Digital Image Processing

Ming-Sui (Amy) Lee

Lecture 02

Announcement

- **Class Information**
  - **Teaching Assistant**
    - 郭柏辰 @532
      Office Hours: 14:00 ~ 16:00, Monday
    - 黃聖凱 @532
      Office Hours: 16:00 ~ 18:00, Thursday
    - Email: dip.mslee@gmail.com

Announcement

- Class Information
  - Class website
    - https://ceiba.ntu.edu.tw/1062DIP
    - 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. 21, 2018

Image Enhancement

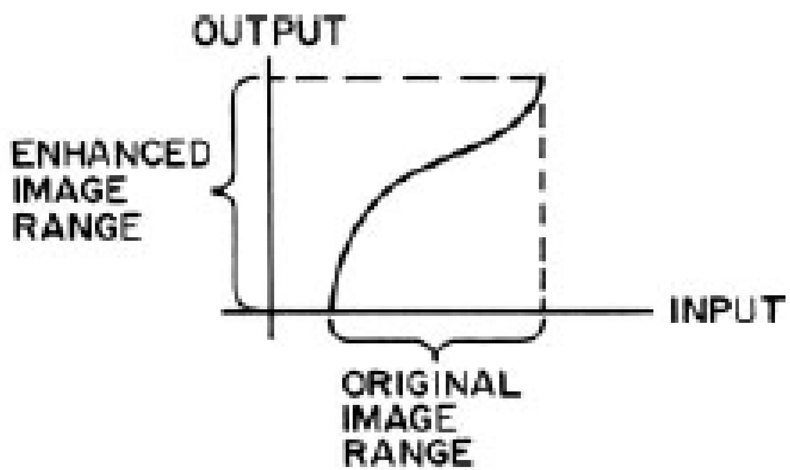
Image Enhancement

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

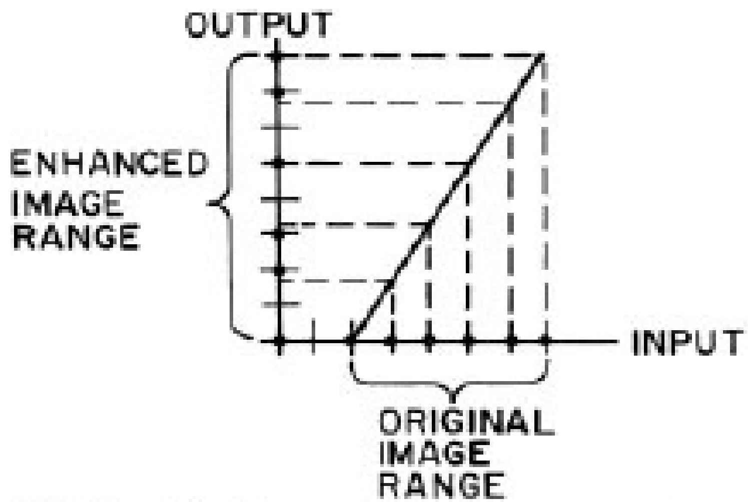
- **Two Approaches**
  - Contrast Manipulation
  - Histogram Modification

Contrast Manipulation

- **Transfer Function** relation between input & output
  - Linear
  - Nonlinear
  - Piecewise



