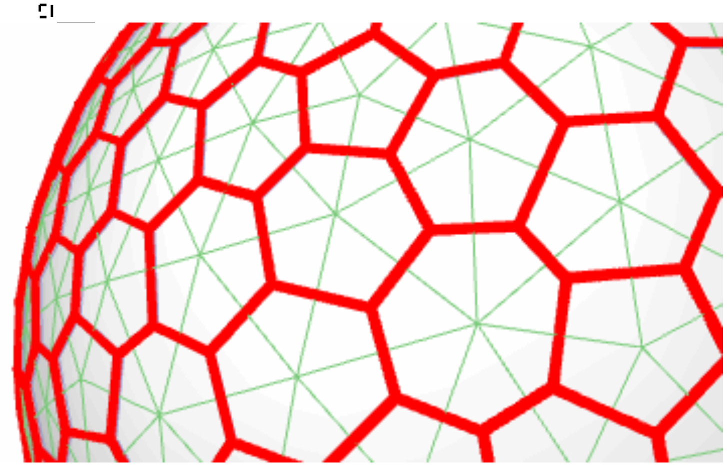
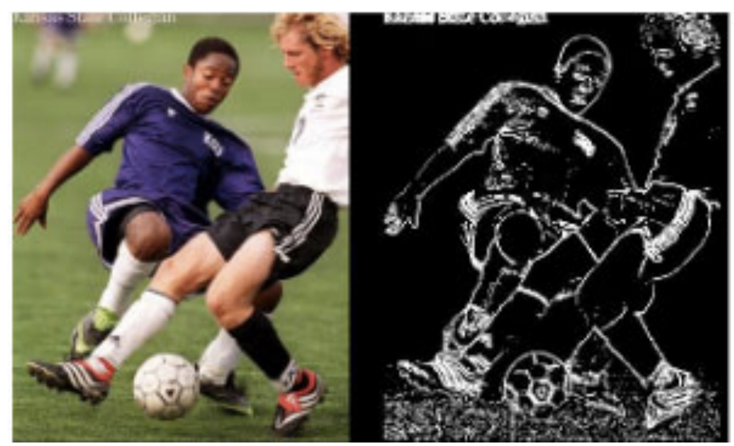


DIGITAL IMAGE PROCESSING

Lecture 6

Wavelets (cont), Lines and edges

Tammy Riklin Raviv
Electrical and Computer Engineering
Ben-Gurion University of the Negev

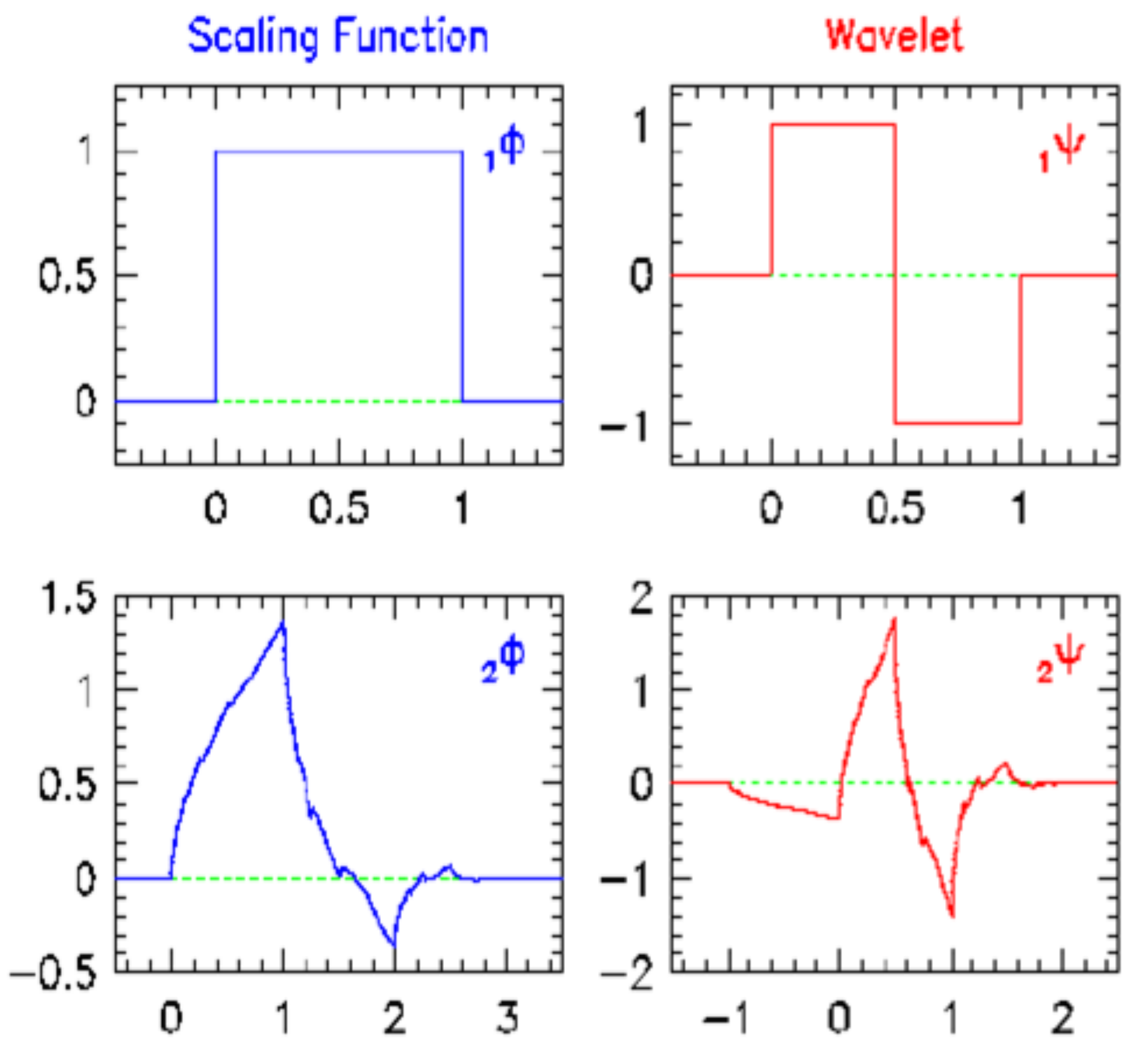


before we move on

- Wavelets … a few more slides

Wavelet functions examples

- Haar function
- Daubechies function



Haar Wavelets

$$D_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \Rightarrow D_1^{-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} a+b \\ a-b \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix} \Rightarrow \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} c \\ d \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} c+d \\ c-d \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Haar Wavelets

$$c[m/2] = \frac{1}{\sqrt{2}}[v[m] + v[m+1]] \quad m = 0,2,4,\ \ldots.\ ,M-1$$

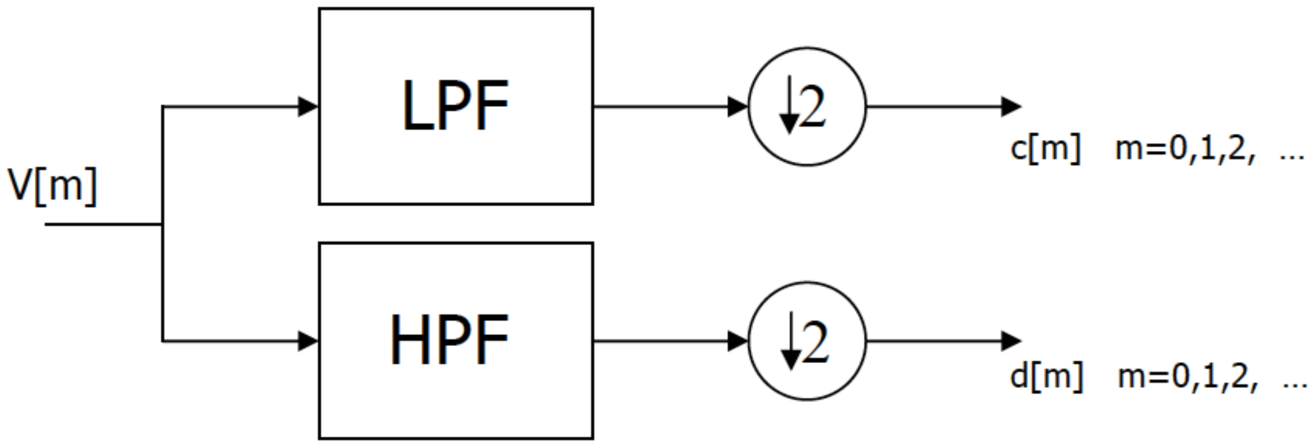
$$d[m/2] = \frac{1}{\sqrt{2}}[v[m] - v[m+1]] \quad m = 0,2,4,\ \ldots.\ ,M-1$$

Given a sequence of M items, partition into pairs. Replace each pair by the sum and difference of the pair of items.

Haar Wavelets

$$c[{}^{m}\!/_{2}] = \frac{1}{\sqrt{2}}[v[m] + v[m+1]] \quad m = 0,2,4, \ldots. ,M-1$$

$$d[{}^{m}\!/_{2}] = \frac{1}{\sqrt{2}}[v[m] - v[m+1]] \quad m = 0,2,4, \ldots. ,M-1$$



Haar Wavelets

Very simple numerical example:

$\underline{v}$=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

$$c = \frac{1}{\sqrt{2}}\left[(1+2)\quad(3+4)\quad(5+6)\quad(7+8)\quad(9+10)\quad(11+12)\quad(13+14)\quad(15+16)\right] =$$

$$= \frac{1}{\sqrt{2}}\left[3\quad 7\quad 11\quad 15\quad 19\quad 23\quad 27\quad 31\right]$$

$$d = \frac{1}{\sqrt{2}}\left[(1-2)\quad(3-4)\quad(5-6)\quad(7-8)\quad(9-10)\quad(11-12)\quad(13-14)\quad(15-16)\right] =$$

$$= \frac{1}{\sqrt{2}}\left[-1\quad -1\quad -1\quad -1\quad -1\quad -1\quad -1\quad -1\right]$$

Haar Wavelets

Very simple numerical example:

$\underline{v}$=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

$$c = \frac{1}{\sqrt{2}}[0 \quad 3 \quad 0 \quad 7 \quad 0 \quad 11 \quad 0 \quad 15 \quad 0 \quad 19 \quad 0 \quad 23 \quad 0 \quad 27 \quad 0 \quad 31 \quad 0]$$

$$d = \frac{1}{\sqrt{2}}[0 \quad -1 \quad 0 \quad -1 \quad 0 \quad -1 \quad 0 \quad -1 \quad 0 \quad -1 \quad 0 \quad -1 \quad 0 \quad -1 \quad 0 \quad -1 \quad 0]$$

Haar Wavelets

Very simple numerical example:

$\underline{v}$=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

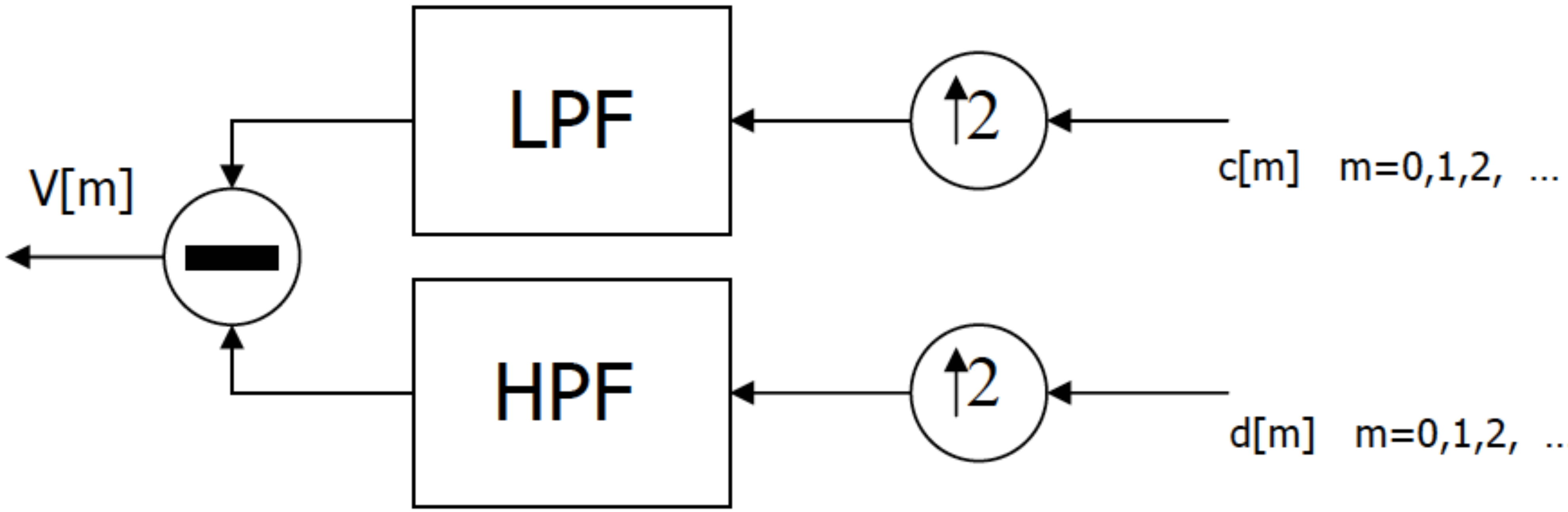
$$LPF\{c\} = \frac{1}{2}\left[(0+3)(3+0)(0+7)(7+0)\ \ \ldots\ldots\ \ (0+31)(31+0)\right] =$$

$$= \frac{1}{2}\left[3 \quad 3 \quad 7 \quad 7 \quad 11 \quad 11 \quad 15 \quad 15 \quad 19 \quad 19 \quad 23 \quad 23 \quad 27 \quad 27 \quad 31 \quad 31\right]$$

$$HPF\{d\} = \frac{1}{2}\left[(0+1)(-1-0)(0+1)(-1-0)(0+1)\ \ \ldots\ldots\right] =$$

$$= \frac{1}{2}\left[+1 \quad -1 \quad +1 \quad -1 \quad +1 \quad -1 \quad +1 \quad \ldots\ldots\right]$$

Haar Wavelets

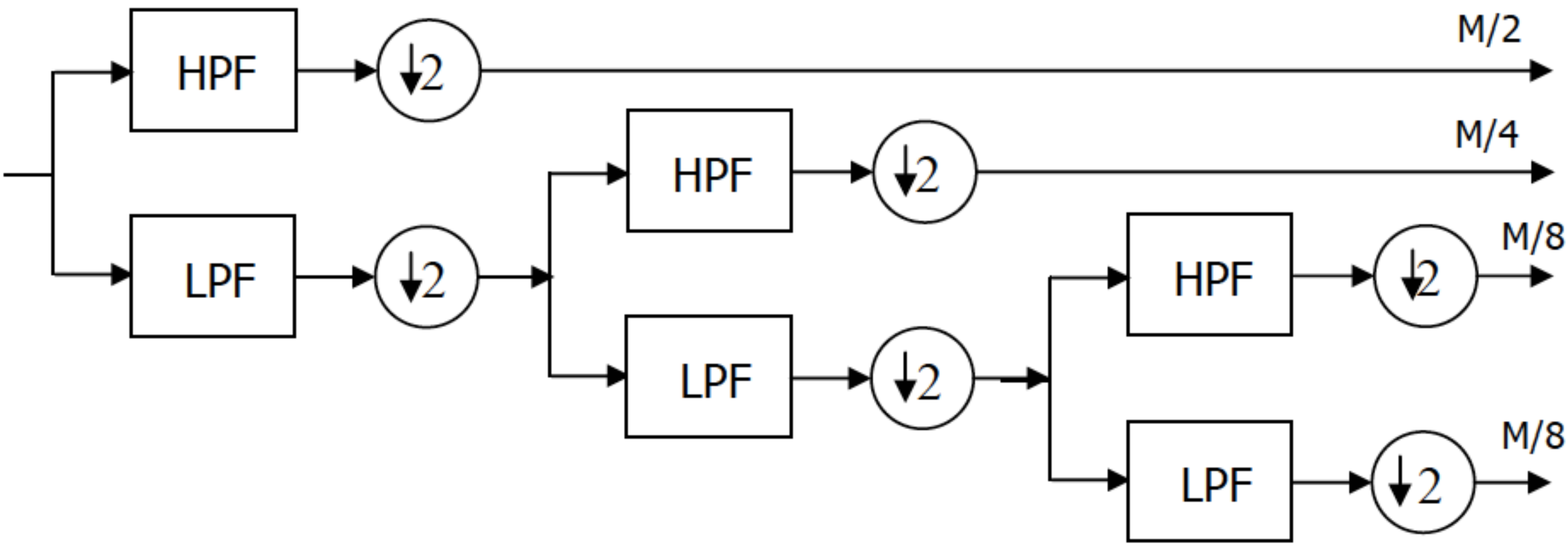


Haar Wavelets

$$D_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$D_3 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Haar Wavelets



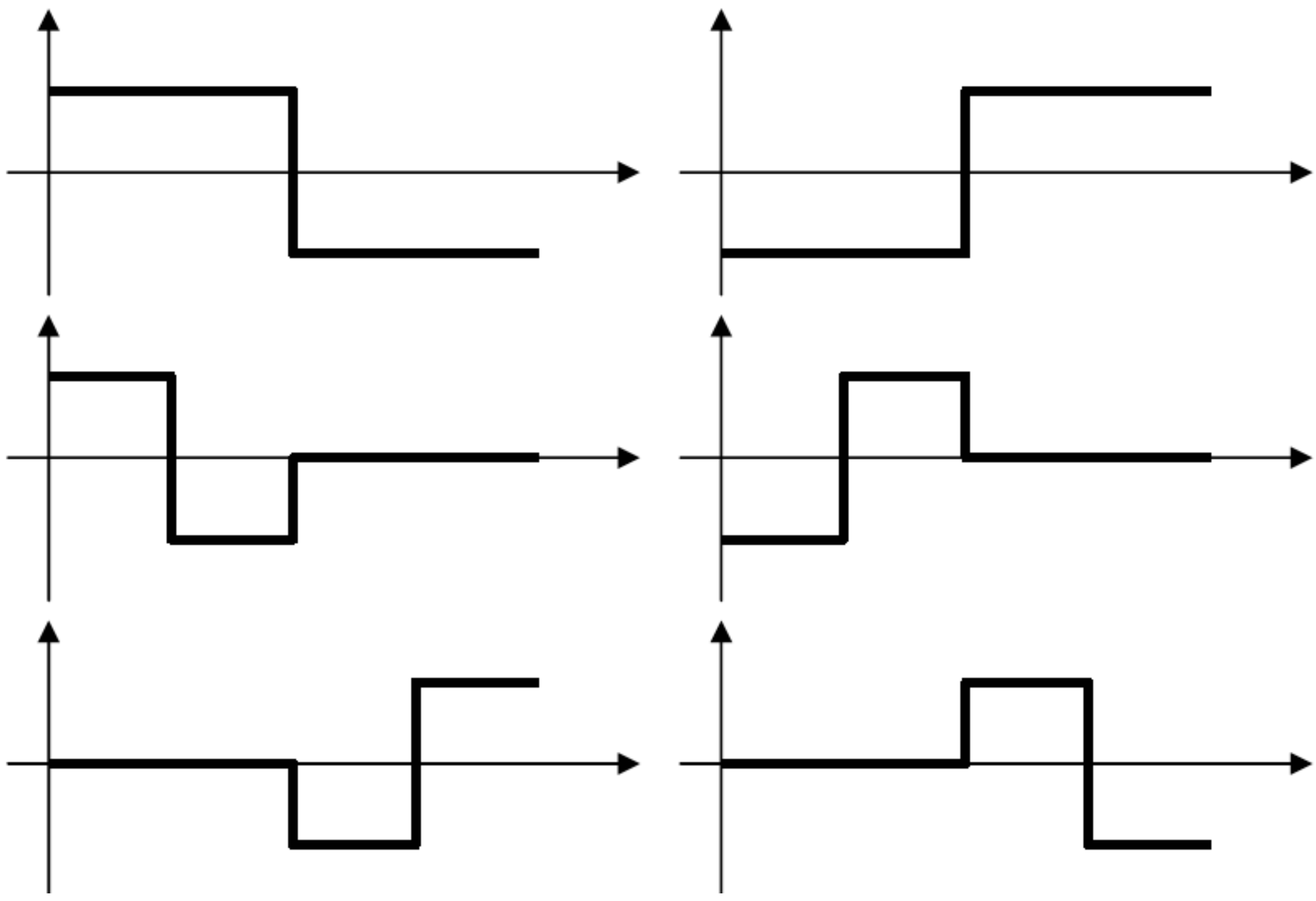
Haar Transform

Haar Wavelets

$$W_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Transform Matrix

Haar Wavelets



Properties of Daubechies wavelets

I. Daubechies, *Comm. Pure Appl. Math.* 41 (1988) 909.

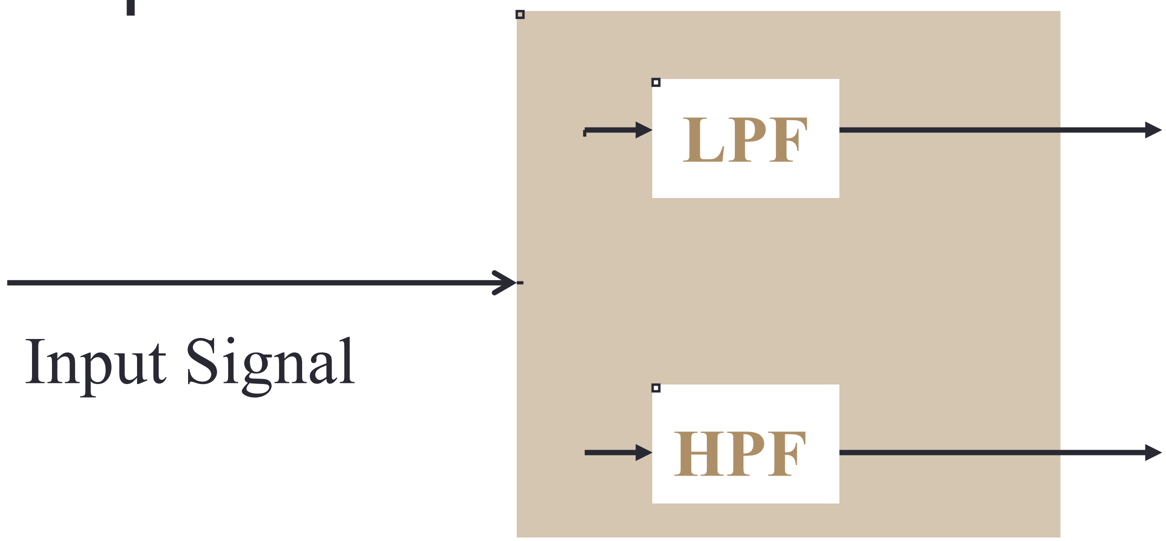
- Compact support
  - finite number of filter parameters / fast implementations
  - high compressibility
  - fine scale amplitudes are very small in regions where the function is smooth / sensitive recognition of structures
- Identical forward / backward filter parameters
  - fast, exact reconstruction
  - very asymmetric

Mallat* Filter Scheme

- Mallat was the first to implement this scheme, using a well known filter design called "two channel sub band coder", yielding a *'Fast Wavelet Transform'*

Approximations and Details:

- Approximations: High-scale, low-frequency components of the signal
- Details: low-scale, high-frequency components

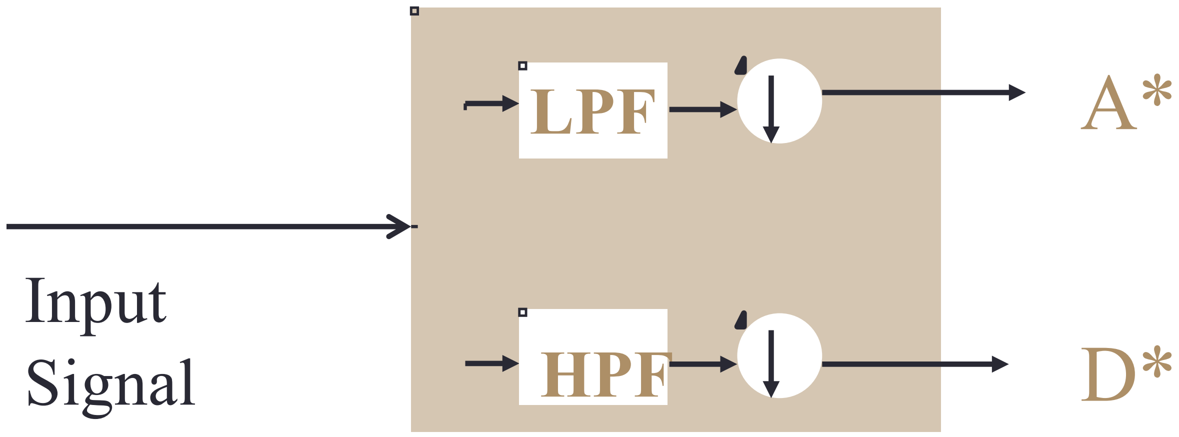


Decimation

- The former process produces twice the data it began with: N input samples produce N approximations coefficients and N detail coefficients.
- To correct this, we *Down sample* (or: *Decimate)* the filter output by two, by simply throwing away every second coefficient.

