CAP5415-Computer Vision
Lecture 3-Edge Maps and Histograms

Ulas Bagci

bagci@ucf.edu

Outline

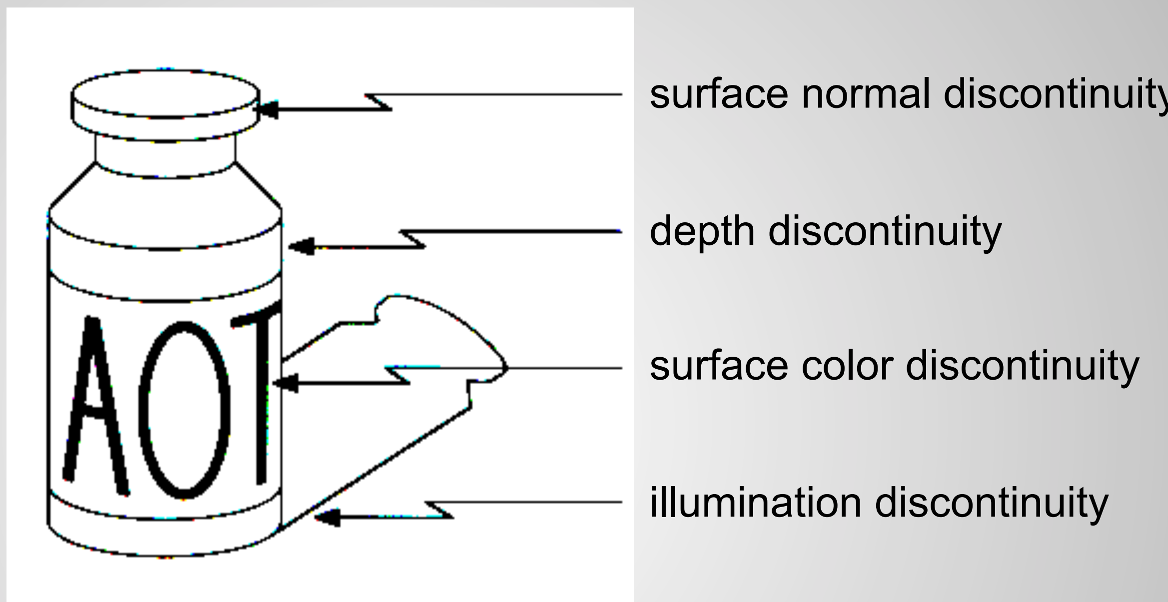
- Continue edge detection and filtering methods,
- Histogram-based analysis

- *Read Szeliski, Chapter 3.*
- *Read Shah, Chapter 2.*
- *Read/Program CV with Python, Chapters 1 and 2.*

Revisiting Edge Detection

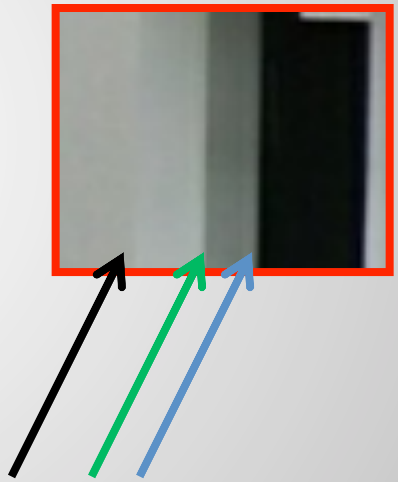
- Goal: Identify sudden changes (discontinuities) in an image
  - Most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
  - Marks the border of an object

