

Digital Image Processing

Ming-Sui (Amy) Lee
Lecture 02

Announcement

■ Class Information

- Class website
 - NTU COOL
 - Syllabus / Lecture #1
 - [Lecture #2](#)
 - [Submission guideline](#)
 - [Homework #1](#)
 - Please be sure to read the guideline carefully
 - Deadline: 00:00 on Mar. 18, 2021

Image Enhancement

[Image Enhancement]

■ Goal of Image Enhancement

- make images more appealing
- no theory, ad-hoc rules, derived with insights

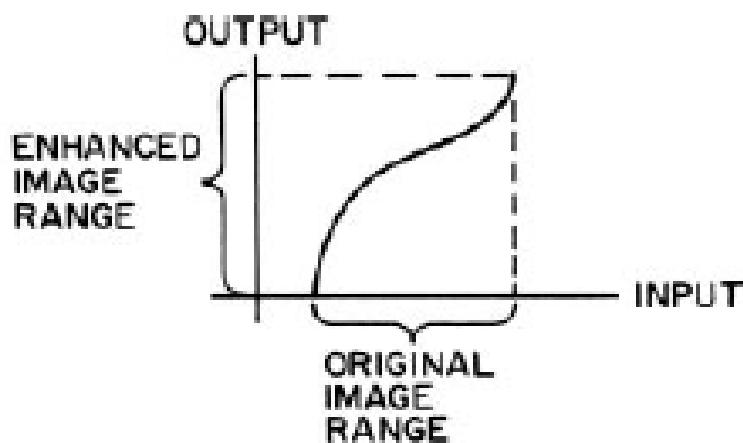
■ Two Approaches

- Contrast Manipulation
- Histogram Modification

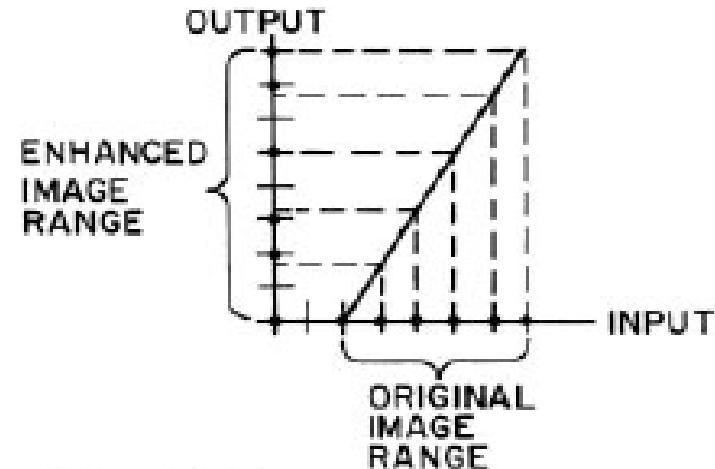
Contrast Manipulation

Transfer Function

- Linear
- Nonlinear
- Piecewise



Continuous Image

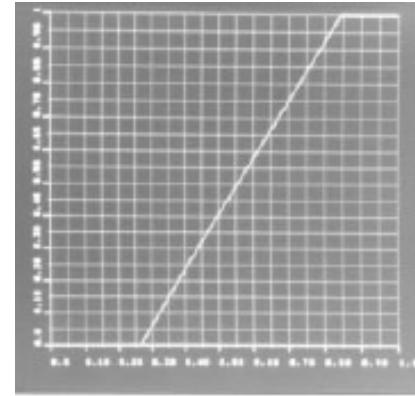
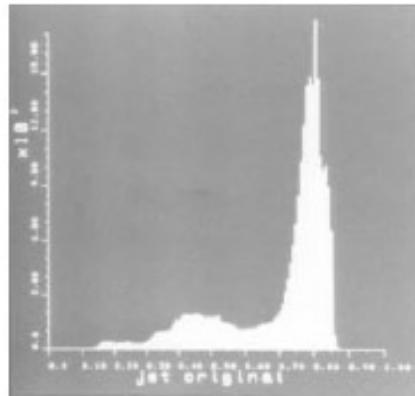


Quantized Image

Contrast Manipulation

Linear scaling and clipping

$$G(j,k) = T[F(j,k)] \quad 0 \leq F(j,k) \leq 1$$

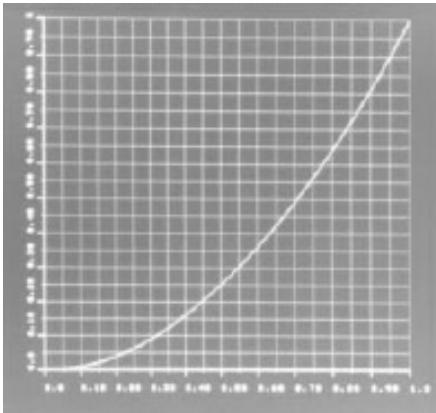


Contrast Manipulation

Power-Law



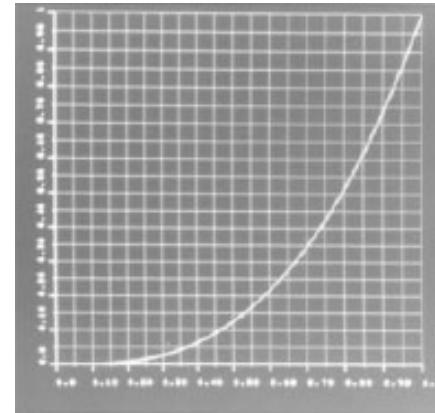
$$G(j,k) = [F(j,k)]^p \quad 0 \leq F(j,k) \leq 1$$



