Review on Image Processing

Gang Pan

Zhejiang University

Credit: Jiaya Jia

Image Processing

What is an image?

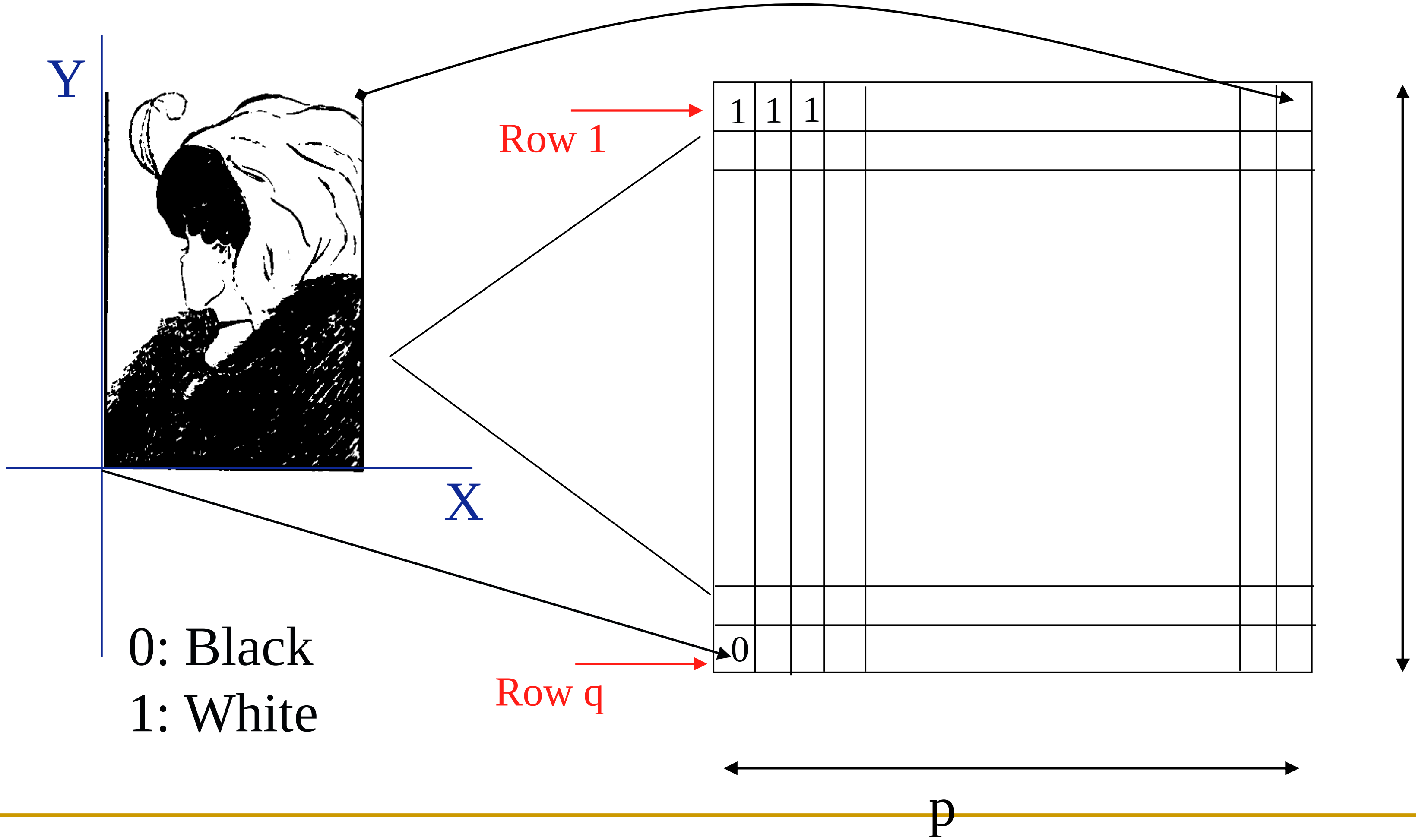
- We can think of an **image** as a function, *f*, from R^2 to R:
  - $f(x, y)$ gives the **intensity** at position (x, y)
  - Realistically, we expect the image only to be defined over a rectangle, with a finite range:
    - $f: [a,b]\text{x}[c,d] \rightarrow [0,1]$
- A color image is just three functions pasted together. We can write this as a "vector-valued" function:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Dealing with Images

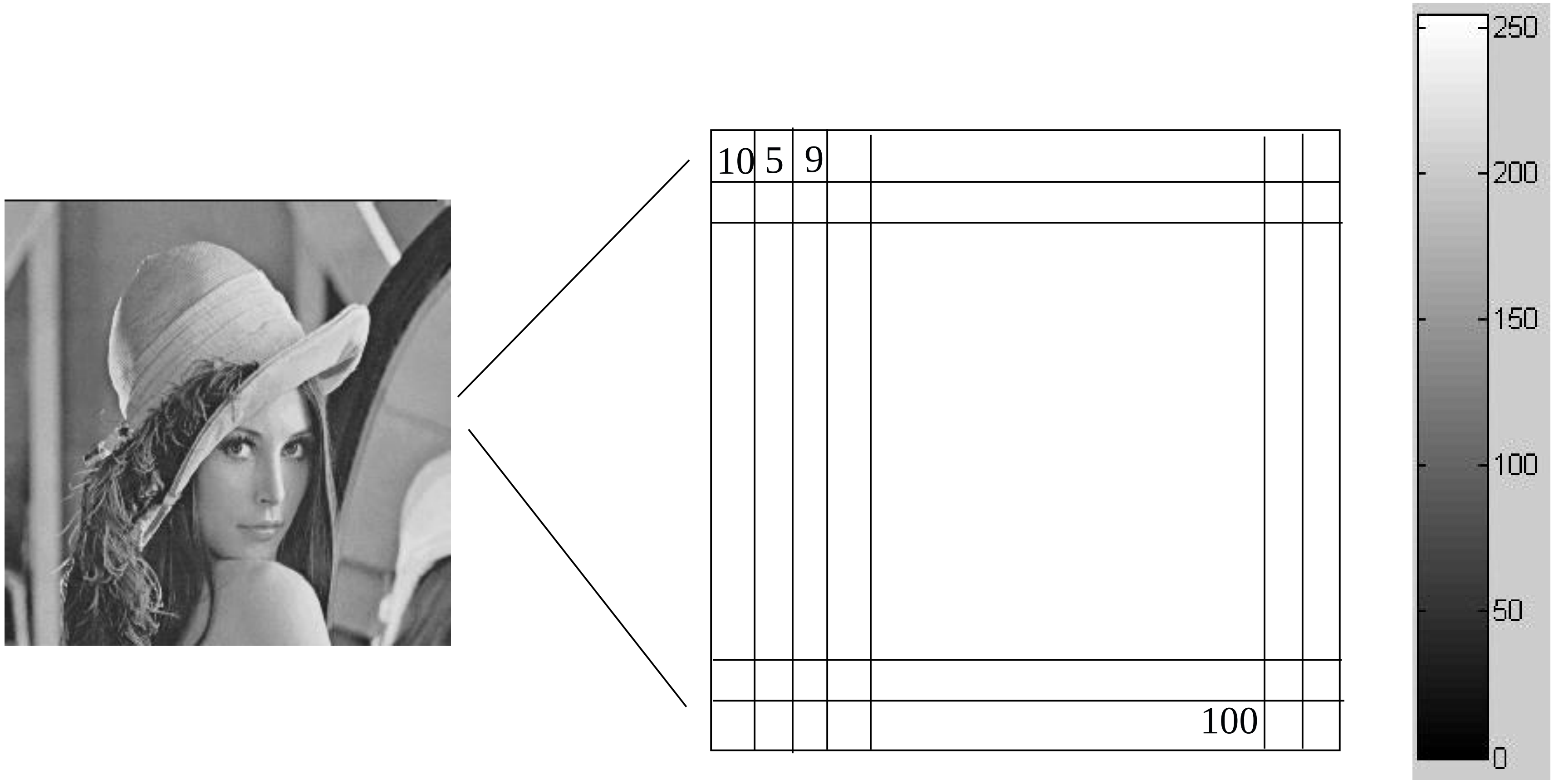
- Binary
- Gray Scale
- Color

Binary Image



0: Black
1: White

Gray Scale Image



Color Image (RGB)

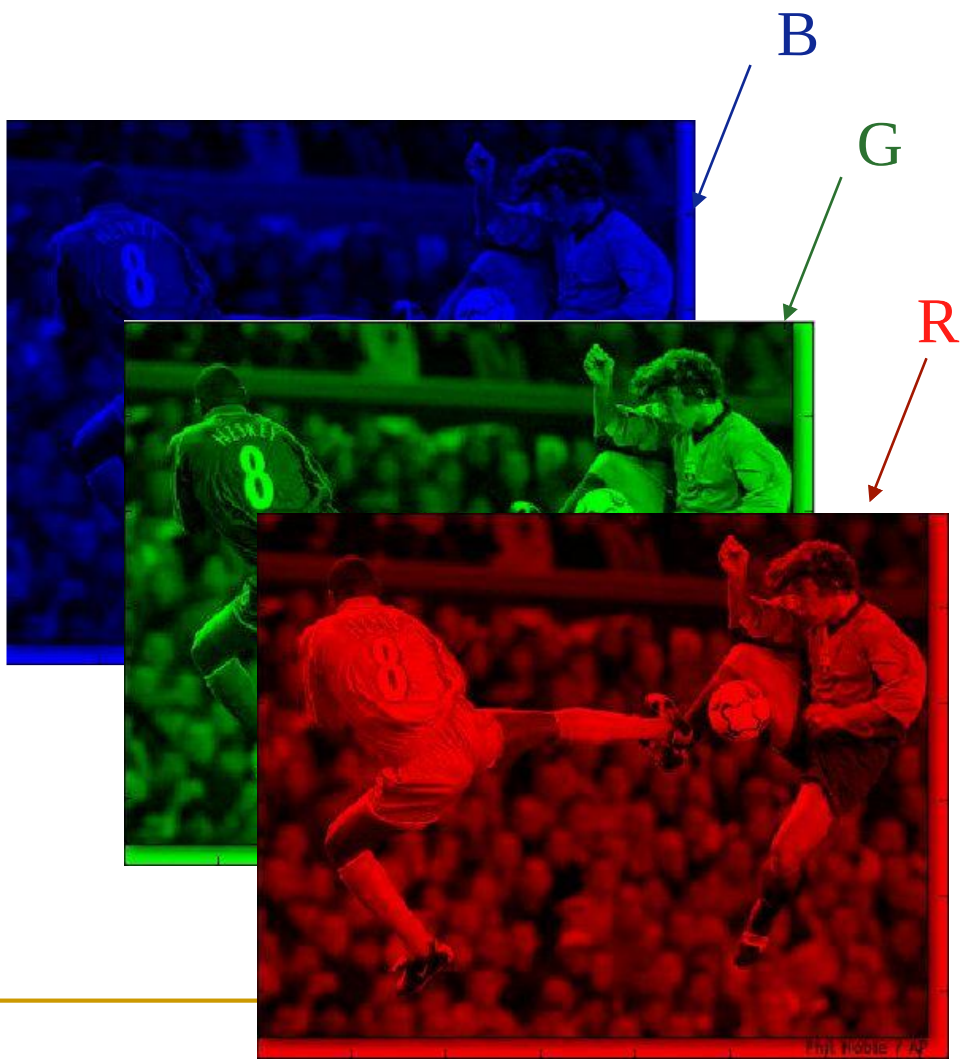
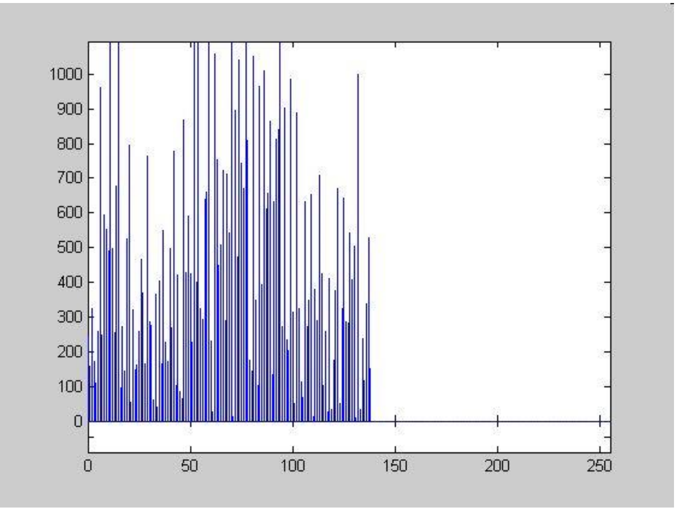