Origins of Edges

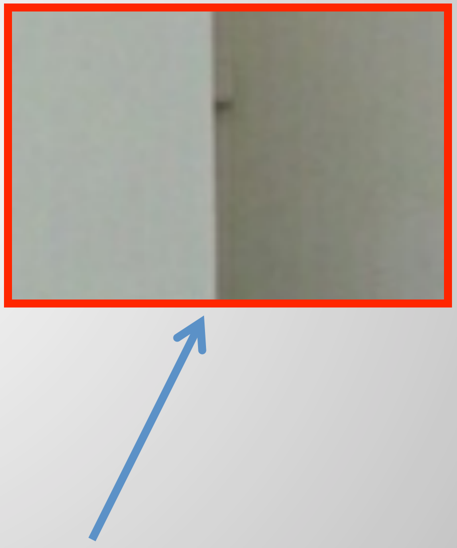


- Edges are caused by a variety of factors

Close-up Edges

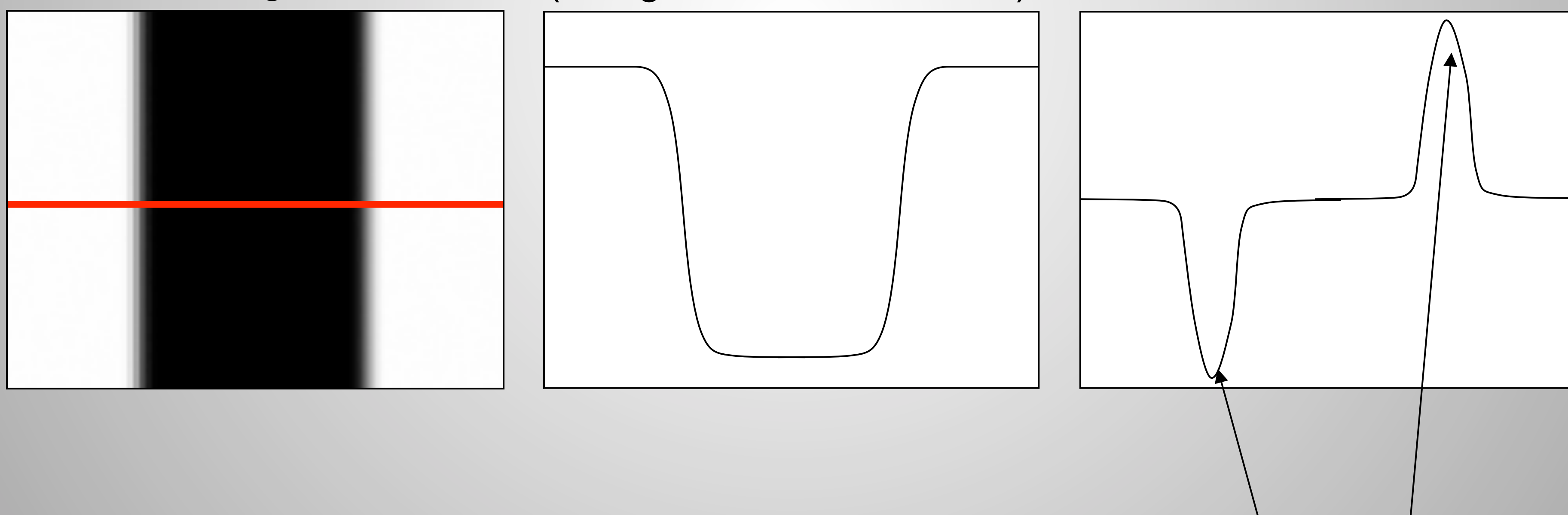


Close-up Edges



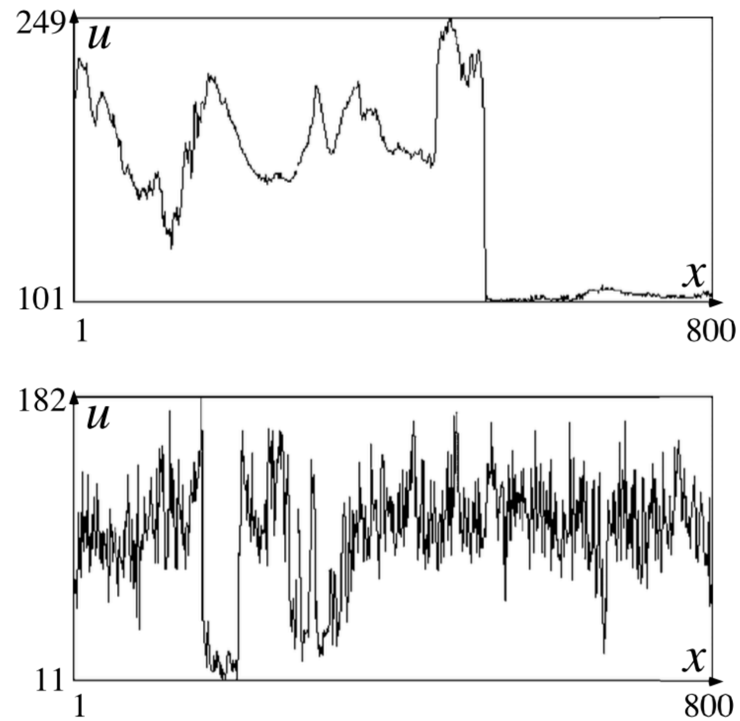
Characterizing Edges

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

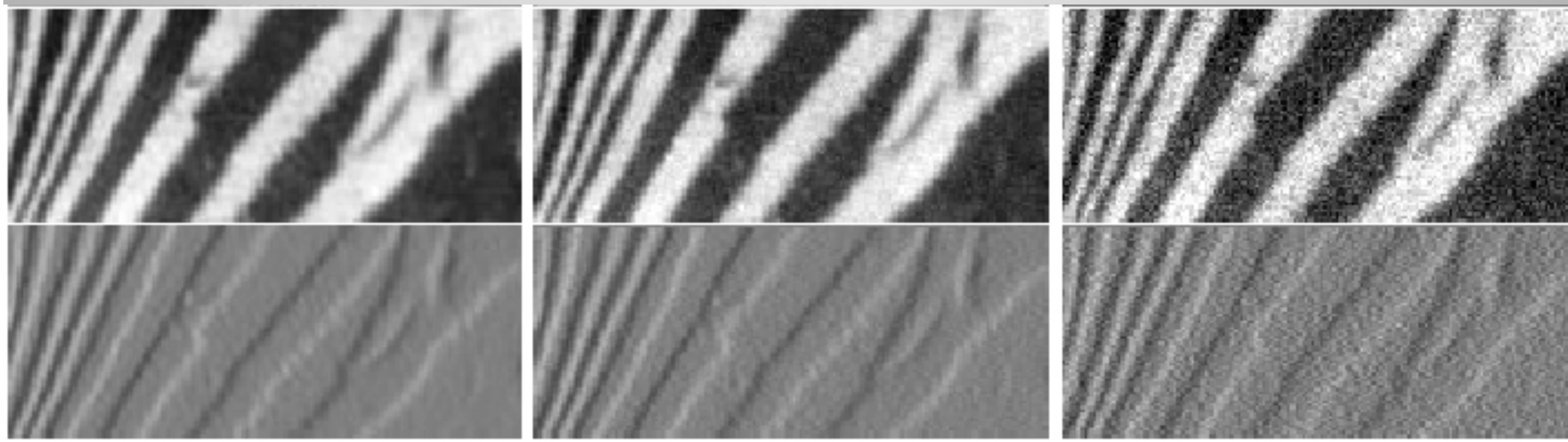


Slide Credit: James Hays

Intensity Profile



Effects of Noise

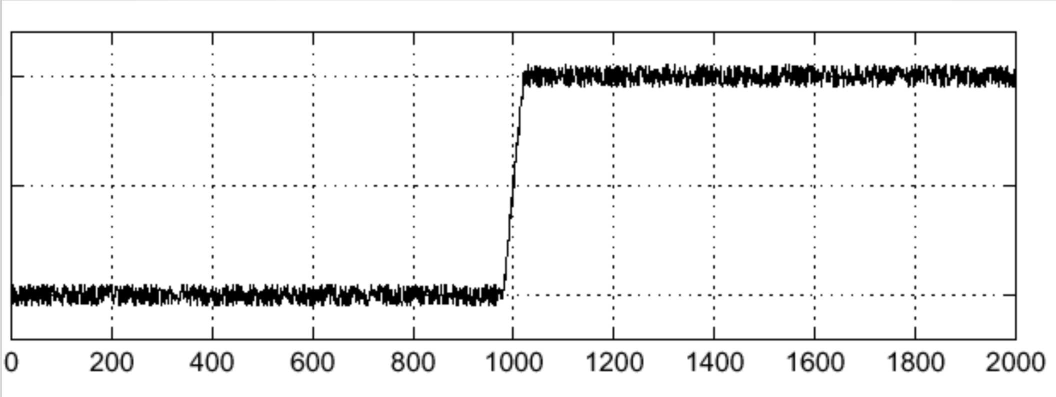


Increasing noise →

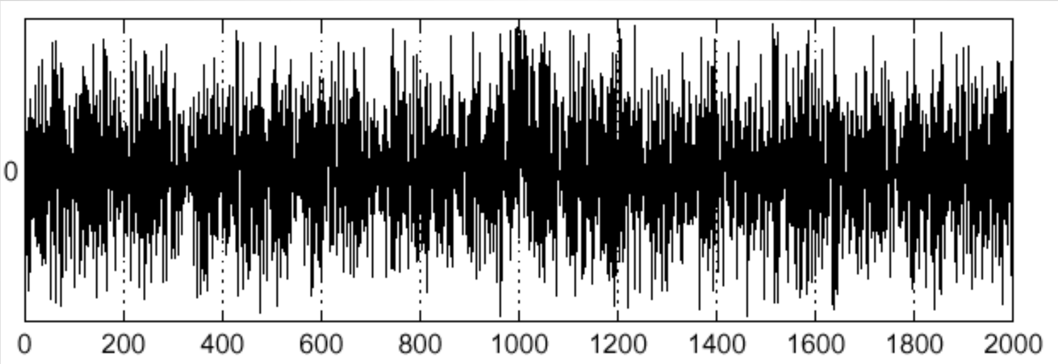
Zero mean additive Gaussian noise

Effects of Noise

$f(x)$



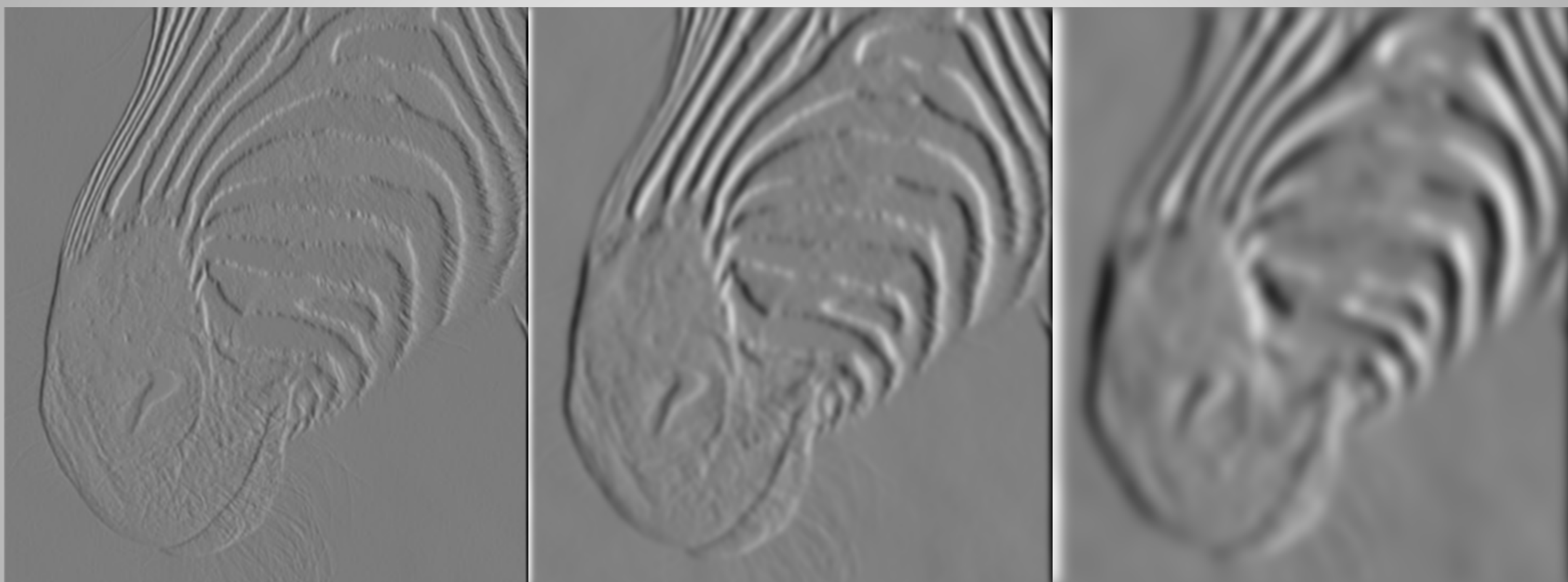
$\frac{d}{dx}f(x)$



Where is the edge?

Credit to Seitz.

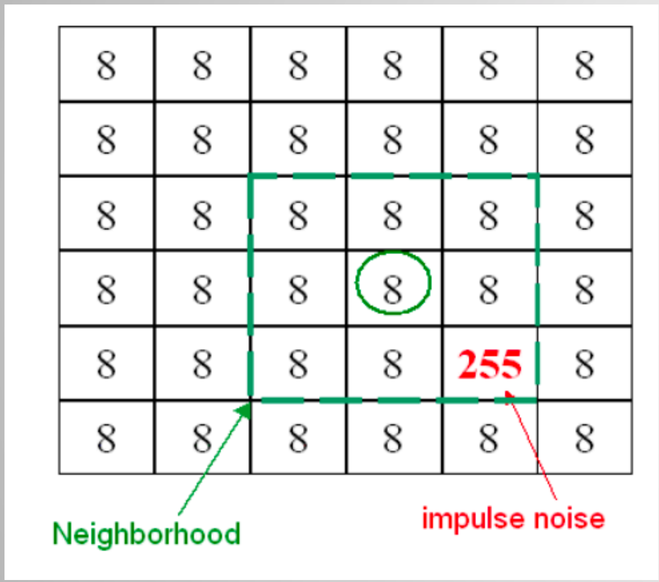
Solution: Smoothing



1 pixel 3 pixels 7 pixels

- Smoothing removes noise, but blurs edge.

Revisiting Median Filtering

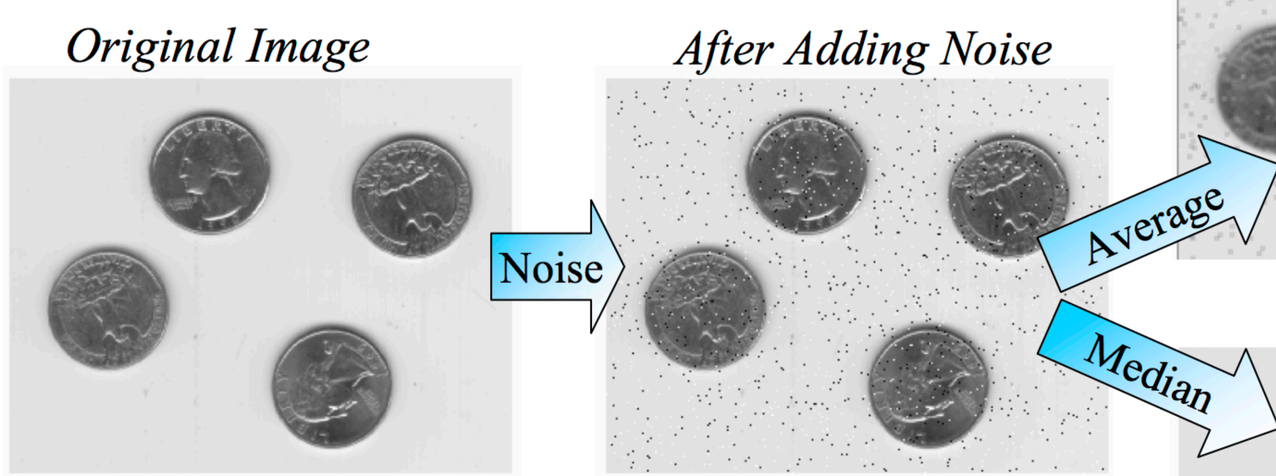


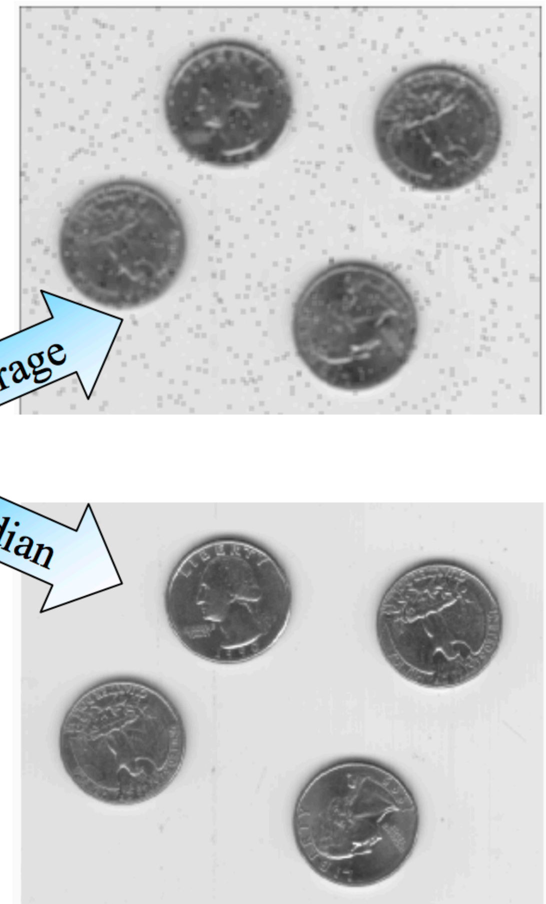
[8 8 8 8 8 8 8 8 **255**]

median

Revisiting Median Filtering

- Original image "*Eight.tif*" with added 'salt-and-pepper' noise then filtered with a (3-by-3) averaging filter and a (3-by-3) median filter.