(a) Square function



(b) Square output



(c) Cube function



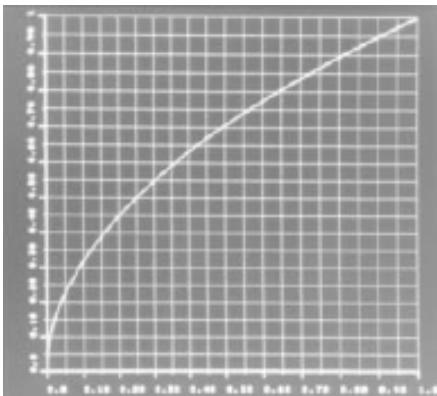
(d) Cube output 7

Contrast Manipulation

■ Power-Law



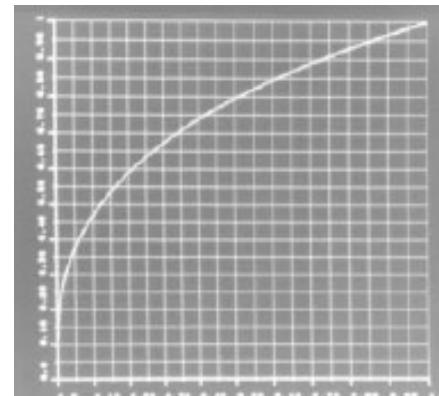
$$G(j,k) = [F(j,k)]^p \quad 0 \leq F(j,k) \leq 1$$



(a) Square root function



(b) Square root output



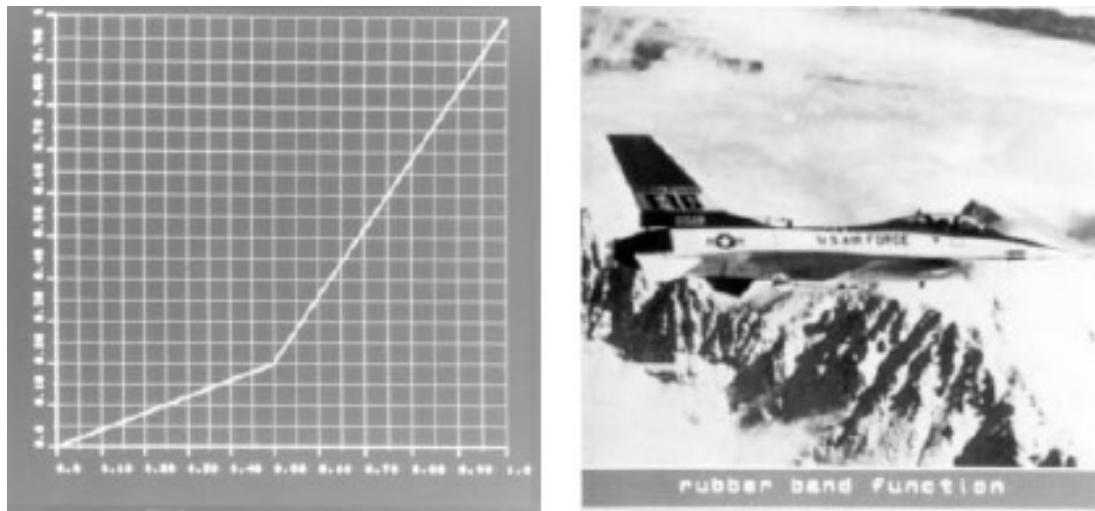
(c) Cube root function



(d) Cube root output

Contrast Manipulation

- Rubber Band Transfer Function
 - Piecewise linear transformation
 - Inflection point (control point)



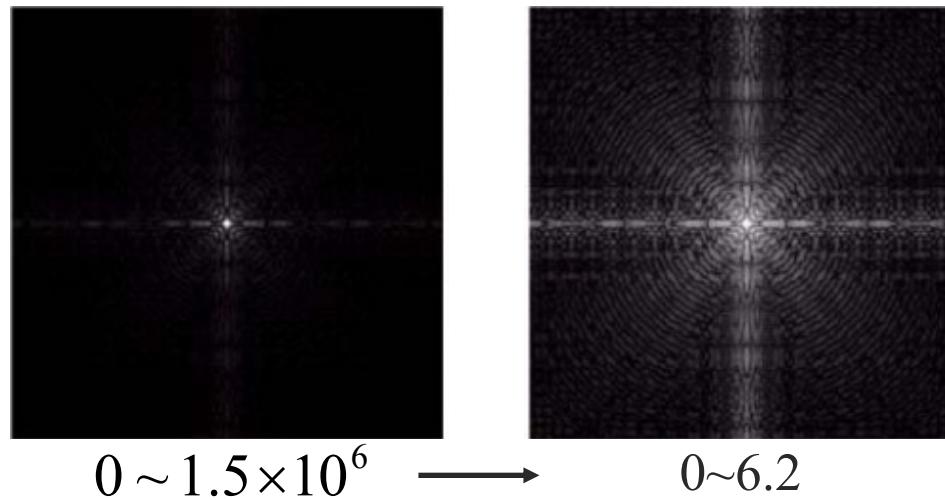
Can choose the area where we want to stretch or reduce the contrast

[Contrast Manipulation]

■ Logarithmic Point Transformation

$$G(j, k) = \frac{\log_e \{1 + aF(j, k)\}}{\log_e \{2.0\}} \quad 0 \leq F(j, k) \leq 1$$

