

CSc I6716
Fall 2010

Part I
Feature Extraction (1)

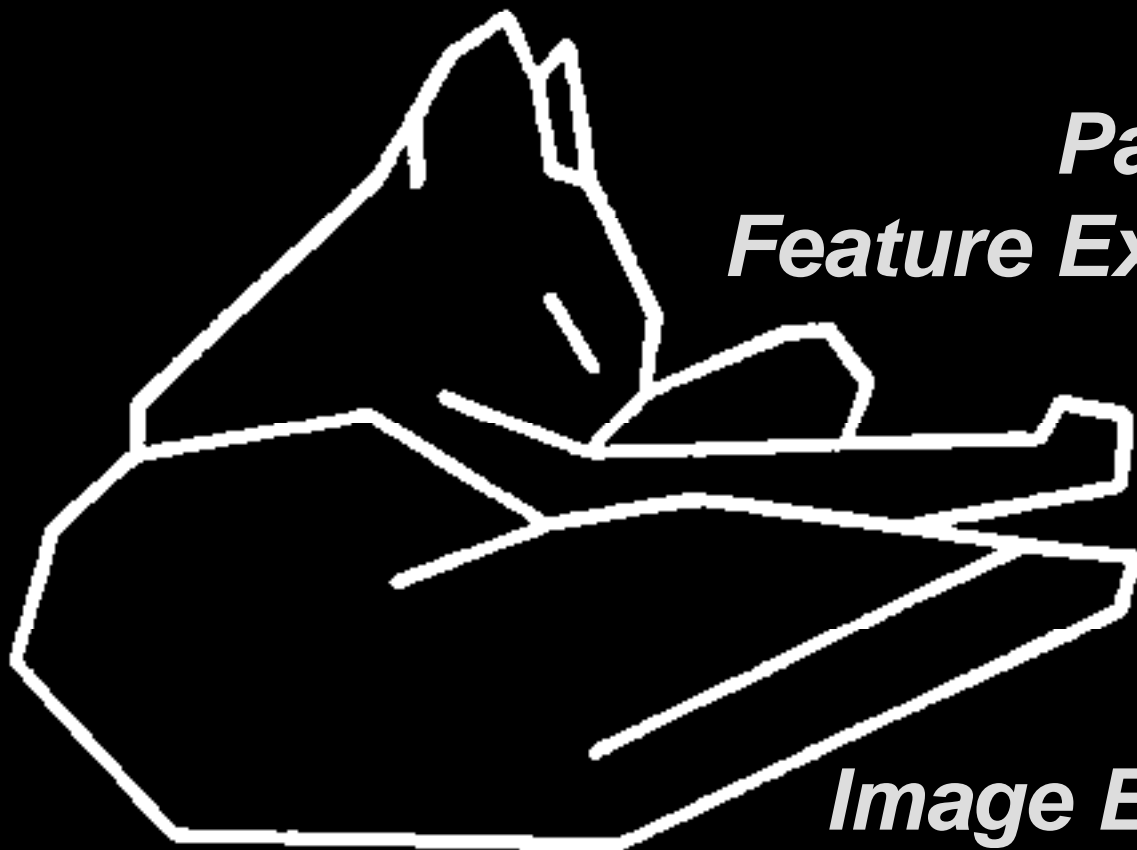
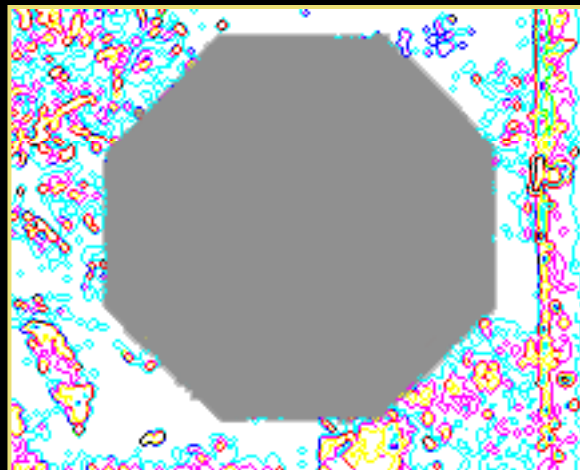
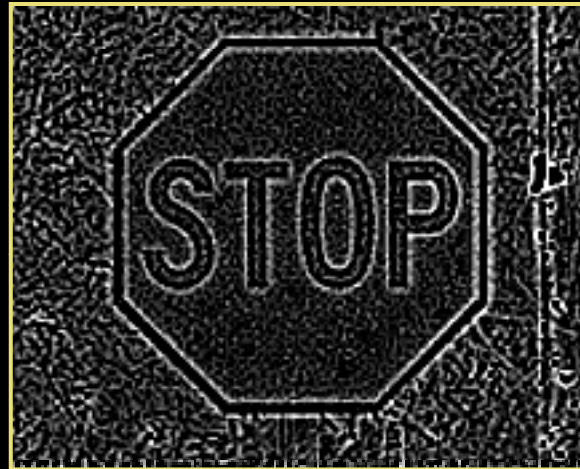
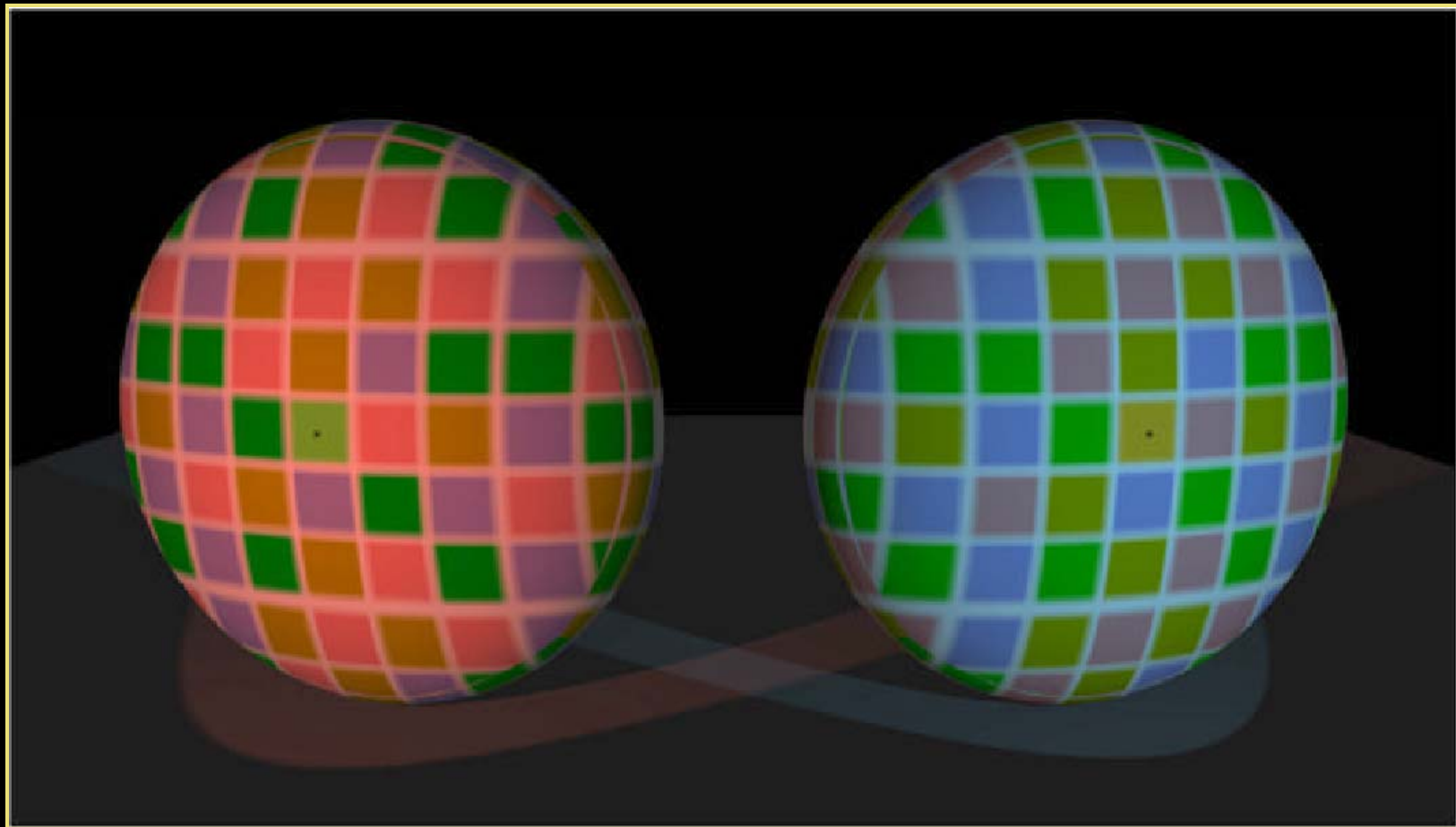


Image Enhancement

Zhigang Zhu, City College of New York zhu@cs.ccny.cuny.edu

- Local, meaningful, detectable parts of the image.

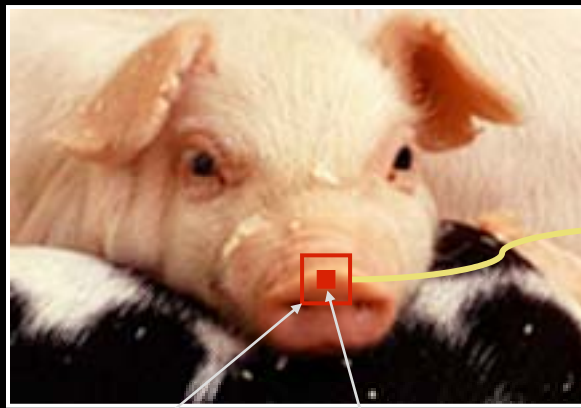




Squares with dots in them are the same color

- Image Enhancement
 - Brightness mapping
 - Contrast stretching/enhancement
 - Histogram modification
 - Noise Reduction
 -
- Mathematical Techniques
 - Convolution
 - Gaussian Filtering
- Edge and Line Detection and Extraction
- Region Segmentation
- Contour Extraction
- Corner Detection

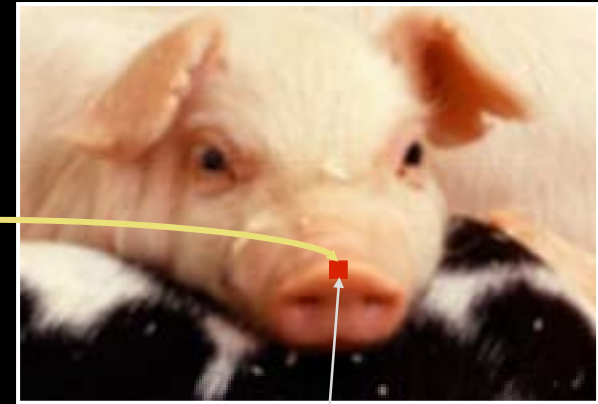
- Goal: improve the ‘visual quality’ of the image
 - for human viewing
 - for subsequent processing
- Two typical methods
 - spatial domain techniques....
 - ◆ operate directly on image pixels
 - frequency domain techniques....
 - ◆ operate on the Fourier transform of the image
- No general theory of ‘visual quality’
 - General assumption: if it looks better, it is better
 - Often not a good assumption



neighborhood N

$I(x,y)$

$T(I(x,y))$



$I'(x,y)$

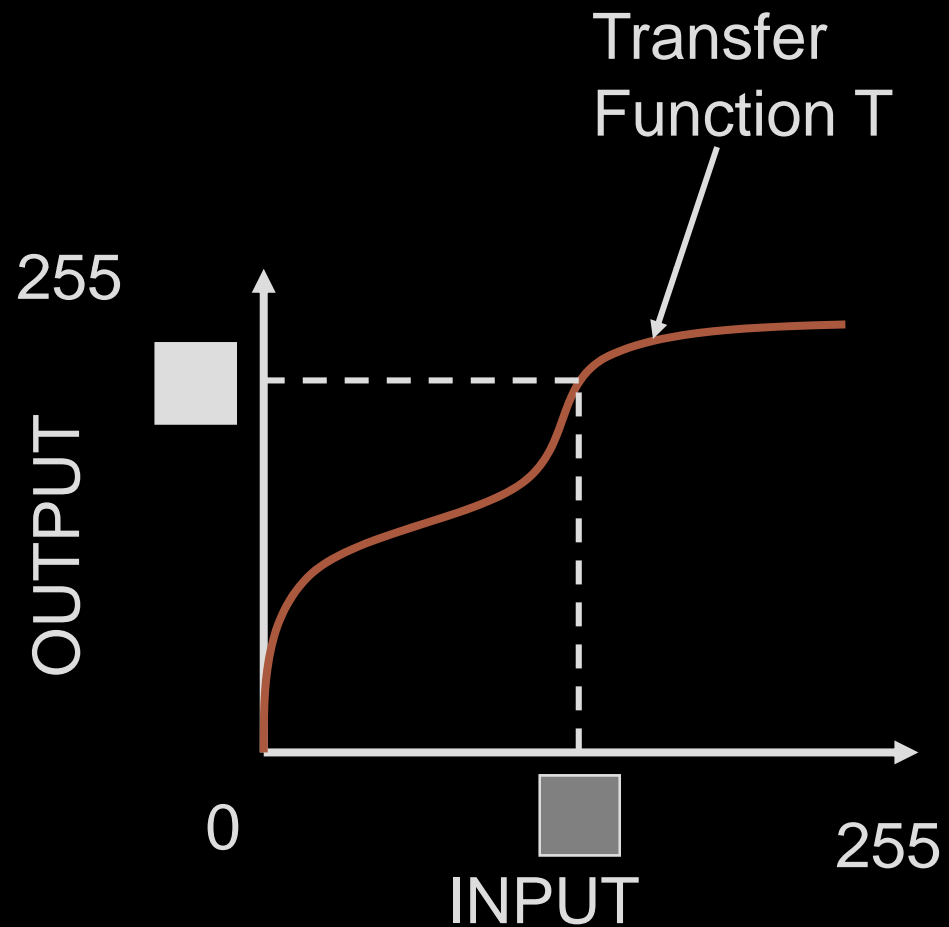
- Transformation T
 - point - pixel to pixel
 - area - local area to pixel
 - global - entire image to pixel
- Neighborhoods
 - typically rectangular
 - typically an odd size: 3x3, 5x5, etc
 - centered on pixel $I(x,y)$
- Many IP algorithms rely on this basic notion