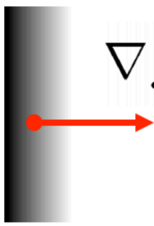


➢ Observation:

The median filter does a better job of removing 'salt-and-pepper' noise, with less blurring of edges.

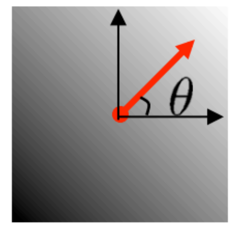
Revisit: Image Gradient

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



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

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



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

The gradient points in the direction of most rapid increase in intensity

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?

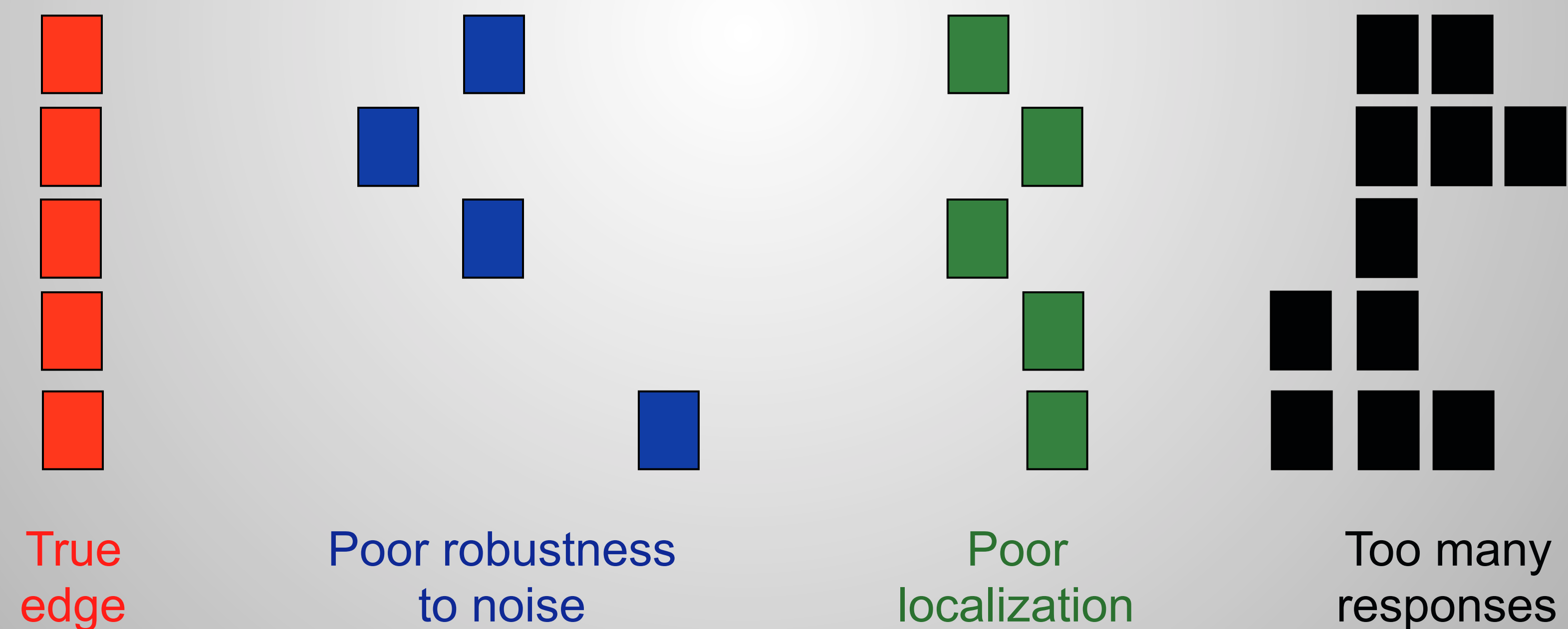
The *edge strength* is given by the gradient magnitude

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

Source: Steve Seitz

Design Criteria for Edge Detection Problems

- Good Detection: minimize prob. of FP (detecting spurious edges) and FN (missing real edges)
- Good Localization: must be as close as possible to the true edges
- Single Response: must return one point only for each true edge point

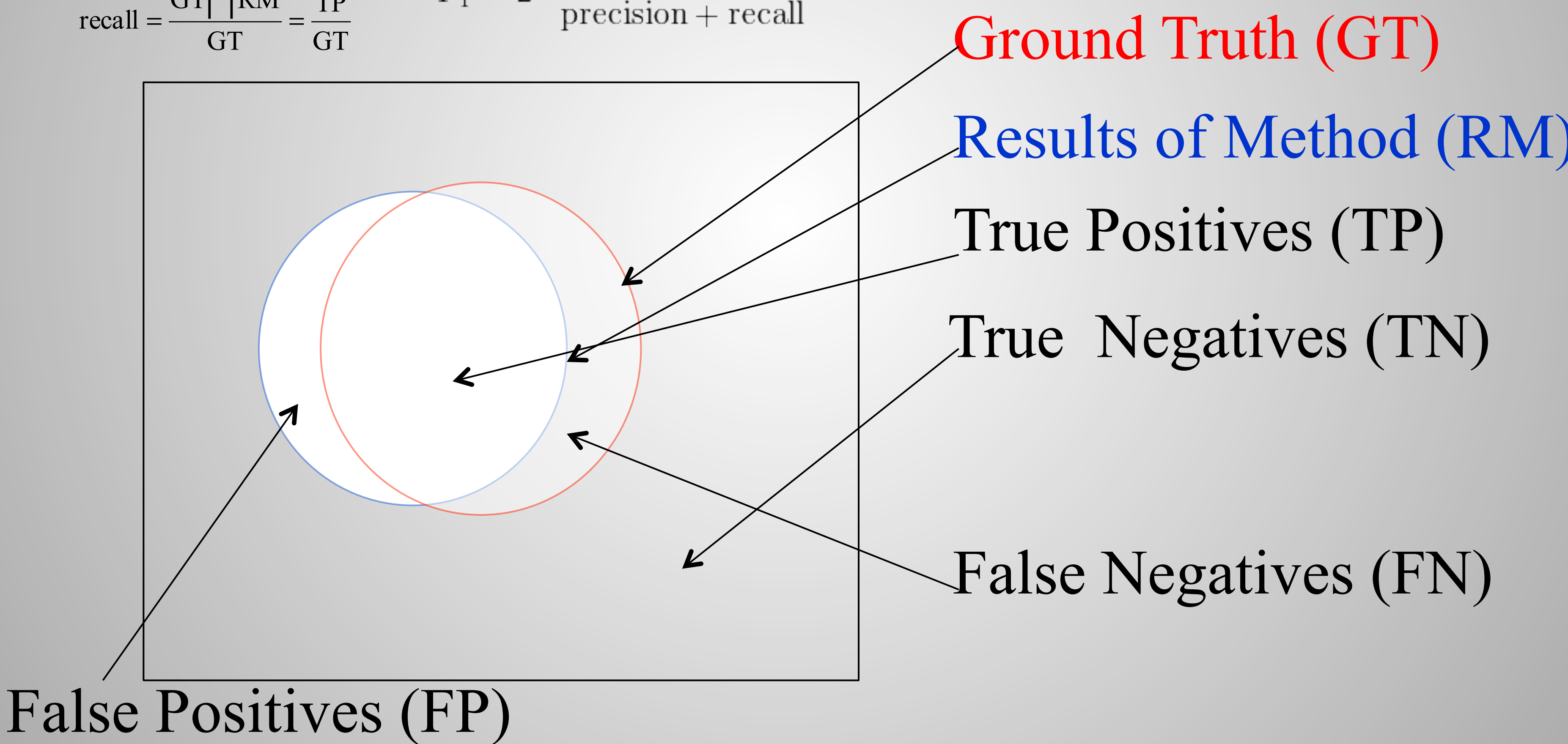


Evaluate Edge Detection

$$\text{precision} = \frac{\text{GT} \bigcap \text{RM}}{\text{RM}} = \frac{\text{TP}}{\text{RM}}$$

$$\text{recall} = \frac{\text{GT} \bigcap \text{RM}}{\text{GT}} = \frac{\text{TP}}{\text{GT}}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



Basic Comparisons of Edge Operators

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

Good Localization
Noise Sensitive
Poor Detection

Roberts (2 x 2):

0	1
-1	0

1	0
0	-1

Sobel (3 x 3):

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	1

Sobel (5 x 5):

-1	-2	0	2	1
-2	-3	0	3	2
-3	-5	0	5	3
-2	-3	0	3	2
-1	-2	0	2	1

1	2	3	2	1
2	3	5	3	2
0	0	0	0	0
-2	-3	-5	-3	-2
-1	-2	-3	-2	-1

Poor Localization
Less Noise Sensitive
Good Detection

Example: Laplacian of Gaussian (LoG) and Canny Edge Detector

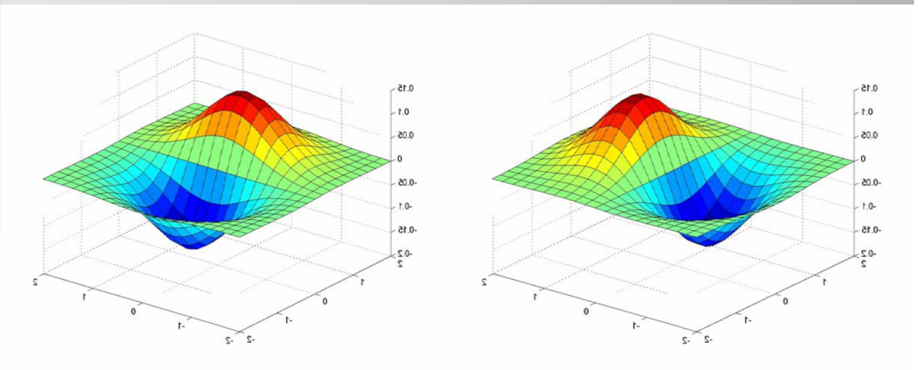
Marr and Hildreth Filtering, 1980.

