Low-level Vision Processing Algorithms

Speaker: Ito, Dang

Supporter: Ishii, Toyama and Y. Murakami

Adaptive Systems Lab

The University of Aizu

会津大学

THE UNIVERSITY OF AIZU

Overview

- Introduction
  - What is Vision Processing?
- Basic Knowledge
  - Image formation
  - Transformation
- Low level Algorithm
  - Filtering
  - Edge/Border Detection
  - Conner Detection
- Conclusion

Overview

会津大学
THE UNIVERSITY OF AIZU

- Introduction
  - What is Vision Processing?
- Basic Knowledge
  - Image formation
  - Transformation
- Low level Algorithm
  - Filtering
  - Edge/Border Detection
  - Conner Detection
- Conclusion

What is Vision Processing?

- Make computers understand images and video:
  - Images to Models

What kind of scene?

Where are the cars?

How far is the building?

…

Slide credit James Hays

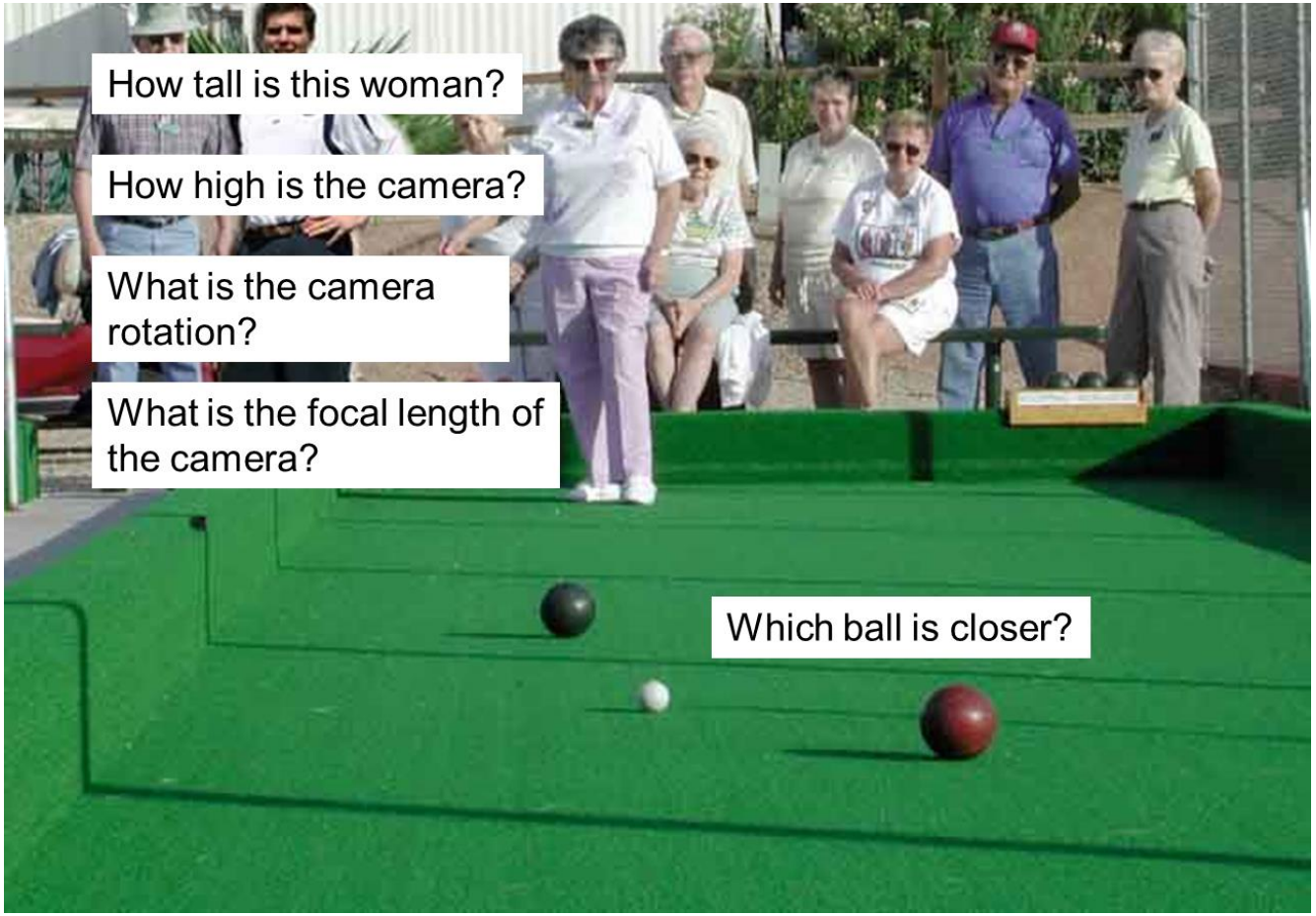
Application of Vision Processing

- Optical Character Recognition
- Face detection
- Object recognition
- Shape/motion capture
- Medical Imaging

Why we need?

会津大学
THE UNIVERSITY OF AIZU

- Human limitations
  - measurement accuracy, darkness, etc…



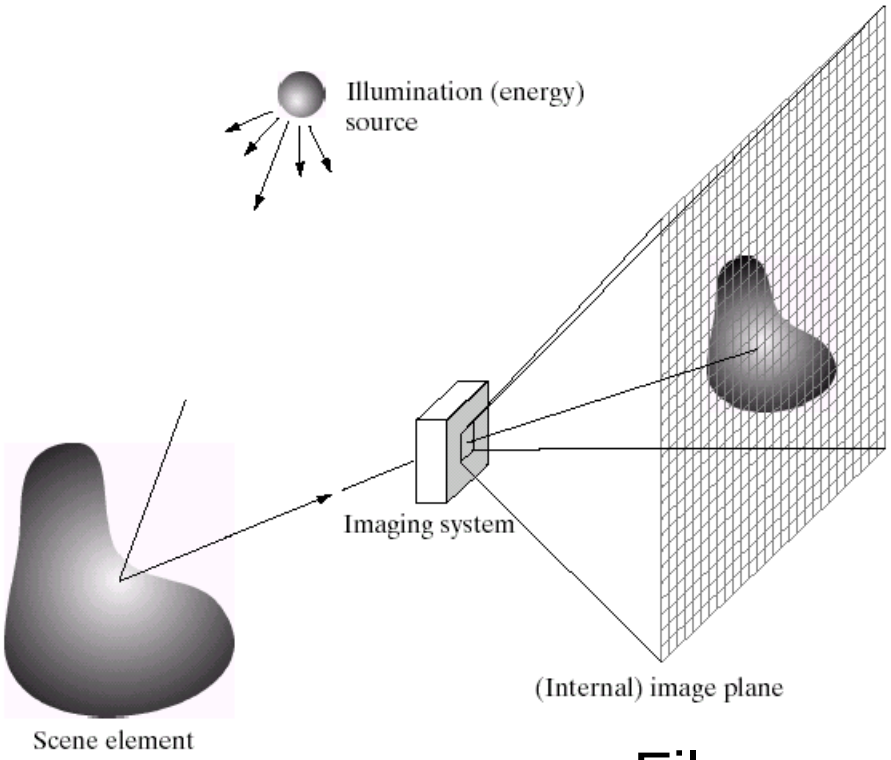
Overview

- Introduction
  - What is Vision Processing?
- Basic Knowledge
  - Image formation
  - Transformation
- Low level Algorithms
  - Filtering
  - Border Detection
  - Edge Detection
- Conclusion

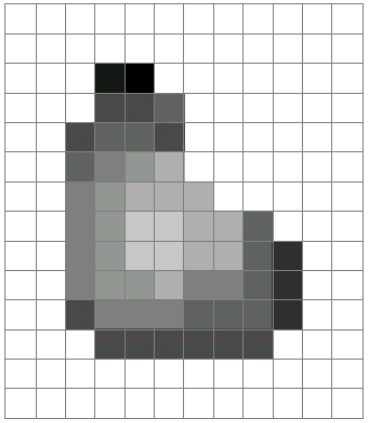
Basic Knowledge

- Image and Color.
- Camera, Lens.
- Color representation.
- Fourier Transform.

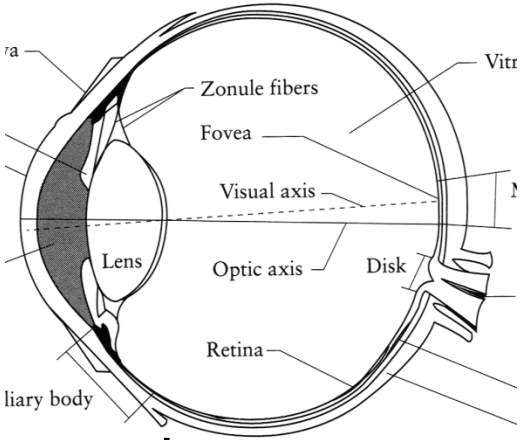
Image Formation



Film



Digital Camera

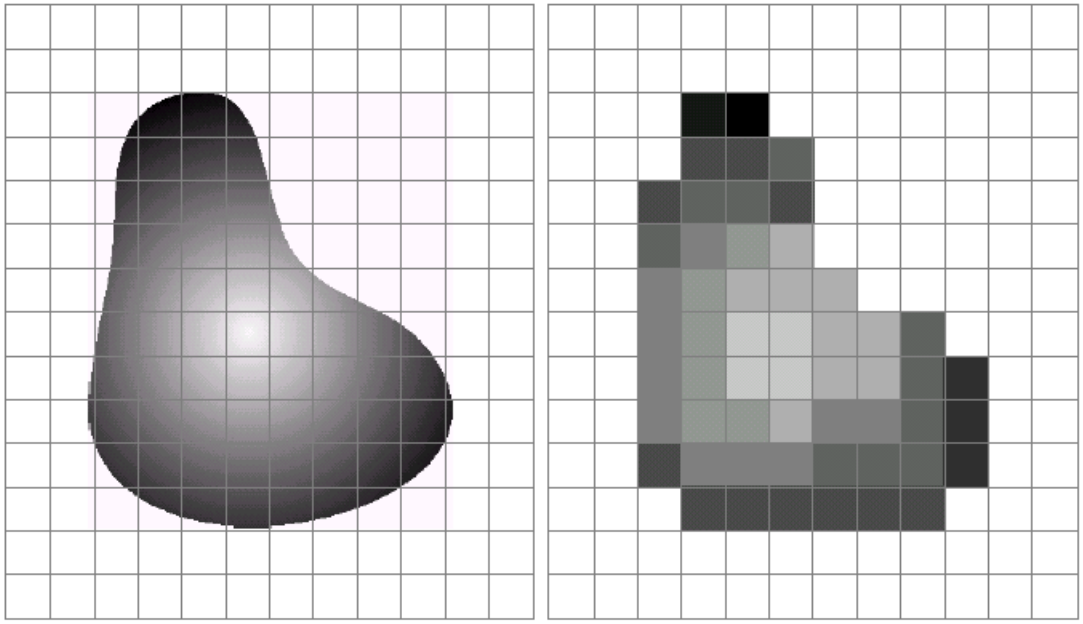


The Eye

Digital camera

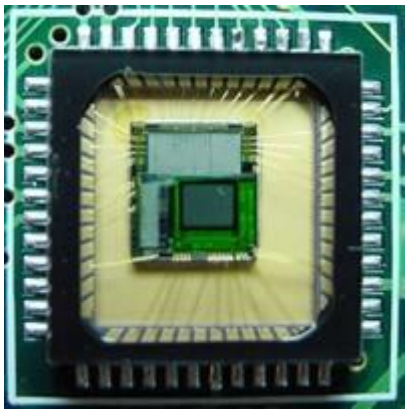
- A digital camera replaces film with a sensor array
  - Each cell in the array is light-sensitive diode that converts photons to electrons
  - Two common types
    - Charge Coupled Device (CCD)
    - CMOS
  - http://electronics.howstuffworks.com/digital-camera.htm

Sensor Array



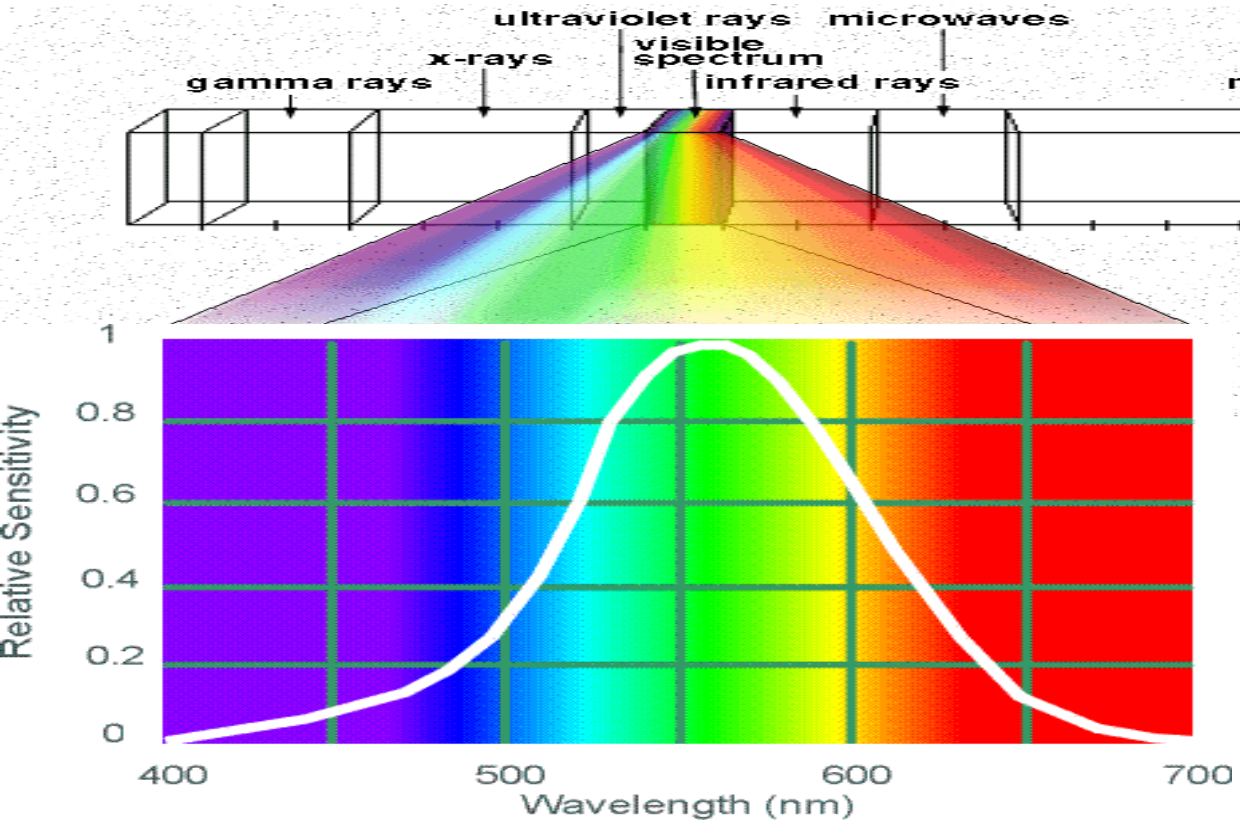
a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.



CMOS sensor

Electromagnetic Spectrum

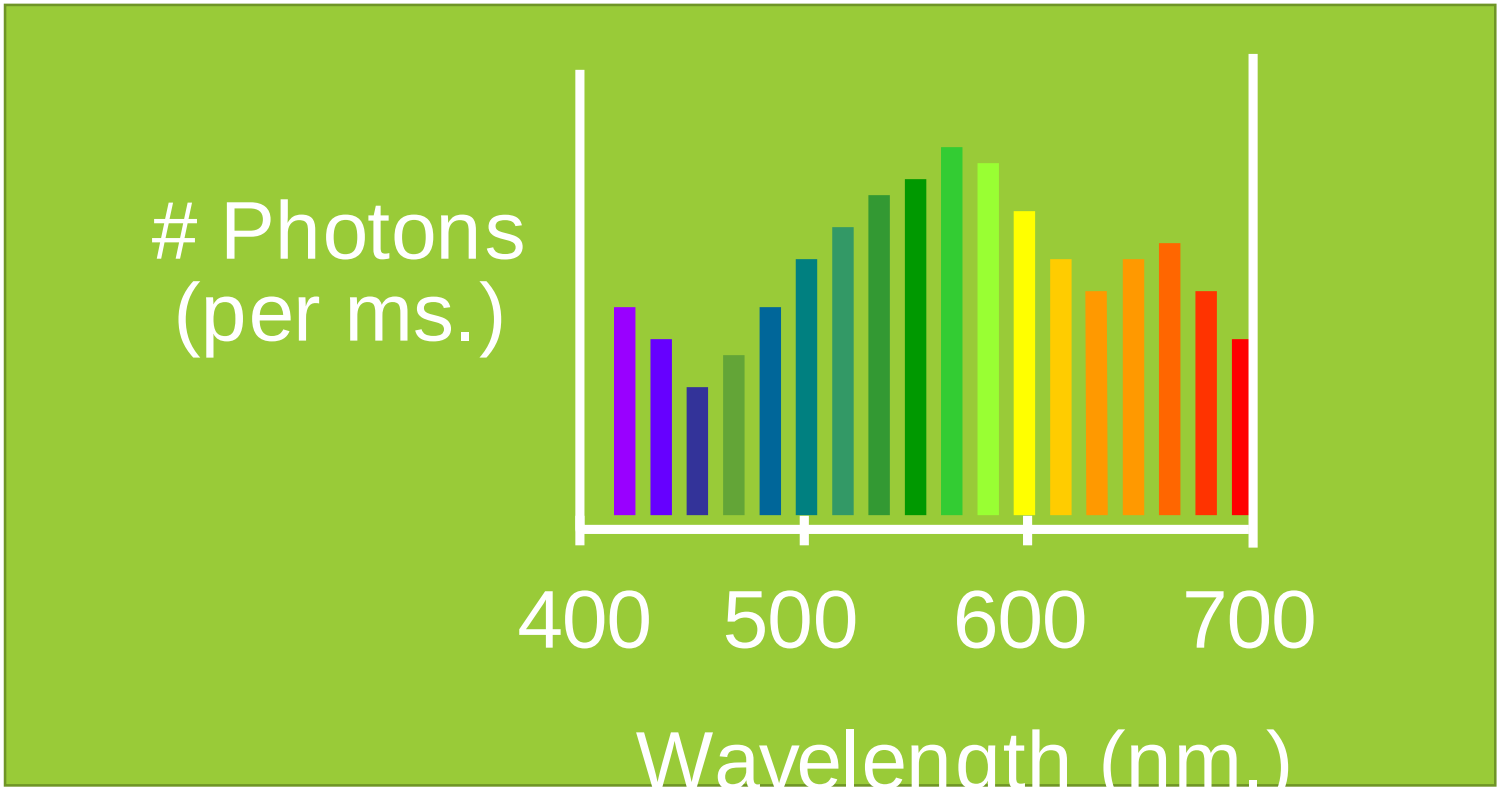


Human Luminance Sensitivity Function

http://www.yorku.ca/eye/photopik.htm

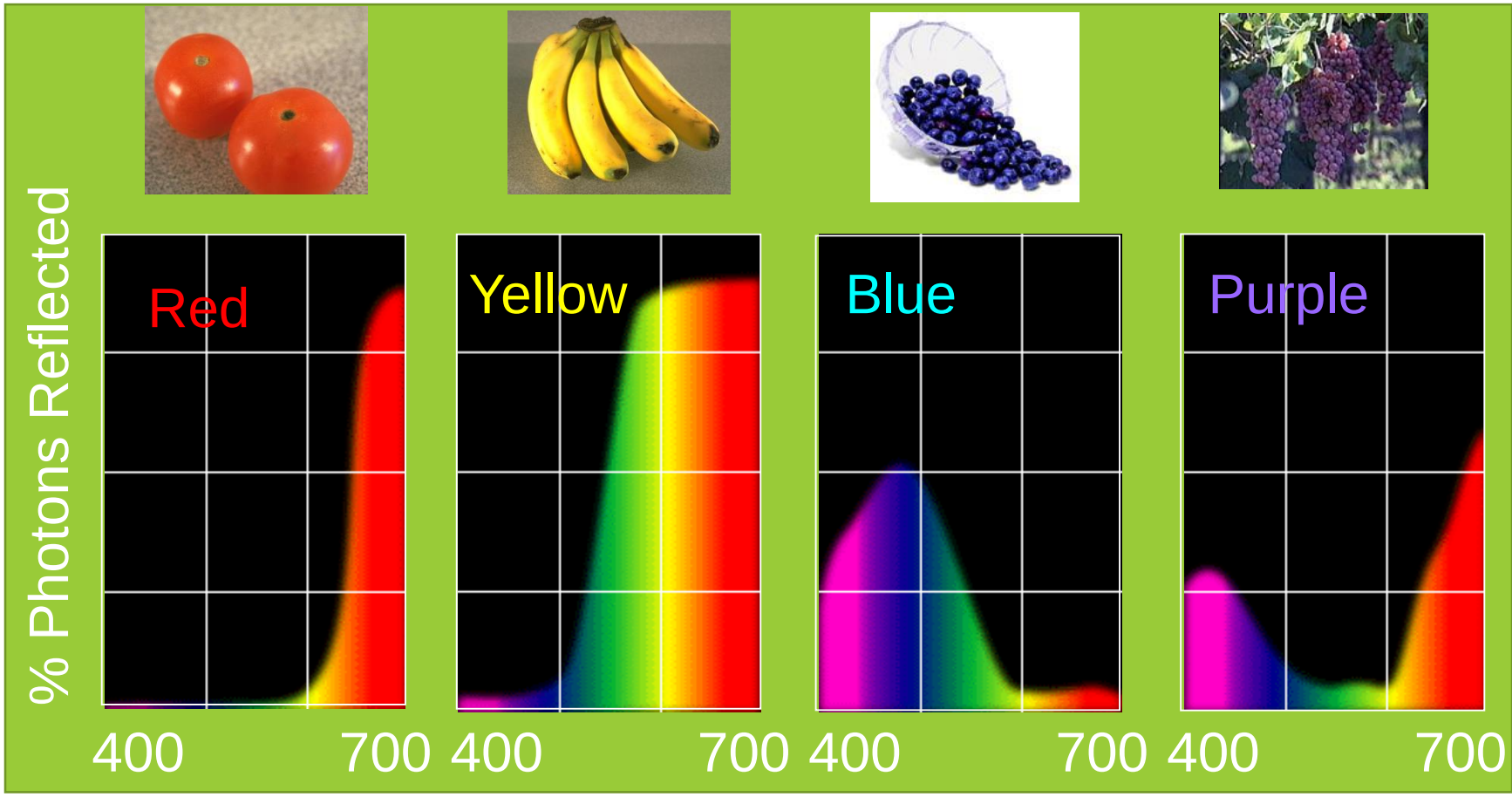
The Physics of Light

Any patch of light can be completely described physically by its spectrum: the number of photons (per time unit) at each wavelength 400 - 700 nm.

