Introduction to Wavelets in Image Processing

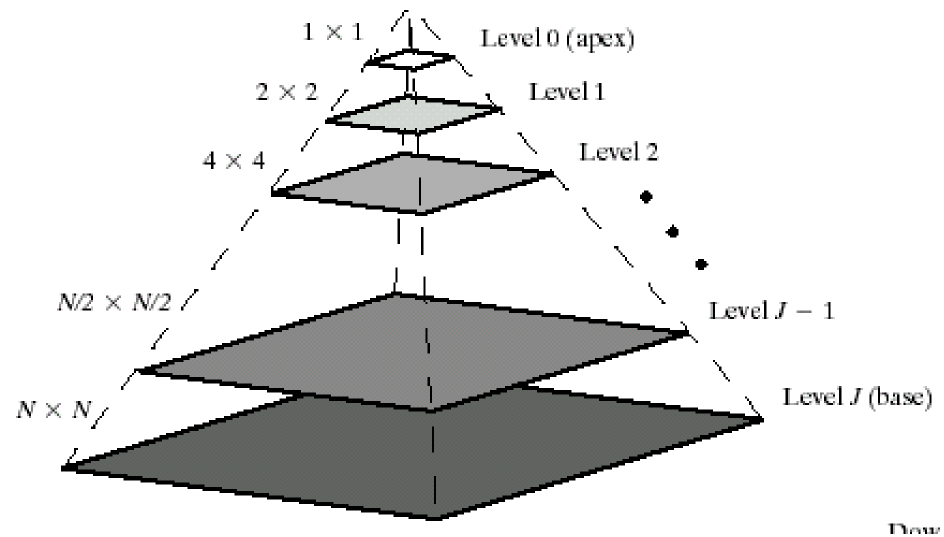
Pyramid Representation

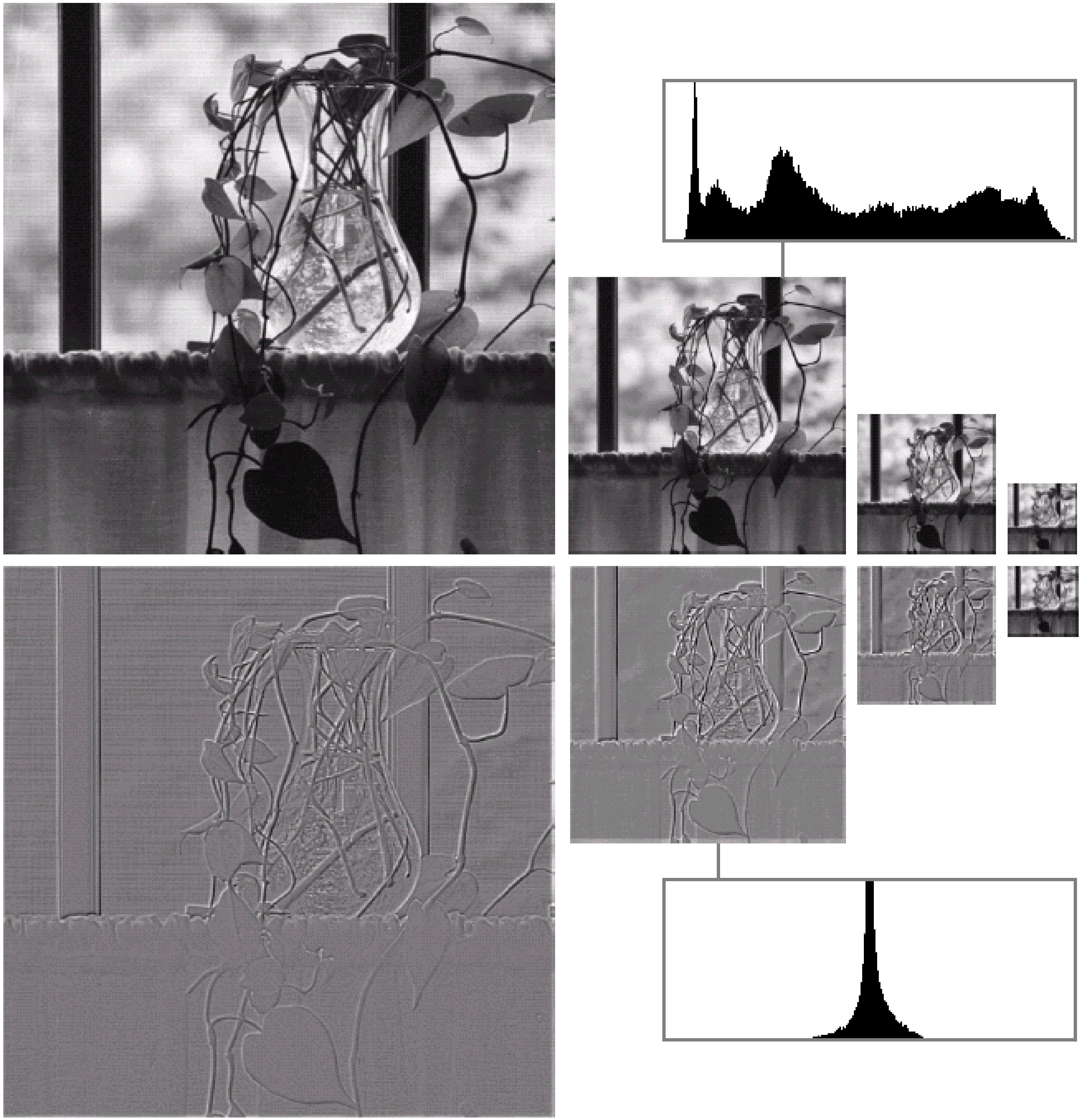
- Recall that we can create a multi-resolution pyramid of images
  多分辨率图像金字塔

- At each level, we just store the differences (residuals) between the image at that level and the predicted image from the next level

- We can reconstruct the image by just adding up all the residuals

- Advantage: residuals are easier to store





a
b

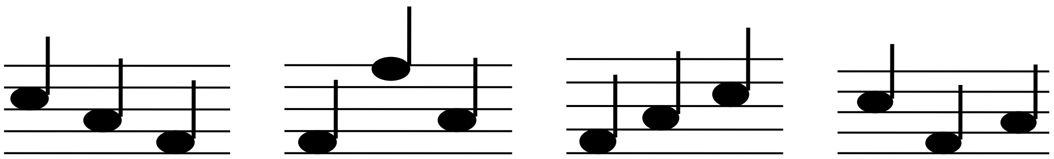
FIGURE 7.3 Two image pyramids and their statistics: (a) a Gaussian (approximation) pyramid and (b) a Laplacian (prediction residual) pyramid.

Wavelets

- Wavelets are a more general way to represent and analyze multiresolution images
- Can also be applied to 1D signals
- Very useful for
  - image compression (e.g., in the JPG-2000 standard)
  - removing noise

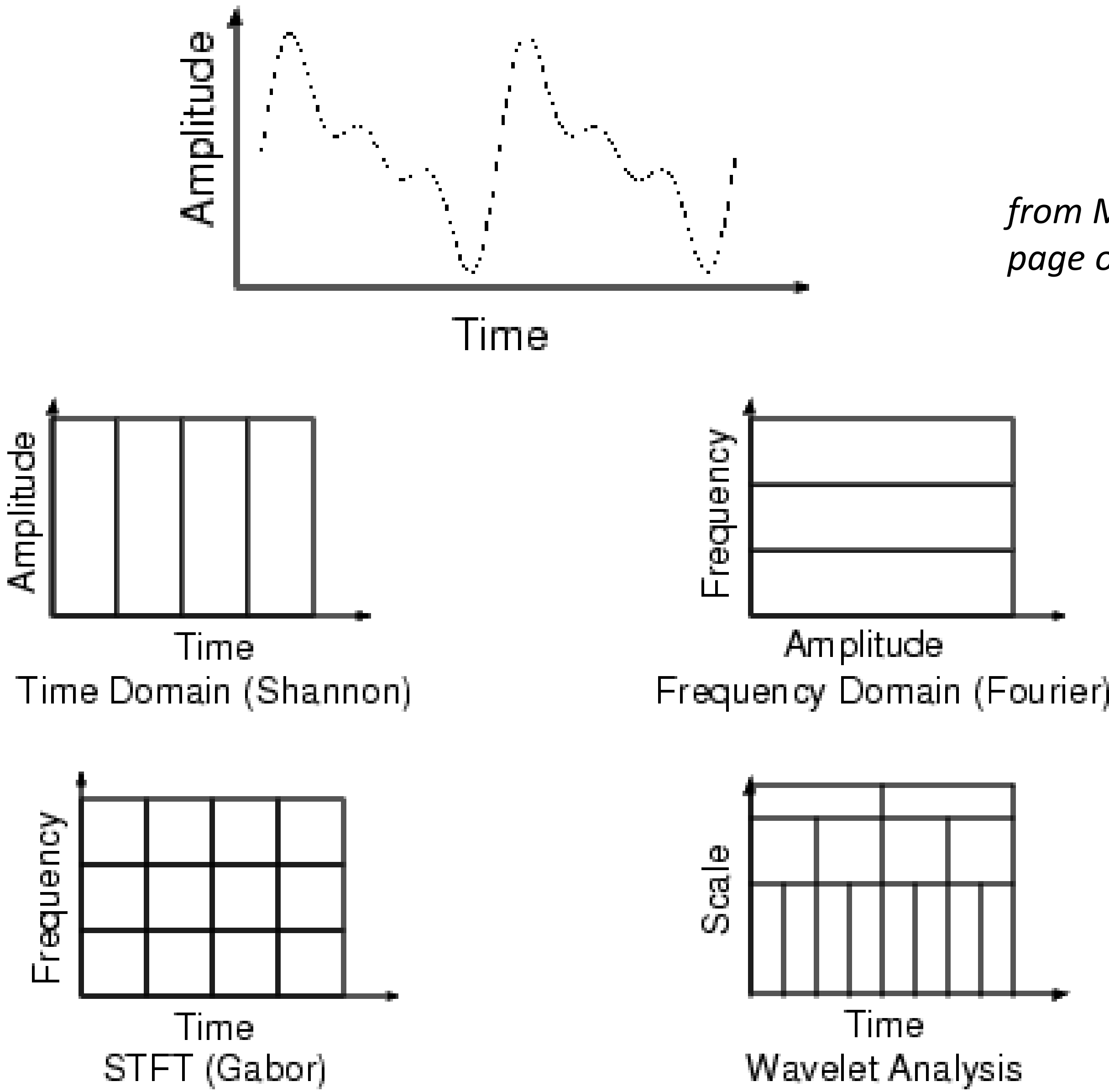
Wavelet Analysis

- Motivation
  - Sometimes we care about both frequency as well as time
  - Example: Music