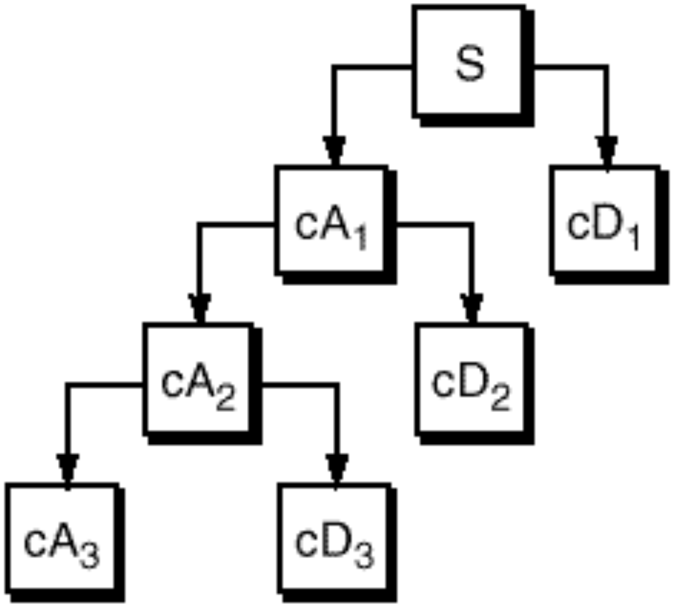
Decimation (cont'd)

So, a complete one stage block looks like:

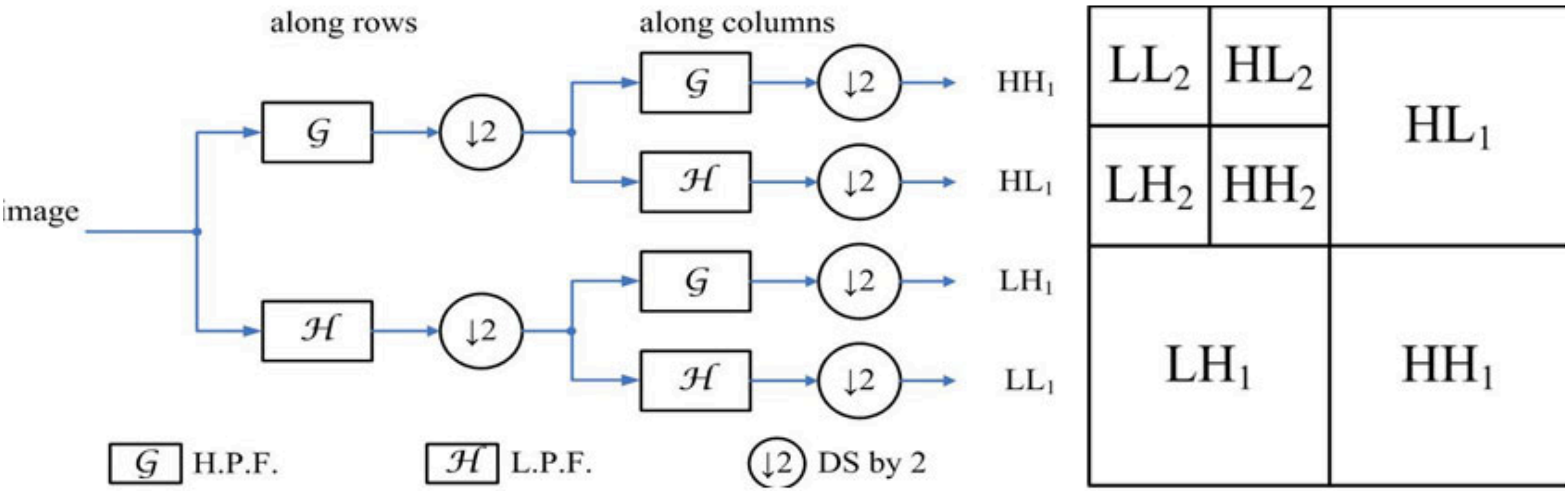


Multi-level Decomposition

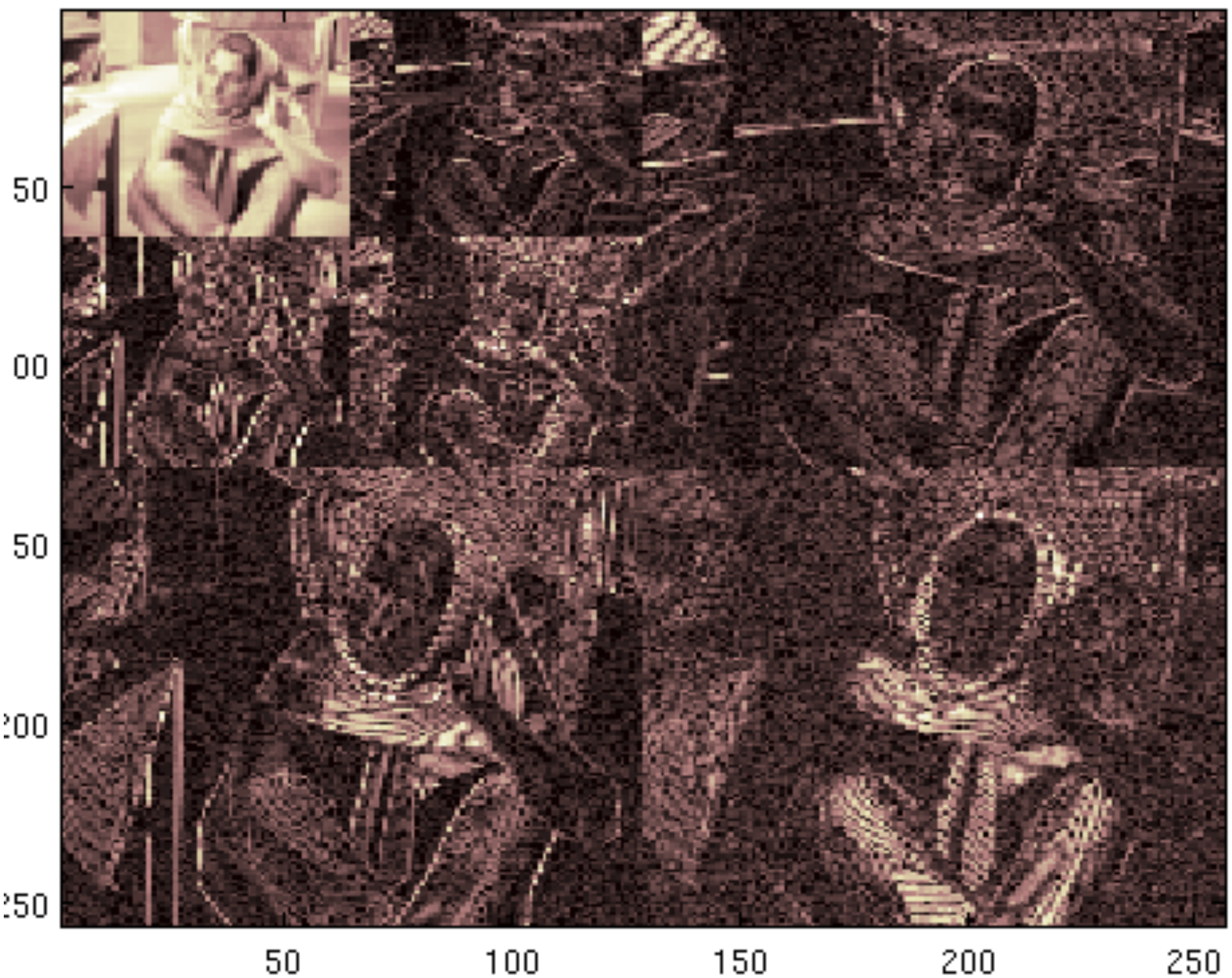
- Iterating the decomposition process, breaks the input signal into many lower-resolution components: *Wavelet decomposition tree*:

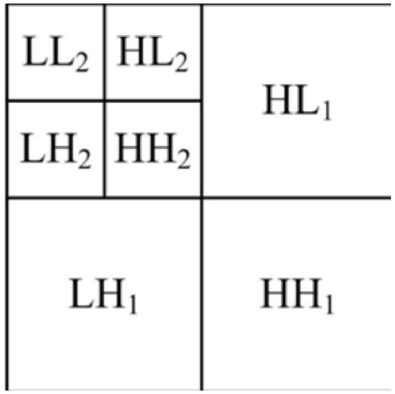


2D Wavelet Decomposition

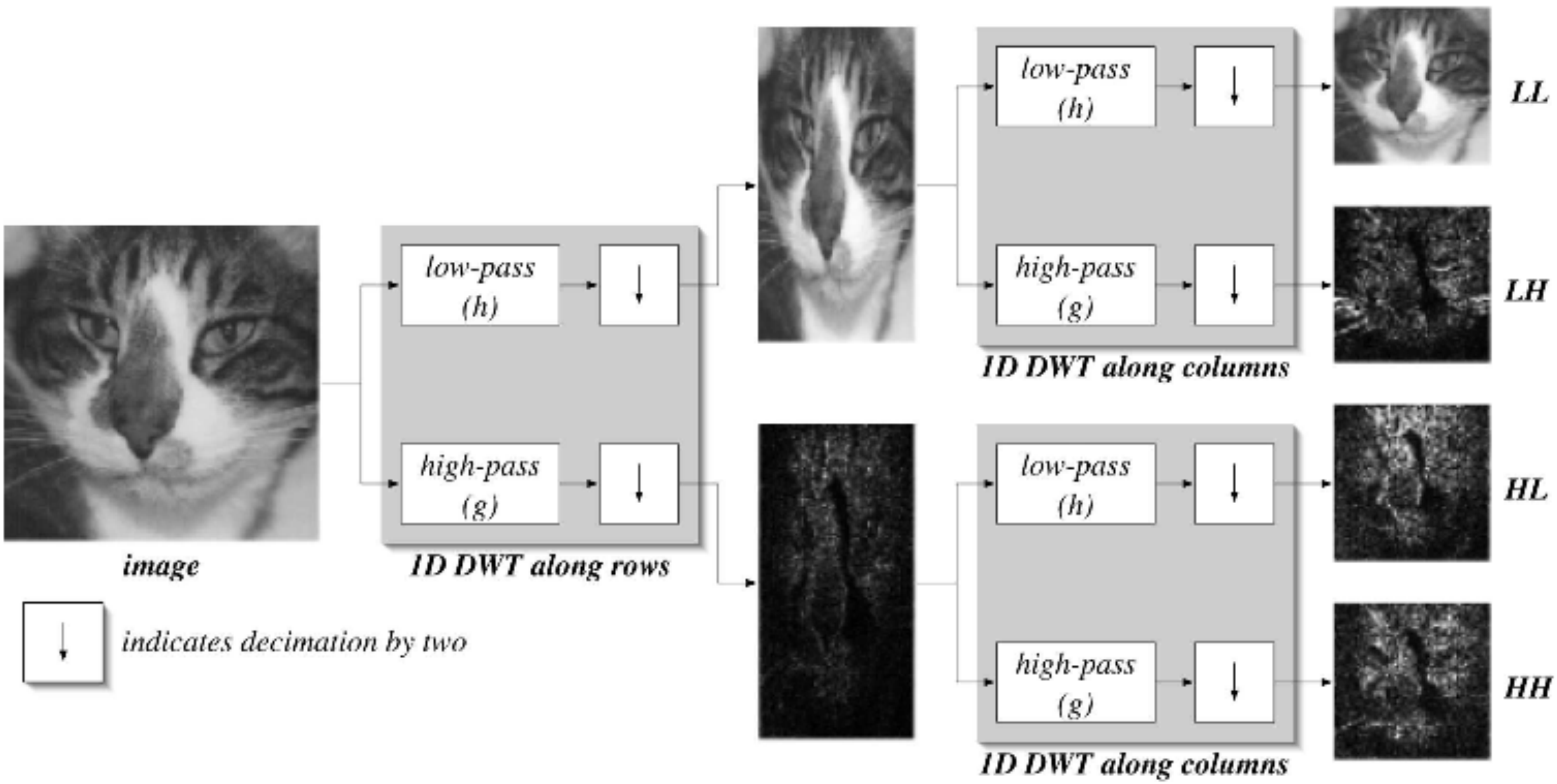


2D Wavelet transform



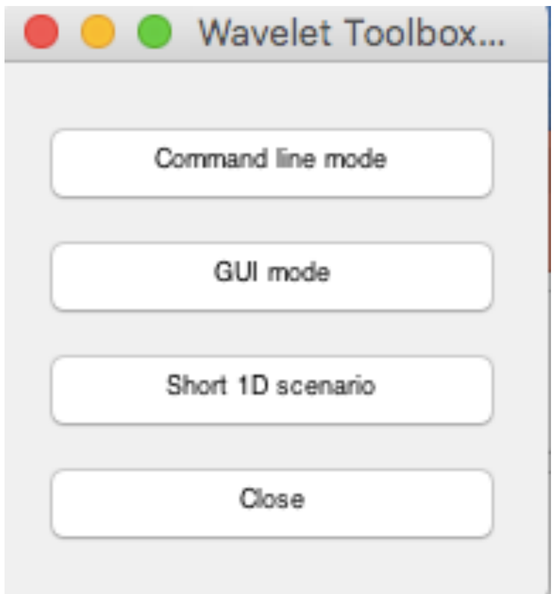


2D Wavelet transform

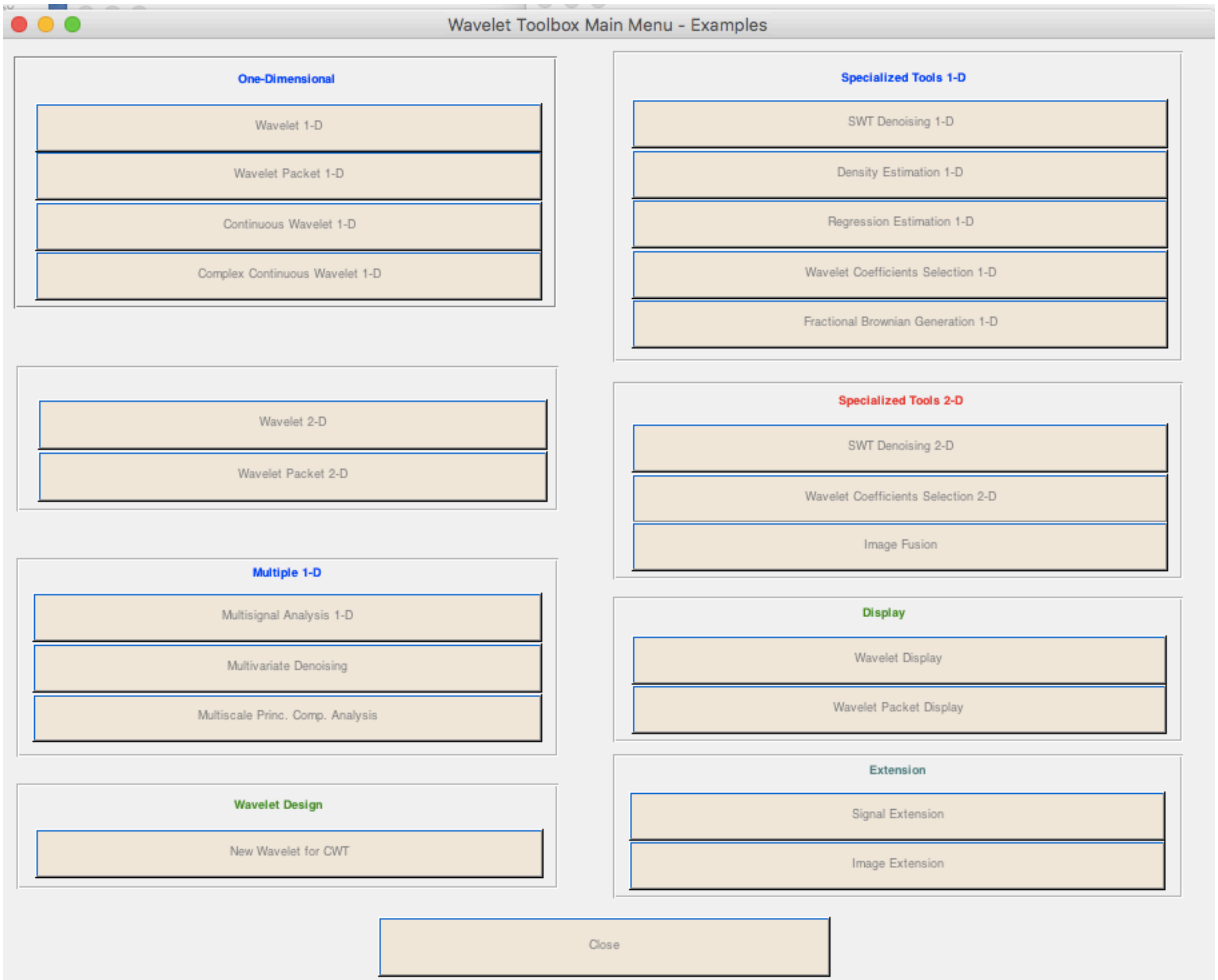


Wavelets in Matlab

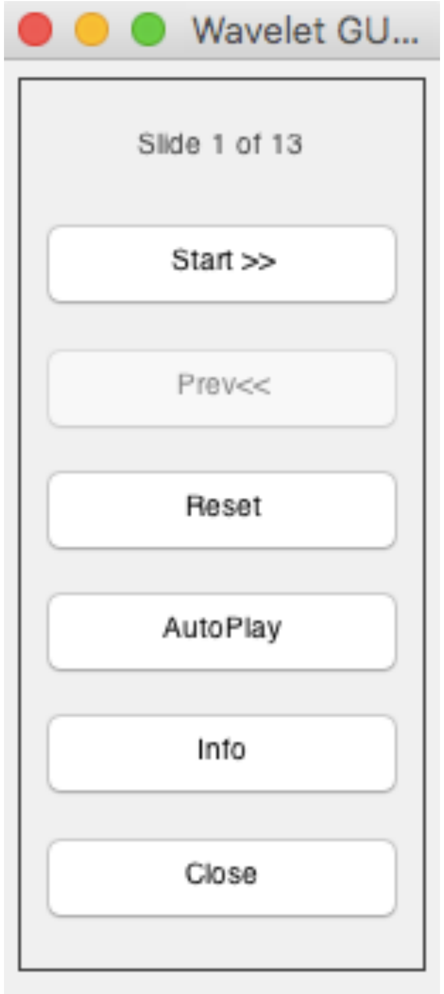
```
>> wavedemo
```



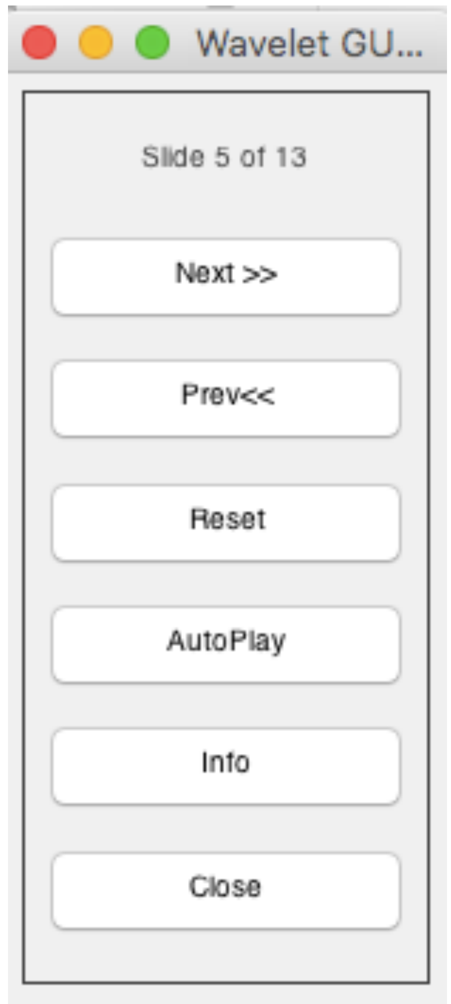
Wavelets in Matlab



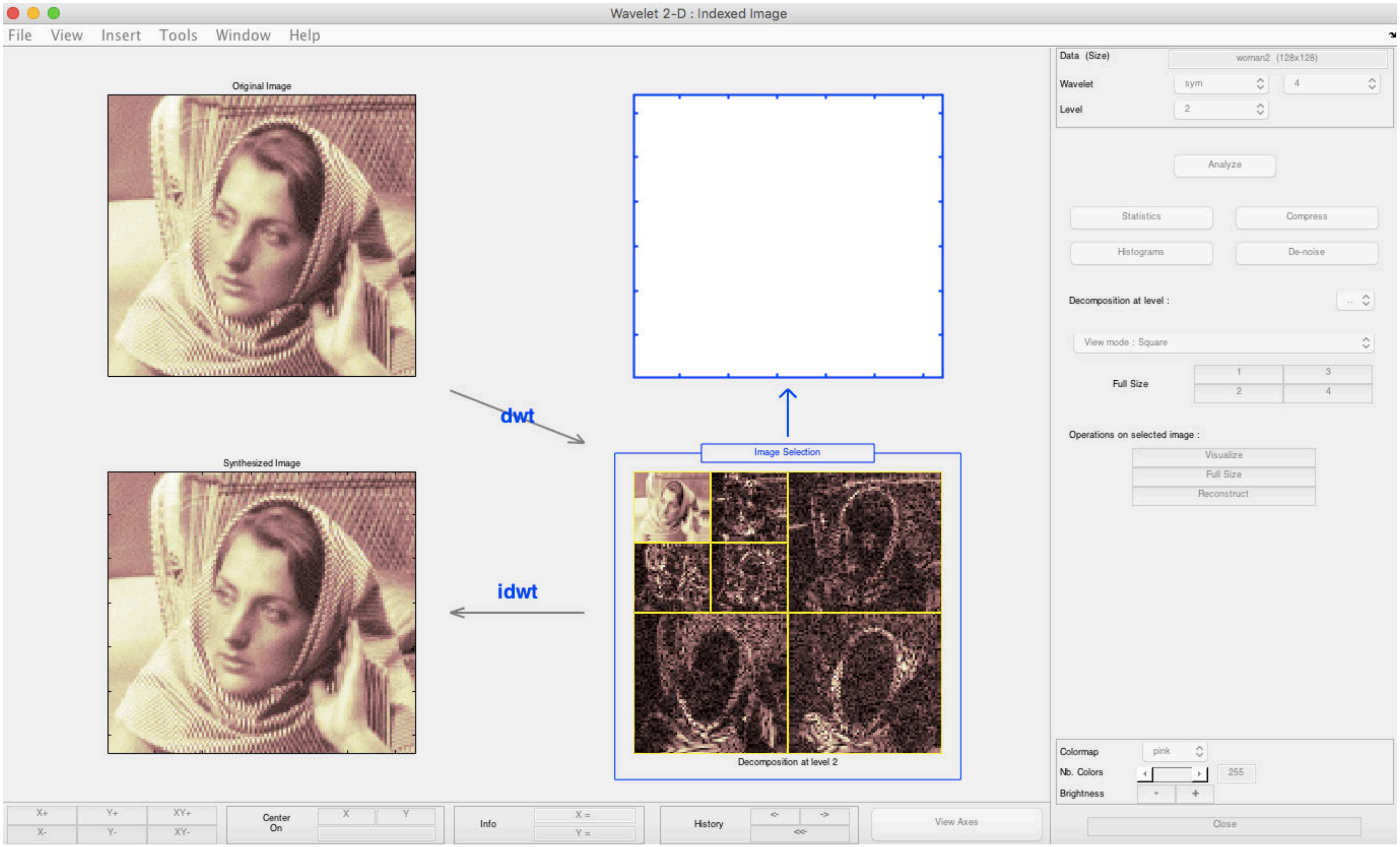
Wavelets in Matlab



Wavelets in Matlab

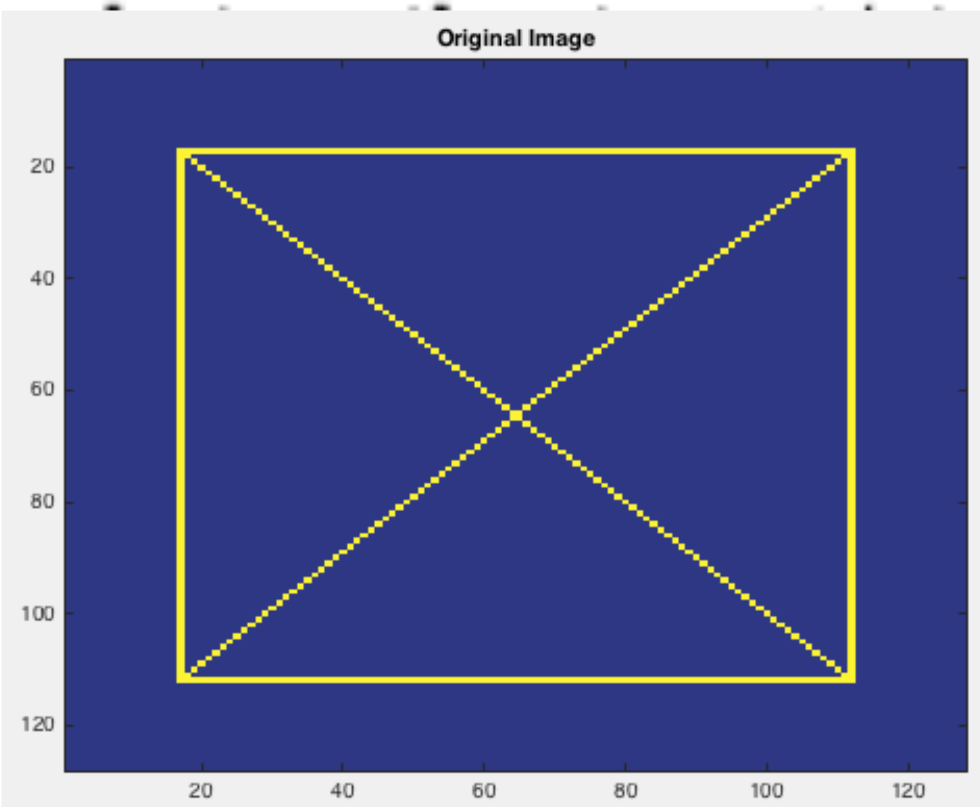


Wavelets in Matlab



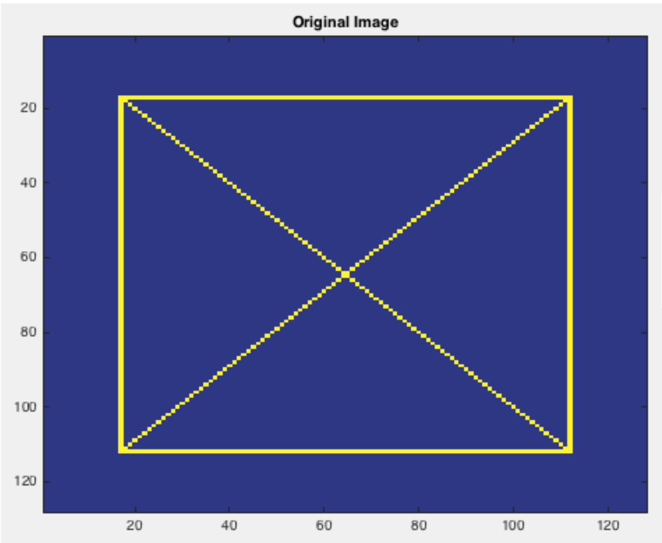
Wavelets in Matlab

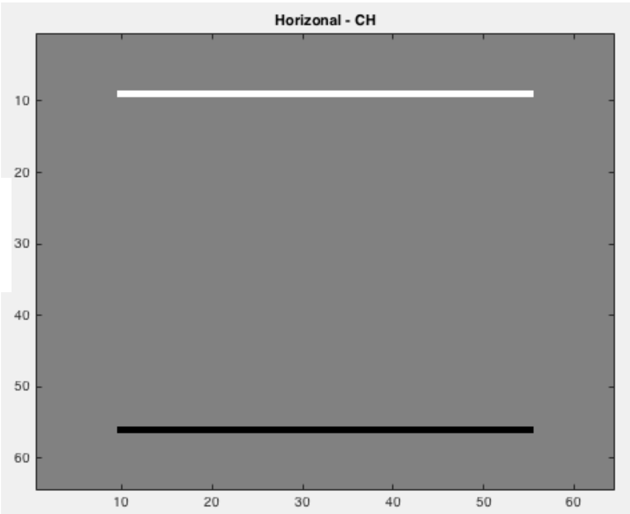
```
>> load xbox;
>> figure;
>> imagesc(xbox)
>> title('Original Image')
```



Wavelets in Matlab

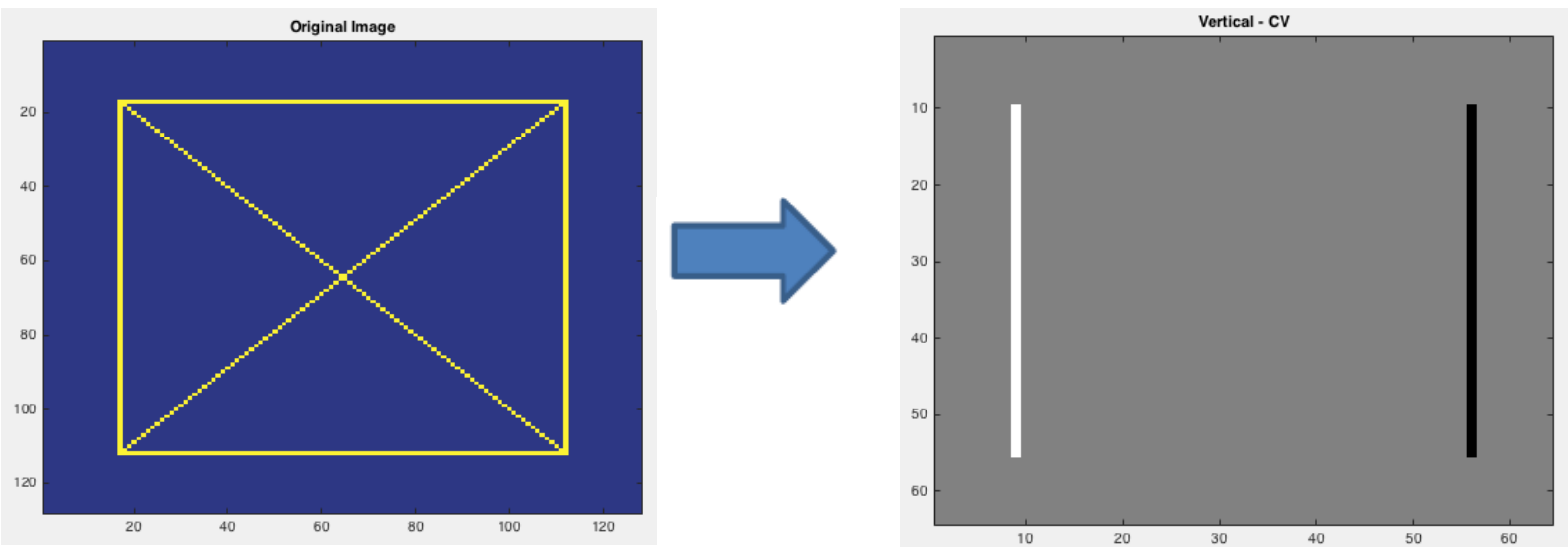
```
>> [CA,CH,CV,CD] = dwt2(xbox,'haar','mode','sym');

>> figure;colormap gray;
>> imagesc(CH)
```





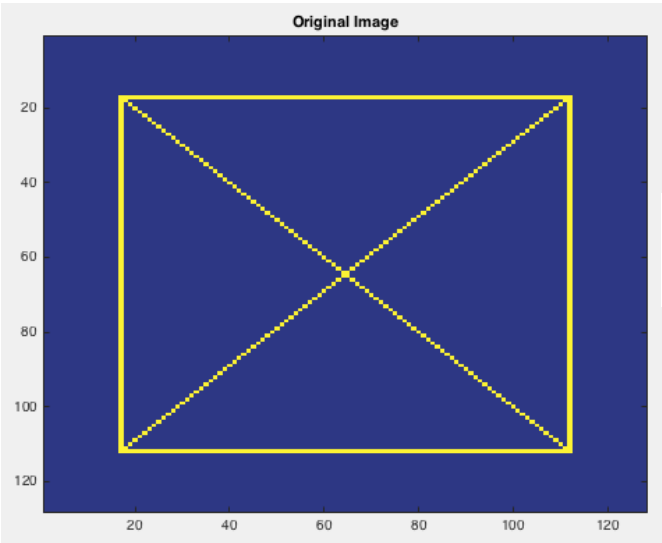
Wavelets in Matlab

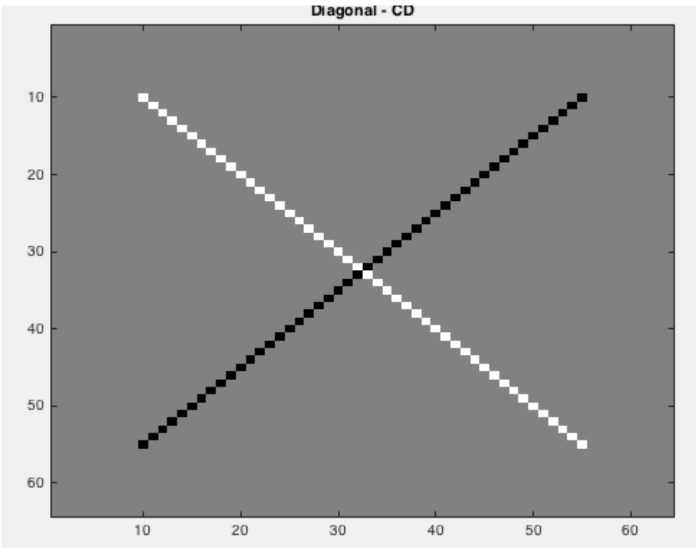
```
>> [CA,CH,CV,CD] = dwt2(xbox,'haar','mode','sym');
>> figure;colormap gray;
>> imagesc(CV)
>> title('Vertical - CV');
```



Wavelets in Matlab

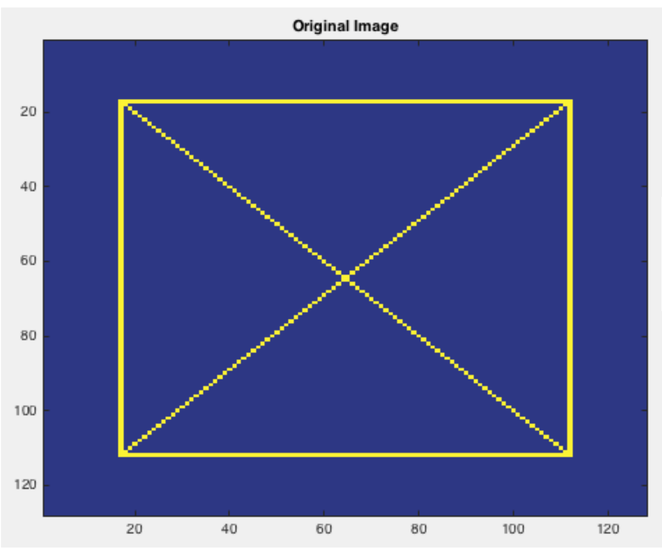
```
>> [CA,CH,CV,CD] = dwt2(xbox,'haar','mode','sym');
>> figure;colormap gray;
>> imagesc(CD)
>> title('Diagonal - CD');
```

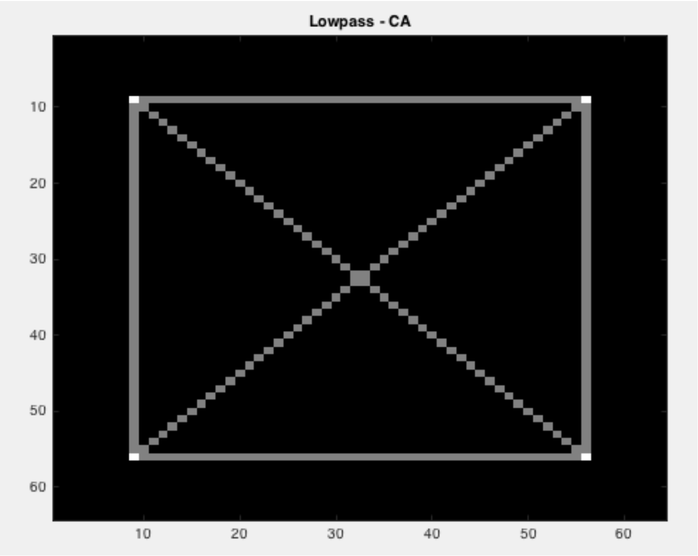




Wavelets in Matlab

```
>> [CA,CH,CV,CD] = dwt2(xbox,'haar','mode','sym');
>> figure;colormap gray;
>> imagesc(CA)
>> title('Lowpass - CA');
```





Wavelets in Matlab

```
>> [Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('haar')
```

Lo - Lowpass

Hi - Highpass

D - Decomposition

R - Reconstruction

```
Lo_D =

    0.7071    0.7071


Hi_D =

   -0.7071    0.7071


Lo_R =

    0.7071    0.7071


Hi_R =

    0.7071   -0.7071
```

Wavelets in Matlab

```
>> RGB = imread('someImage.png');
>> I = rgb2gray(RGB);
>> wname = 'db5';
>> wname = 'haar';
>> [CA,CH,CV,CD] = dwt2(I,wname,'mode','sym');
```

Alternatively

```
>> [Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('haar')

>> [CA,CH,CV,CD] = dwt2(xbox,Lo_D,Hi_D,'mode','sym');
```

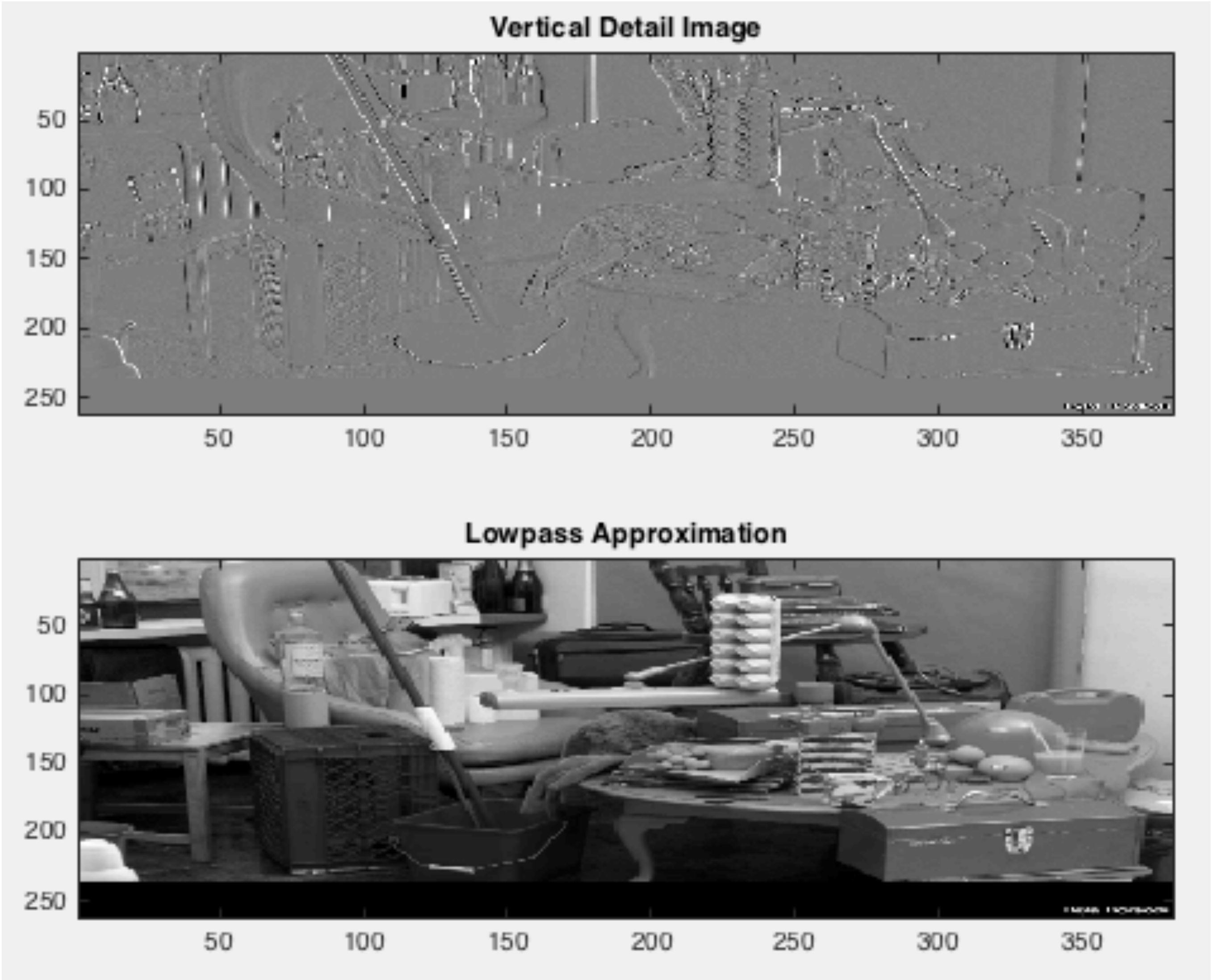
Wavelets in Matlab

```
>> RGB = imread('someImage.png');
>> I = rgb2gray(RGB);
>> wname = 'db5';
>> wname = 'haar';
>> [CA,CH,CV,CD] = dwt2(I,wname,'mode','sym');
```

```
>> subplot(211)
imagesc(CV); title('Vertical Detail Image');
colormap gray;
subplot(212)
imagesc(CA); title('Lowpass Approximation');
```

Wavelets in Matlab

Haar



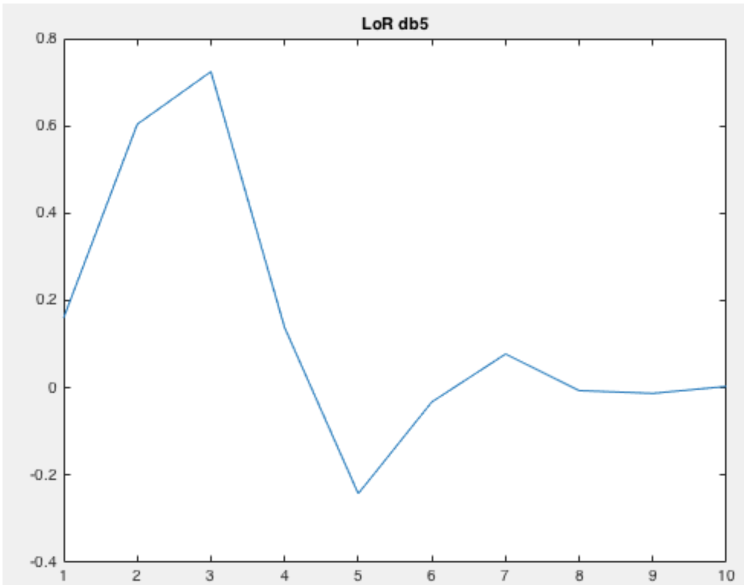
Wavelets in Matlab

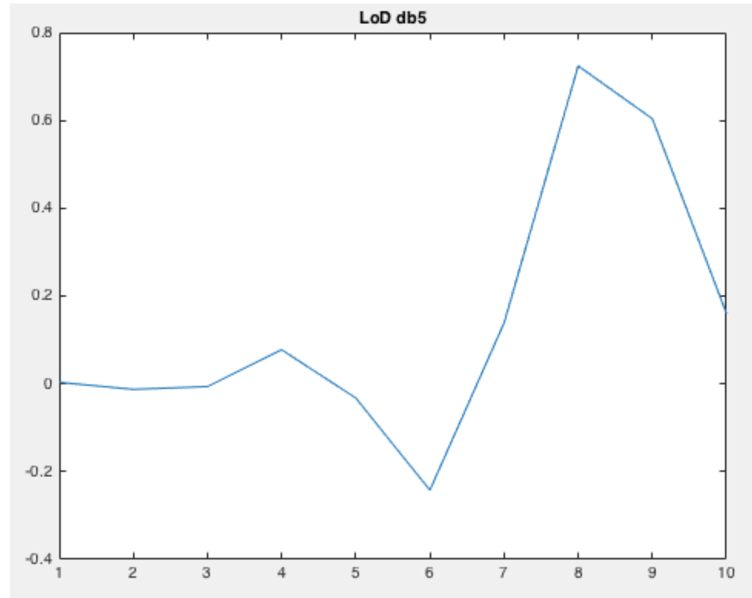
```
>> [Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('db2')

 Lo_D =

   -0.1294    0.2241    0.8365    0.4830


 Hi_D =

   -0.4830    0.8365   -0.2241   -0.1294


 Lo_R =

    0.4830    0.8365    0.2241   -0.1294


 Hi_R =

   -0.1294   -0.2241    0.8365   -0.4830
```

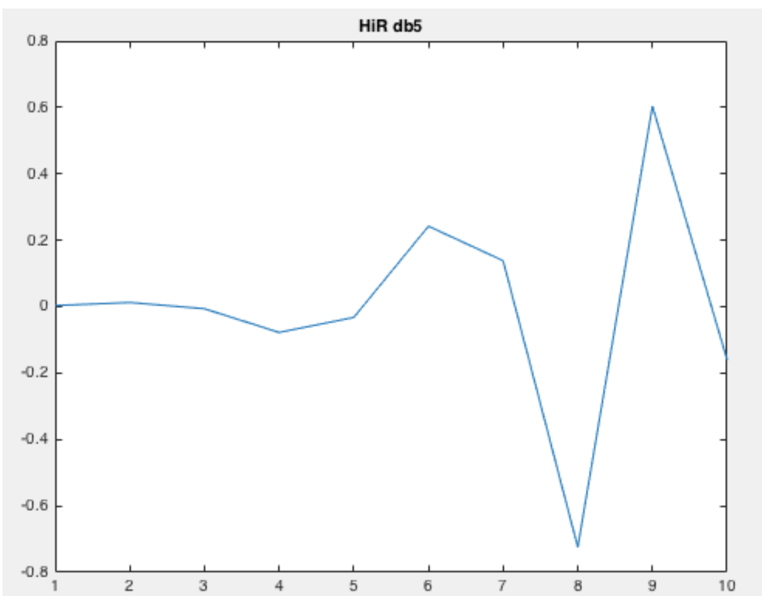
Wavelets in Matlab

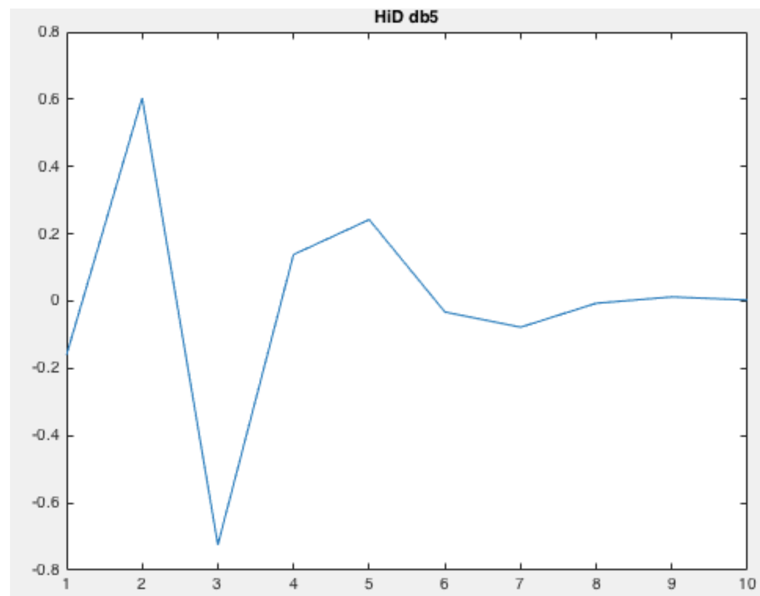
```
>> [Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('db5')

Lo_D =

    0.0033   -0.0126   -0.0062    0.0776   -0.0322   -0.2423    0.1384    0.7243    0.6038    0.1601


Hi_D =

   -0.1601    0.6038   -0.7243    0.1384    0.2423   -0.0322   -0.0776   -0.0062    0.0126    0.0033


Lo_R =

    0.1601    0.6038    0.7243    0.1384   -0.2423   -0.0322    0.0776   -0.0062   -0.0126    0.0033


Hi_R =

    0.0033    0.0126   -0.0062   -0.0776   -0.0322    0.2423    0.1384   -0.7243    0.6038   -0.1601
```