$$I'(x,y) = T(I(x,y))$$

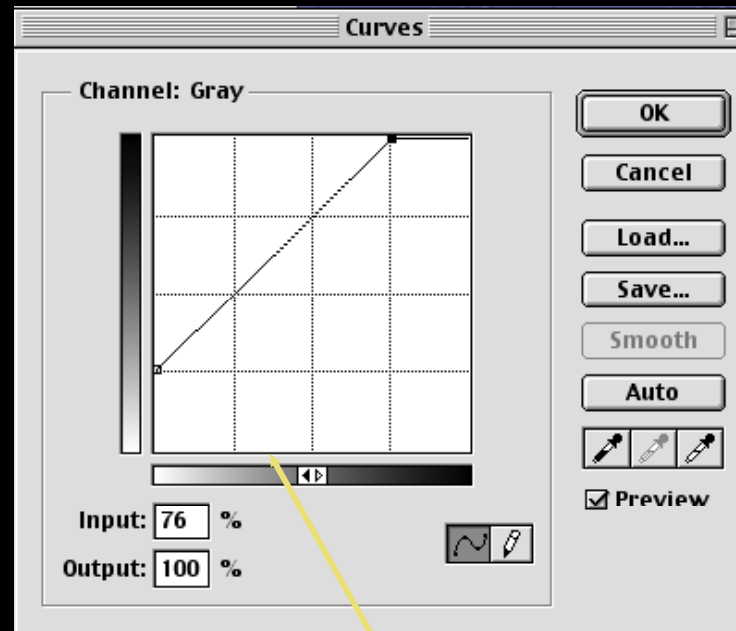
$$O = T(I)$$

$$O = T(I)$$

Input pixel value, I , mapped to output pixel value, O , via transfer function T .

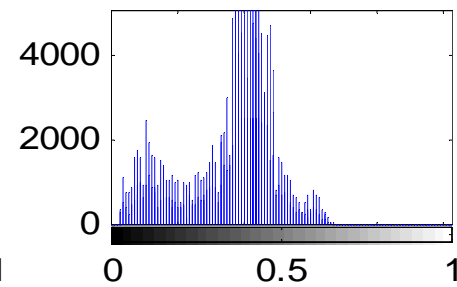
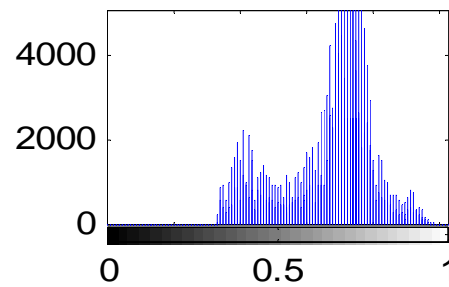
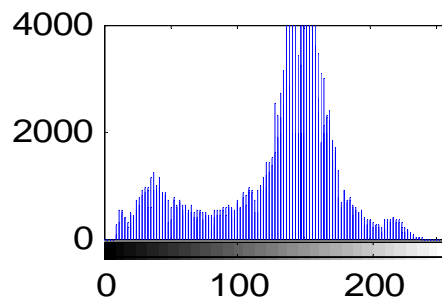
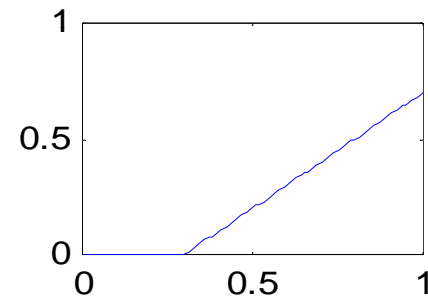
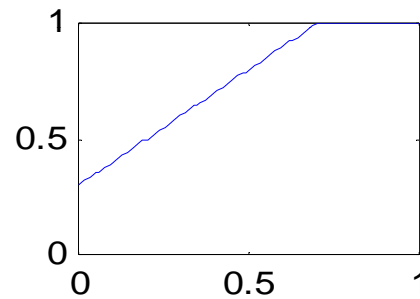
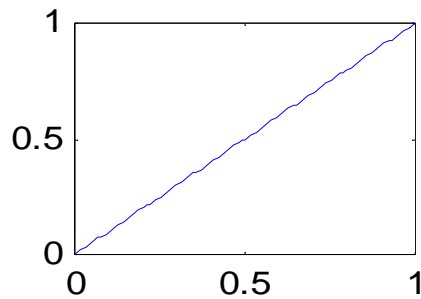


- Photoshop 'adjust curve' command



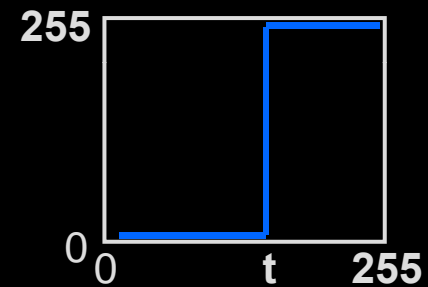
Output gray value $I'(x,y)$

Input gray value $I(x,y)$



- T is a point-to-point transformation
 - only information at $I(x,y)$ used to generate $I'(x,y)$
- Thresholding

$$I'(x,y) = \begin{cases} I_{\max} & \text{if } I(x,y) > t \\ I_{\min} & \text{if } I(x,y) \leq t \end{cases}$$

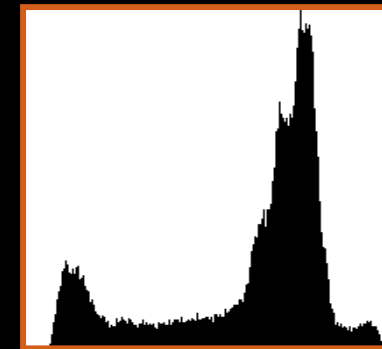
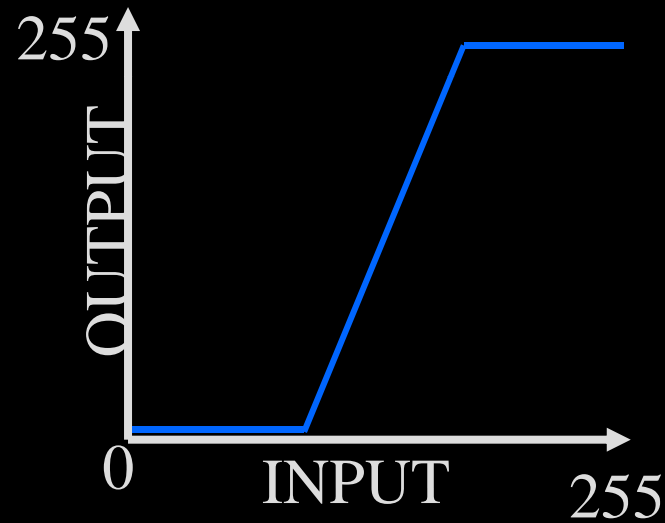
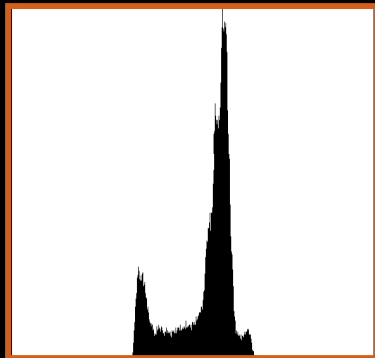


t=89

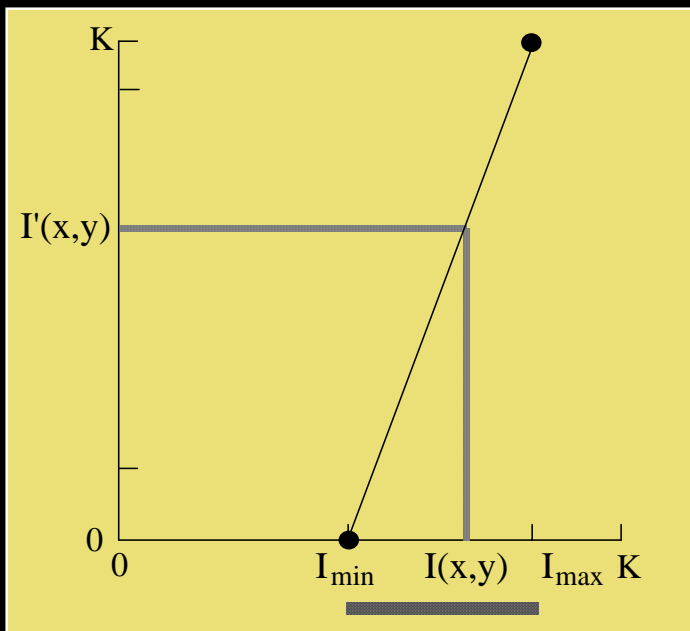
Introduction to

Computer Vision

Point Transforms: Linear Stretch



- Consider the case where the original image only utilizes a small subset of the full range of gray values:
- New image uses full range of gray values.
- What's F ? {just the equation of the straight line}



Input image $I(x,y)$

Gray scale range: $[I_{\min}, I_{\max}]$

Output image $I'(x,y) = F [I(x,y)]$

Desired gray scale range: $[0, K]$

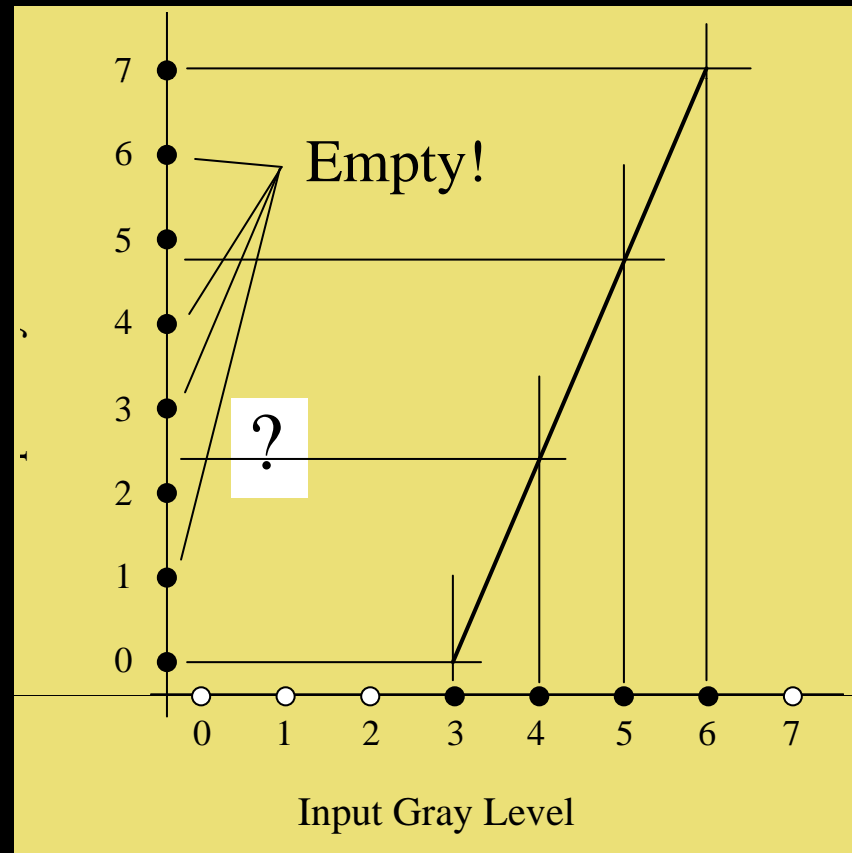
- F is the equation of the straight line going through the point $(I_{\min}, 0)$ and (I_{\max}, K)

$$I'(x,y) = \frac{K}{I_{\max} - I_{\min}} I(x,y) - \frac{K}{I_{\max} - I_{\min}} I_{\min}$$


$$I' = mI + b$$

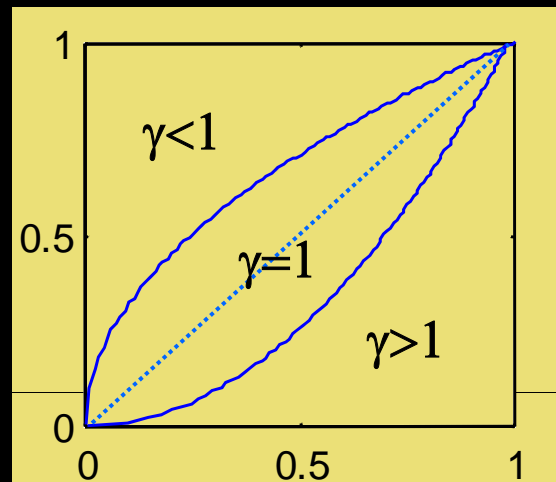
- useful when the image gray values do not fill the available range.
- Implement via lookup tables