(a) (b) (c) (d)

  - Time domain operations tell us “when”
  - Fourier domain operations tell us “frequency”



from Matlab help page on wavelets

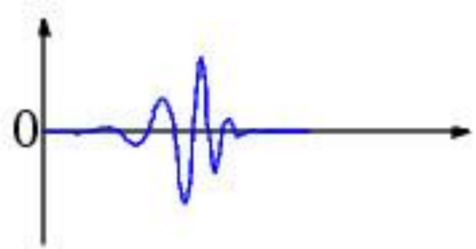
Continuous Wavelet Transform

- Define a function $\psi(x)$
  - assume $\psi(x)$ band-limited and its dc component = 0

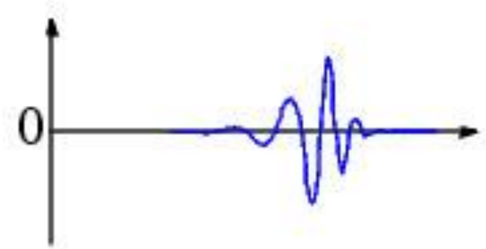
- Create scaled and shifted versions of $\psi(x)$

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right)$$

- Example:

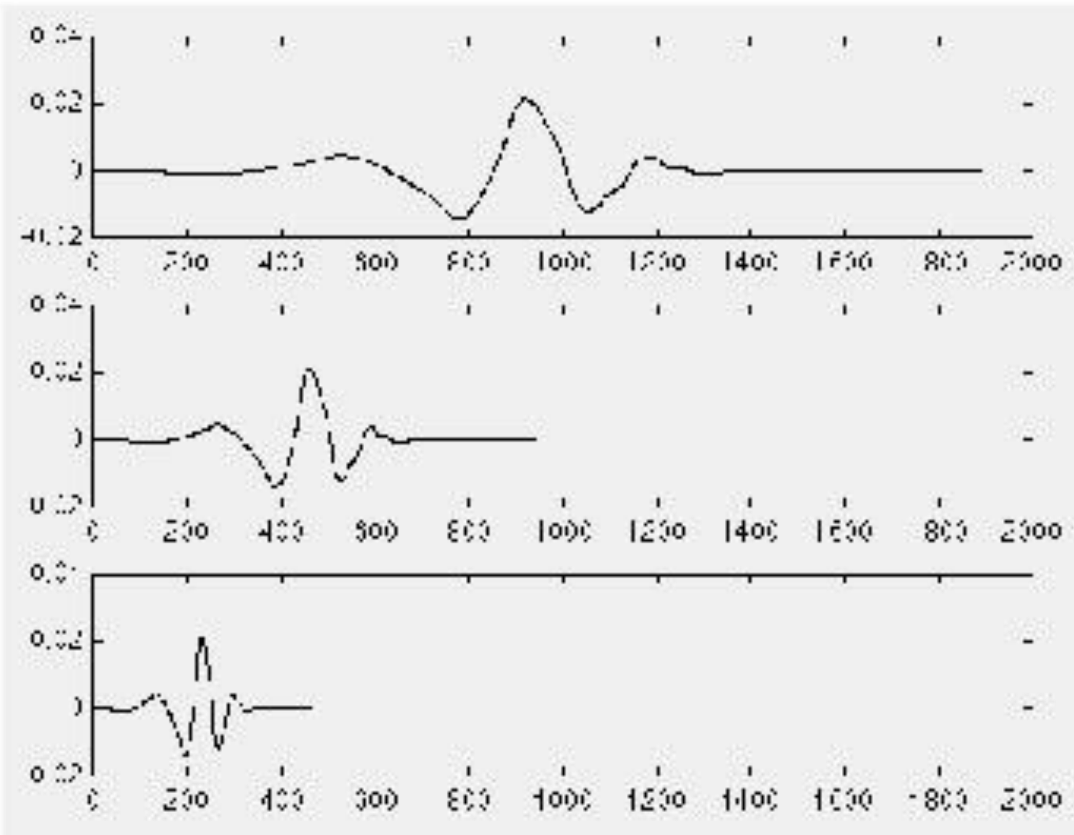


Wavelet function
$\psi(t)$



Shifted wavelet function
$\psi(t-k)$

Example of scaling



$$f(t) = \psi(t) \quad ; \quad a = 1$$

$$f(t) = \psi(2t) \; ; \quad a = \frac{1}{2}$$

$$f(t) = \psi(4t) \; ; \quad a = \frac{1}{4}$$

Continuous Wavelet Transform

- Define the continuous wavelet transform of *f(x)*:

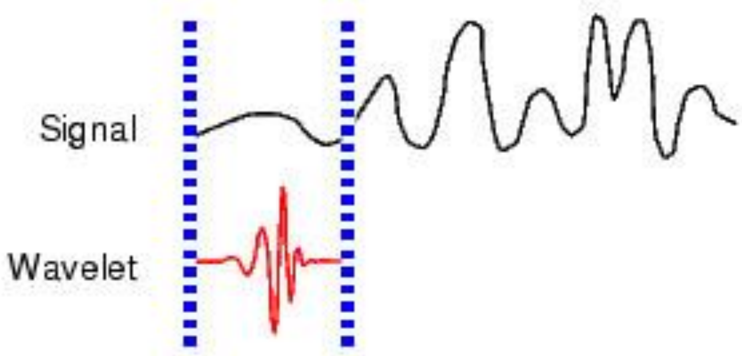
$$W_\varphi(s,\tau) = \int_{-\infty}^{\infty} f(x)\psi_{s,\tau}(x)\,dx$$

- This transforms a continuous function of one variable into a continuous function of two variables: translation and scale

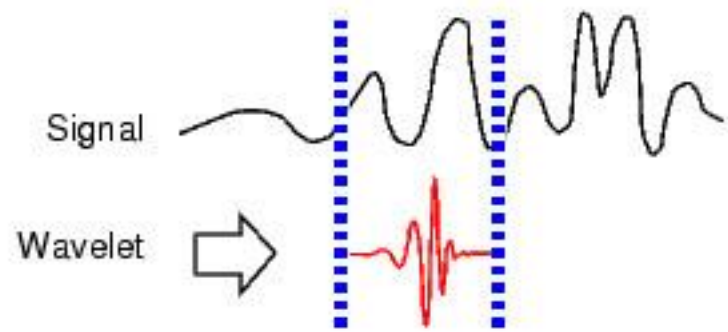
- The wavelet coefficients measure how closely correlated the wavelet is with each section of the signal

- For compact representation, choose a wavelet that matches the shape of the image components
  - Example: Haar wavelet for black and white drawings