Fourier Spectrum

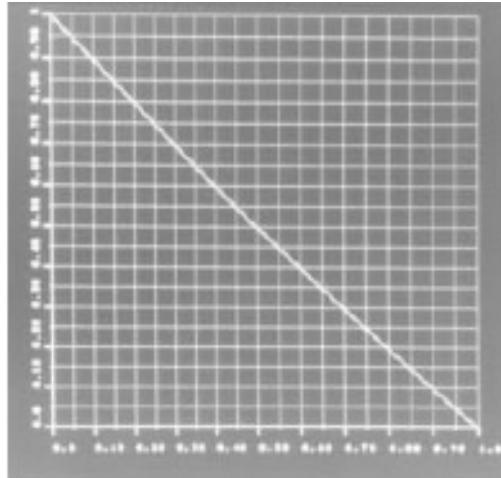


Useful for scaling image arrays with a very wide dynamic range

Contrast Manipulation

Reverse Function

$$G(j,k) = 1 - F(j,k) \quad 0 \leq F(j,k) \leq 1$$



(a) Reverse function



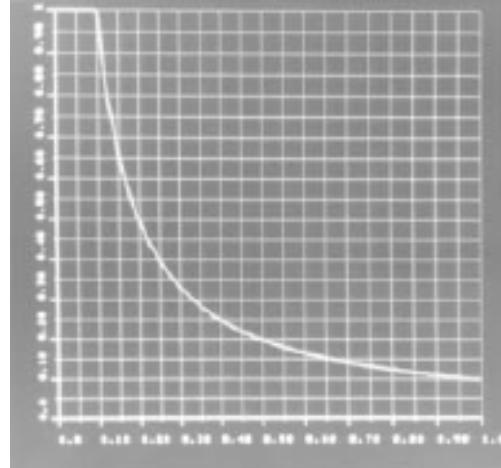
(b) Reverse function output

Able to see more details in dark areas of an image

Contrast Manipulation

Inverse Function

$$G(j,k) = \begin{cases} 1 & 0 \leq F(j,k) \leq 0.1 \\ \frac{0.1}{F(j,k)} & 0.1 \leq F(j,k) \leq 1 \end{cases}$$



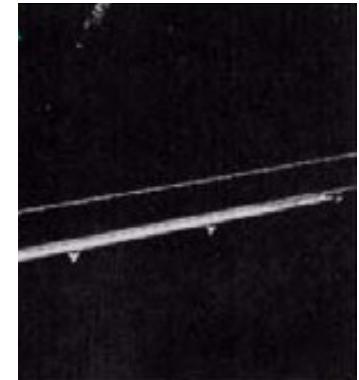
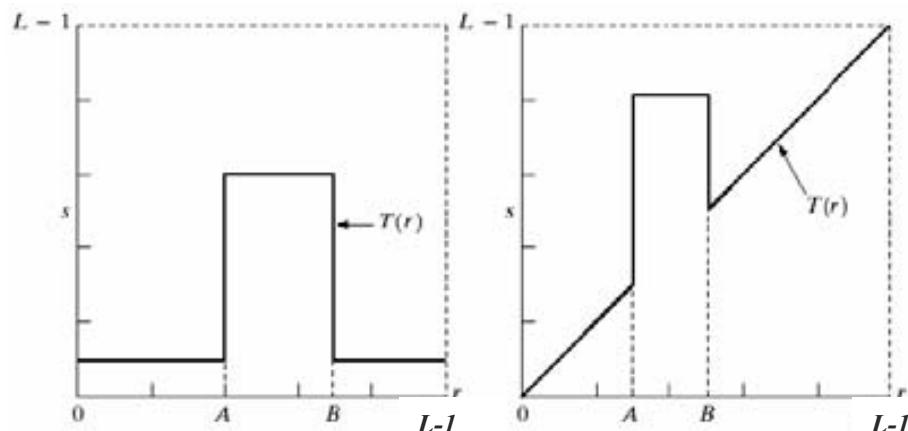
(c) Inverse function



(d) Inverse function output

Contrast Manipulation

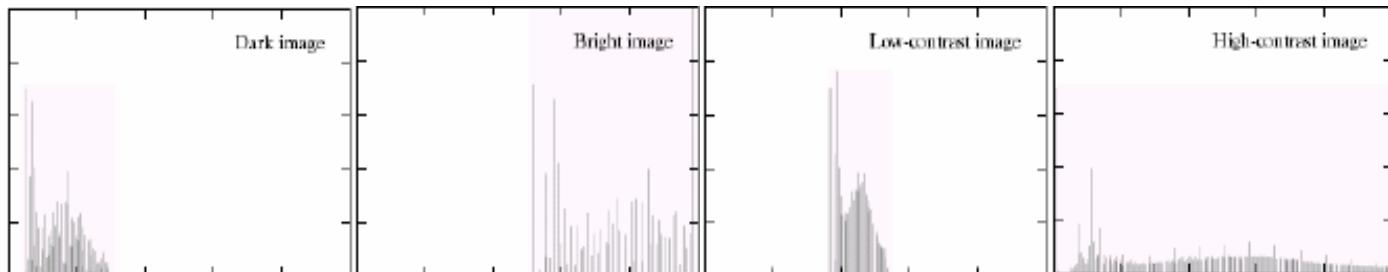
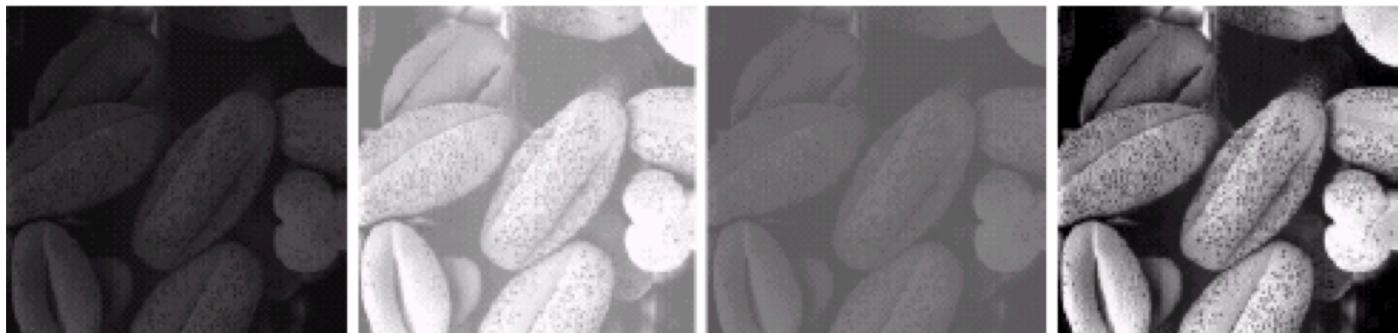
■ Amplitude-Level Slicing (Gray-Level Slicing)