Image Histogram 图像直方图



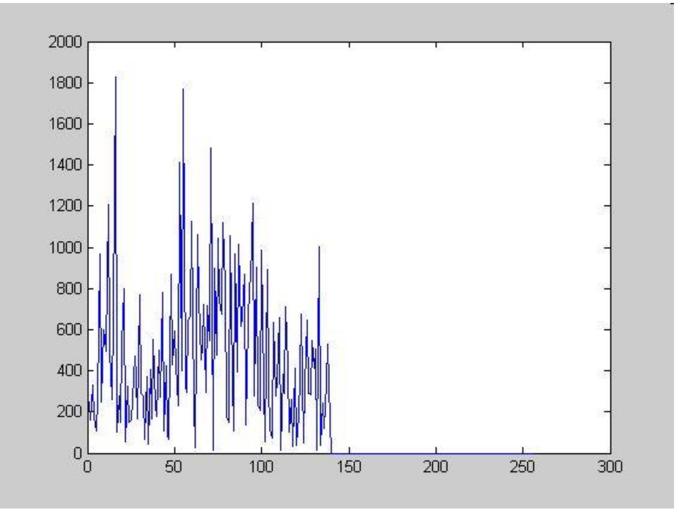


Image Noise

- Light Variations
- Camera Electronics
- Surface Reflectance
- Lens

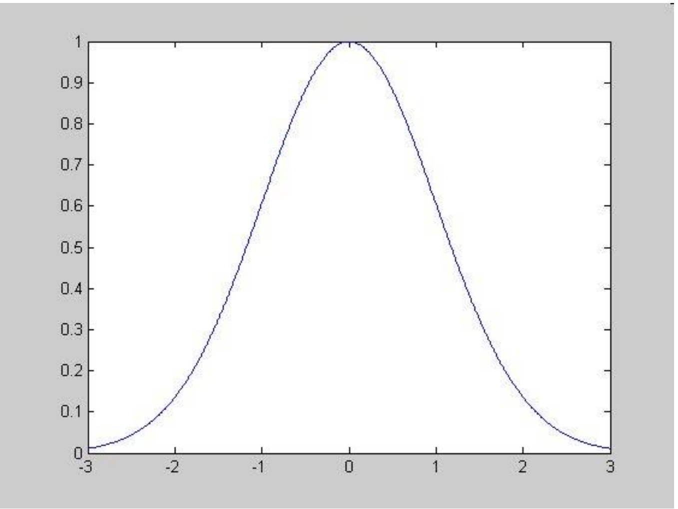
Image Noise

- Let *I(i,j)* be the true pixel values and *n(i,j)* be the noise added to the pixel *(i,j)*

$$\hat{I}(i,j) = I(i,j) + n(i,j) \qquad \text{(Additive Noise)}$$

Gaussian Noise (White Noise)

$$n(i,j) = e^{\frac{-x^2}{2\sigma^2}}$$



Salt and Pepper Noise

$$\hat{I}(i,j) = \begin{cases} I(i,j) & p < l \\ s_{\min} + q(s_{\max} - s_{\min}) & p \geq l \end{cases}$$

$$p, q \in [0,1] \text{ (Uniformly distributed random variables)}$$

$$l = \text{Threshold}$$

Image Processing

- An **image processing** operation typically defines a new image g in terms of an existing image f.
- We can transform either the range of f.

$$g(x, y) = t(f(x, y))$$

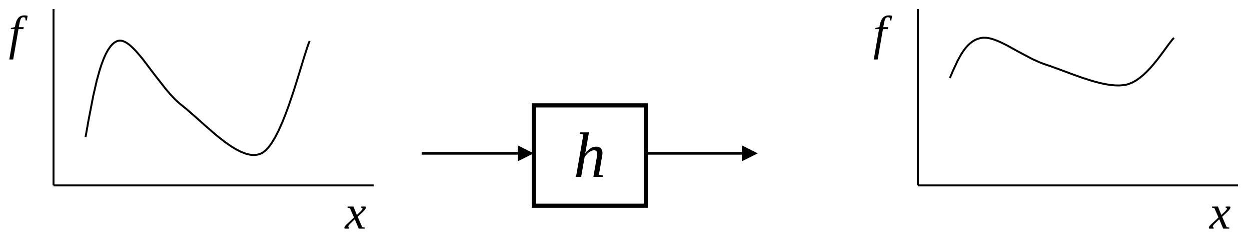
- Or the domain of f:

$$g(x, y) = f(t_x(x, y), t_y(x, y))$$

- What kinds of operations can each perform?

Image Processing

- image filtering: change ***range*** of image
  - *g(x) = h(f(x))*



- image warping: change ***domain*** of image
  - *g(x) = f(h(x))*

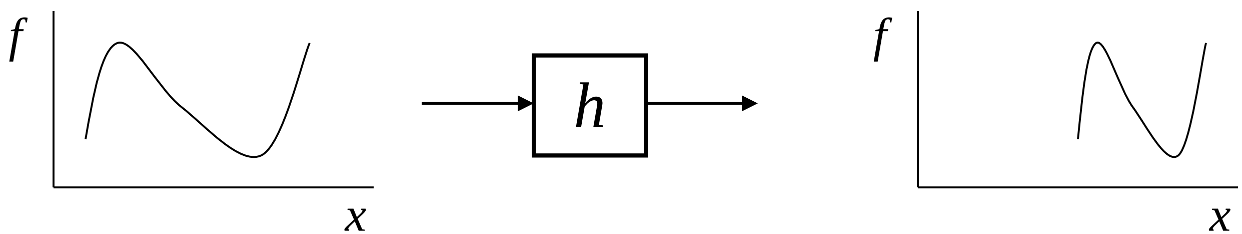
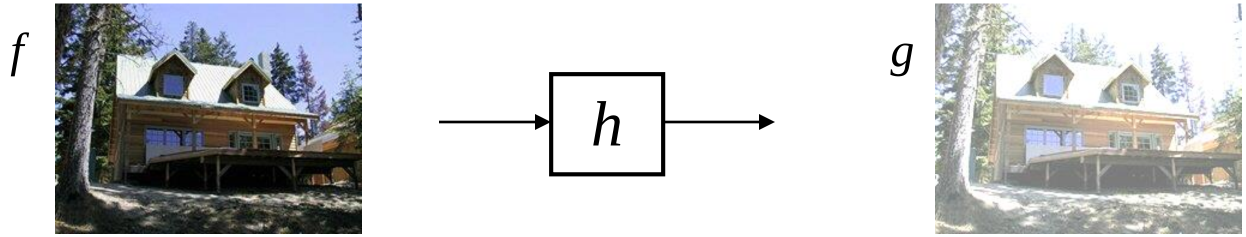
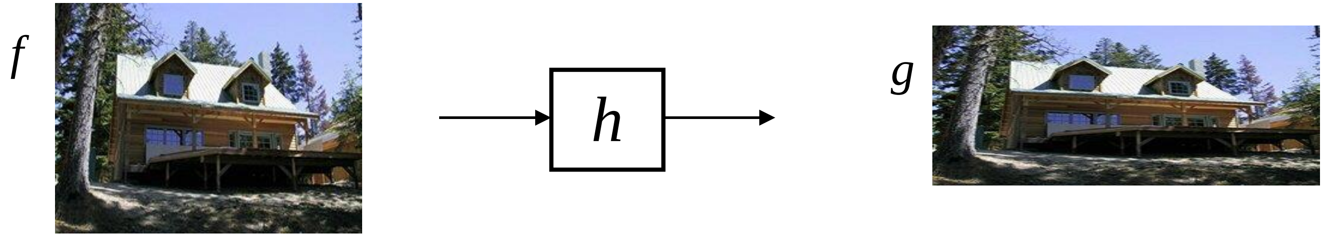


Image Processing

- image filtering: change ***range*** of image
  - $g(x) = h(f(x))$



- image warping: change ***domain*** of image
  - $g(x) = f(h(x))$



Point Processing

- The simplest kind of range transformations are these independent of position x,y:

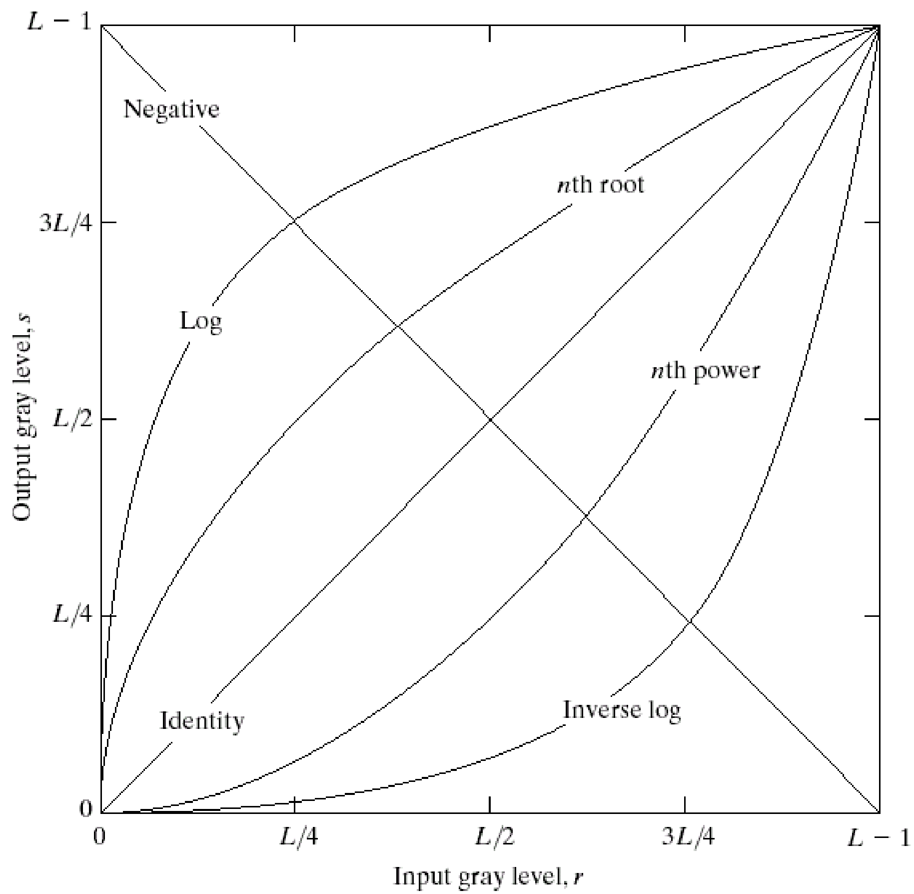
$$g = t(f)$$

- This is called **point processing**.