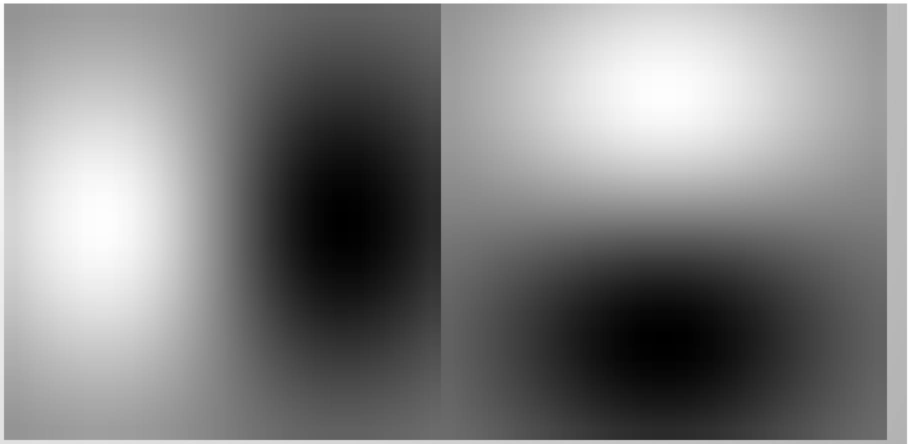
- Smooth Image with Gaussian Filter
- Applying the Laplacian for a Gaussian-filtered image can be done in one step of convolution.
- Find zero-crossings
- Find slope of zero-crossings
- Apply threshold to slope and mark edges

J. Canny. 1986

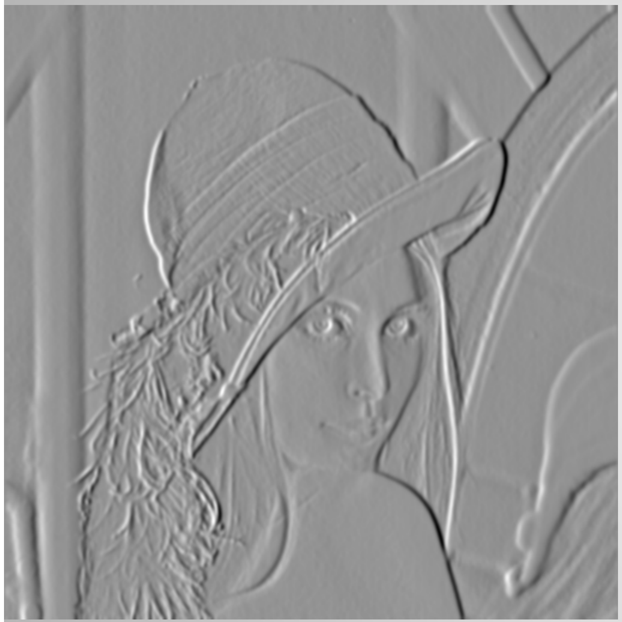
- Smooth Image with Gaussian filter
- Compute Derivative of filtered image
- Find Magnitude and Orientation of gradient
- Apply Non-max suppression
- Apply Thresholding (Hysteresis)

Example: Canny





Example: Canny-Gradients

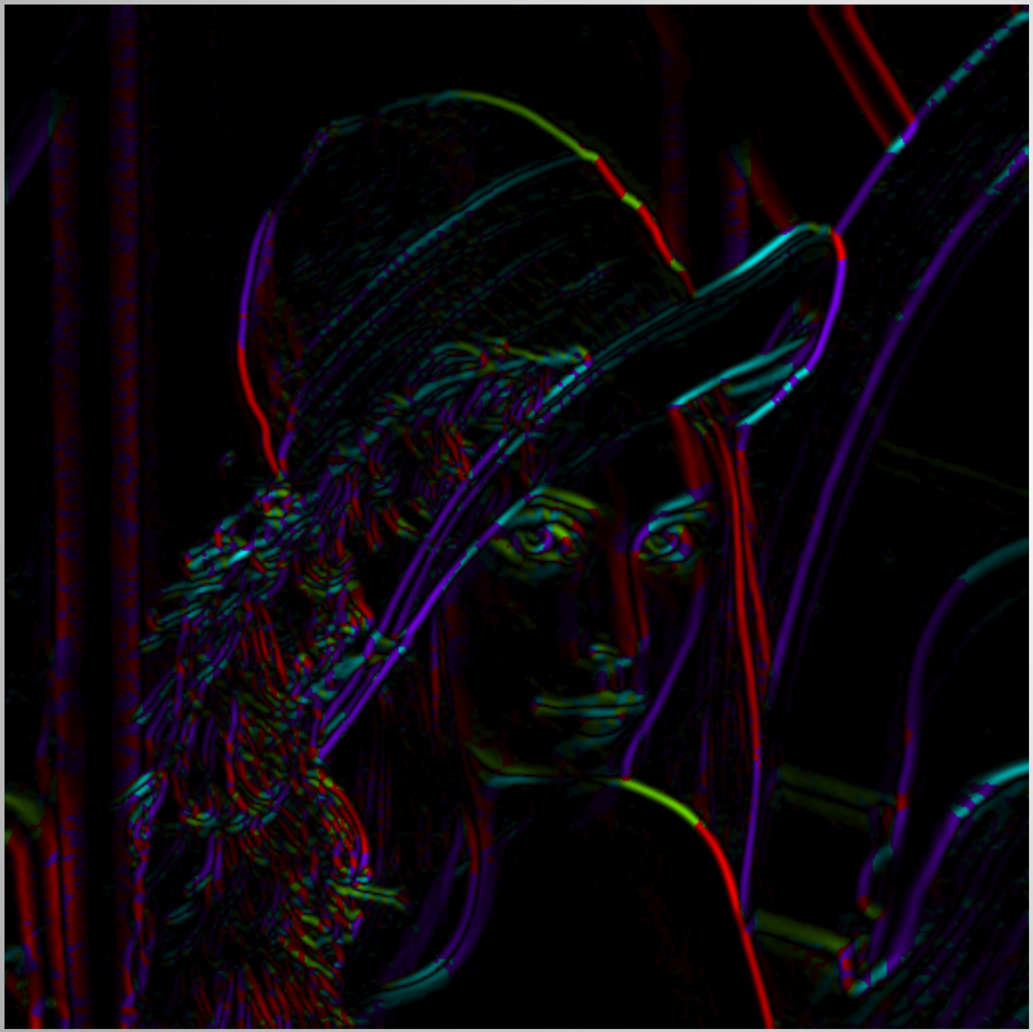


X-Derivative of Gaussian

Y-Derivative of Gaussian

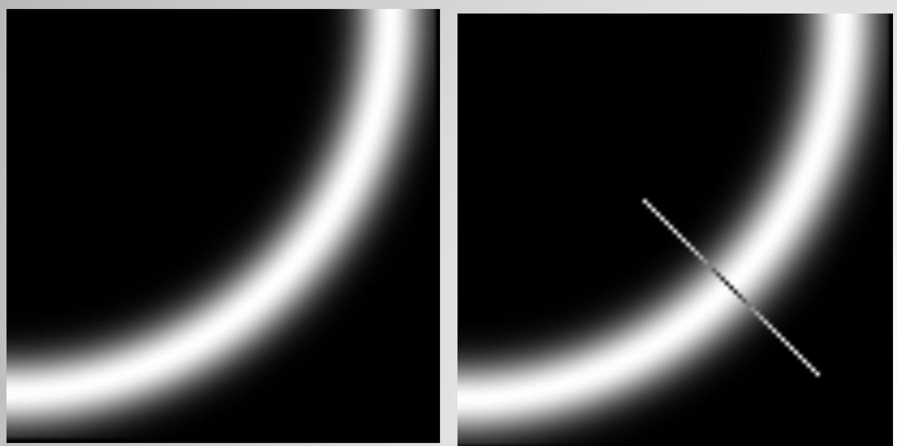
Gradient Magnitude

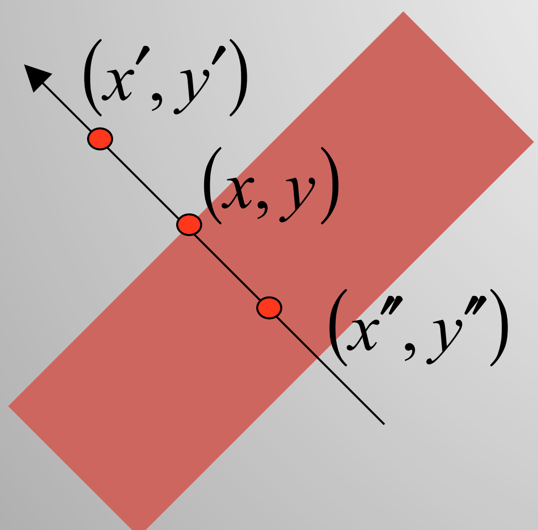
Example: Gradient Orientation



theta = atan2(gy, gx)

Example: Non-maximum suppression





$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\Delta S|(x',y') \\ & \& \, |\Delta S|(x,y) > |\Delta S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$

x’ and **x**’’ are the neighbors of **x** along normal direction to an edge

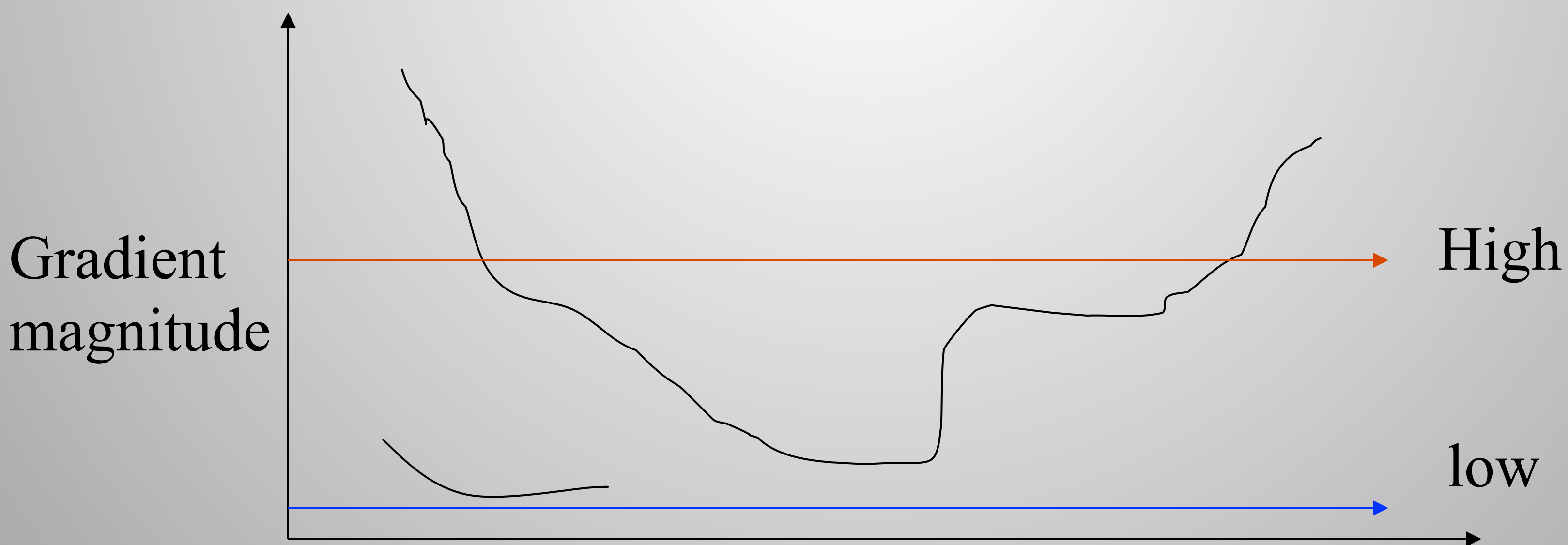
Example: Non-maximum suppression

Before Non-Max Suppression

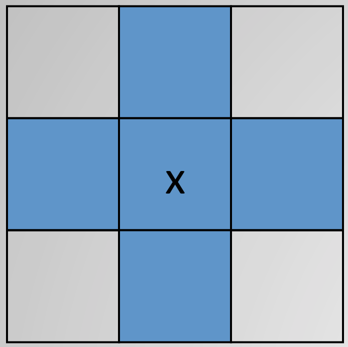
After Non-Max Suppression

Example: Hysteresis Thresholding [L, H]

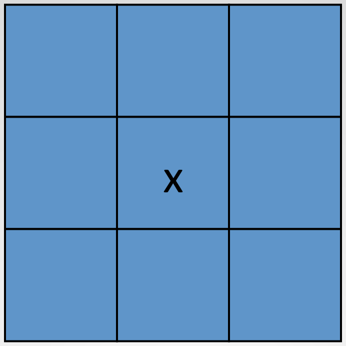
- If the gradient at a pixel is
  - above "**High**", declare it as an '**edge pixel**'
  - below "**Low**", declare it as a "**non-edge-pixel**"
  - **between** "low" and "high"
    - Consider its neighbors iteratively then declare it an "edge pixel" if it is **connected** to an 'edge pixel' **directly** or via pixels **between** "low" and "high".