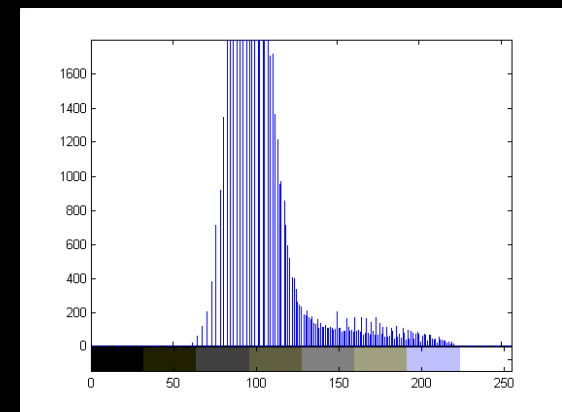
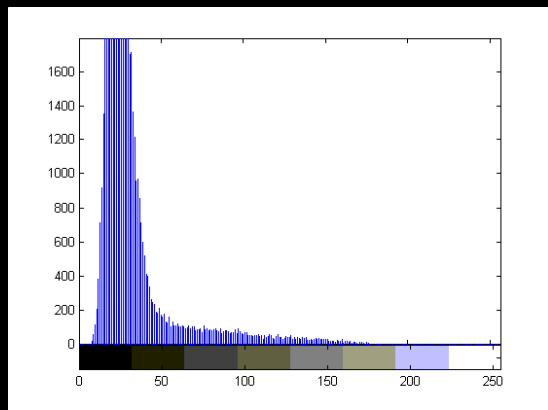
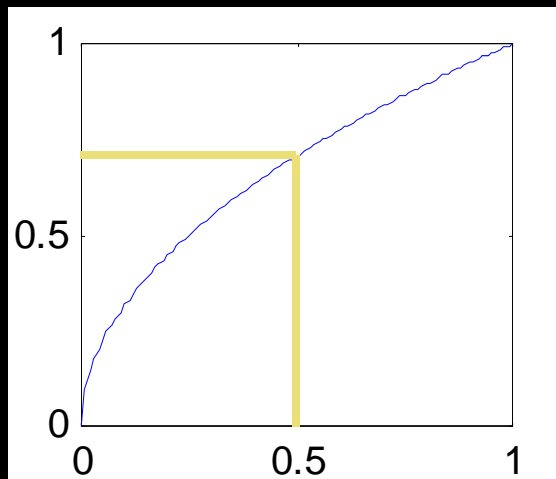
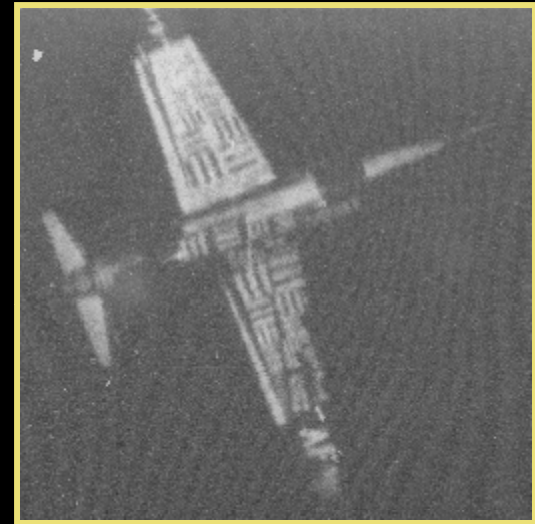
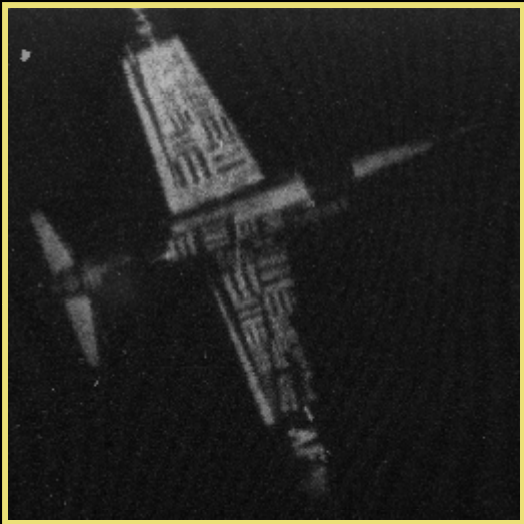
- Have assumed a continuous grayscale.
- What happens in the case of a discrete grayscale with K levels?



$$O = I^\gamma$$

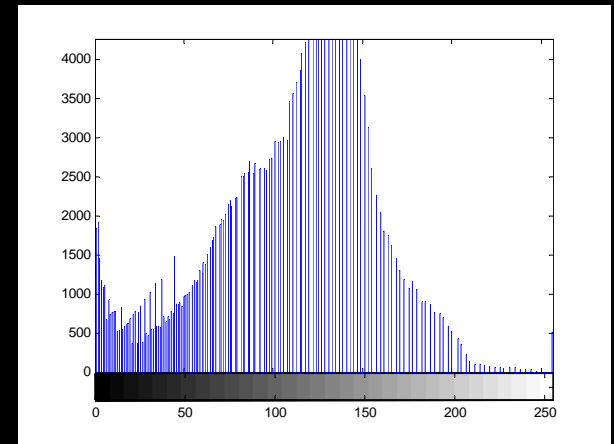
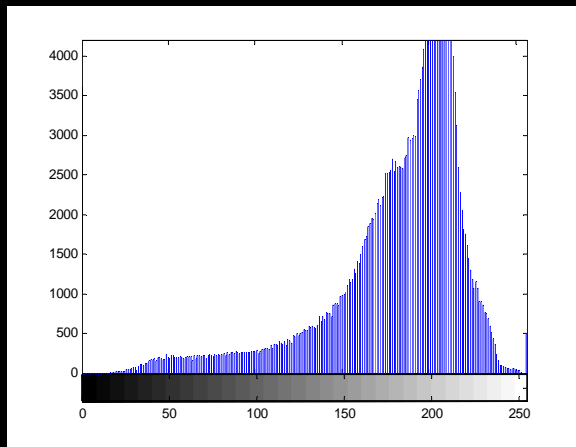
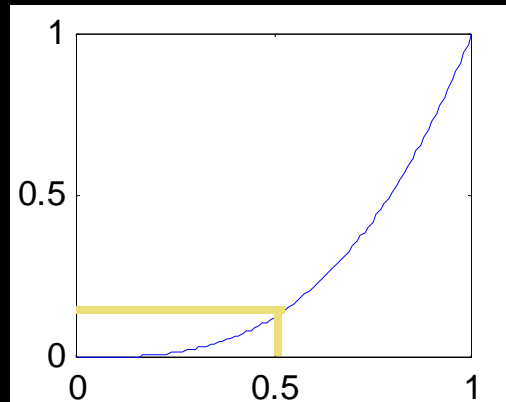
- $\gamma < 1$ to enhance contrast in dark regions
- $\gamma > 1$ to enhance contrast in bright regions.





Introduction to
Computer Vision

$\gamma=3.0$

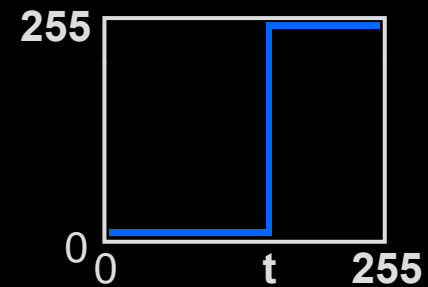


- Technique can be applied to color images
 - same curve to all color bands
 - different curves to separate color bands:



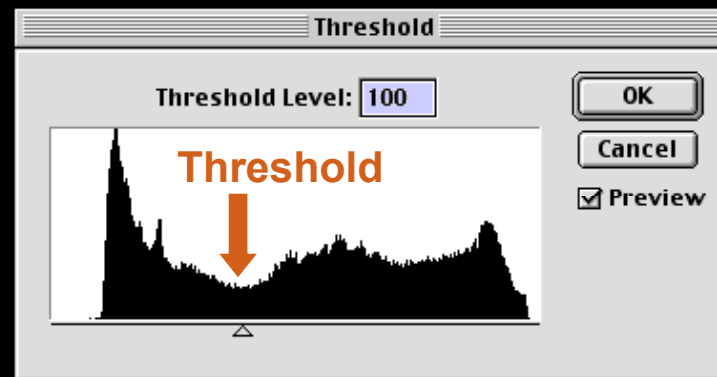
- T is a point-to-point transformation
 - only information at $I(x,y)$ used to generate $I'(x,y)$
- Thresholding

$$I'(x,y) = \begin{cases} I_{\max} & \text{if } I(x,y) > t \\ I_{\min} & \text{if } I(x,y) \leq t \end{cases}$$



t=89

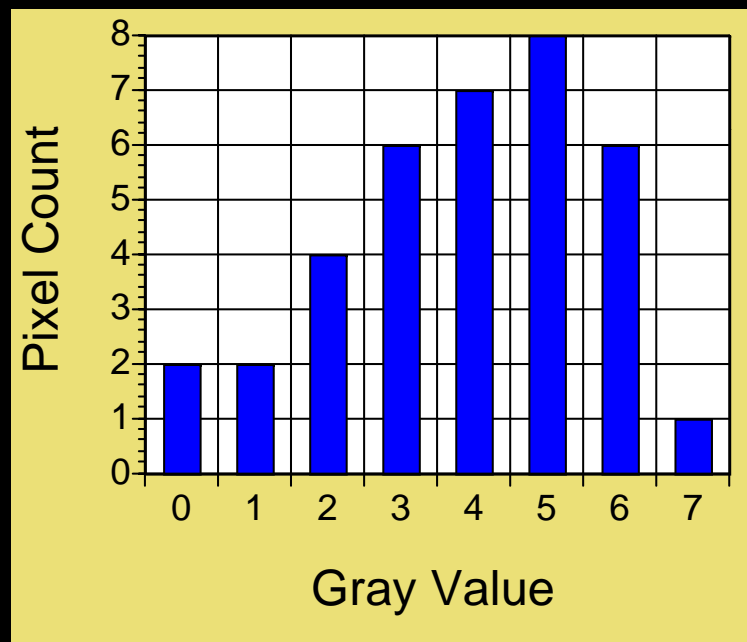
- Arbitrary selection
 - select visually
- Use image histogram



- The image shows the spatial distribution of gray values.
- The image histogram discards the spatial information and shows the relative frequency of occurrence of the gray values.

Image	Gray Value	Count	Rel. Freq.
0 3 3 2 5 5	0	2	.05
1 1 0 3 4 5	1	2	.05
2 2 2 4 4 4	2	4	.11
3 3 4 4 5 5	3	6	.17
3 4 5 5 6 6	4	7	.20
7 6 6 6 6 5	5	8	.22
	6	6	.17
	7	1	.03
	Sum=	36	1.00

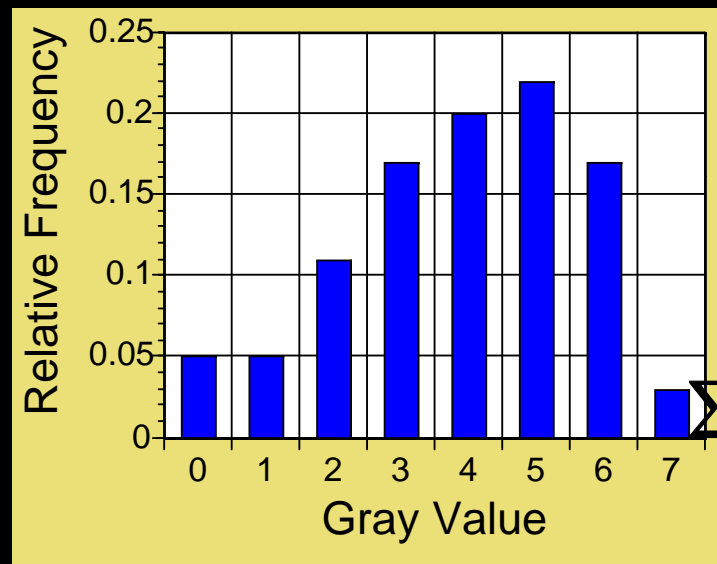
- The histogram typically plots the absolute pixel count as a function of gray value:



For an image with dimensions M by N

$$\sum_{i=I_{\min}}^{I_{\min}} H(i) = MN$$

- The graph of relative frequency of occurrence as a function of gray value is also called a histogram:

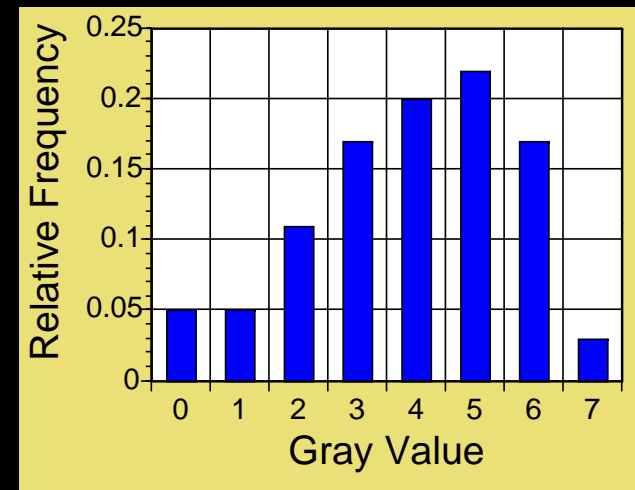


- Interpreting the relative frequency histogram as a probability distribution, then:

$$P(I(x,y) = i) = H(i)/(M \times N)$$

- Interpreting the relative frequency histogram as a probability distribution, then:

