Digital Image Processing

Ming-Sui (Amy) Lee

Mar. 02, 2016

Announcement

- **Class Information**
  - **Teaching Assistant**
    - 湯敬淦 @532
      Office Hours: 14:00 ~ 16:00, Tuesday
    - 簡均容 @532
      Office Hours: 10:00 ~ 12:00, Thursday
    - Email: dip.mslee@gmail.com

Announcement

- Class Information
  - Class website
    - https://ceiba.ntu.edu.tw/1042DIP
    - Syllabus
    - Lecture #1
    - Lecture #2
    - Submission guideline
    - Sample codes
    - Homework #1

Announcement

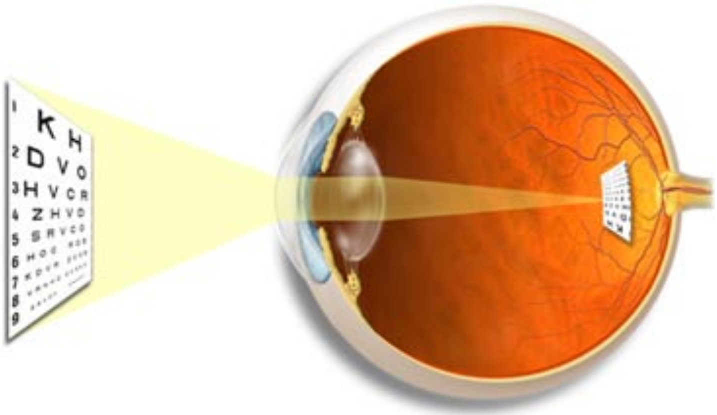
- Class Information
  - Homework
    - Please be sure to read the guideline carefully
      - Submission guideline
    - Homework #1
    - Sample codes
    - Deadline: 11:59 am on Mar. 15, 2016

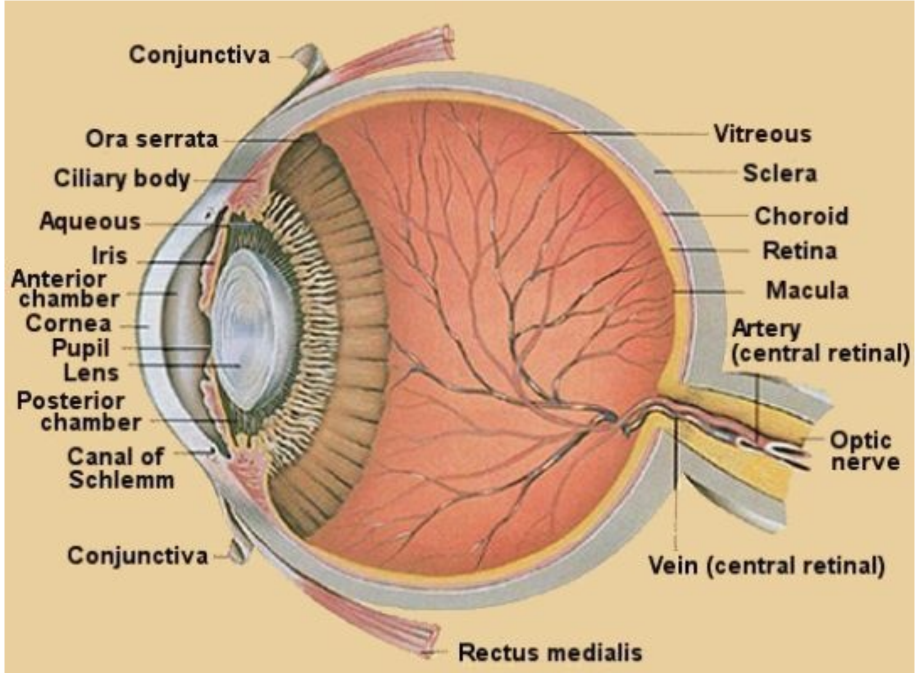
Digital Image Fundamentals

Image Quality

- **Objective / subjective**
  - Machine/human beings
  - Mathematical and Probabilistic/ human intuition and perception

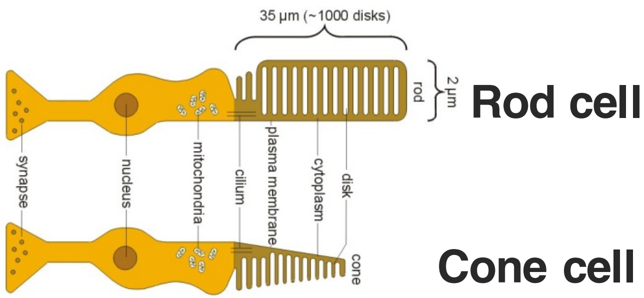
Structure of the Human Eye



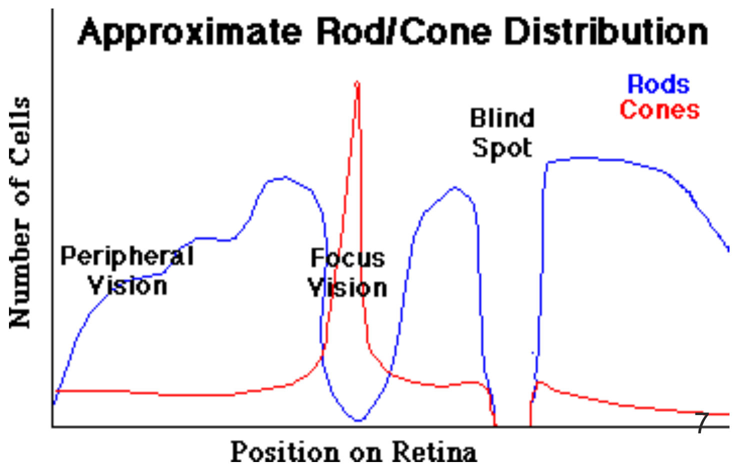


photoreceptor cells

75~150 million



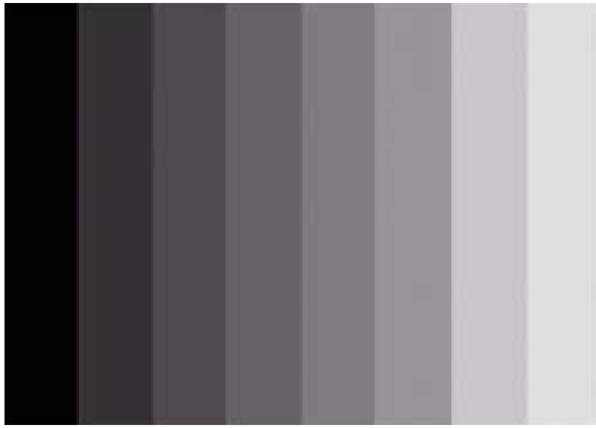
6~7 million

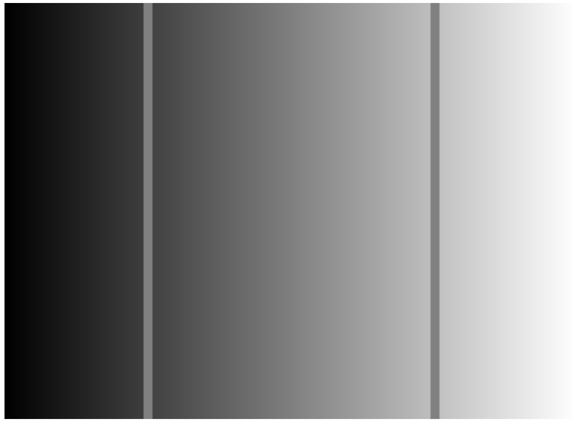


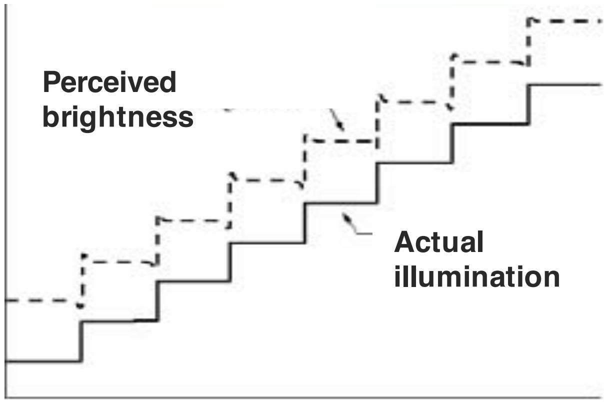
http://en.wikipedia.org/wiki/Blind_spot_(vision)