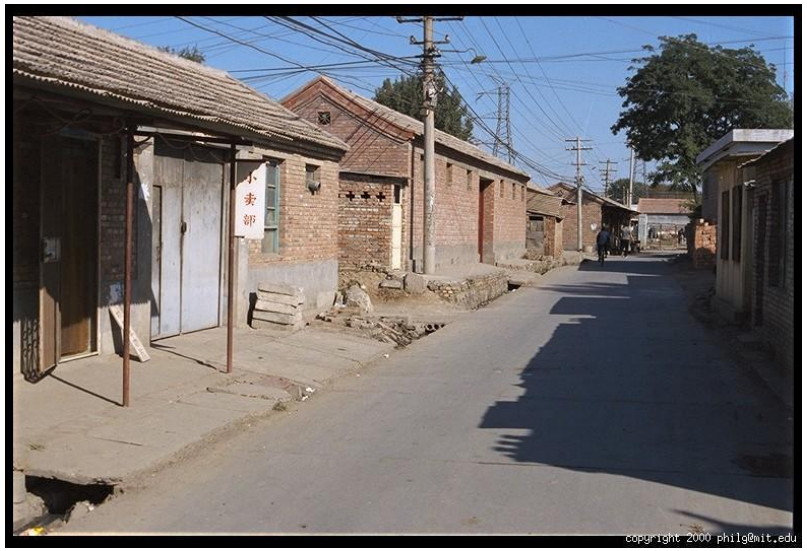


The Physics of Light

Some examples of the reflectance spectra of surfaces



Color Image

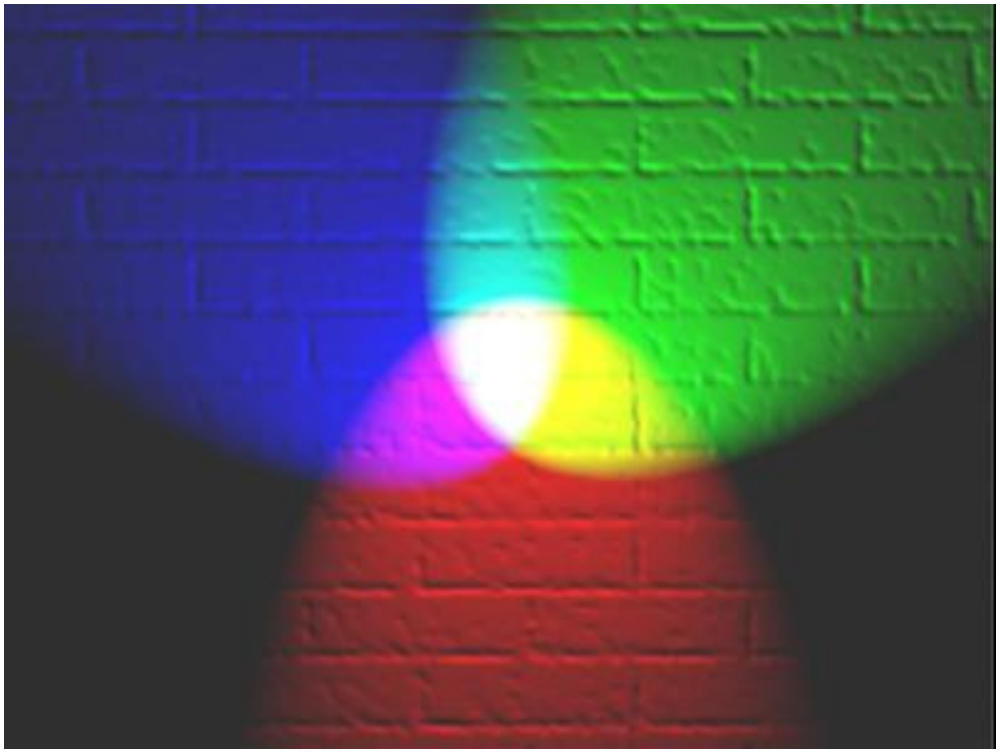

copyright 2000 philg@mit.edu


R
G
B
copyright 2000 philg@mit.edu

Color spaces

会津大学
THE UNIVERSITY OF AIZU

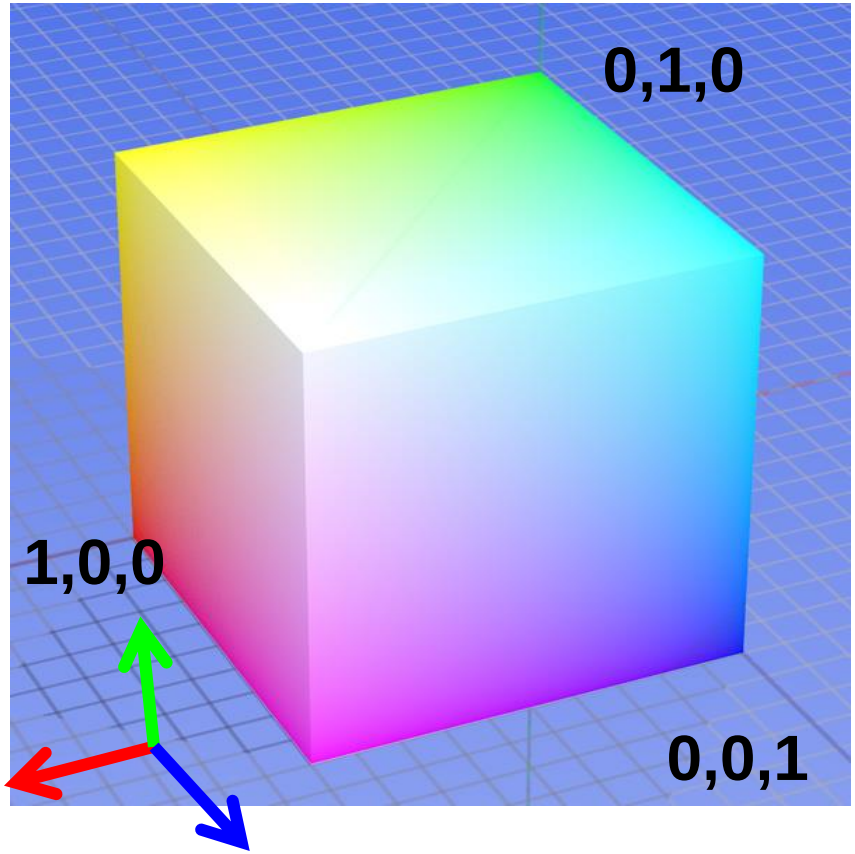
- How can we represent color?



http://en.wikipedia.org/wiki/File:RGB_illumination.jpg

Color spaces: RGB

Default color space



R
(G=0,B=0)

G
(R=0,B=0)

B
(R=0,G=0)

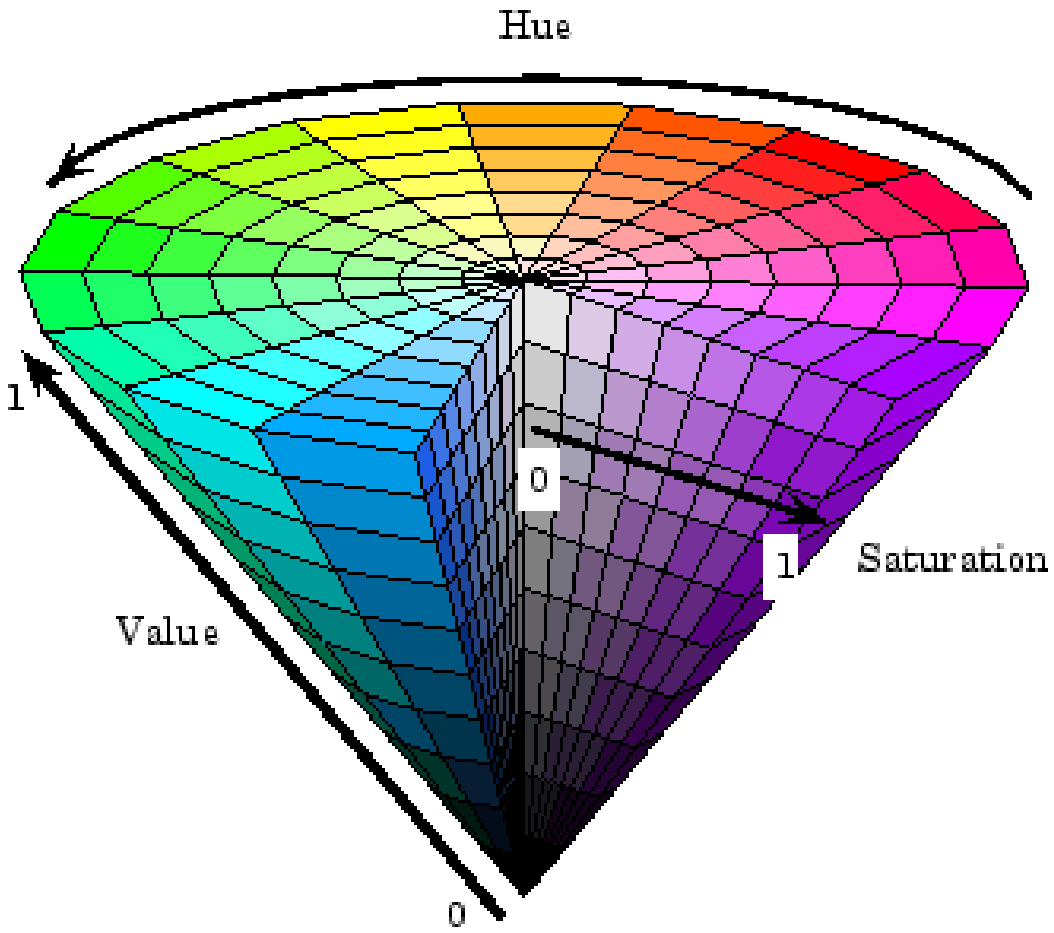
Some drawbacks

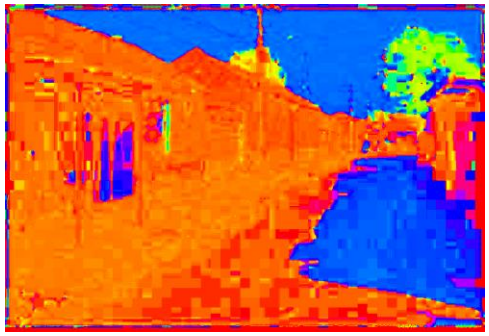
- Strongly correlated channels
- Non-perceptual

Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Color spaces: HSV

Intuitive color space





H

(S=1,V=1)

S

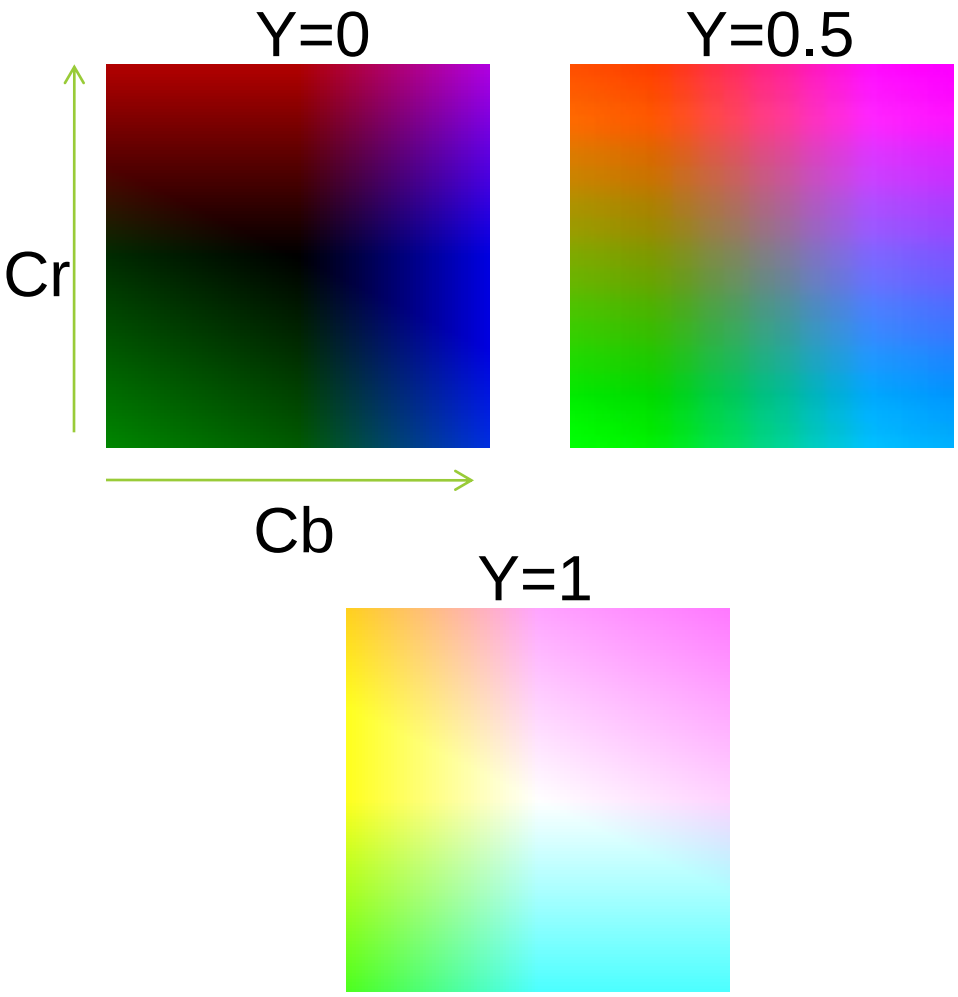
(H=1,V=1)

V

(H=1,S=0)

Color spaces: YCbCr

Fast to compute, good for compression, used by TV



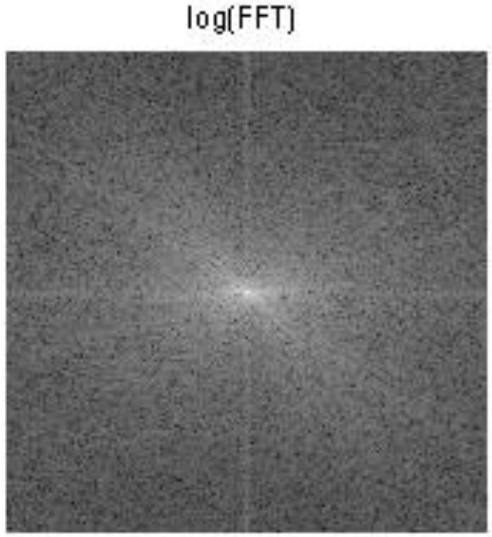
Y
(Cb=0.5,Cr=0.5)

Cb
(Y=0.5,Cr=0.5)

Cr
(Y=0.5,Cb=05)

Fourier Transform

__Any__ univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.

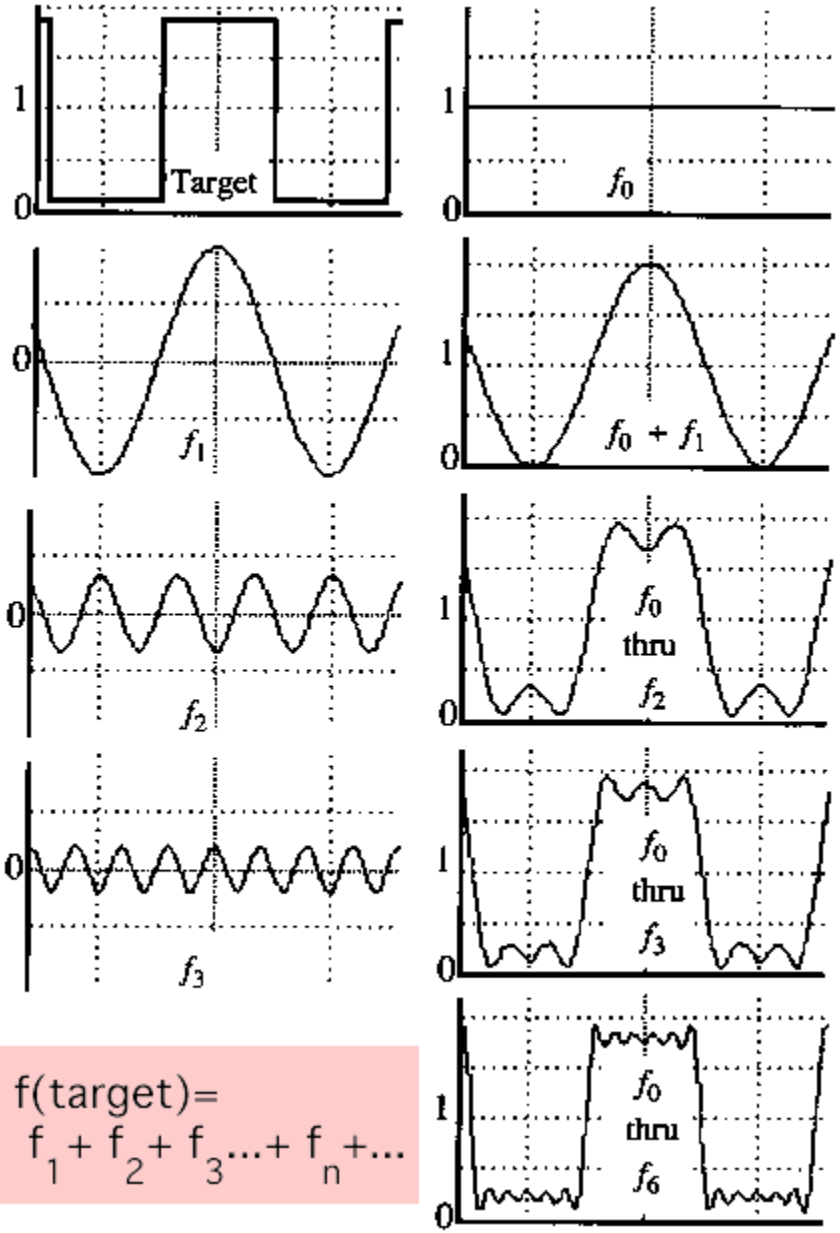


A sum of sines

Our building block:

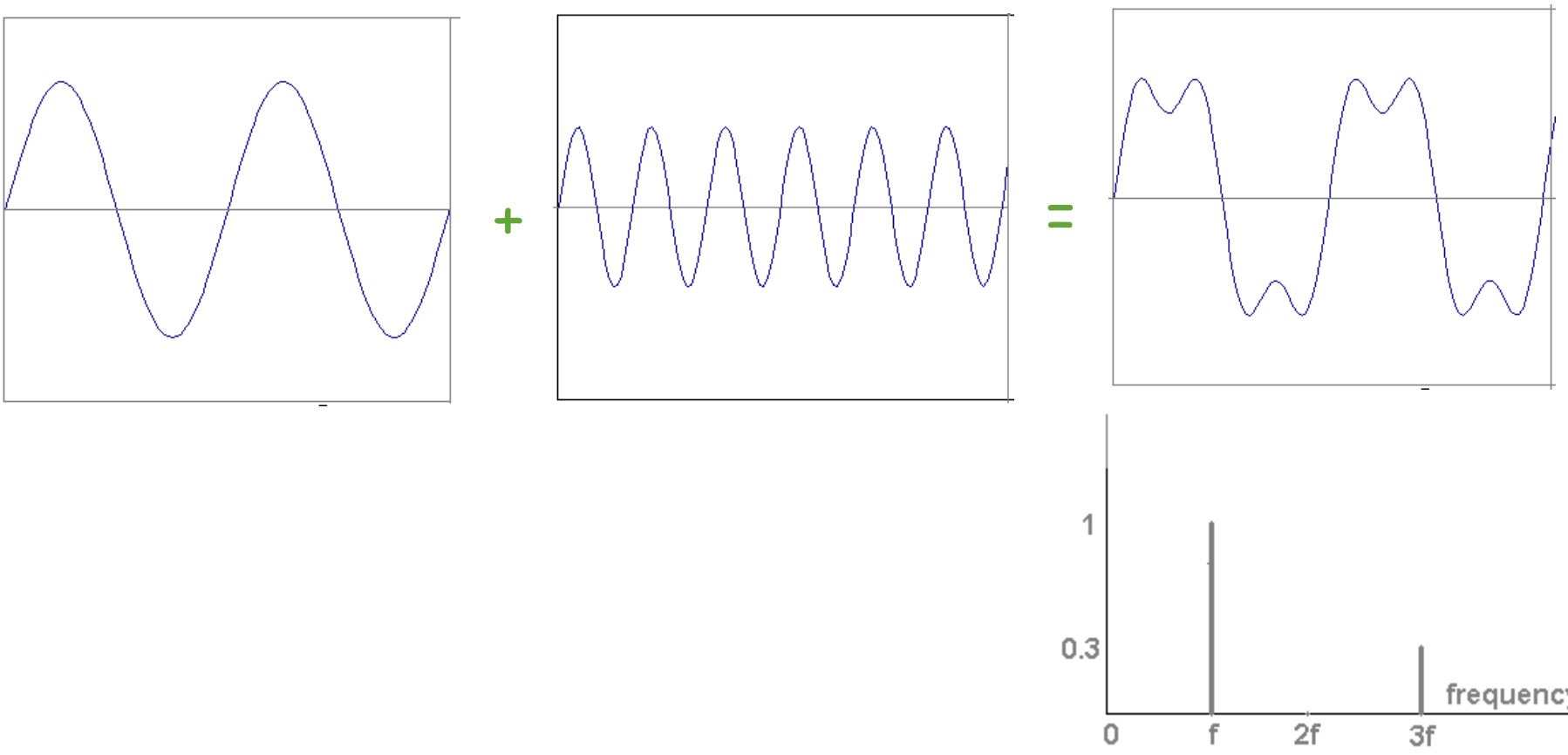
$$A\sin(\omega x + \phi)$$

Add enough of them to get any signal *g(x)* you want!



Frequency Spectra

- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$

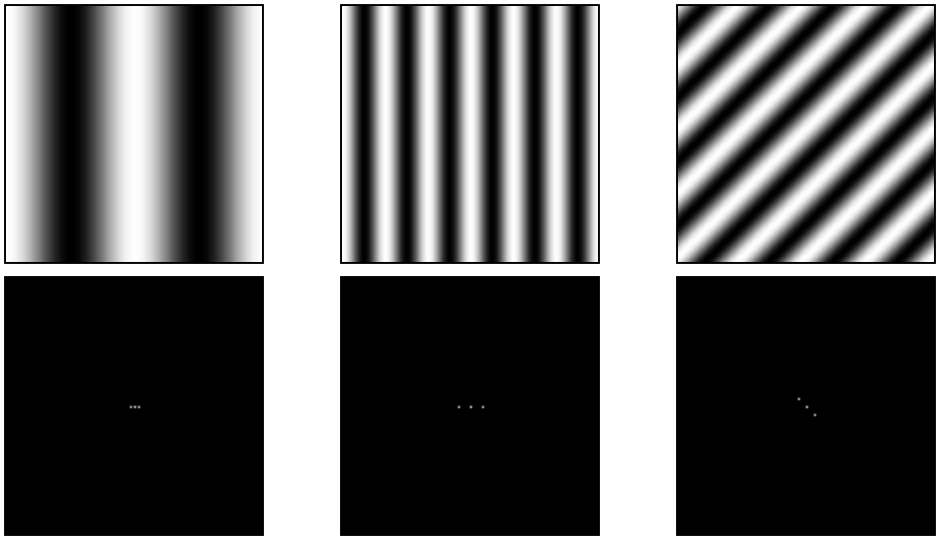


Fourier analysis in images

会津大学
THE UNIVERSITY OF AIZU

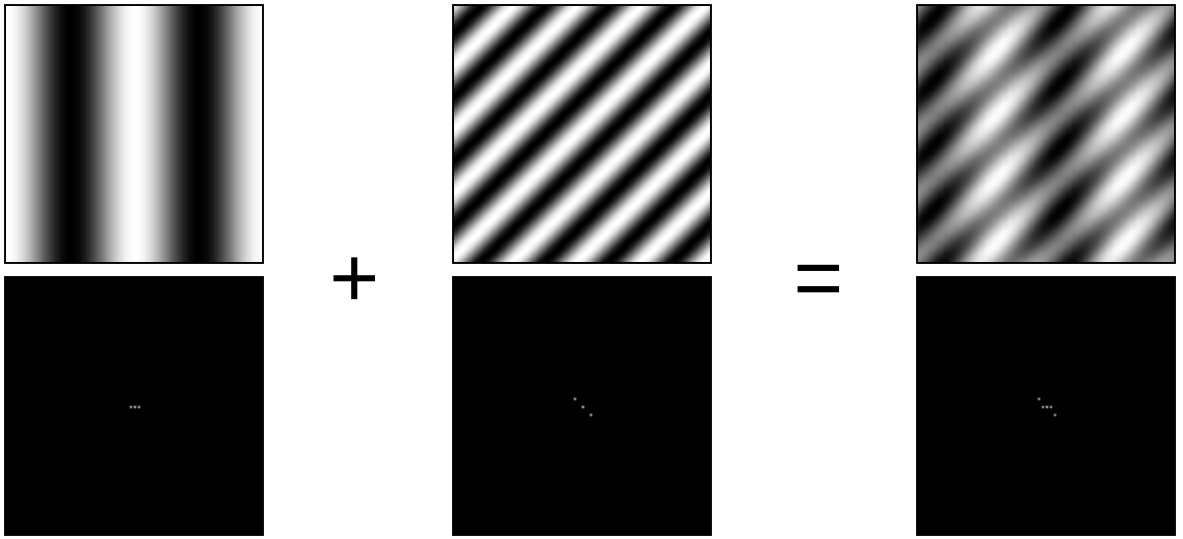
Intensity Image

Fourier Image



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Signals can be composed



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

More: http://www.cs.unm.edu/~brayer/vision/fourier.html

Overview

- Introduction
  - What is Vision Processing?
- Basic Knowledge
  - Image formation
  - Transformation
- Low level Algorithms
  - Filtering
  - Border Detection
  - Edge Detection
- Conclusion

Low Level Algorithms

- Filtering
- Simple Block matching.
- Border Detection
- Edge Detection

The big picture…

Feature Detection

e.g. DoG

Feature Description

e.g. SIFT

NEXT COSCO?

database of local descriptors

Matching / Indexing / Detection

Slide credit Fei Fei Li

Image Filtering

- Compute function of local neighborhood at each position.
- Why:
  - Enhance images:
    - Denoise, resize, increase contrast, so on.
  - Extract information from images
    - Texture, edges, distinctive points, so on.
  - Detect patterns:
    - Template matching

Image Filtering (2)

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

会津大学
THE UNIVERSITY OF AIZU

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0								

$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image Filtering (3)

$g[\cdot,\cdot]$ $\frac{1}{9}$

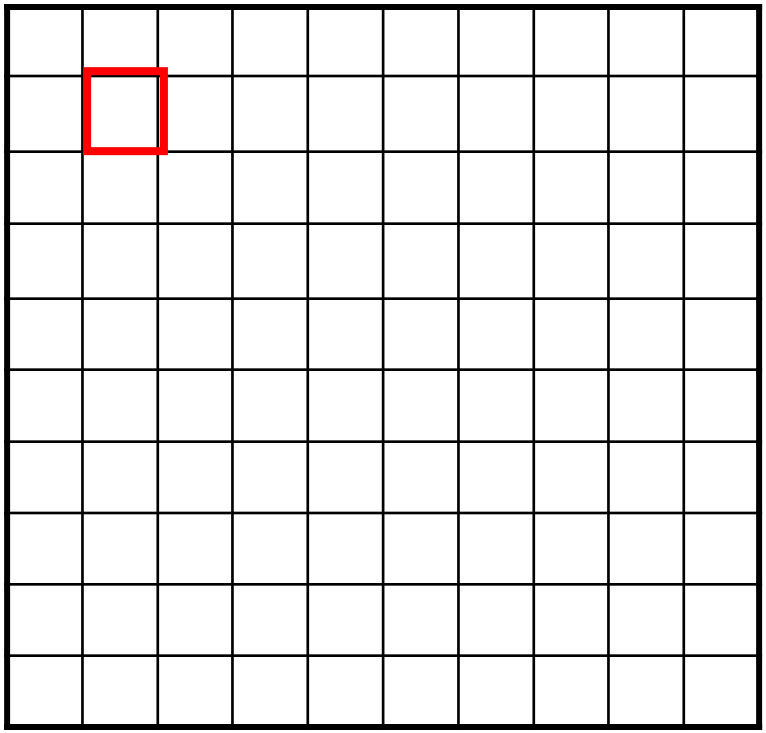
1	1	1
1	1	1
1	1	1

会津大学
THE UNIVERSITY OF AIZU

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$



$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image Filtering (4)

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

会津大学
THE UNIVERSITY OF AIZU

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20						

$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image Filtering (5)

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

会津大学
THE UNIVERSITY OF AIZU

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

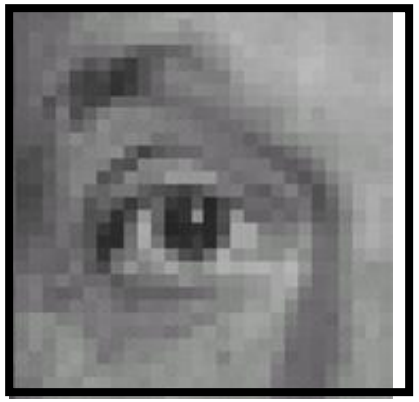
$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Example of Filters

Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Original

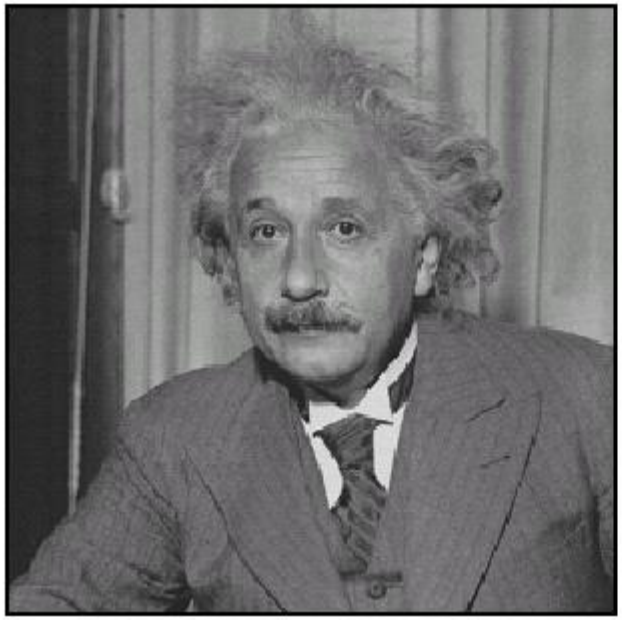
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter

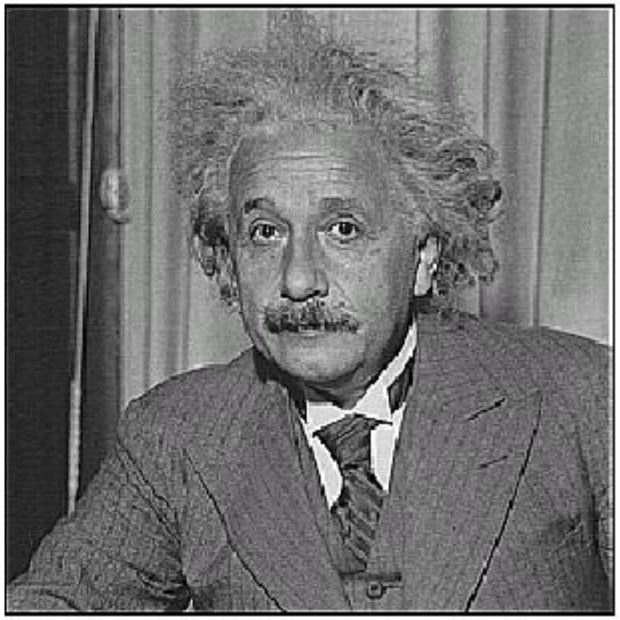
- Accentuates differences with local average

Source: D. Lowe

Sharpening



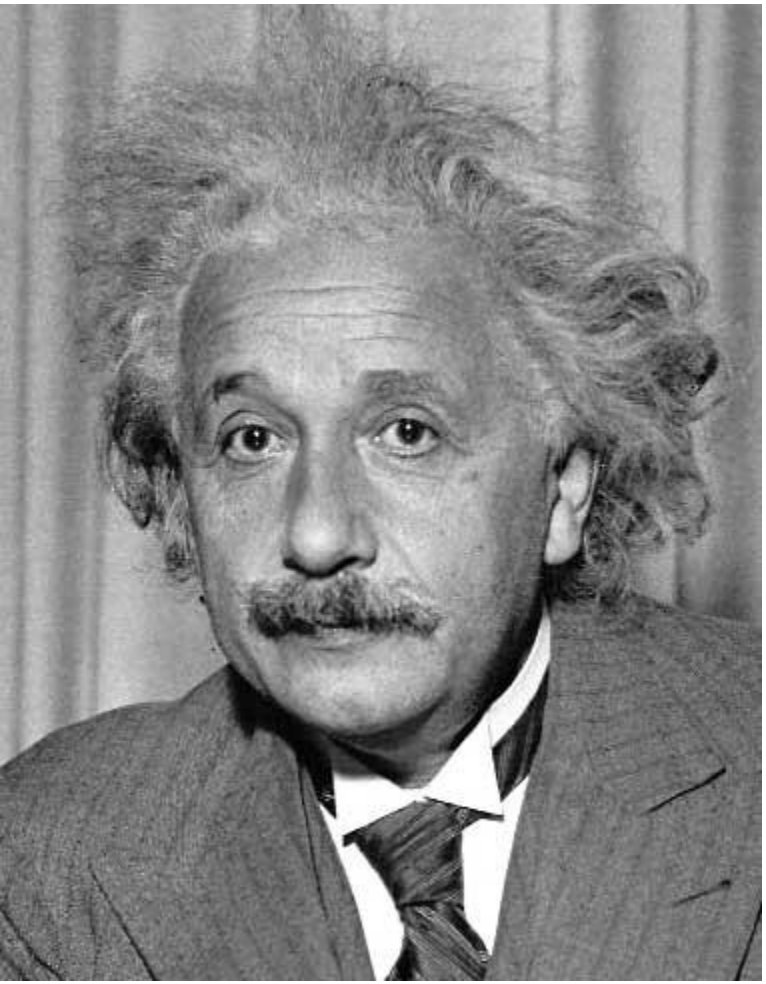
before



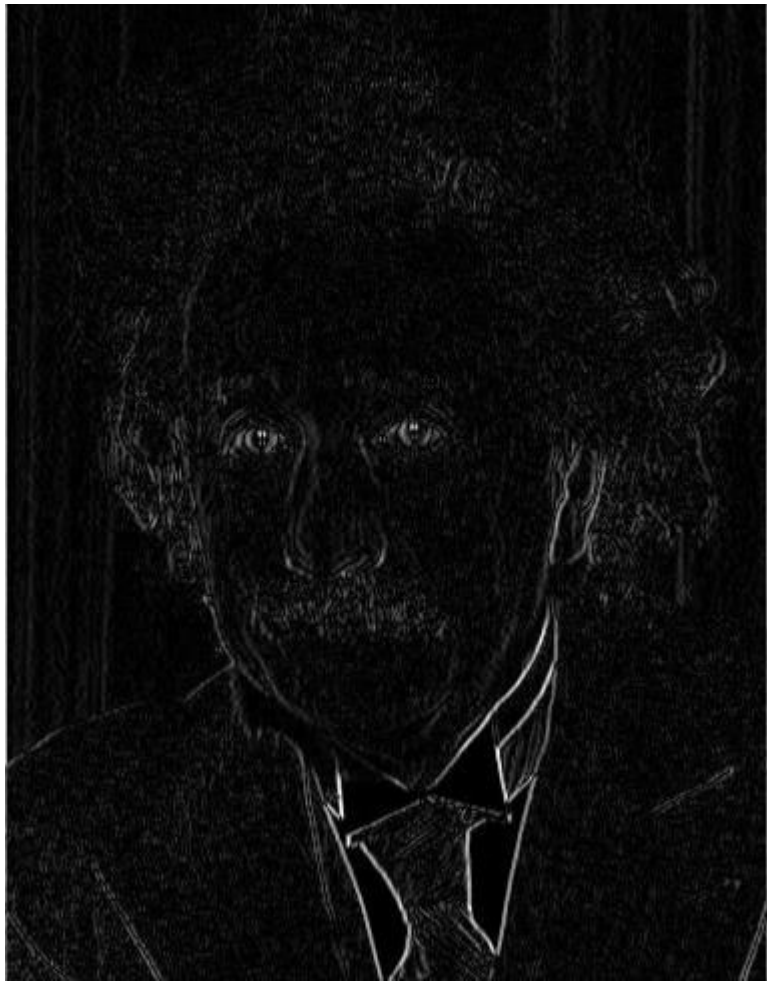
after

Source: D. Lowe

Sobel Filter: Vertical

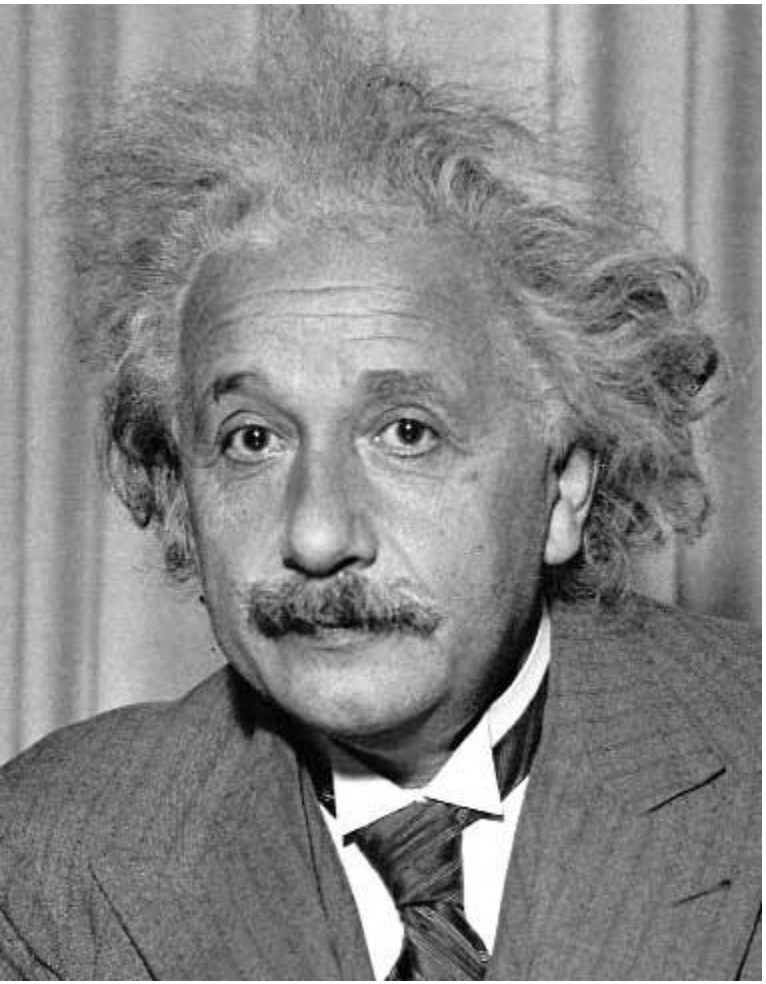


1	0	-1
2	0	-2
1	0	-1



Vertical Edge
(absolute value)

Sobel Filter: Horizontal

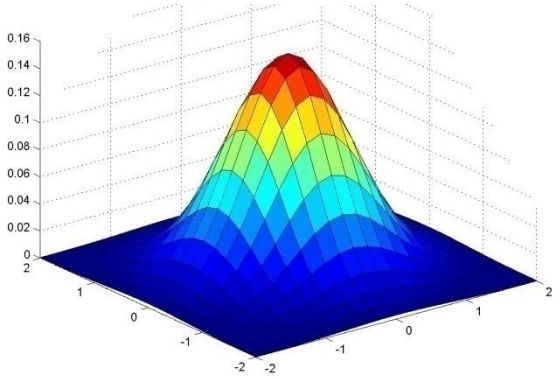


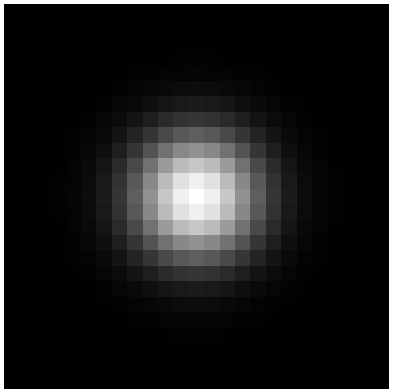
1	2	1
0	0	0
-1	-2	-1

Horizontal Edge
(absolute value)

Gaussian Filter

- Weight contributions of neighboring pixels by nearness.
- Gaussian Filter is low pass filter.



