenvalues are real
  - Eigenvectors are orthogonal
  - $\mathbf{1} \in \mathcal{N}(L)$
  - The Laplacian is a differencing operator



$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Analyzing Graphs

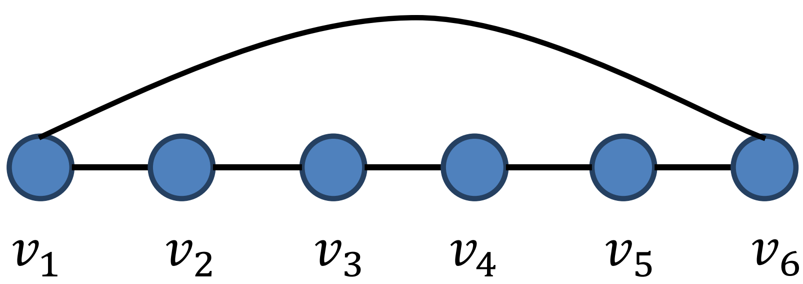
- **Properties of the Laplacian Matrix** L
  - Eigenvalues are real
  - Eigenvectors are orthogonal
  - $\mathbf{1} \in \mathcal{N}(L)$
  - The Laplacian is a differencing operator



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Analyzing Graphs

- **Properties of the Laplacian Matrix L**
  - Eigenvalues are real
  - Eigenvectors are orthogonal
  - $\mathbf{1} \in \mathcal{N}(L)$
  - The Laplacian is a differencing operator

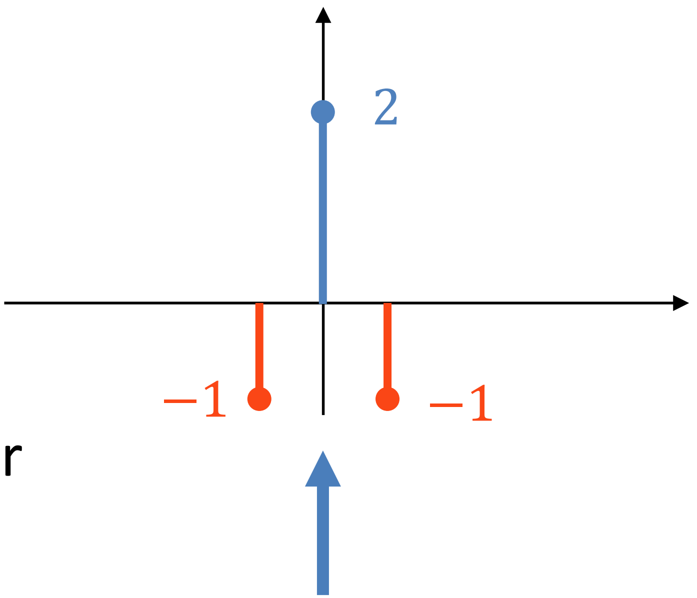


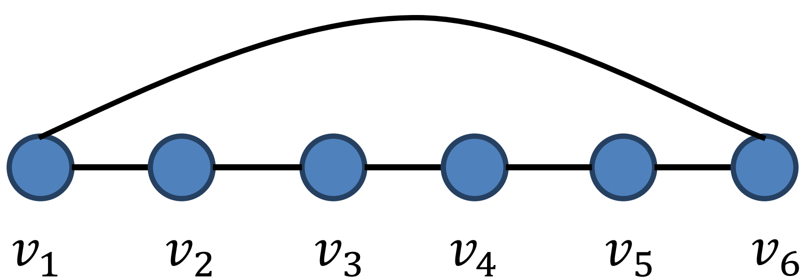
$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Analyzing Graphs

- **Properties of the Laplacian Matrix L**
  - Eigenvalues are real
  - Eigenvectors are orthogonal
  - $\mathbf{1} \in \mathcal{N}(L)$
  - The Laplacian is a differencing operator