Human Visual Perception

- **Perceived brightness is NOT a simple function of intensity**

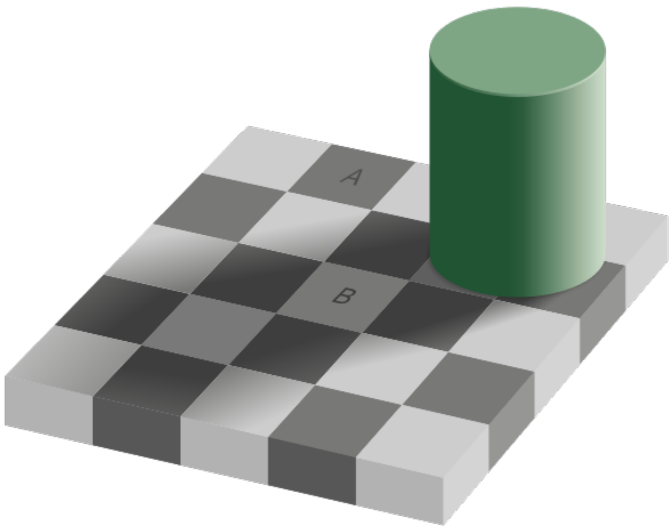


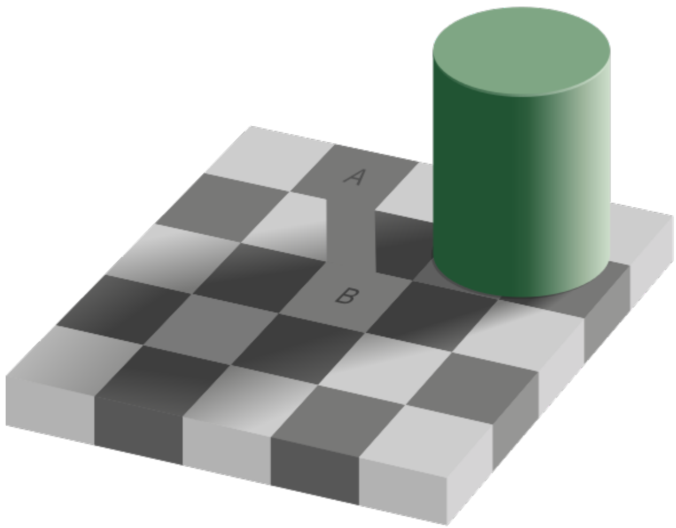


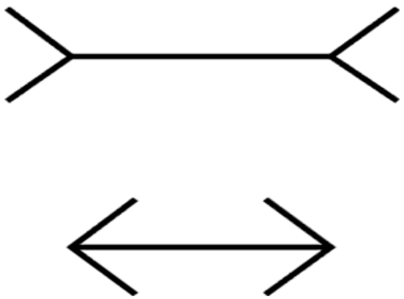


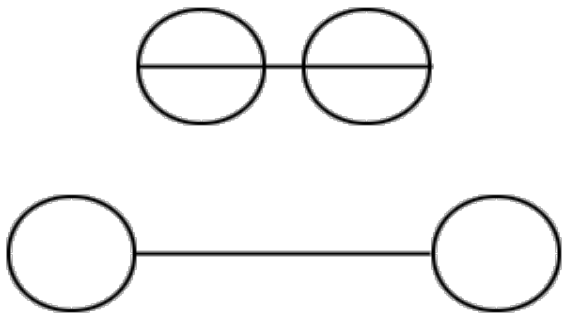
Human Visual Perception

- **Optical Illusion**



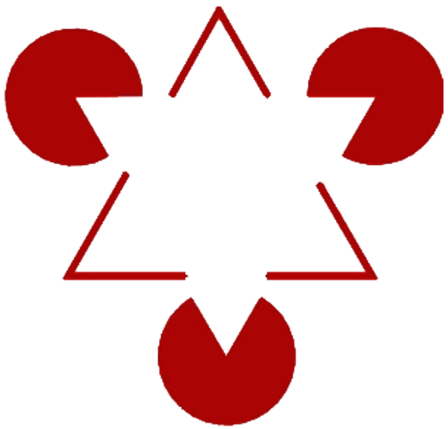


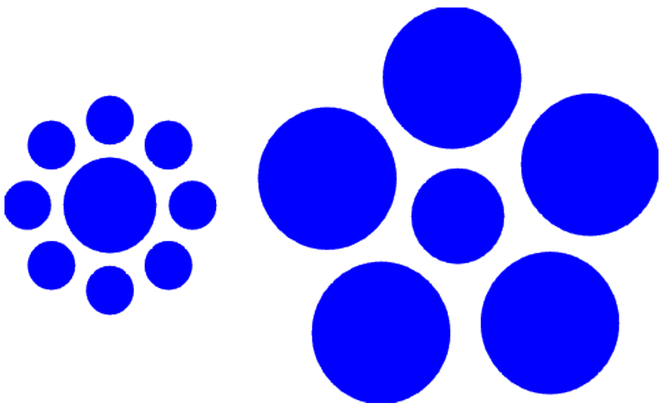


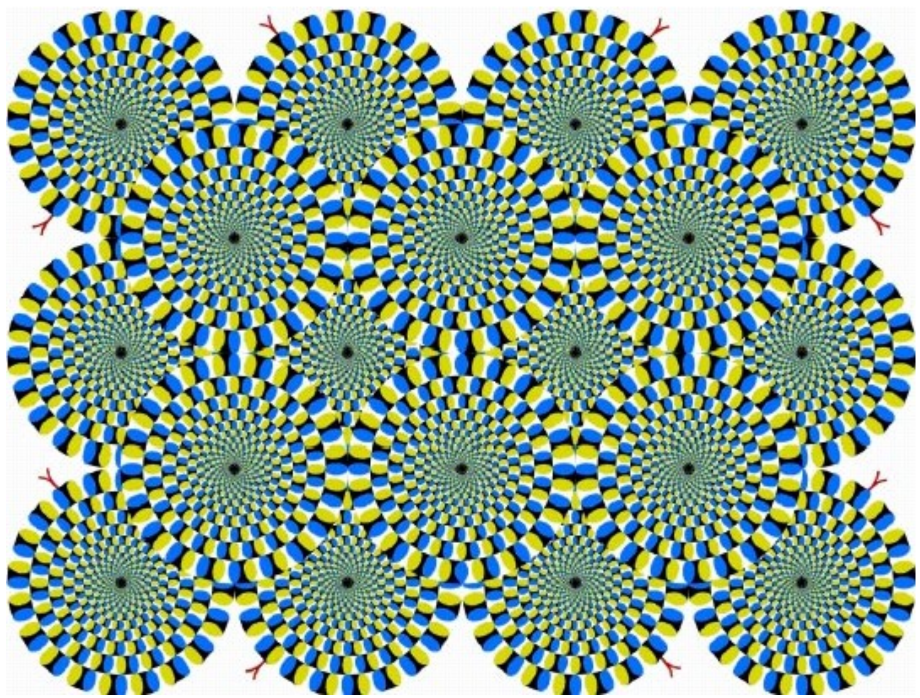


Human Visual Perception

- **Optical Illusion**



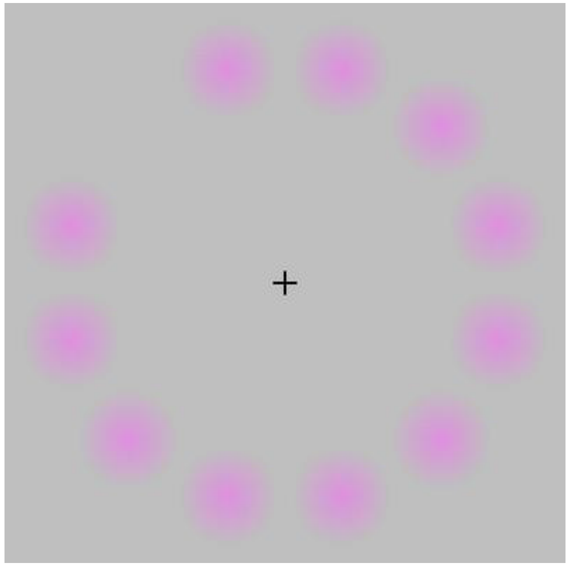


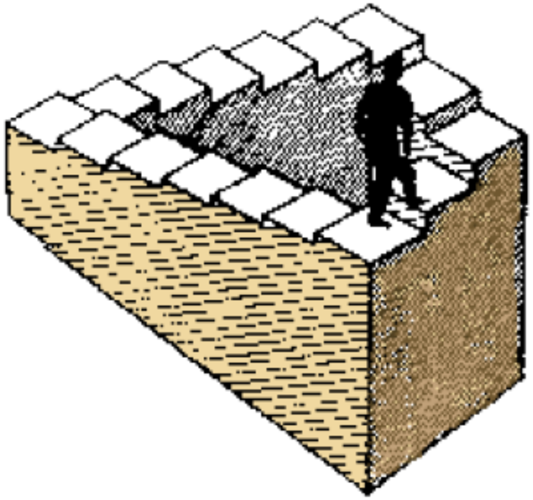


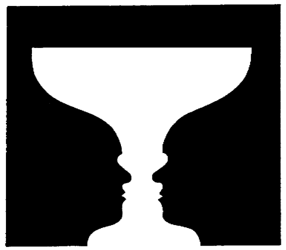
http://architecturehell.wordpress.com/

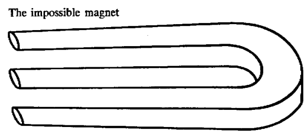
Human Visual Perception

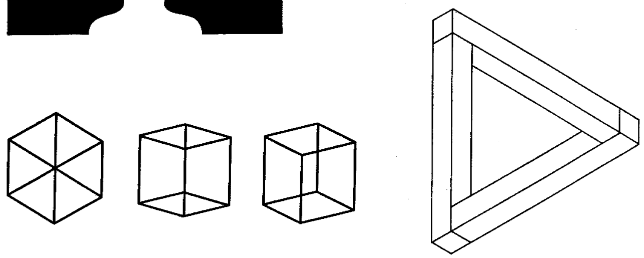
- **Optical Illusion**











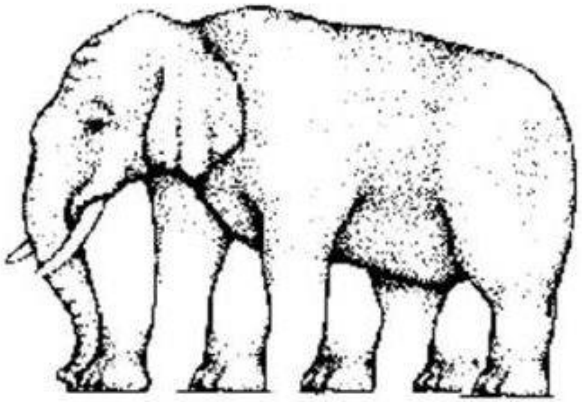
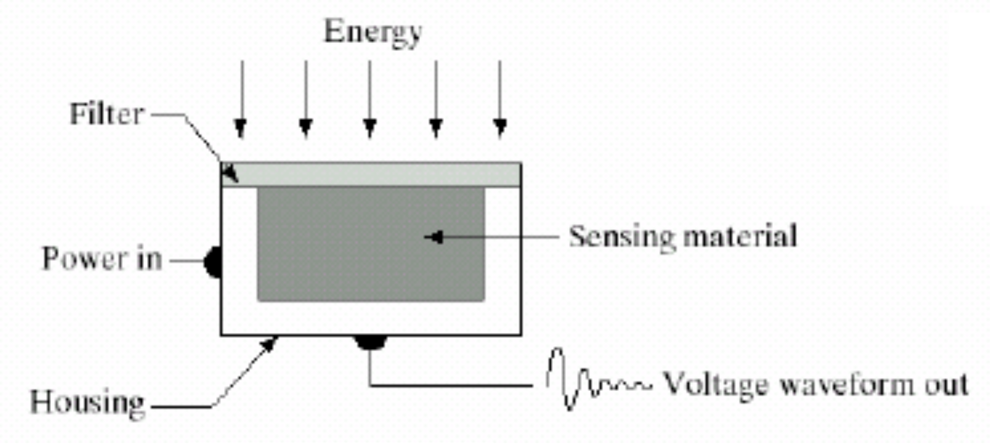


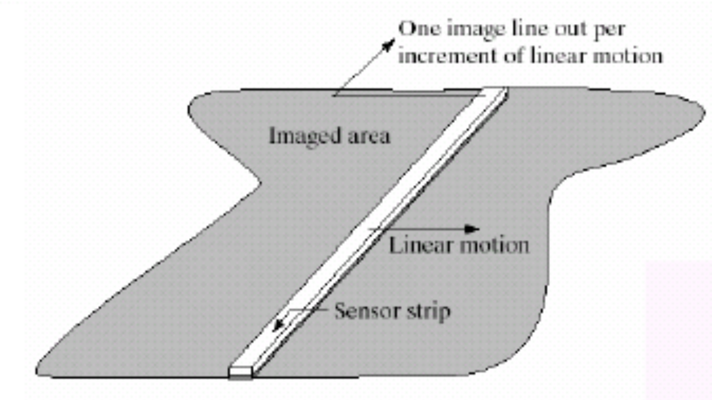
Image Sensing and Acquisition

- Illumination Source
  - EM energy, ultrasound, synthesized, …
- Scene Element
  - Objects, human organs, buried mineral,…
- Sensing Material
  - Single sensor: photodiode
  - Sensor strips: require extensive processing
  - Sensor arrays: CCD & CMOS