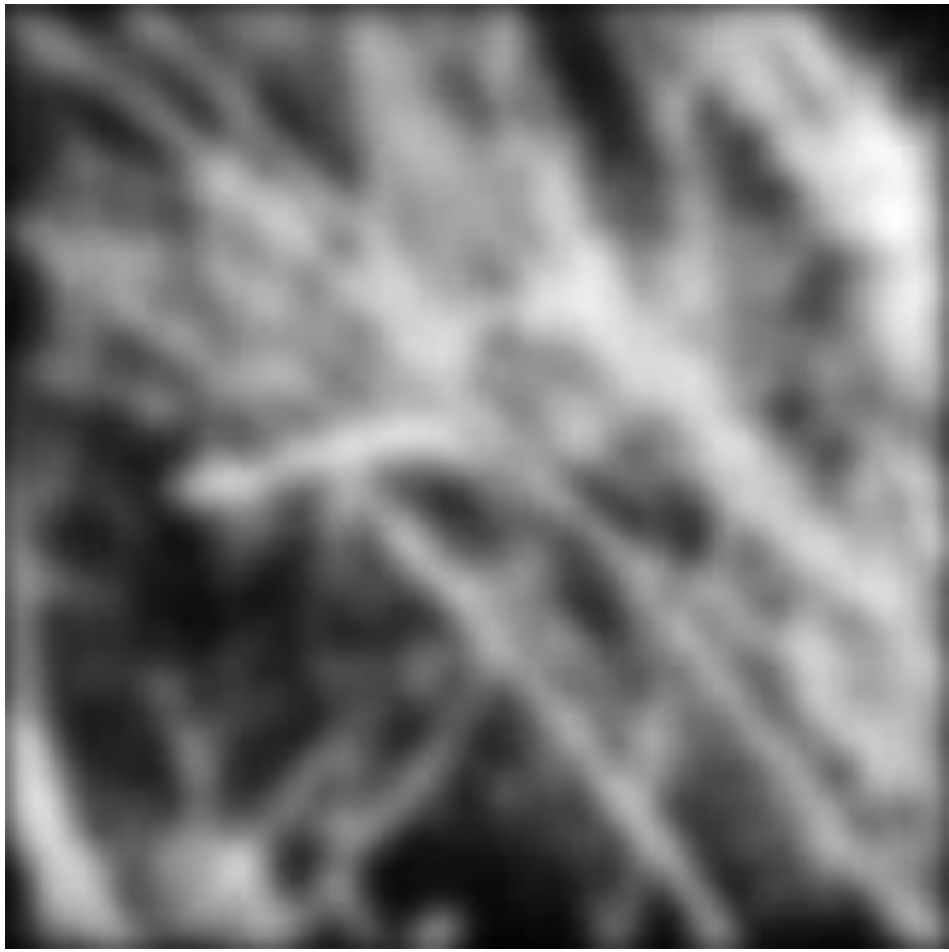


0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

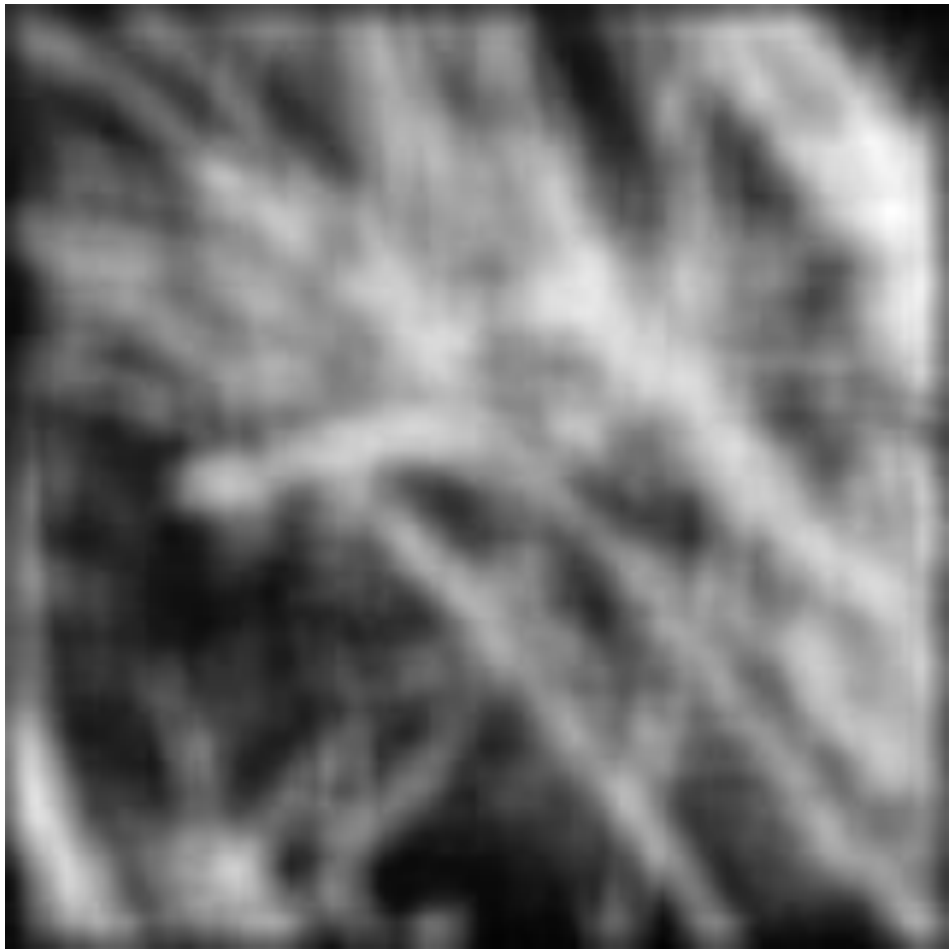
5 x 5, $\sigma = 1$

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Gaussian Filter



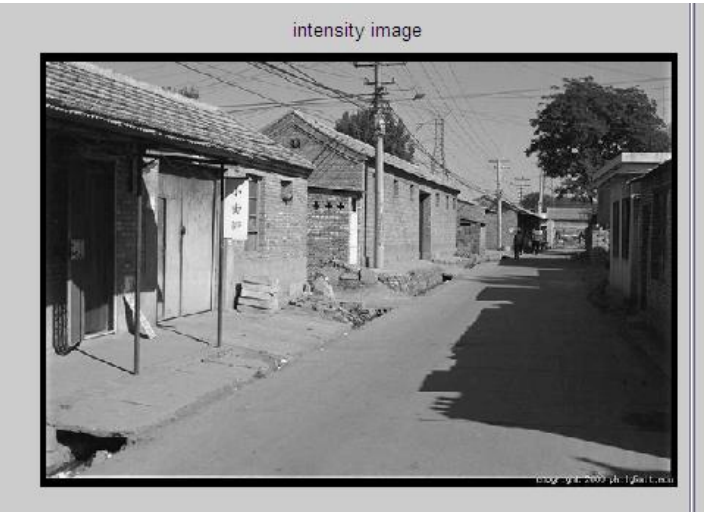
Box Filter

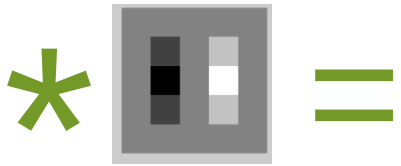


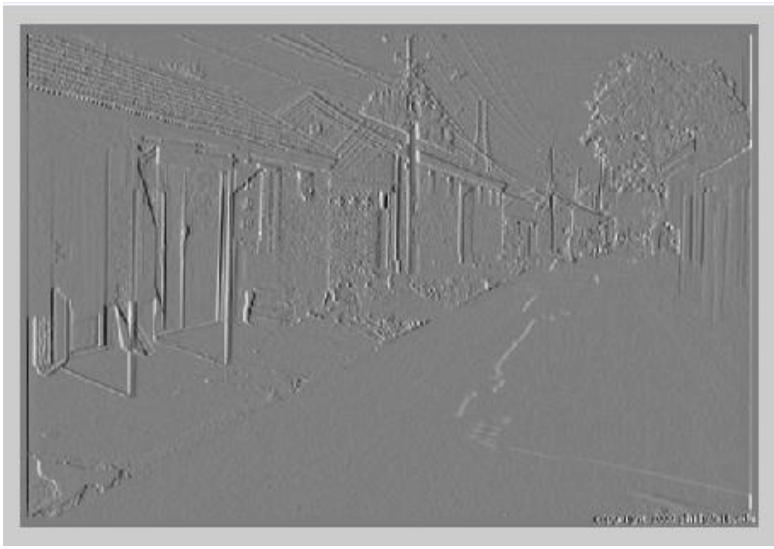
Filtering in spatial domain

会津大学
THE UNIVERSITY OF AIZU

1	0	-1
2	0	-2
1	0	-1

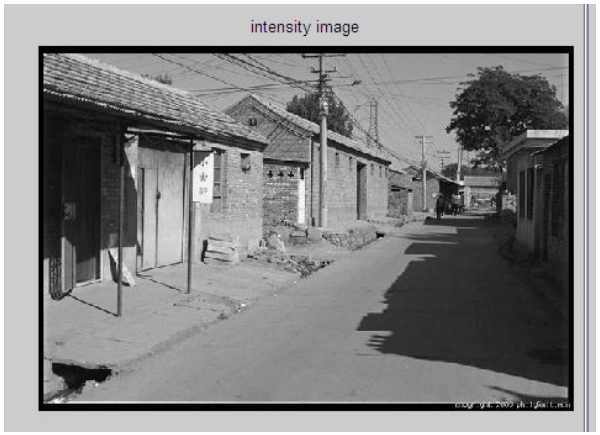




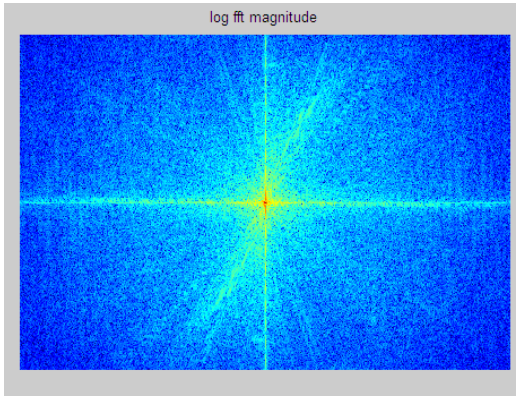


Filtering in frequency domain

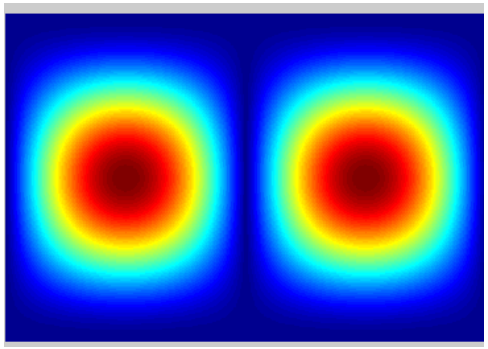
FFT



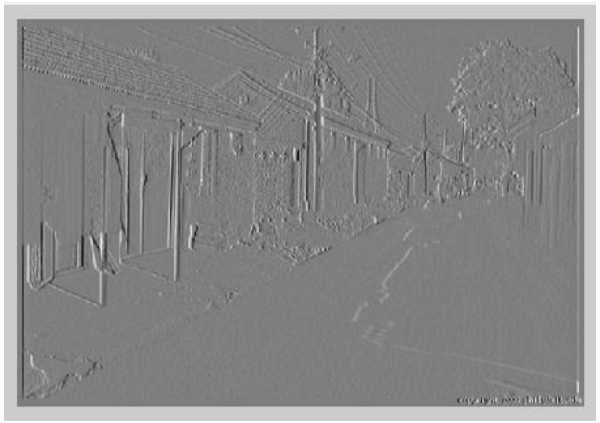
FFT



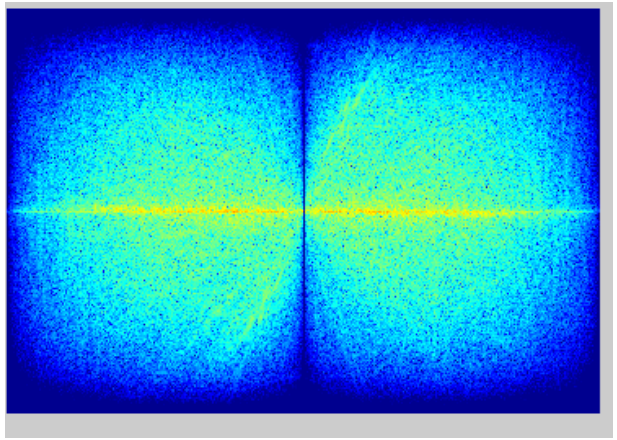
×



=



Inverse FFT

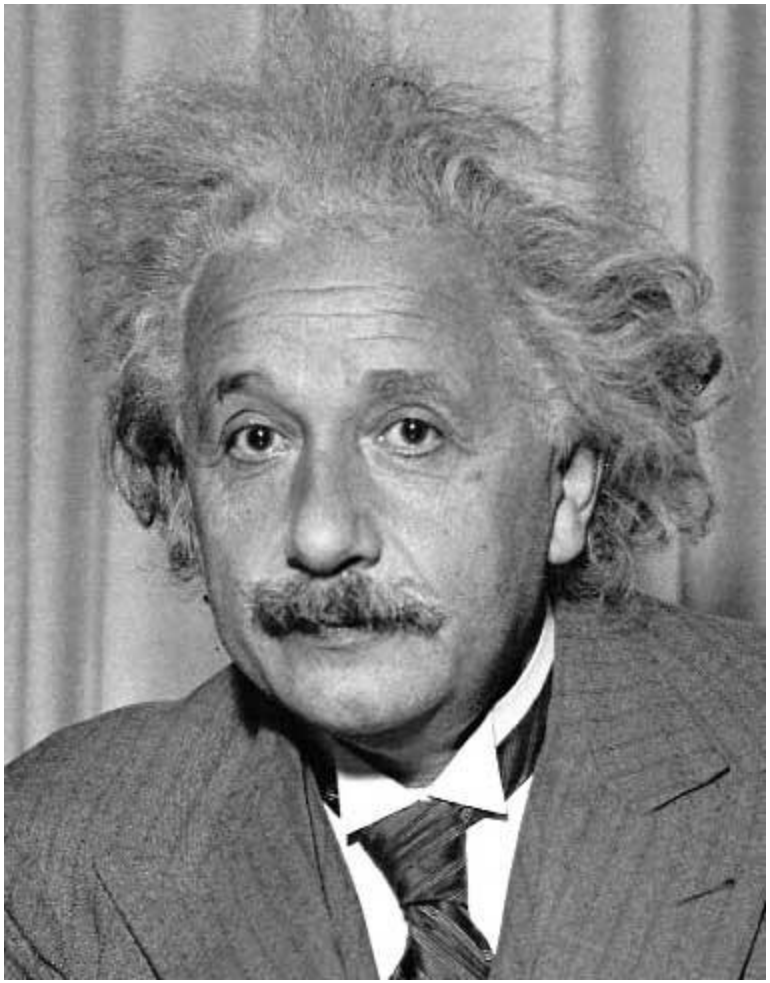


Slide: Hoiem

Template matching

- Goal: find  in image

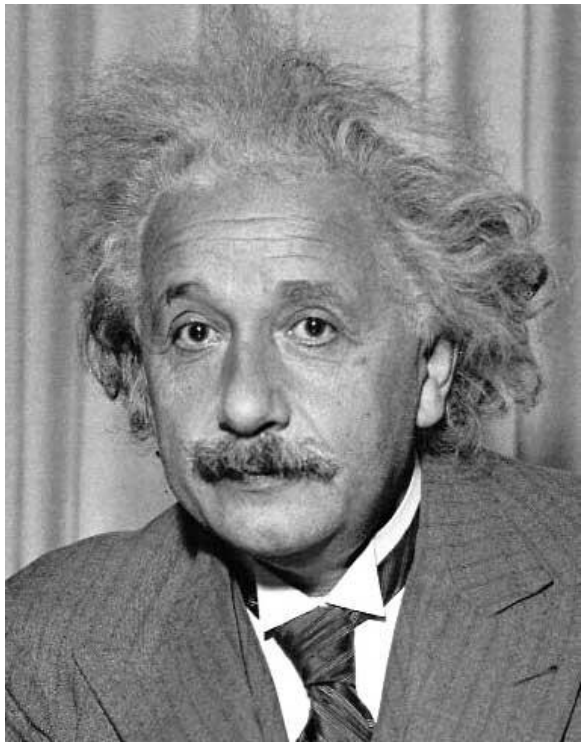
- Main challenge: What is a good similarity or distance measure between two patches?
  - Correlation
  - Zero-mean correlation
  - Sum Square Difference
  - Normalized Cross Correlation