Histogram Modification

Goal

- Rescale the original image so that the histogram of the enhanced image follows some desired form



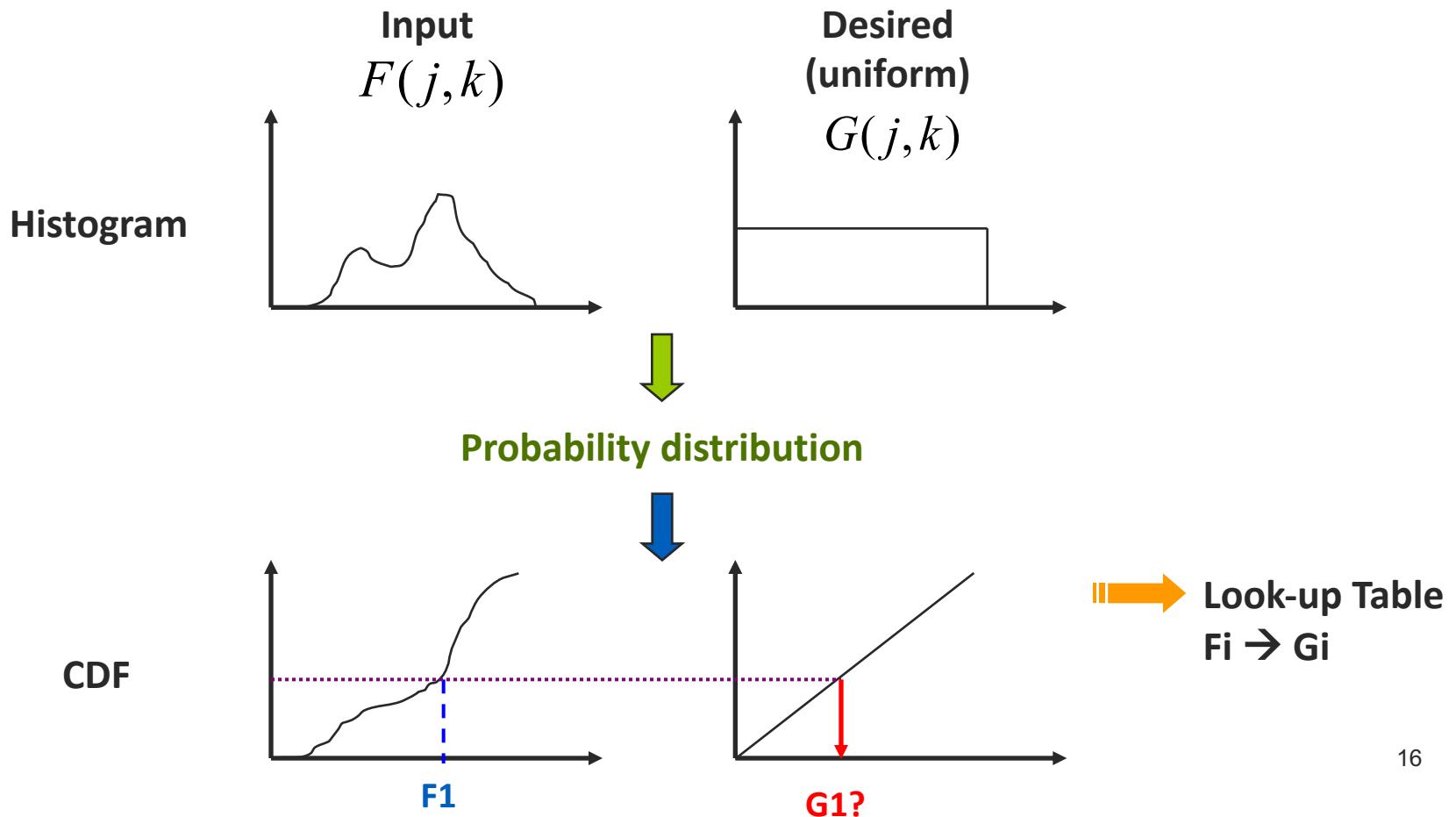
[Histogram Modification]

■ Histogram Equalization

- make the output histogram to be uniformly distributed
 - Transfer function
 - Bucket filling