- **Important:** every pixel for himself – spatial information is completely lost!

Basic Point Processing

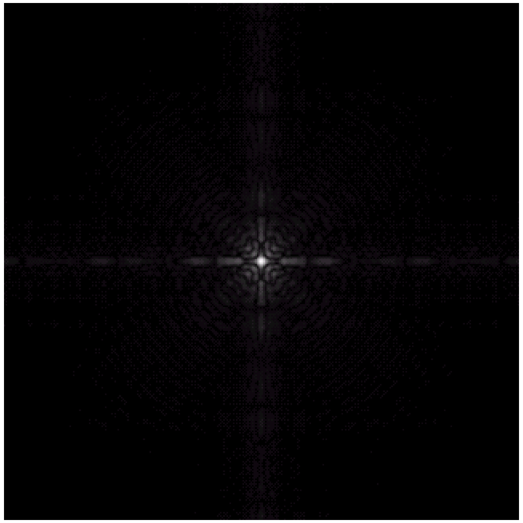
FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Log

a b

FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Power-law transformations

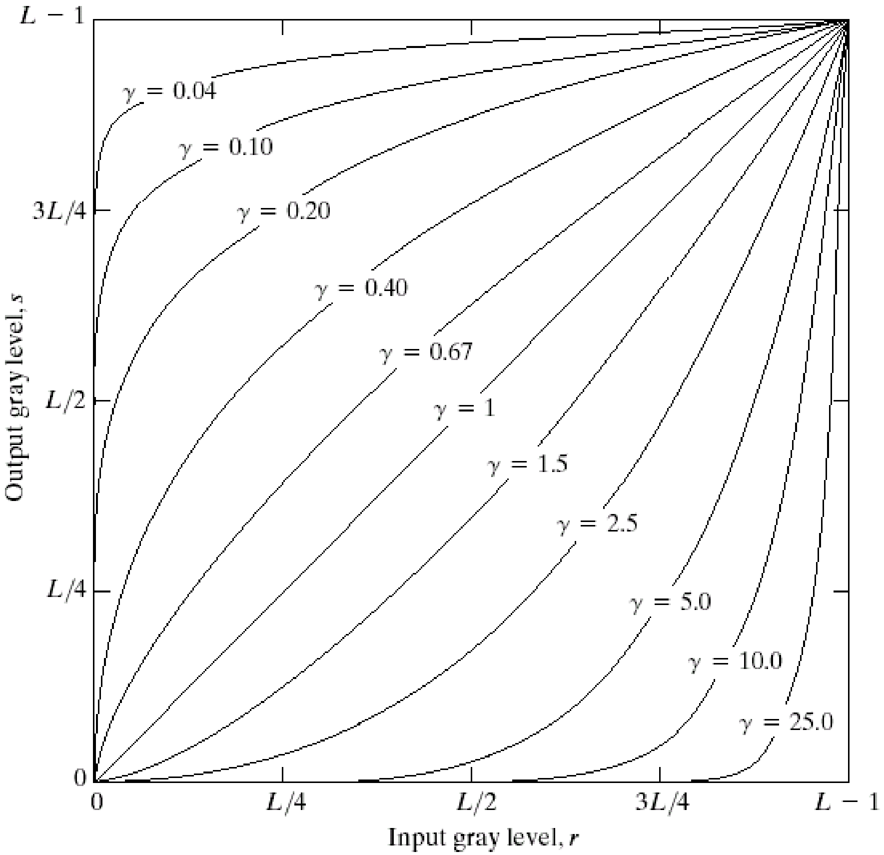


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

$$s = cr^{\gamma}$$

Image Enhancement

a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Example: Gamma Correction

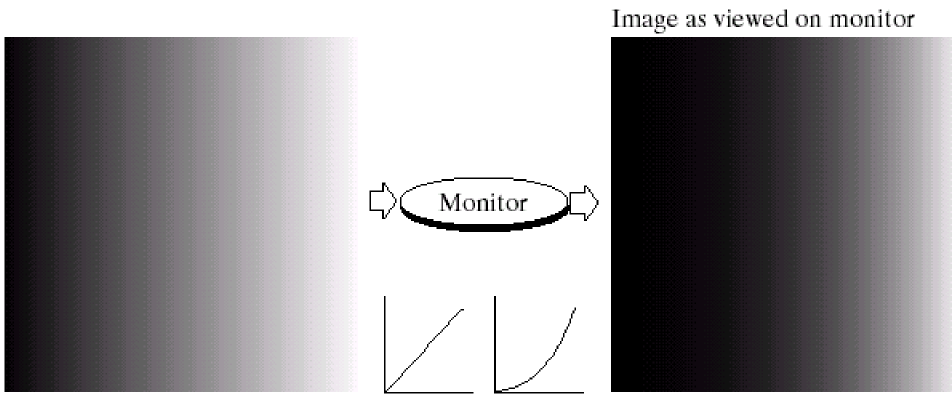


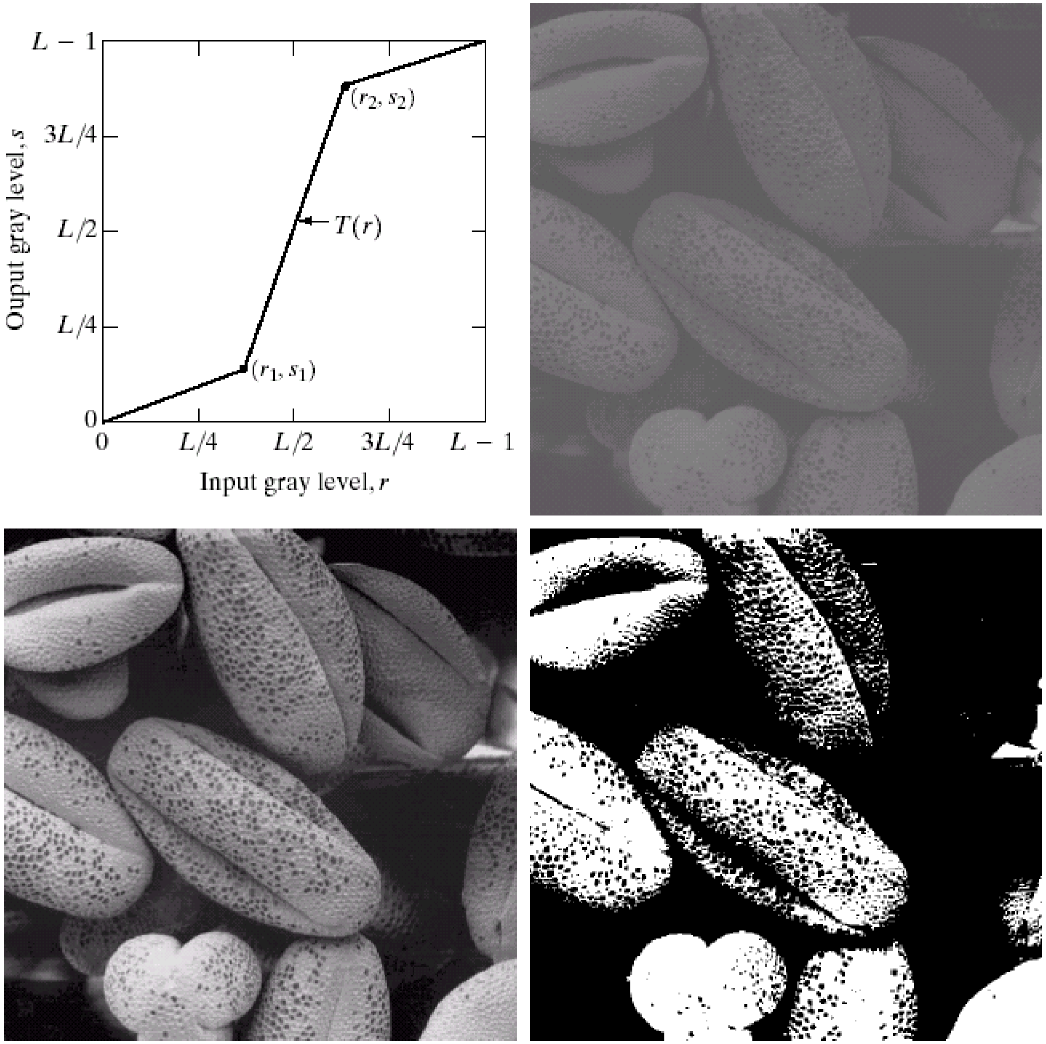
FIGURE 3.7 (a) Linear-wedge gray-scale image. (b) Response of monitor to linear wedge. (c) Gamma-corrected wedge. (d) Output of monitor.