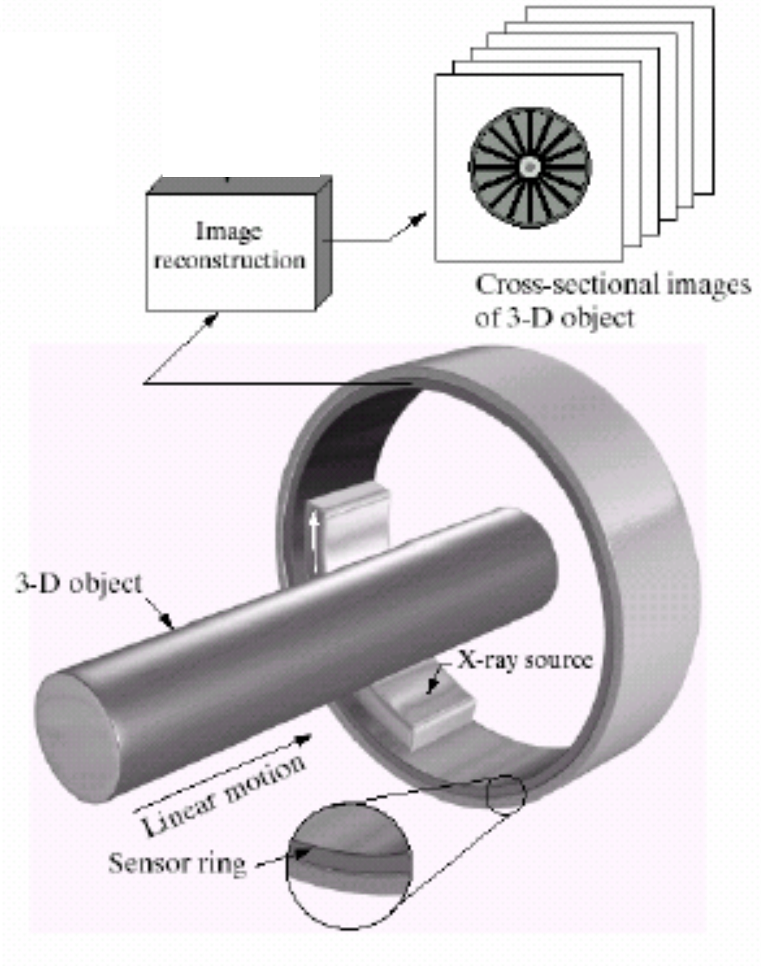
Image Sensing and Acquisition



Single sensor



Sensor Strip



Circular Sensor Strip

Image Sensing and Acquisition

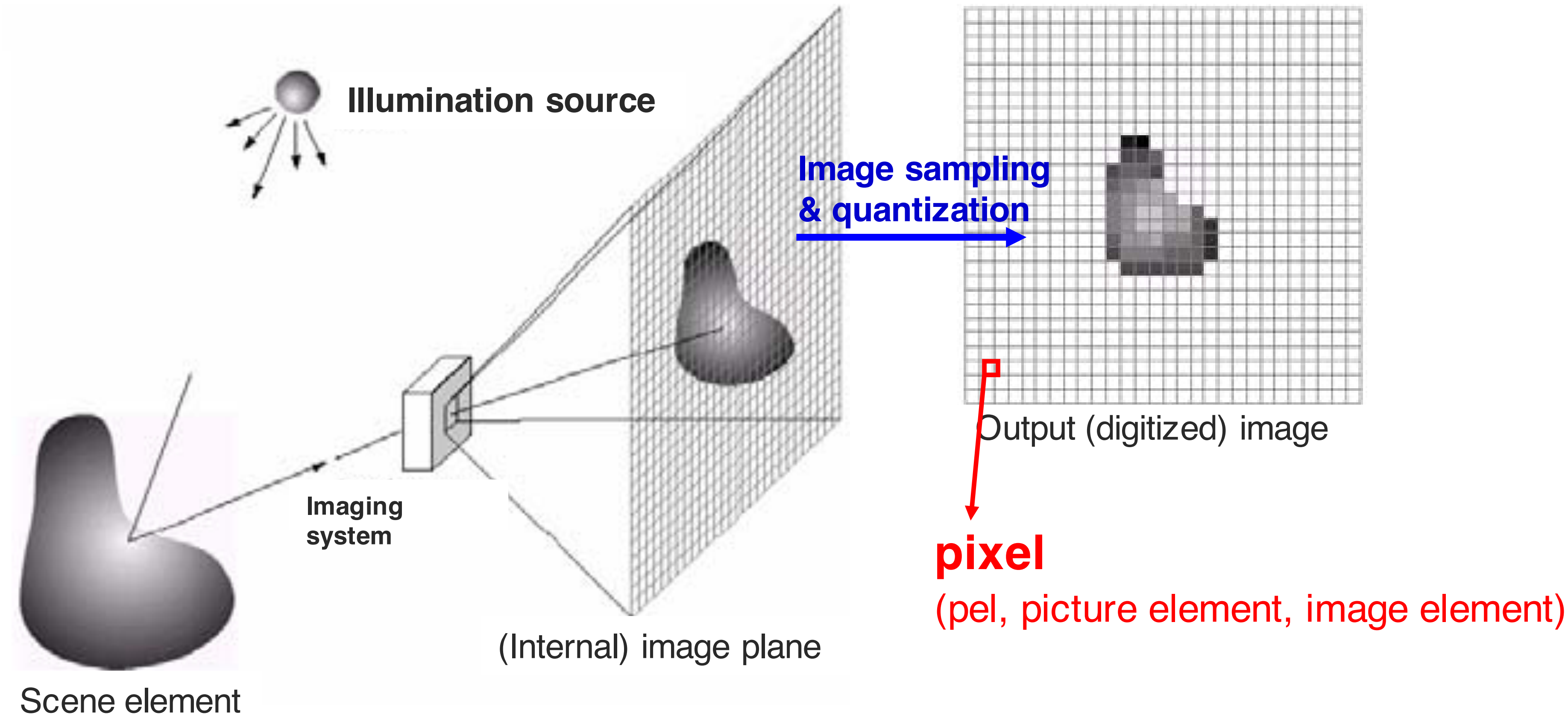
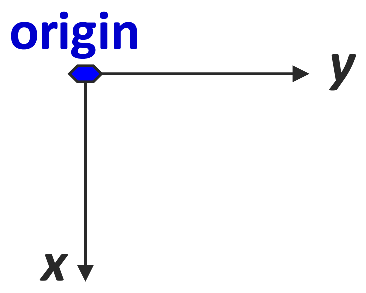


Image Formation Model

- **An image → 2D function**

$$0 < f(x, y) < \infty$$

where *x* and *y* are spatial coordinates



- **Categorized by two components**

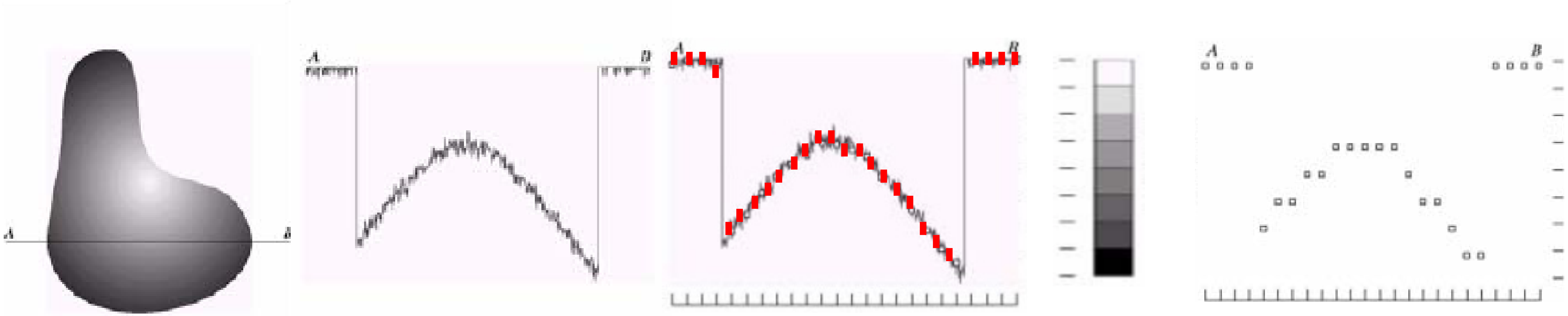
$$f(x, y) = i(x, y)r(x, y)$$

  - **Illumination:** $0 < i(x, y) < \infty$
  - **Reflectance:** $0 < r(x, y) < 1$
    - black velvet/ flat-white wall paint/ snow/ silver-plated metal
      0.1 0.8 0.93 0.9

Image Sampling & Quantization

$$f(x,y) \rightarrow \boxed{\text{A/D}} \rightarrow F(j,k)$$

sampling **quantization**

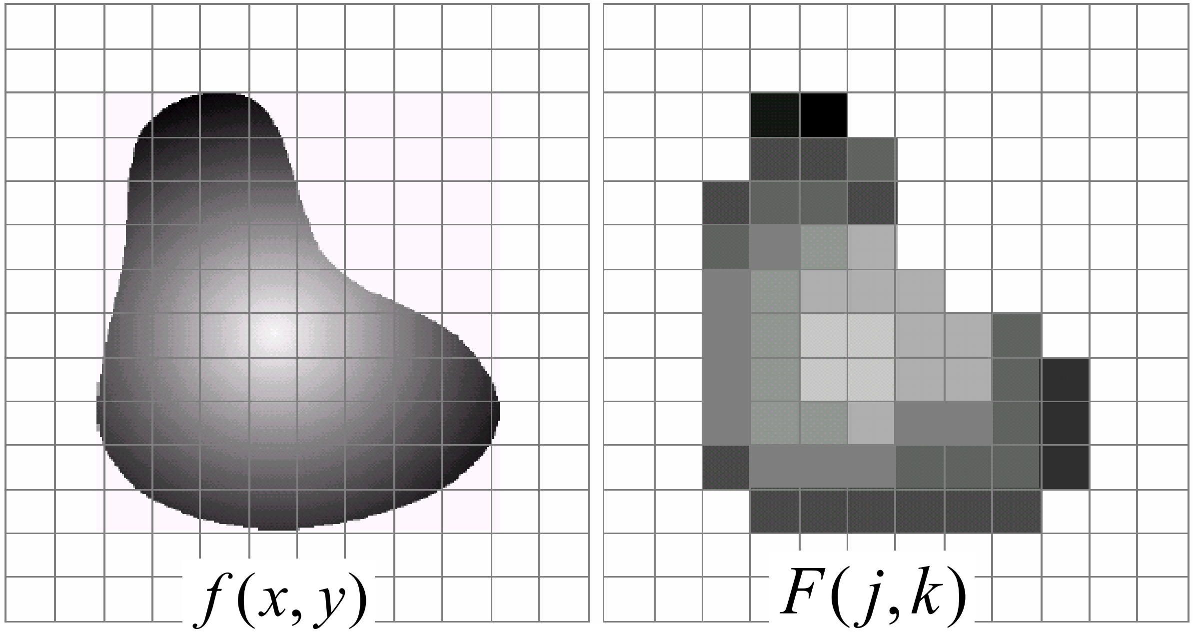


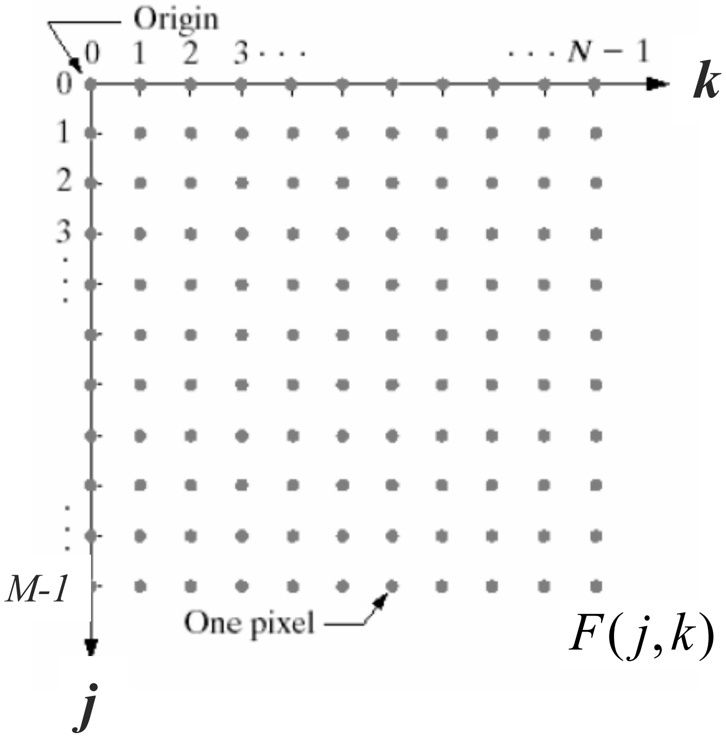
Original image $f(x,y)$ sampling quantization $F(j,k)$

sampling $\downarrow$ coordinates

quantization $\downarrow$ amplitude

Image Sampling & Quantization





$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \qquad F(j,k) = \begin{bmatrix} F(0,0) & F(0,1) & \cdots & F(0,N-1) \\ F(1,0) & F(1,1) & \cdots & F(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ F(M-1,0) & F(M-1,1) & \cdots & F(M-1,N-1) \end{bmatrix}$$