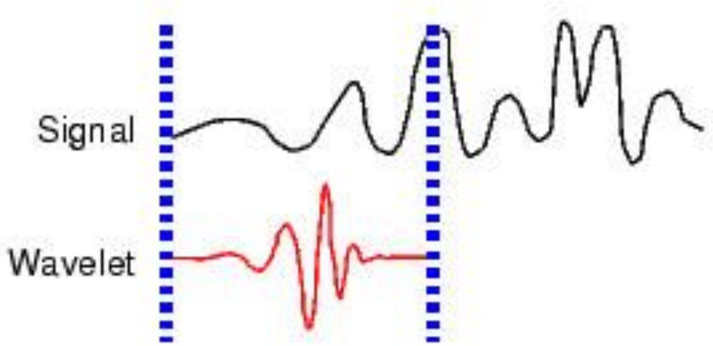
Example



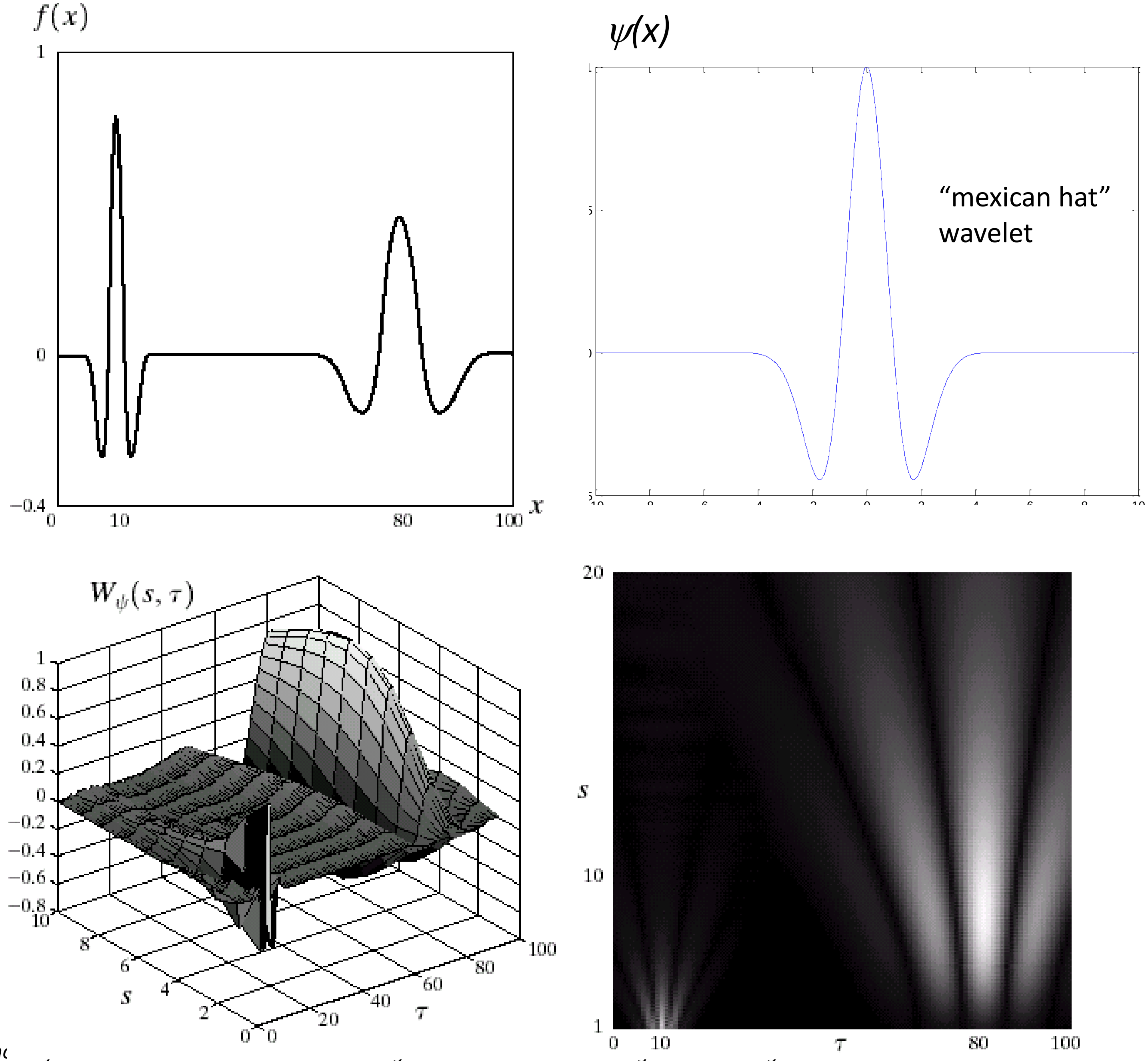
Low value for $W_{\psi}(s,\tau)$



Higher value of $W_{\psi}(s,\tau_2)$



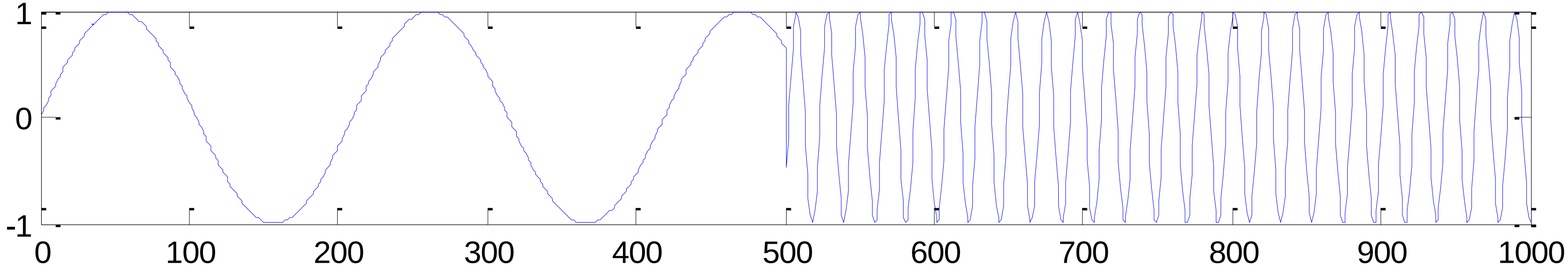
Different scale

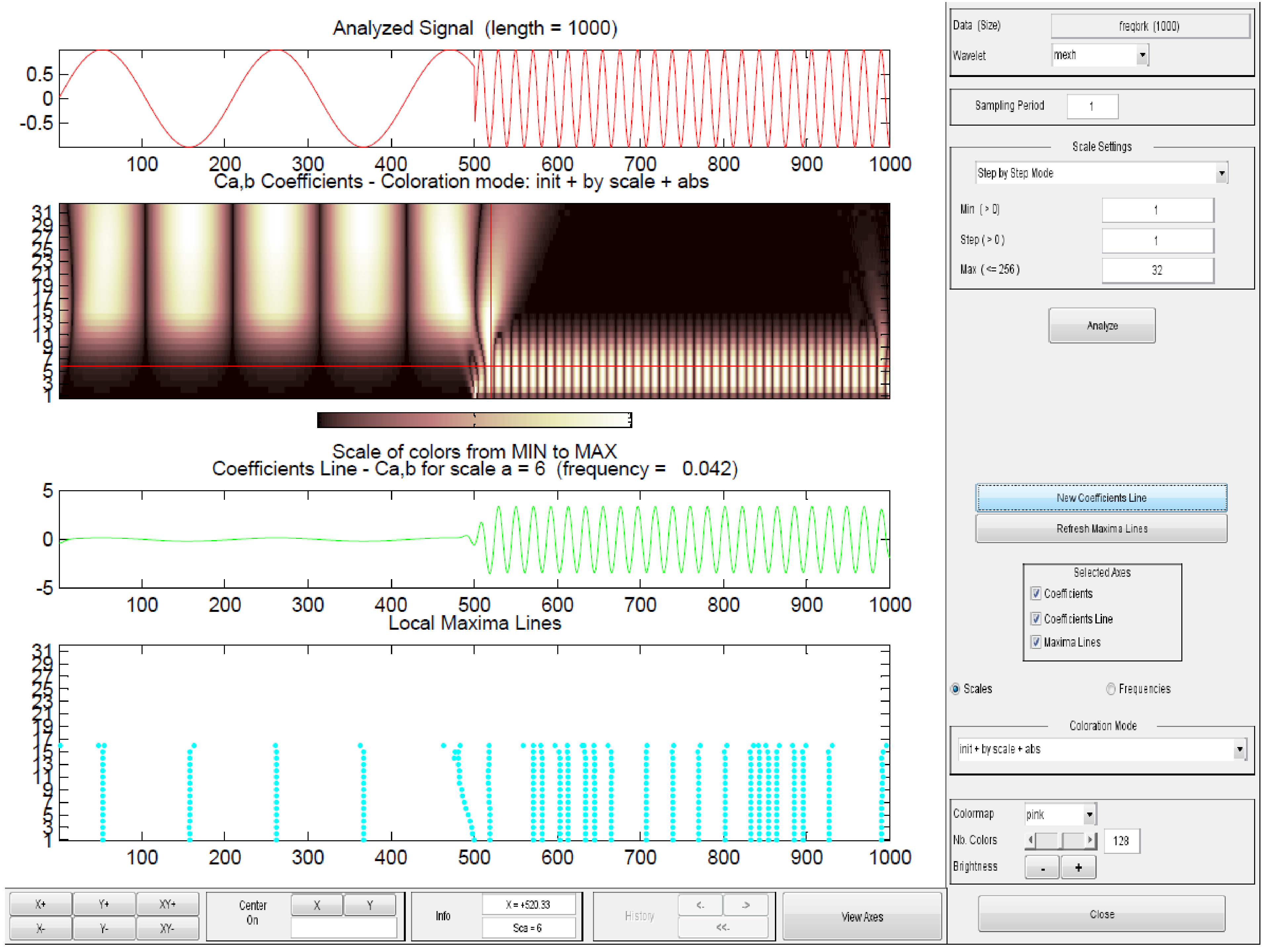


f(x)
1
0
−0.4
0
10
80
100
x
ψ(x)
“mexican hat”
wavelet
W_ψ(s, τ)
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
10
8
6
4
2
0
s
0
20
40
60
80
100
τ
20
s
10
1
0
10
τ
80
100

Matlab Demo

- Run “wavemenu”
  - Choose “Continuous wavelet 1D”
  - Choose “Example analysis” -> “frequency breakdown with mexh”
  - Look at magnitude of coefficients (right click on coefficients to select scale, then hit the button “new coefficients line")




Analyzed Signal (length = 1000)
Ca,b Coefficients - Coloration mode: init + by scale + abs
Scale of colors from MIN to MAX
Coefficients Line - Ca,b for scale a = 6 (frequency = 0.042)
Local Maxima Lines
Data (Size)
freqbrk (1000)
Wavelet
mexh
Sampling Period
1
Scale Settings
Step by Step Mode
Min (> 0)
1
Step (> 0)
1
Max (<= 256)
32
Analyze
New Coefficients Line
Refresh Maxima Lines
Selected Axes
Coefficients
Coefficients Line
Maxima Lines
Scales
Frequencies
Coloration Mode
init + by scale + abs
Colormap
pink
Nb. Colors
128
Brightness
Close
X+
Y+
XY+
X-
Y-
XY-
Center On
X
Y
Info
X = +520.33
Sca = 6
History
View Axes

Inverse Transform

- Inverse continuous wavelet transform

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_\psi(s,\tau) \frac{\psi_{s,\tau}(x)}{s^2} \, d\tau \, ds$$

- where

$$C_\psi = \int_{-\infty}^\infty \frac{\left|\Psi(\mu)\right|}{|\mu|} d\mu$$

- and $\Psi(\mu)$ is the Fourier transform of $\psi(x)$

Discrete Wavelet Transform

- Don't need to calculate wavelet coefficients at every possible scale
- Can choose scales based on powers of two, and get equivalent accuracy

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$$

- We can represent a discrete function *f(n)* as a weighted summation of wavelets ψ*(n),* plus a coarse approximation φ*(n)*

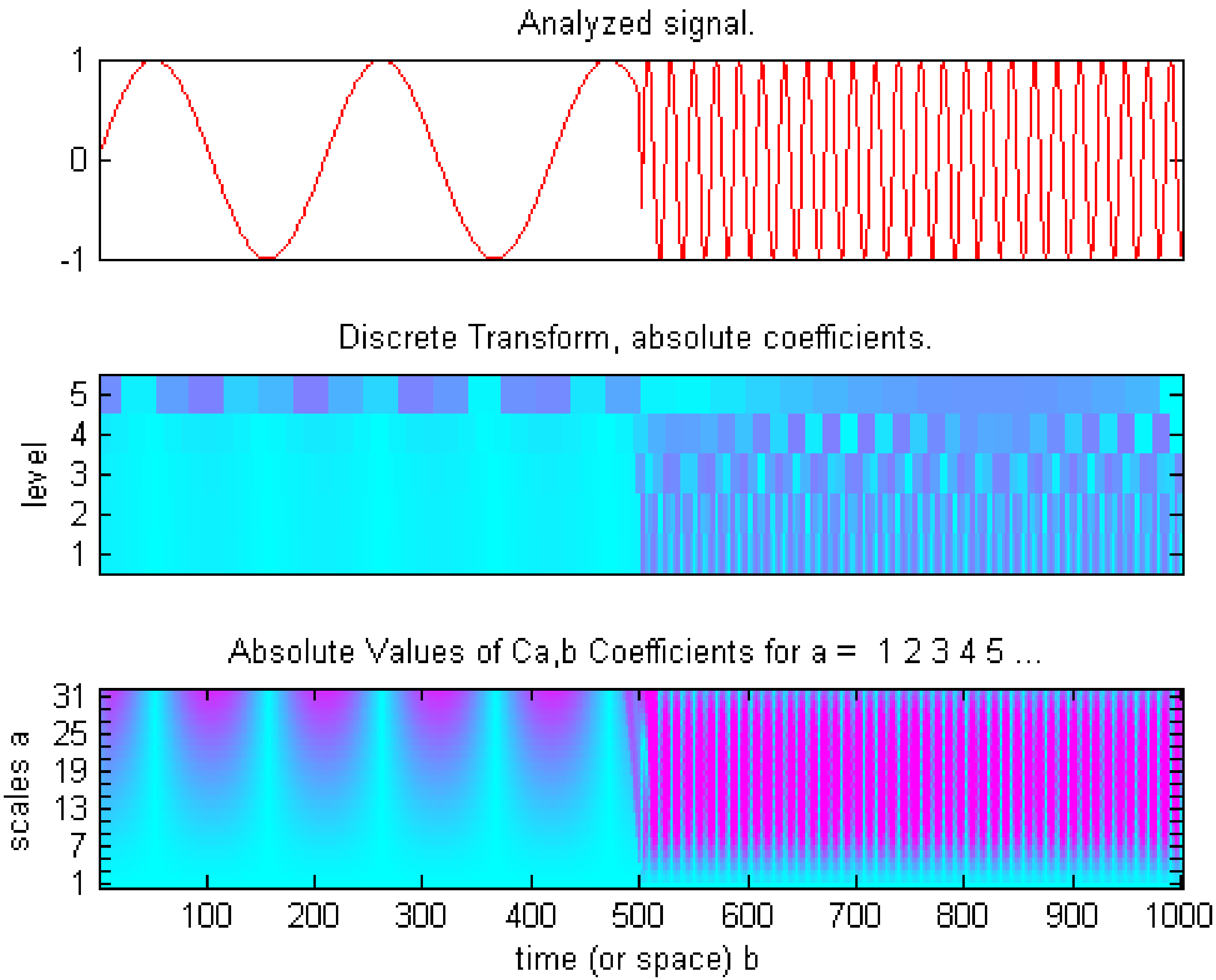
$$f(n) = \frac{1}{\sqrt{M}}\sum_k W_\varphi(j_0,k)\varphi_{j_0,k}(n) \; + \; \frac{1}{\sqrt{M}}\sum_{j=j_0}^{\infty}\sum_k W_\psi(j,k)\psi_{j,k}(n)$$