$$s = r^{\gamma}$$

$$e.g.\quad 0.25 = 0.5^{2.0}$$

http://www.cs.berkeley.edu/~efros/java/gamma/gamma.html

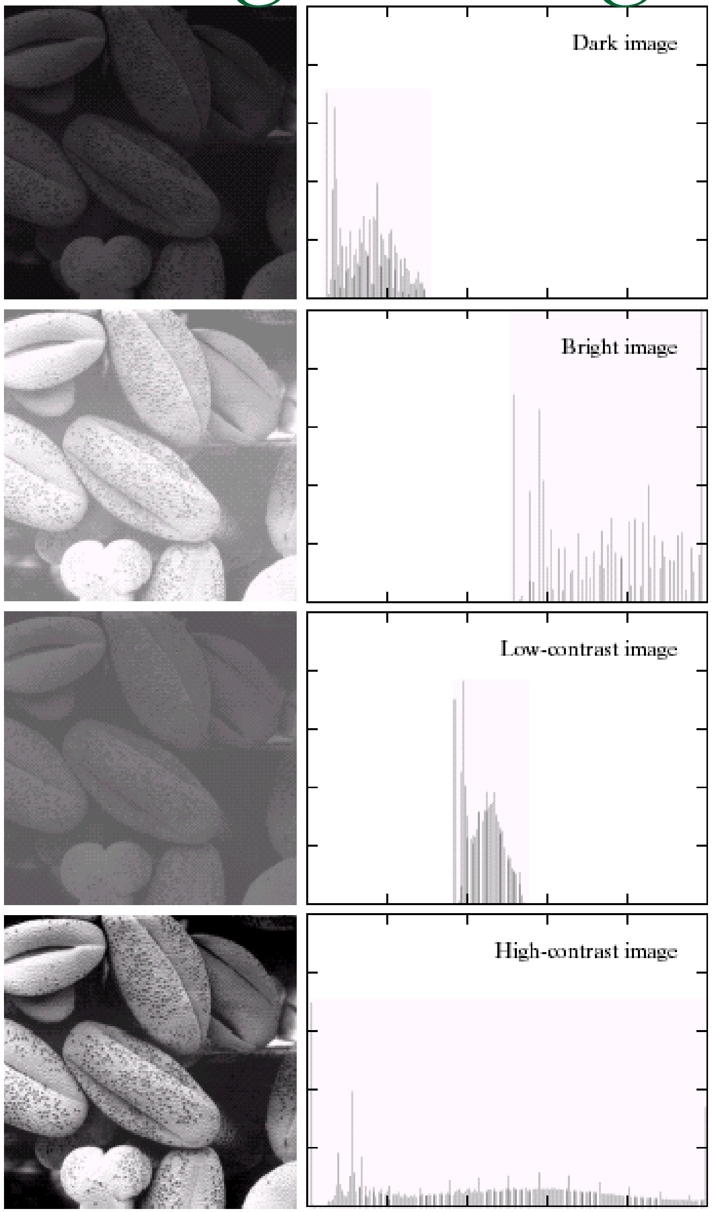
Contrast Stretching

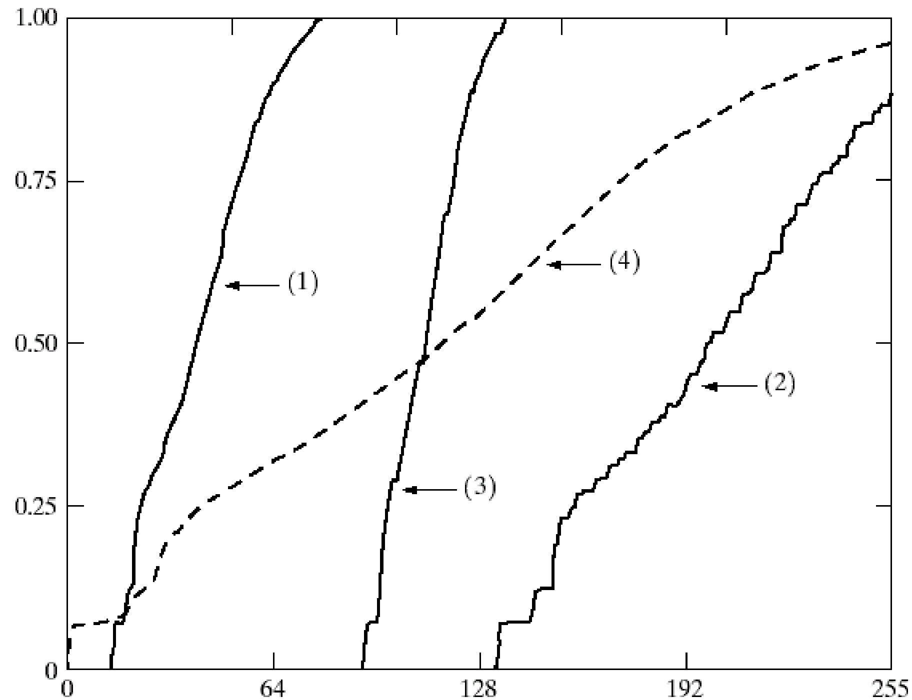


a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Image Histograms





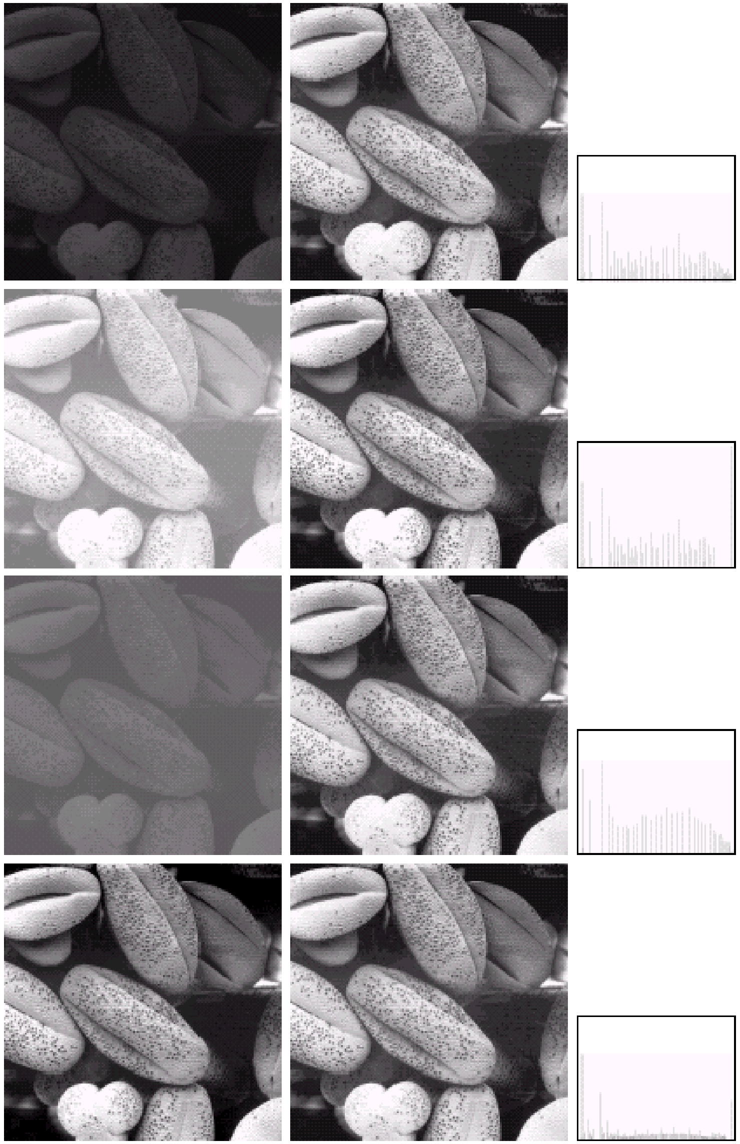
Cumulative Histograms 累积直方图

$$s = \sum_{0}^{r} h(r)$$

a b

FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Histogram Equalization (直方图均衡化)

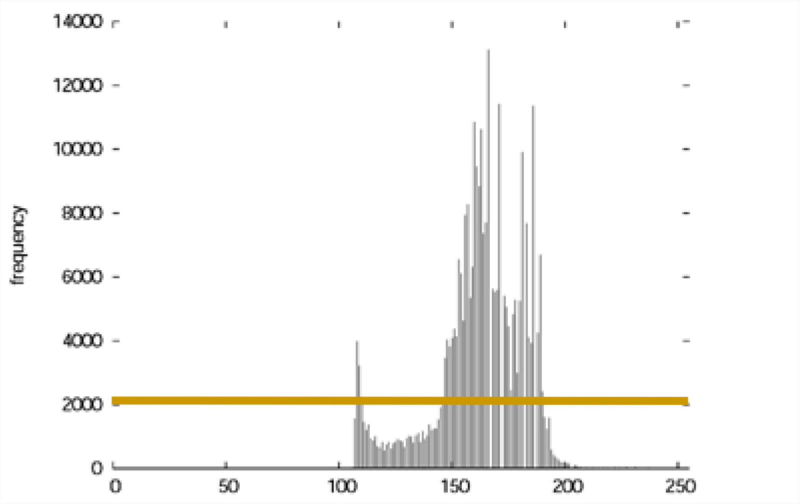


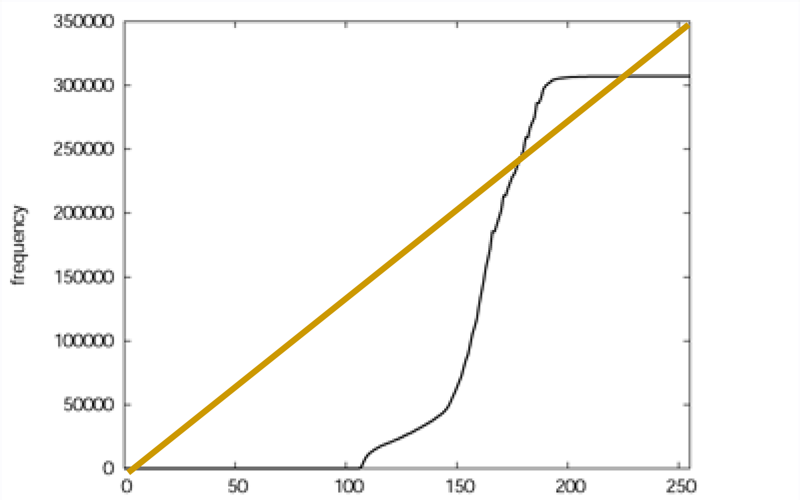
a b c

FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

Histogram Equalization

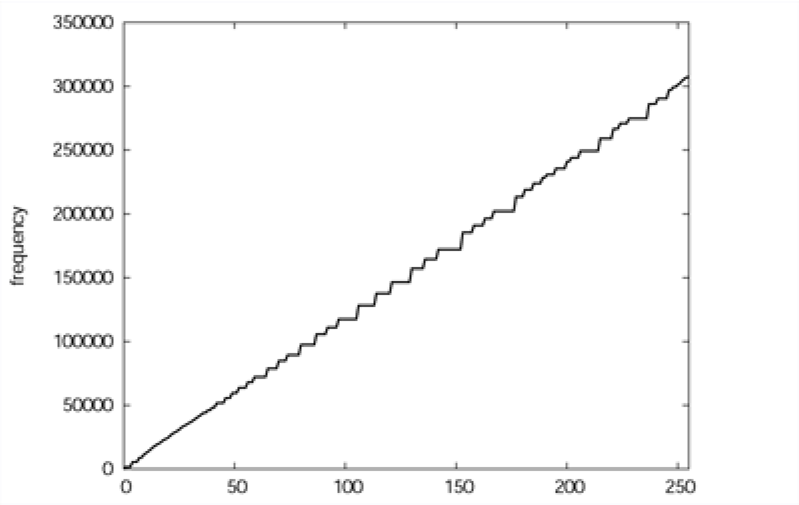
- Note how the image is extremely *grey*; it lacks detail since the range of colours seems limited to mid grey-levels. We can verify this by looking at the image histogram.
- W calculate the *cumulative frequencies* within the image. The cumulative frequency for grey level *g* is defined as the sum of the histogram data 0 to *g*. We can graph the cumulative frequencies for our image:

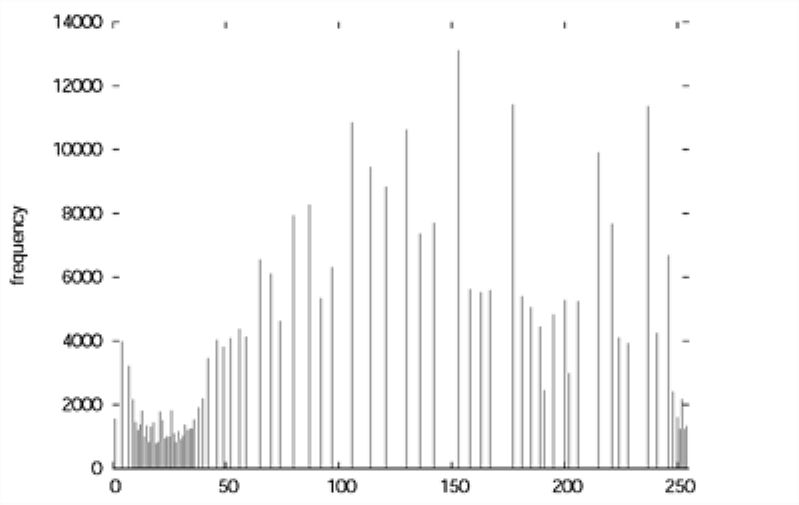




Histogram Equalization

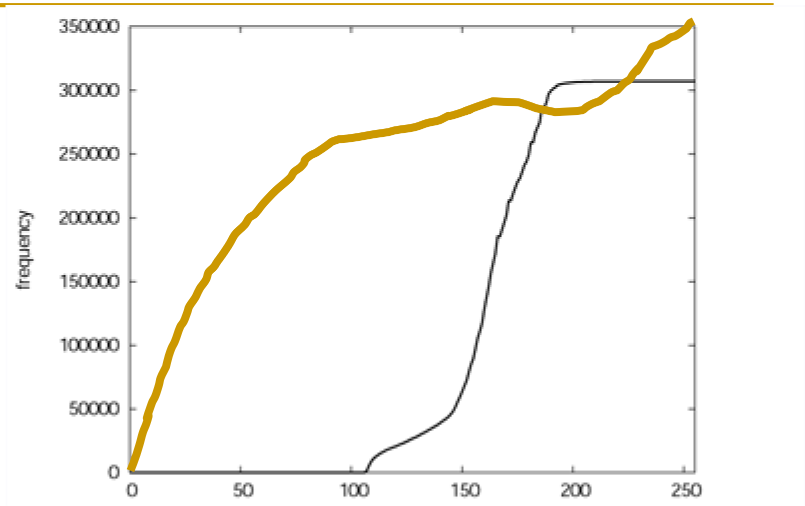
- alpha = 255 / *numPixels*
- for each *pixel*
  - g(x,y) = cumulativeFrequency[f(x,y)] * alpha
- end for