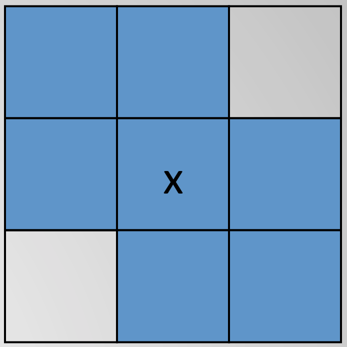
Example: Hysteresis Thresholding [L, H]



4 connected



8 connected



6 connected

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

Example: Final Canny Edges

Effect of Gaussian Kernel (smoothing)

original

Canny with $\sigma = 1$

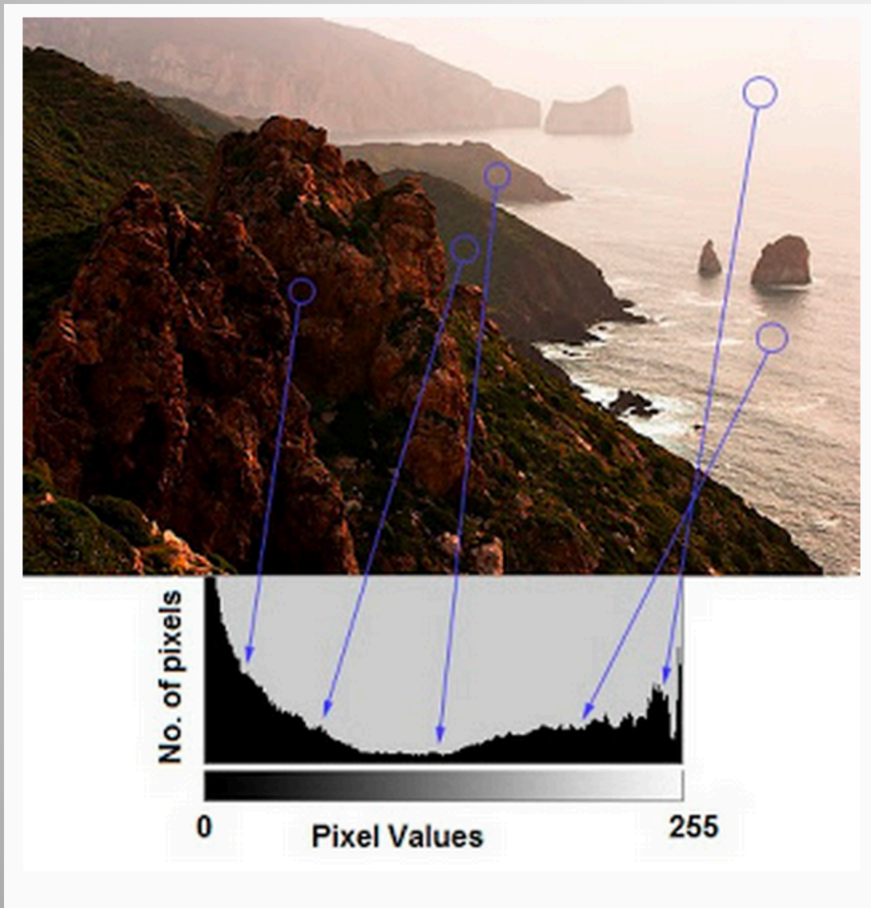
Canny with $\sigma = 2$

The choice of σ depends on desired behavior

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

Image Histogram

- A histogram represents tabulated frequencies, typically by using bars in a graphical diagram.



- It provides a natural bridge between Images and a probabilistic description.

Ex: What is the probability of pixel ***p*** at location (x,y) has brightness ***z*** ?

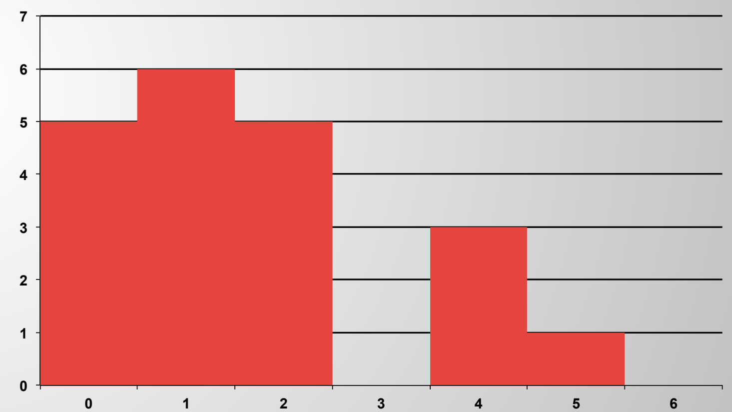
- Histogram of a digital image typically has many local minima and maxima, which may complicate its further Processing.

- This problem can be solved by local smoothing of the histogram.

Histogram

0	1	1	2	4
2	1	0	0	2
5	2	0	0	4
1	1	2	4	1

image



histogram

Histogram and PDF

- Assume a scalar image, A, and its histogram H.

$$h(u) = H(u)/|\Omega|$$

- Denominator is the size of histogram (num. of pixels)
- h -> PDF, relative frequencies are set between 0 and 1.

Entropy / Histogram

- If a probability density ***h*** is known, then image information content can be estimated regardless of its interpretation using entropy ***E***.

$$E(X) = -\sum_{k=1}^{n} h(x_k) \log h(x_k)$$

- Shannon, 1948 [Information entropy].
- Amount of uncertainty about an event associated with a given PDF

Gradation Functions

- When recording image data, there are often particular problems: lighting, motion blur, noise,…
  - Uniform illumination is desired
  - Smoothing (denoising)
  - Sharpening

Gradation Functions

- When recording image data, there are often particular problems: lighting, motion blur, noise,…
  - Uniform illumination is desired
  - Smoothing (denoising)
  - Sharpening

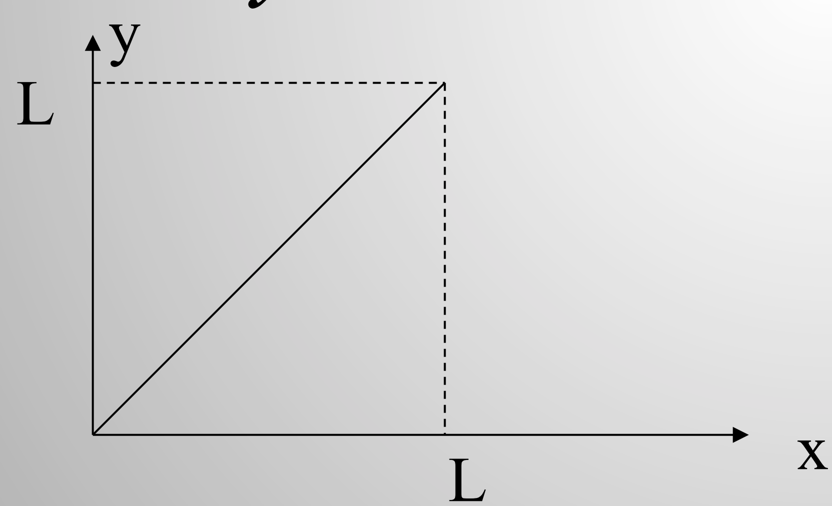
Gradation Functions

- We transform an image **A** into a new image $\mathbf{A}_{\mathbf{new}}$ of the same size, by mapping a grey level ***u*** at pixel location ***p*** in A by a gradation function ***g*** onto a grey level ***v=g(u)***.

Gradation Functions

- We transform an image **A** into a new image $\mathbf{A}_{\mathbf{new}}$ of the same size, by mapping a grey level $\boldsymbol{u}$ at pixel location $\boldsymbol{p}$ in A by a gradation function $\boldsymbol{g}$ onto a grey level $\boldsymbol{v=g(u)}$.

$$y = x$$

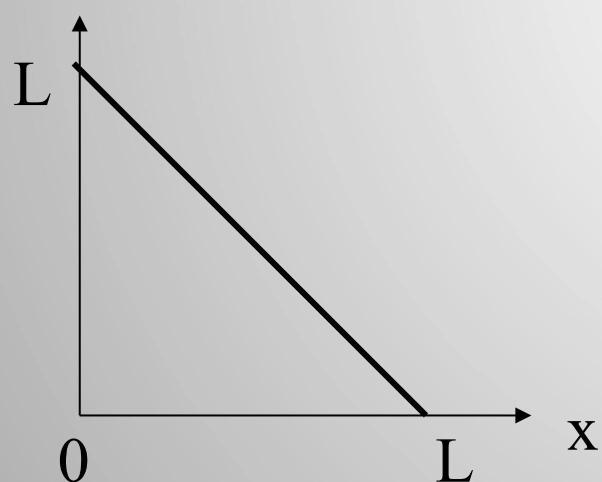


No influence on visual quality at all

Gradation Functions

- We transform an image **A** into a new image $\mathbf{A}_{\mathbf{new}}$ of the same size, by mapping a grey level ***u*** at pixel location ***p*** in A by a gradation function ***g*** onto a grey level ***v=g(u)***.