where j_0 is an arbitrary starting scale, and *n = 0,1,2, … M*

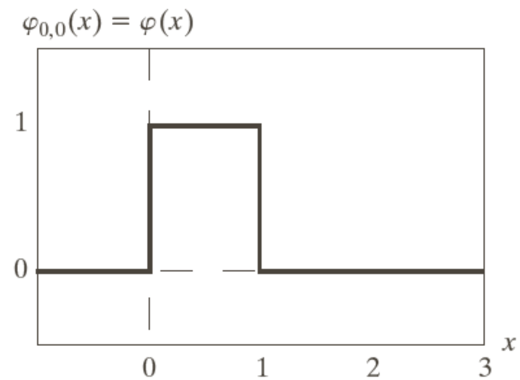
"Approximation" coefficients	"Detail" coefficients
$W_\Phi(j_0,k) = \frac{1}{\sqrt{M}}\sum_x f(x)\,\varphi_{j_0,k}(x)$	$W_\Psi(j,k) = \frac{1}{\sqrt{M}}\sum_x f(x)\,\psi_{j,k}(x)$

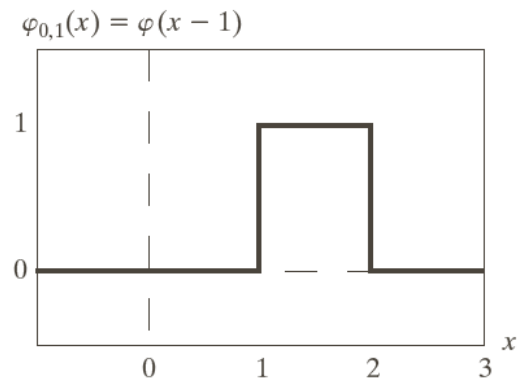
Comparison with CWT

- Usually you don't need to compute the continuous transform
- A signal (with finite energy) can be reconstructed from the discrete transform

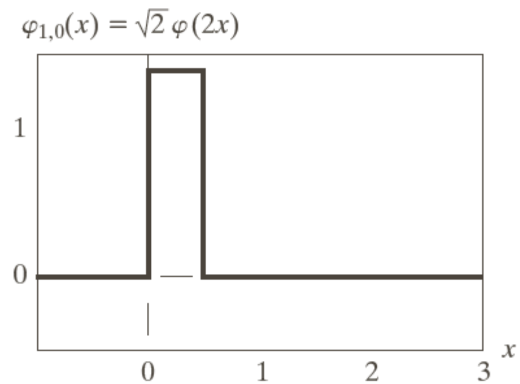


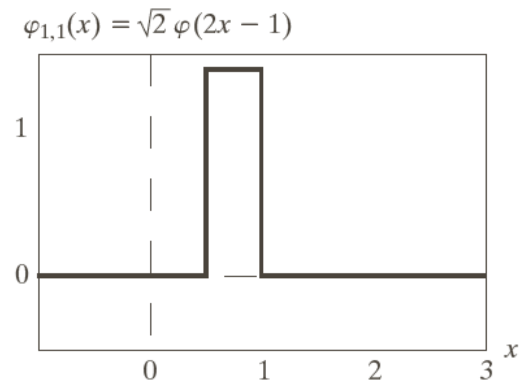
From Matlab help page on wavelets

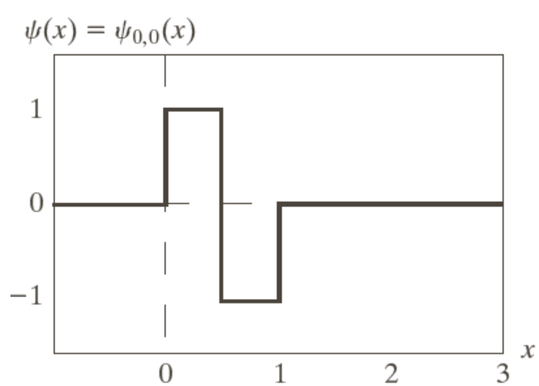


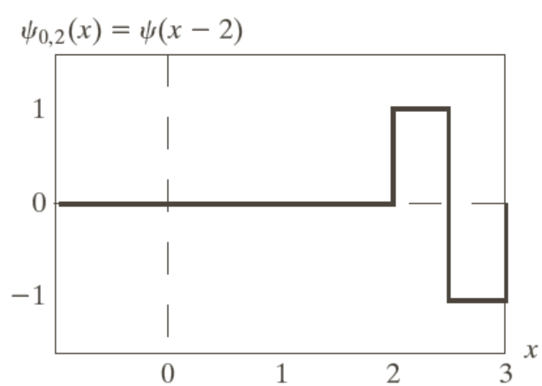


Harr scaling functions









Harr wavelet functions

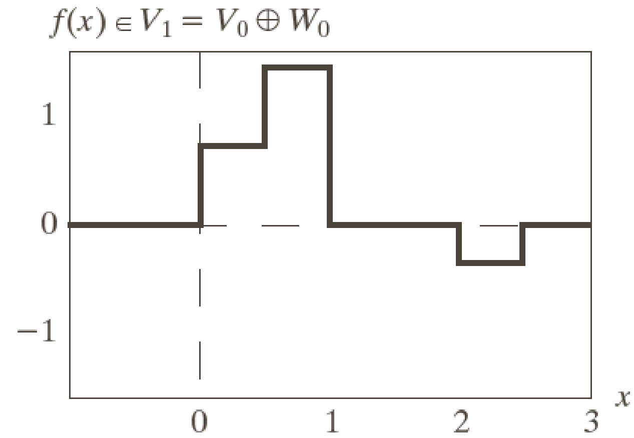
Example

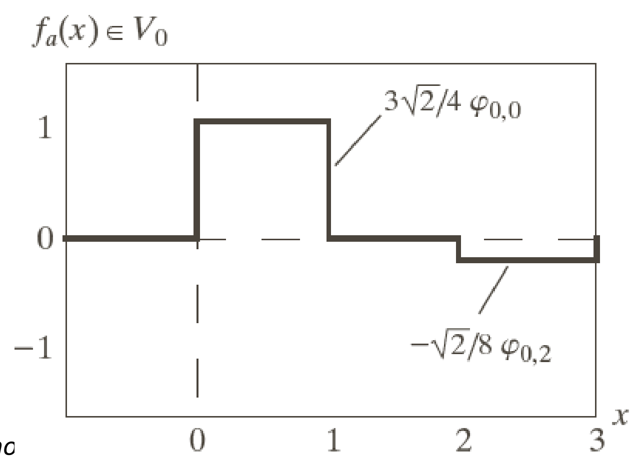
- A function can be represented by a sum of approximation plus detail

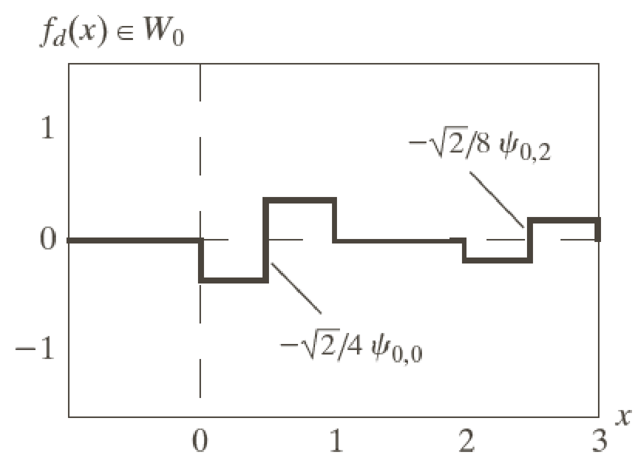
$$f(x) = f_a(x) + f_d(x)$$

$$f_a(x) = \frac{3\sqrt{2}}{4}\varphi_{0,0}(x) - \frac{\sqrt{2}}{8}\varphi_{0,2}(x)$$