Histogram Equalization

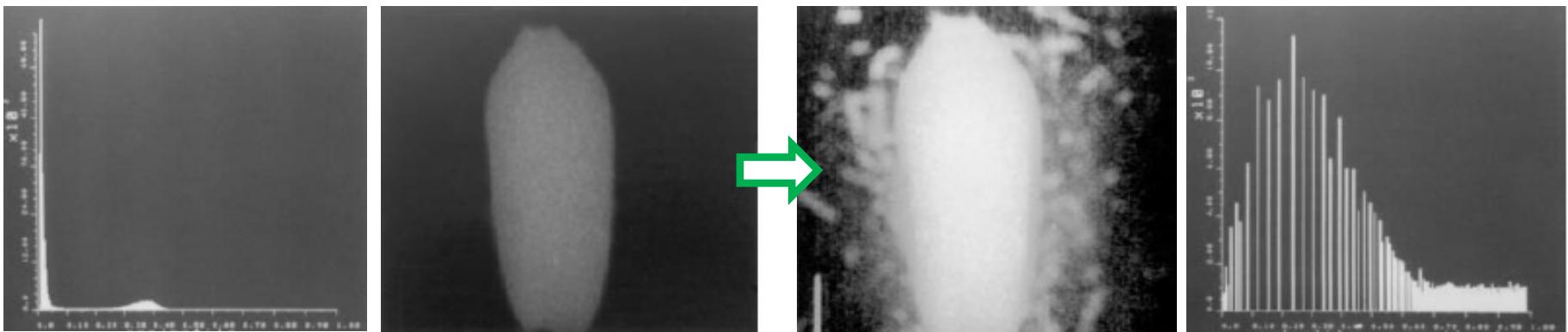
Transfer Function



Histogram Equalization

Transfer Function

- Output histogram not really uniformly distributed
- Still keep the shape
- More flat than the original histogram



Histogram Equalization

Bucket Filling

arbitrary

$F(j,k)$	# of pixels
0	1
1	2
2	5
:	:
255	3

uniform

$G(j,k)$	# of pixels
0	$N/256$
1	$N/256$
2	$N/256$
:	:
255	$N/256$

N : # of total pixels

- Not 1-1 mapping
- Accumulated probability may not end exactly at the boundary of a bin → split it out

[

Reference

]

- Gamma correction
- Tone mapping
- HDR – High Dynamic Range Imaging

Noise Cleaning

Noise Cleaning

■ Noise

- electrical sensor noise
- photographic grain noise
- channel error
- etc.

■ Characteristics of the noise

- discrete
- not spatially correlated
- higher spatial frequency



Noise Cleaning

- Two types of noise
 - Uniform Noise
 - Additive uniform noise, Gaussian noise
 - Impulse Noise
 - Salt and pepper noise
- Solutions
 - Uniform Noise → low-pass filtering
 - Impulse Noise → non-linear filtering

[Basics of Spatial Filtering]

■ Mask

- filter, kernel, template
- $m \times n$
 - $m=2a+1, n=2b+1$,
where a and b are nonnegative integers
 - e.g. 3x3 mask

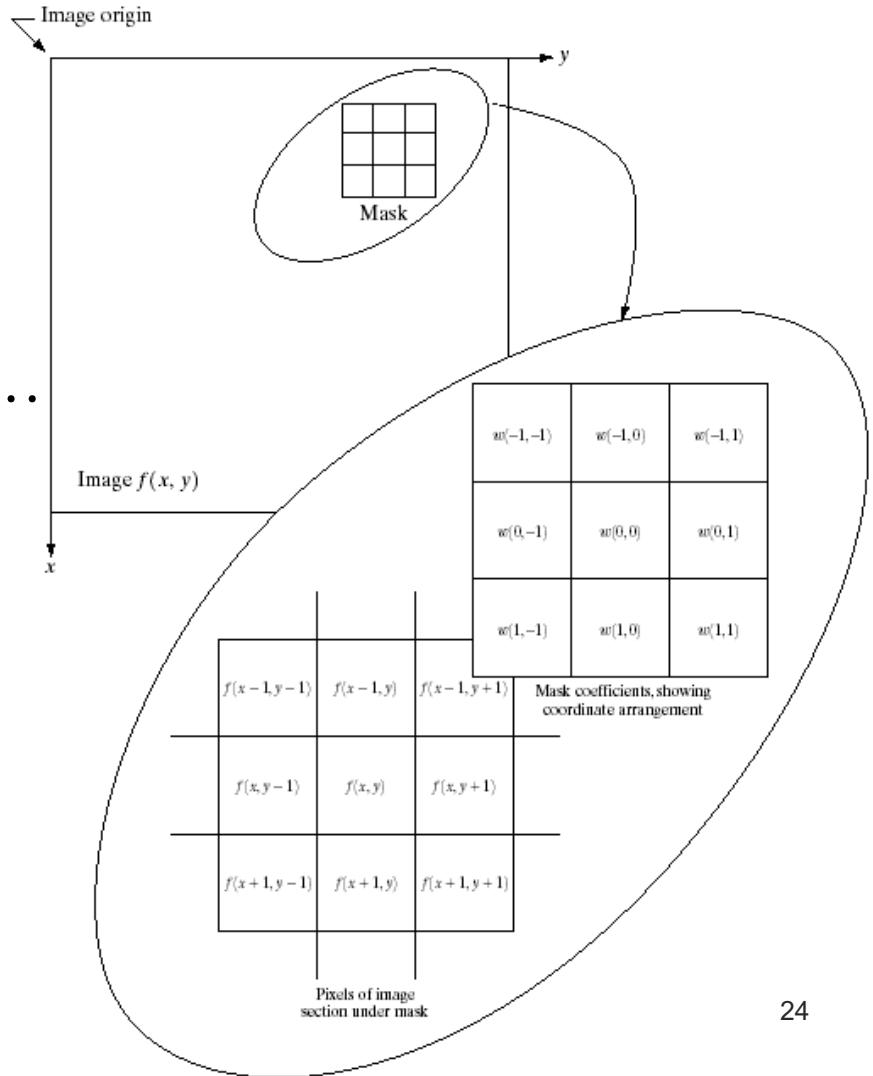
$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$W(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