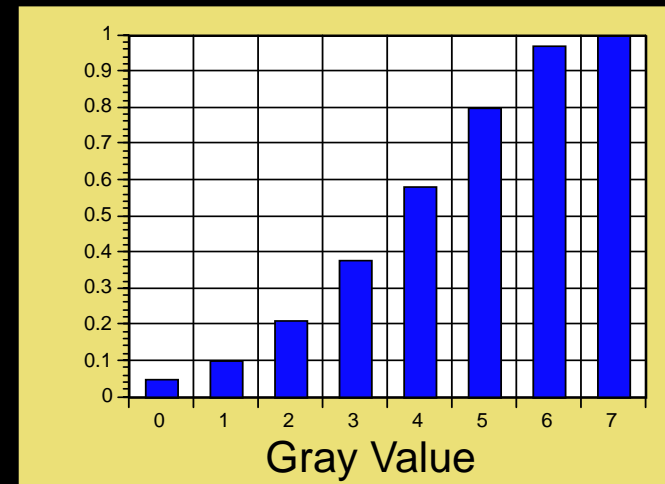
$$P(I(x,y) = i) = H(i)/(M \times N)$$

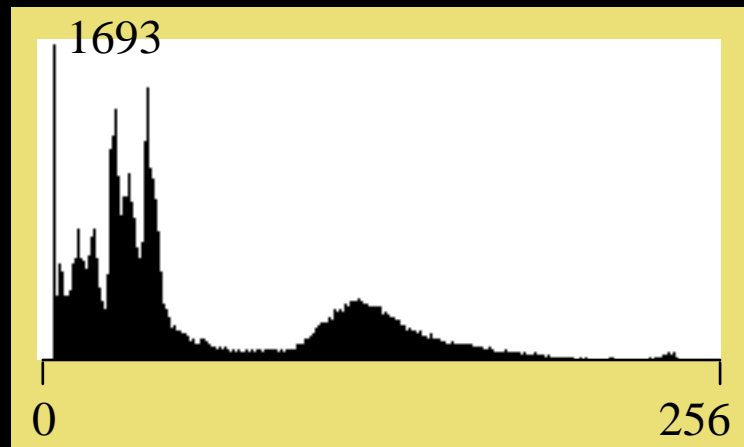
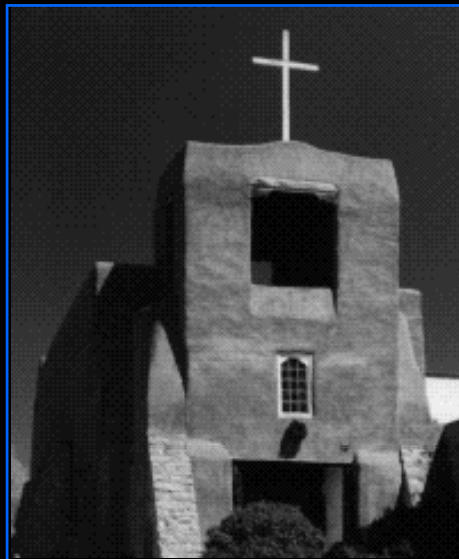
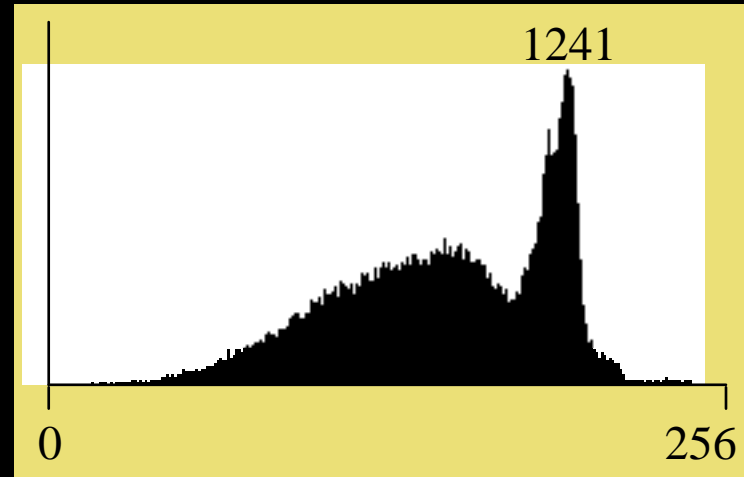


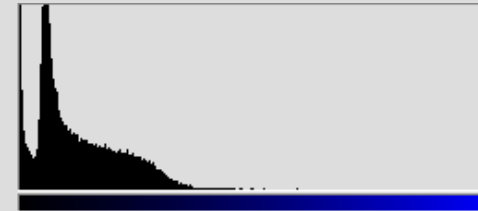
- Curve is called the cumulative distribution function

$$Q(i) = \sum_{k=0}^i P(k)$$

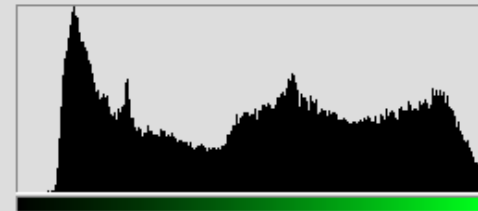
$$CH(i) = \sum_{k=0}^i H(k)$$



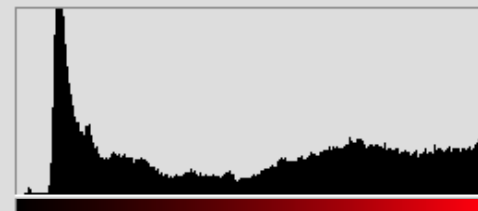




Mean: 25.80 Level:
Std Dev: 23.47 Count:
Median: 18 Percentile:
Pixels: 124416 Cache Level: 1

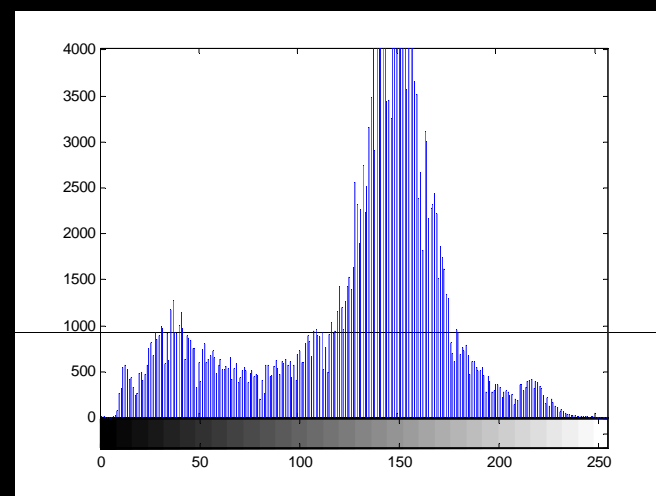


Mean: 130.63 Level:
Std Dev: 69.51 Count:
Median: 137 Percentile:
Pixels: 124416 Cache Level: 1

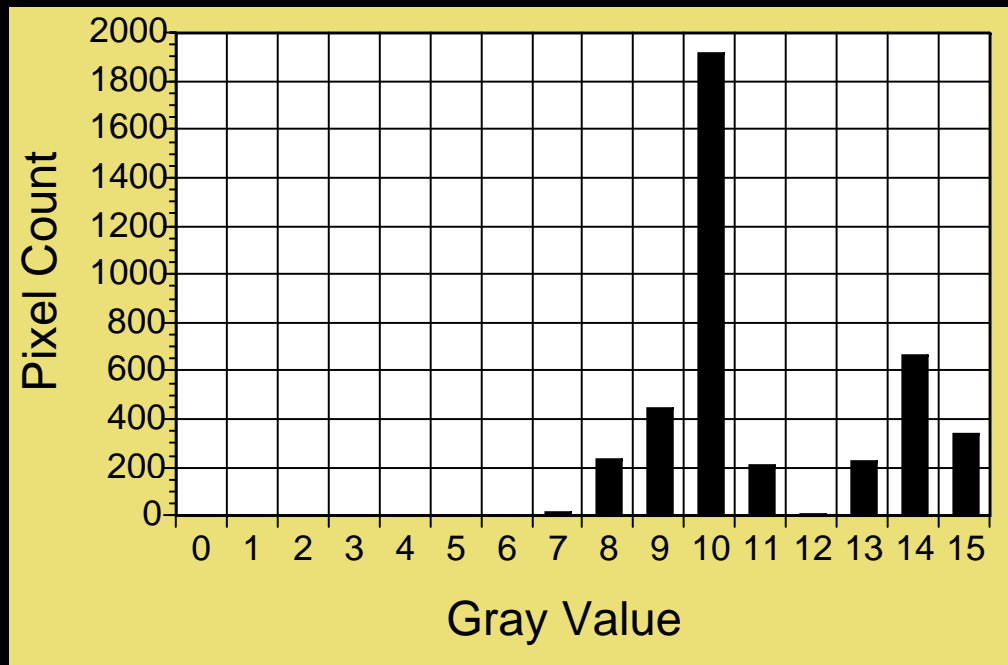


Mean: 139.63 Level:
Std Dev: 85.63 Count:
Median: 156 Percentile:
Pixels: 124416 Cache Level: 1

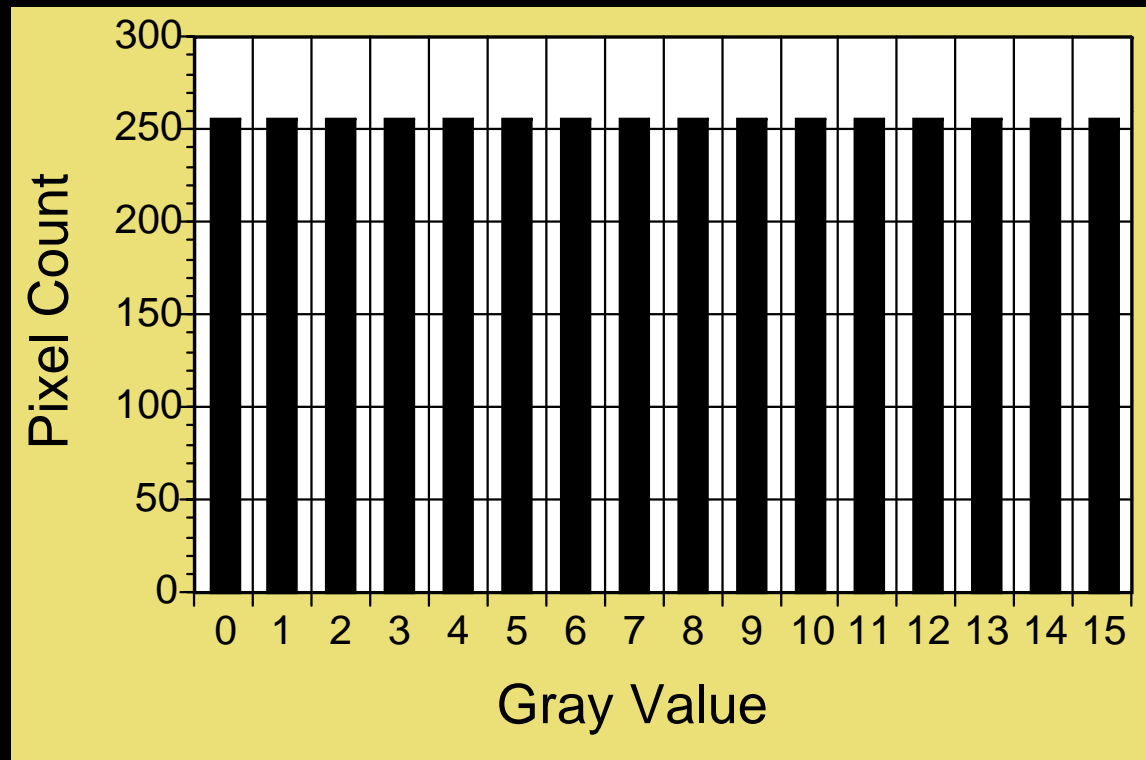
- Image histograms consist of peaks, valleys, and low plains
- Peaks = many pixels concentrated in a few grey levels
- Plains = small number of pixels distributed over a wider range of grey levels



- The goal is to modify the gray levels of an image so that the histogram of the modified image is flat.
 - Expand pixels in peaks over a wider range of gray-levels.
 - “Squeeze” low plains pixels into a narrower range of gray levels.
- Utilizes all gray values equally
- Example Histogram:



- All gray levels are used equally.
- Has a tendency to enhance contrast.



Gray Scale	Actual Count	Desired Count	How to get it
0	0	256	22 from 7, 234 from 8
1	0	256	1 from 8, 255 from 9
2	0	256	195 from 9, 61 from 10
3	0	256	256 from 10
4	0	256	256 from 10
5	0	256	256 from 10
6	0	256	256 from 10
7	22	256	256 from 10
8	235	256	256 from 10
9	450	256	256 from 10
10	1920	256	67 from 10, 189 from 11
11	212	256	23 from 11, 10 from 12, 223 from 13
12	10	256	10 from 13, 246 from 14
13	233	256	256 from 14
14	672	256	170 from 14, 86 from 15
15	342	256	256 from 15
Sum	4096	4096	

How are the gray levels in the original image changed to produce the enhanced image?