Matching with filters

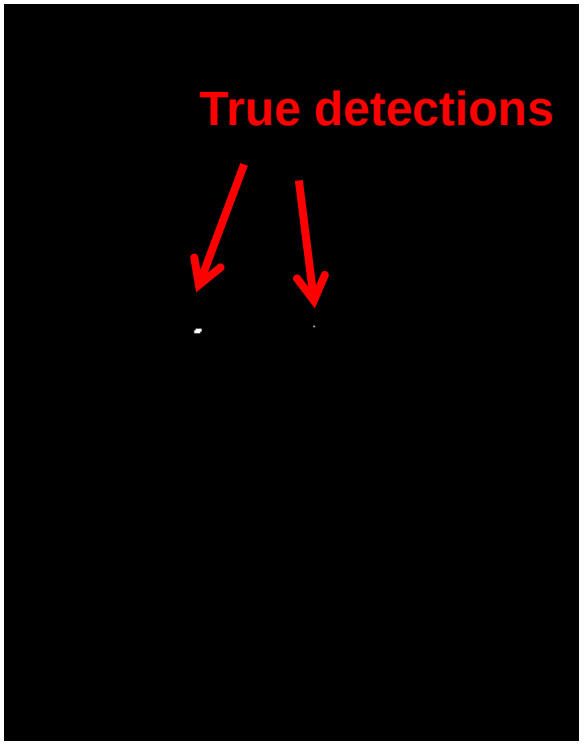
- Goal: find  in image
- Method : SSD

$$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k,n+l])^2$$



Input

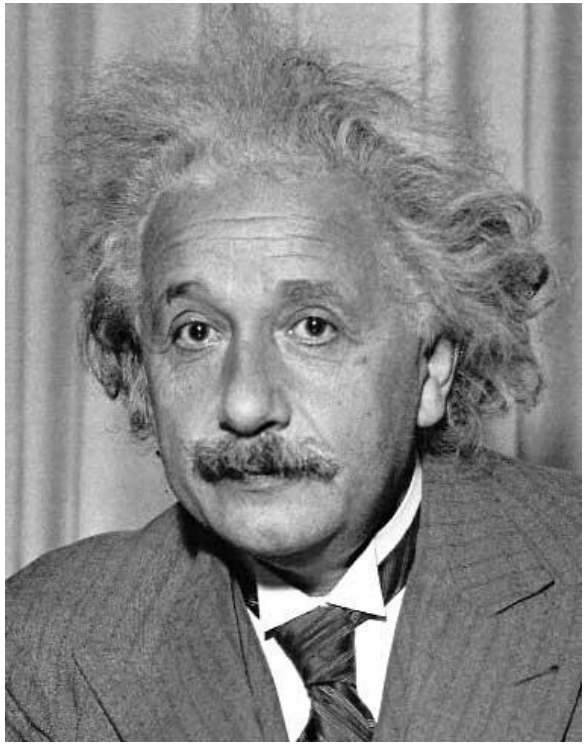
1- sqrt(SSD)



Thresholded Image

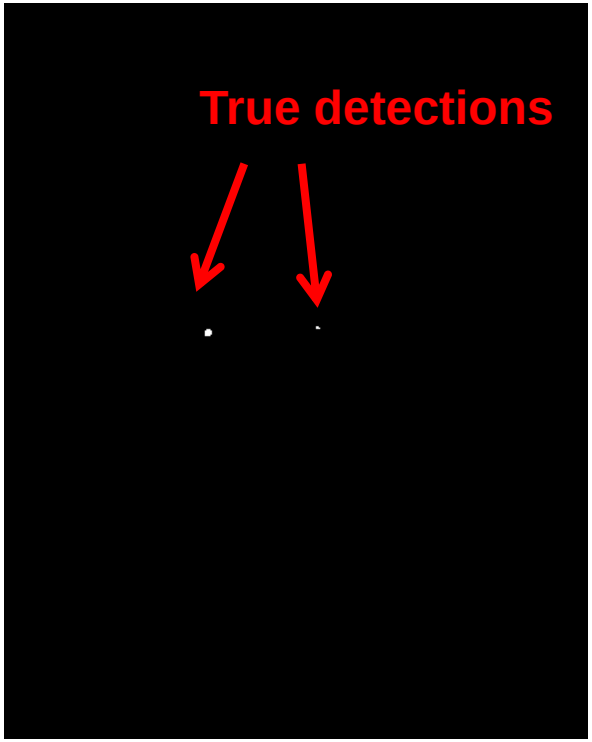
Matching with filters

- Goal: find  in image
- Method 3: Normalized cross-correlation



Input

Normalized X-Correlation



Thresholded Image

Matching with filters

- Goal: find  in image
- Method : Normalized cross-correlation

mean template

mean image patch

$$h[m,n] = \frac{\sum_{k,l}(g[k,l]-\bar{g})(f[m-k,n-l]-\bar{f}_{m,n})}{\left(\sum_{k,l}(g[k,l]-\bar{g})^2 \sum_{k,l}(f[m-k,n-l]-\bar{f}_{m,n})^2\right)^{0.5}}$$

Matlab: `normxcorr2(template, im)`

Compare

- SSD: faster, sensitive to overall intensity
- Normalized cross-correlation: slower, invariant to local average intensity and contrast

Matching smaller / larger eyes?

- Image pyramid: down-sampling and matching

Down-sampling

Imag

w-Res
mage

Source: Forsyth

Matching with Image pyramid

Input: Image, Template

1. Match template at current scale

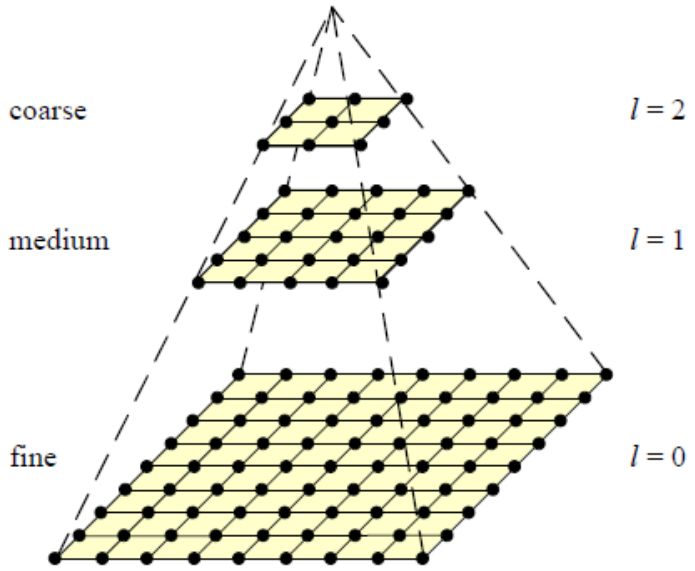
2. Downsample image

3. Repeat 1-2 until image is very small

4. Take responses above some threshold, perhaps with non-maxima suppression

Coarse-to-fine Image Registration

1. Compute Gaussian pyramid
2. Align with coarse pyramid
3. Successively align with finer pyramids
   - Search smaller range

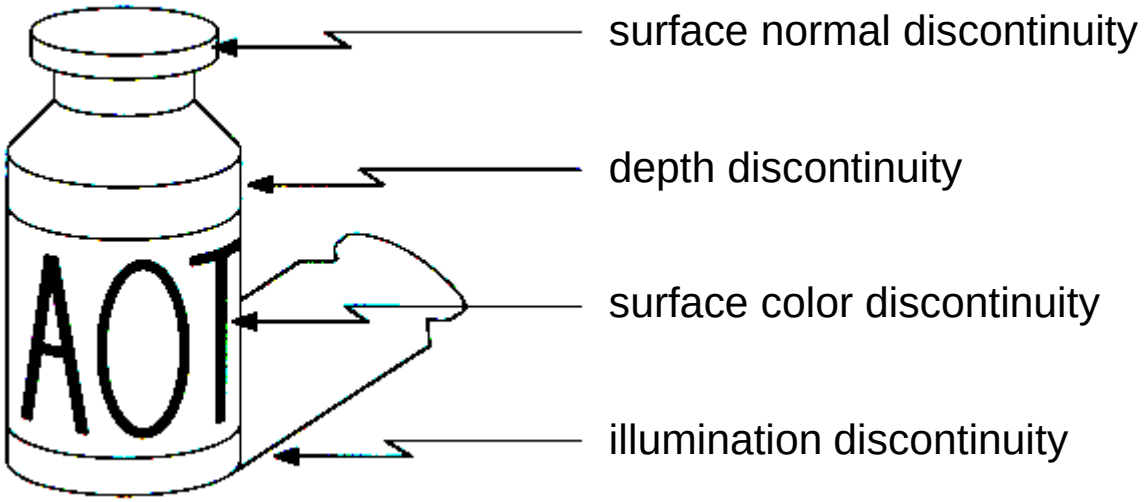


Why is this faster?

Are we guaranteed to get the same result?