Wavelets in Matlab



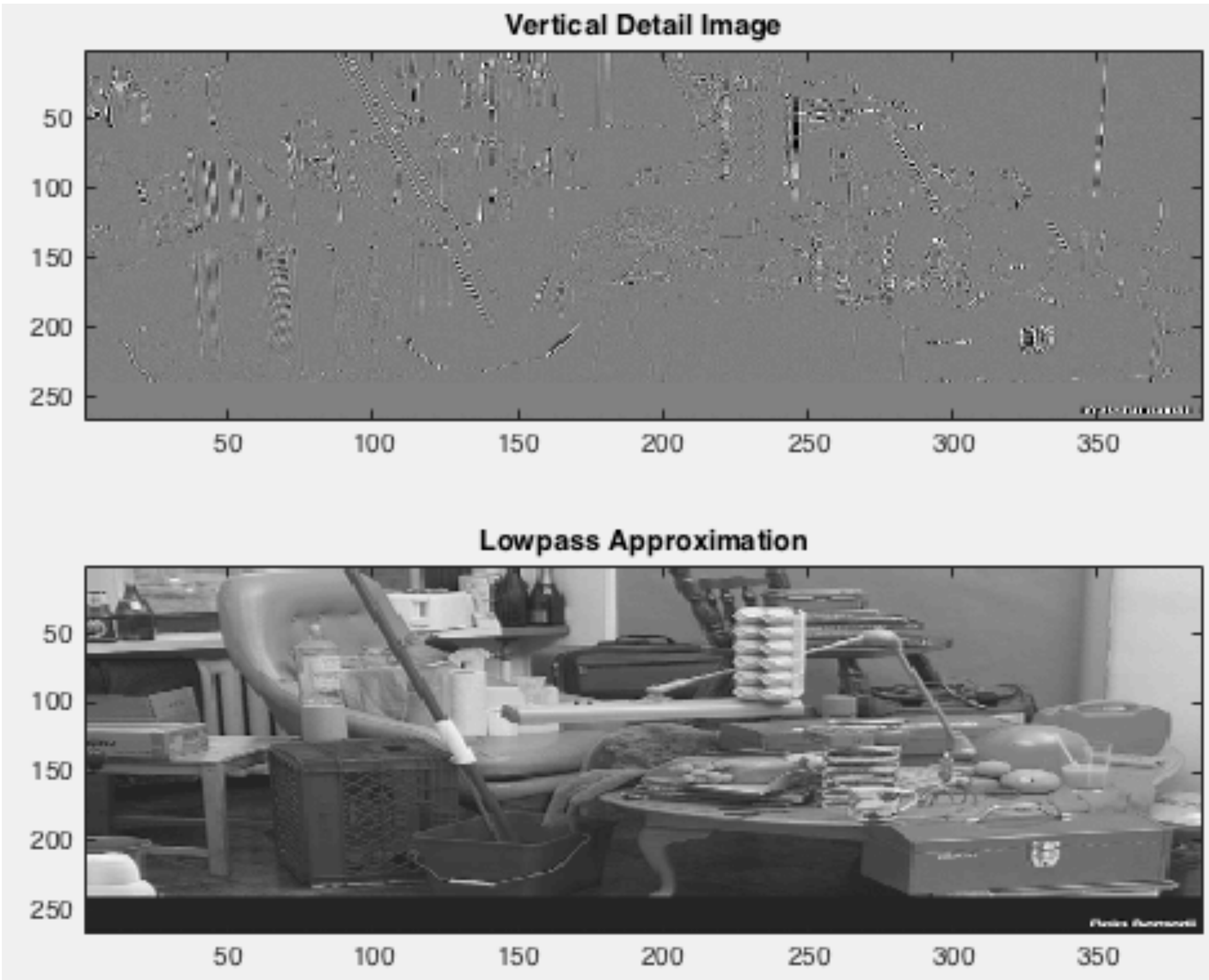






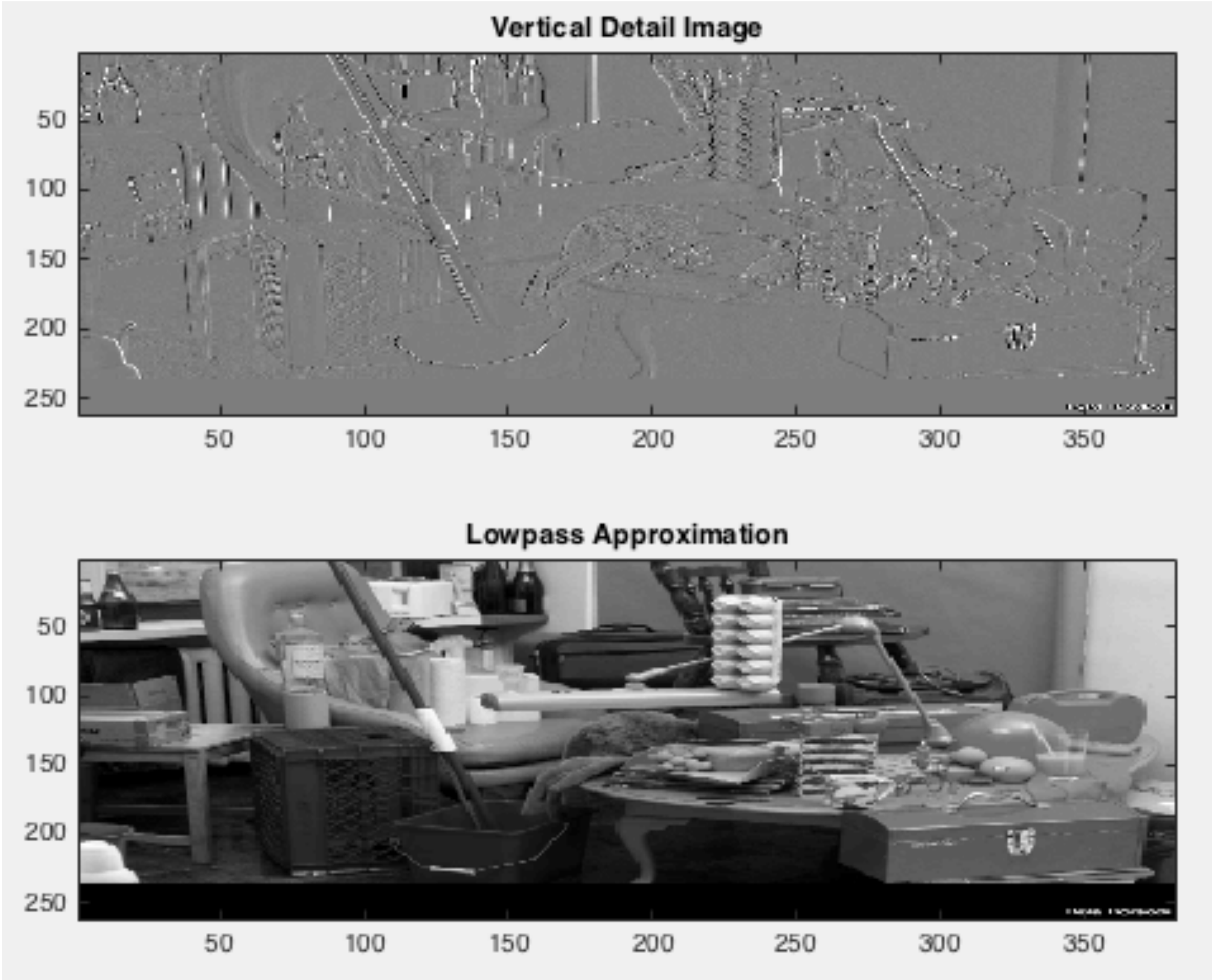
Wavelets in Matlab

db5

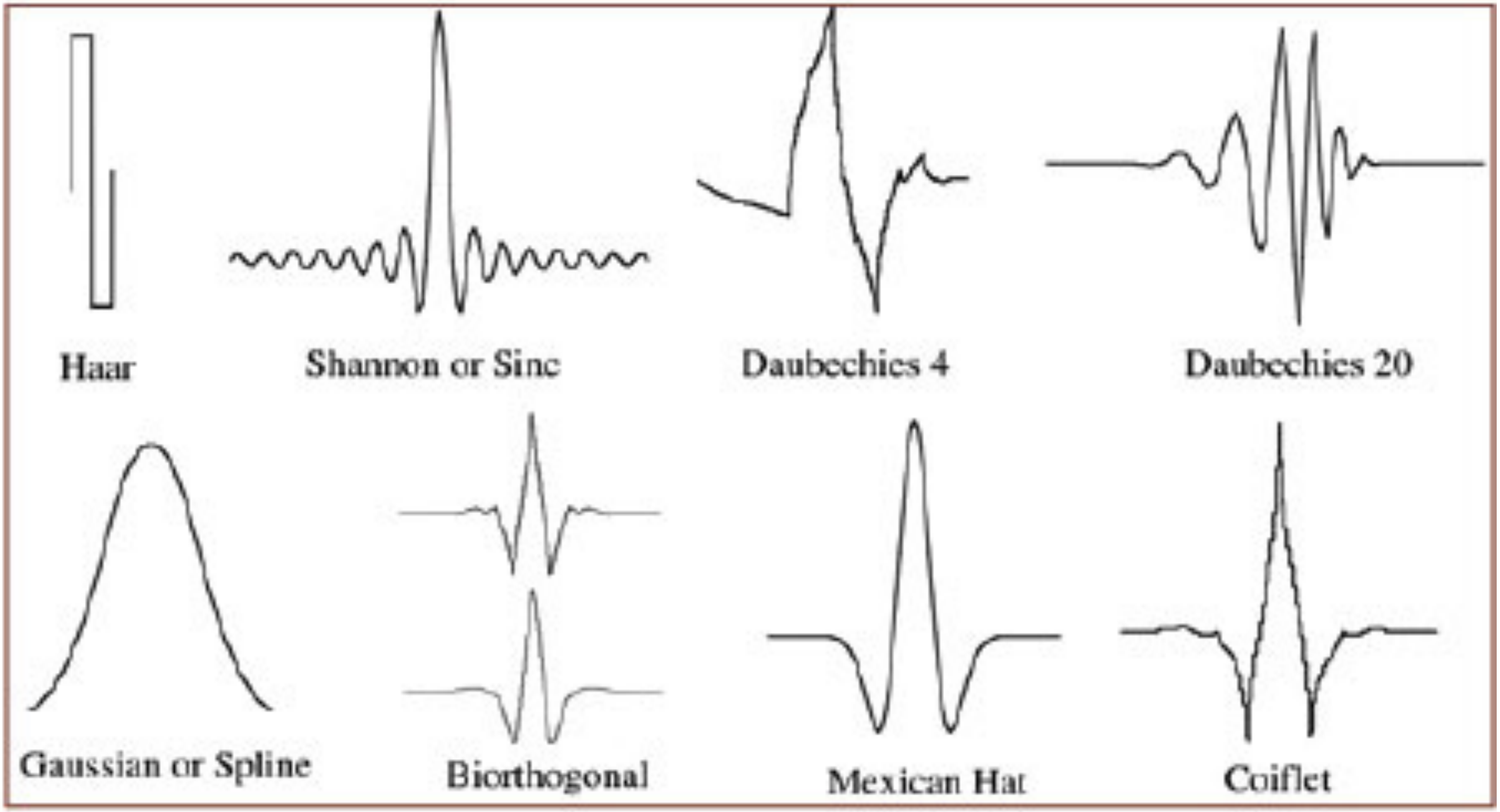


Wavelets in Matlab

Haar



Types of Wavelets



Wavelets in Matlab

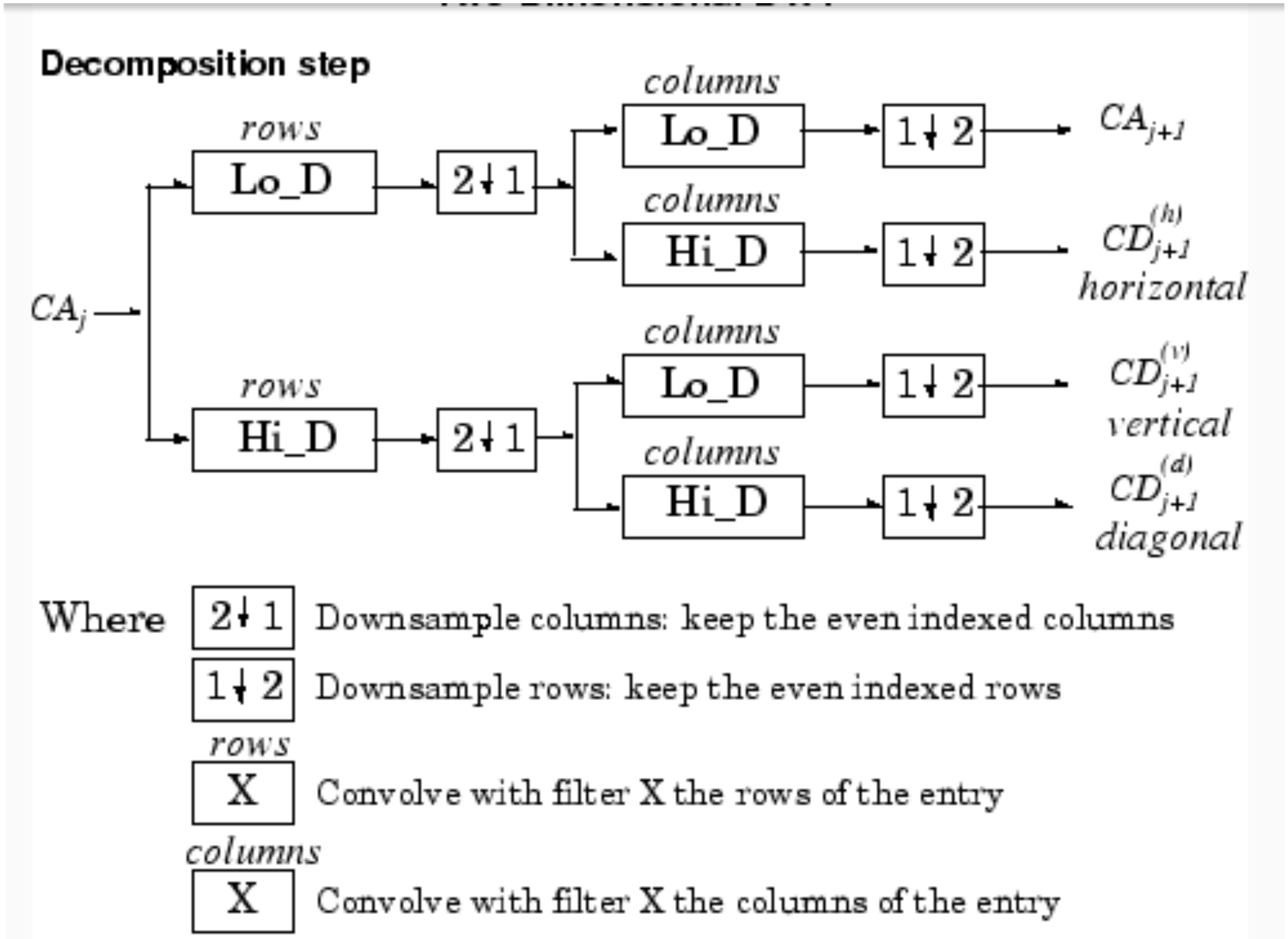
Wavelet Families	Wavelets
Daubechies	'db1' or 'haar', 'db2', ... ,'db10', ... , 'db45'
Coiflets	'coif1', ... , 'coif5'
Symlets	'sym2', ... , 'sym8', ... ,'sym45'
Fejer-Korovkin filters	'fk4', 'fk6', 'fk8', 'fk14', 'fk22'
Discrete Meyer	'dmey'
Biorthogonal	'bior1.1', 'bior1.3', 'bior1.5' 'bior2.2', 'bior2.4', 'bior2.6', 'bior2.8' 'bior3.1', 'bior3.3', 'bior3.5', 'bior3.7' 'bior3.9', 'bior4.4', 'bior5.5', 'bior6.8'
Reverse Biorthogonal	'rbio1.1', 'rbio1.3', 'rbio1.5' 'rbio2.2', 'rbio2.4', 'rbio2.6', 'rbio2.8' 'rbio3.1', 'rbio3.3', 'rbio3.5', 'rbio3.7' 'rbio3.9', 'rbio4.4', 'rbio5.5', 'rbio6.8'

Wavelets in Matlab

```
>> X = idwt2(CA,CH,CV,CD,wname);
```

```
>> figure;imagesc(X)
>> colormap gray
```

Wavelets Pyramid



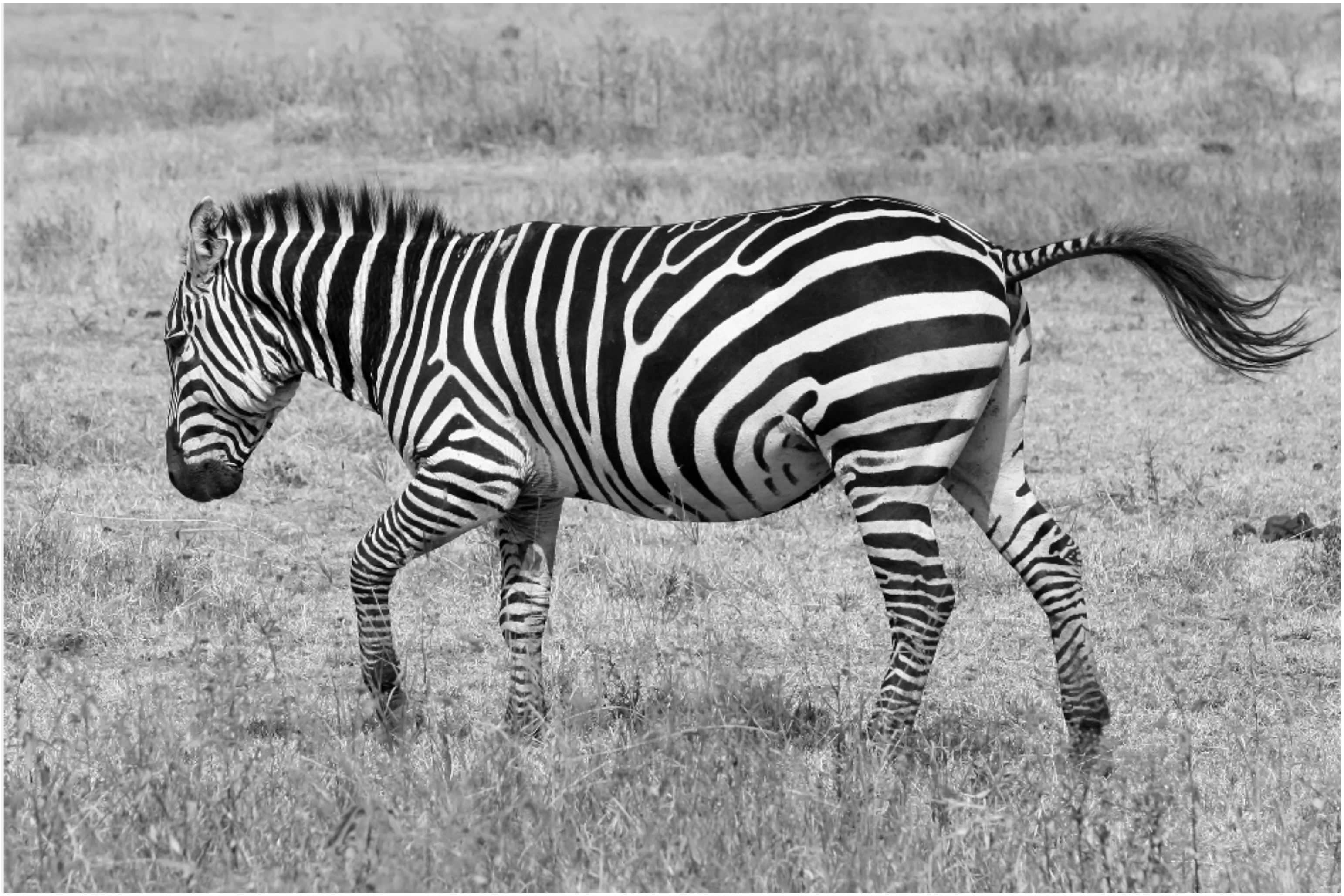
A Note about Features

Image intensities can be used to characterize an object

A Note about Features

Image intensities can be used to characterize an object

Wavelets Features



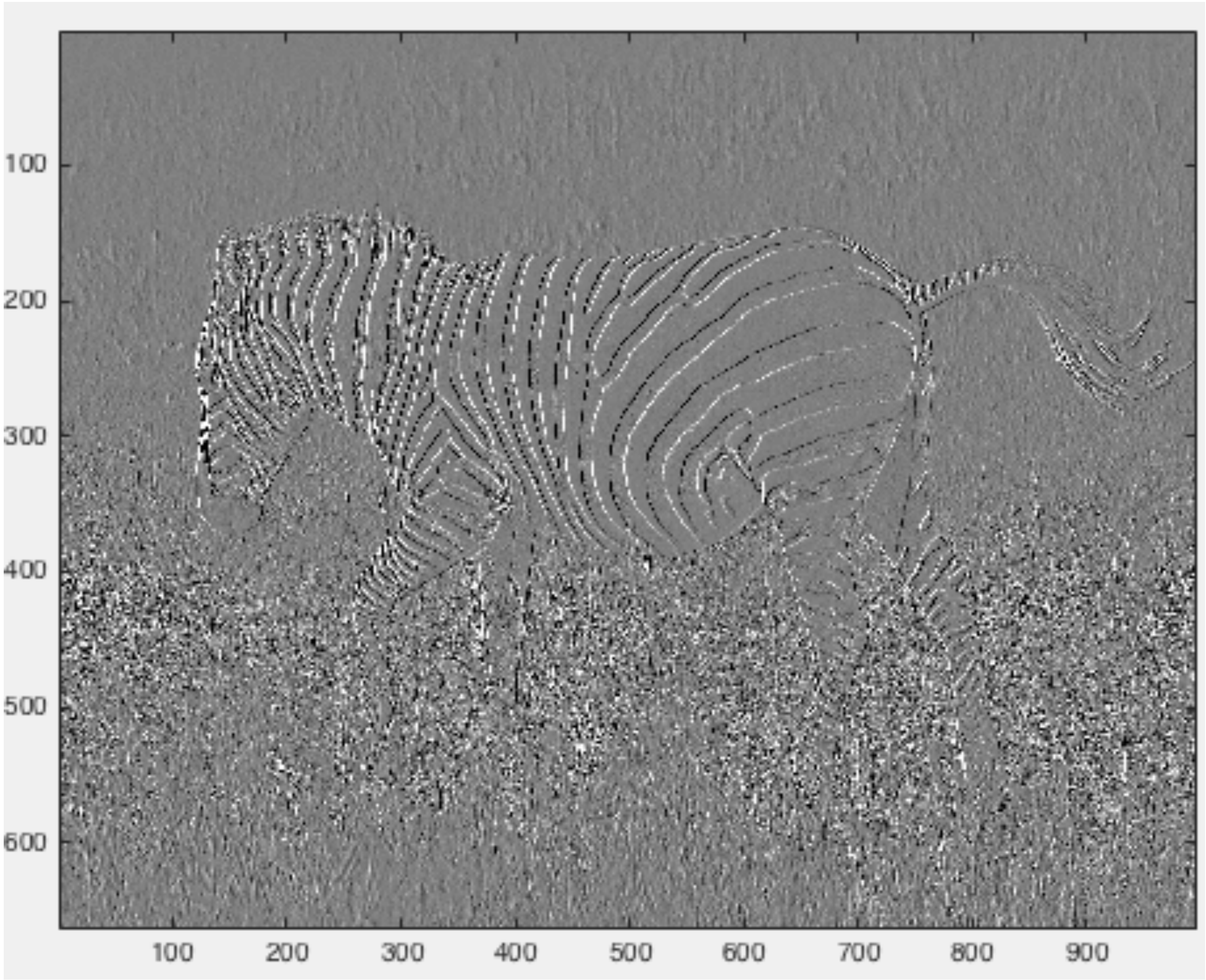
http://myths.e2bn.org/library/1359057790/zebra-running-ngorongoro.jpg

Wavelets Features

```
>> RGB = imread('zebra.jpeg');
>> I = rgb2gray(RGB);
>> figure;imshow(I)
```