Digital Image Representation

- **Dynamic Range**
  - **The range of values spanned by the gray scale**

    $\{0, 1, \ldots, L-1\}$ $\quad L = 2^k$

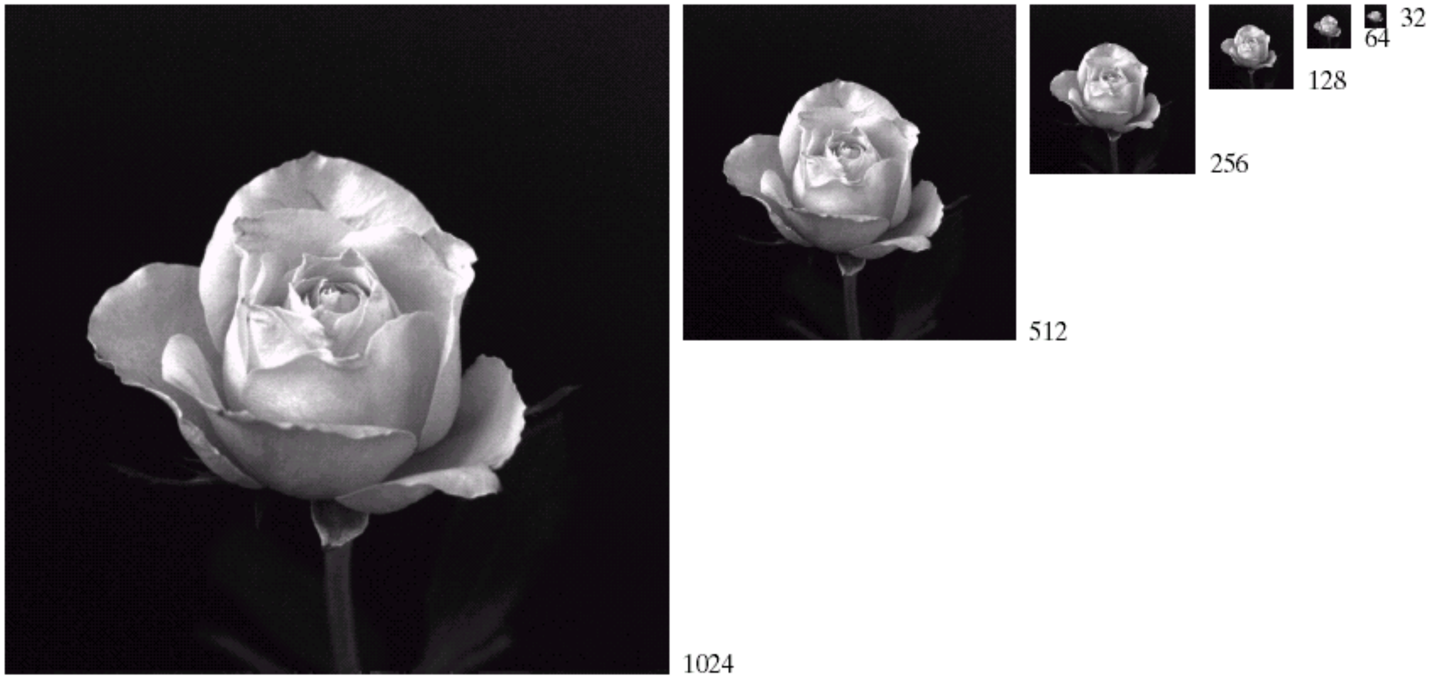
- **Image Size**
  - **for a square image,** $M = N$