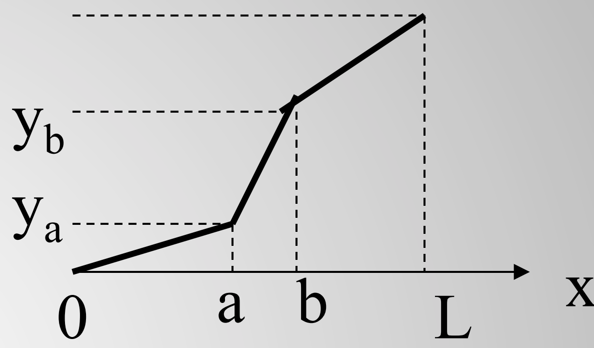
- Digital negative



$$y = L - x$$

Gradation Functions-Contrast Stretching

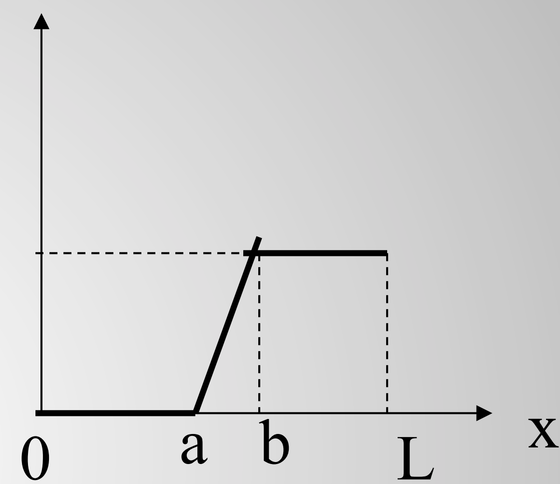
$$y = \begin{cases} \alpha x & 0 \le x < a \\ \beta(x-a) + y_a & a \le x < b \\ \gamma(x-b) + y_b & b \le x < L \end{cases}$$



$$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, y_a = 30, y_b = 200$$

Gradation Functions-Clipping

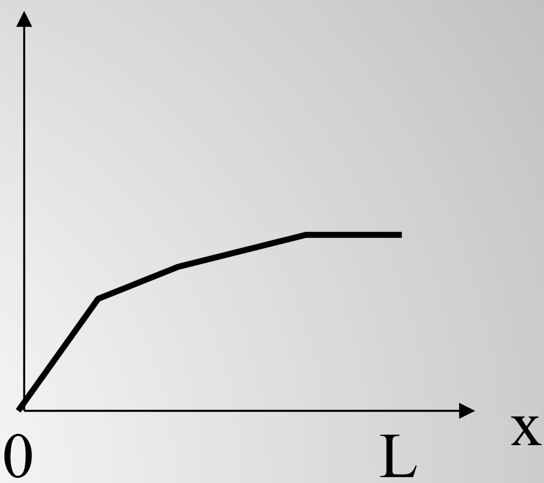
$$y=\begin{cases} 0 & 0\le x<a \\ \beta(x-a) & a\le x<b \\ \beta(b-a) & b\le x<L \end{cases}$$



$a=50, b=150, \beta=2$

Gradation Functions-Range Compression

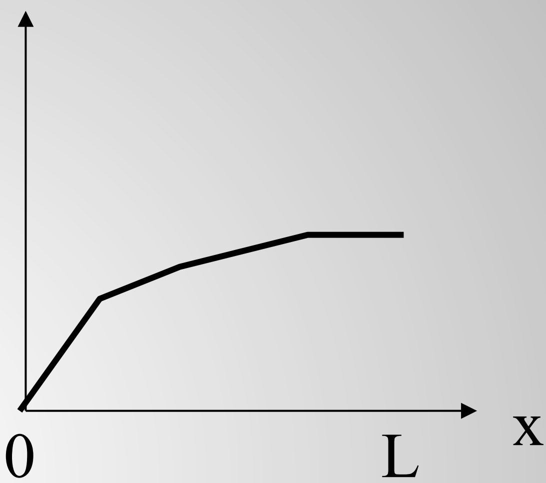
$$y = c\log_{10}(1+x)$$



c=100

Gradation Functions-Range Compression

$$y = c\log_{10}(1+x)$$



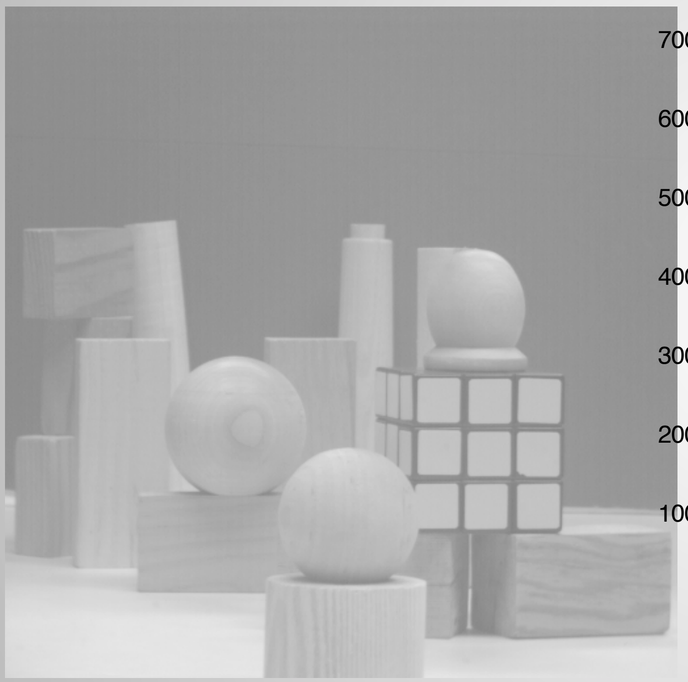
c=100

Gradation Functions

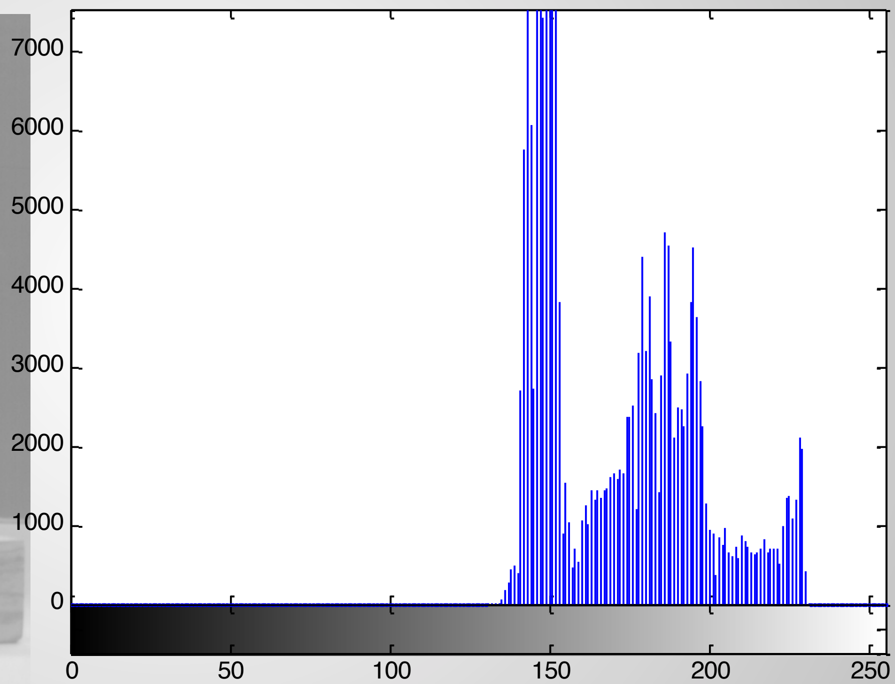
- We transform an image **A** into a new image $\mathbf{A}_{\mathbf{new}}$ of the same size, by mapping a grey level ***u*** at pixel location ***p*** in A by a gradation function ***g*** onto a grey level ***v=g(u)***.

- Histogram Equalization:
  - The aim is to create an image with equally distributed brightness (intensity) levels over the whole brightness (intensity) scale.
  - Enhances contrast for intensity values close to histogram maxima and decreases contrast near minima