$$f_d(x) = \frac{-\sqrt{2}}{4}\psi_{0,0}(x) - \frac{\sqrt{2}}{8}\psi_{0,2}(x)$$

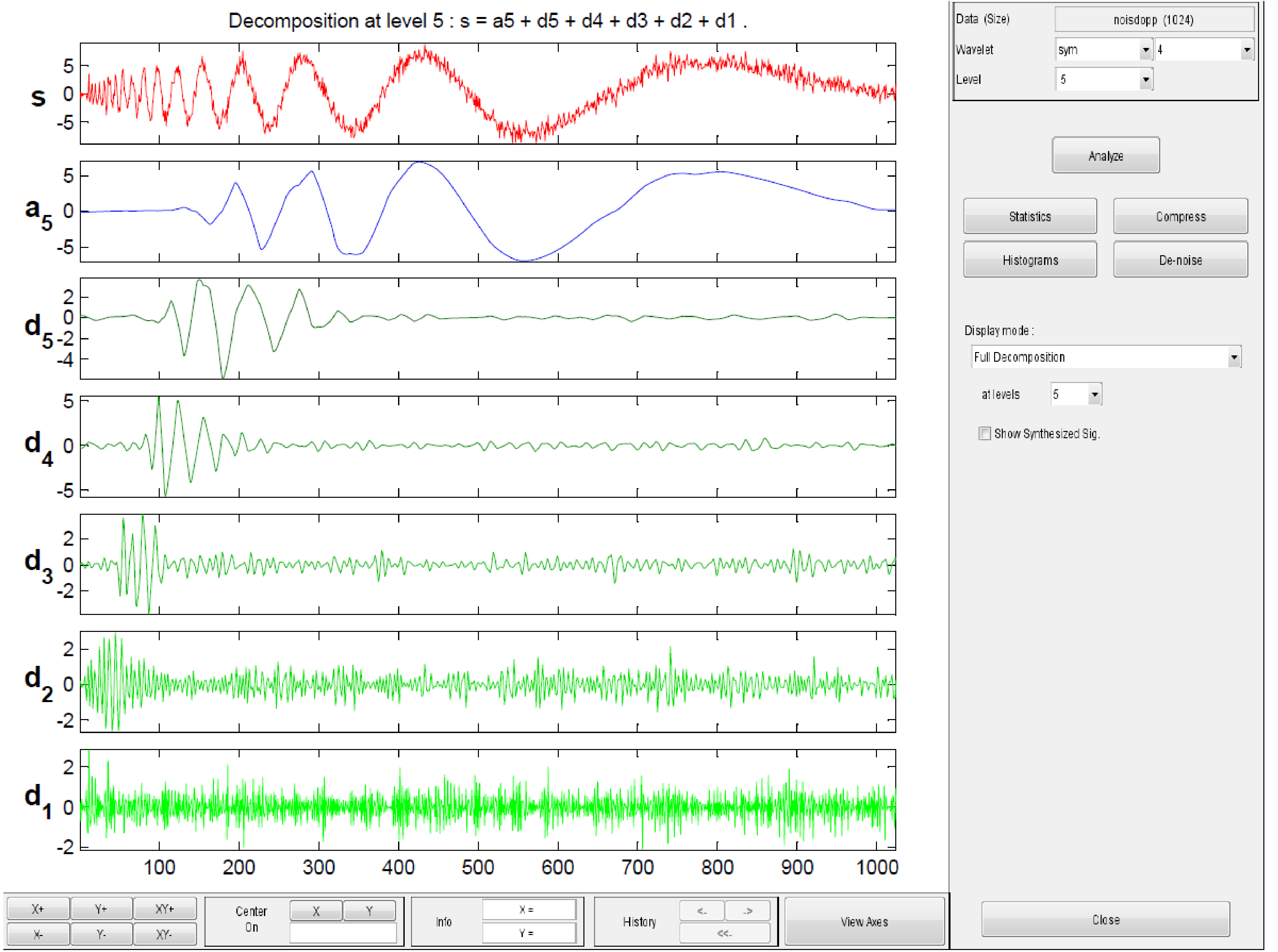


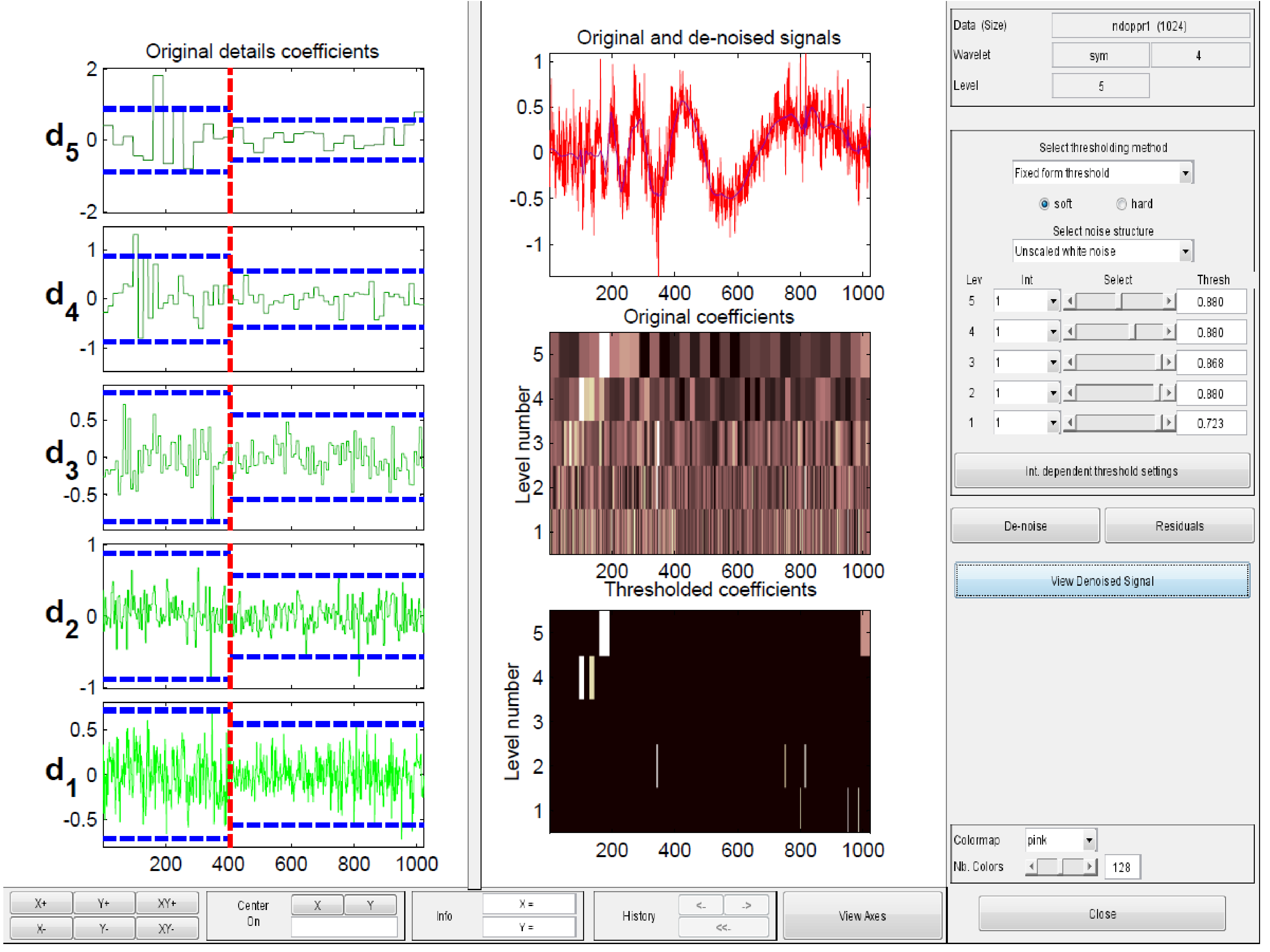




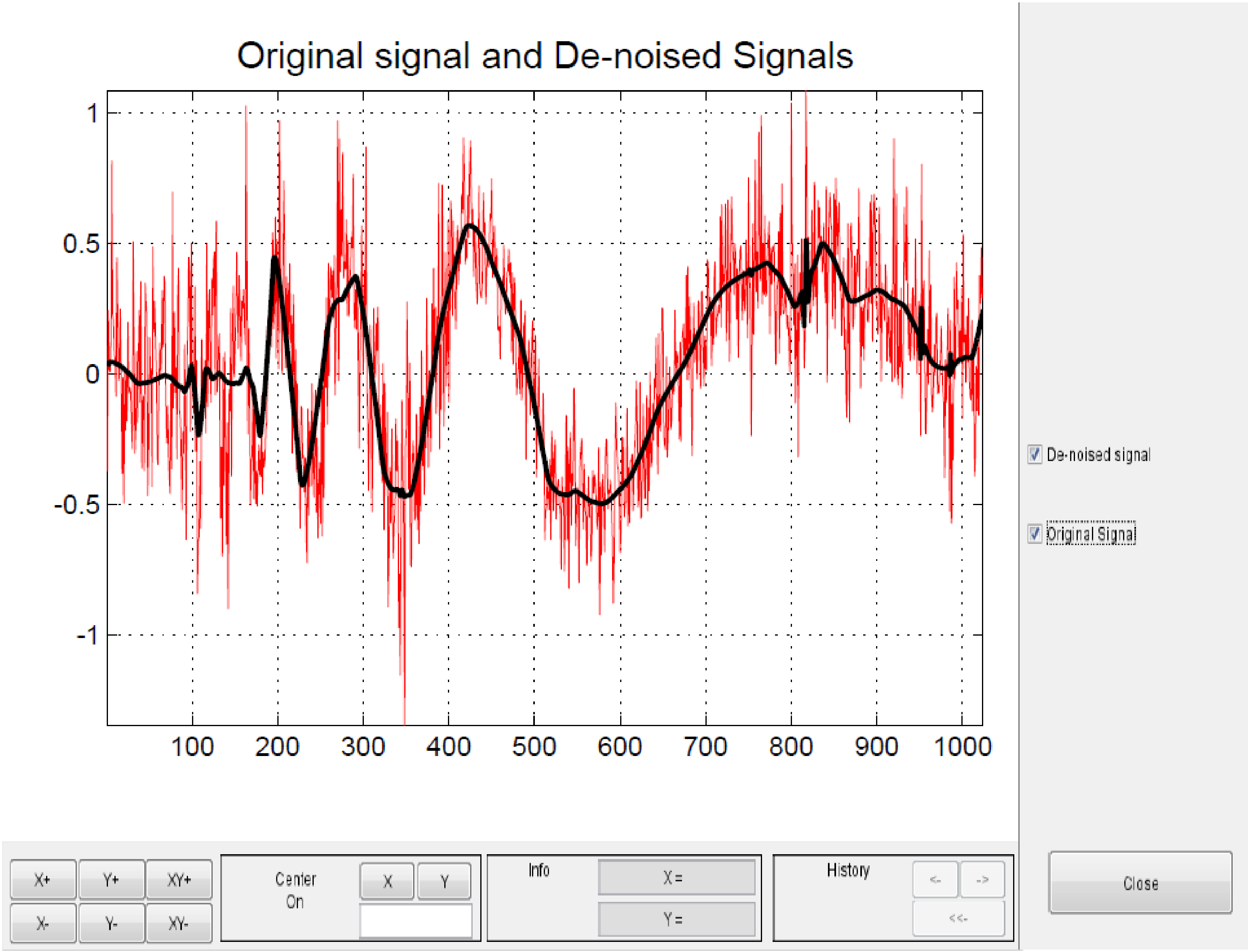
Matlab Demos

- “wavemenu”
- Do 1D discrete wavelet transform on noisy doppler signal, show denoising