Edge/Border Detection

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

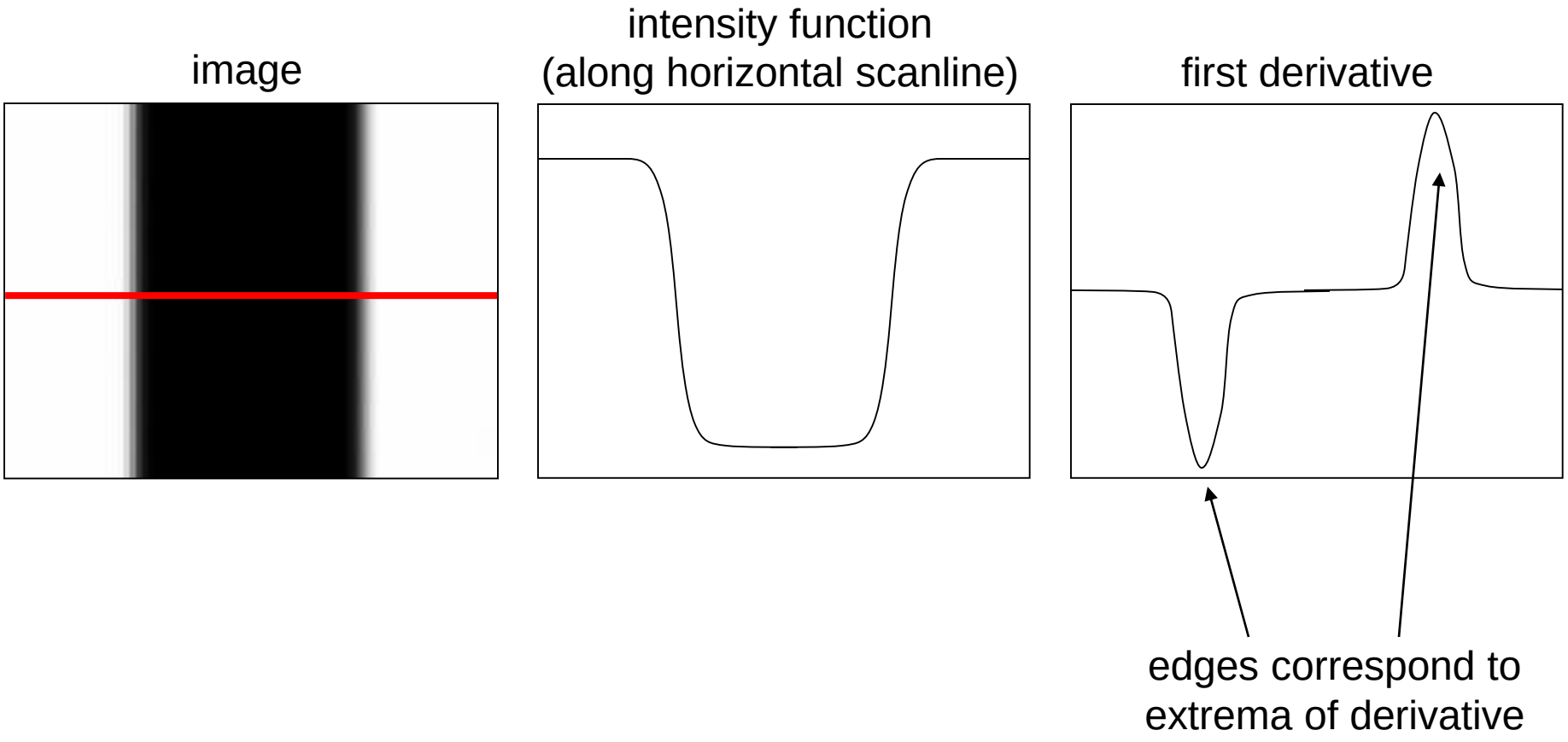


- Edges are caused by a variety of factors

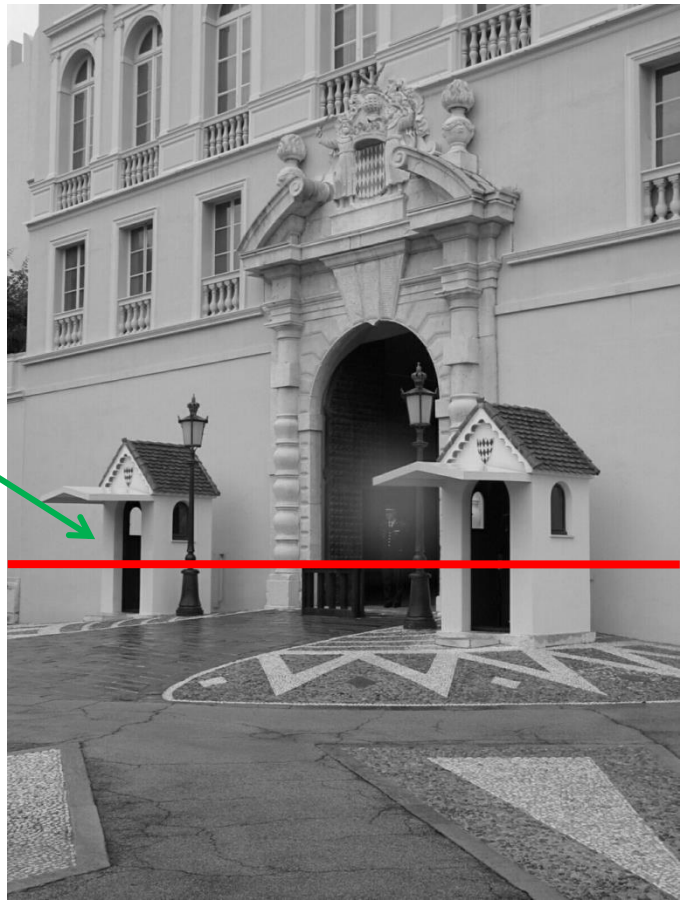
Source: Steve Seitz

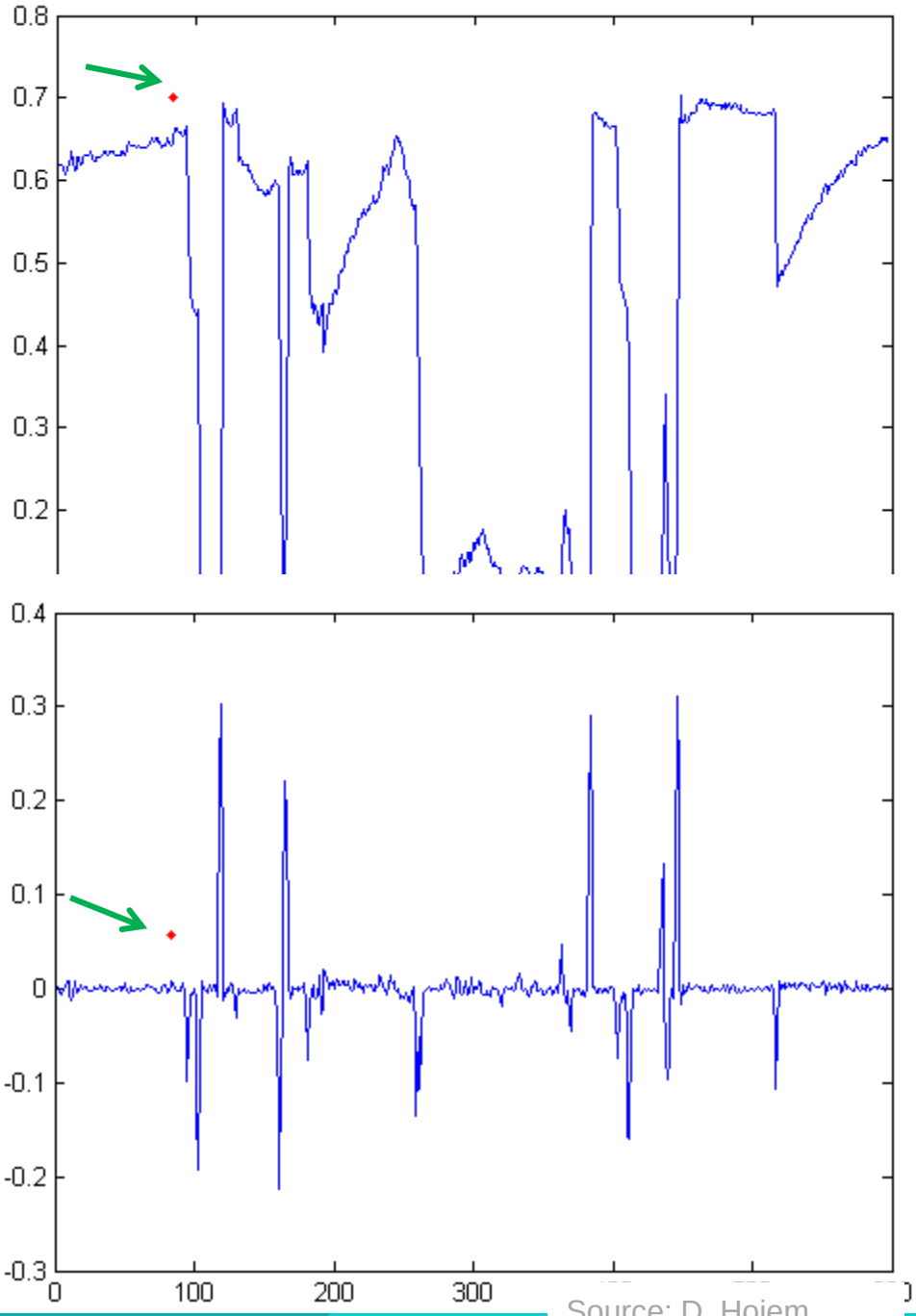
Characterizing edges

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



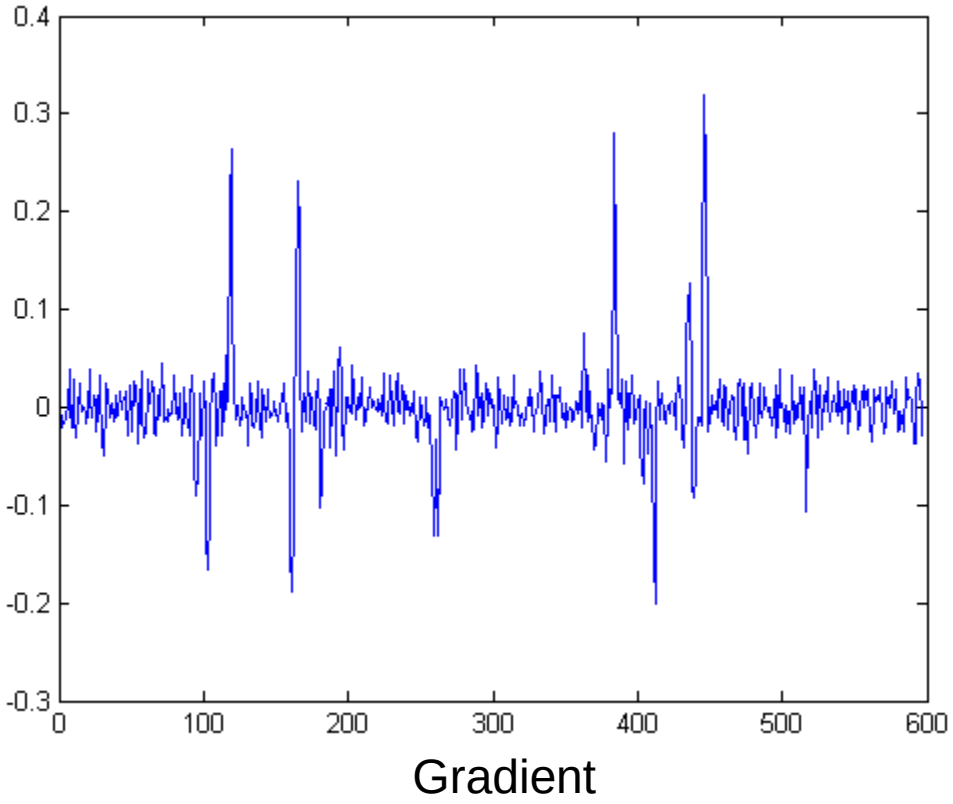
Intensity profile





Source: D. Hoiem

With a little Gaussian noise

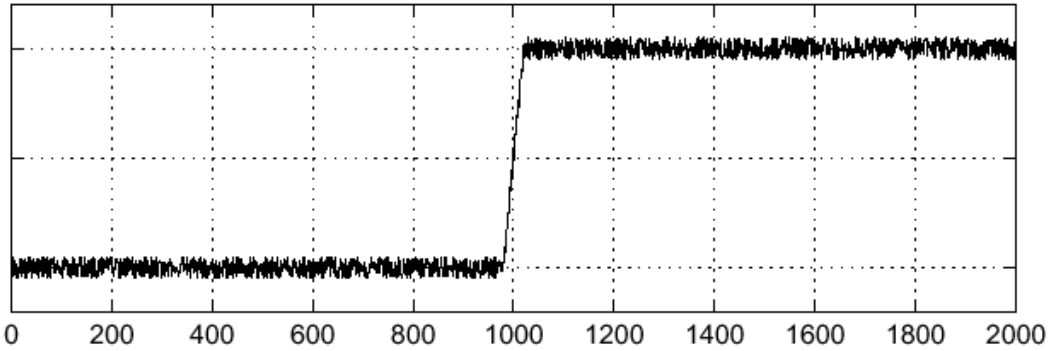


Source: D. Hoiem

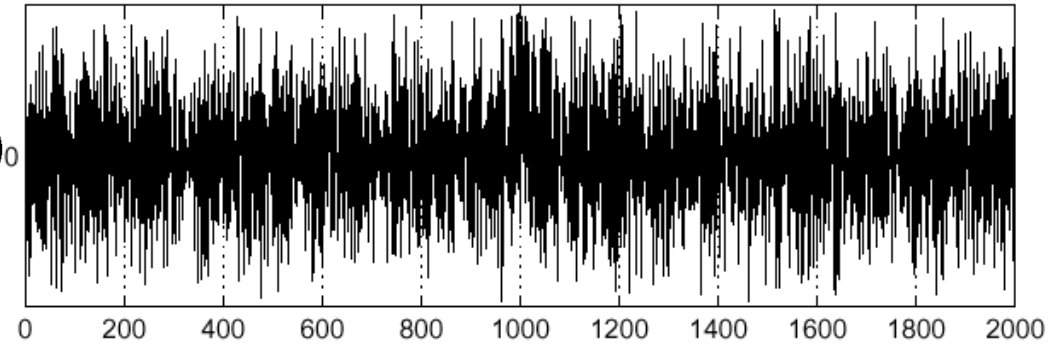
Effects of noise

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

$f(x)$



$\frac{d}{dx} f(x)$

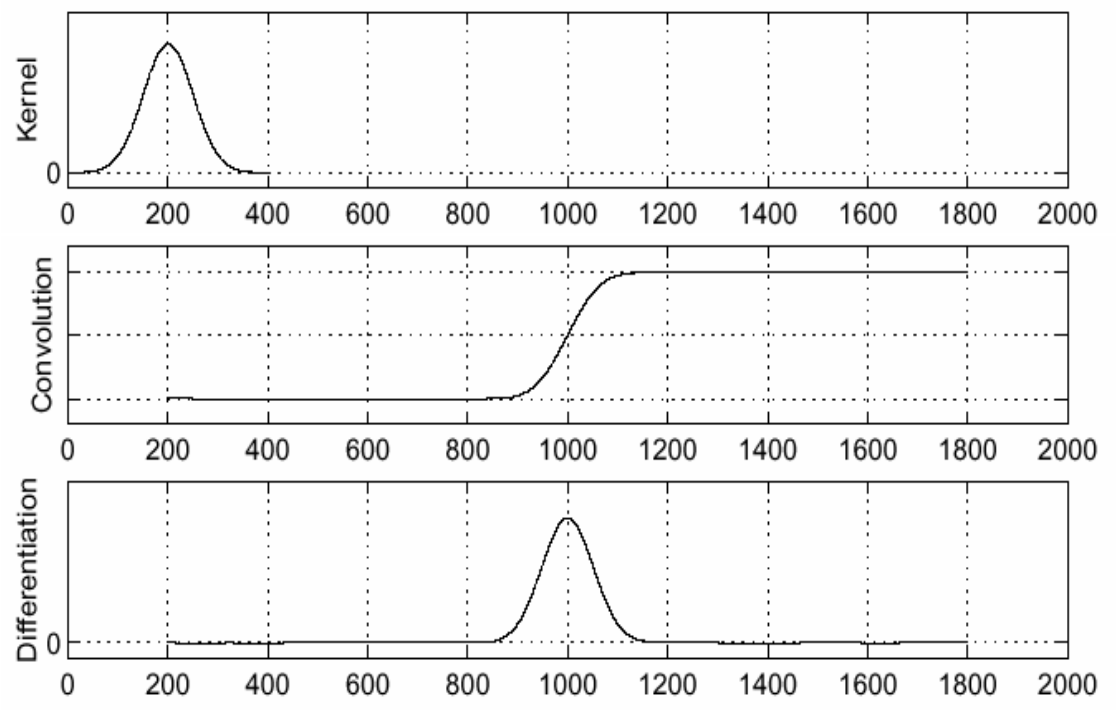


Source: S. Seitz

Solution

f

g



$f * g$

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

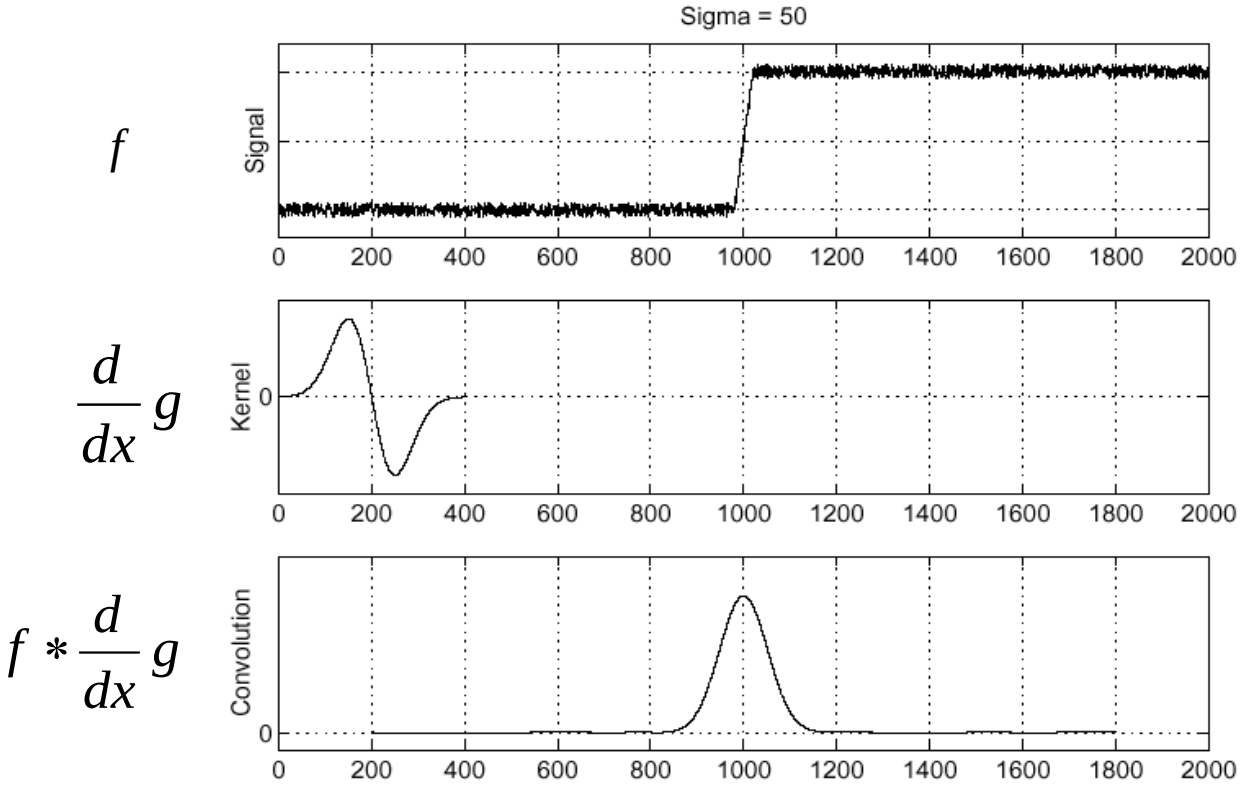
- To find edges, look for peaks in $\frac{d}{dx}(f * g)$

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:



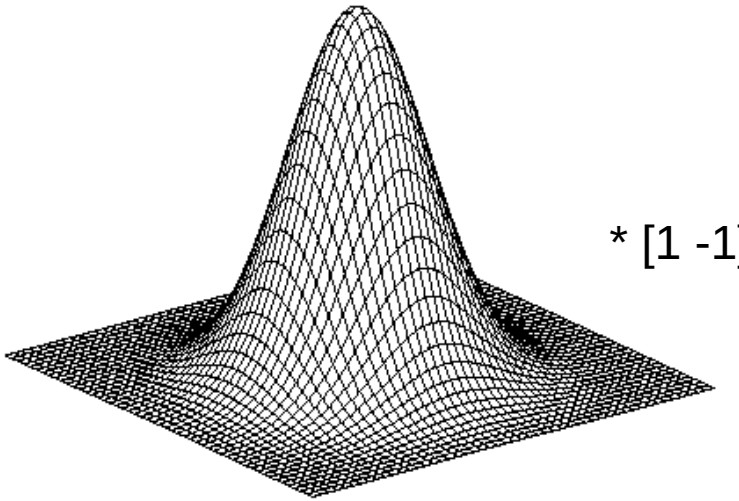
Canny edge detector

- This is probably the most widely used edge detector in computer vision
- Theoretical model: step-edges corrupted by additive Gaussian noise
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization

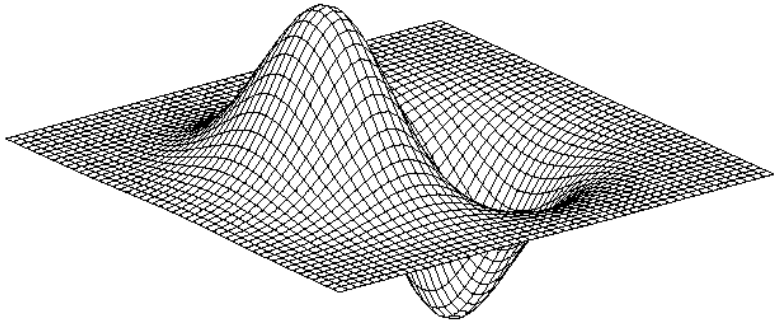
J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Source: L. Fei-Fei

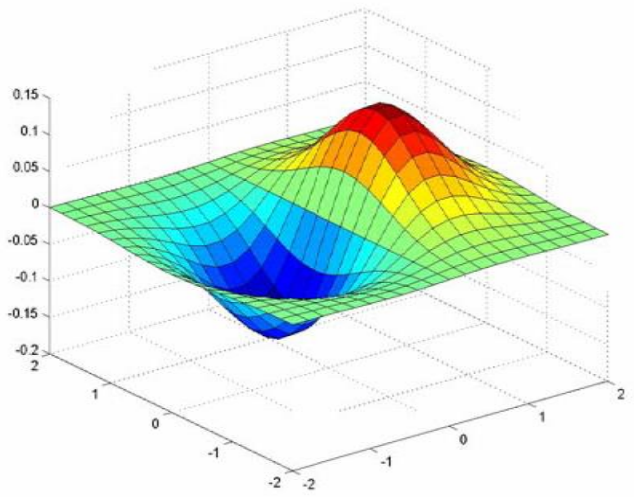
Derivation of Gaussian



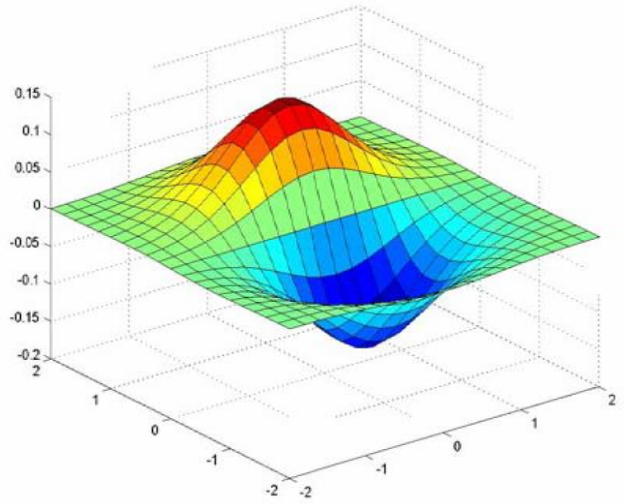
* [1 -1] =



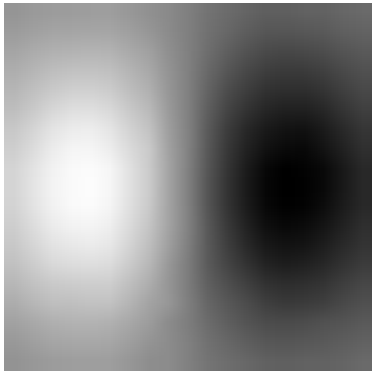
Derivative of Gaussian filter

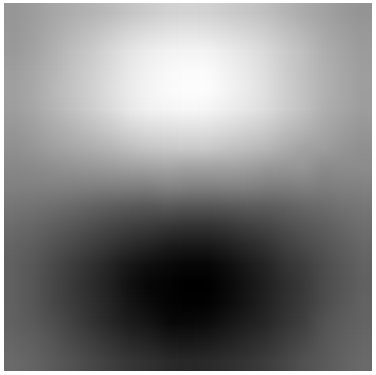


x-direction



y-direction

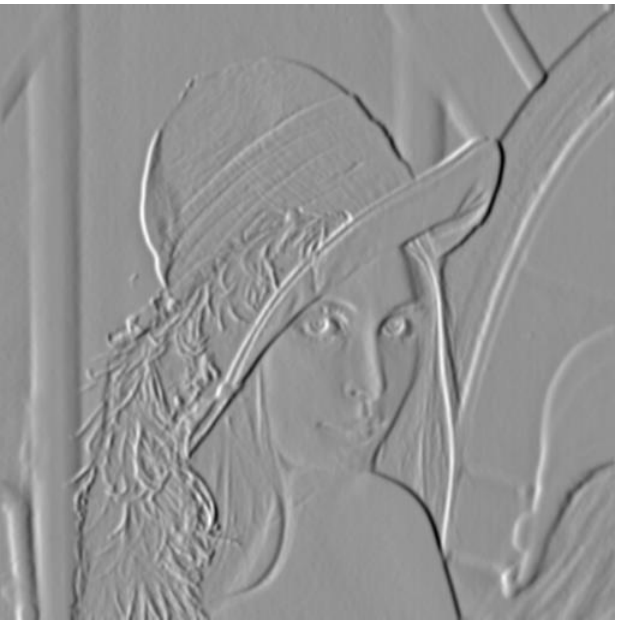




Example

original image (Lena)

Compute Gradients (DoG)



X-Derivative of Gaussian

Y-Derivative of Gaussian

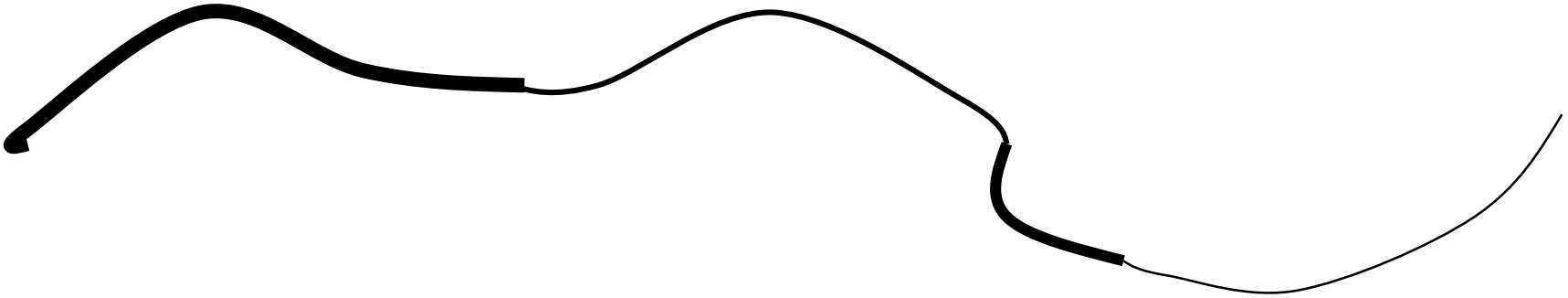
Gradient Magnitude

Hysteresis thresholding

- Threshold at low/high levels to get weak/strong edge pixels
- Do connected components, starting from strong edge pixels

Hysteresis thresholding

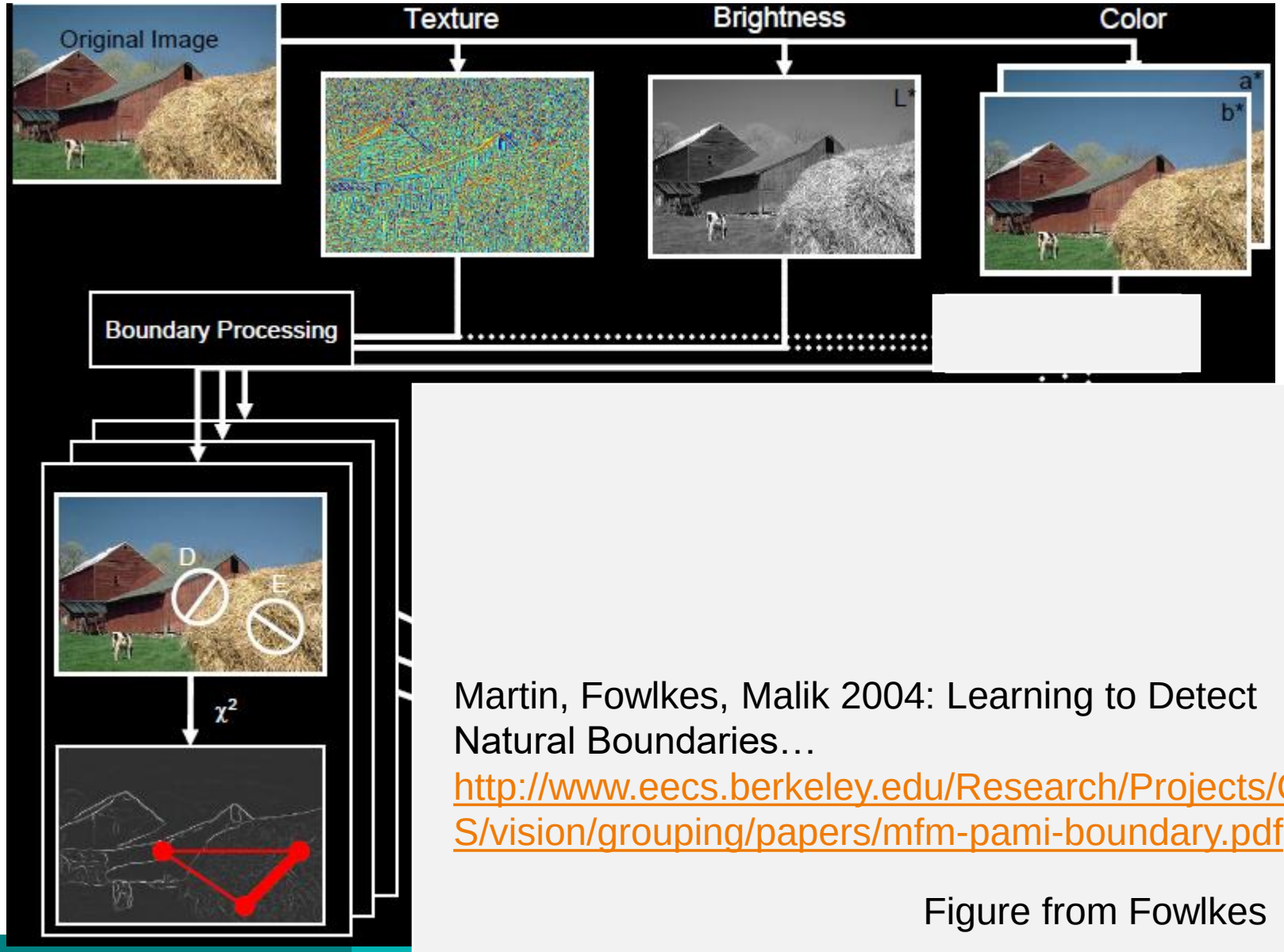
- Check that maximum value of gradient value is sufficiently large
  - drop-outs? use **hysteresis**
    - use a high threshold to start edge curves and a low threshold to continue them.