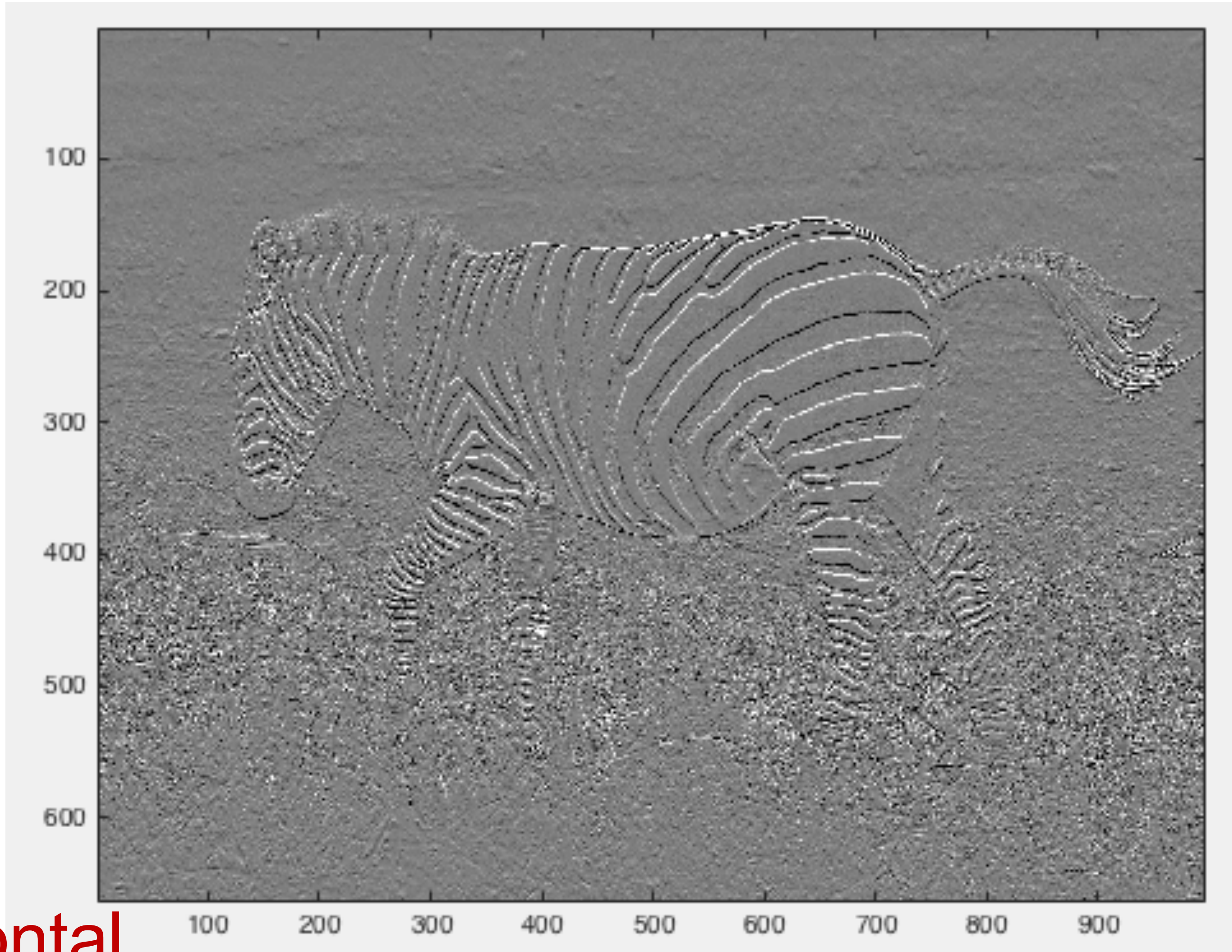
```
>> [CA,CH,CV,CD] = dwt2(I,'haar','mode','sym');
>> figure;colormap gray;
>> imagesc(CV)
>> figure;colormap gray;
>> imagesc(CH)
```

Wavelets Features



Vecrtical

Wavelets Features



Horizontal

Edge detection

- **Goal:** Identify visual changes (discontinuities) in an image.
- Intuitively, semantic information is encoded in edges.
- What are some 'causes' of visual edges?
- Canny edges

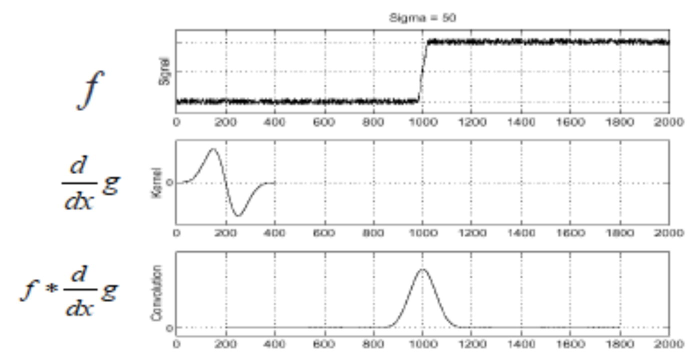
Source: D. Lowe

This class: edges & lines

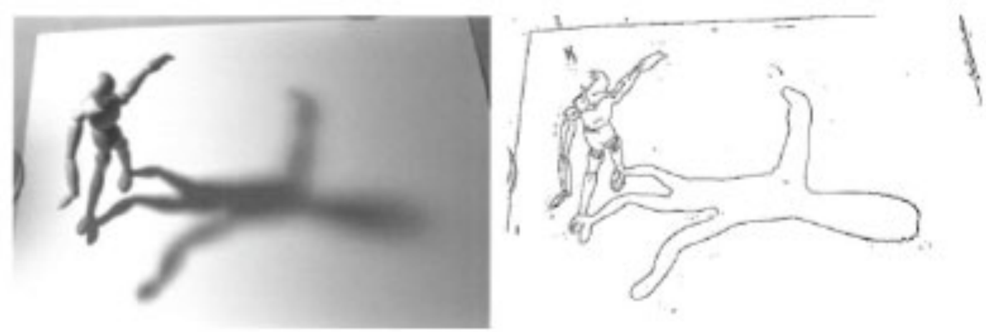
- Edge detection to identify visual change in image

- Derivative of Gaussian and linear combination of convolutions

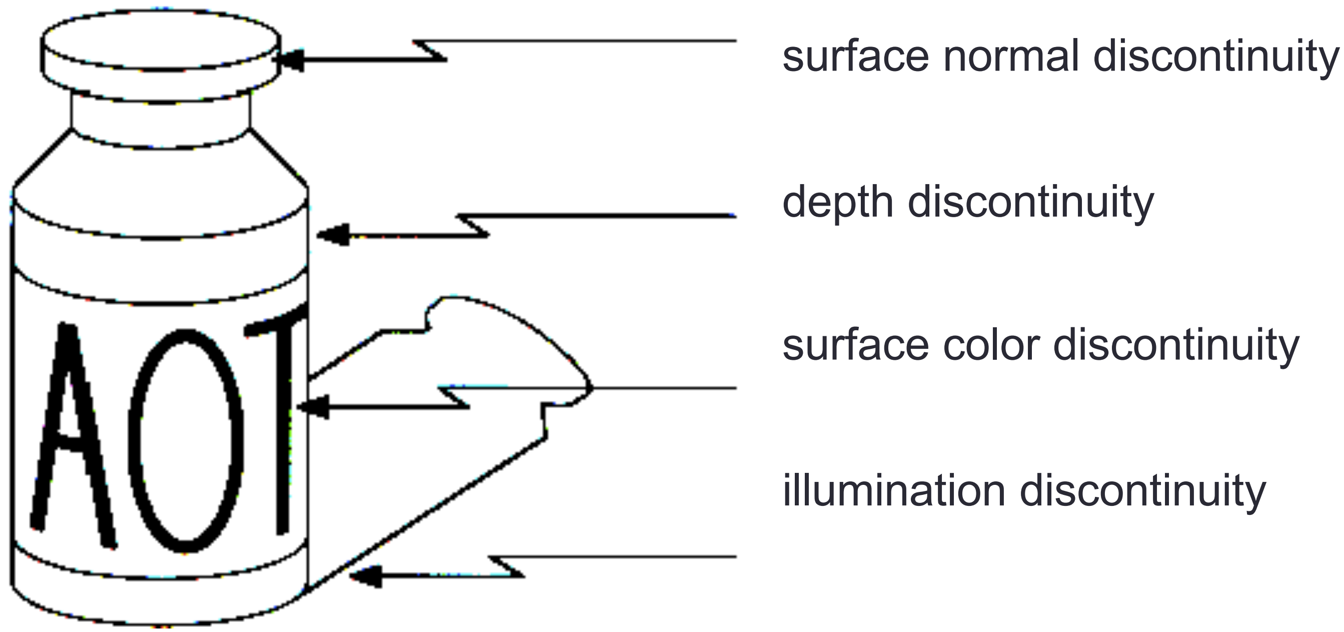
- What is an edge?
  What is a good edge?



canny edges



Origin of Edges



- Edges are caused by a variety of factors

Source: Steve Seitz

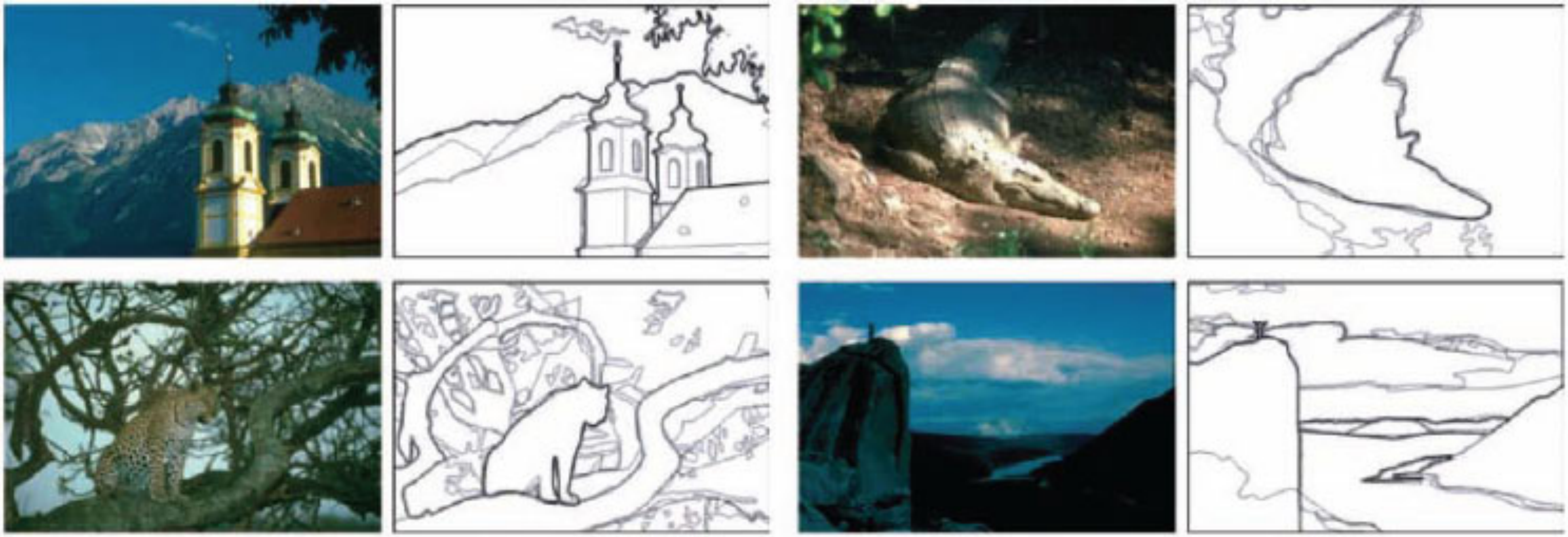
Source: Steve Seitz

Why do we care about edges?

- Extract information
  - Recognize objects

- Help recover geometry and viewpoint

Where do humans see boundaries?



Where do humans see boundaries?

image | human segmentation | gradient magnitude

- Berkeley segmentation database:
  http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

pB slides: Hays

Some questions

- What is a good edge detector?
- Do we lose information when we look at edges? Are edges ‘incomplete’ as a representation of images?

Designing an edge detector

- Criteria for a good edge detector:
  - **Good detection:** the optimal detector should find all real edges, ignoring noise or other artifacts
  - **Good localization**
    - the edges detected must be as close as possible to the true edges
    - the detector must return one point only for each true edge point
- Cues of edge detection
  - Differences in color, intensity, or texture across the boundary
  - Continuity and closure
  - High-level knowledge

Source: L. Fei-Fei

Designing an edge detector

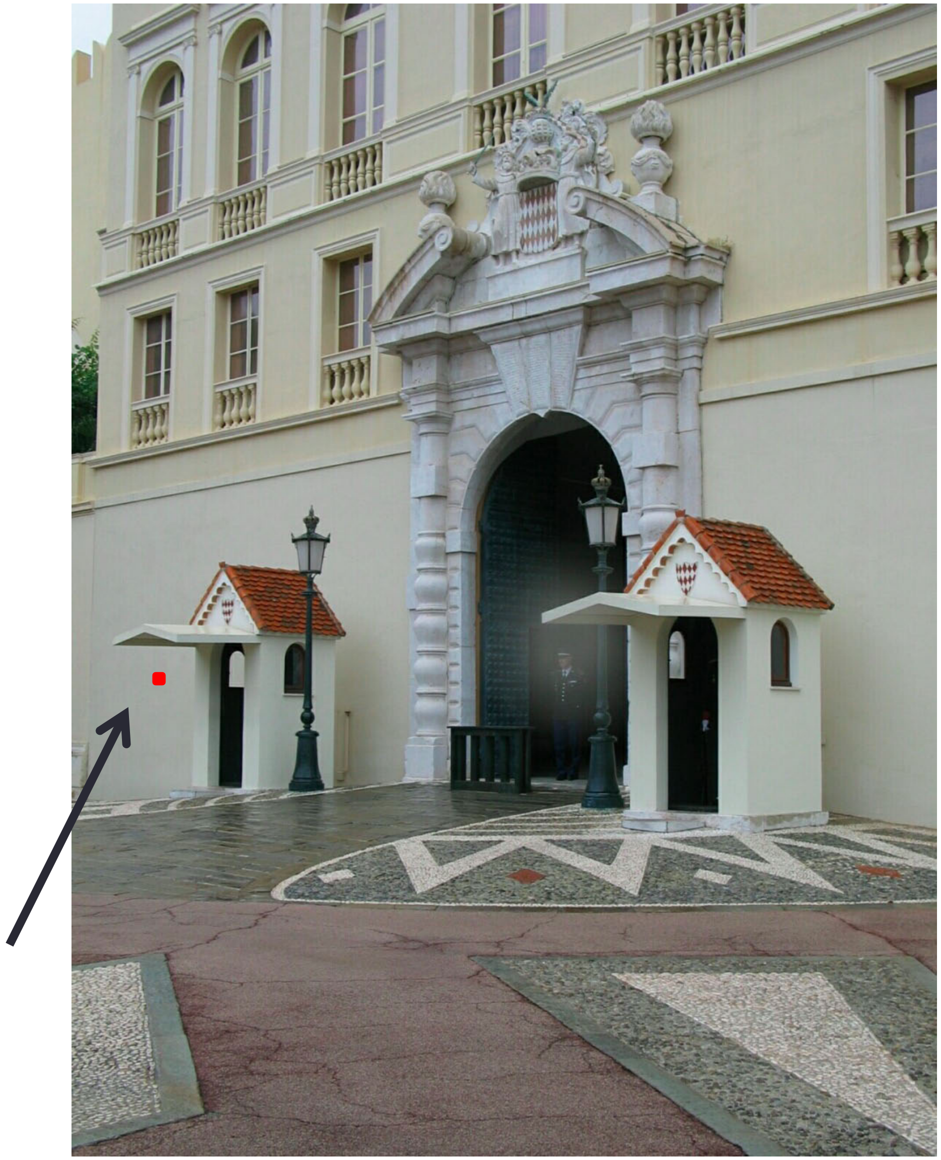
- “All real edges”
  - We can aim to differentiate later on which edges are ‘useful’ for our applications.
  - If we can’t find all things which *could* be called an edge, we don’t have that choice.

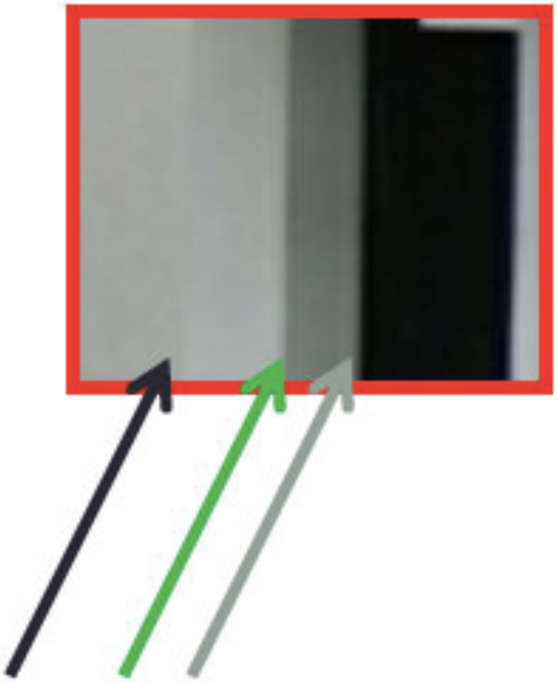
- Is this possible?

Closeup of edges

Source: D. Hoiem

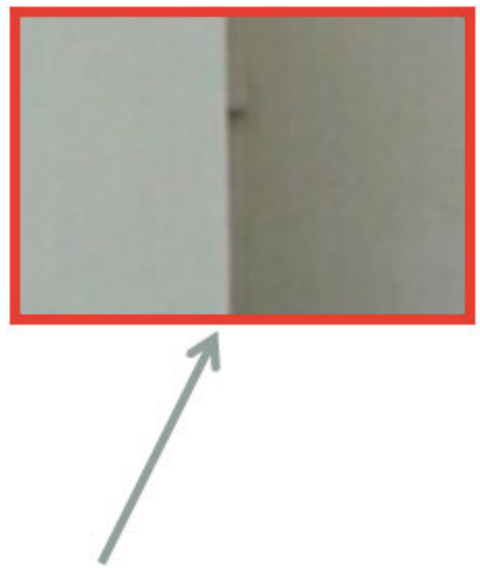
Closeup of edges





Source: D. Hoiem

Closeup of edges



Source: D. Hoiem

Closeup of edges

Characterizing edges

- An edge is a place of rapid change in the image intensity function

image

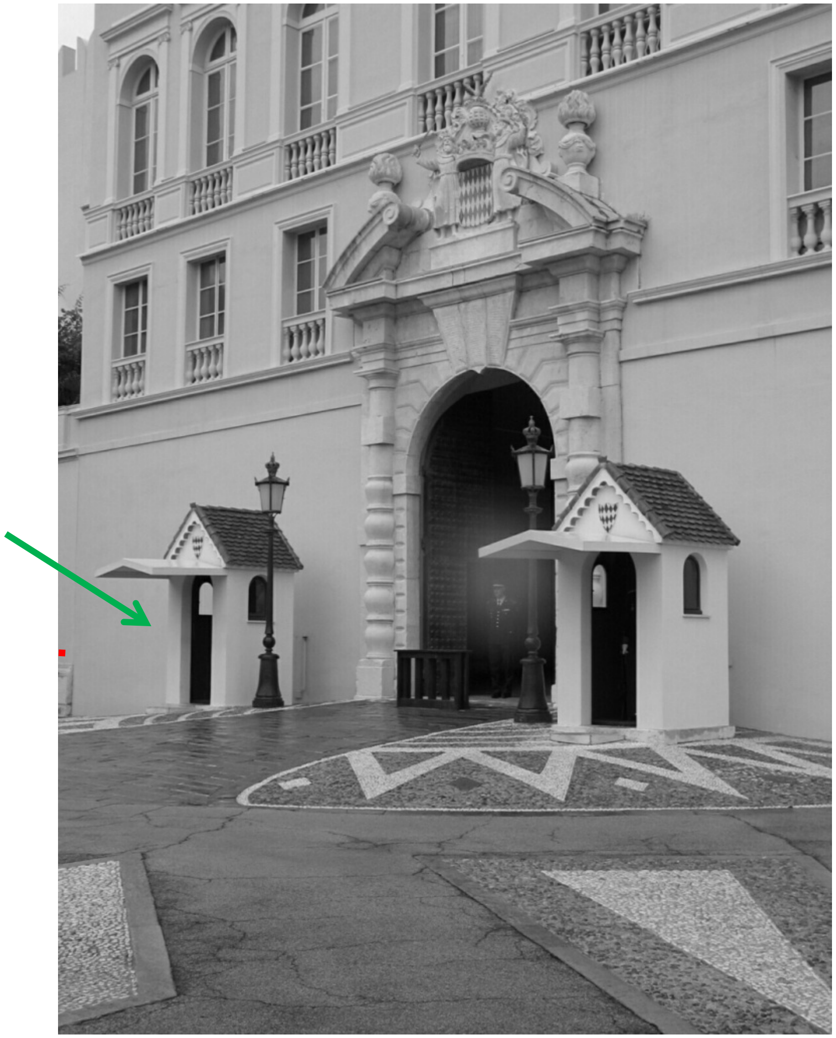
intensity function
(along horizontal scanline)

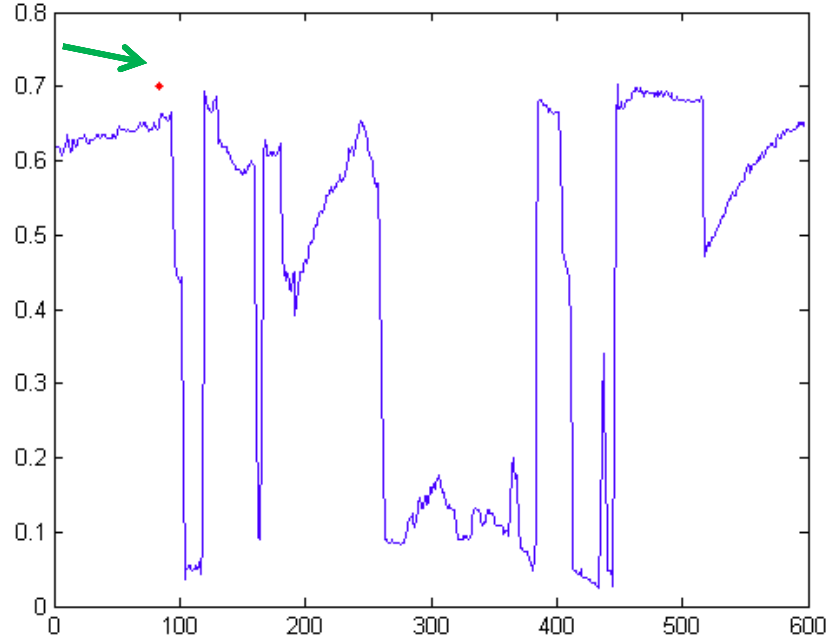
first derivative

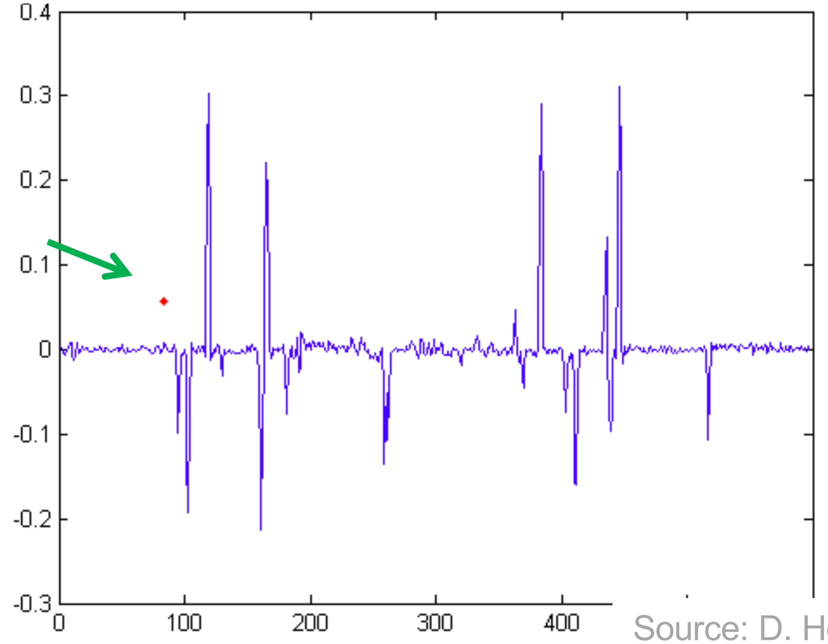
edges correspond to
extrema of derivative

Hays

Intensity profile

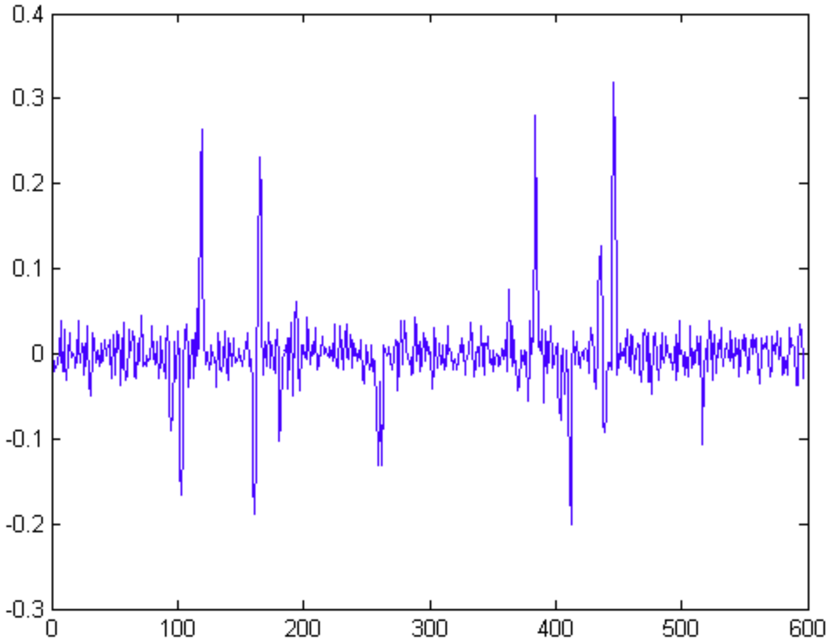






Source: D. Hoiem

With a little Gaussian noise

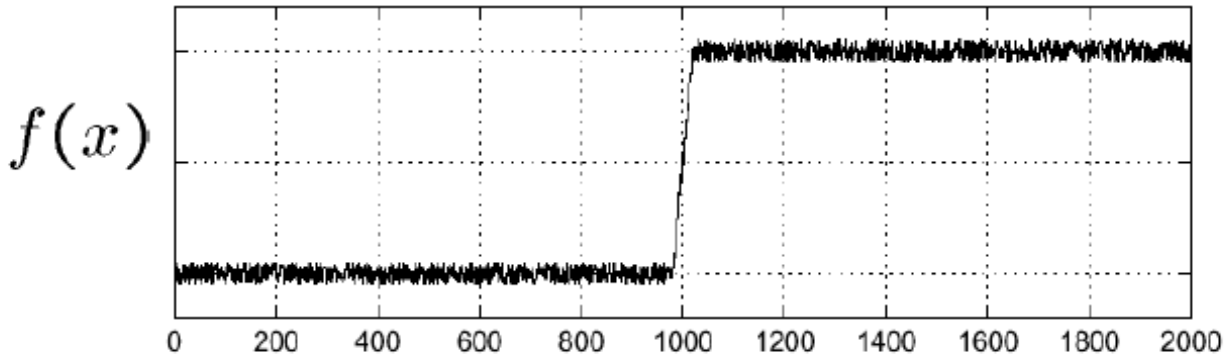


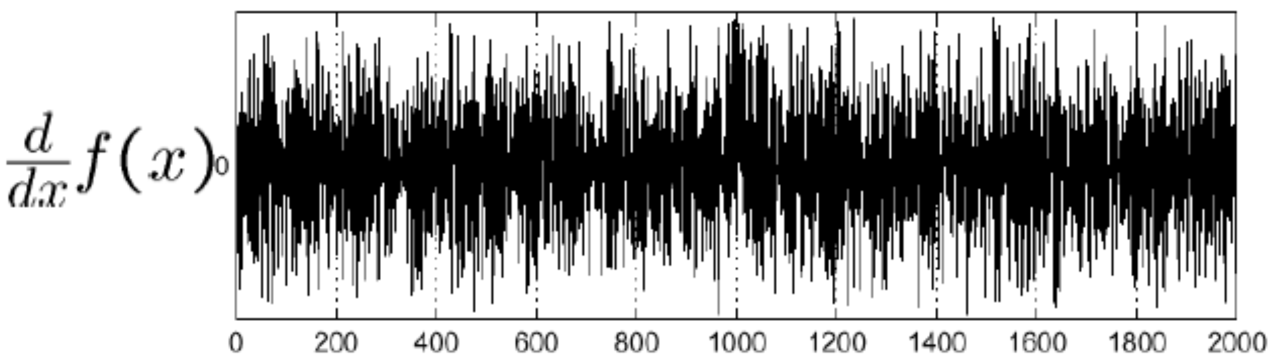
Gradient

Source: D. Hoiem

Effects of noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal





Where is the edge?