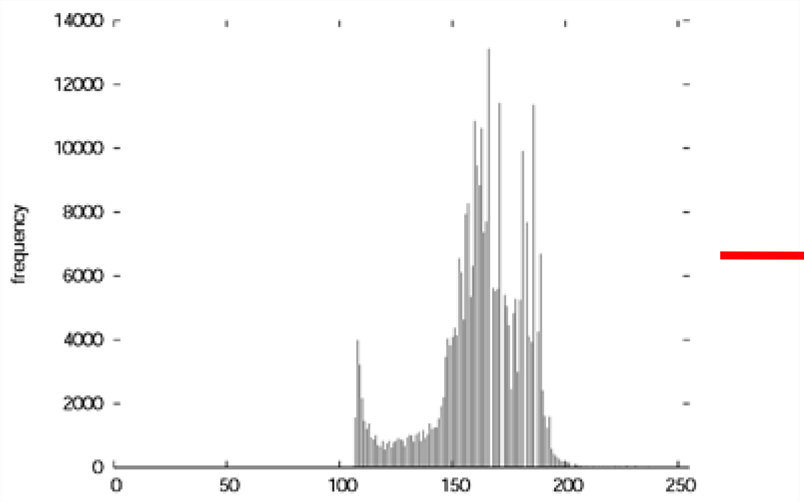


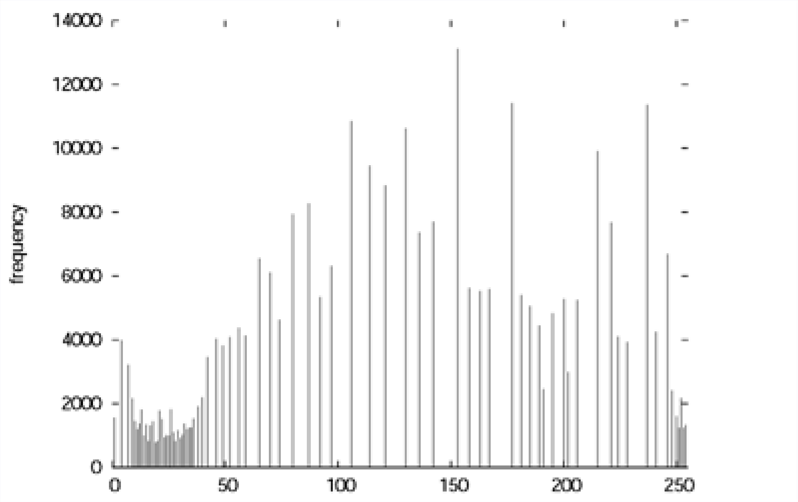


More fun

- Color transfer?







R, G, B of one image

R, G, B of another image

More fun

Linear Filtering

- The output is the linear combination of the neighborhood pixels

$$f(p) = \sum_{q_i \in N(p)} a_i q_i$$

- The coefficients of this linear combination combine to form the "filter-kernel"

1	3	0
2	10	2
4	1	1

Image

$\otimes$

1	0	-1
1	0.1	-1
1	0	-1

Kernel

=

	5	

Filter Output

Convolution

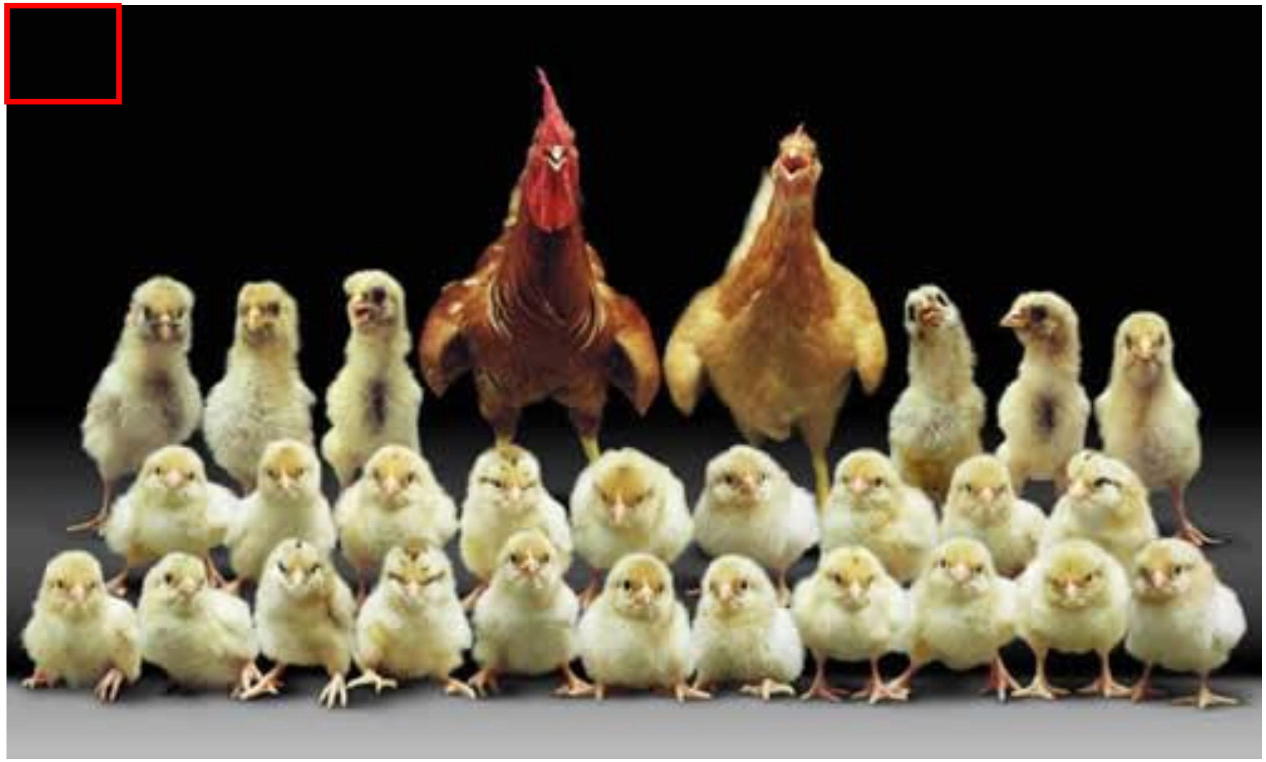
34	20	10	30	38	198	246
28	45	0	1	4	9	2
0	9	0	0	0	2	0
238	5	5	2	9	3	9
8	98	1	8	2	8	2
2	5	4	7	1	6	2
9	3	6	5	3	1	4

Image

-1	1	-1
1	4	1
-1	1	-1

Kernel

Convolution



Convolution

$$f(i,j) = I * H = \sum_k \sum_l I(k,l)\,H(i-k, j-l)$$

I = Image

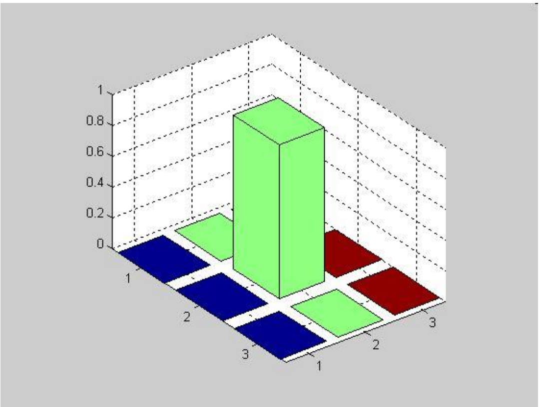
H = Kernel

$$H = \begin{bmatrix} H_1 & H_2 & H_3 \\ H_4 & H_5 & H_6 \\ H_7 & H_8 & H_9 \end{bmatrix} \xrightarrow{X-flip} \begin{bmatrix} H_7 & H_8 & H_9 \\ H_4 & H_5 & H_6 \\ H_1 & H_2 & H_3 \end{bmatrix} \xrightarrow{Y-flip} \begin{bmatrix} H_9 & H_8 & H_7 \\ H_6 & H_5 & H_4 \\ H_3 & H_2 & H_1 \end{bmatrix}$$

$$I = \begin{bmatrix} I_1 & I_2 & I_3 \\ I_4 & I_5 & I_6 \\ I_7 & I_8 & I_9 \end{bmatrix} \otimes \begin{bmatrix} H_9 & H_8 & H_7 \\ H_6 & H_5 & H_4 \\ H_3 & H_2 & H_1 \end{bmatrix} \longrightarrow$$

$$\begin{aligned} I * H = {} & I_1 H_9 + I_2 H_8 + I_3 H_7 \\ & + I_4 H_6 + I_5 H_5 + I_6 H_4 \\ & + I_7 H_3 + I_8 H_2 + I_9 H_1 \end{aligned}$$