Method 1. Choose points randomly.

Method 2. Choice depends on the gray levels of their neighboring points.

Computationally expensive.

Approximations.

- Mapping from one set of grey-levels, I , to a new set, O .
- Ideally, the number of pixels, N_p , occupying each grey level should be:

$$N_p = \frac{M*N}{G}$$

- To approximate this, apply the transform

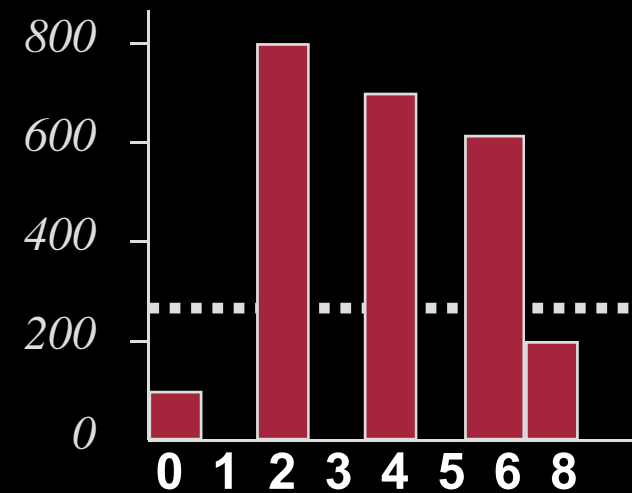
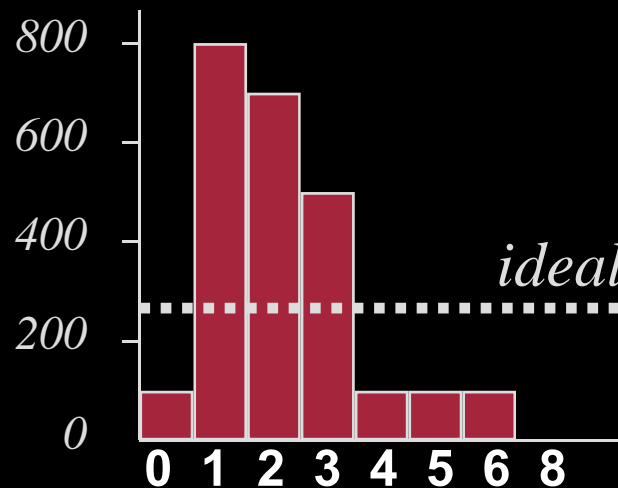
$$i = \text{MAX} \left[0, \text{round} \left\{ \frac{\text{CH}(j)}{N_p} \right\} - 1 \right]$$

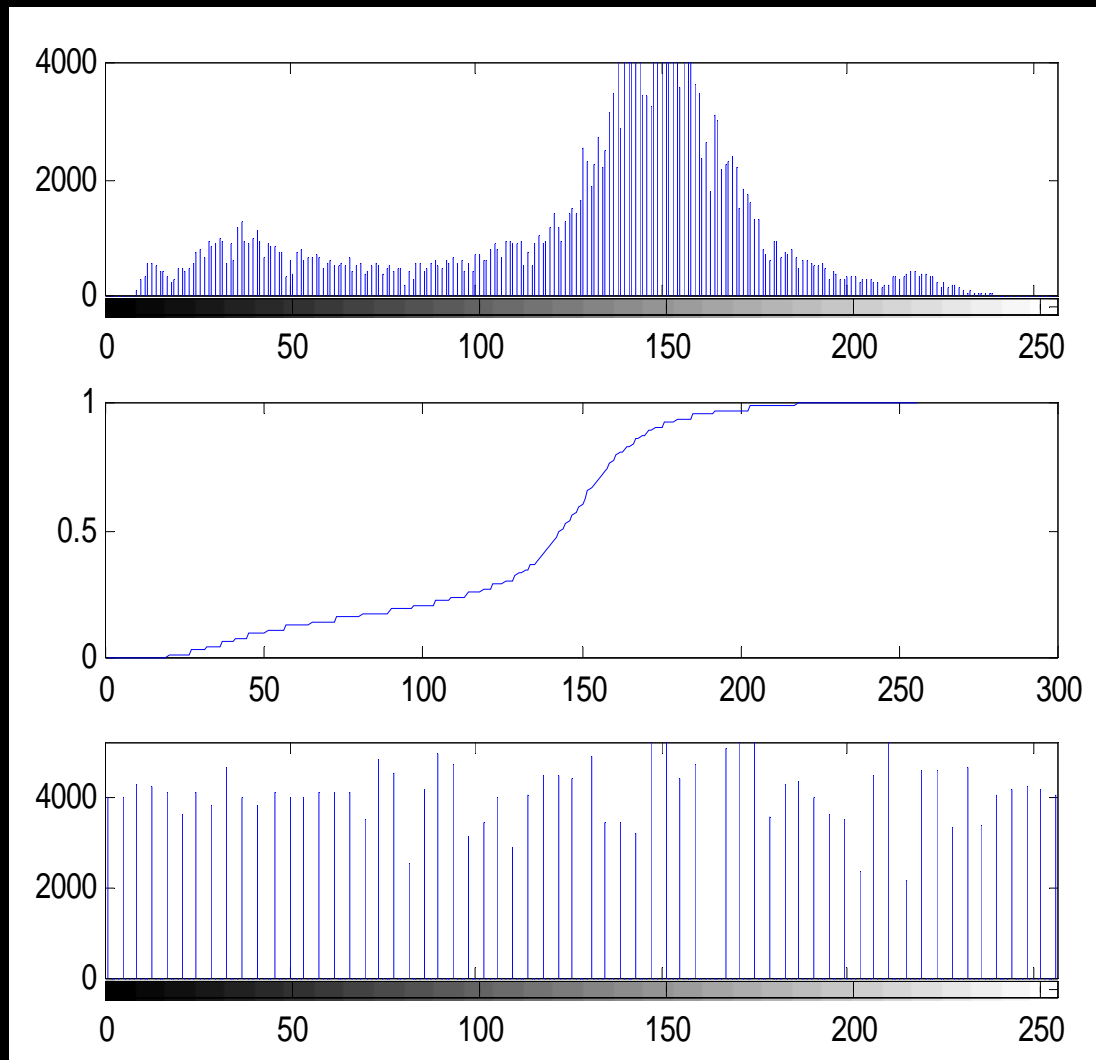
- Where CH is the cumulative histogram (see next slide)
- j is the gray value in the source image
- i is the gray value in the equalized image

$G=8$
 $M \times N = 2400$
 $N_p = 300$

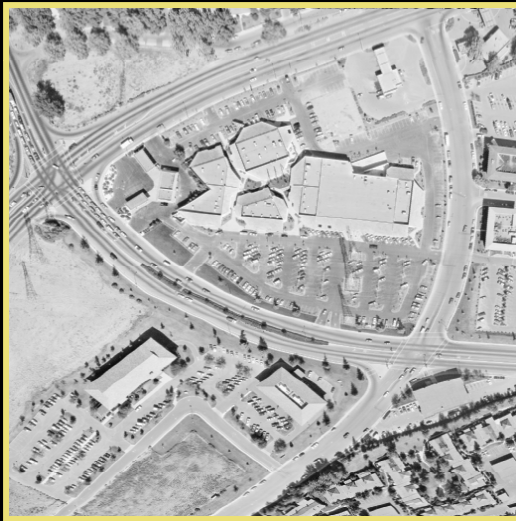
$$CH(j) = \sum_{i=0}^j H(i)$$

j	$H(j)$	$CH(j)$	i
0	100	100	0
1	800	900	2
2	700	1600	4
3	500	2100	6
4	100	2200	6
5	100	2300	7
6	100	2400	7
7	0	2400	7



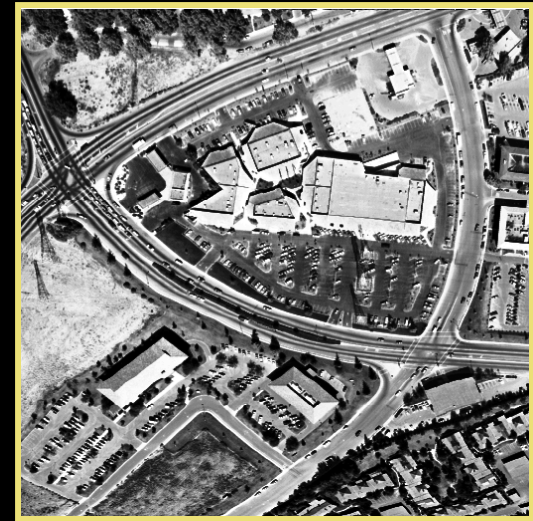
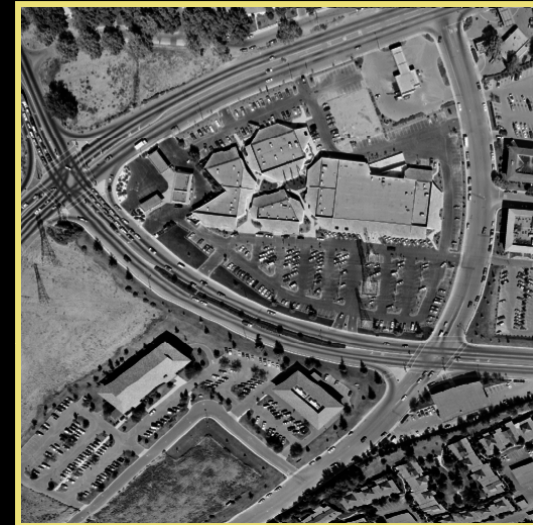


Original



$\gamma > 1$

Histogram
equalization



- Given $M \times N$ input image I with gray scale $p_0 \dots p_k$ and histogram $H(p)$.
- Desire output image O with gray scale $q_0 \dots q_k$ and uniform histogram $G(q)$
- Treat the histograms as a discrete probability density function. Then monotonic transfer function $m = T(n)$ implies:

$$\sum_{i=0}^m G(q_i) = \sum_{j=0}^n H(p_j)$$

- The sums can be interpreted as discrete distribution functions

- The histogram entries in G must be:
(because G is uniform)

$$G(i) = \frac{M \times N}{q_k - q_0}$$

- Plug this into previous equation to get:

$$\sum_{i=0}^m \frac{M \times N}{q_k - q_0} = \sum_{i=0}^n H(p_i)$$

- Now translate into continuous domain (can't really get uniform distributions in the discrete domain)

$$\int_{q_0}^{q_m} \frac{M \times N}{q_k - q_0} ds = \int_{p_0}^{p_n} H(s) ds$$

$$\frac{MN(q_m - q_0)}{q_k - q_0}$$

- Solve last equation for q to get:

$$q_m = T(p) = \frac{q_k - q_0}{MN} \int_{p_0}^{p_n} H(s) ds + q_0$$

- Map back into discrete domain to get:

$$q_m = T(p) = \frac{q_k - q_0}{MN} \sum_{i=p_0}^{p_n} H(i) + q_0$$

- Leads to an algorithm.....

- For an $N \times M$ image of G gray levels, say 0-255
 - Create image histogram
 - For cumulative image histogram H_c

- Set

$$T(p) = \text{round} \left(\frac{G-1}{NM} H_c(p) \right)$$

- Rescan input image and write new output image by setting

$$g_q = T(g_p)$$

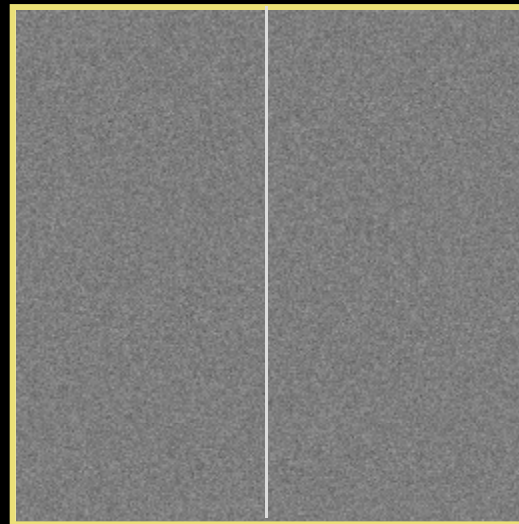
- Acquisition process degrades image
- Brightness and contrast enhancement implemented by pixel operations
- No one algorithm universally useful
- $\gamma > 1$ enhances contrast in bright images
- $\gamma < 1$ enhances contrast in dark images
- Transfer function for histogram equalisation proportional to cumulative histogram