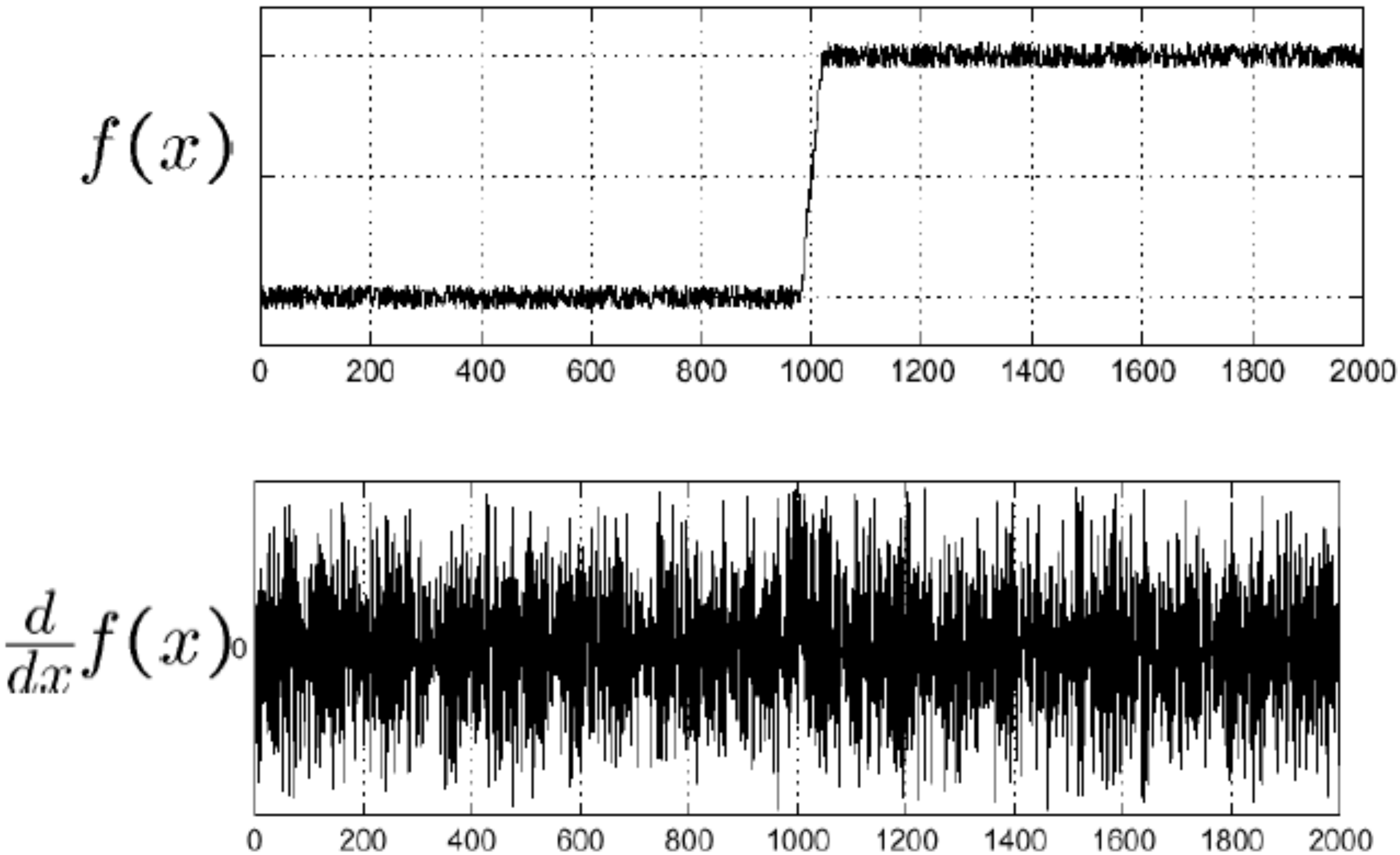
Source: S. Seitz

Effects of noise

- Difference filters respond strongly to noise
  - Image noise results in pixels that look very different from their neighbors
  - Generally, the larger the noise the stronger the response
- What can we do about it?

Source: D. Forsyth

Solution: smooth first



Where is the edge?

Source: S. Seitz

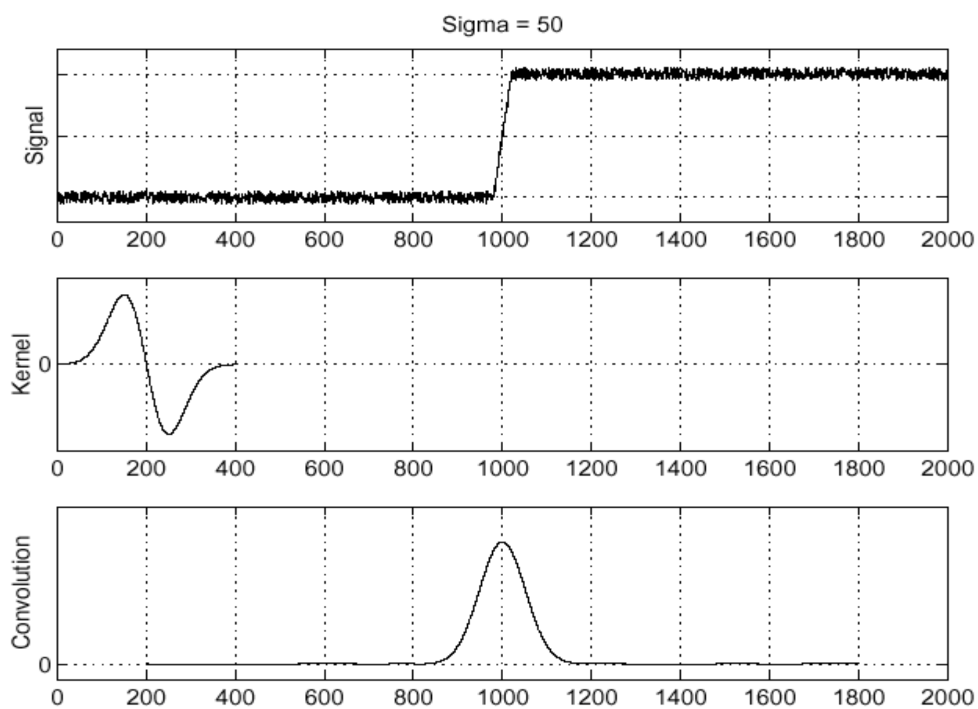
Derivative theorem of convolution

- Differentiation is convolution, and convolution is associative: $$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$
- This saves us one operation:

f

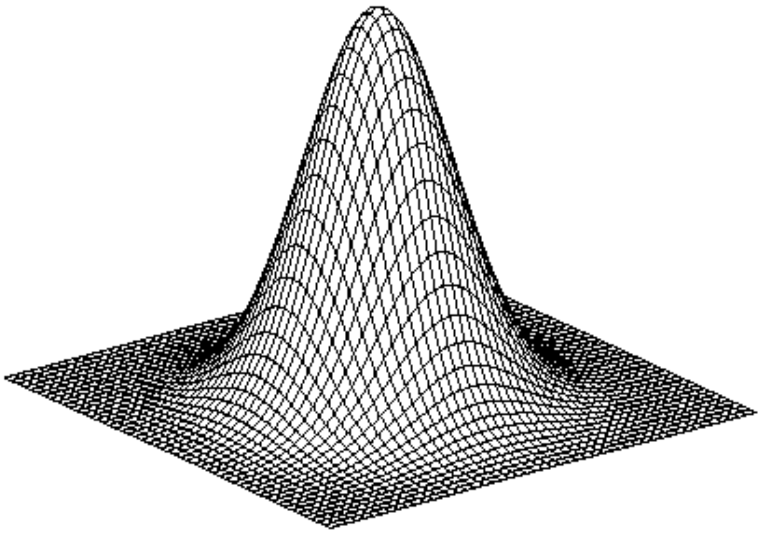
$\frac{d}{dx} g$

$f * \frac{d}{dx} g$

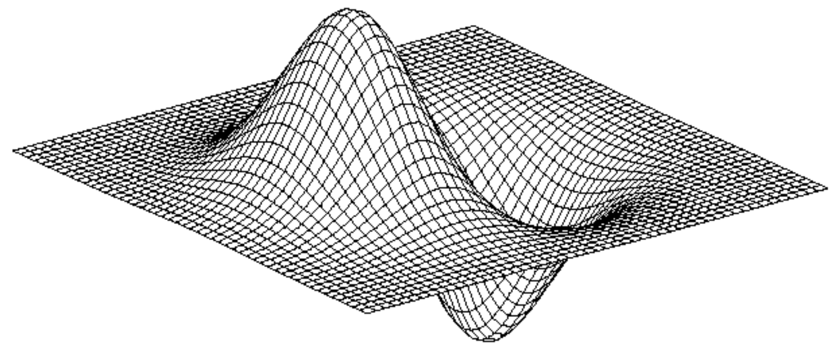


Source: S. Seitz

Derivative of 2D Gaussian filter



* [1 -1] =



Hays

Defining edges in 2D

- Rapid change in intensity

$$\boldsymbol{J}(\boldsymbol{x}) = \nabla I(\boldsymbol{x}) = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})(\boldsymbol{x}).$$

Gradient

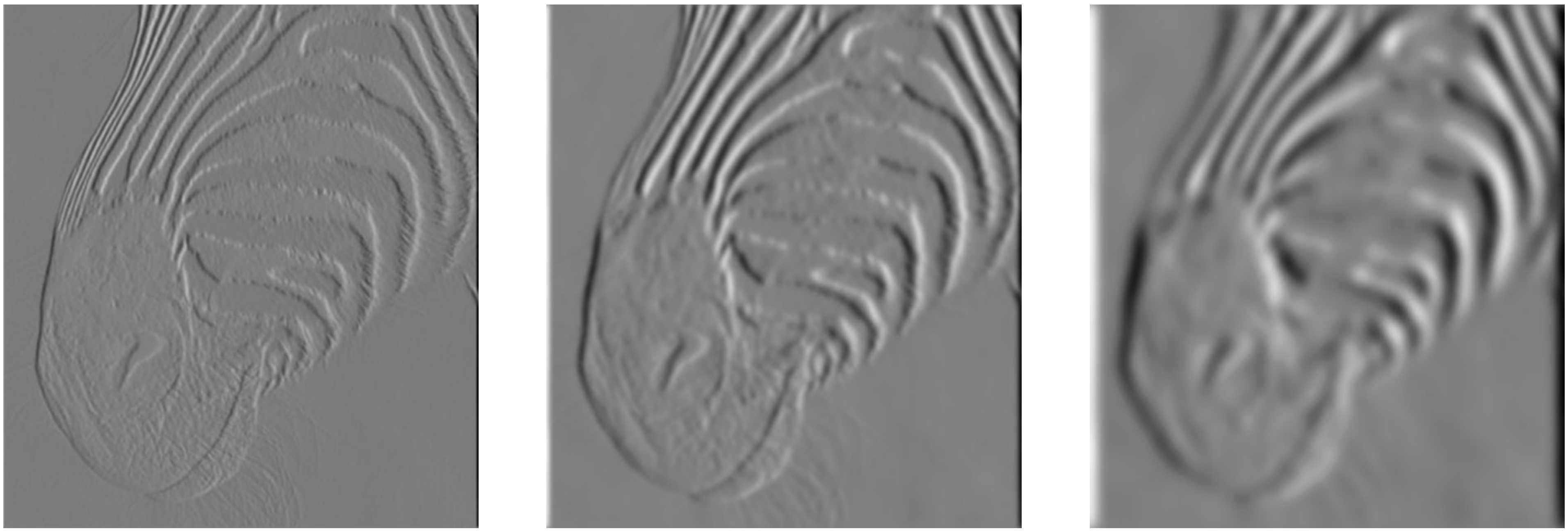
$$\boldsymbol{J}_\sigma(\boldsymbol{x}) = \nabla[G_\sigma(\boldsymbol{x}) * I(\boldsymbol{x})] = [\nabla G_\sigma](\boldsymbol{x}) * I(\boldsymbol{x}),$$

gradient of the smoothed image

$$\nabla G_\sigma(\boldsymbol{x}) = (\frac{\partial G_\sigma}{\partial x}, \frac{\partial G_\sigma}{\partial y})(\boldsymbol{x}) = [-x \quad -y]\frac{1}{\sigma^3}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

horizontal and vertical derivatives of the Gaussian kernel function

Tradeoff between smoothing and localization



1 pixel 3 pixels 7 pixels

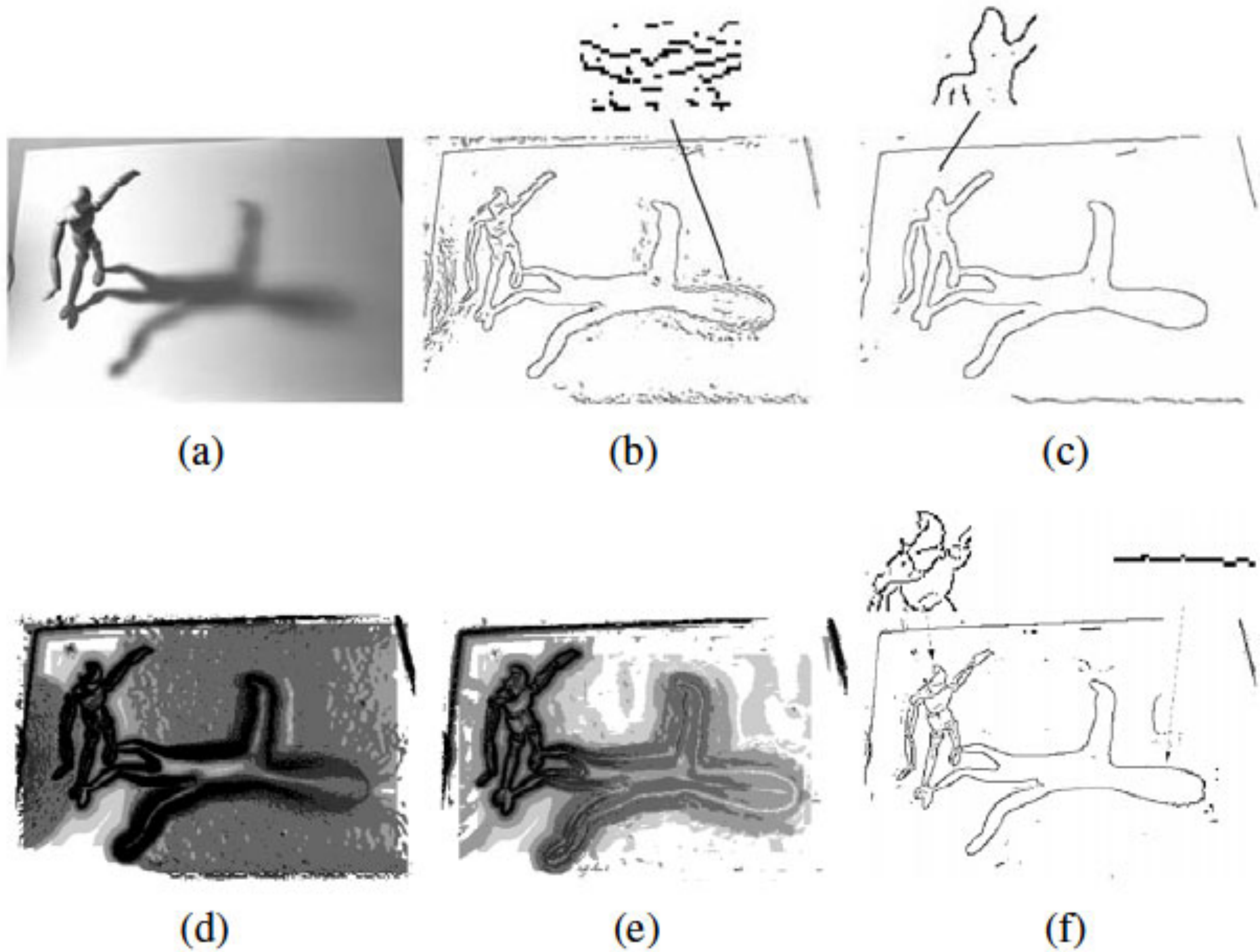
- Smoothed derivative removes noise, but blurs edge. Also finds edges at different “scales”.

Source: D. Forsyth

Scale selection and blur estimation

- How σ is determined ?
- Elder & Zucker (1998)
- Given a known image noise level, compute, for every pixel, the minimum scale at which an edge can be reliably detected

Scale selection and blur estimation



Elder – Are Edges Incomplete? 1999

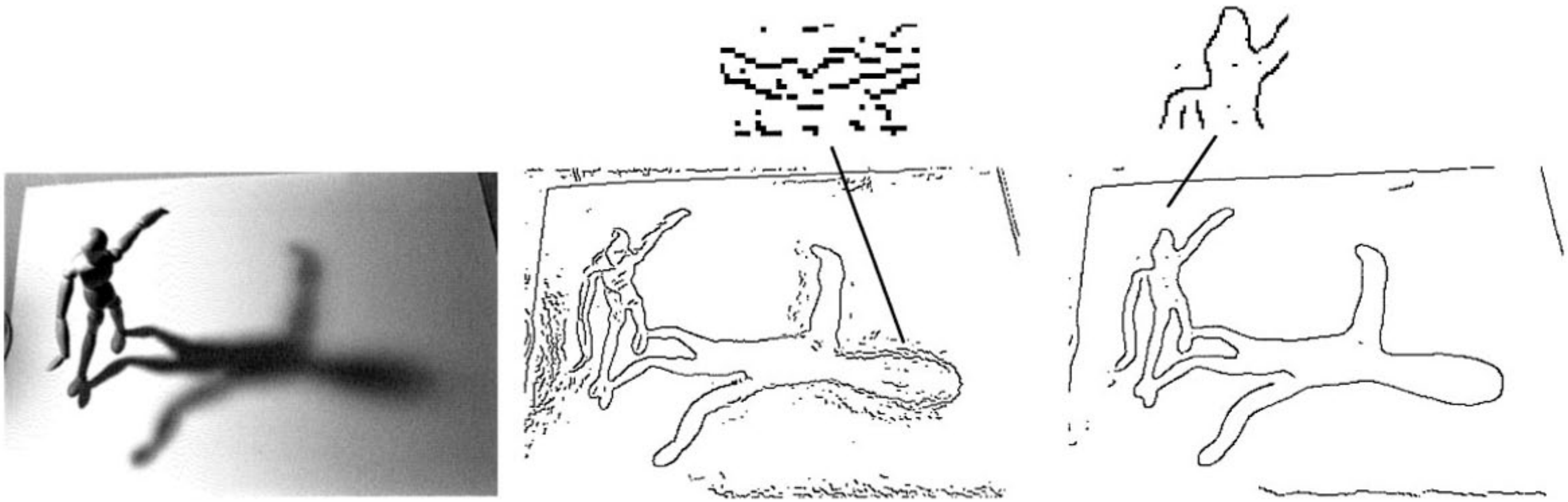


Figure 2. The problem of local estimation scale. Different structures in a natural image require different spatial scales for local estimation. The original image contains edges over a broad range of contrasts and blur scales. In the middle are shown the edges detected with a Canny/Deriche operator tuned to detect structure in the mannequin. On the right is shown the edges detected with a Canny/Deriche operator tuned to detect the smooth contour of the shadow. Parameters are ($\alpha = 1.25, \omega = 0.02$) and ($\alpha = 0.5, \omega = 0.02$), respectively. See (Deriche, 1987) for details of the Deriche detector.

What information would we need to
'invert' the edge detection process?

Elder – Are Edges Incomplete? 1999

Edge 'code':
- position,
- gradient magnitude,
- gradient direction,
- blur.

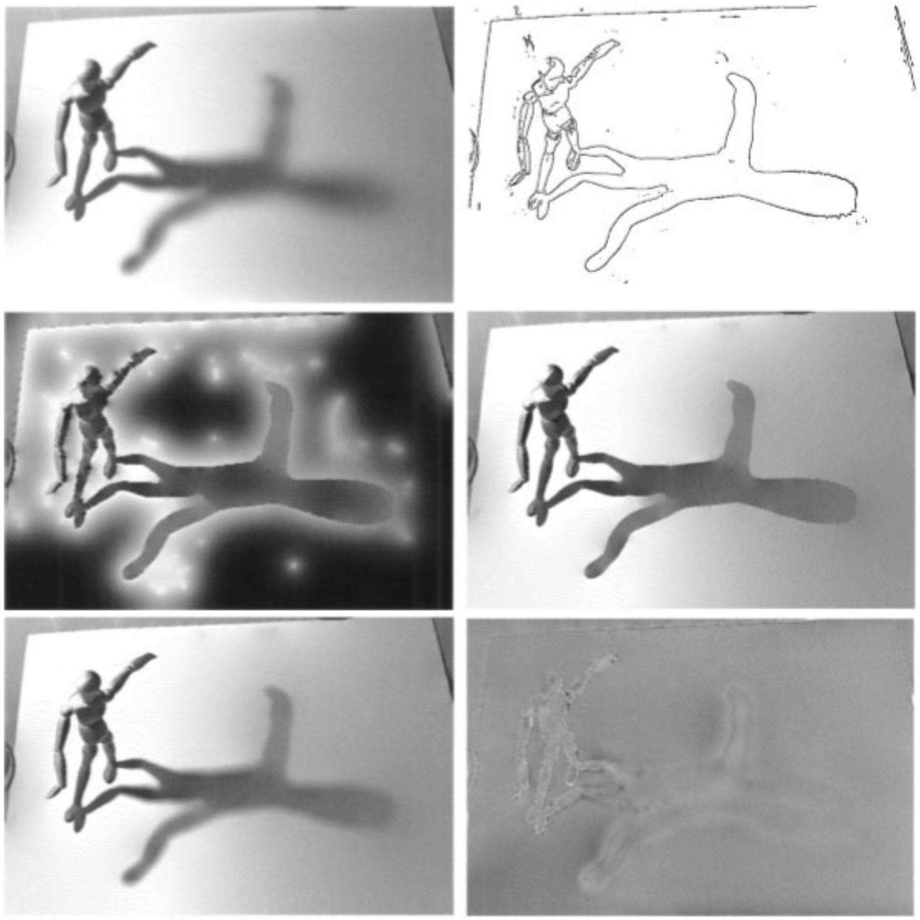


Figure 8. *Top left*: Original image. *Top right*: Detected edge locations. *Middle left*: Intermediate solution to the heat equation. *Middle right*: Reconstructed luminance function. *Bottom left*: Reblurred result. *Bottom right*: Error map (reblurred result—original). Bright indicates overestimation of intensity, dark indicates underestimation. Edge density is 1.7%. RMS error is 10.1 grey levels, with a 3.9 grey level DC component, and an estimated 1.6 grey levels due to noise removal.