■ Spatial Filtering/Convolution

$$\begin{aligned} G(j,k) = & w(-1,-1)F(j-1,k-1) + w(-1,0)F(j-1,k) + \dots \\ & + w(0,0)F(j,k) + \dots \\ & + w(1,0)F(j+1,k) + w(1,1)F(j+1,k+1) \end{aligned}$$

Basics of Spatial Filtering

$$\begin{aligned}
 G(j,k) = & w(-1,-1)F(j-1, k-1) \\
 & + w(-1,0)F(j-1, k) + \dots \\
 & + w(0,0)F(j, k) + \dots \\
 & + w(1,0)F(j+1, k) \\
 & + w(1,1)F(j+1, k+1)
 \end{aligned}$$

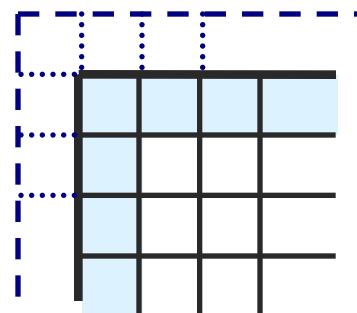
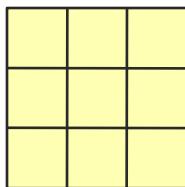


Q: Boundary pixels?

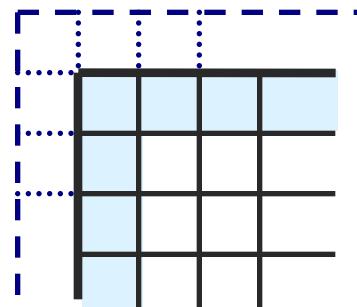
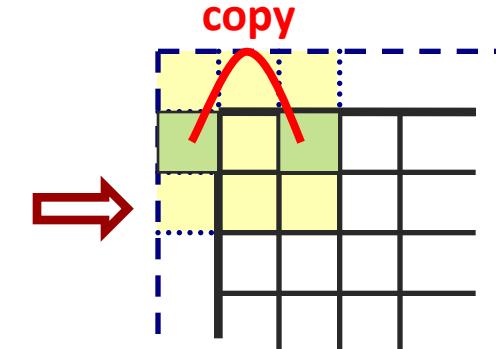
Basics of Spatial Filtering

■ Boundary Extension (3x3 mask)

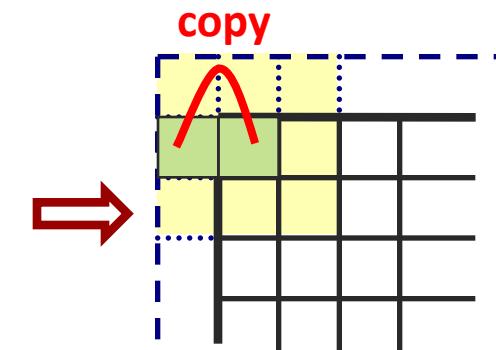
e.g.
3x3 mask, w



odd



even



Q: 5x5 mask?

Noise Cleaning

■ Uniform noise

- Perform low-pass filtering

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- General form

$$H = \frac{1}{(b+2)^2} \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}$$

e.g.

$$F = \begin{bmatrix} 0 & 0 & 180 & 180 \\ 0 & 0 & 180 & 180 \\ 0 & 0 & 180 & 180 \\ 0 & 0 & 180 & 180 \end{bmatrix}$$

High Frequency Noise Removal

Low-pass filtering

- Normalized to unit weighting
- Averaging
- Smaller/Larger filter size ?



3x3



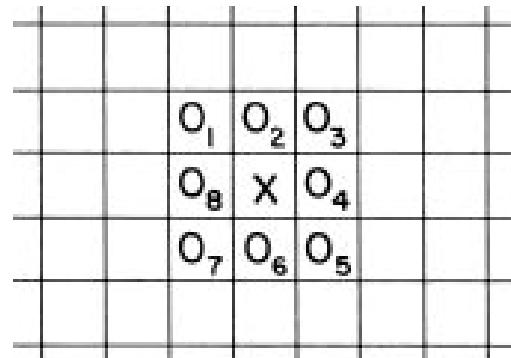
7x7

Noise Cleaning

- Impulse noise
 - black: pixel value =0 → dead sensor
 - white: pixel value=255 → saturated sensor
- Solutions
 - Outlier detection
 - Median filtering
 - Pseudo-median filtering (PMED)

[Impulse Noise Removal]

■ Outlier detection



$$\text{if } \left| x - \frac{1}{8} \sum_{i=1}^8 O_i \right| > \varepsilon \quad \text{then } x = \frac{1}{8} \sum_{i=1}^8 O_i$$

How to choose ε ?
Larger window?