Decomposition at level 5 : s = a5 + d5 + d4 + d3 + d2 + d1 .
s
a5
d5
d4
d3
d2
d1
100 200 300 400 500 600 700 800 900 1000
Data (Size)
noisdopp (1024)
Wavelet
sym
4
Level
5
Analyze
Statistics
Compress
Histograms
De-noise
Display mode :
Full Decomposition
at levels
5
Show Synthesized Sig.
X+ Y+ XY+
X- Y- XY-
Center On
X Y
Info
X =
Y =
History
<- ->
<<-
View Axes
Close

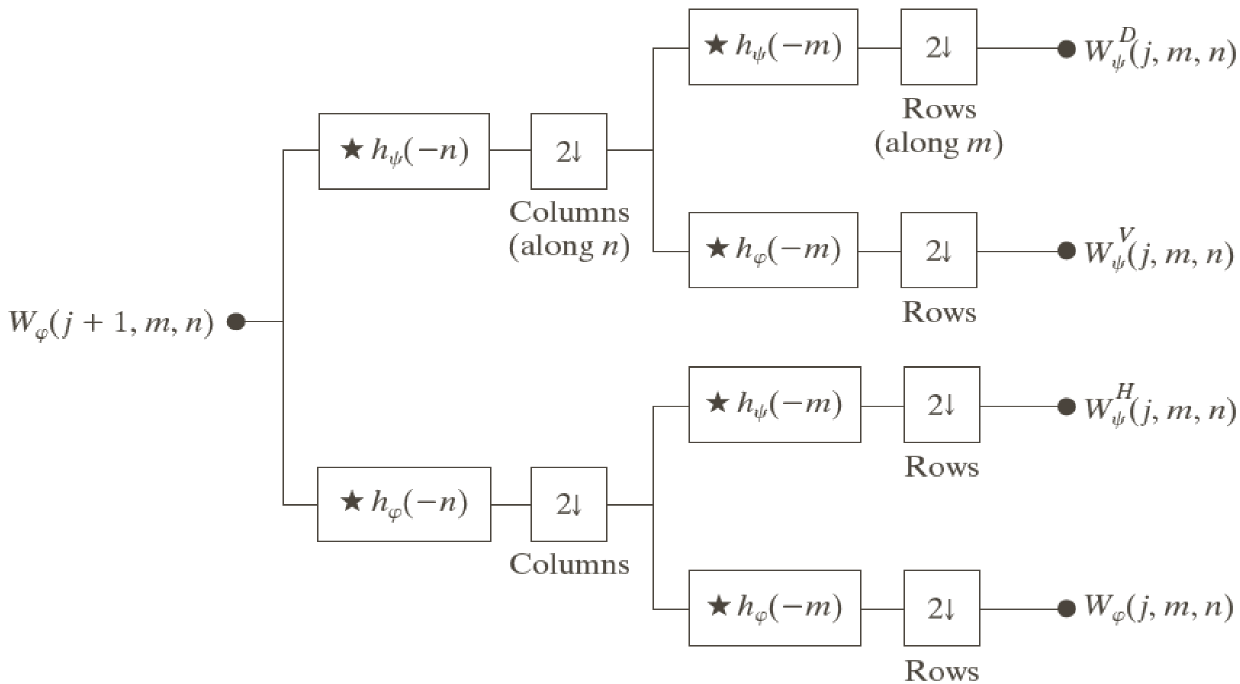


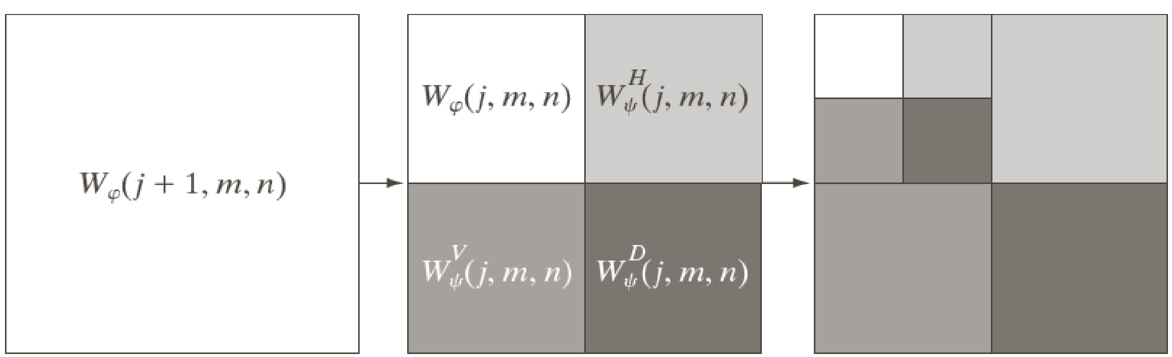
Original details coefficients
d5
d4
d3
d2
d1
Original and de-noised signals
Original coefficients
Level number
Thresholded coefficients
Level number
Data (Size)
ndoppr1 (1024)
Wavelet
sym
4
Level
5
Select thresholding method
Fixed form threshold
soft
hard
Select noise structure
Unscaled white noise
Lev
Int
Select
Thresh
0.880
0.880
0.868
0.880
0.723
Int. dependent threshold settings
De-noise
Residuals
View Denoised Signal
Colormap
pink
Nb. Colors
128
Close
X+
Y+
XY+
X-
Y-
XY-
Center On
X
Y
Info
X =
Y =
History
View Axes



Original signal and De-noised Signals
1
0.5
0
-0.5
-1
100
200
300
400
500
600
700
800
900
1000
De-noised signal
Original Signal
X+
Y+
XY+
X-
Y-
XY-
Center
On
X
Y
Info
X =
Y =
History
<-
->
<<-
Close

Expanding to Two Dimensions





Colorado

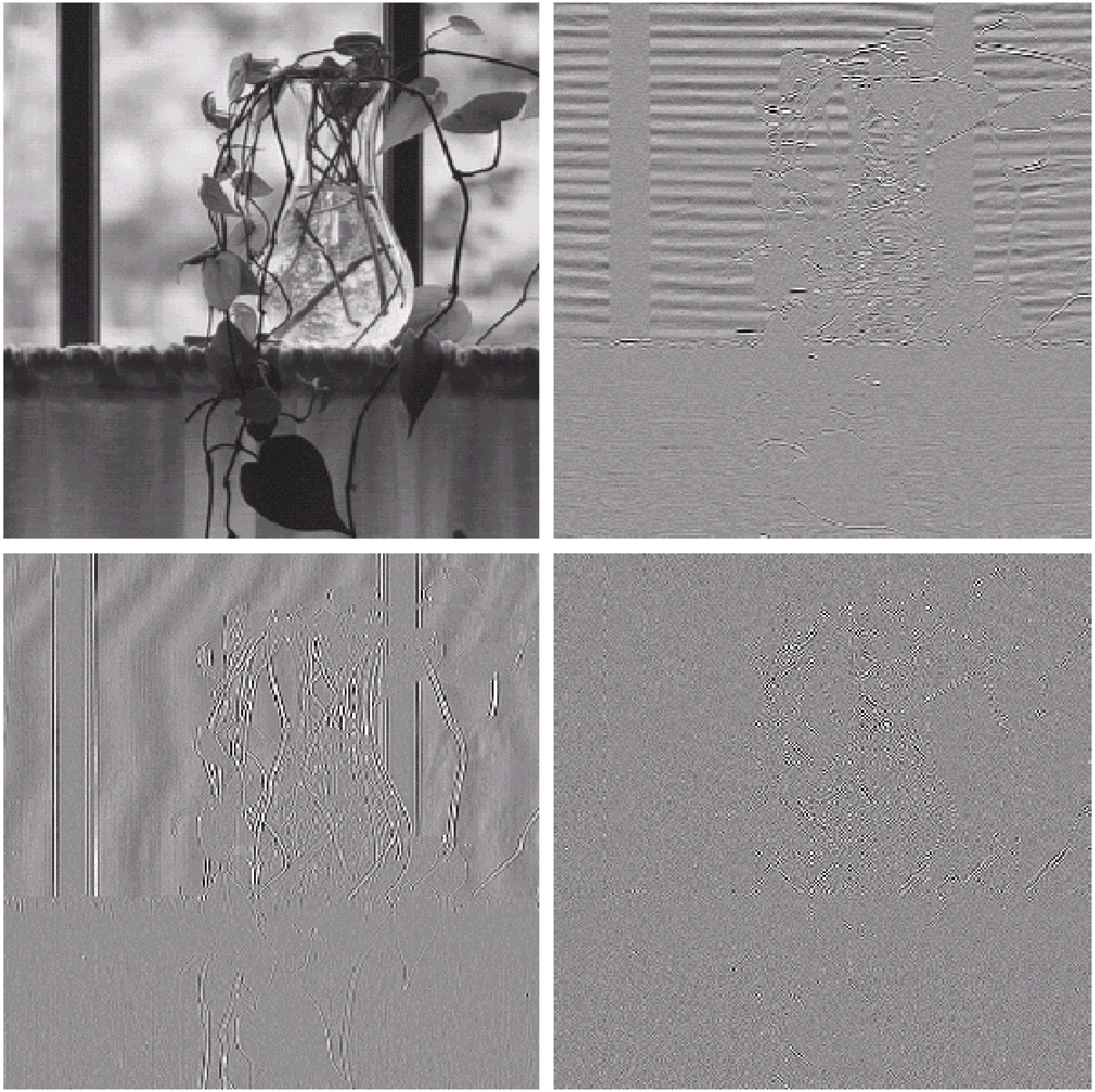


FIGURE 7.7 A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.5.