Implementation issues

- The gradient magnitude is large along a thick "trail" or "ridge," so how do we identify the actual edge points?
- How do we link the edge points to form curves?

Source: D. Forsyth

Edge thining

- We wish to get single pixels at discrete locations along the edge contours.
- Can be done by looking for maxima in the edge strength (gradient magnitude) in a direction perpendicular to the edge orientation, i.e., along the gradient direction.
- Finding this maximum corresponds to taking a directional derivative of the strength field in the direction of the gradient and then looking for zero crossings.

$$S_\sigma(\boldsymbol{x}) = \nabla \cdot \boldsymbol{J}_\sigma(\boldsymbol{x}) = [\nabla^2 G_\sigma](\boldsymbol{x}) * I(\boldsymbol{x})].$$

Laplacian

Laplacian of Gaussian (LoG) kernel (Marr and Hildreth 1980).

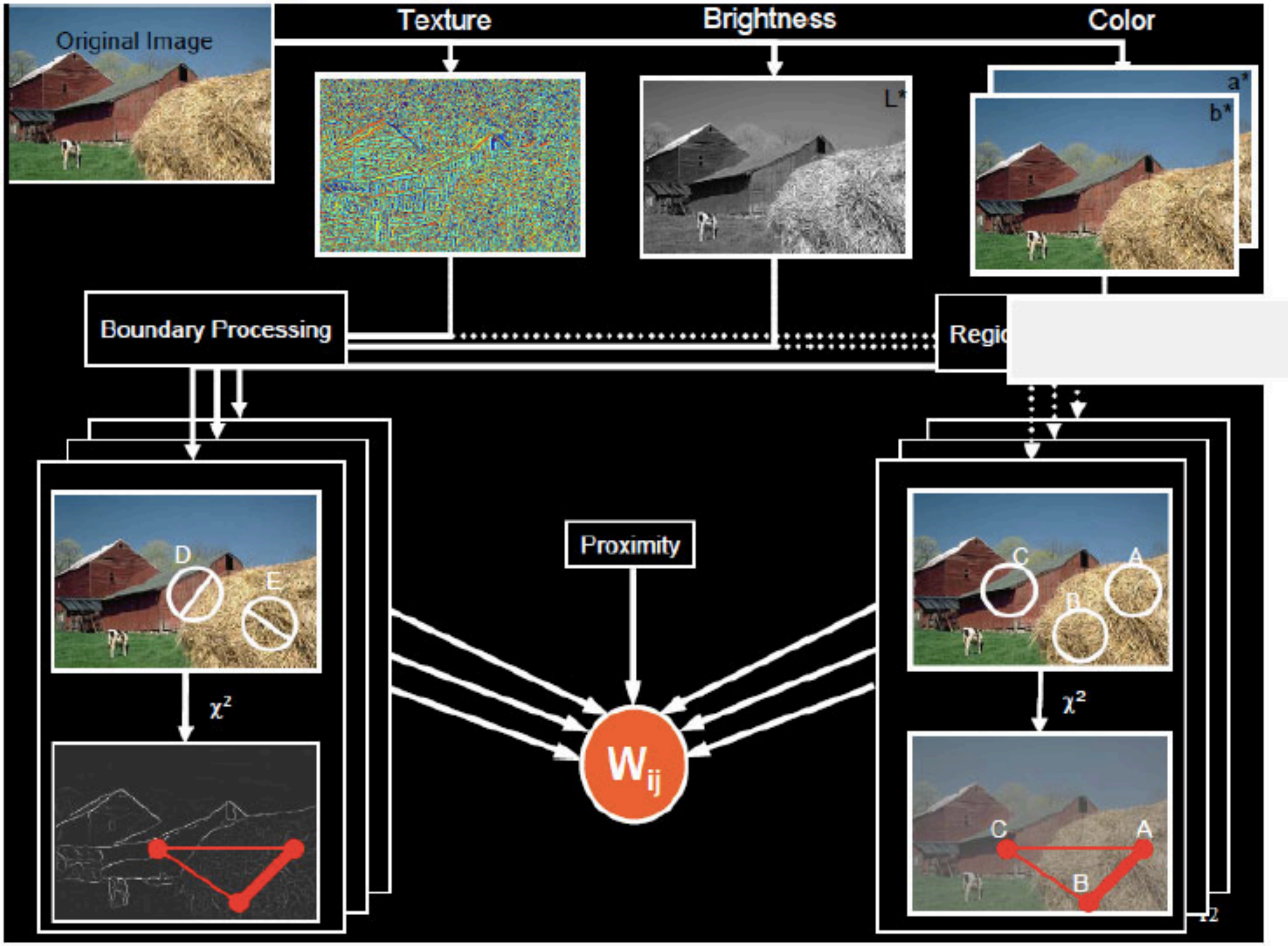
$$\nabla^2 G_\sigma(x) = \frac{1}{\sigma^3}\left(2 - \frac{x^2+y^2}{2\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

This kernel can be split into two separable parts:

$$\nabla^2 G_\sigma(x) = \frac{1}{\sigma^3}\left(1 - \frac{x^2}{2\sigma^2}\right)G_\sigma(x)G_\sigma(y) + \frac{1}{\sigma^3}\left(1 - \frac{y^2}{2\sigma^2}\right)G_\sigma(y)G_\sigma(x)$$

LoG → Difference of Gaussian (DoG) computation

Combining edge feature cues



Combined brightness, color, texture boundary detector (Martin, Fowlkes, and Malik 2004)

Brightness

Color

Texture

Combined

Human

Boundary Detector

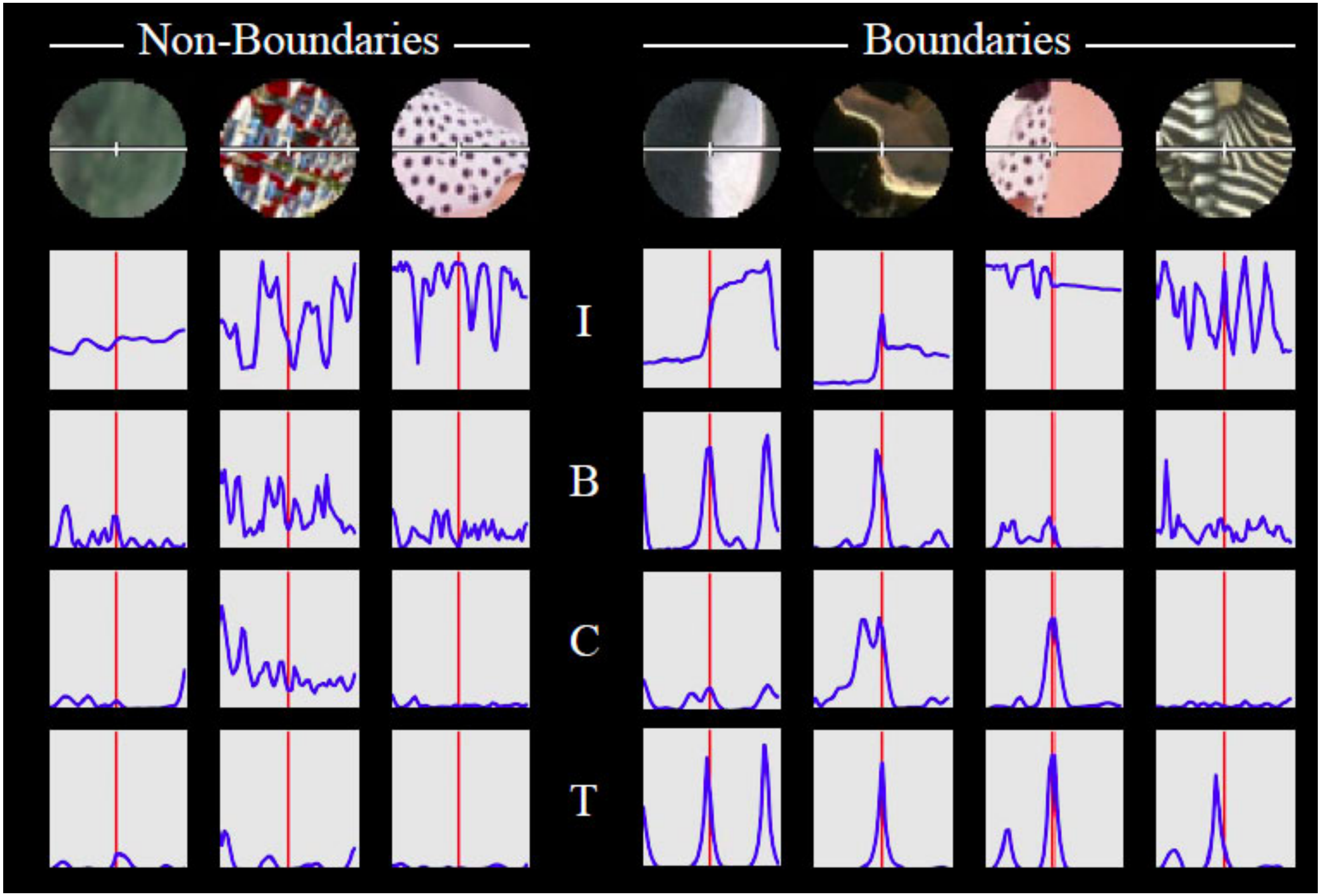
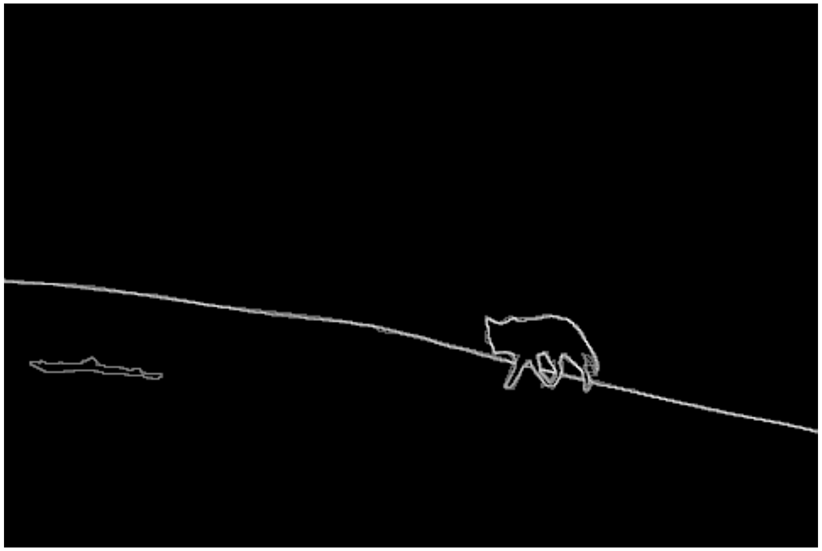


Figure from Fowlkes

Results

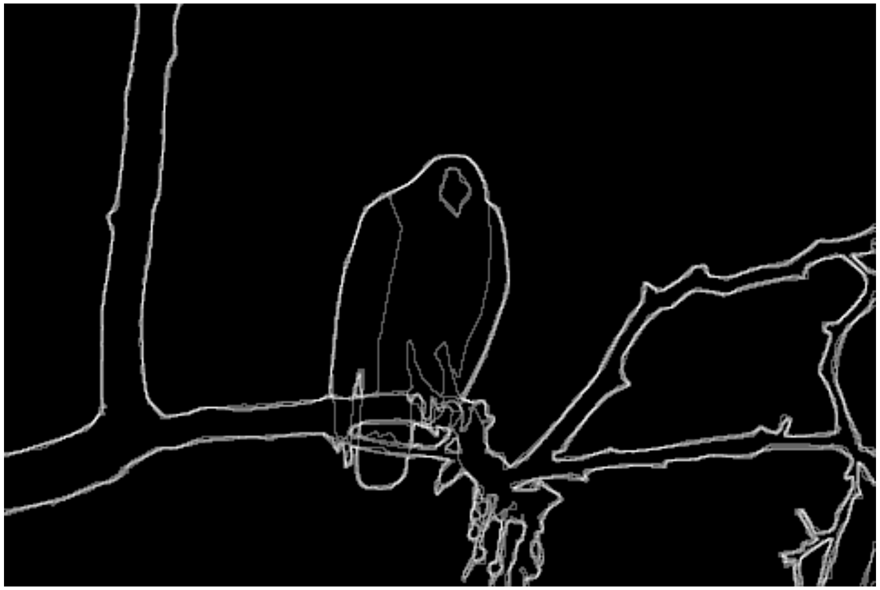
Automatic



Human

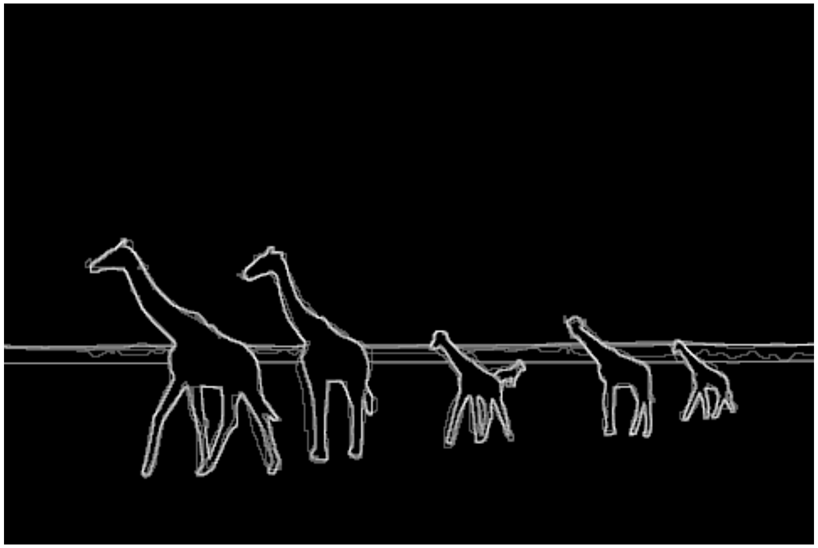
Results

Automatic

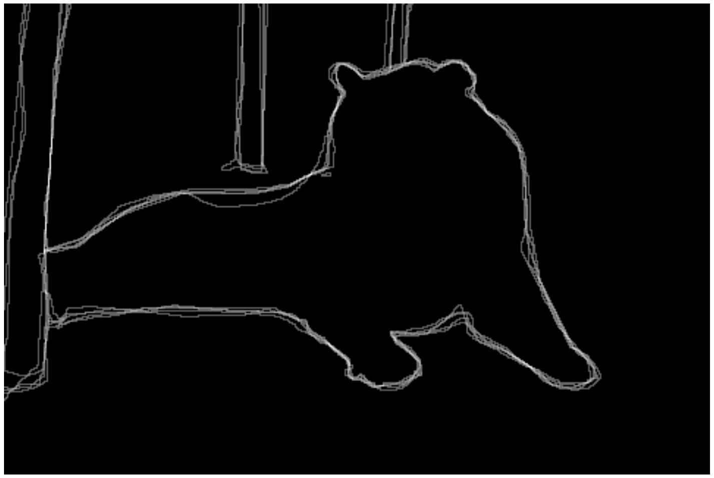


Human

Results



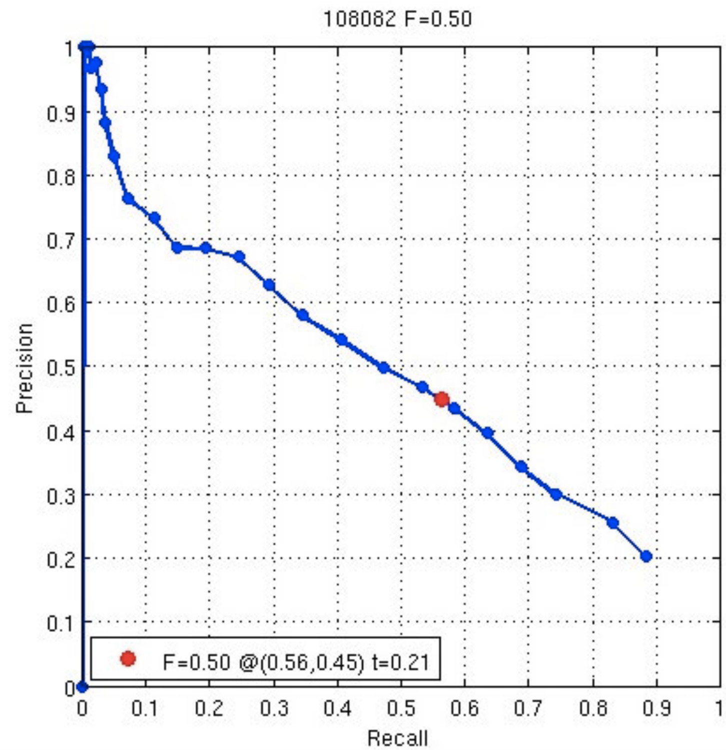
Results



For more:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/108082-color.html

Scoring Edge Detectors

Ren et al.
NIPS2012
(color)
(0.50)

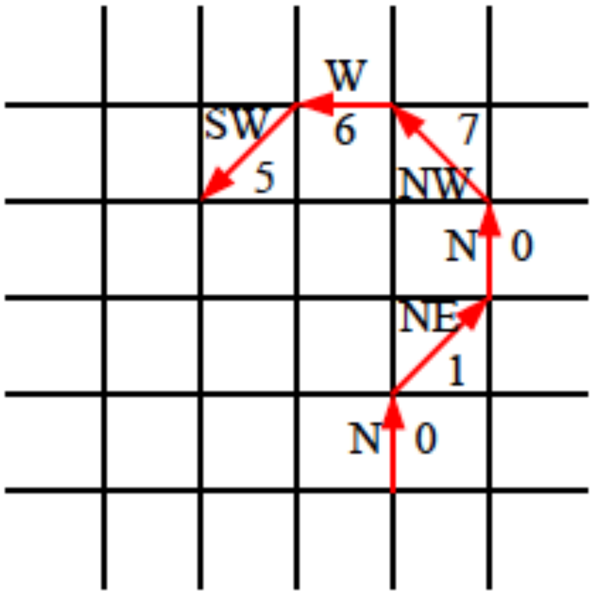


Precision is the probability that a machine-generated boundary pixel is a true boundary pixel. Recall is the probability that a true boundary pixel is detected.

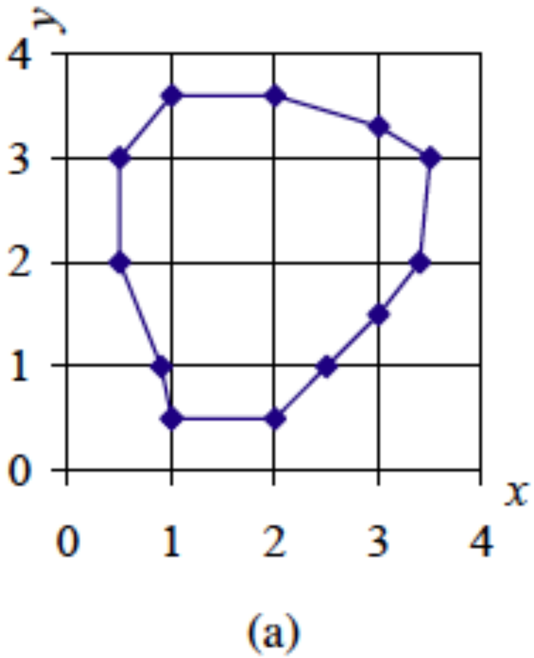
The traditional F-measure or balanced F-score (**F_1 score**) is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Edge Linking

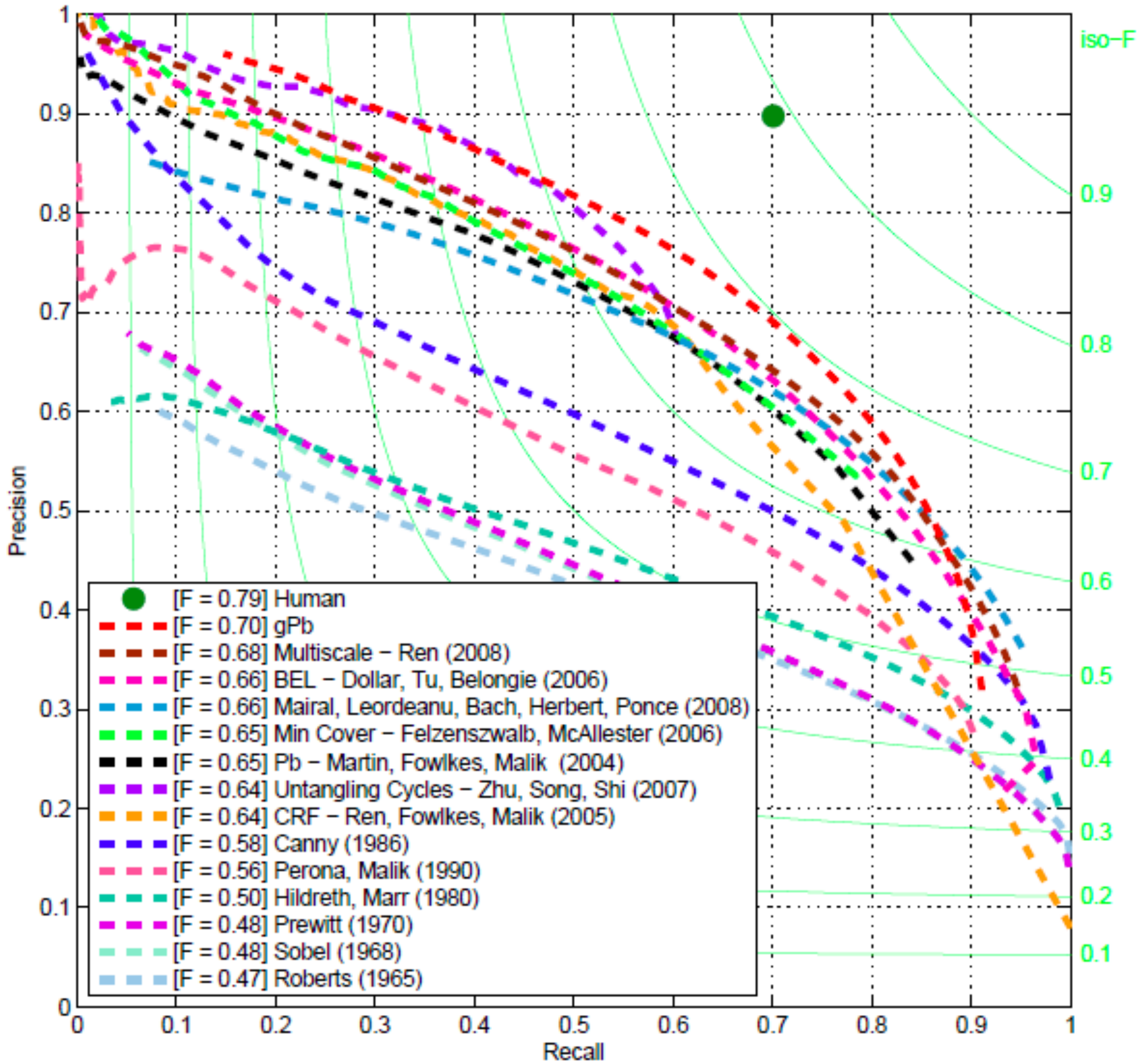


Chain code



Arc length parameterization

45 years of boundary detection



, and Malik. TPAMI 2011 (pdf)

State of edge detection

- Local edge detection works well
  - ‘False positives’ from illumination and texture edges (depends on our application).
- Some methods take into account longer contours
- Modern methods that actually “learn” from data.
- Poor use of object and high-level information.

Hays

Canny edge detector

- Probably the most widely used edge detector in computer vision.

- Theoretical model: step-edges corrupted by additive Gaussian noise.
- Canny showed that first derivative of Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization.

J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

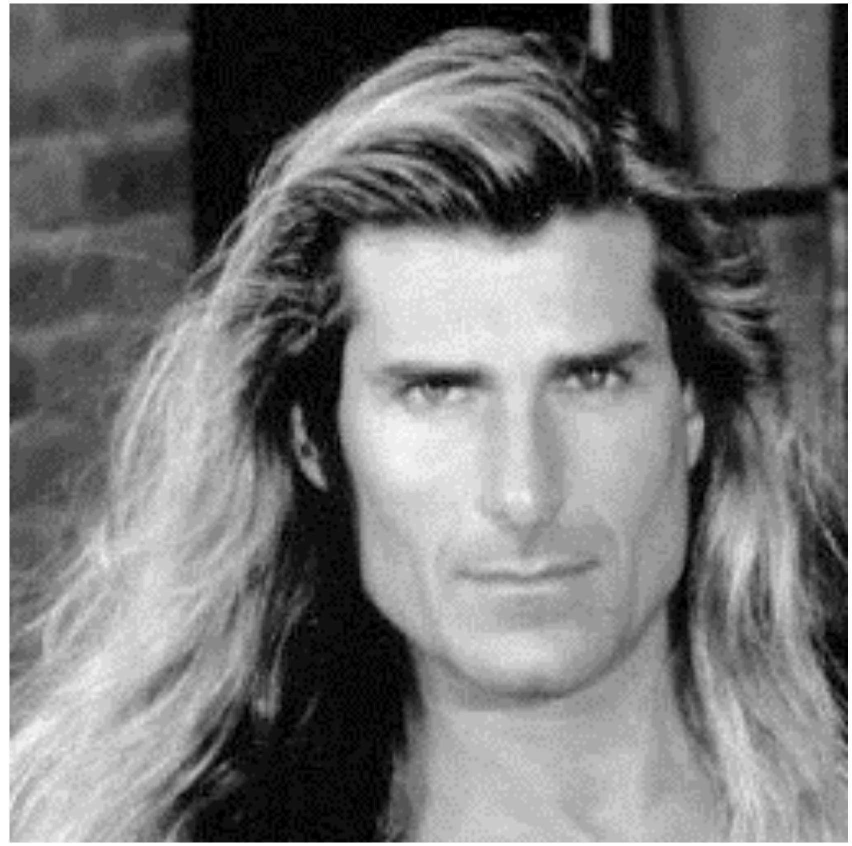
22,000 citations!