[Impulse Noise Removal]

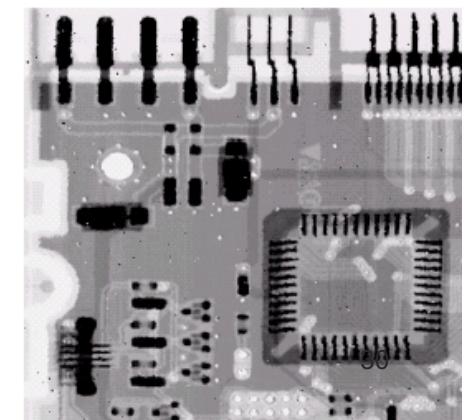
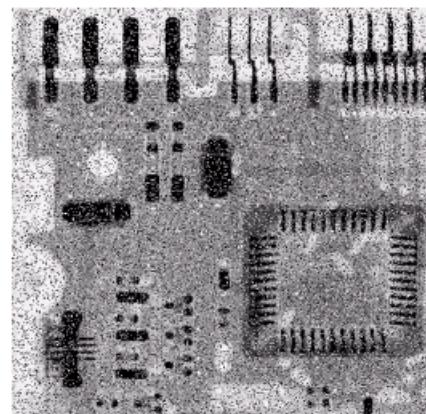
■ Median filtering

a_1, \dots, a_N where N is odd

- sort those values in order
- pick the middle one in the sorted list
- e.g. 3x3 mask:

$$I = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 8 & 7 \\ 1 & 5 & 6 \end{bmatrix}$$

→ Median is 3



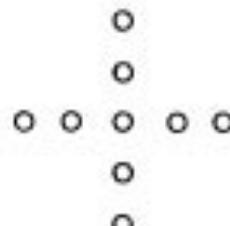
[Impulse Noise Removal]

■ Median filtering

- Preserve sharp edges
- Effective in removing impulse noise
- 1D/2D (directional)
 - e.g. 2D



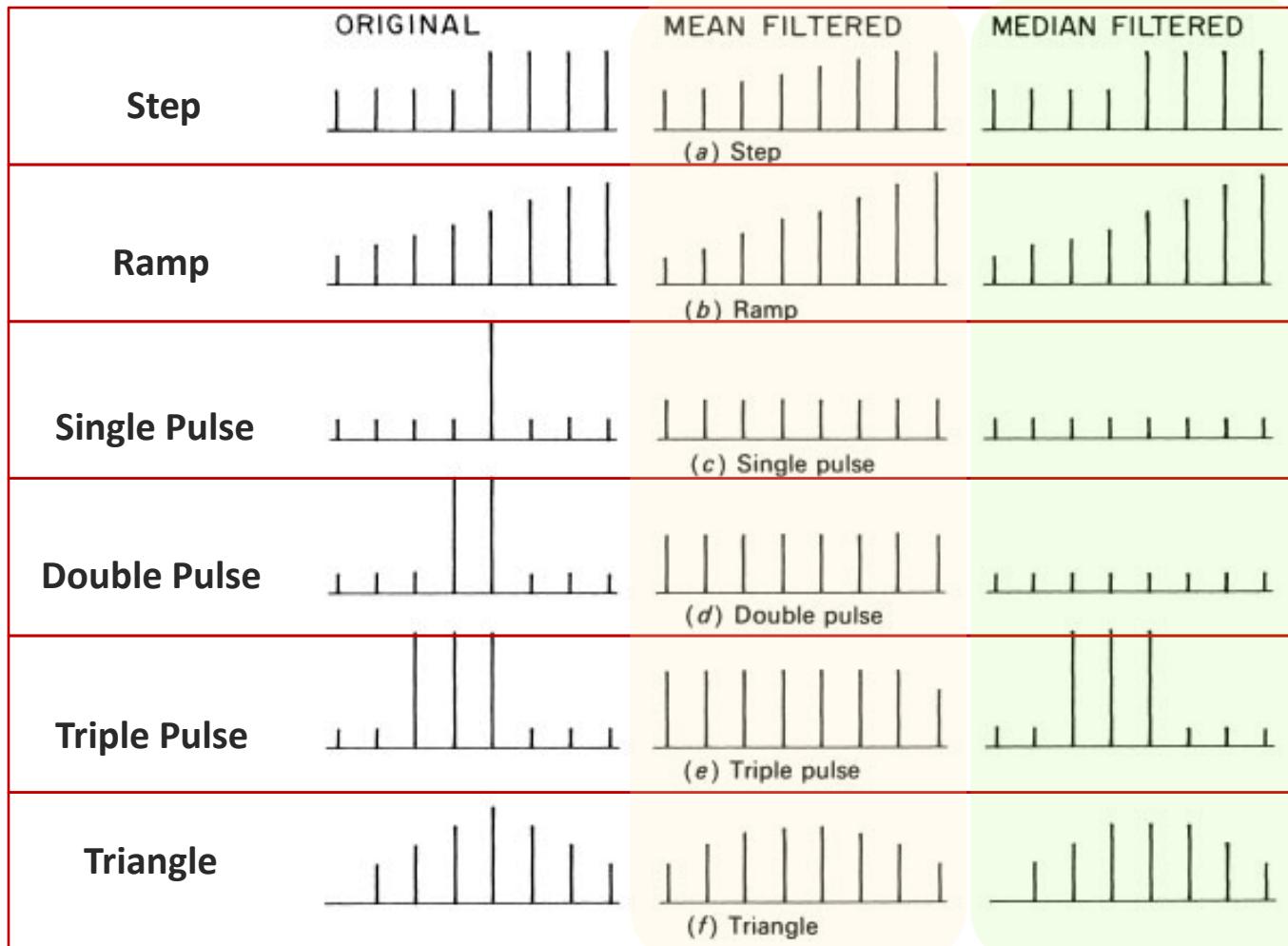
square



cross

Impulse Noise Removal

- e.g. 1D (window size = 5)



[Impulse Noise Removal]