- What is noise?
- How is noise reduction performed?
 - Noise reduction from first principles
 - Neighbourhood operators
 - ◆ linear filters (low pass, high pass)
 - ◆ non-linear filters (median)



image

+



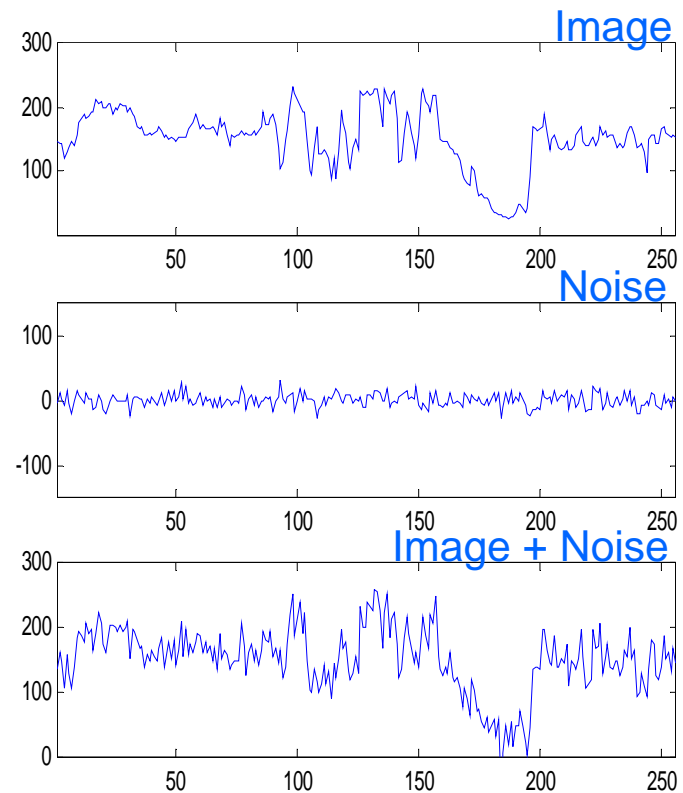
noise

=

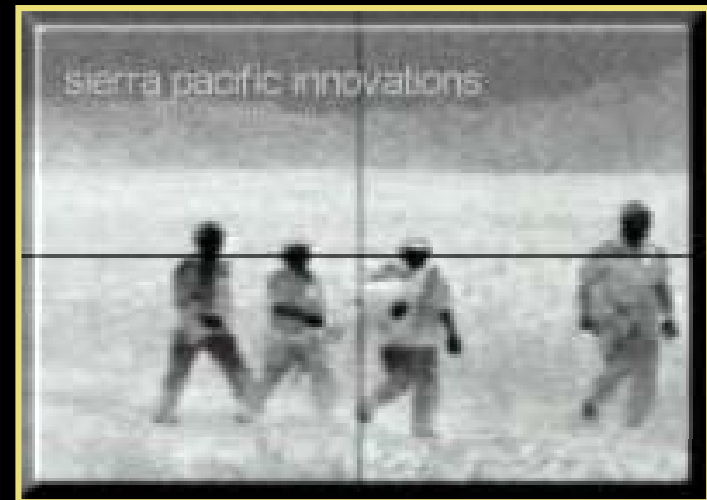
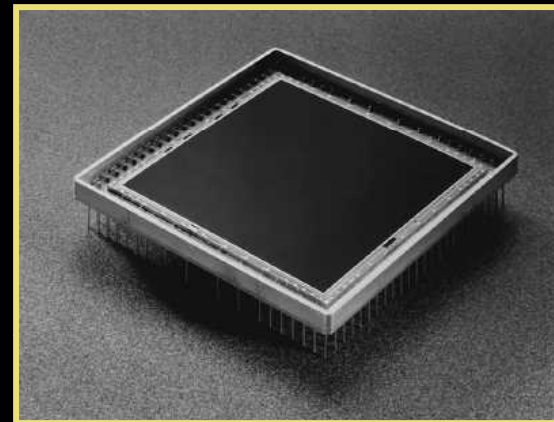


'grainy' image

- Plot of image brightness.
- Noise is additive.
- Noise fluctuations are rapid
 - high frequency.



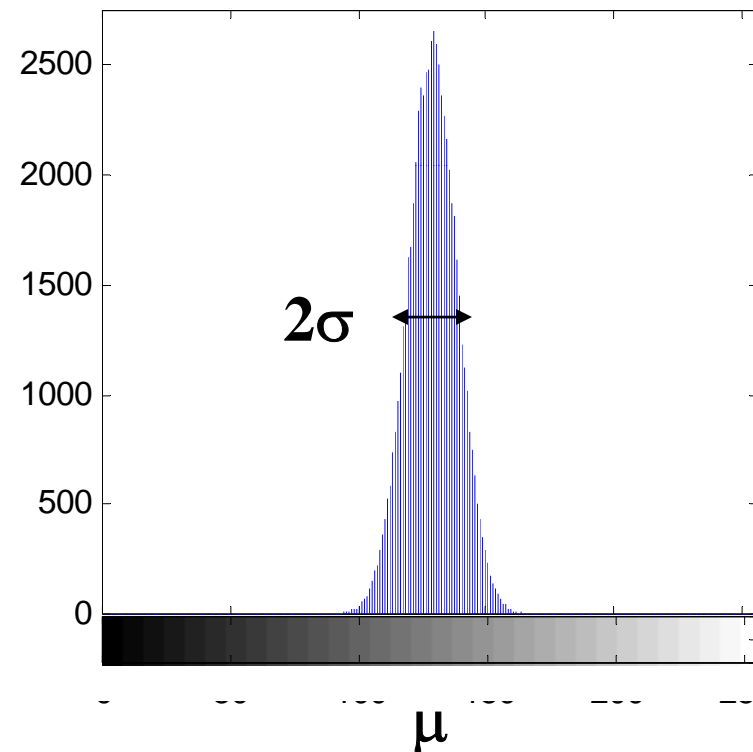
- Sources of noise = CCD chip.
- Electronic signal fluctuations in detector.
- Caused by thermal energy.
- Worse for infra-red sensors.
- Electronics
- Transmission



Radiation from the long wavelength IR band is used in most infrared imaging applications

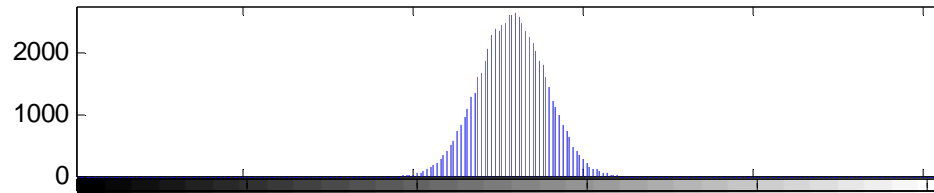
- Plot noise histogram
- Typical noise distribution is normal or Gaussian
- Mean(noise) $\mu = 0$
- Standard deviation σ

$$\eta(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right]$$

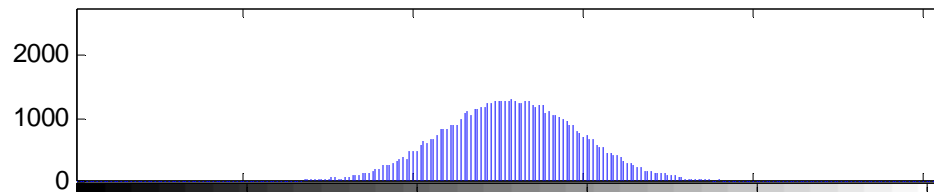


- Integral under curve is 1

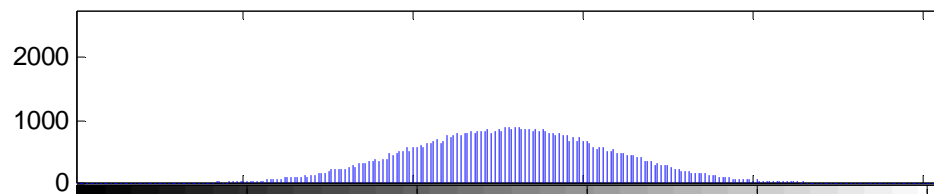
$\sigma=10$



$\sigma=20$



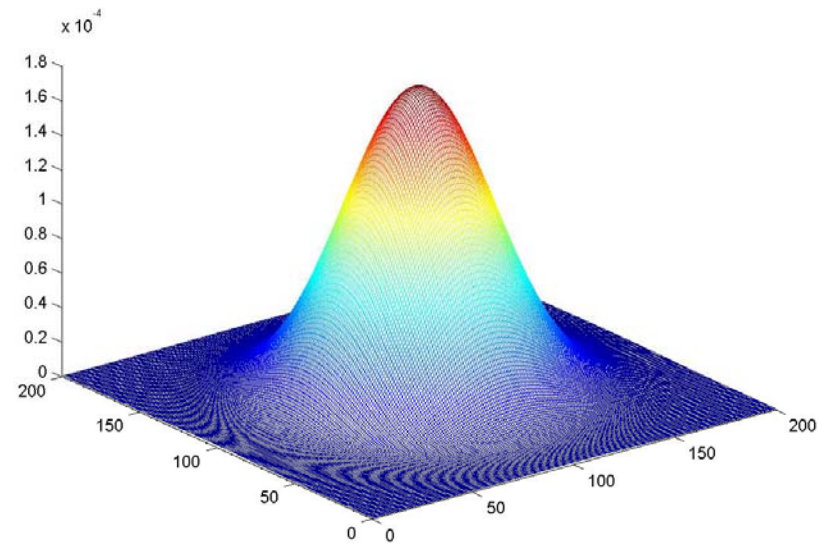
$\sigma=30$



Introduction to

Computer Vision

Two Dimensional Gaussian



- Noise varies above and below uncorrupted image.

Image

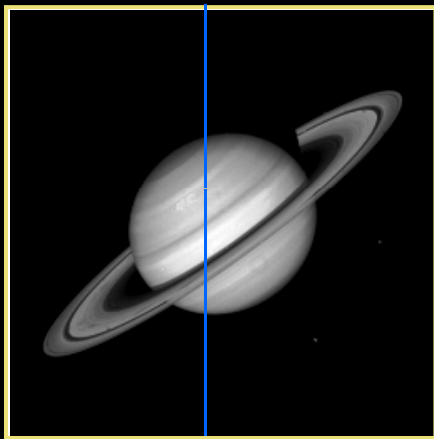
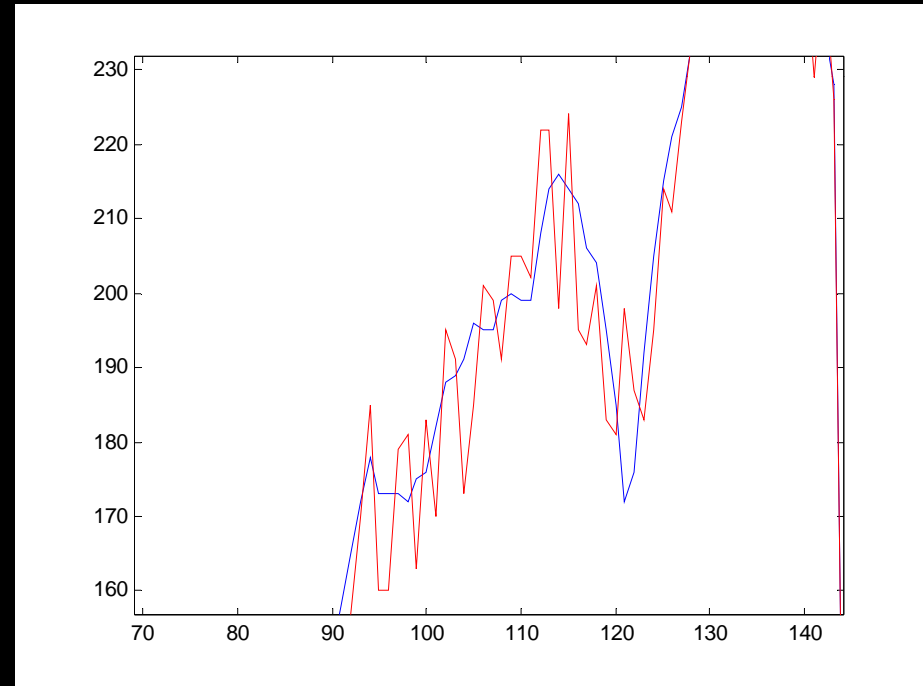
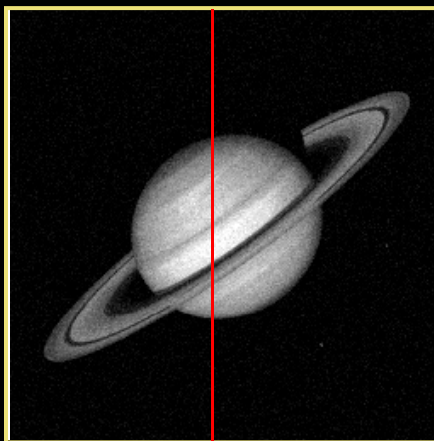
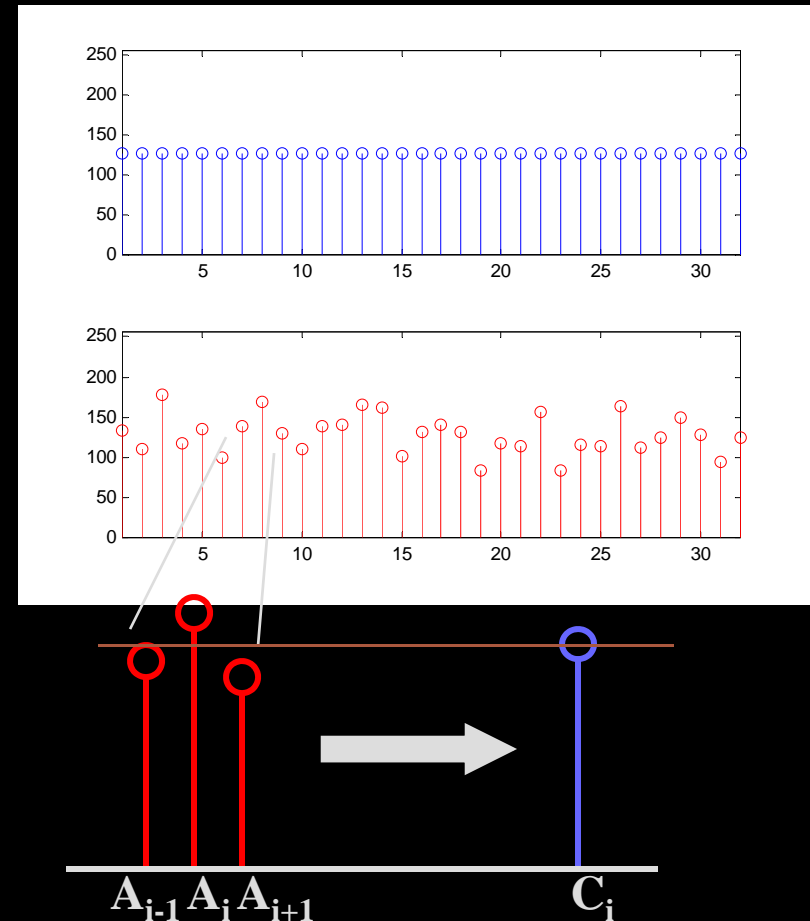