Laplacian Filter

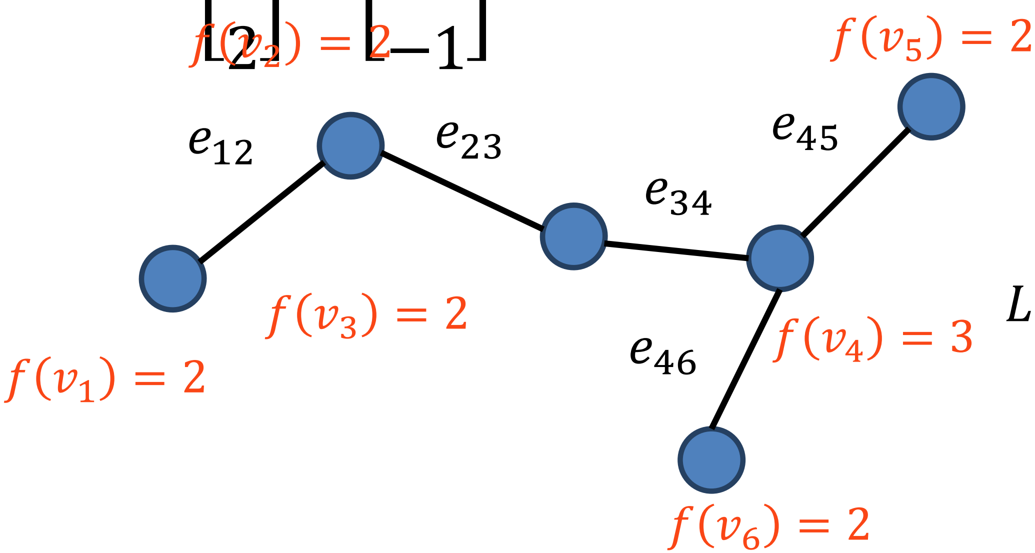




$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Analyzing Graphs

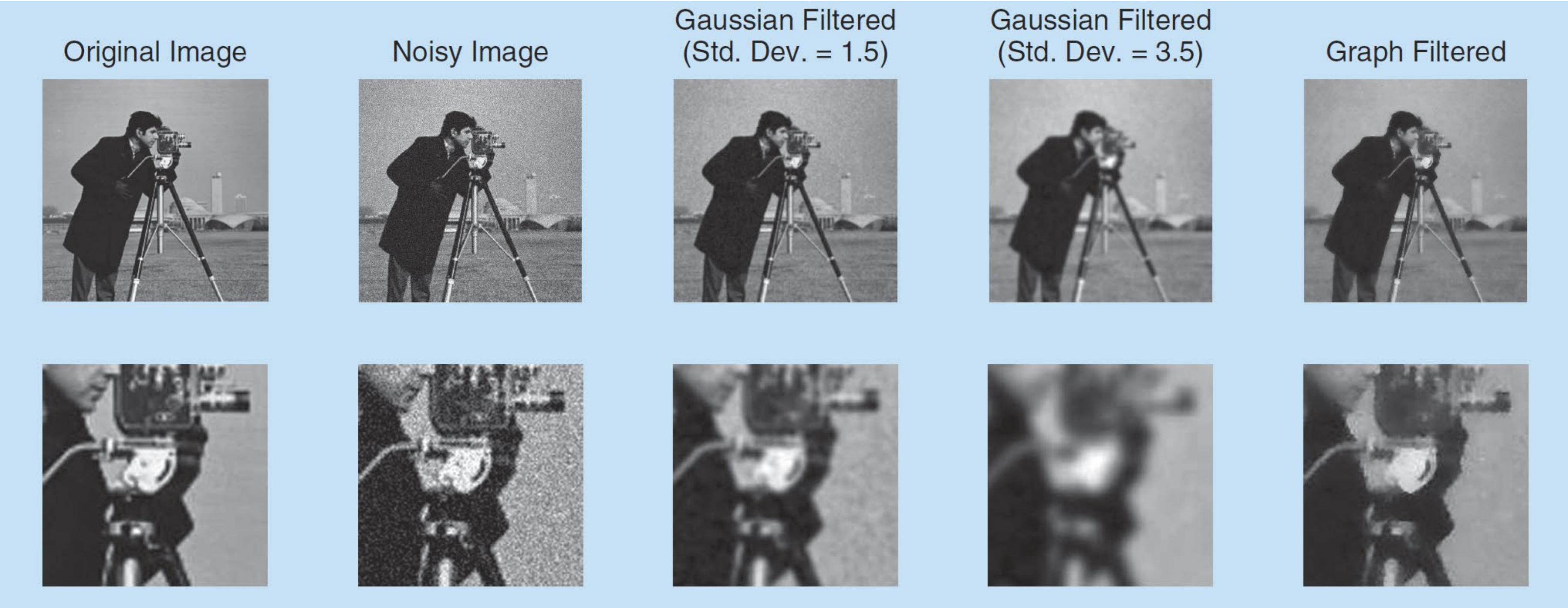
- **Properties of the Laplacian Matrix L**
  - $L\begin{bmatrix}2\\2\\2\\3\\2\\2\end{bmatrix} = \begin{bmatrix}0\\0\\-1\\3\\-1\\-1\end{bmatrix}$



$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Analyzing Graphs

- **Application of Smoothness**
  - Make a graph of related components (i.e., not edges) and optimize smoothness



From “The Emerging Field of Signal Processing on Graphs,” Signal Proc. Mag.