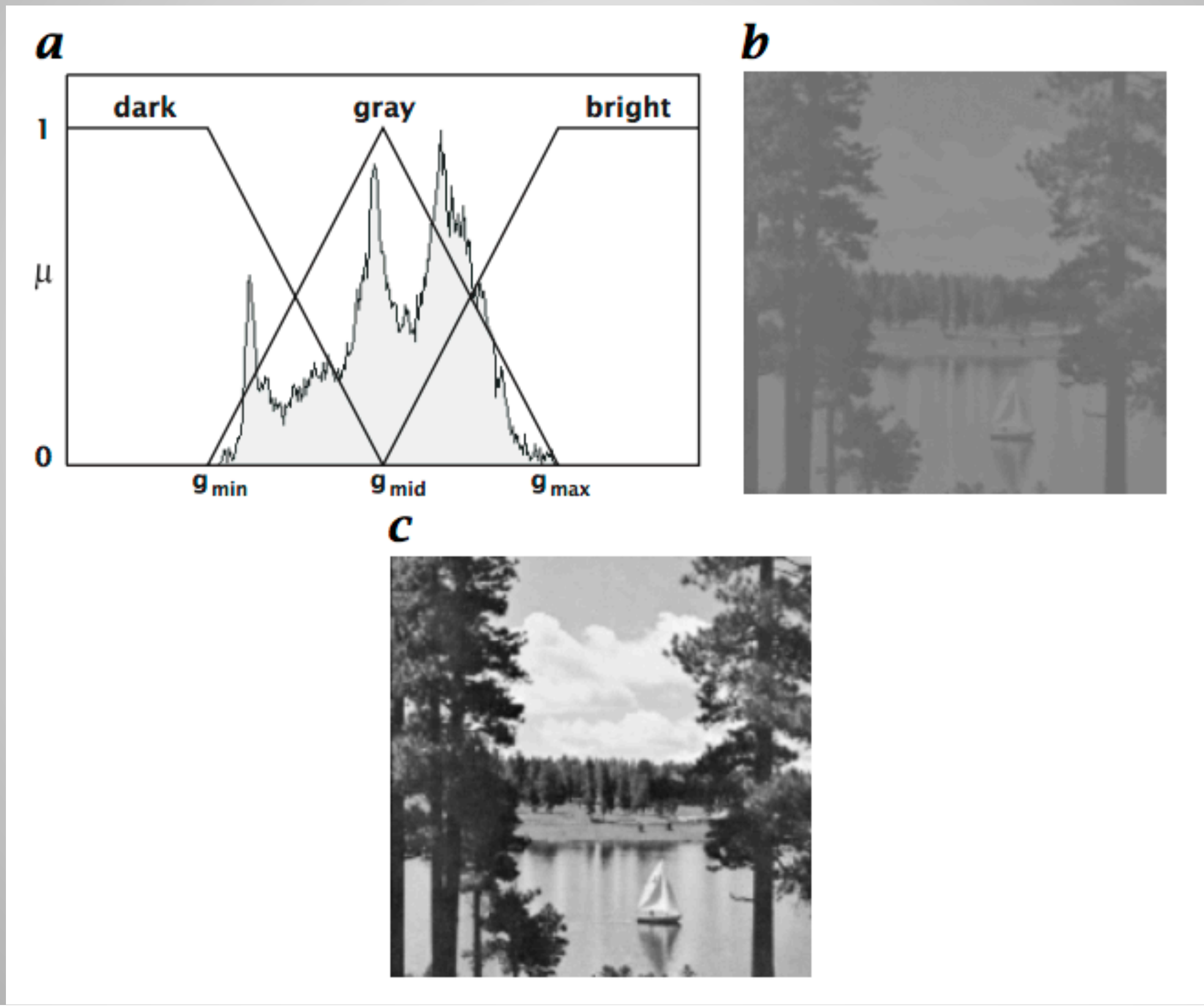
Histogram Equalization



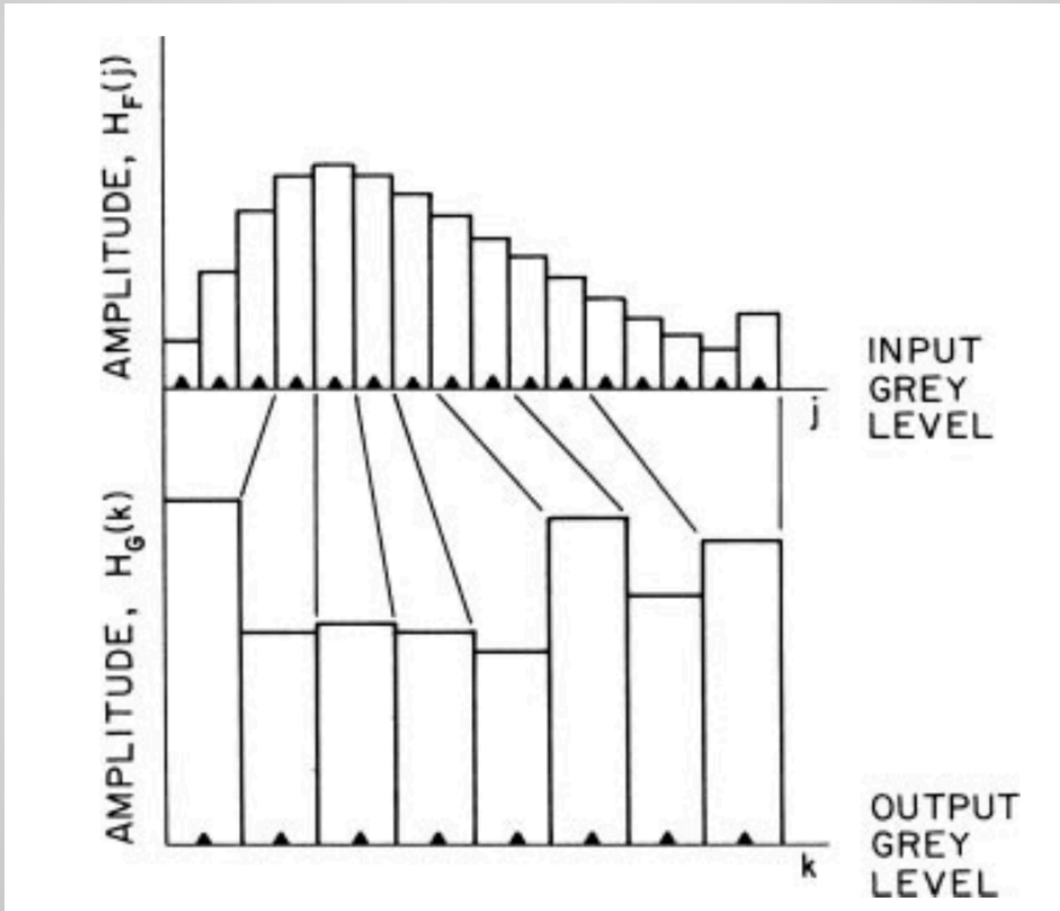
Over-exposed image



Histogram Equalization



Histogram Equalization



Algorithm for Histogram Equalization

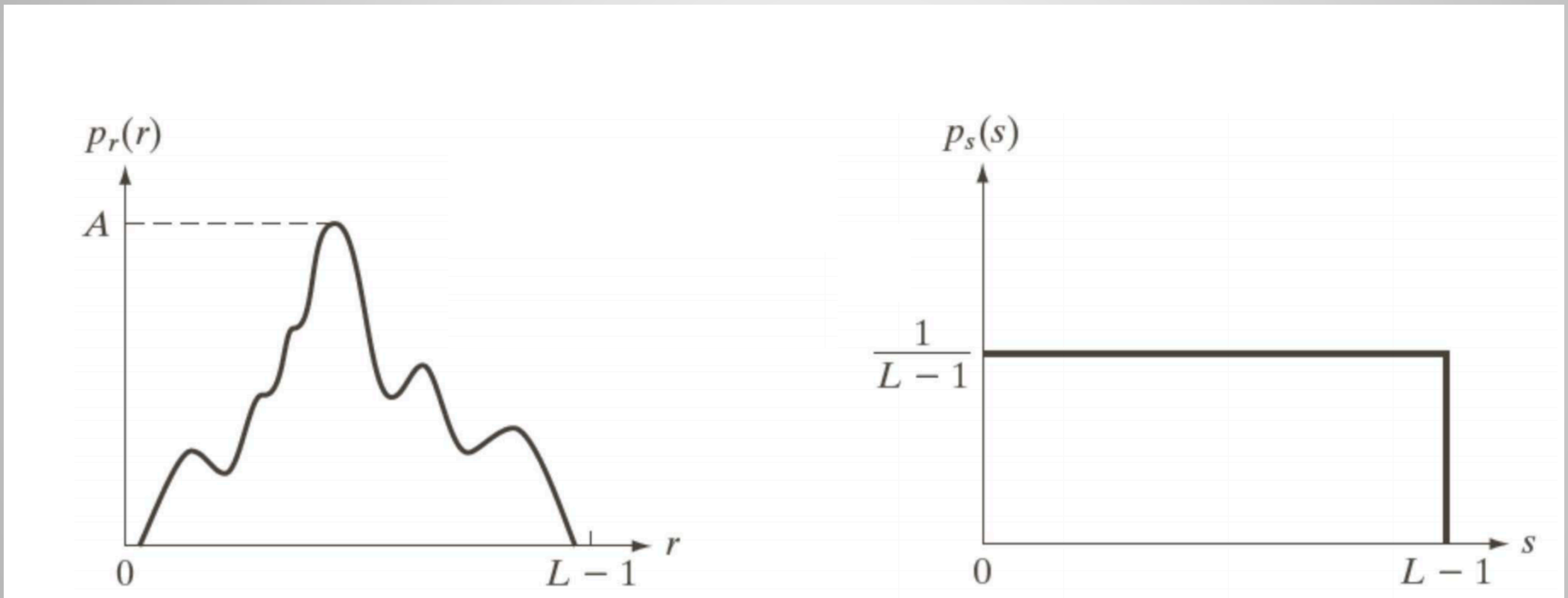
Normalized histogram p (PDF) of an image f, whose intensity values span from 0 to L-1

$$p_n = \frac{\text{number of pixels with intensity } n}{\text{total number of pixels}} \qquad n = 0, 1, ..., L-1.$$

The histogram equalized image ***g*** will be defined by

$$g_{i,j} = \operatorname{floor}((L-1)\sum_{n=0}^{f_{i,j}} p_n),$$

Histogram Transform



Histogram Equalization – Discrete Case

- for discrete case we have:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= \frac{(L-1)}{M \cdot N}\sum_{j=0}^{k} n_j \qquad 0 \le k \le L-1$$

- assume $L = 8,\ M = N = 64,\ M \cdot N = 4096$

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

- $s_0 = T(r_0) = 7 \cdot \sum_{j=0}^{0} p_r(r_j) = 7 \cdot p_r(0) = 1.33 \rightarrow 1$
- $s_1 = T(r_1) = 7 \cdot \sum_{j=0}^{1} p_r(r_j) = 7 \cdot (p_r(0) + p_r(1)) = 3.08 \rightarrow 3$
- $s_2 = T(r_2) = 7 \cdot \sum_{j=0}^{2} p_r(r_j) = 7 \cdot (p_r(0) + p_r(1) + p_r(2)) = 4.55 \rightarrow 5$

$s_3 = 5.67 \rightarrow 6,\ s_4 = 6.23 \rightarrow 6, s_5 = 6.65 \rightarrow 7, s_6 = 6.86 \rightarrow 7, s_7 = 7.00 \rightarrow 7$

Histogram Equalization - Discrete

- final transform:

$$r_0 \to s_0 = 1 \Rightarrow 790 \text{ pixels map to } 1$$

$$r_1 \to s_1 = 3 \Rightarrow 1023 \text{ pixels map to } 3$$

$$r_2 \to s_2 = 5 \Rightarrow 850 \text{ pixels map to } 5$$

$$r_3 \to s_3 = 6 \Rightarrow 656 + 329 = 985 \text{ pixels map to } 6$$

$$r_4 \to s_4 = 6 \Rightarrow 656 + 329 = 985 \text{ pixels map to } 6$$

$$r_5 \to s_5 = 7 \Rightarrow 245 + 122 + 81 = 458 \text{ pixels map to } 7$$

$$r_6 \to s_6 = 7 \Rightarrow 245 + 122 + 81 = 458 \text{ pixels map to } 7$$

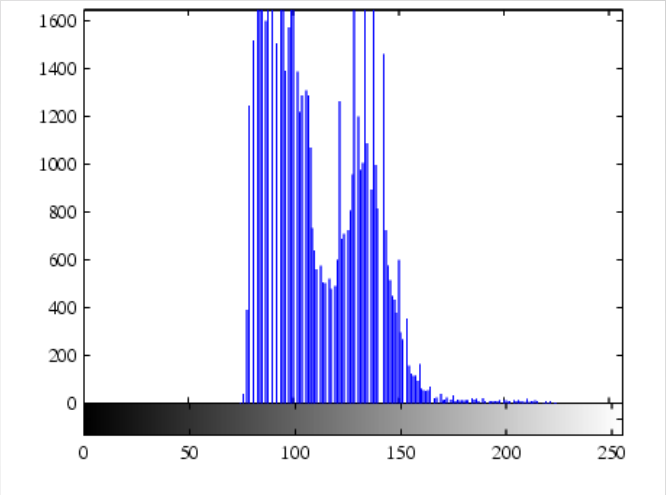
$$r_7 \to s_7 = 7 \Rightarrow 245 + 122 + 81 = 458 \text{ pixels map to } 7$$