Continuous Image



Quantized Image

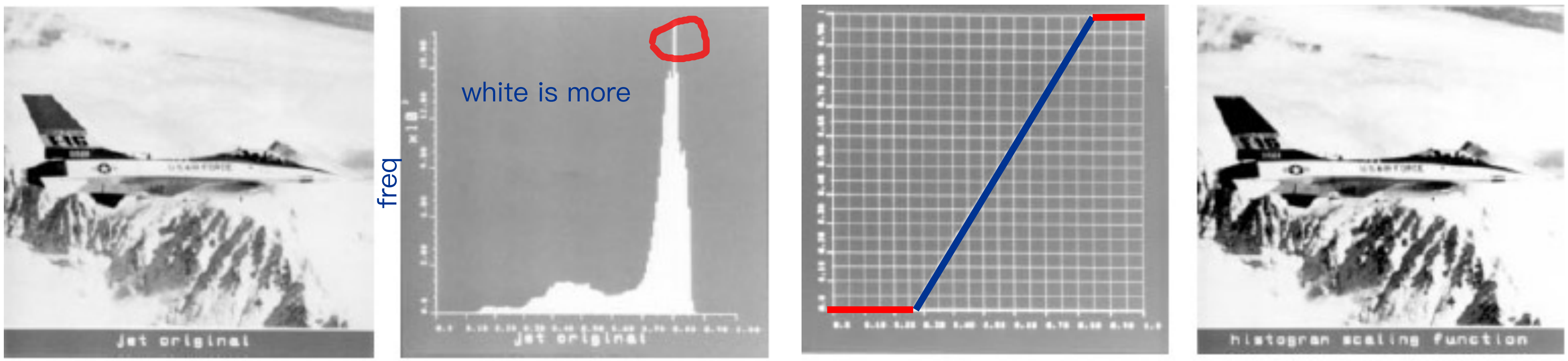
Contrast Manipulation

- **Linear scaling and clipping**

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

intensity

gray scale –> normalize –> 0 … 1



(a) Original

(b) Original histogram

intensity

(c) Transfer function

red lines(clipping): input –> same output

blue line: linear scaling

(d) Contrast stretched

Both black and white become intense

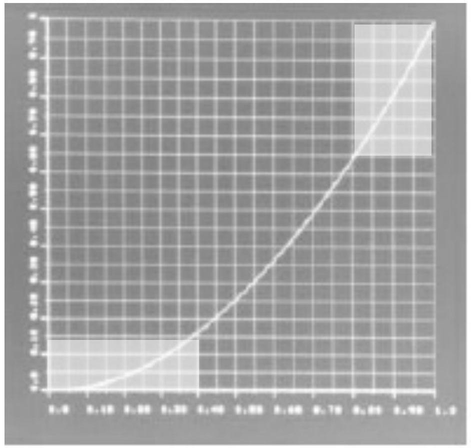
Contrast Manipulation

- **Power-Law**

p > 1: darkness become detailed

255: bomb; with normalization, it's okay.

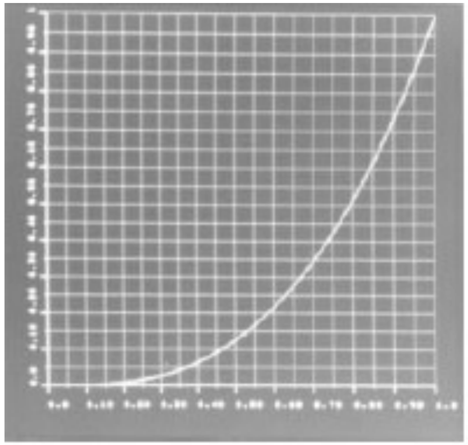
$$G(j,k) = \left[F(j,k)\right]^p \qquad 0 \le F(j,k) \le 1$$



(a) Square function

p = 2

(b) Square output



(c) Cube function

p = 3

(d) Cube output

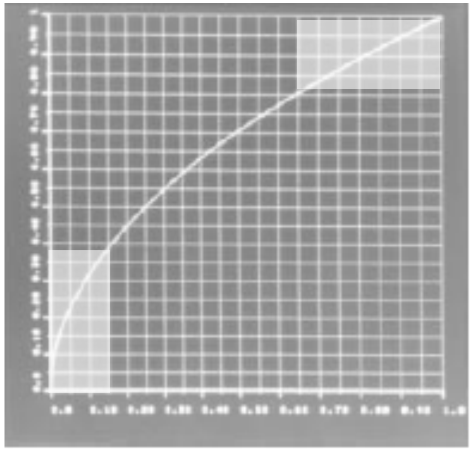
We care about cloud!