- Median filtering
 - Fast computation
 - Approximation of median
 - e.g. 5-element filter
 - a, b, c, d, e
 - $\text{MED}(a, b, c, d, e)$
 - = $\max(\min(a,b,c), \min(a,b,d), \dots)$
 - = $\min(\max(a,b,c), \max(a,b,d), \dots)$
 - there are 10 possible choices
 - could be narrowed down

[Impulse Noise Removal]

■ Pseudomedian filtering (PMED)

- e.g. 5-element filter

$a, b, c, d, e \rightarrow$ spatially ordered

MAXMIN = A (under estimated)

$$= \max(\min(a,b,c) , \min(b,c,d) , \min(c,d,e))$$

MINMAX = B (over estimated)

$$= \min(\max(a,b,c) , \max(b,c,d) , \max(c,d,e))$$

$\rightarrow \underline{\text{PMED}(a, b, c, d, e)}$

$$= 0.5 * (A + B) = \underline{0.5 * (\text{MAXMIN} + \text{MINMAX})}$$

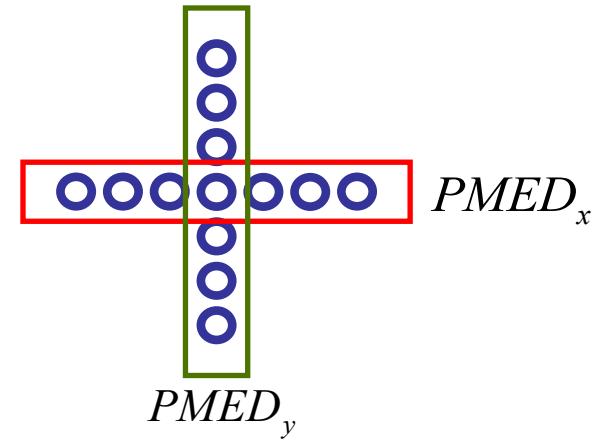
$$\sim \text{MED}(a, b, c, d, e)$$

[Impulse Noise Removal]

■ Pseudomedian filtering (PMED)

○ 2D case

$$PMED = \frac{1}{2} (PMED_x + PMED_y)$$



$$\begin{aligned} PMED &= \frac{1}{2} \max(MAXMIN(x_c), MAXMIN(y_R)) \\ &+ \frac{1}{2} \min(MINMAX(x_c), MINMAX(y_R)) \end{aligned}$$

[Impulse Noise Removal]

- Pseudomedian filtering (PMED)
 - MAXMIN
 - Remove salt noise
 - MINMAX
 - Remove pepper noise
 - May cascade two operations
 - Remove salt and pepper noise

[Impulse Noise Removal]



Original noisy image



MAXMIN



MINMAX of MAXMIN



MINMAX



MAXMIN of MINMAX

Q: same results?

Quality Measurement

■ Peak signal-to-noise ratio (PSNR)

- Mean squared error (MSE)

$$MSE = \frac{1}{w^* h} \sum_j \sum_k [F(j, k) - F'(j, k)]^2$$

- The PSNR is defined as

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right)$$

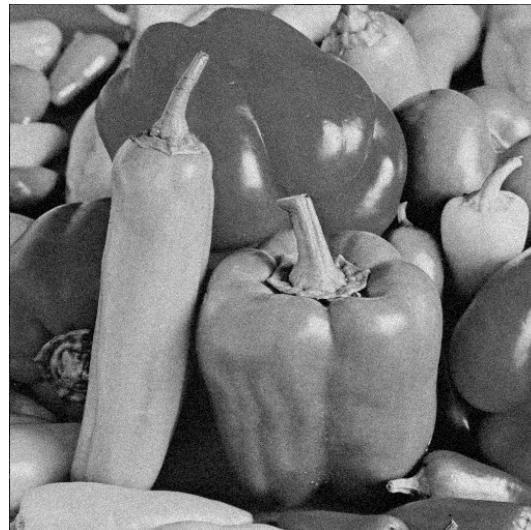
[

Example

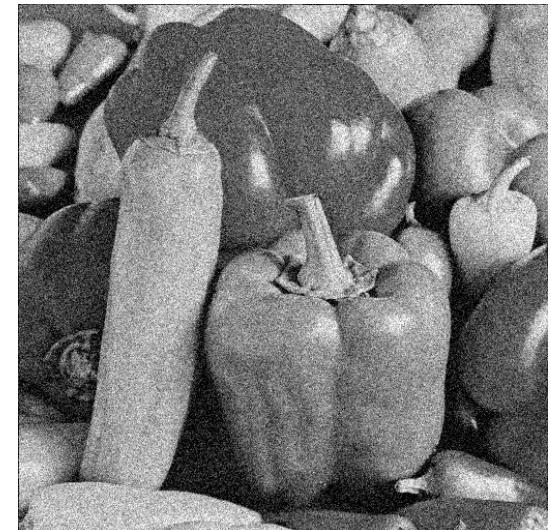
]



Original image



Gaussian noise ($\sigma=10$)
PSNR : 28.18dB



Gaussian noise ($\sigma=30$)
PSNR : 18.81dB

Q: Represent perceived visual quality?



Reference



■ EPLL

- Zoran, D., and Weiss, Y., “From learning models of natural image patches to whole image restoration,” in **IEEE International Conference on Computer Vision (ICCV)**, pp. 479-486, 2011.

■ BM3D

- Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., “Image denoising by sparse 3-D transform-domain collaborative filtering,” in **IEEE Transactions on image processing**, Vol. 16, No. 8, pp. 2080-2095, 2007.