    total number of bits required to store the image: $b = N^2 \cdot k$

N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

Downsampling

- **1024x1024 → 32x32**
  - **Downsampled by a factor of 2**



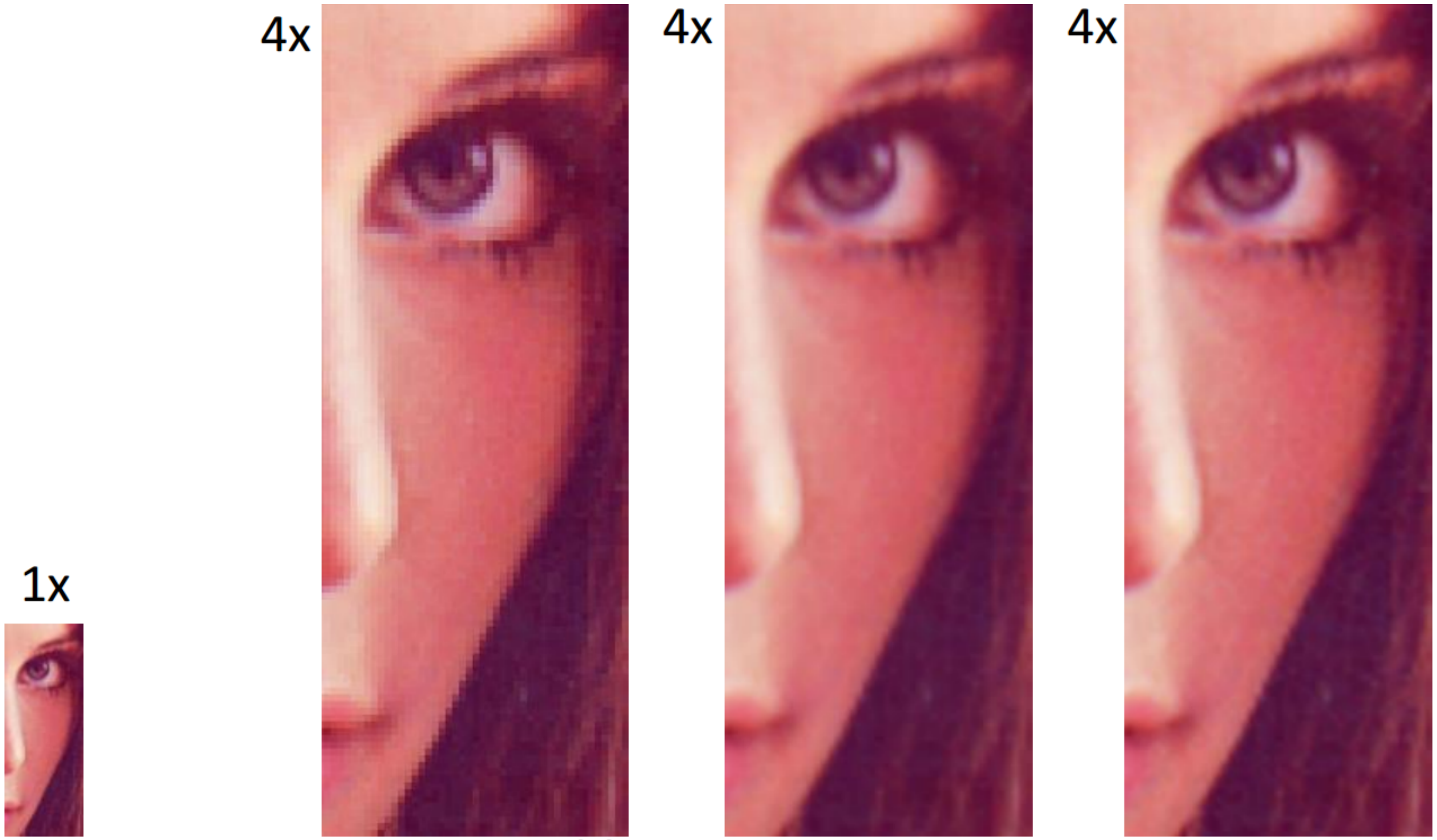
Re-Sampling

- Zero-Order-Hold Method (ZOH)
  - Row and column duplication

Re-Sampling

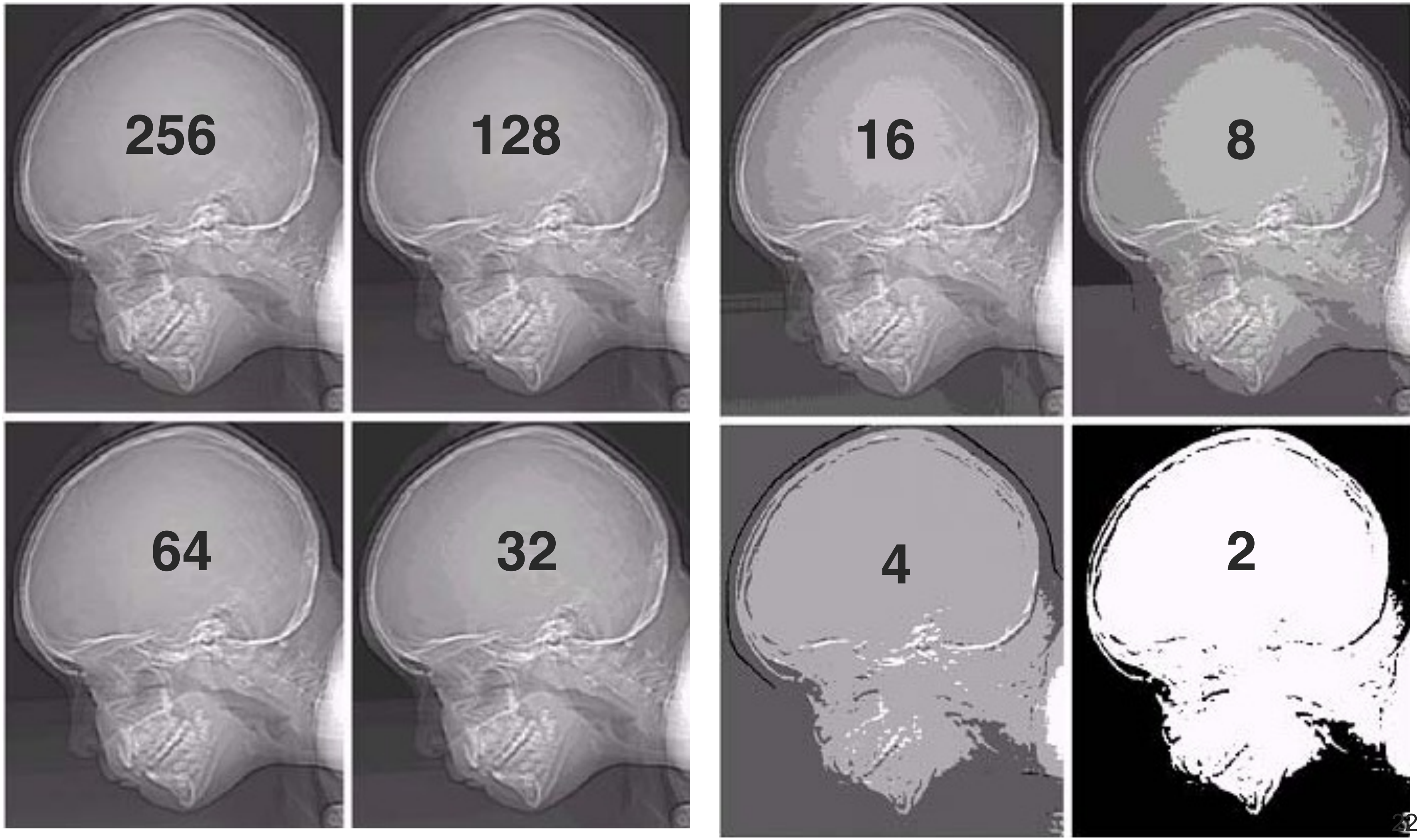
4x 4x 4x

1x



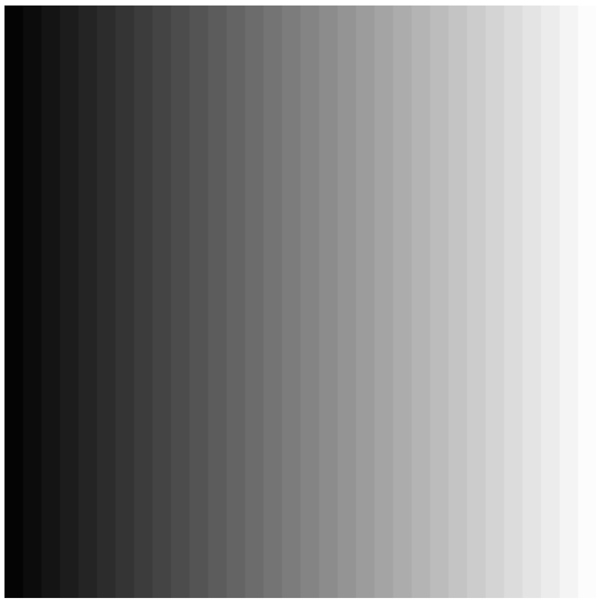
Original Nearest neighbor bilinear bicubic

L=256,128,64,32,16,8,4,2



Digital Image Representation

- 8-bit image is commonly used
  - Storage
  - Human perception



32 steps (5 bits) in gray level

64 steps (6 bits) in gray level

Image Enhancement

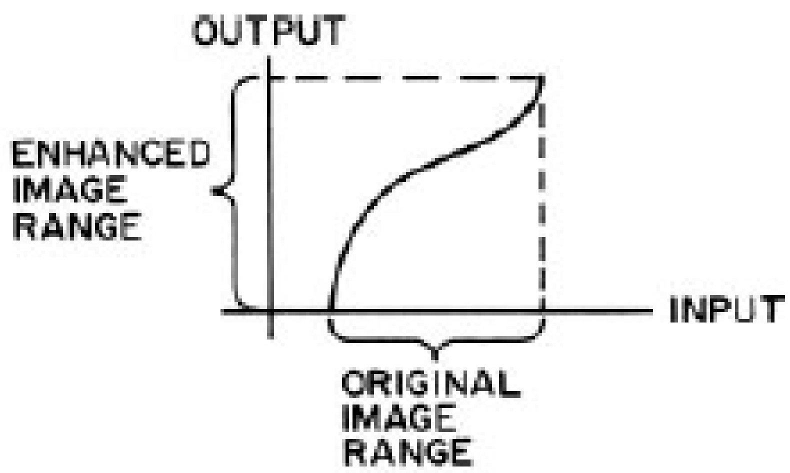
Image Enhancement

- **Goal of Image Enhancement**
  - make images more appealing
  - no theory, ad-hoc rules, derived with insights

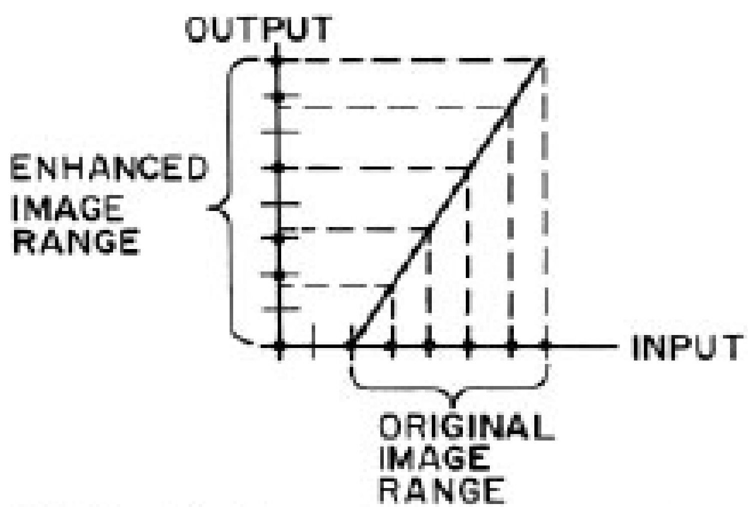
- **Two Approaches**
  - Contrast Manipulation
  - Histogram Modification

Contrast Manipulation

- Transfer Function
  - Linear
  - Nonlinear
  - Piecewise



Continuous Image

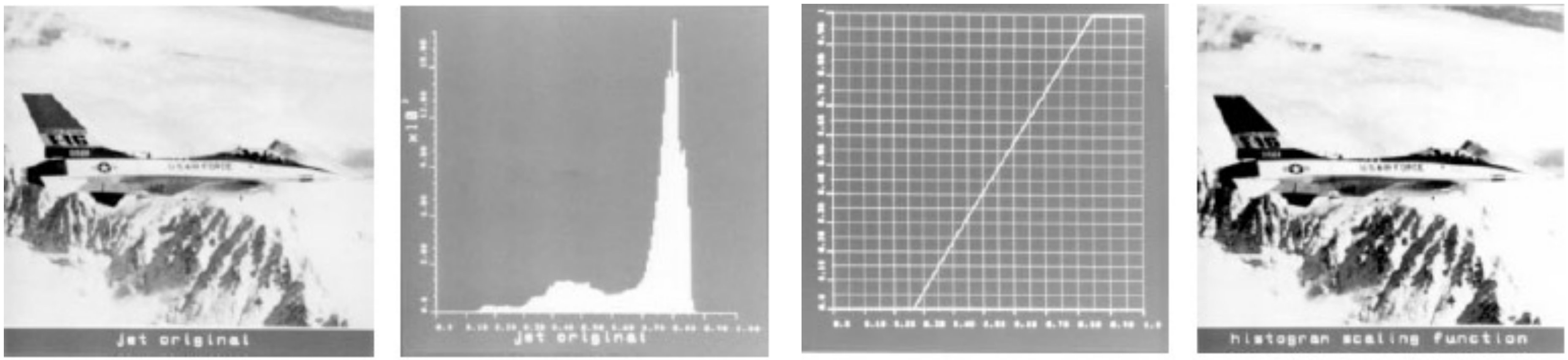


Quantized Image

Contrast Manipulation

- **Linear scaling and clipping**

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

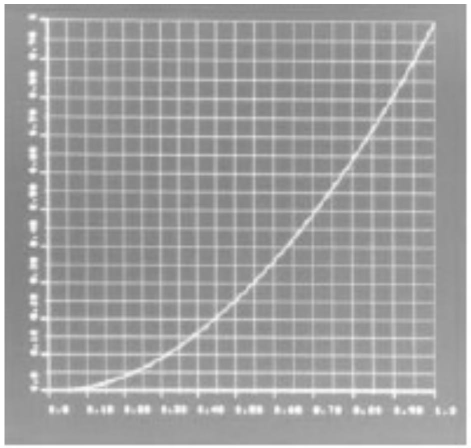


(*a*) Original (*b*) Original histogram (*c*) Transfer function (*d*) Contrast stretched

Contrast Manipulation

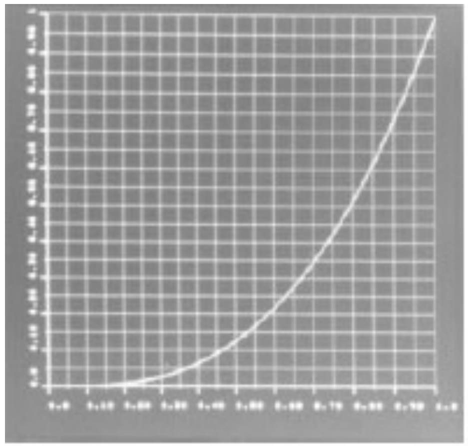
- **Power-Law**

$$G(j,k) = \left[F(j,k)\right]^p \qquad 0 \le F(j,k) \le 1$$



(a) Square function

(b) Square output



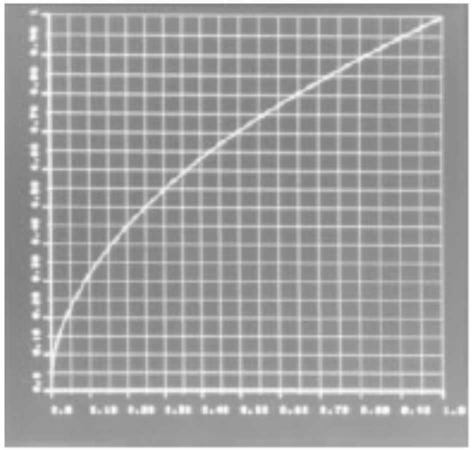
(c) Cube function

(d) Cube output

Contrast Manipulation

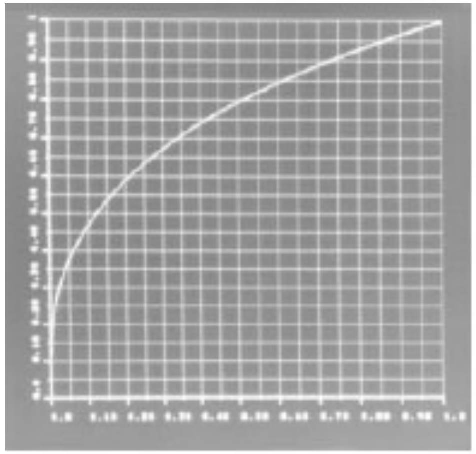
- **Power-Law**

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



(a) Square root function

(b) Square root output

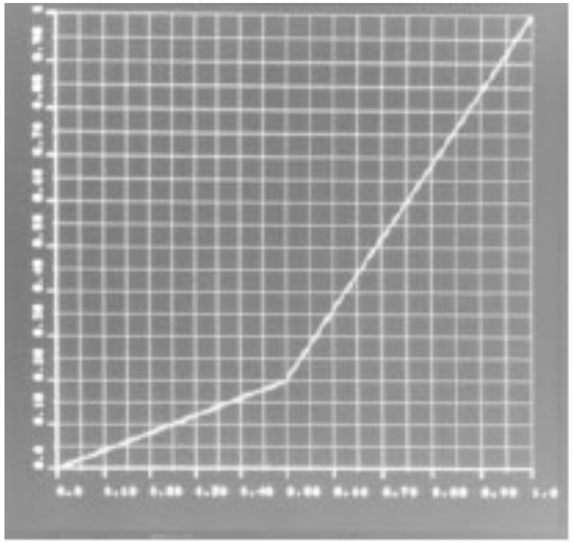


(c) Cube root function

(d) Cube root output

Contrast Manipulation

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



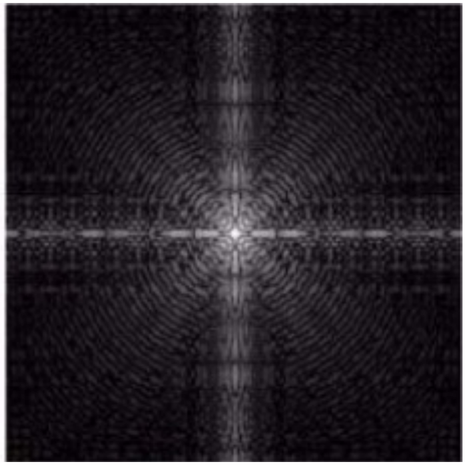
Can choose the area where we want to stretch or reduce the contrast