Linear Filtering

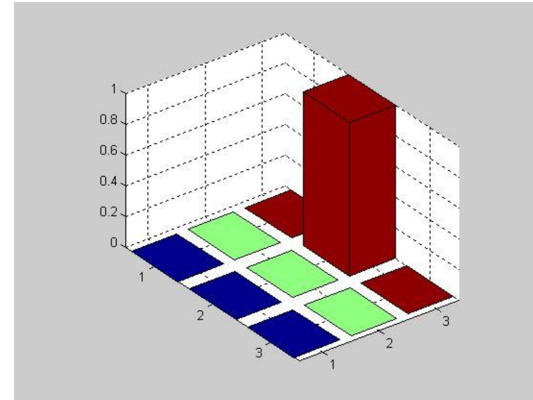


*

0	0	0
0	1	0
0	0	0

=

Linear Filtering



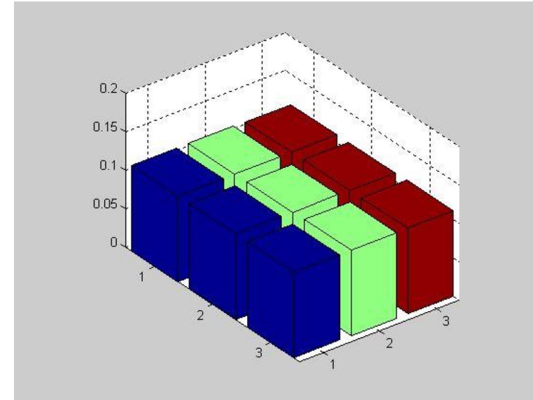
*

0	0	0
0	0	1
0	0	0

=

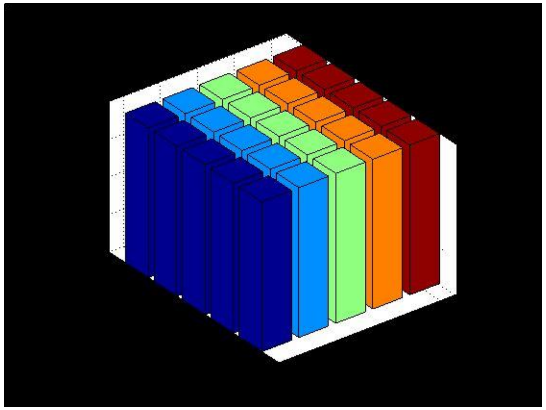
想左移动一个像素单位

Linear Filtering



$$* \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} =$$

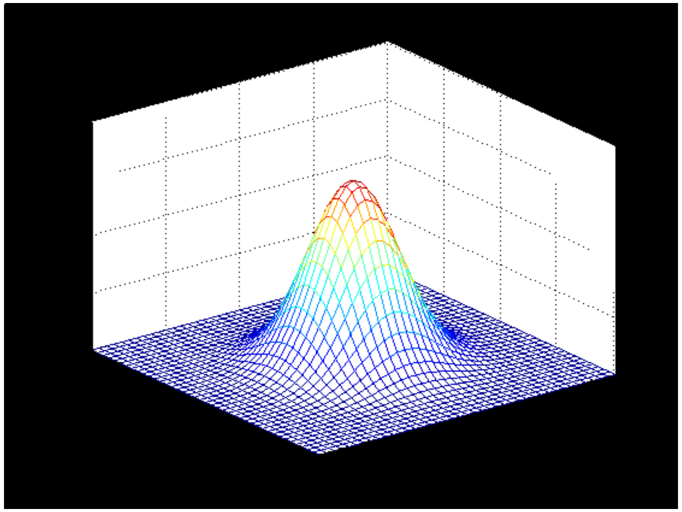
Linear Filtering



$$* \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} =$$

Gaussian Filter

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

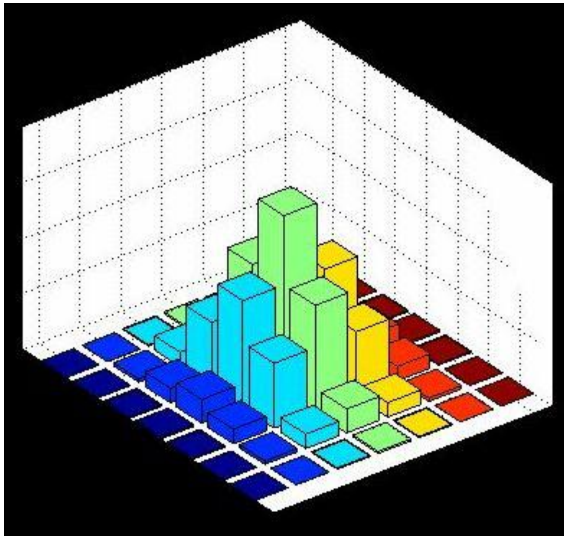


$$H(i,j)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left((i-k-1)^2+(j-k-1)^2\right)}{2\sigma^2}\right)$$

where $H(i,j)$ is $(2k+1)\times(2k+1)$ array

Linear Filtering (Gaussian Filter)

*



=

Gaussian Vs Average

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Smoothing by Averaging

Gaussian Smoothing

Noise Filtering

Gaussian Noise

After Averaging

After Gaussian Smoothing

Noise Filtering

Salt & Pepper Noise

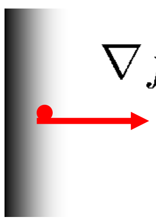
After Averaging

After Gaussian Smoothing

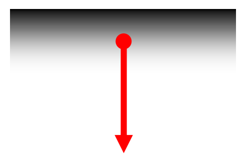
Image gradient

- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

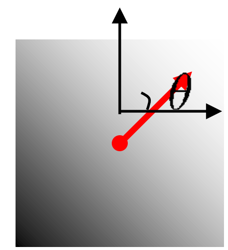
- The gradient points in the direction of most rapid change in intensity



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

- The *gradient direction* is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

  - how does this relate to the direction of the edge?

- The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

First-order derivative

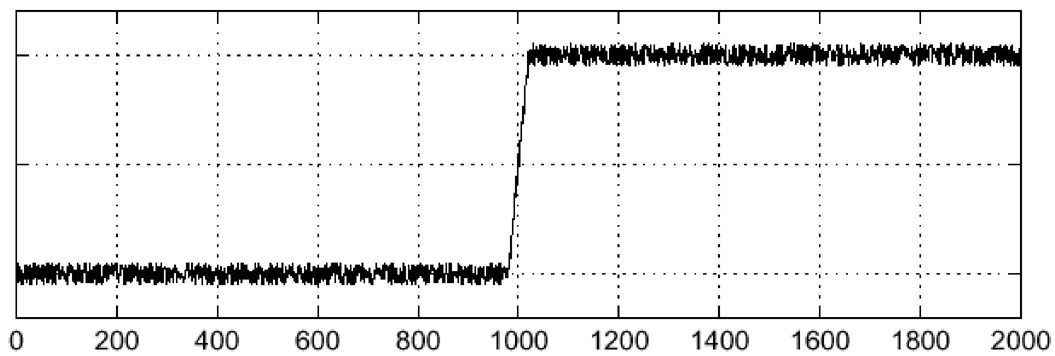
- a basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

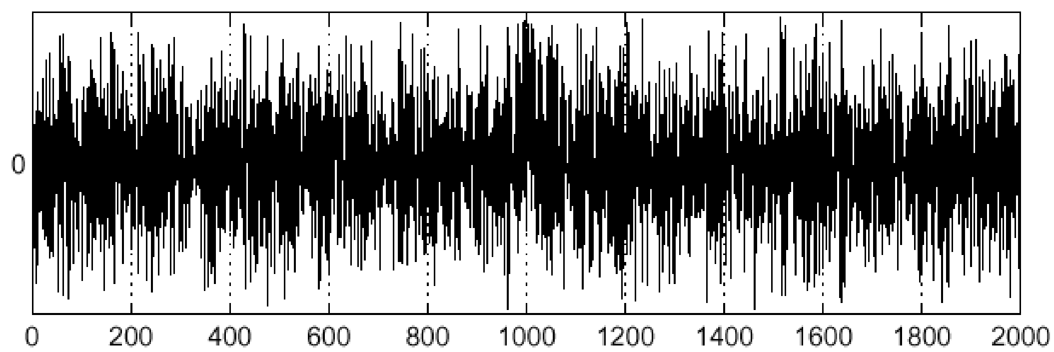
Effects of noise

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

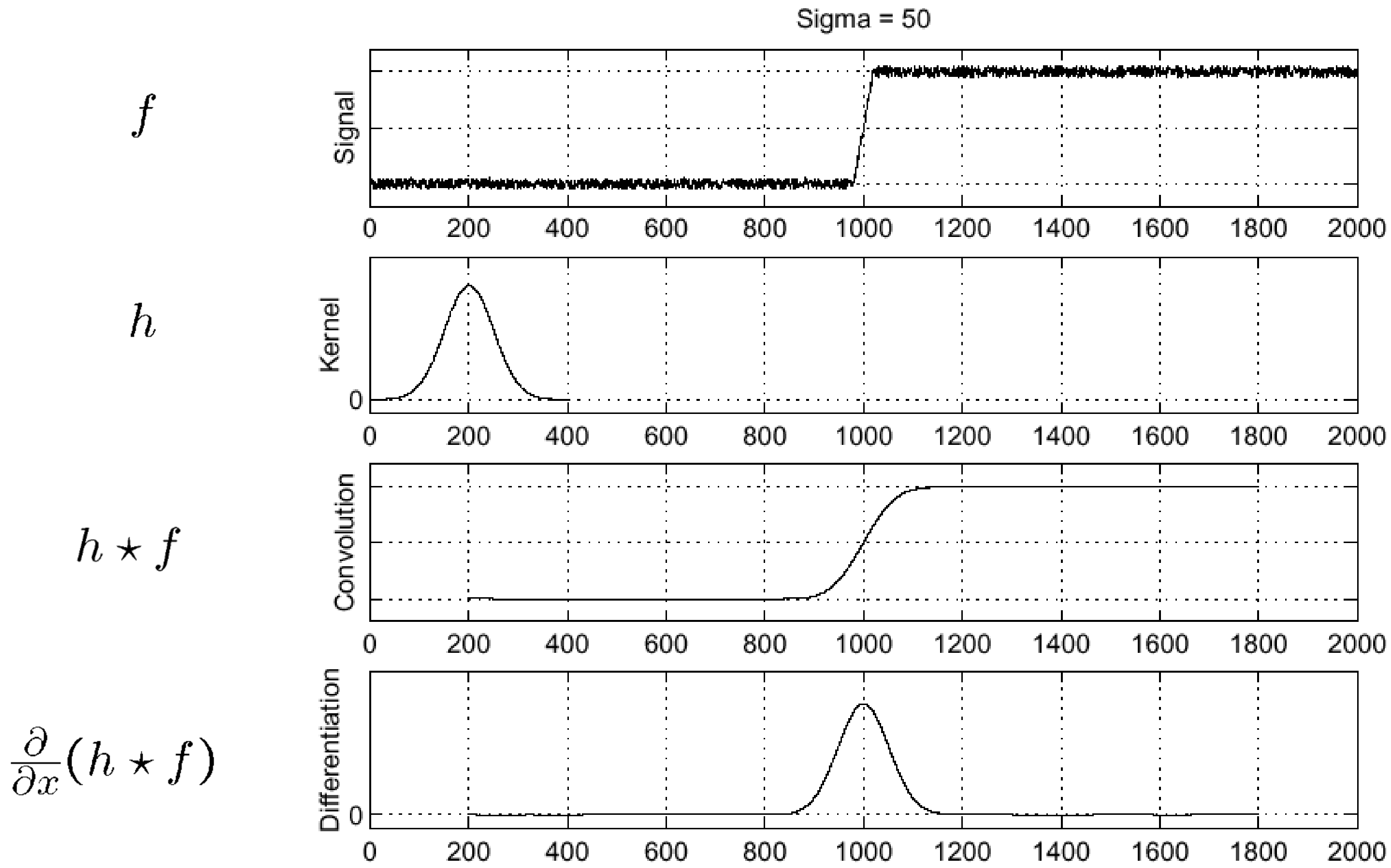
$$f(x)$$



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



Solution: smooth first



- Where is the edge?
- Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative theorem of convolution

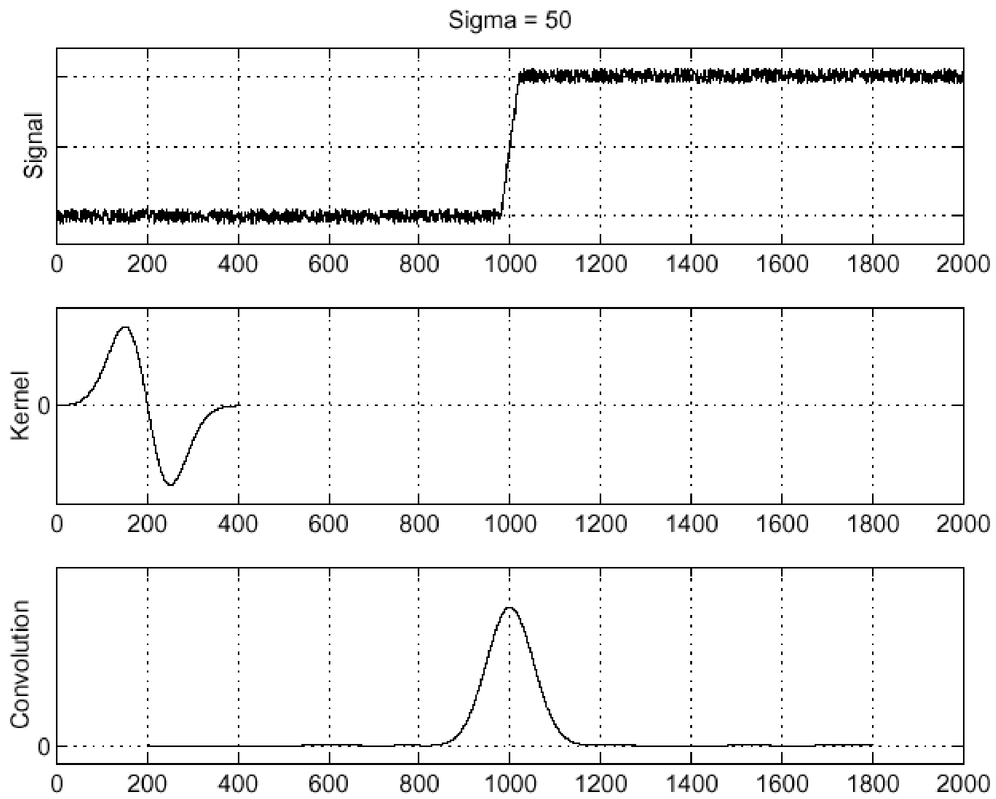
- This saves us one operation:

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$



Second-order derivative

- similarly, we define the second-order derivative of a one-dimensional function f(x)

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

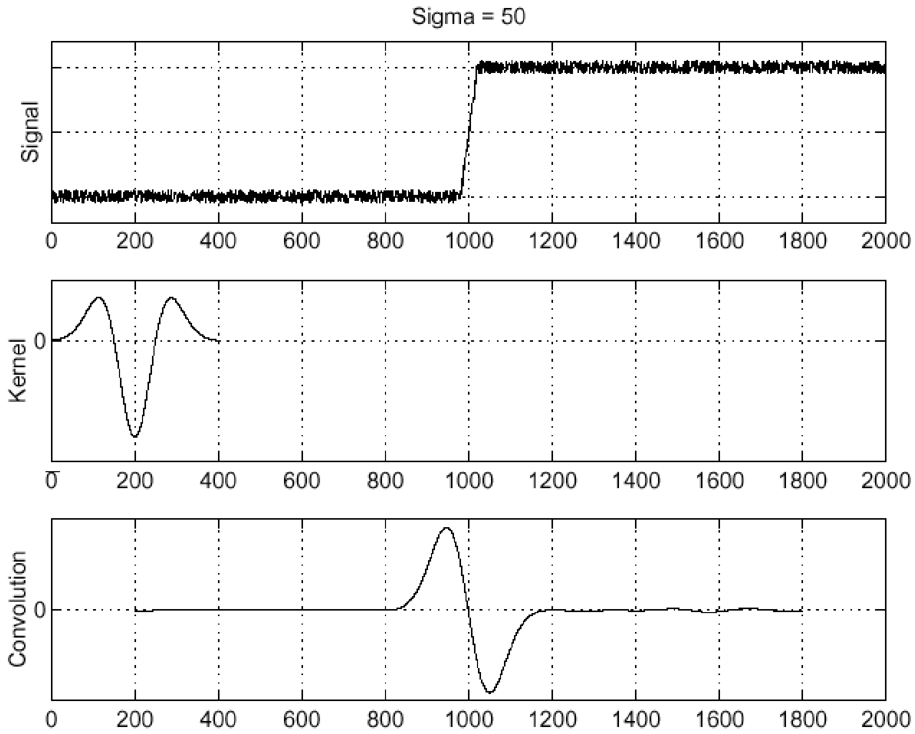
Laplacian of Gaussian

- Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

f

$\frac{\partial^2}{\partial x^2}h$

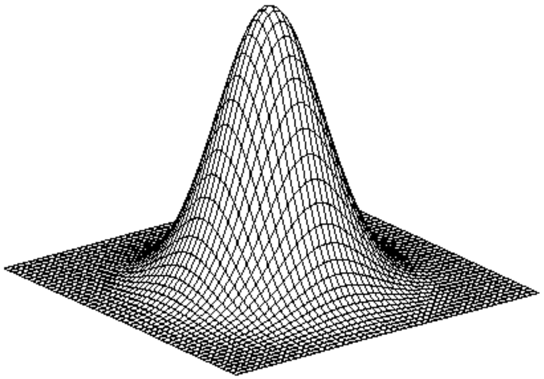
$(\frac{\partial^2}{\partial x^2}h) \star f$



Laplacian of Gaussian operator

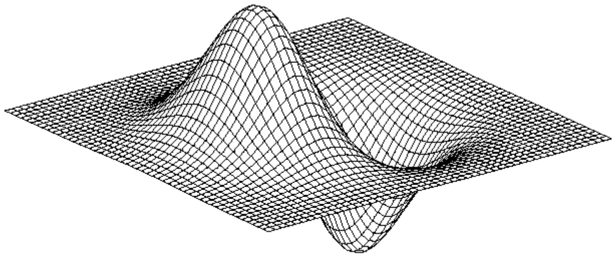
- Where is the edge?
- Zero-crossings of bottom graph

2D edge detection filters



Gaussian

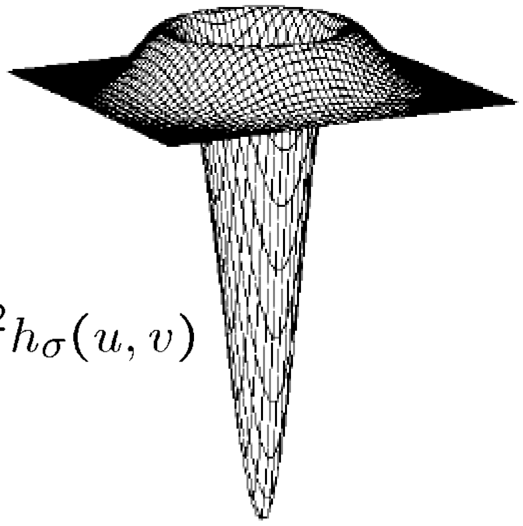
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

- ∇^2 is the **Laplacian** operator:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

First and Second-order derivative of f(x,y)

- Laplacian operator

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)]$$

What is Laplacian masks

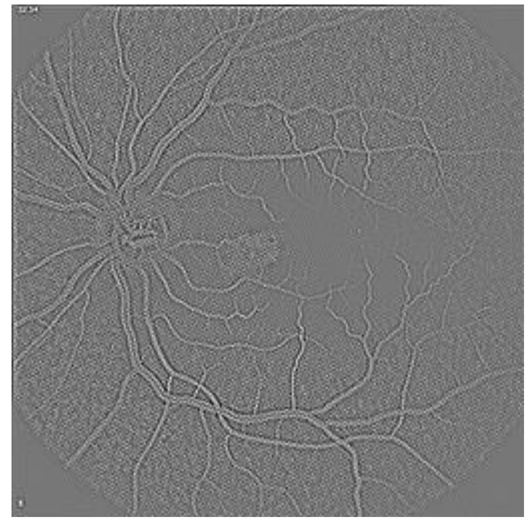
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

- as it is a derivative operator,
  - it highlights gray-level discontinuities in an image
  - it deemphasizes regions with slowly varying gray levels
- tends to produce images that have
  - grayish edge lines and other discontinuities, all superimposed on a dark featureless background.

Sharpening

+



=

Unsharp masking

$$f_s(x,y) = f(x,y) - \bar{f}(x,y)$$

Mask = original image – blurred image

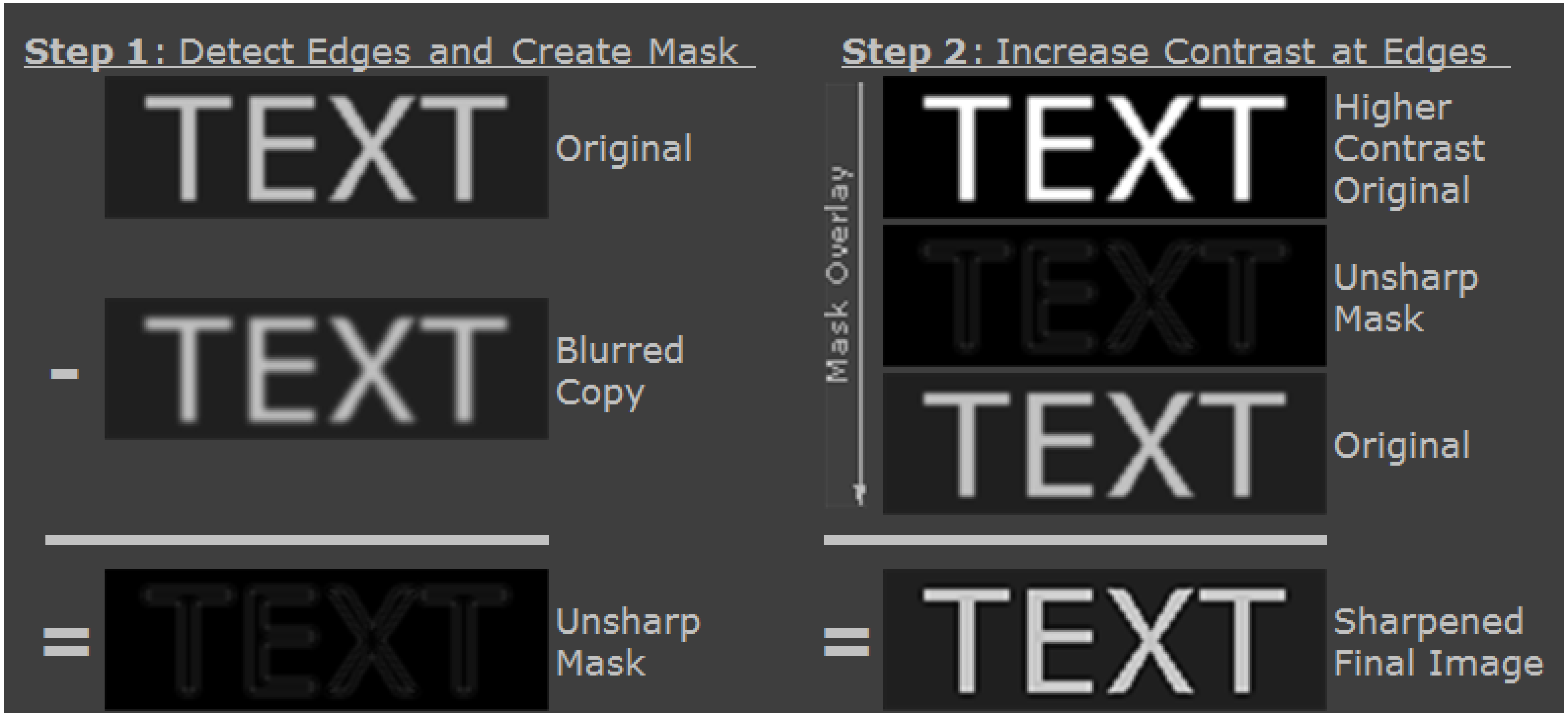
$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y)$$

$$\Downarrow$$

$$f_{hb}(x,y) = (A-1)f(x,y) + (f(x,y) - \bar{f}(x,y))$$
$$= (A-1)f(x,y) + f_s(x,y)$$

$A \geq 1$

Unsharp masking



Transform

Linear image transformations

- In analyzing images, a common approach is making a change of basis.