Contrast Manipulation

- **Power-Law**

p < 1: whiteness become detailed

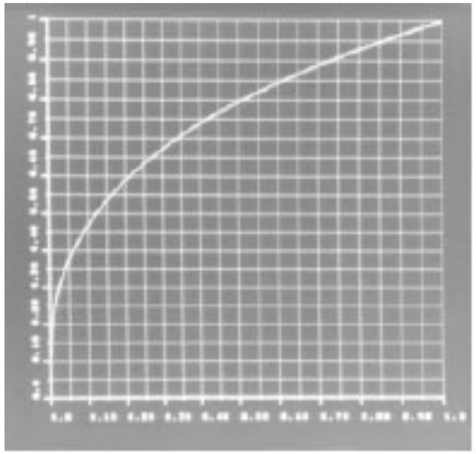
$$G(j,k) = \left[F(j,k)\right]^p \quad 0 \le F(j,k) \le 1$$



(a) Square root function

p = 1 / 2

(b) Square root output



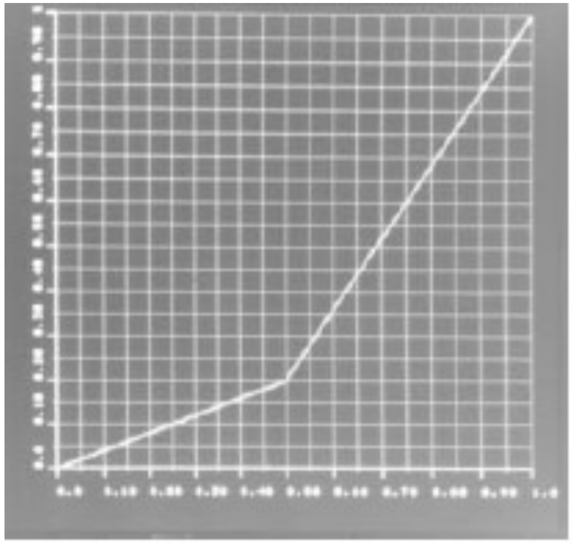
(c) Cube root function

p = 1 / 3

(d) Cube root output

Contrast Manipulation

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



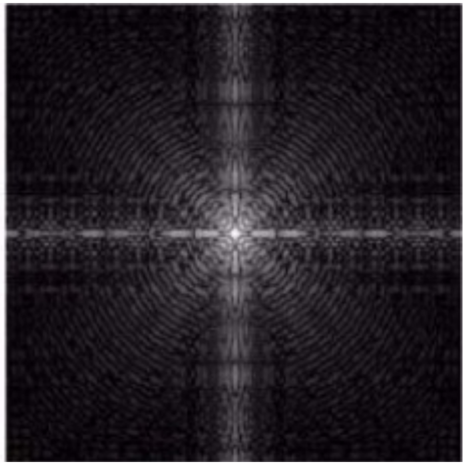
Can choose the area where we want to stretch or reduce the contrast

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



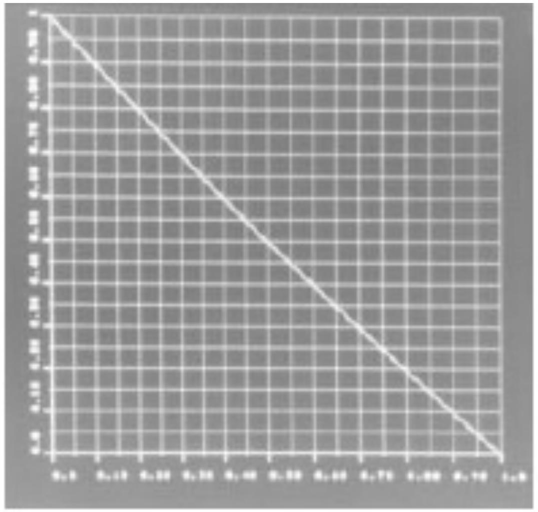
$0 \sim 1.5 \times 10^6 \longrightarrow$ 0~6.2