Contrast Manipulation

- **Logarithmic Point Transformation**

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

Fourier Spectrum



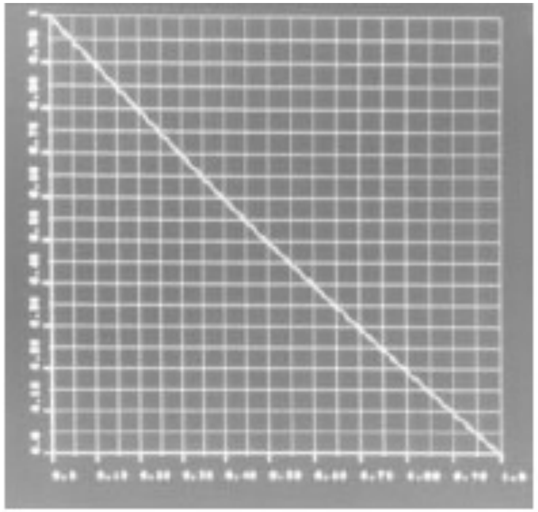
$0 \sim 1.5 \times 10^6 \longrightarrow$ 0~6.2

Useful for scaling image arrays with a very wide dynamic range

Contrast Manipulation

- **Reverse Function**

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



(a) Reverse function

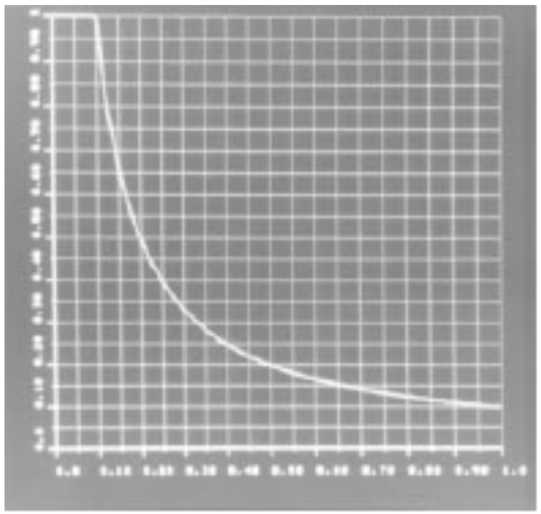
(b) Reverse function output

Able to see more details in dark areas of an image

Contrast Manipulation

- **Inverse Function**

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

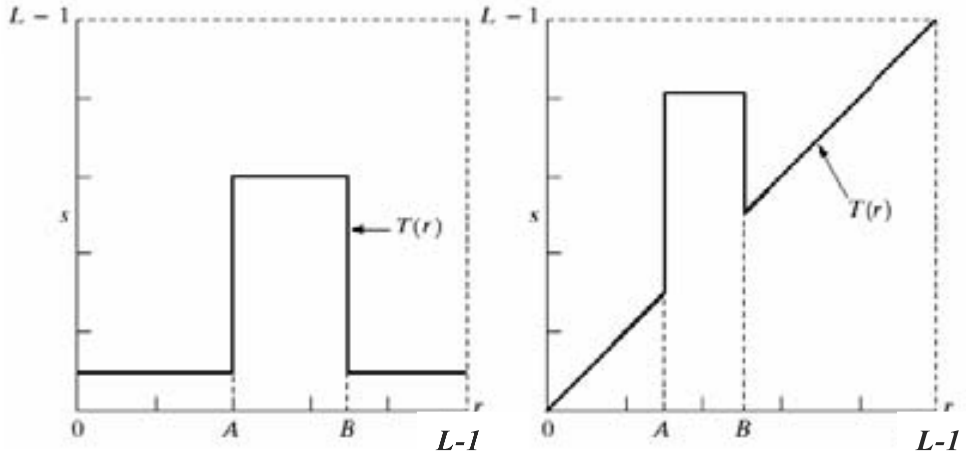


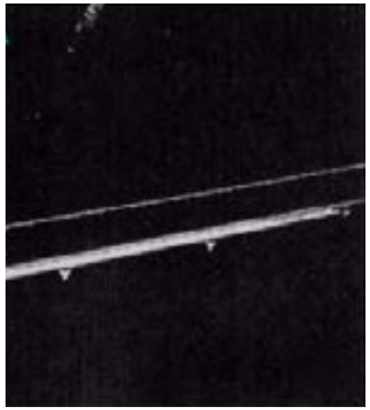
(c) Inverse function

(d) Inverse function output

Contrast Manipulation

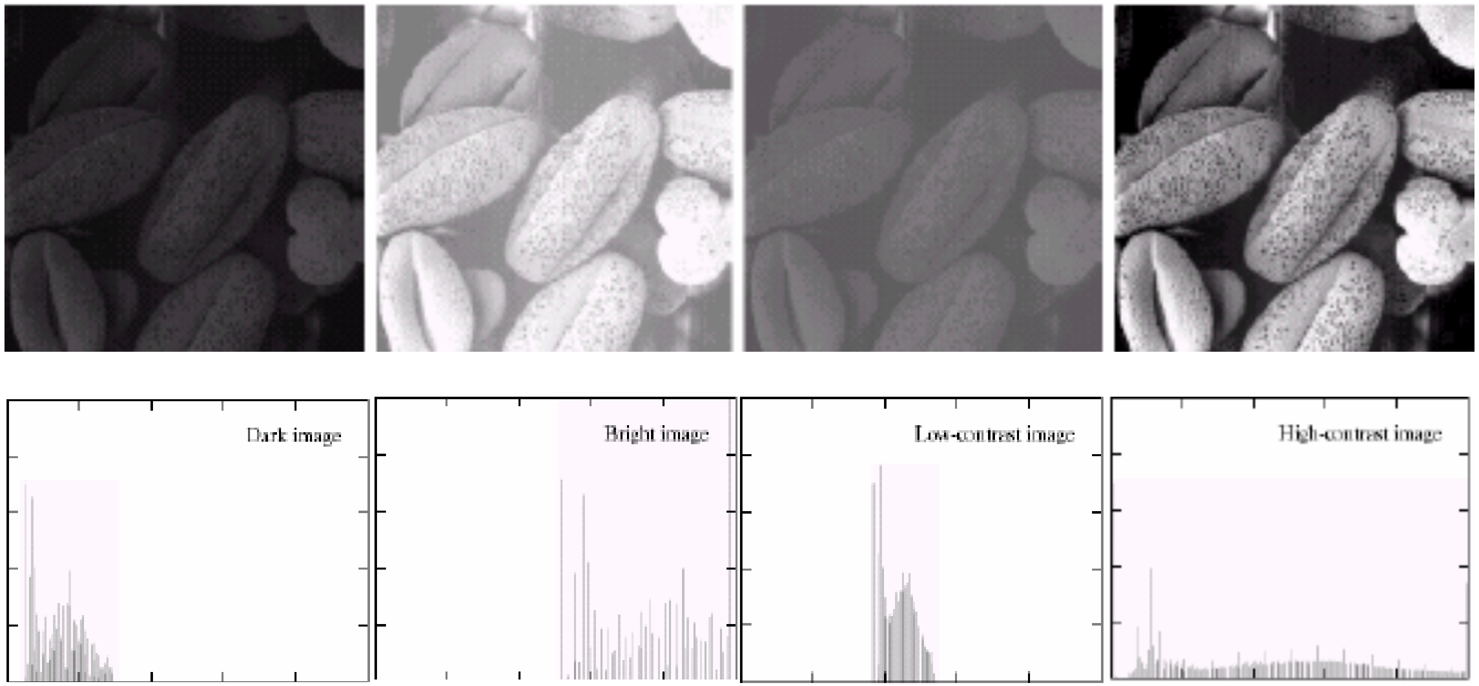
- **Amplitude-Level Slicing (Gray-Level Slicing)**