transformed image

$$\vec{F} = U\vec{f}$$

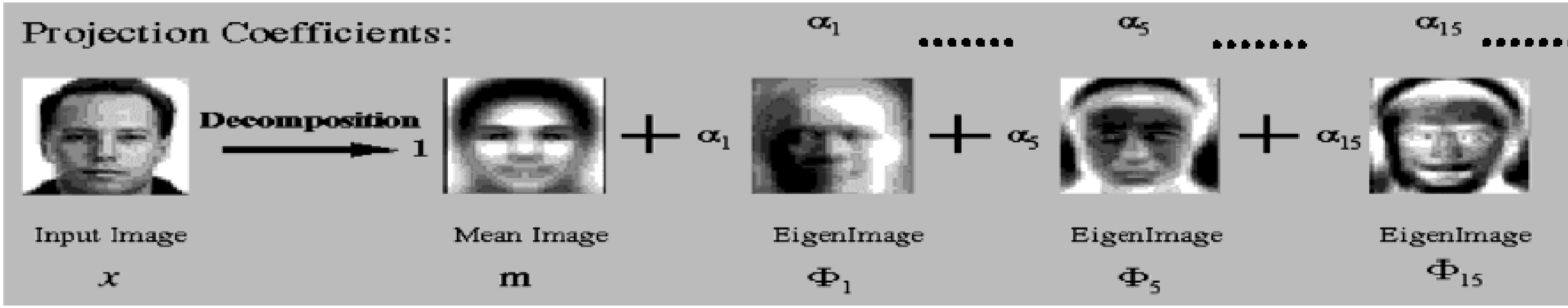
Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform

- Same basis functions can be constructed for the inverse transform

$$\vec{f} = U^{-1}\vec{F}$$

分解成特征脸的线性组合



Fourier transform is invertible

- Continuous

$$F(g(x,y))(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-i2\pi(ux+vy)}dxdy$$

- Discrete
  - Forward

$$F[m,n]=\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} f[k,l]e^{-\pi i\left(\frac{km}{M}+\frac{\ln}{N}\right)}$$

  - Inverse

$$f[k,l]=\frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} F[m,n]e^{+\pi i\left(\frac{km}{M}+\frac{\ln}{N}\right)}$$

Phase and Magnitude

- Fourier transform of a real function is complex
- We can express the Fourier transform in polar coordinates:

$$F(u,v) = |F(u,v)|\, e^{i\phi(u,v)}$$

- where the magnitude is

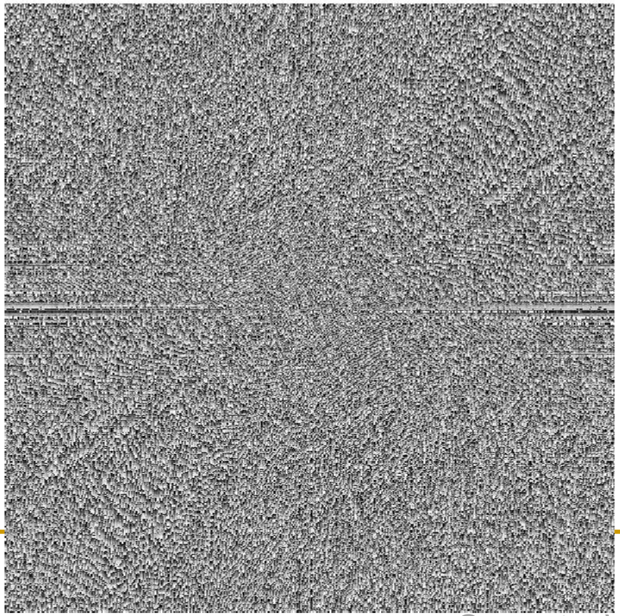
$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

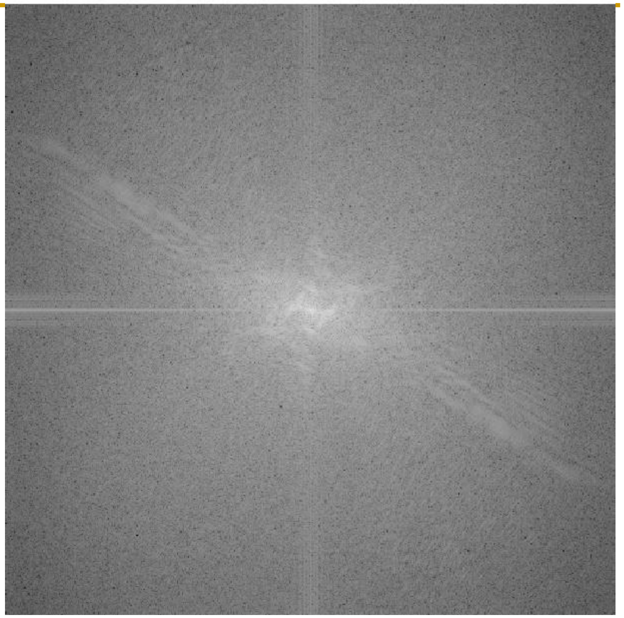
- and the phase is

$$\phi(u,v) = \tan^{-1}[\frac{I(u,v)}{R(u,v)}]$$

- Interesting fact
  - all natural images have about the same magnitude transform

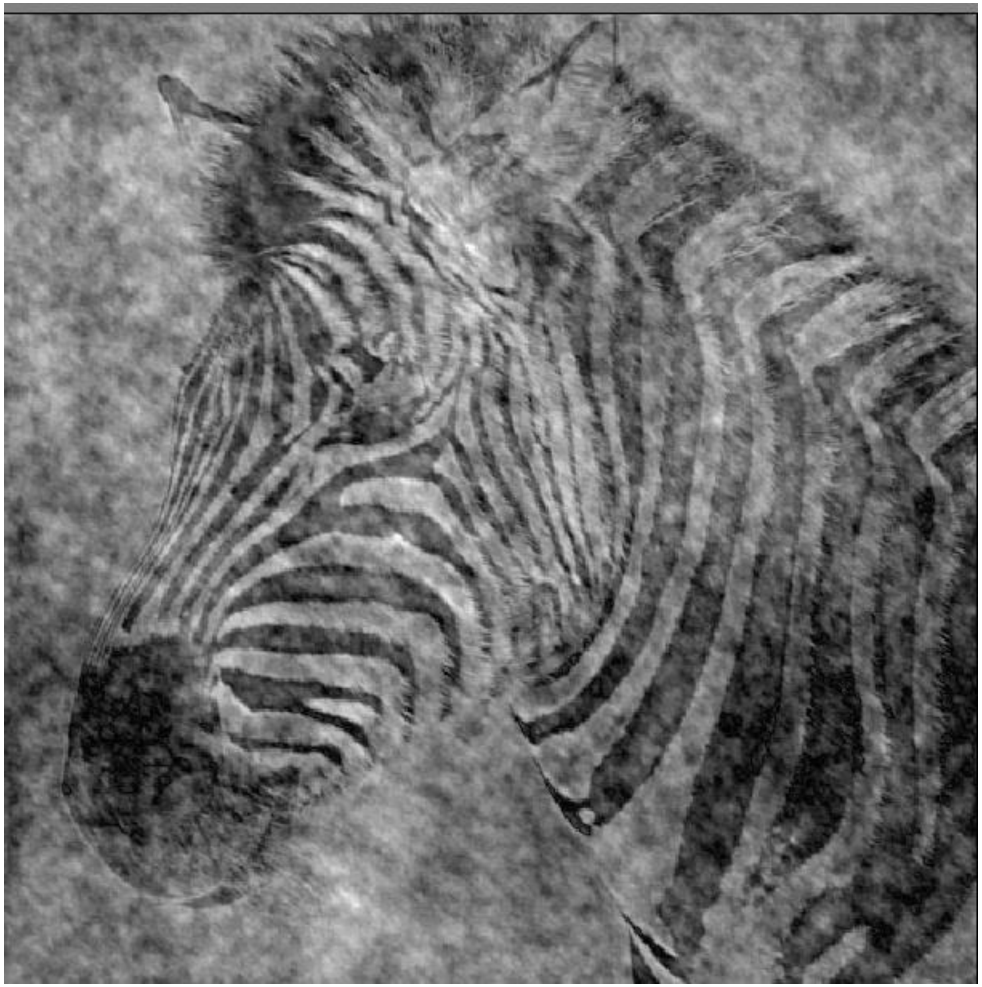
Cheetah Image
Fourier Magnitude (above)
Fourier Phase (below)



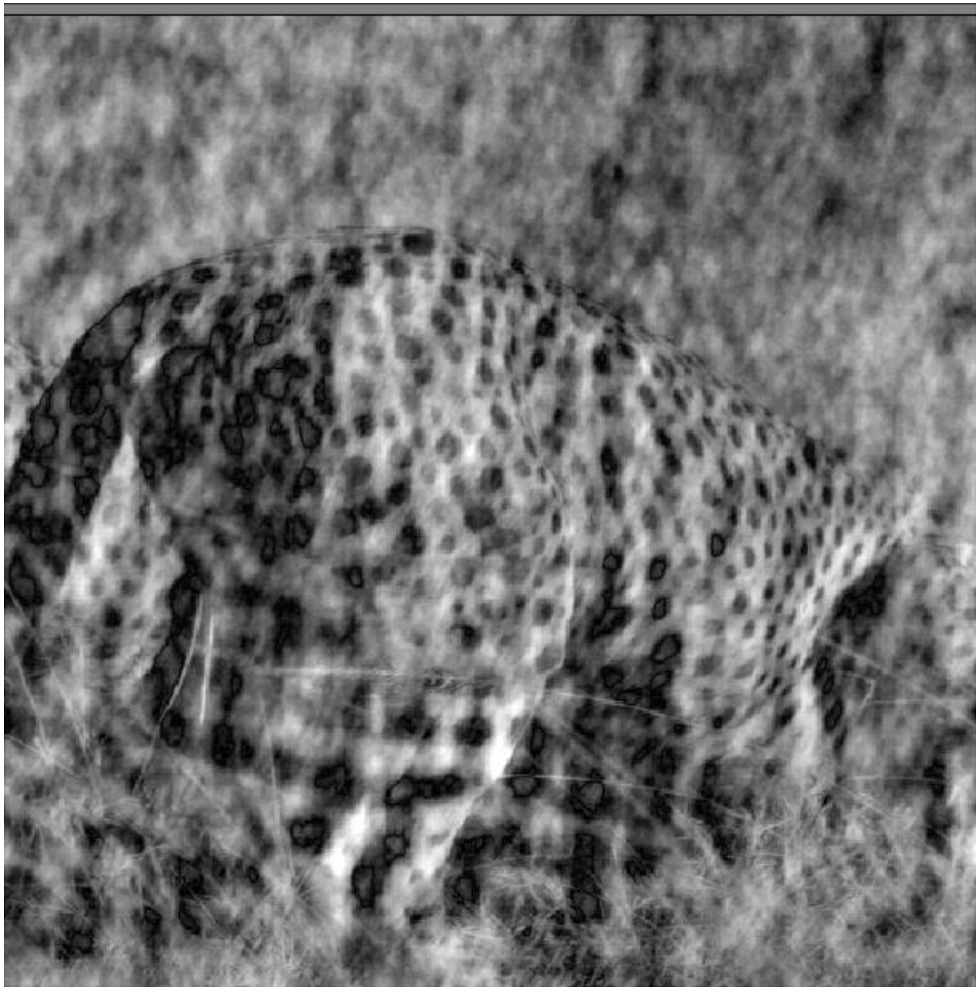


Zebra Image
Fourier Magnitude (above)
Fourier Phase (below)

Reconstruction with
Zebra phase,
Cheetah Magnitude



Reconstruction with Cheetah phase, Zebra Magnitude



Why Fourier transform important?

- Useful for convolution

$$f = g \otimes h \Leftrightarrow F = G * H$$

Reading

- Chapter 4, Digital Image Processing (2nd edition), Prentice Hall.