Source: S. Seitz

Final Canny Edges

pB boundary detector



Martin, Fowlkes, Malik 2004: Learning to Detect Natural Boundaries…
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/papers/mfm-pami-boundary.pdf

Figure from Fowlkes

pB Boundary Detector

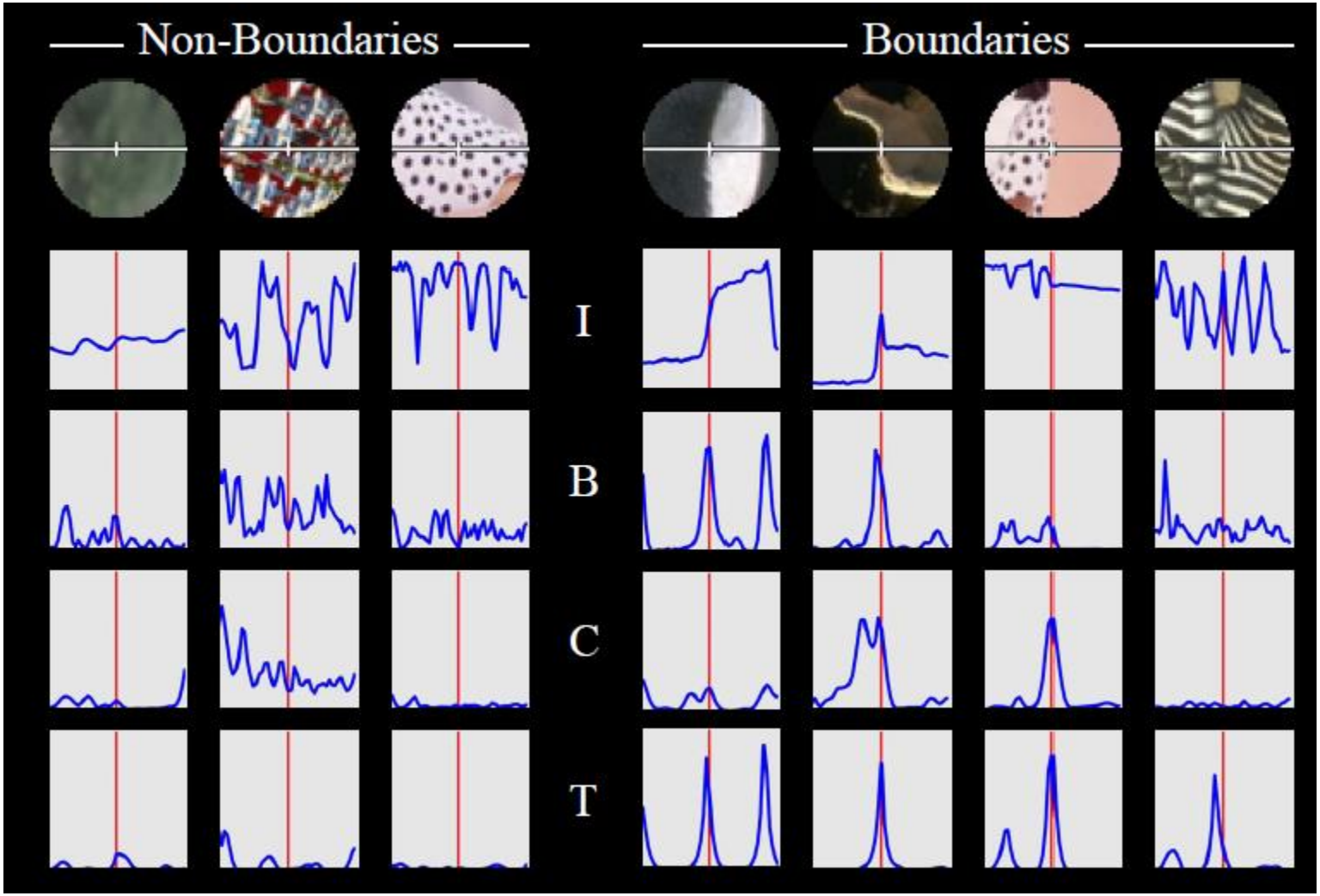


Figure from Fowlkes

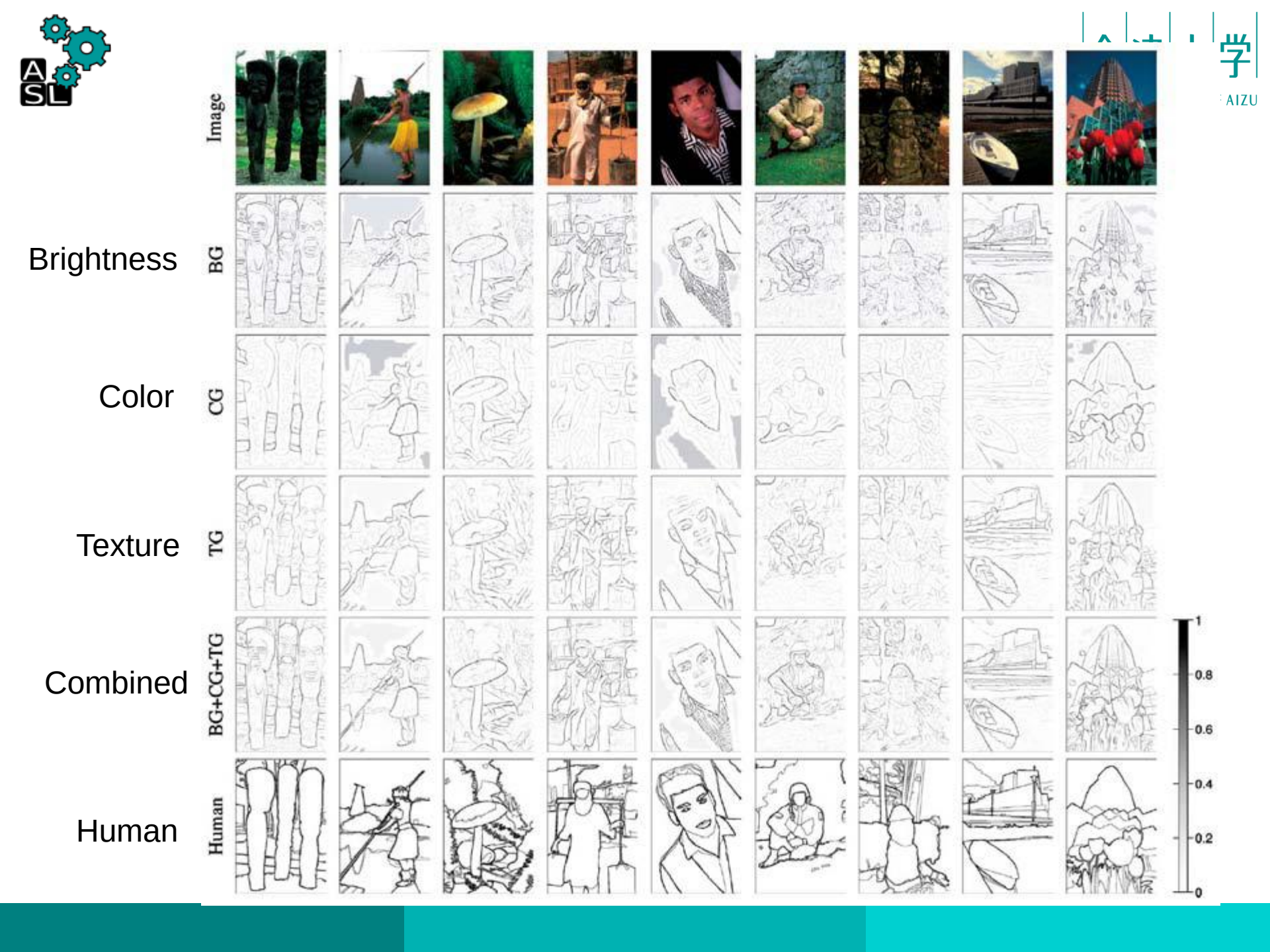
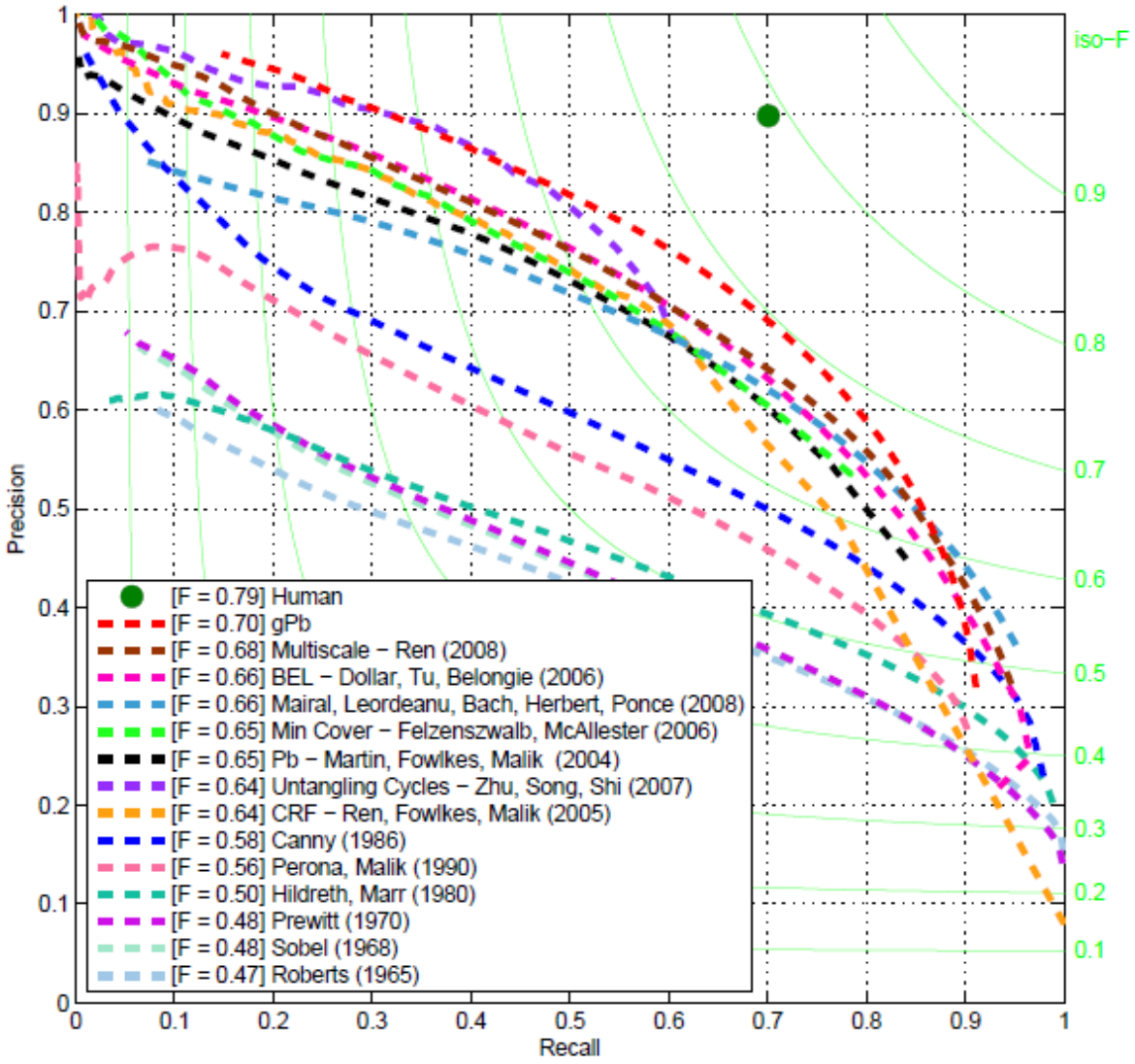


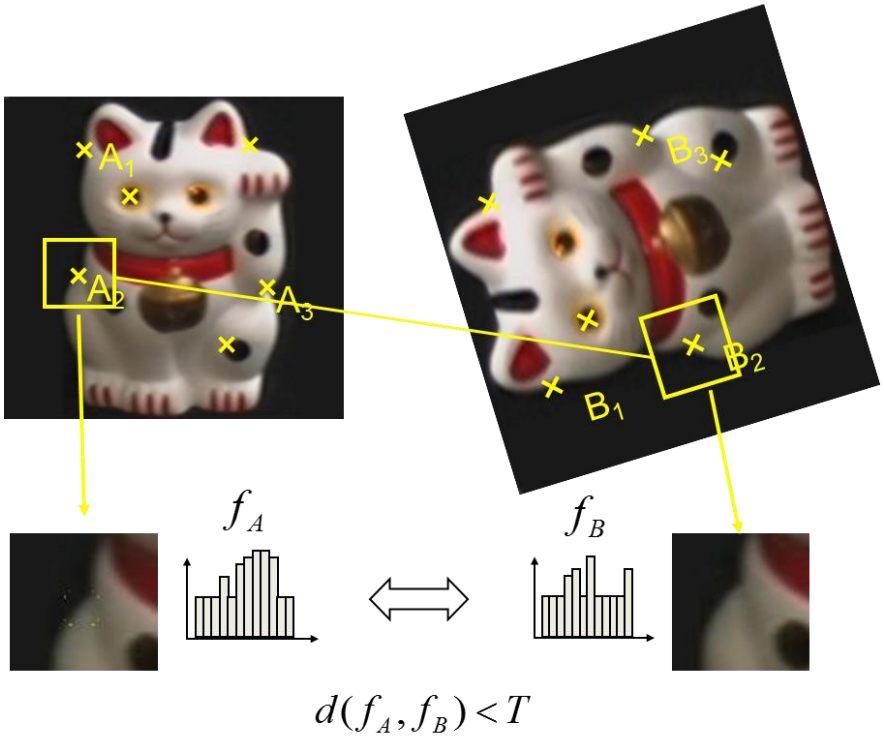
Image
BG
CG
TG
BG+CG+TG
Human
Brightness
Color
Texture
Combined
Human
1
0.8
0.6
0.4
0.2
0

45 years of boundary detection



Conner Detection

- For matching, interesting points are most concerned.
- Most of interesting points are conner:

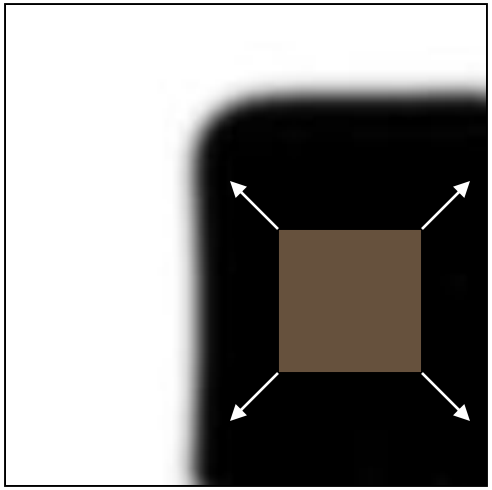


1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

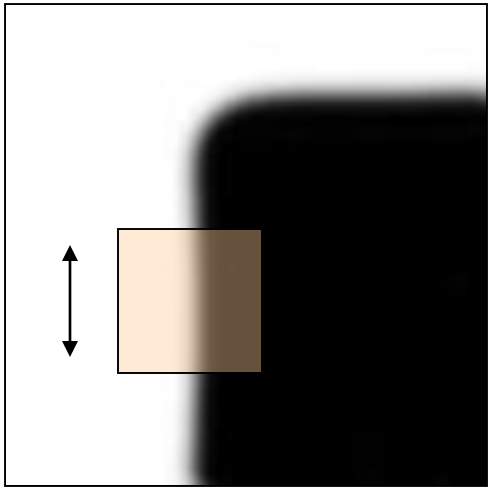
K. Grauman, B. Leibe

Corner Detection: Basic Idea

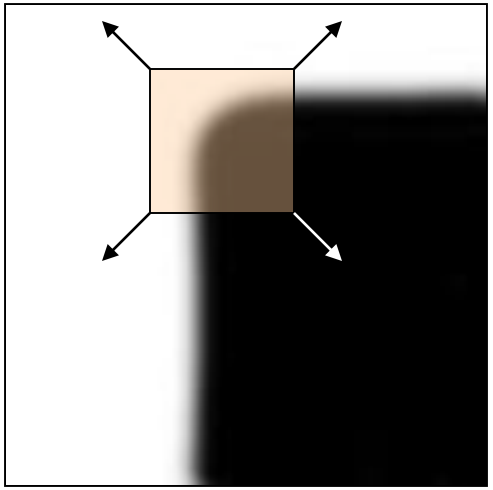
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region:
no change in all directions



"edge":
no change along the edge direction



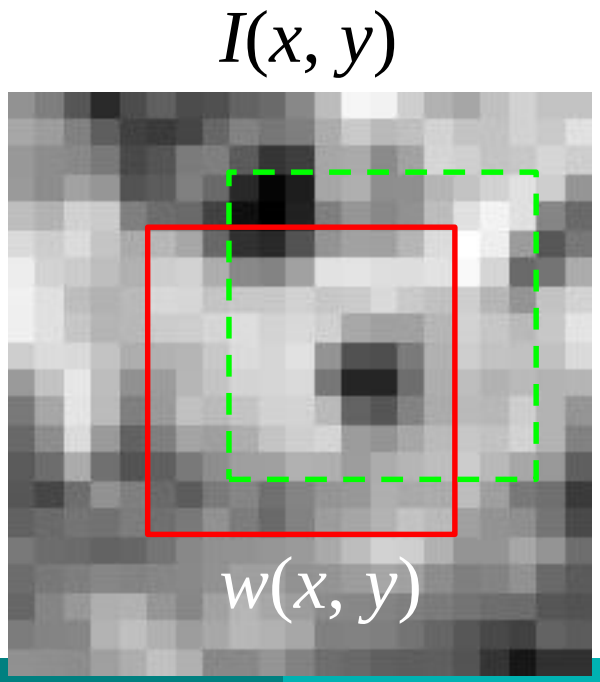
"corner":
significant change in all directions

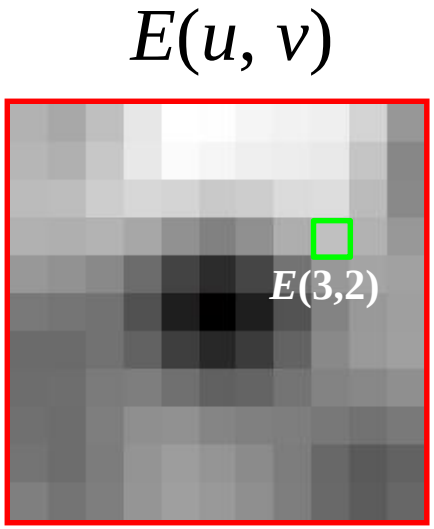
Source: A. Efros

Corner Detection: Mathematics

Change in appearance of window $w(x,y)$ for the shift $[u,v]$:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$





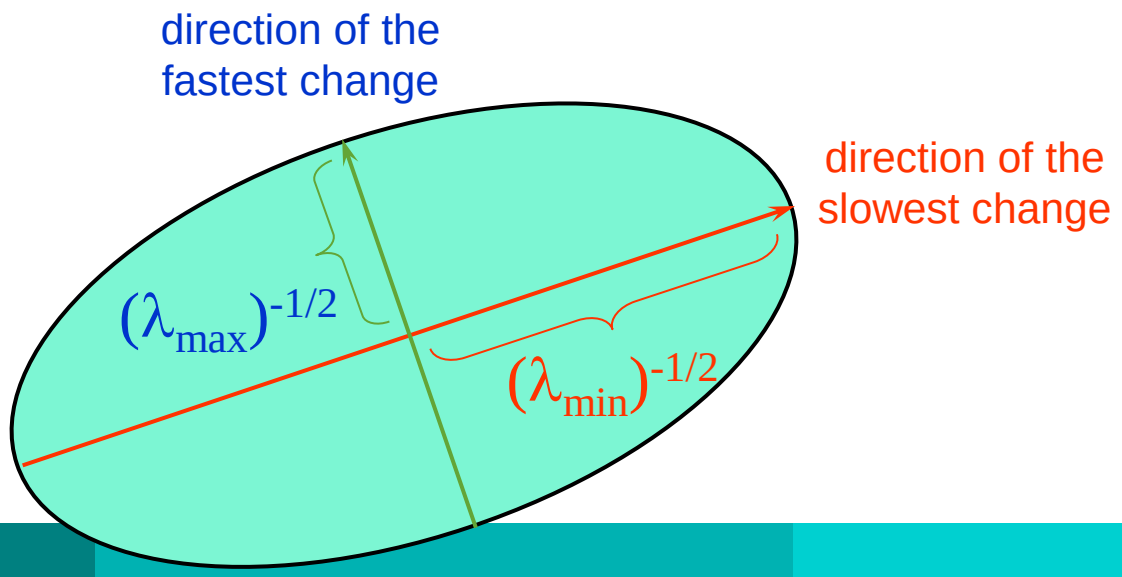
Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $\begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.