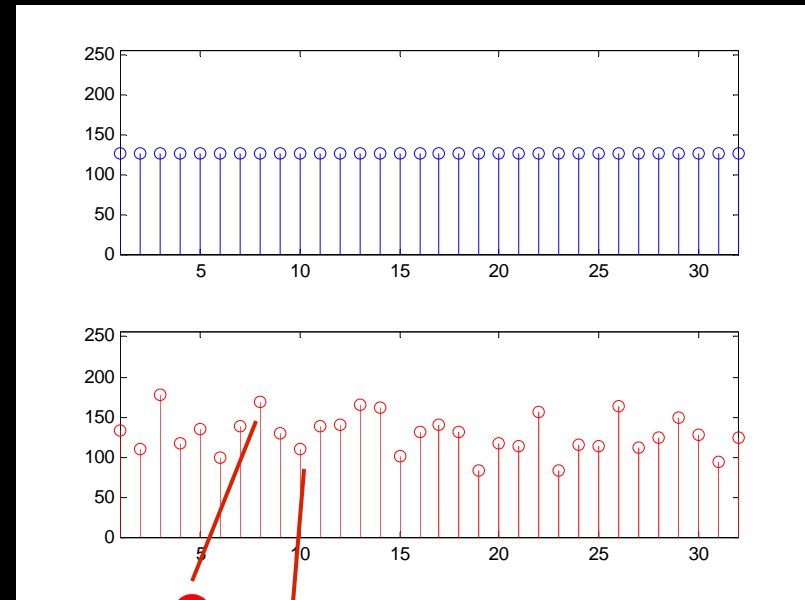
Image
+
Noise



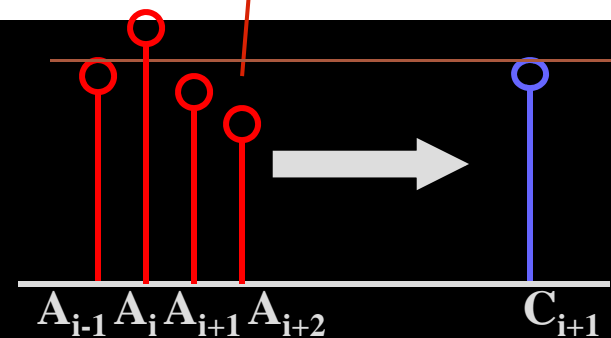
- How do we reduce noise?
- Consider a uniform 1-d image and add noise.
- Focus on a pixel neighbourhood.
- Averaging the three values should result in a value closer to original uncorrupted data
- Especially if gaussian mean is zero



- Averaging 'smooths' the noise fluctuations.
- Consider the next pixel A_{i+1}
- Repeat for remainder of pixels.



$$C_{i+1} = \frac{A_i + A_{i+1} + A_{i+2}}{3}$$



- All pixels can be averaged by **convolving** 1-d image A with mask B to give enhanced image C.
- Weights of B must equal one when added together.
 - Why?

$$C = A * B$$

$$B = [B_1 \quad B_2 \quad B_3]$$

$$C_i = A_{i-1} \times B_1 + A_i \times B_2 + A_{i+1} \times B_3$$

$$B = \frac{1}{3} [1 \quad 1 \quad 1]$$

$$C_i = \frac{A_{i-1} + A_i + A_{i+1}}{3}$$

- Consider the problem of blurring a 2D image
- Here's the image and a blurred version:



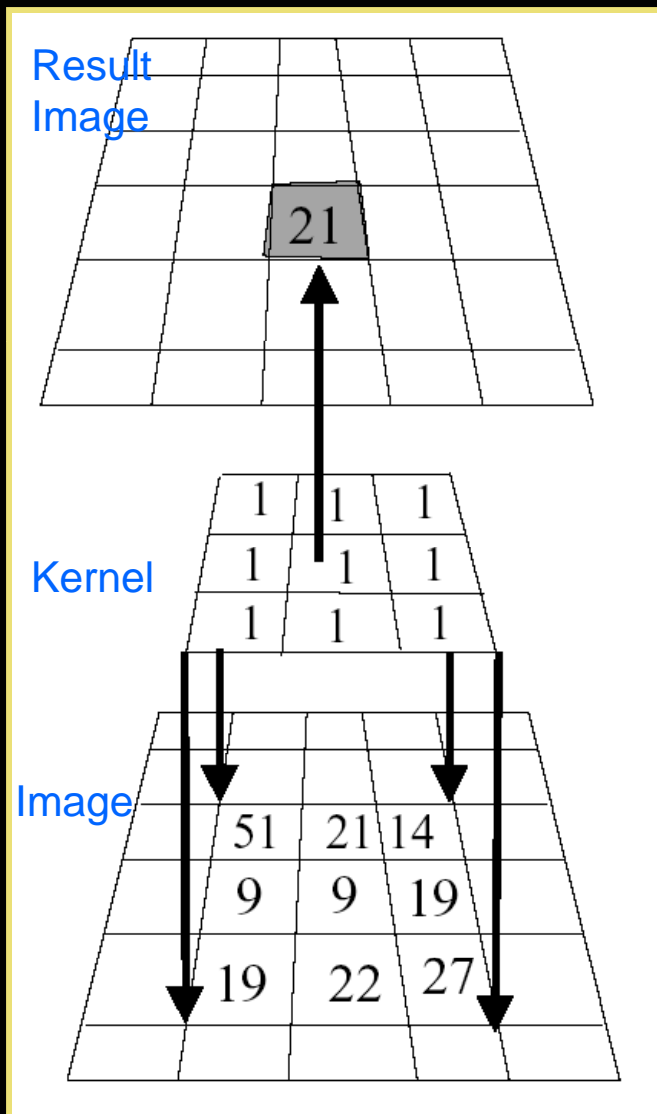
- How would we do this?
 - What's the 2D analog of 1D convolution?

- $C = A * B$
- B becomes

$$B = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



- Extremely important concept in computer vision, image processing, signal processing, etc.
- Lots of related mathematics (**we won't do**)
- General idea: reduce a filtering operation to the repeated application of a mask (or filter kernel) to the image
 - Kernel can be thought of as an $N \times N$ image
 - N is usually odd so kernel has a central pixel
- In practice
 - (flip kernel)
 - Align kernel center pixel with an image pixel
 - Pointwise multiply each kernel pixel value with corresponding image pixel value and add results
 - Resulting sum is normalized by kernel weight
 - Result is the value of the pixel centered on the kernel



- Kernel is aligned with pixel in image, multiplicative sum is computed, normalized, and stored in result image.
- Process is repeated across image.
- What happens when kernel is near edge of input image?

$$\left[\begin{array}{l} 1*51 + 1*21 + 1*14 + \\ 1*8 + 1*9 + 1*19 + \\ 1*19 + 1*22 + 1*27 \end{array} \right] 1/9 = 21$$

- missing samples are zero
- missing samples are gray
- copying last lines
- reflected indexing (mirror)
- circular indexing (periodic)
- truncation of the result

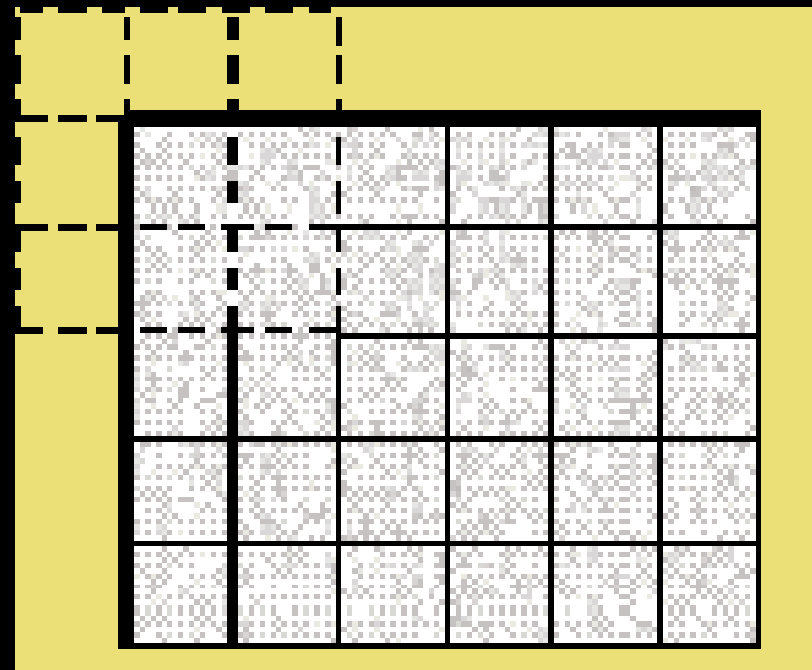


Image size = $M_1 \times N_1$

Mask size = $M_2 \times N_2$

Convolution size =
 $M_1 - M_2 + 1 \times N_1 - N_2 + 1$

Typical Mask sizes
= 3×3 , 5×5 , 7×7 ,
 9×9 , 11×11

What is the convolved
image size for a 128
 $\times 128$ image and
 7×7 mask?

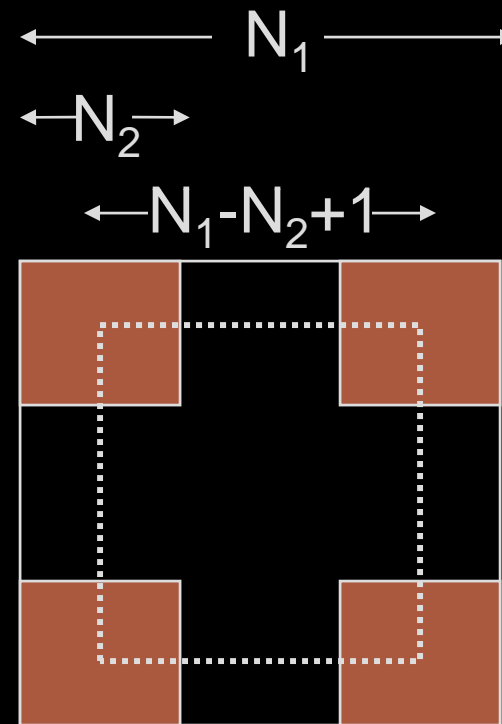


Image Data:

10	12	40	16	19	10
14	22	52	10	55	41
10	14	51	21	14	10
32	22	9	9	19	14
41	18	9	22	27	11
10	7	8	8	4	5