Histogram Modification

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

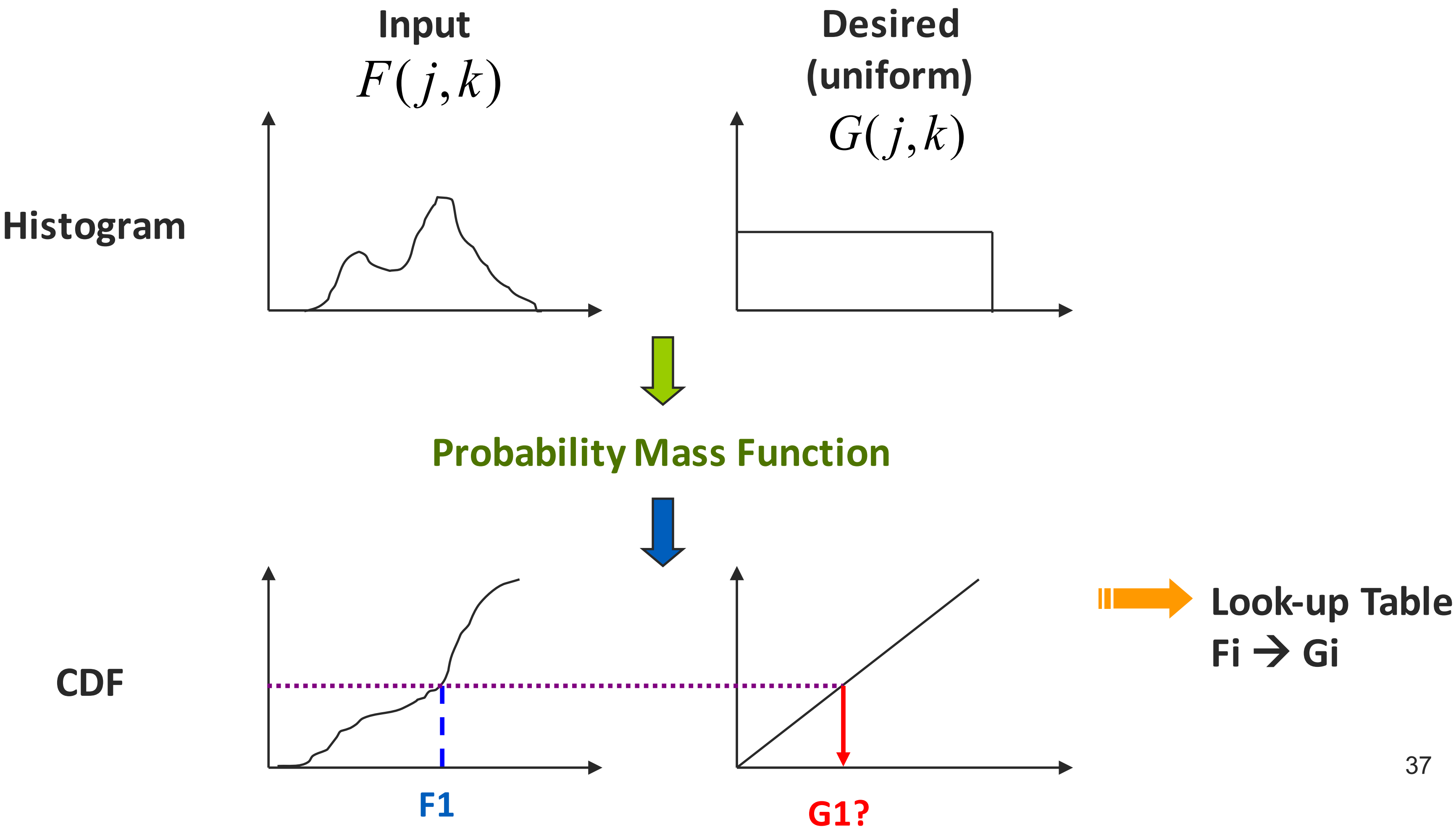


Histogram Modification

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

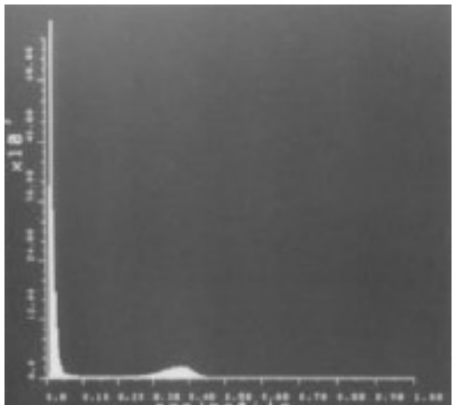
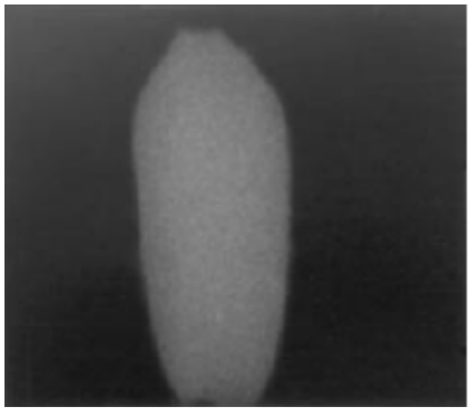
Histogram Equalization

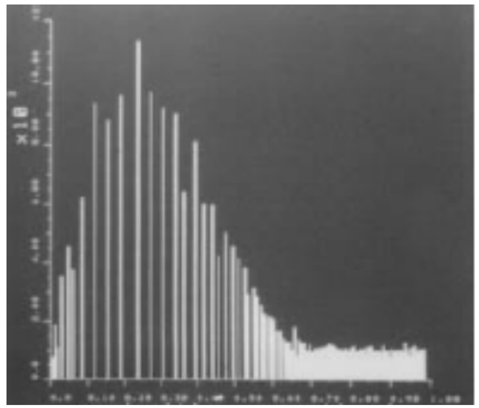
- **Transfer Function**



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

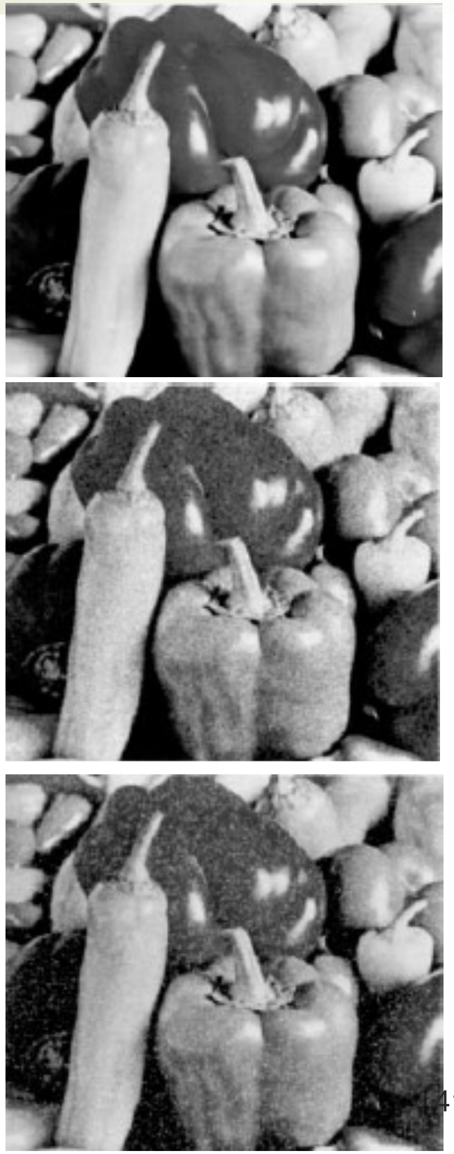
N: # of total pixels

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

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 x n
    - m=2a+1, n=2b+1,
      where a and b are nonnegative integers
    - e.g. 3x3 mask

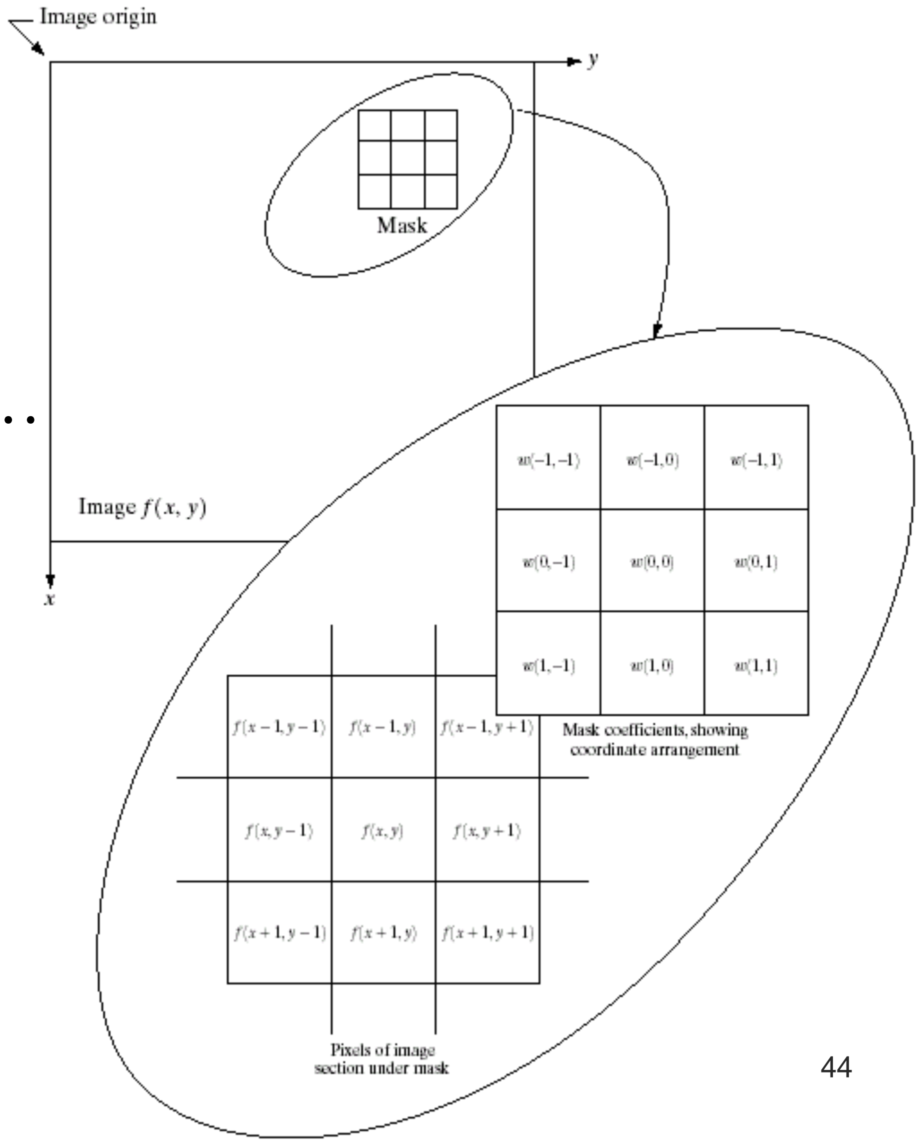
$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	W(0,0)	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

- **Spatial Filtering/Convolution**

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

Basics of Spatial Filtering

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

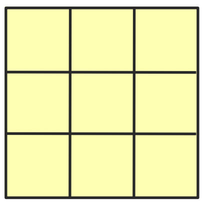


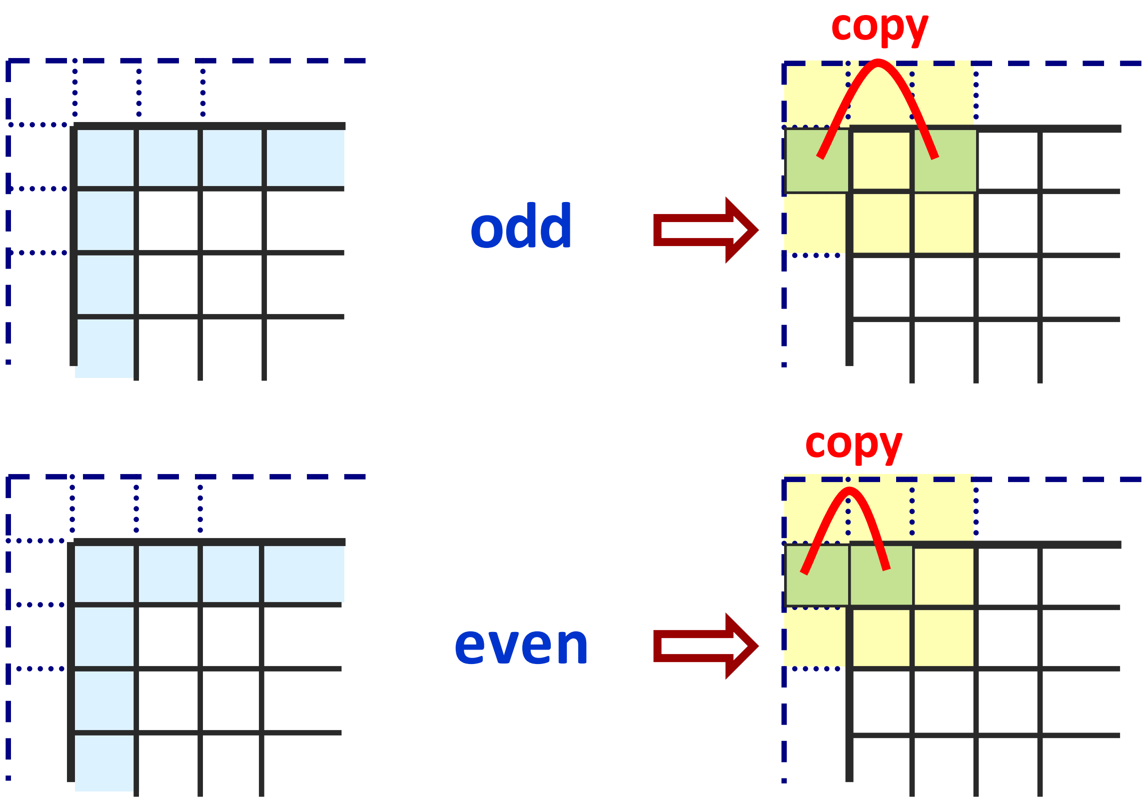
Q: Boundary pixels?