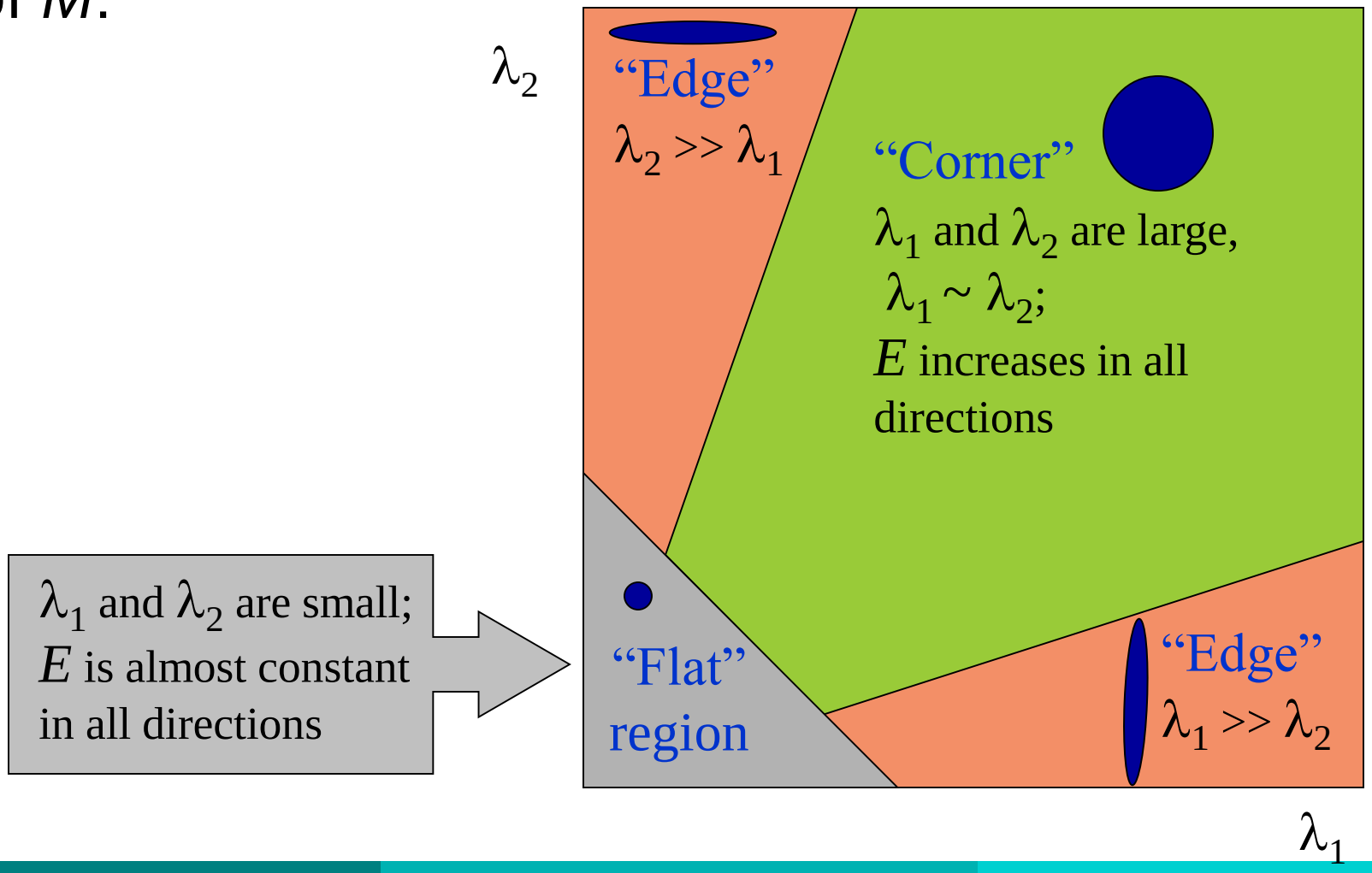
Diagonalization of M: $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by R



Interpreting the eigenvalues

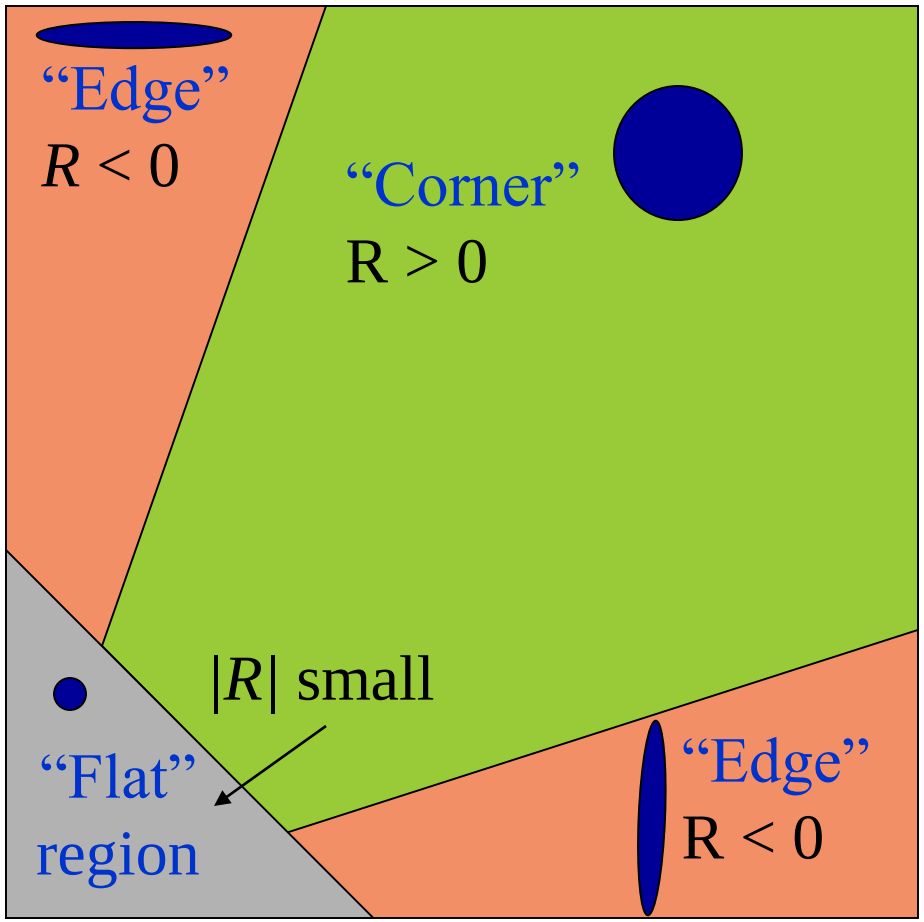
Classification of image points using eigenvalues of *M*:



Corner response function

$$R = \det(M) - \alpha \,\mathrm{trace}(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)



Harris corner detector

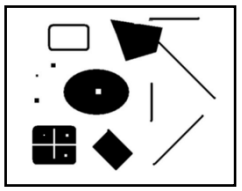
1) Compute M matrix for each image window to get their *cornerness* scores.
2) Find points whose surrounding window gave large corner response (f> threshold)
3) Take the points of local maxima, i.e., perform non-maximum suppression

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

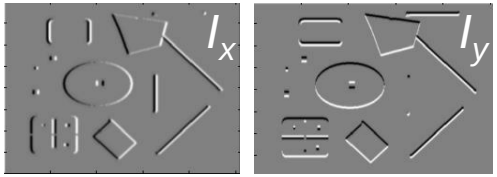
Harris Detector [Harris88]

- Second moment matrix

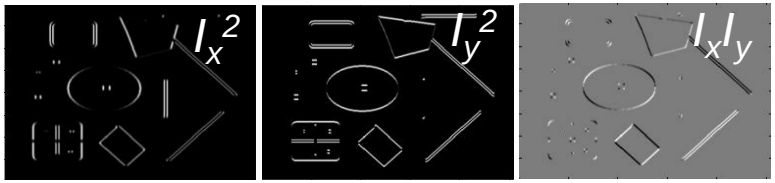
$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



1. Image derivatives (optionally, blur first)



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$

$$\det M = \lambda_1 \lambda_2$$
$$\text{trace}\, M = \lambda_1 + \lambda_2$$

4. Cornerness function – both eigenvalues are strong

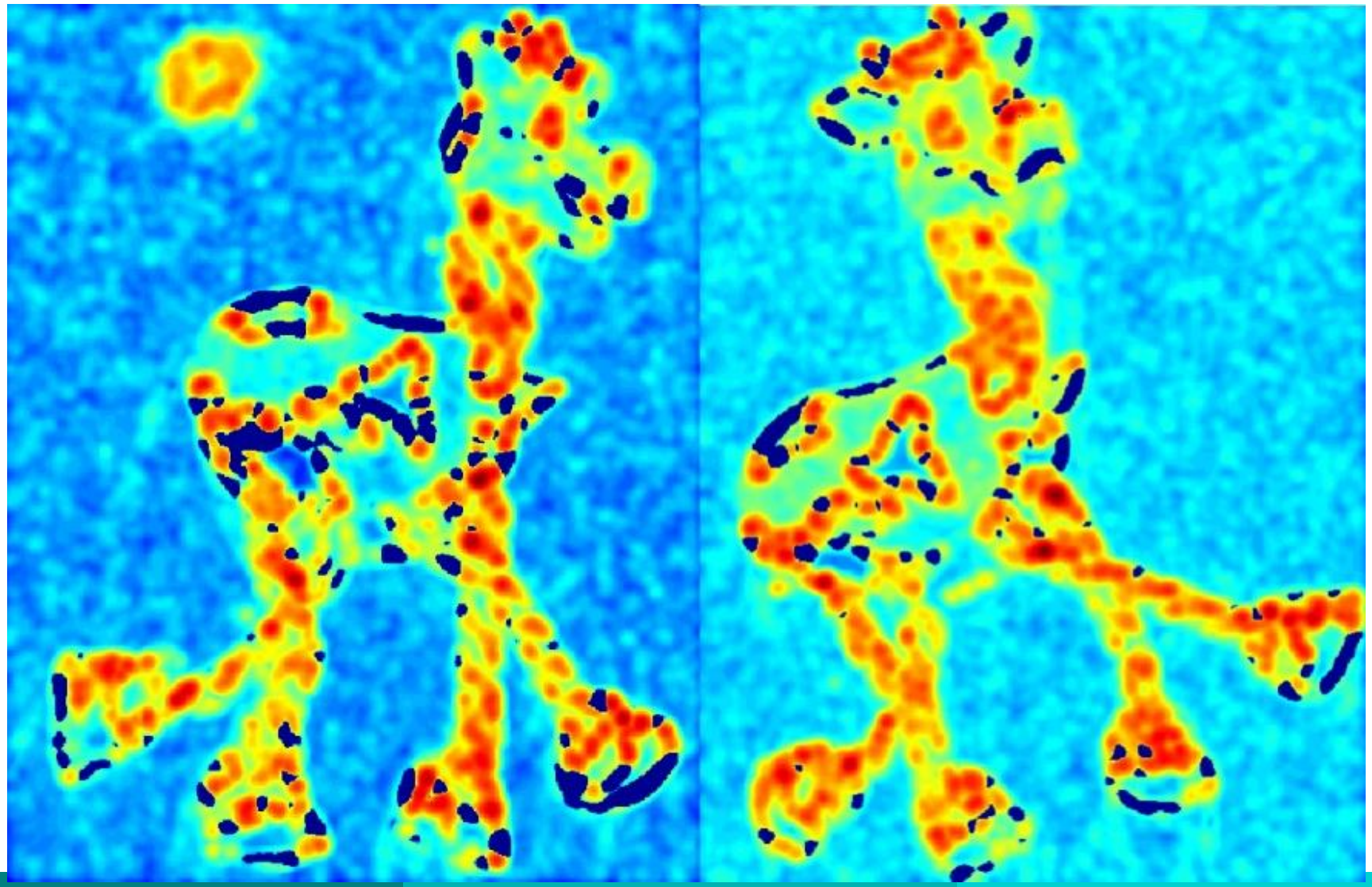
$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I,\sigma_D))^2] =$$
$$g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Non-maxima suppression

Harris Detector: Steps

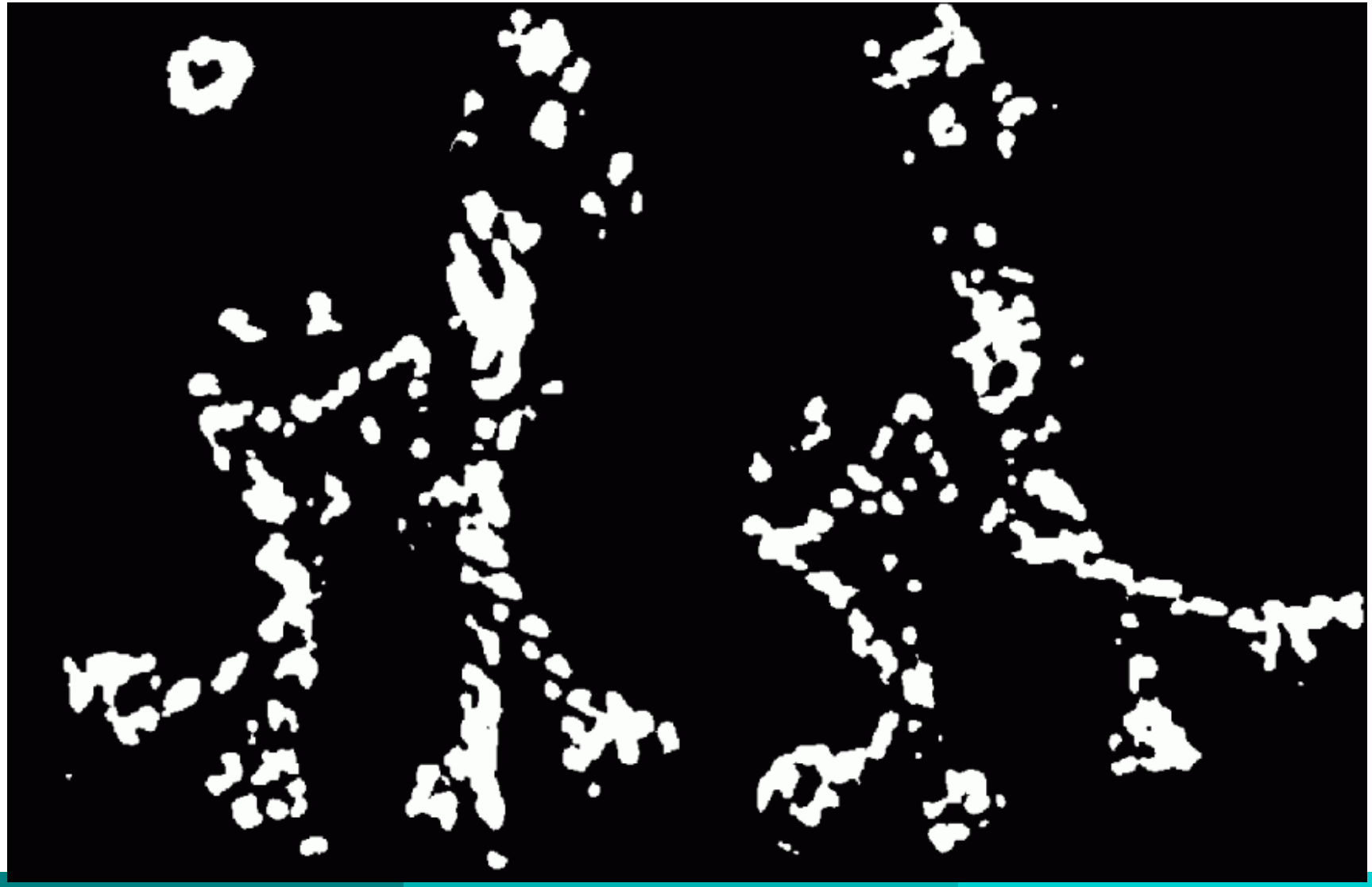
Harris Detector: Steps

Compute corner response R



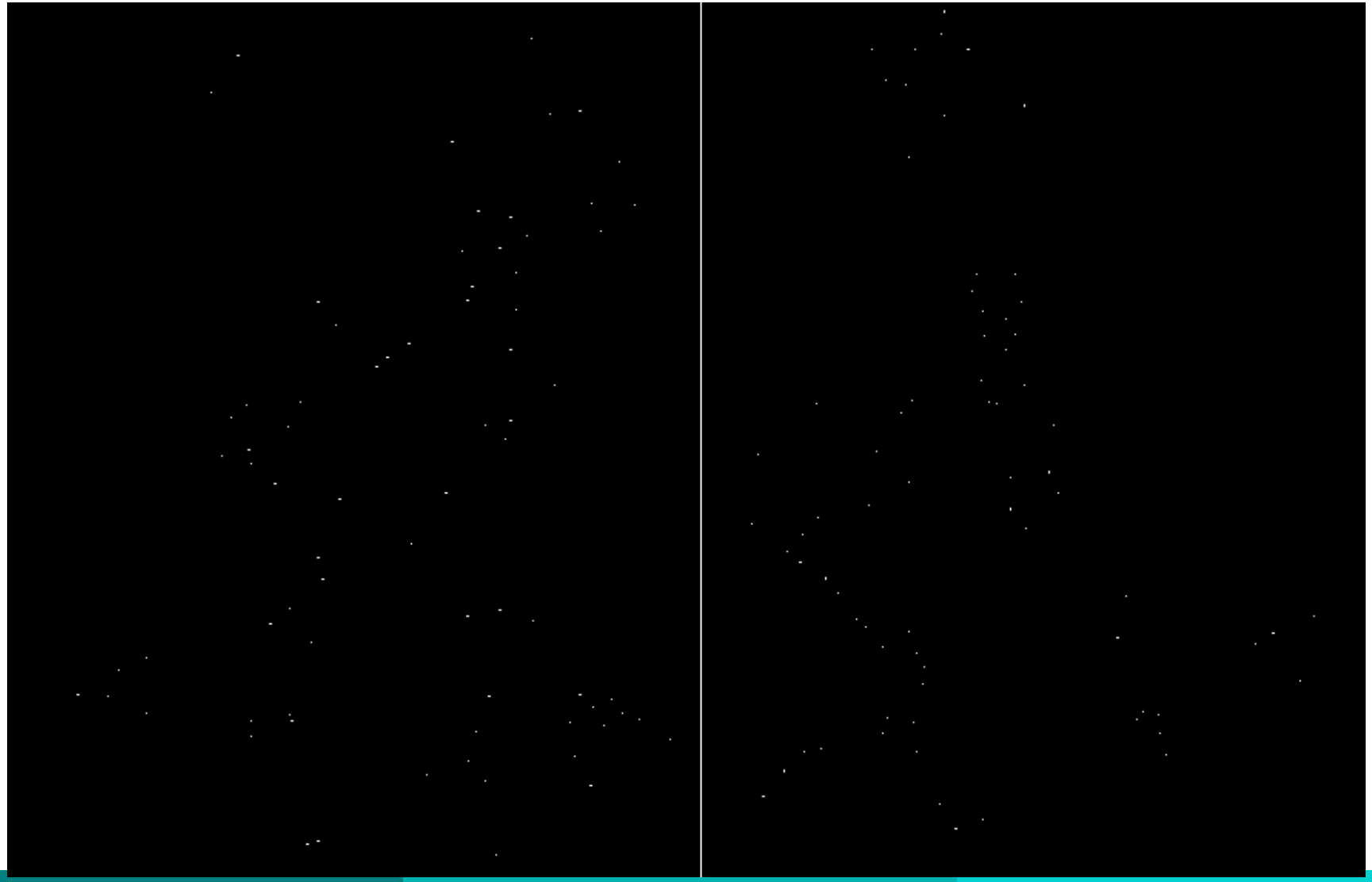
Harris Detector: Steps

Find points with large corner response: R>threshold



Harris Detector: Steps

Take only the points of local maxima of R



Harris Detector: Steps

Conclusion

- “Vision is the act of knowing what is where by looking.” –Aristotle
- The motivation of vision processing is let computer understand the image/video.
- To do it, the image is firstly analyzed to distract the feature points. In this presentation:
  - Basic knowledge for vision processing.
  - Filtering/Matching.
  - Border/Edge detection.
  - Conner detection.
- For more advanced processing, we need further algorithm and analysis.

References

- CS 143 Introduction to Computer Vision - James Hays ; http://cs.brown.edu/courses/cs143/
- http://www.cse.psu.edu/~rtc12/CSE486/
- Computer Vision: Algorithms and Applications - Richard Szeliski http://szeliski.org/Book

Thank you for your attention !

Let’s do Question and Answer !!!