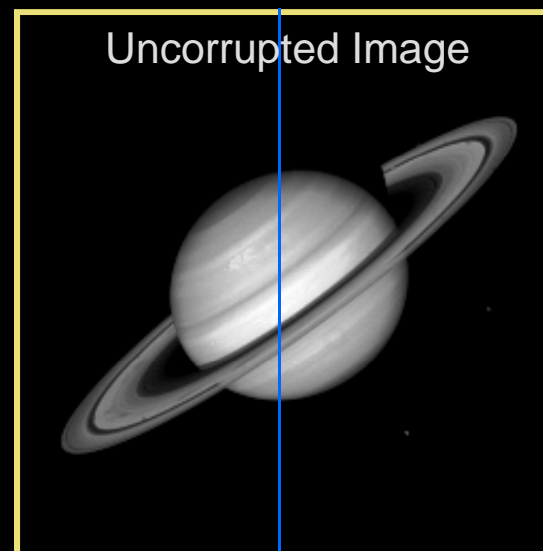
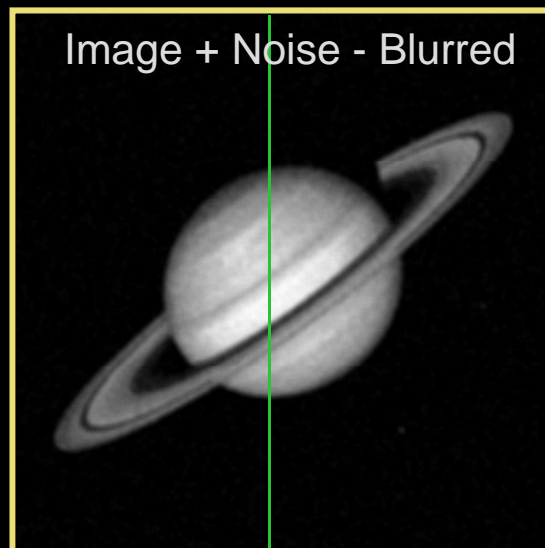
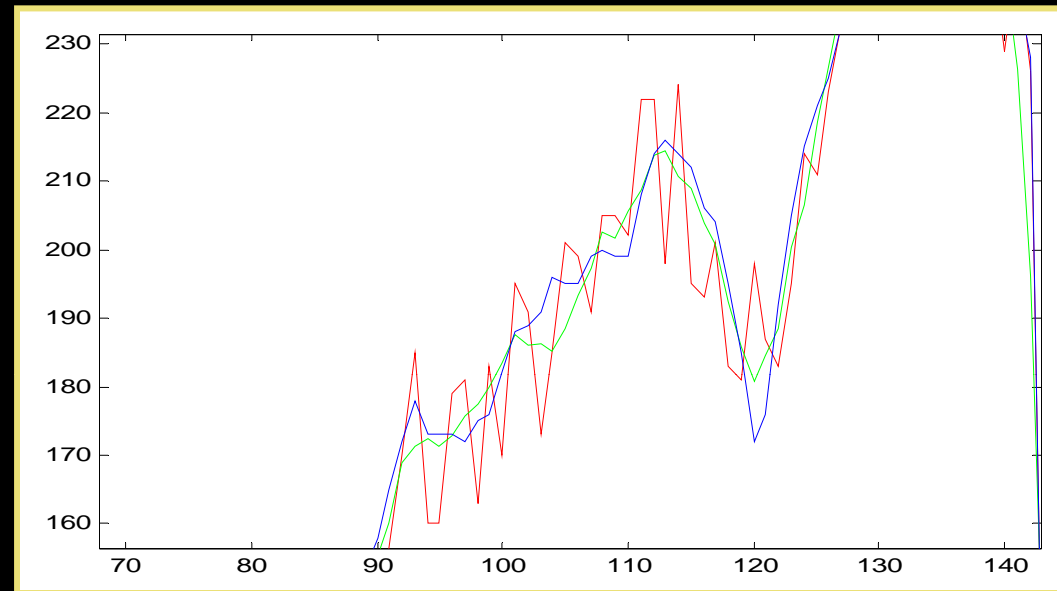
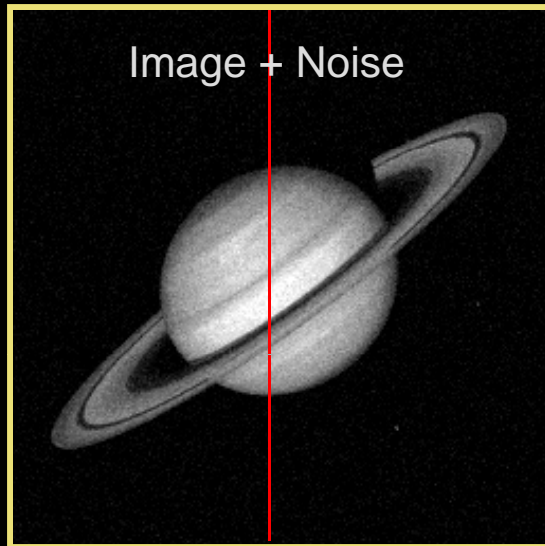
Mask:

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

$$S = \sum_{k=-1}^1 \sum_{m=-1}^1 M(k,m)$$

$$I_f(i,j) = \frac{1}{S} \sum_{k=-1}^1 \sum_{m=-1}^1 I(i+k,j+m) * M(k,m) = I \otimes M$$

- Commutative: $f_1(t) * f_2(t) = f_2(t) * f_1(t)$
- Distributive: $f_1(t) * [f_2(t) + f_3(t)] = f_1(t) * f_2(t) + f_1(t) * f_3(t)$
- Associative: $f_1(t) * [f_2(t) * f_3(t)] = [f_1(t) * f_2(t)] * f_3(t)$
- Shift: if $f_1(t) * f_2(t) = c(t)$, then
 - $f_1(t) * f_2(t-T) = f_1(t-T) * f_2(t) = c(t-T)$
- Convolution with impulse: $f(t) * \delta(t) = f(t)$
- Convolution with shifted impulse: $f(t) * \delta(t-T) = f(t-T)$



- Technique relies on high frequency noise fluctuations being 'blocked' by filter. Hence, **low-pass** filter.
- Fine detail in image may also be smoothed.
- Balance between keeping image fine detail and reducing noise.

- Saturn image coarse detail
- Boat image contains fine detail
- Noise reduced but fine detail also smoothed



Image



Blurred Image



- Nonlinear Filter
- Compute median of points in neighborhood
- Has a tendency to blur detail less than averaging
- Works very well for 'shot' or 'salt and pepper' noise



Original

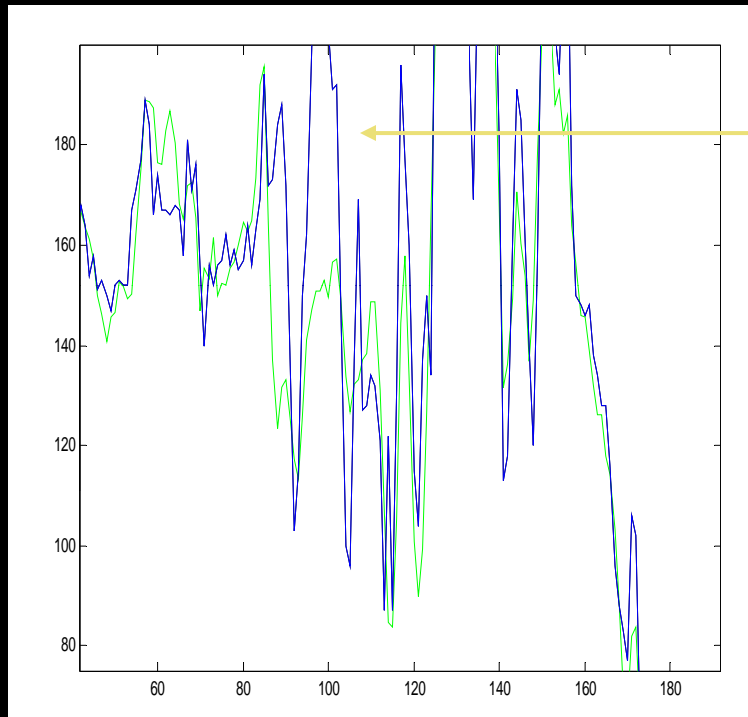


Low-pass

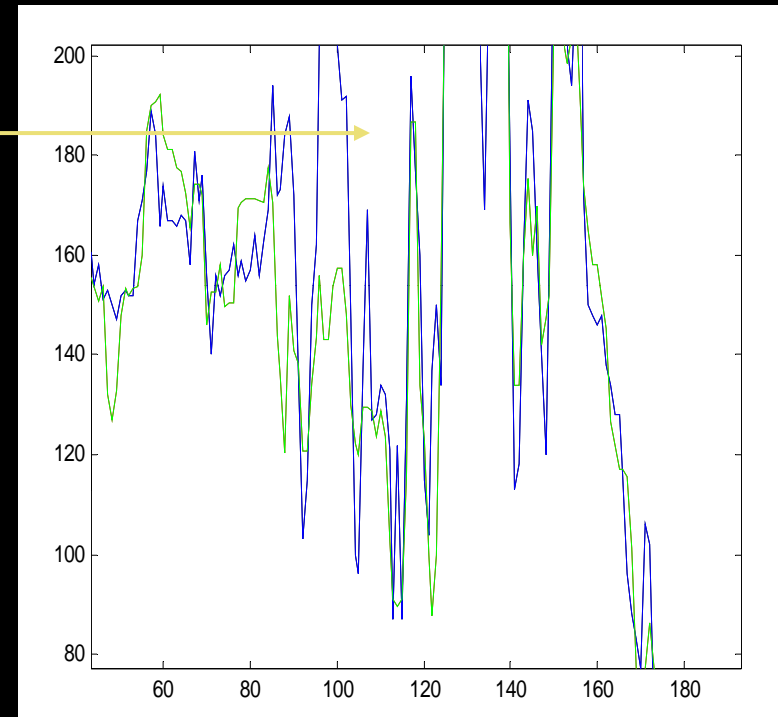


Median

Low-pass



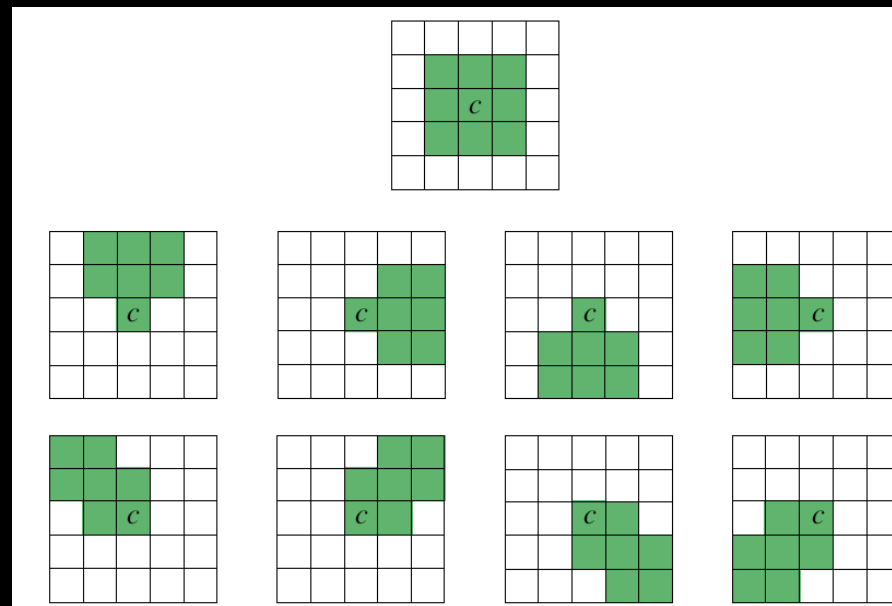
Median



- Low-pass: fine detail smoothed by averaging
- Median: fine detail passed by filter

- Smooth (average, blur) an image without disturbing
 - Sharpness or
 - positionOf the edges
- Nagao-Maysuyama Filter
- Kuwahara Filter
- Anisotropic Diffusion Filtering (Perona & Malik,
- Spatially variant smoothing

- Calculate the variance within nine subwindows of a 5x5 moving window
- Output value is the mean of the subwindow with the lowest variance
- Nine subwindows used:



■ Principle:

- divide filter mask into four regions (a, b, c, d).
- In each compute the mean brightness and the variance
- The output value of the center pixel (abcd) in the window is the mean value of that region that has the smallest variance.



a	a	ab	b	b
a	a	ab	b	b
ac	ac	abcd	bd	bd
c	c	cd	d	d
c	c	cd	d	d





Original



Median
(1 iteration)



Median
(10 iterations)



Kuwahara



Original



Perona, Malik, 1987

$$\frac{\partial u}{\partial t} = \operatorname{div} \left(\frac{Du}{|Du|^2 + \lambda^2} \right)$$



Rudin, Osher,
Fatemi, 1992

$$\frac{\partial u}{\partial t} = \operatorname{div} \left(\frac{Du}{|Du|} \right)$$



Alvarez, Lions, 1992

$$\frac{\partial u}{\partial t} = \frac{|Du|}{|k * Du|} \operatorname{div} \left(\frac{Du}{|Du|} \right)$$



Weickert, 1994

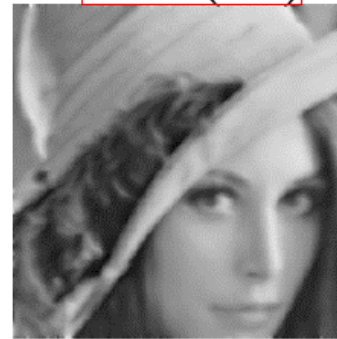
$$\frac{\partial u}{\partial t} = D^2 u(d, d)$$

$$d = SEigen(k * (Du \otimes Du))$$



Caselles, Sbert, 1997

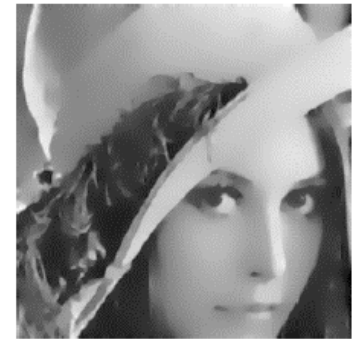
$$\frac{\partial u}{\partial t} = \frac{1}{|Du|^2} D^2 u(Du, Du)$$



Zhong Carmona, 1998

$$\frac{\partial u}{\partial t} = D^2 u(d, d)$$

$$d = SEigen(D^2 u)$$



Sochen, Kimmel,
Malladi, 1998

$$\frac{\partial u}{\partial t} = \operatorname{div} \left(\frac{Du}{\sqrt{|Du|^2 + 1}} \right)$$



- In general:
 - Anisotropic diffusion estimates an image from a noisy image
 - Useful for restoration, etc.
 - Only shown smoothing examples
 - Read the literature

- Set of general techniques
- Varied goals:
 - enhance perceptual aspects of an image for human observation
 - preprocessing to aid a vision system achieve its goal
- Techniques tend to be simple, ad-hoc, and qualitative
- Not universally applicable
 - results depend on image characteristics
 - determined by interactive experimentation
- Can be embedded in larger vision system
 - selection must be done carefully
 - some techniques introduce artifacts into the image data
- Developing specialized techniques for specific applications can be tedious and frustrating.

Summary

- What is noise?
 - Gaussian distribution
- Noise reduction
 - first principles
- Neighbourhood
 - low-pass
 - median

Conclusion

- Averaging pixels corrupted by noise cancels out the noise.
- Low-pass can blur image.
- Median may retain fine image detail that may be smoothed by averaging.