a	d^V
d^H	d^D

a(m,n): approximation

d^V(m,n): detail in vertical

d^H(m,n): detail in horizontal

d^D(m,n): detail in diagonal

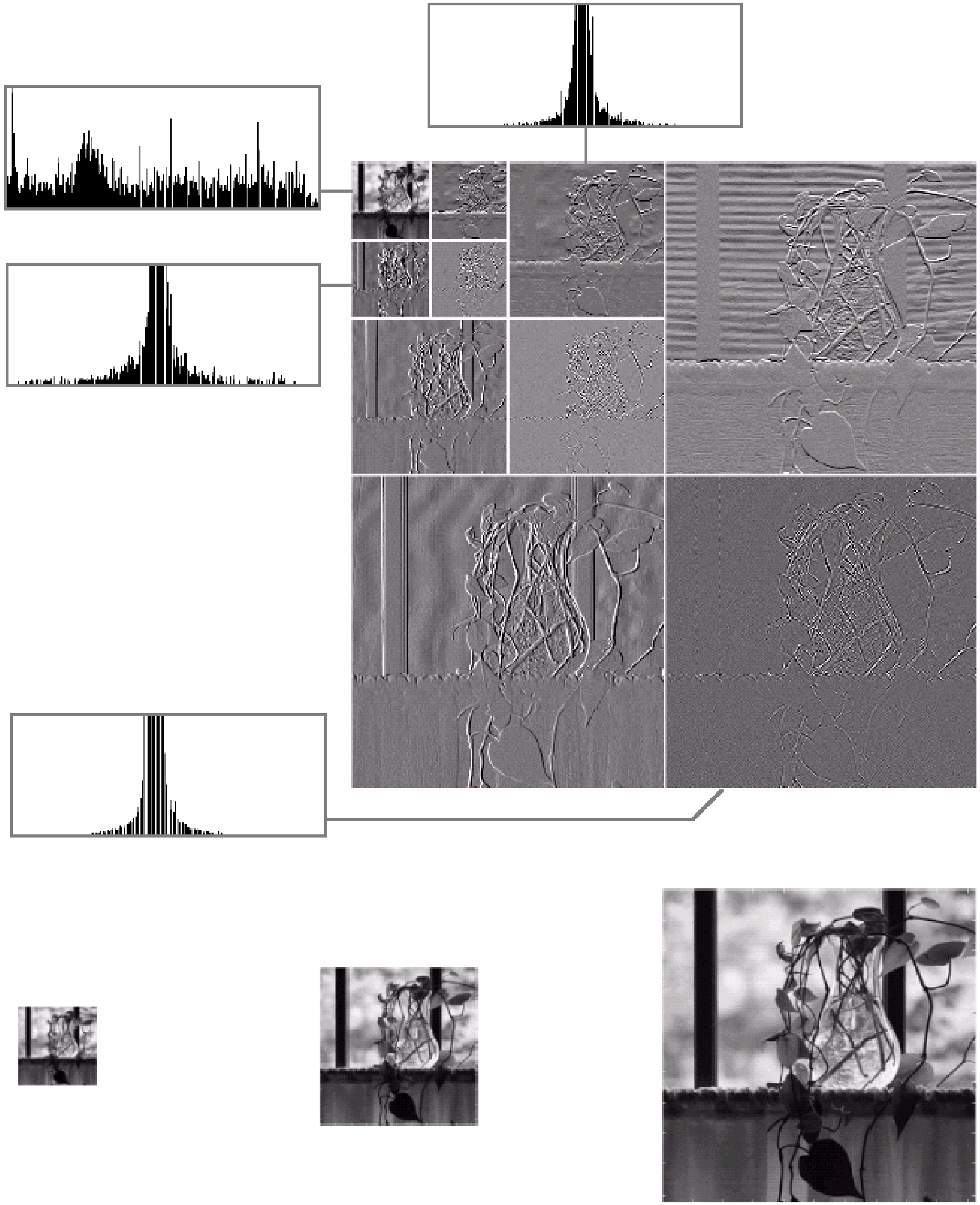


FIGURE 7.8 (a) A discrete wavelet transform using Haar basis functions. Its local histogram variations are also shown; (b)–(d) Several different approximations (64 × 64, 128 × 128, and 256 × 256) that can be obtained from (a).

Colorado

Use of Wavelets in Processing

- Approach:
  - Compute the 2D wavelet transform
  - Alter the transform
  - Compute the inverse transform
- Examples:
  - De-noising
  - Compression
  - Image fusion

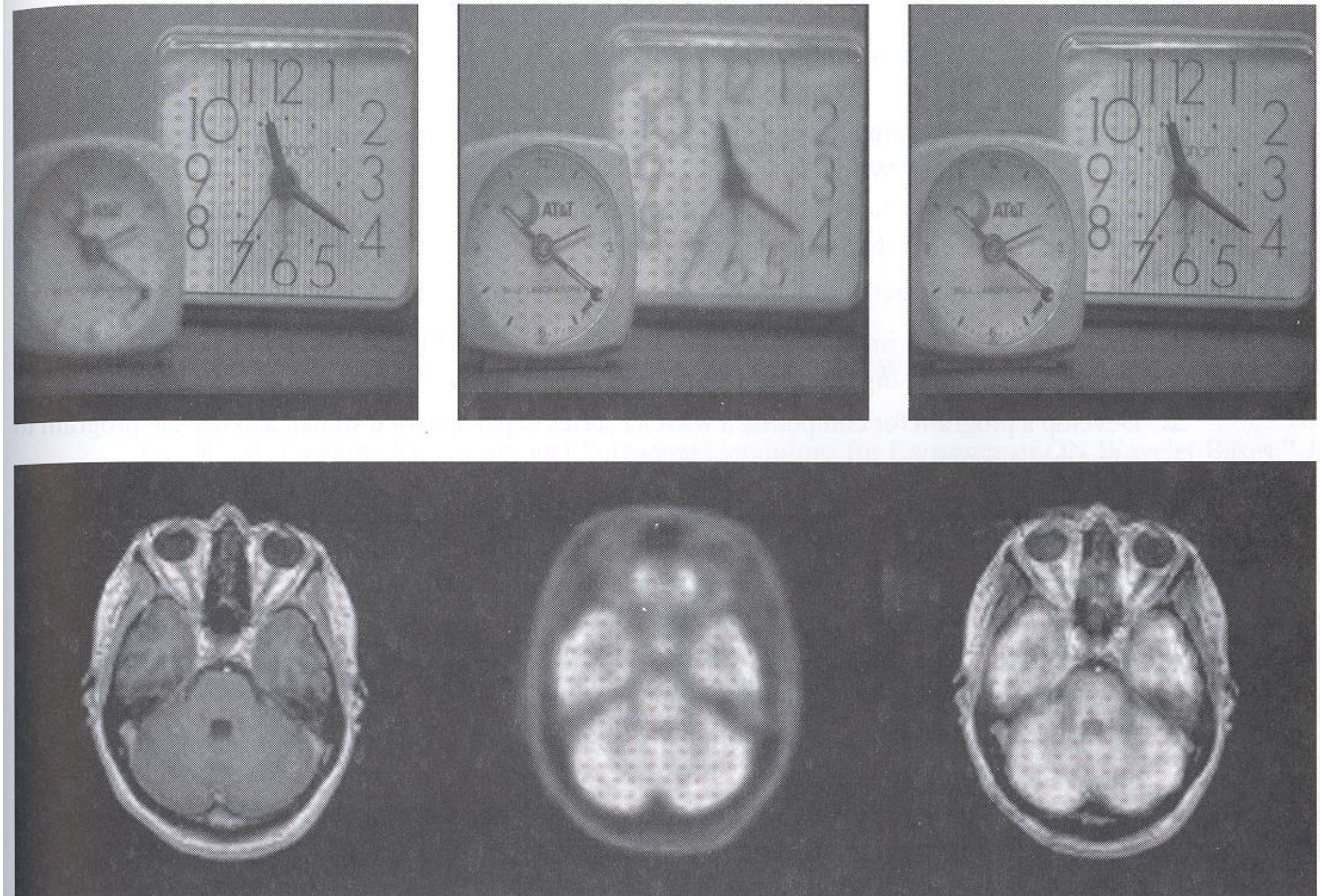
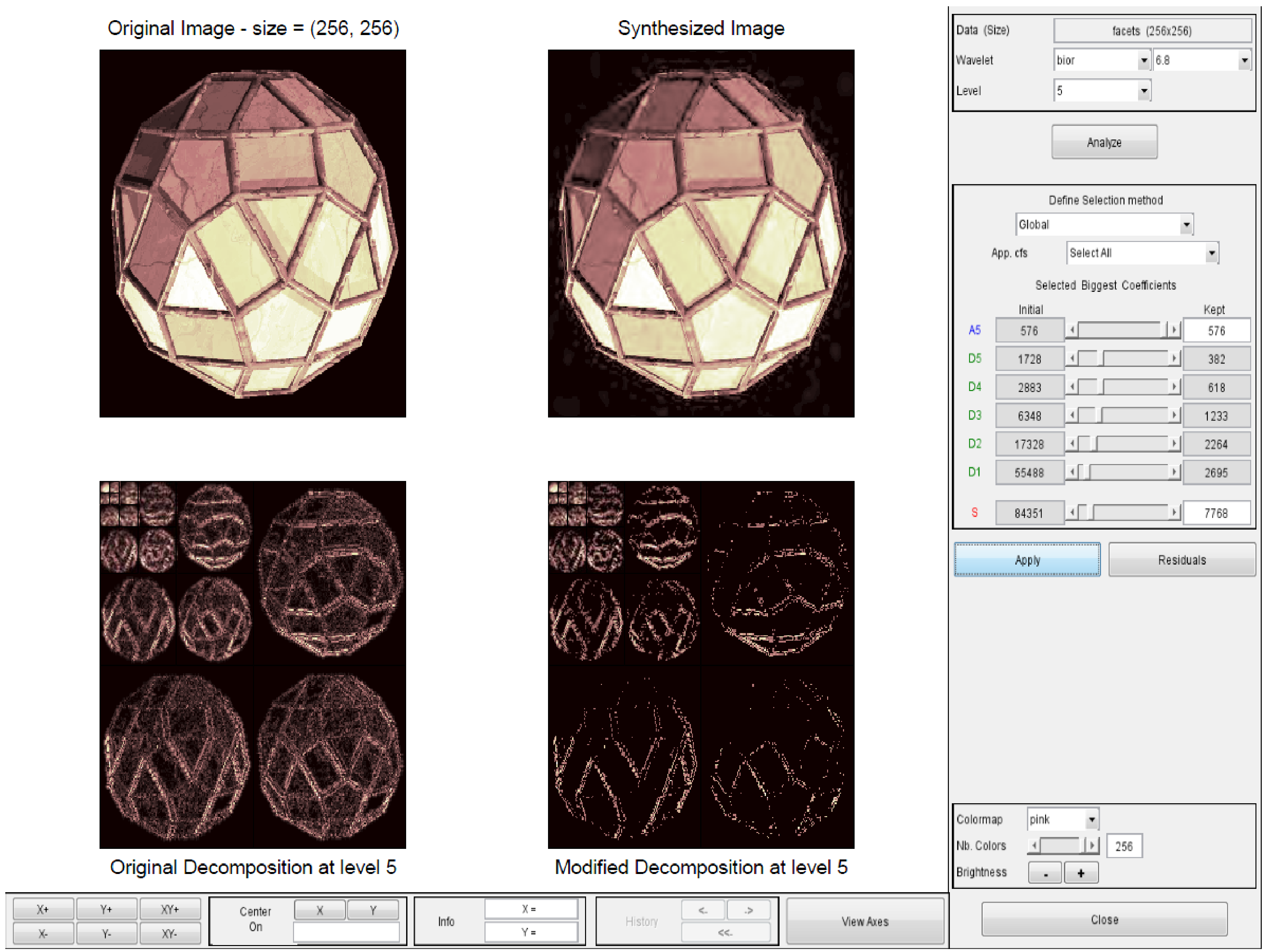