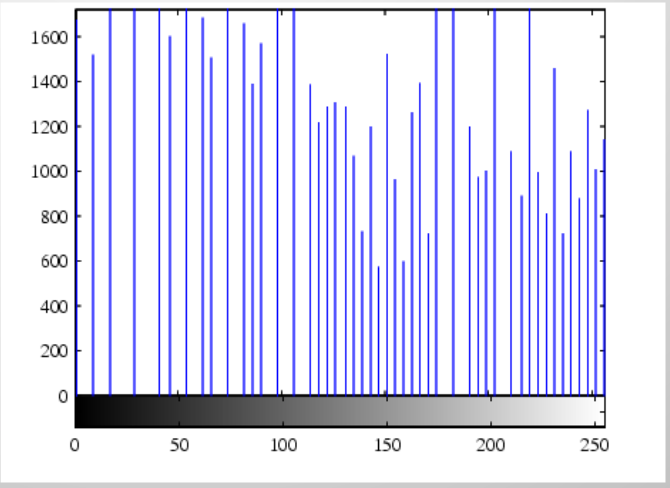
Histogram Equalization Example 1

before

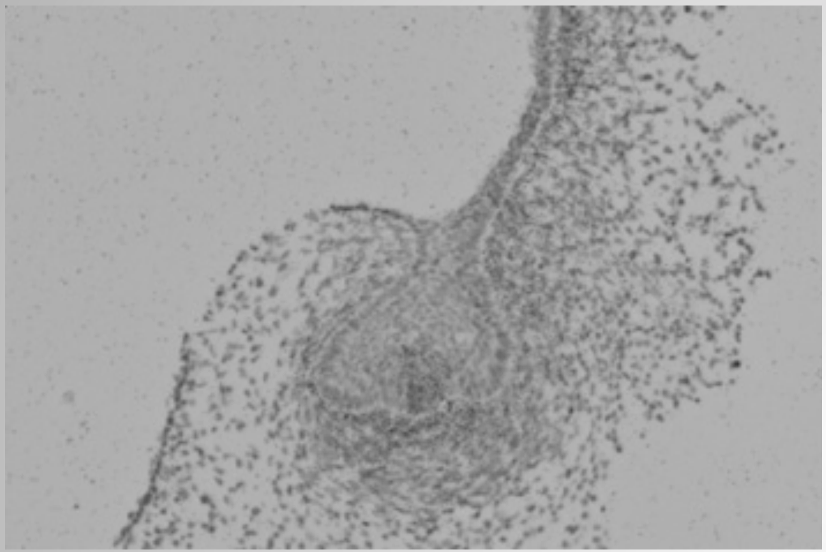
after

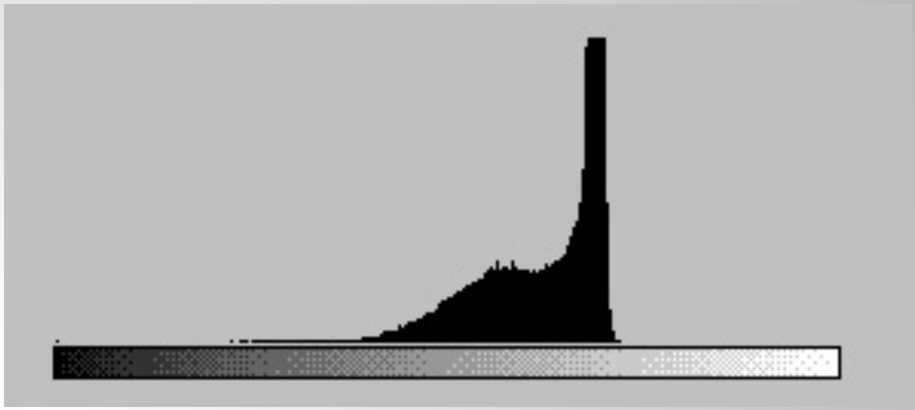
Histogram Equalization Example 2

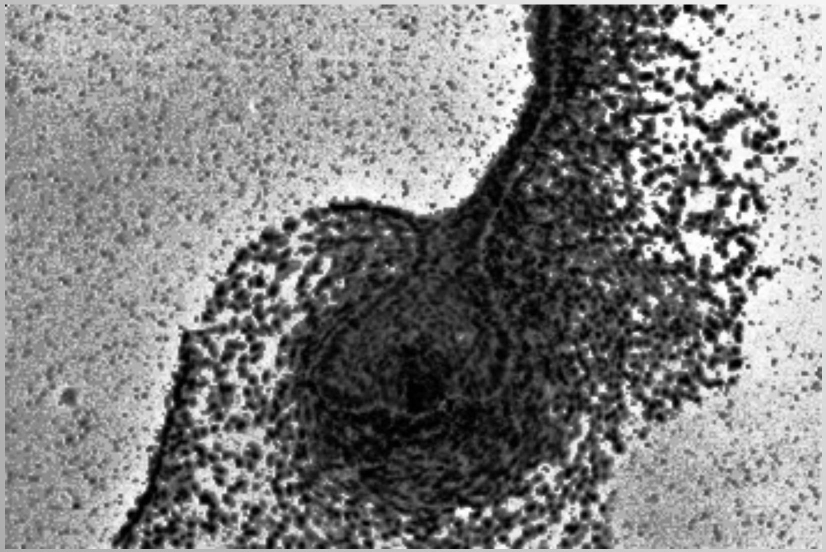


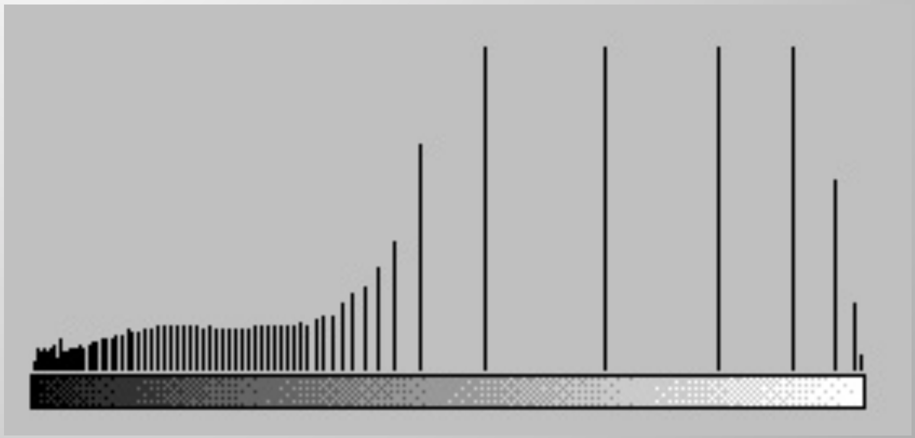


Histogram Equalization Example 3

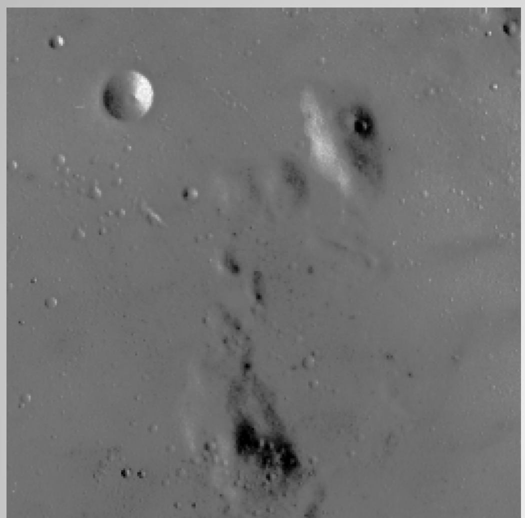


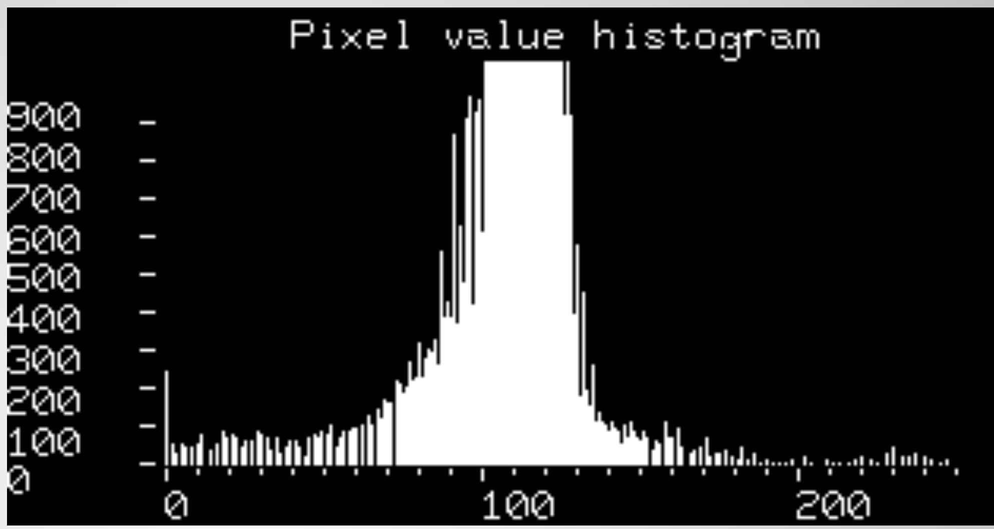


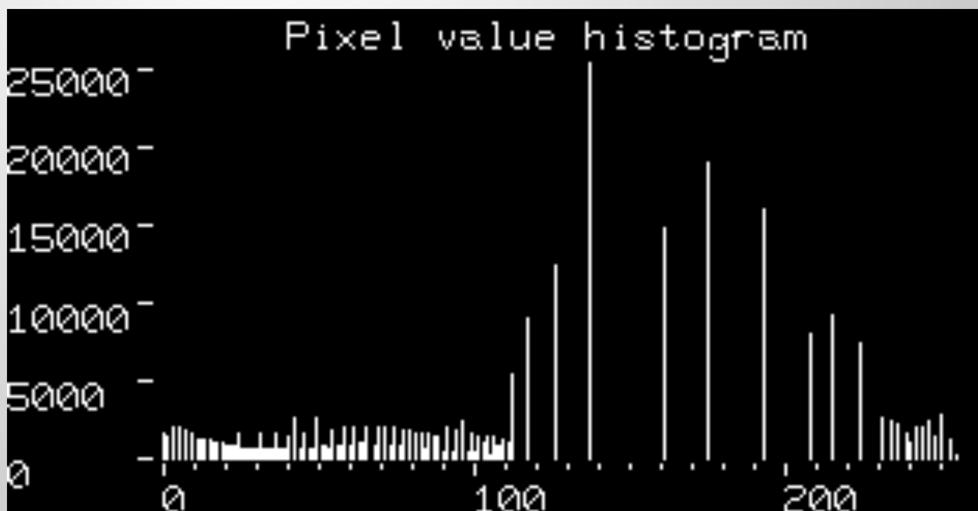




Histogram Equalization Example 4







Histogram Equalization Example 5

Histogram Equalization Example 6

Summary

- Edge Detection
- Noise and Smoothing
- Canny Edge Detector
- Histograms
- Histogram Equalization

Programming Assignment #1

- Image display and filtering
- Deadline: September 10, 2015.

- REMINDERS
  - Deadline for PA0 (Bonus) is 3rd September, 2015.
  - Submit online.