L. Fei-Fei

Examples: Controversy and Appropriateness

‘Lena’

Alexander Sawchuk @ USC, 1973



‘Fabio’

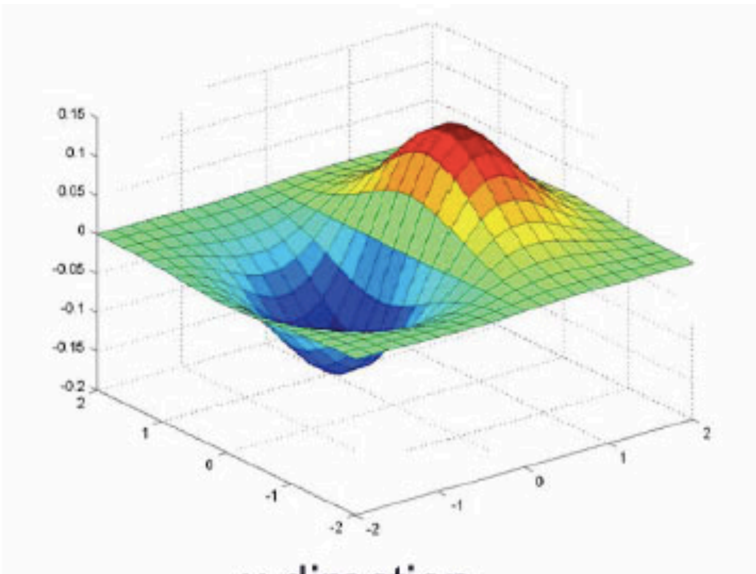
Deanna Needell @ Claremont McKenna, 2012

Canny edge detector

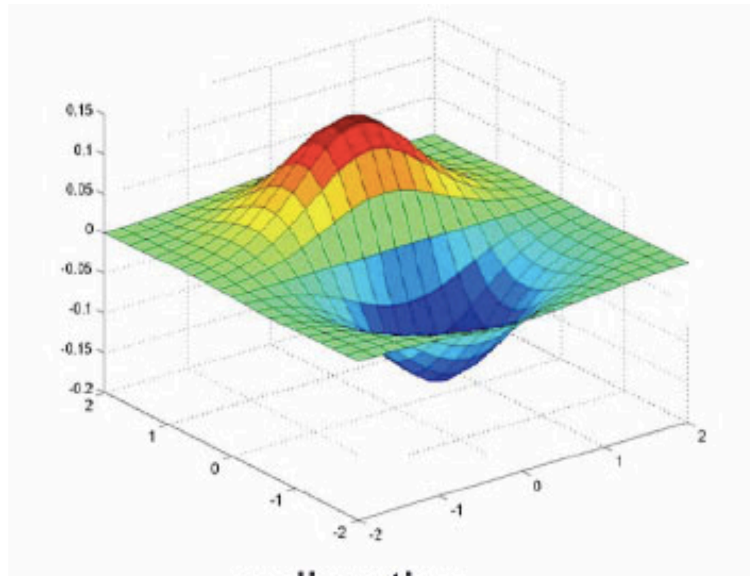
1. Filter image with x, y derivatives of Gaussian

Source: D. Lowe, L. Fei-Fei

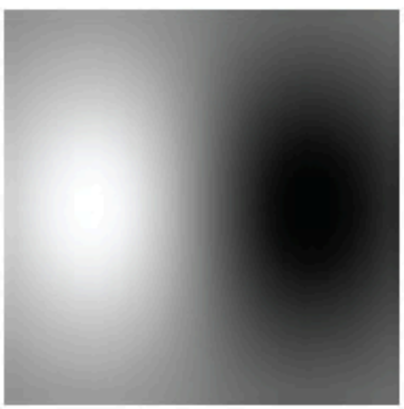
Derivative of Gaussian filter

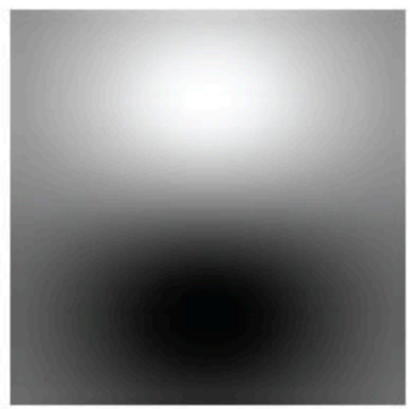


x-direction

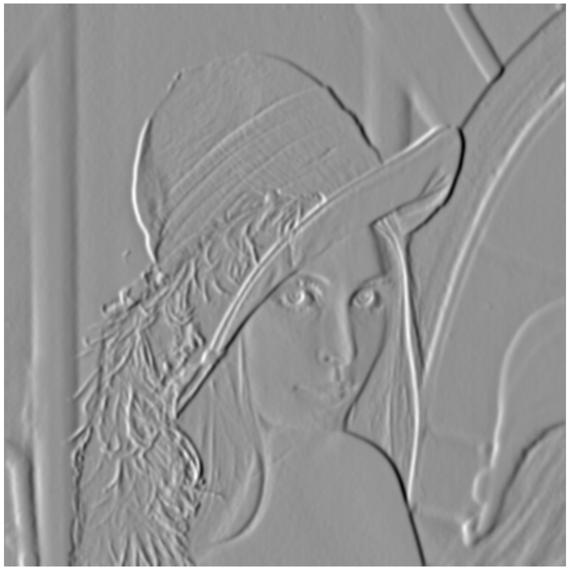


y-direction





Compute Gradients



X-Derivative of Gaussian

What's the difference?

Y-Derivative of Gaussian

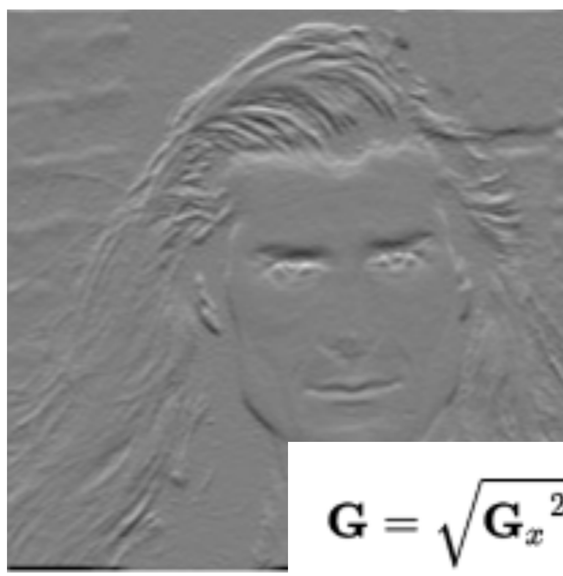
Canny edge detector

1. Filter image with x, y derivatives of Gaussian
2. Find magnitude and orientation of gradient

Source: D. Lowe, L. Fei-Fei

Compute Gradient Magnitude

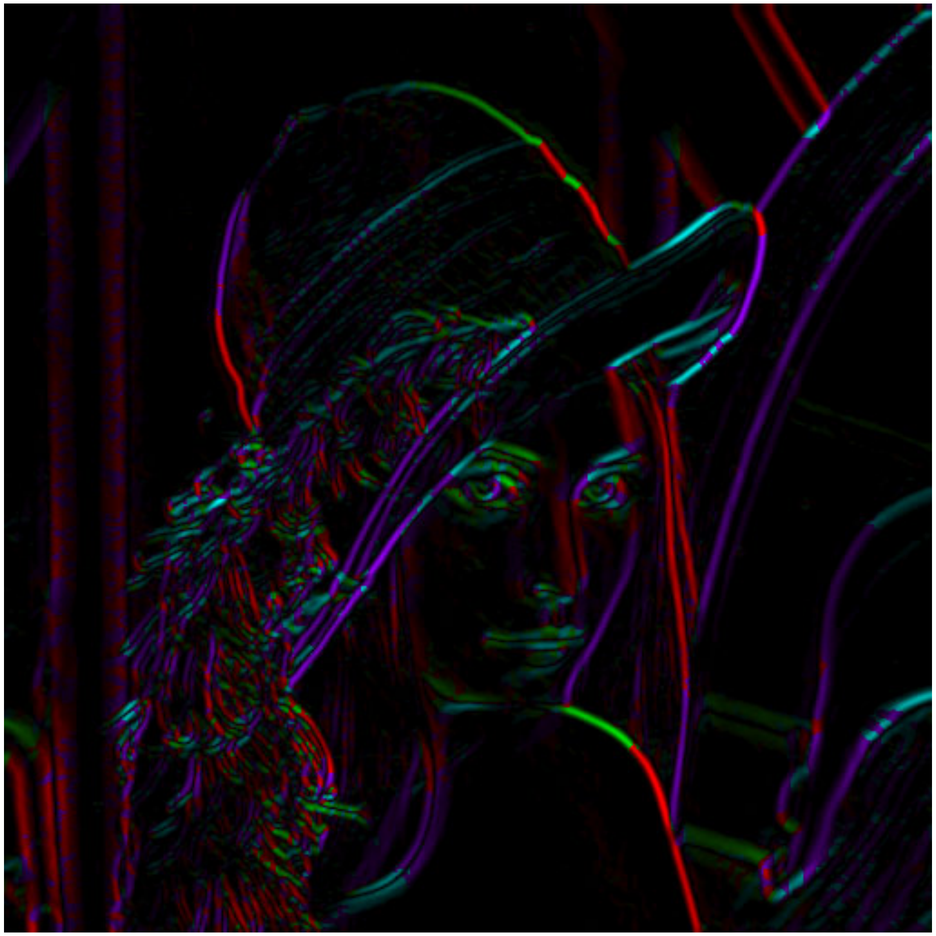
sqrt(X-Deriv.ofGaussian ^2 + Y-Deriv.ofGaussian ^2) = gradient magnitude



$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

Compute Gradient Orientation

- Threshold magnitude at minimum level
- Get orientation via   theta = atan2(gy, gx)

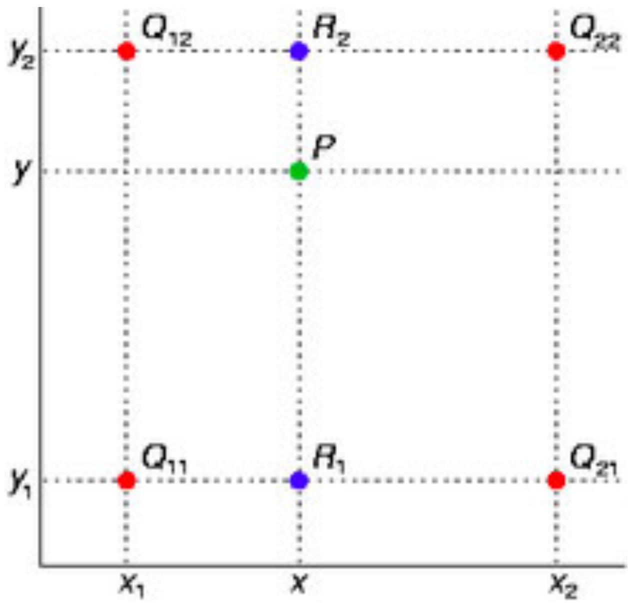


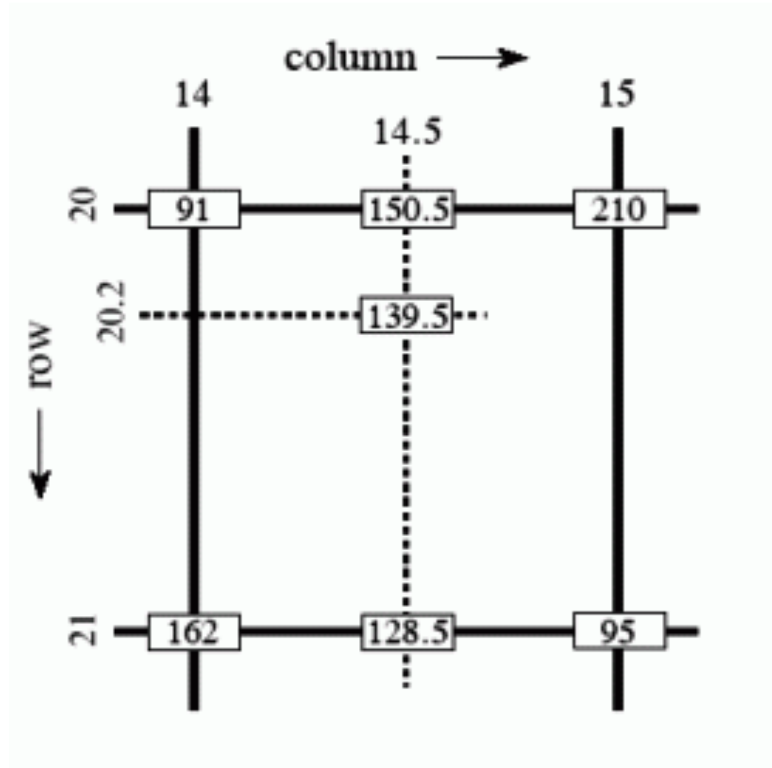
Canny edge detector

1. Filter image with x, y derivatives of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin multi-pixel wide “ridges” to single pixel width

Source: D. Lowe, L. Fei-Fei

Sidebar: Bilinear Interpolation





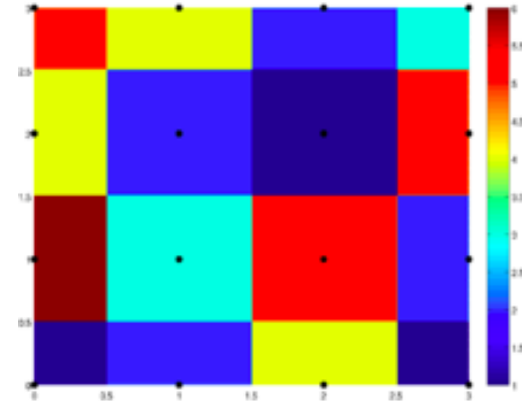
$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}.$$

http://en.wikipedia.org/wiki/Bilinear_interpolation

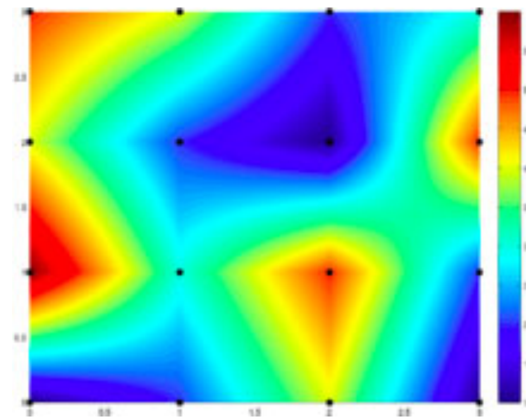
Sidebar: Interpolation options

- imx2 = imresize(im, 2, interpolation_type)

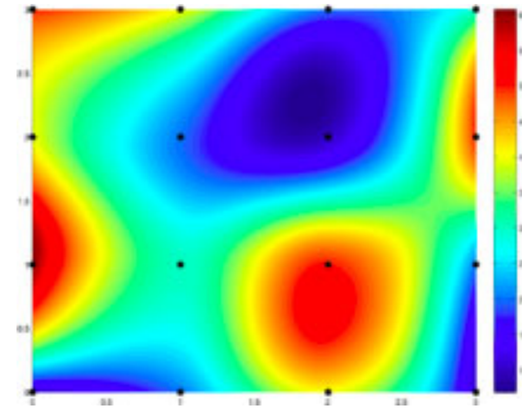
- 'nearest'
  - Copy value from nearest known
  - Very fast but creates blocky edges



- 'bilinear'
  - Weighted average from four nearest known pixels
  - Fast and reasonable results



- 'bicubic' (default)
  - Non-linear smoothing over larger area (4x4)
  - Slower, visually appealing, may create negative pixel values



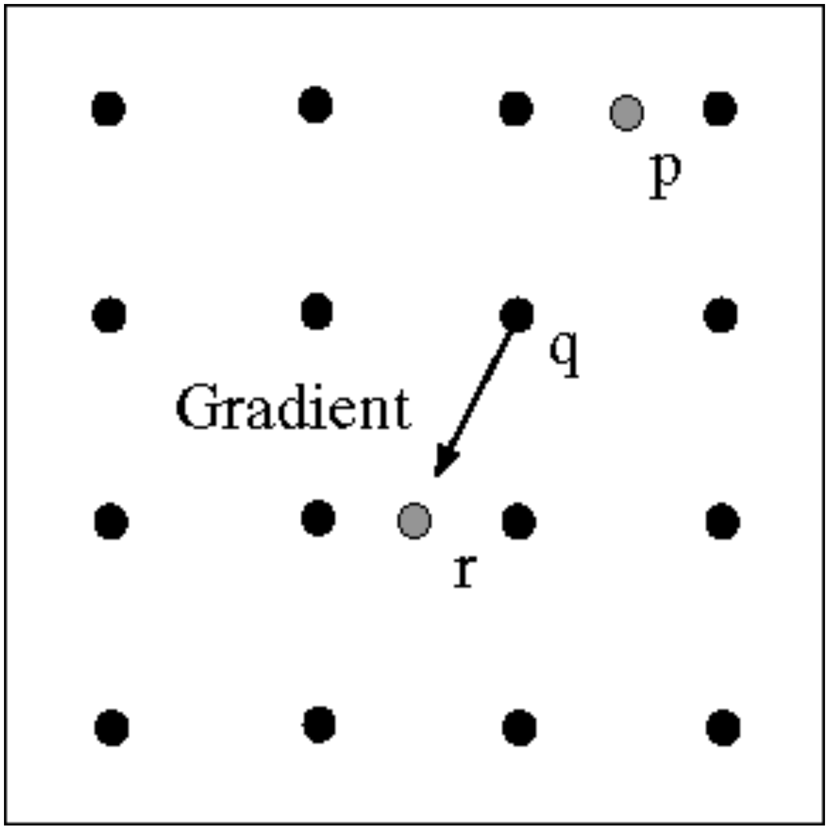
Examples from http://en.wikipedia.org/wiki/Bicubic_interpolation

Non-maximum suppression

Non-maximum supression is often used along with edge detection algorithms.
The image is scanned along the image gradient direction, and if pixels are not part of the local maxima they are set to zero.
This has the effect of supressing all image information that is not part of local maxima.

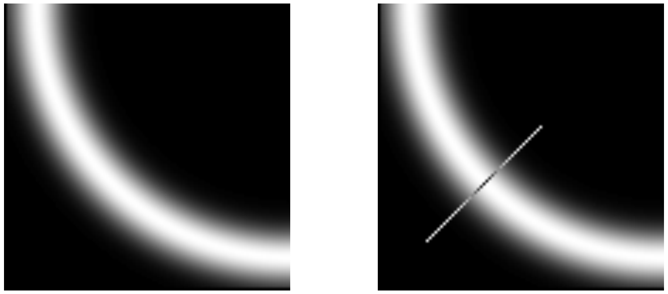
Source: D. Forsyth

Non-maximum suppression for each orientation



At pixel q:
We have a maximum if the value is larger than those at both p and at r.

Interpolate along gradient direction to get these values.



Source: D. Forsyth

Before Non-max Suppression

Gradient magnitude

James Hays

After non-max suppression

Gradient magnitude

James Hays

Canny edge detector

1. Filter image with x, y derivatives of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin multi-pixel wide “ridges” to single pixel width
4. ‘Hysteresis’ Thresholding:
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them
   - ‘Follow’ edges starting from strong edge pixels
     - Connected components (Szeliski 3.3.4)

Source: D. Lowe, L. Fei-Fei

‘Hysteresis’ thresholding

- Two thresholds – high and low
- Grad. mag. > high threshold? = strong edge
- Grad. mag. < low threshold? noise
- In between = weak edge

- ‘Follow’ edges starting from strong edge pixels
- Continue them into weak edges
    - Connected components (Szeliski 3.3.4)

Source: S. Seitz

Hysteresis thresholding

- Threshold at low/high levels to get weak/strong edge pixels
- ‘Follow’ edges starting from strong edge pixels
  - Connected components

Final Canny Edges

Effect of σ (Gaussian kernel spread/size)

Original

Canny with $\sigma = 1$

Canny with $\sigma = 2$

The choice of σ depends on desired behavior

- large σ detects large scale edges
- small σ detects fine features

Source: S. Seitz

Canny edge detector

1. Filter image with x, y derivatives of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin multi-pixel wide “ridges” to single pixel width
4. ‘Hysteresis’ Thresholding:
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them
   - ‘Follow’ edges starting from strong edge pixels
     - Connected components (Szeliski 3.3.4)

- MATLAB: edge(image, ‘canny’)

Source: D. Lowe, L. Fei-Fei

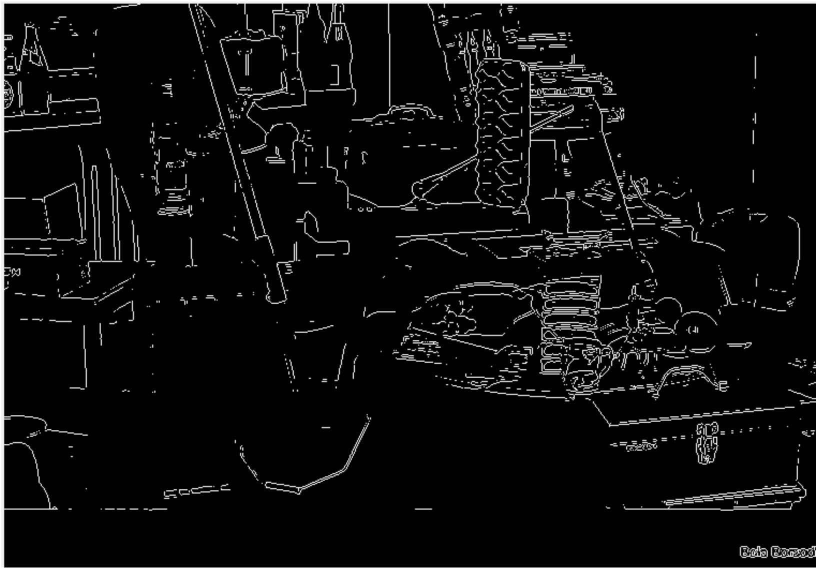
edge() in Matlab

- BW = edge(I,method,threshold,direction,'nothinning')
- BW = edge(I,method,threshold,direction,sigma)
-

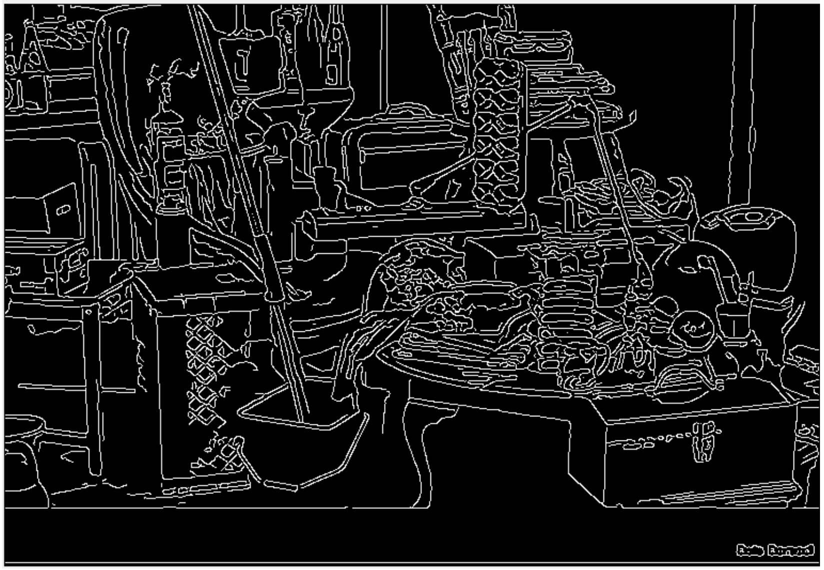
Method
'Canny'
'log' (Laplacian of Gaussian)
'Prewitt'
'Roberts'
'Sobel'
'zerocross'

edge() in Matlab

Prewitt



Canny



Next Class: Hough transform