Figure 14–36 Wavelet transform image fusion: (a), (b) images taken at different focus settings; (c) fused image; (d) MRI image; (e) PET image; (f) fused image (Courtesy Henry Hui Li, reprinted by permission from [28])

Matlab Examples (“wavemenu”)

- De-noising
  - Choose “SWT de-noising 2D”
  - Set threshold value to zero out coefficients below the threshold
- Compression
  - Choose “Wavelet coefficients selection 2D”
- Fusion
  - Choose “Image fusion”



Original Image - size = (256, 256)
Synthesized Image
Original Decomposition at level 5
Modified Decomposition at level 5
Data (Size)
facets (256x256)
Wavelet
bior
6.8
Level
5
Analyze
Define Selection method
Global
App. cfs
Select All
Selected Biggest Coefficients
Initial
Kept
A5 576 576
D5 1728 382
D4 2883 618
D3 6348 1233
D2 17328 2264
D1 55488 2695
S 84351 7768
Apply
Residuals
Colormap
pink
Nb. Colors
256
Brightness
-
+
Close
X+ Y+ XY+
X- Y- XY-
Center On
X
Y
Info
X =
Y =
History
<-
->
<<-
View Axes

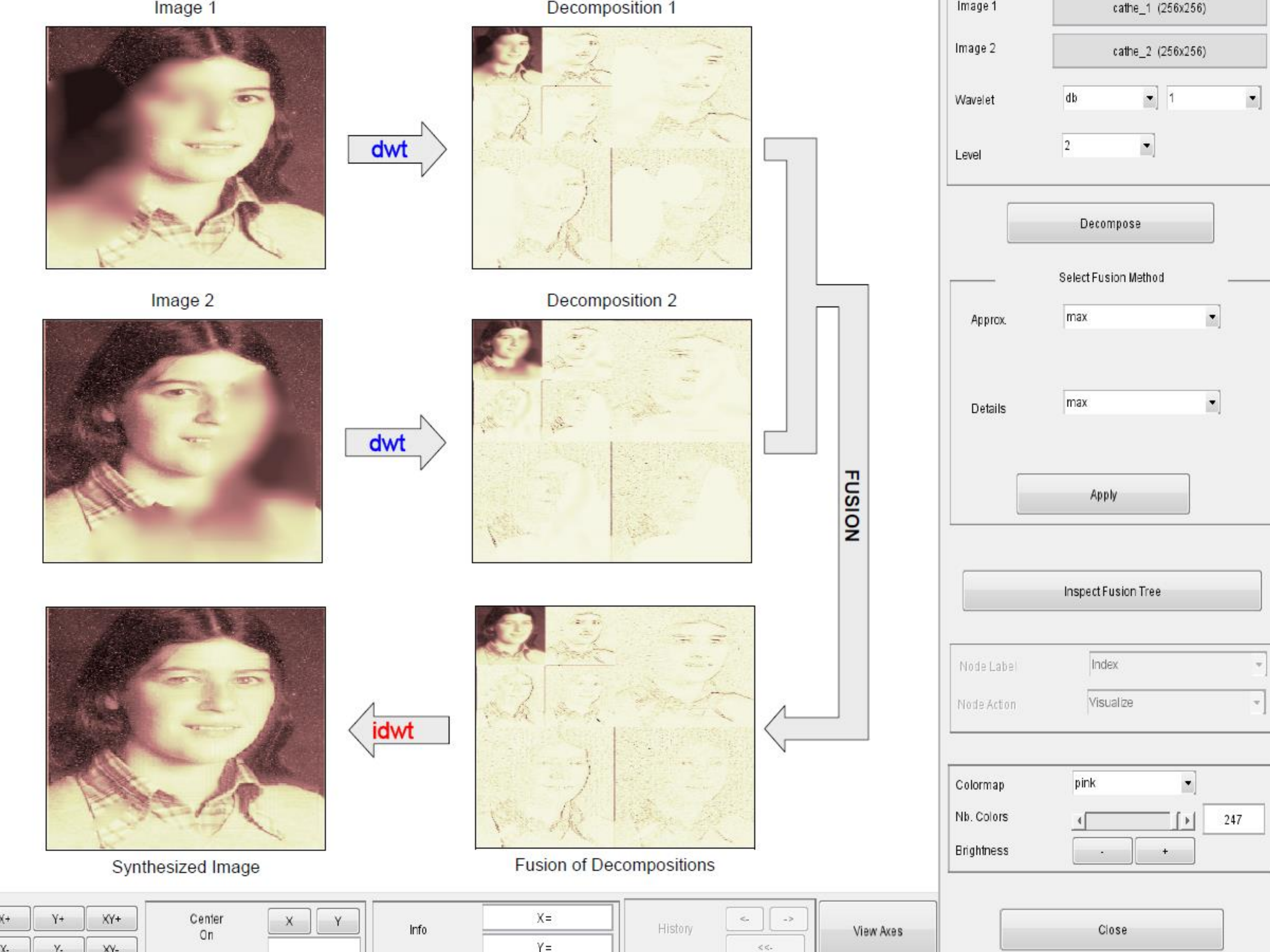


Image 1
Decomposition 1
dwt
Image 2
Decomposition 2
dwt
FUSION
idwt
Synthesized Image
Fusion of Decompositions
Image 1
cathe_1 (256x256)
Image 2
cathe_2 (256x256)
Wavelet
db
1
Level
2
Decompose
Select Fusion Method
Approx.
max
Details
max
Apply
Inspect Fusion Tree
Node Label
Index
Node Action
Visualize
Colormap
pink
Nb. Colors
247
Brightness
-
+
Close
X+
Y+
XY+
X-
Y-
XY-
Center On
X
Y
Info
X=
Y=
History
<-
->
<<-
View Axes

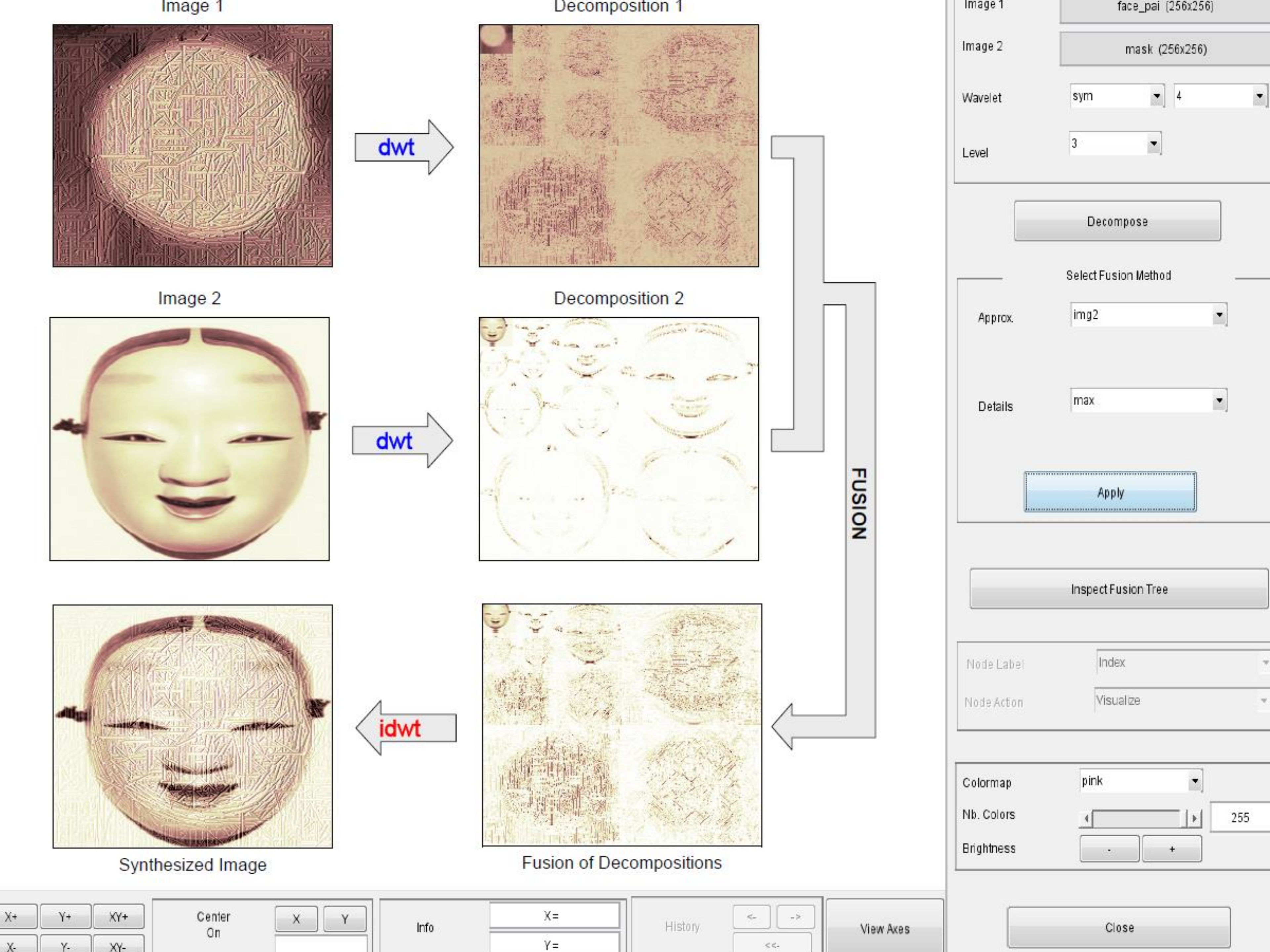


Image 1
Decomposition 1
dwt
Image 2
Decomposition 2
dwt
FUSION
idwt
Synthesized Image
Fusion of Decompositions
Image 1
face_pai (256x256)
Image 2
mask (256x256)
Wavelet
sym
4
Level
3
Decompose
Select Fusion Method
Approx.
img2
Details
max
Apply
Inspect Fusion Tree
Node Label
Index
Node Action
Visualize
Colormap
pink
Nb. Colors
255
Brightness
-
+
Close
X+
Y+
XY+
X-
Y-
XY-
Center On
X
Y
Info
X=
Y=
History
<-
->
<<-
View Axes

Summary / Questions

- Wavelets represent the scale of features in an image, as well as their position.
  - Can also be applied to 1D signals.
- They are useful for a number of applications including image compression.
- We can use them to process images:
  - Compute the 2D wavelet transform
  - Alter the transform
  - Compute the inverse transform
- What are some other applications of wavelet processing?