Basics of Spatial Filtering

- **Boundary Extension (3x3 mask)**

e.g.
3x3 mask, w





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

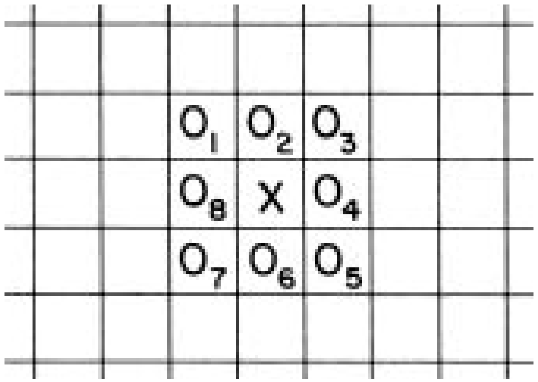
Noise Cleaning

- **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**



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

How to choose ε ?

Larger window?

Impulse Noise Removal

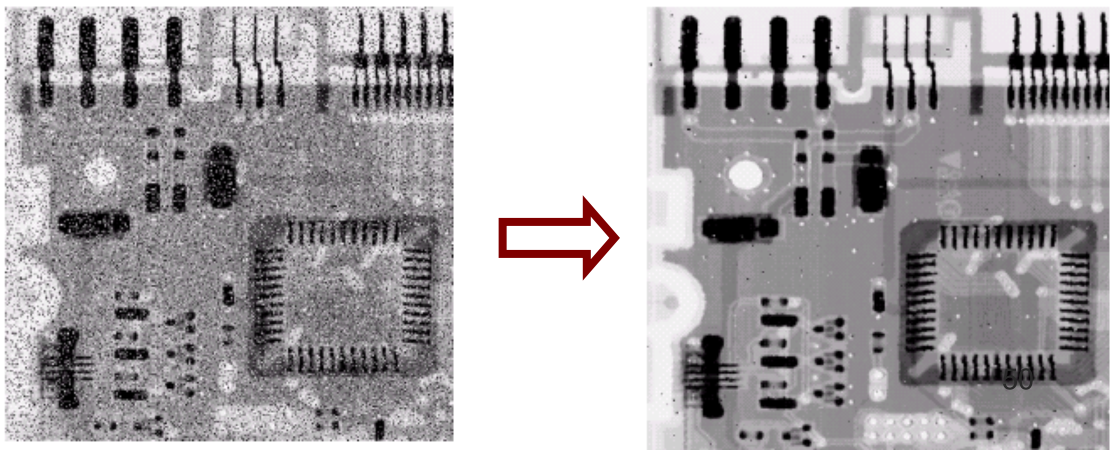
- **Median filtering**

  $$a_1, \ldots, a_N \quad where \quad N \quad is \quad odd$$

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

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

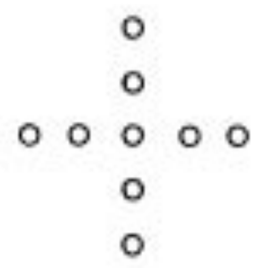
→ Median is 3



Impulse Noise Removal

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

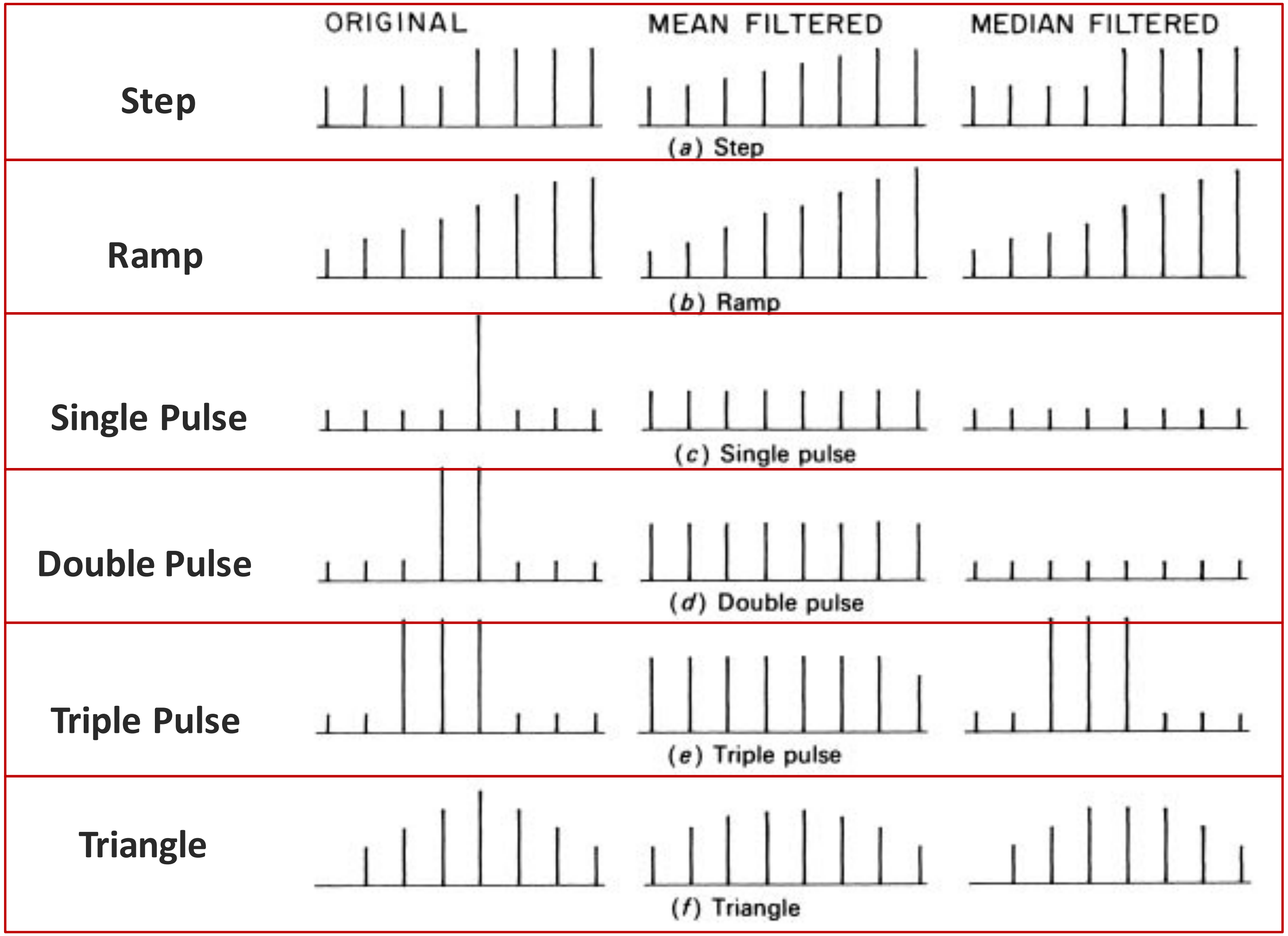
square



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

        → MED(a, b, c, d, e)

        =max(min(a,b,c) , min(a,b,d), …)

        =min(max(a,b,c) , max(a,b,d), …)

        → 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 → 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))

    → PMED(a, b, c, d, e)

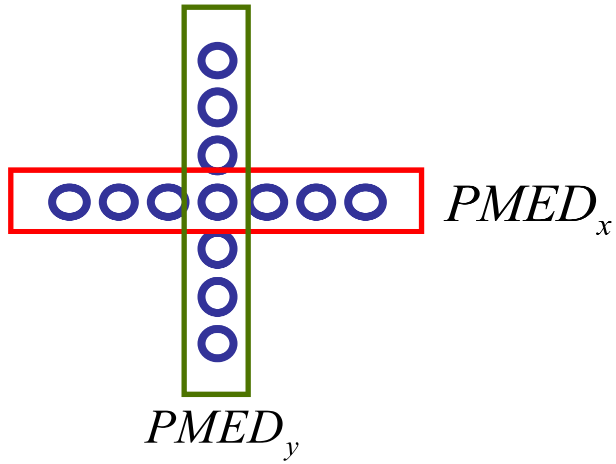
    = 0.5 * (A + B) = 0.5 * (MAXMIN + MINMAX)

    ~ MED(a, b, c, d, e)

Impulse Noise Removal

- **Pseudomedian filtering (PMED)**
  - **2D case**

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



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

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

Q: same results?

MINMAX

MAXMIN of MINMAX

Quality Measurement

- Peak signal-to-noise ratio (PSNR)
  - Mean squared error (MSE)

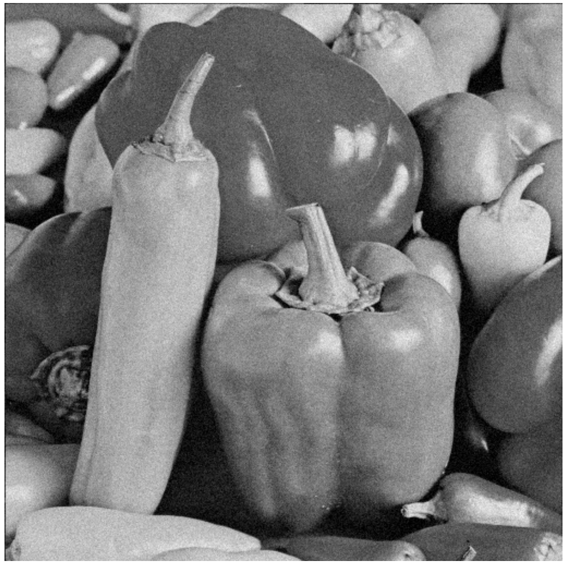
$$MSE = \frac{1}{w*h}\sum_j \sum_k \left[F(j,k) - F'(j,k)\right]^2$$

  - The PSNR is defined as

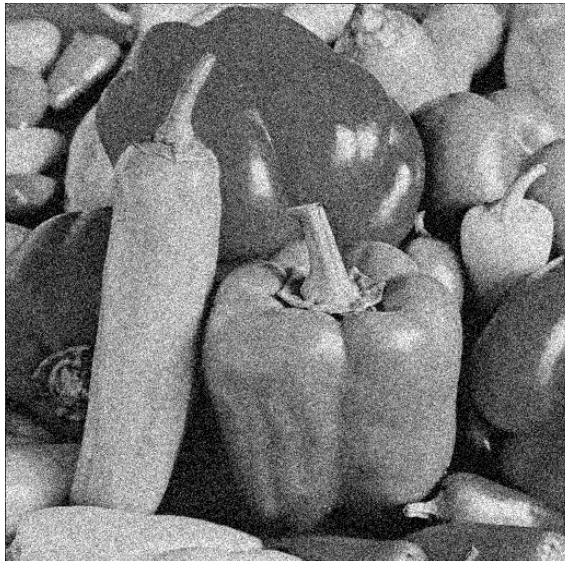
$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right)$$

Example

Original image



Gaussian noise (σ=10)
PSNR : 28.18dB



Gaussian noise (σ=30)
PSNR : 18.81dB

Q: Represent perceived visual quality?