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

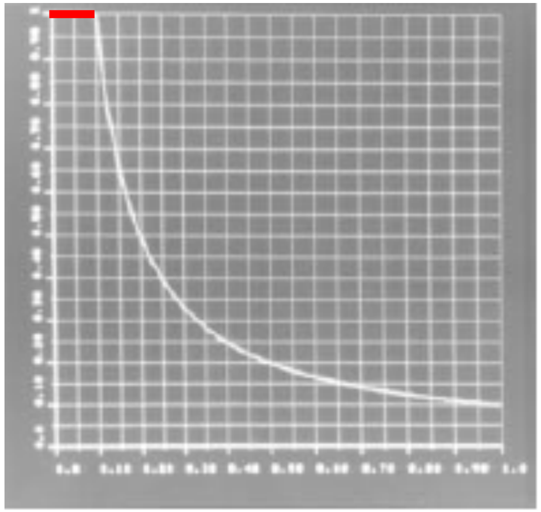
Able to see more details in dark areas of an image

Contrast Manipulation

- **Inverse Function**

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

focus on mountain

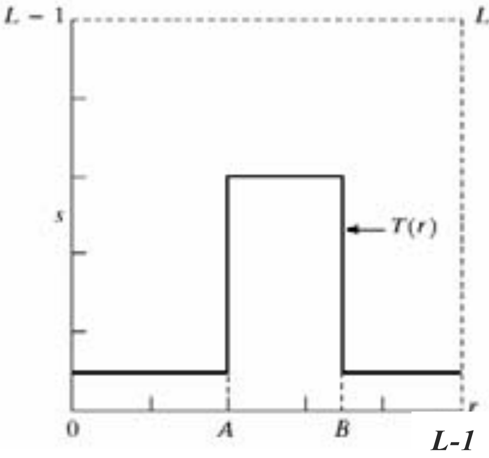


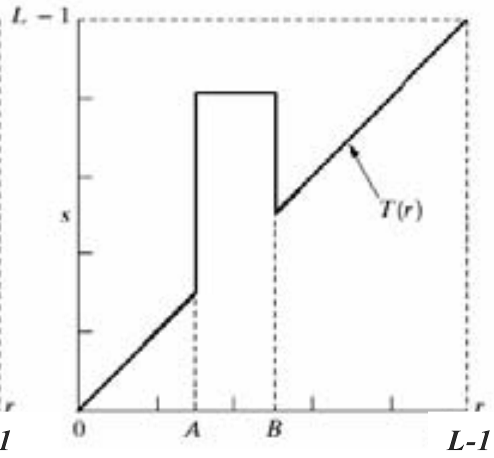
(c) Inverse function

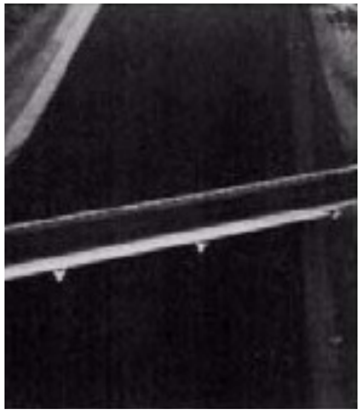
(d) Inverse function output

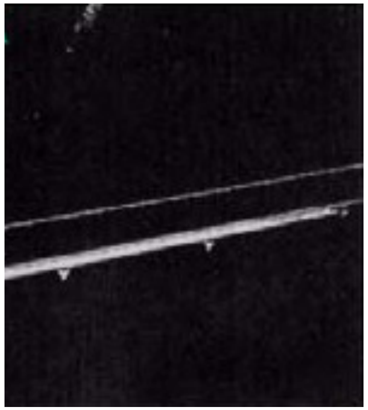
Contrast Manipulation

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



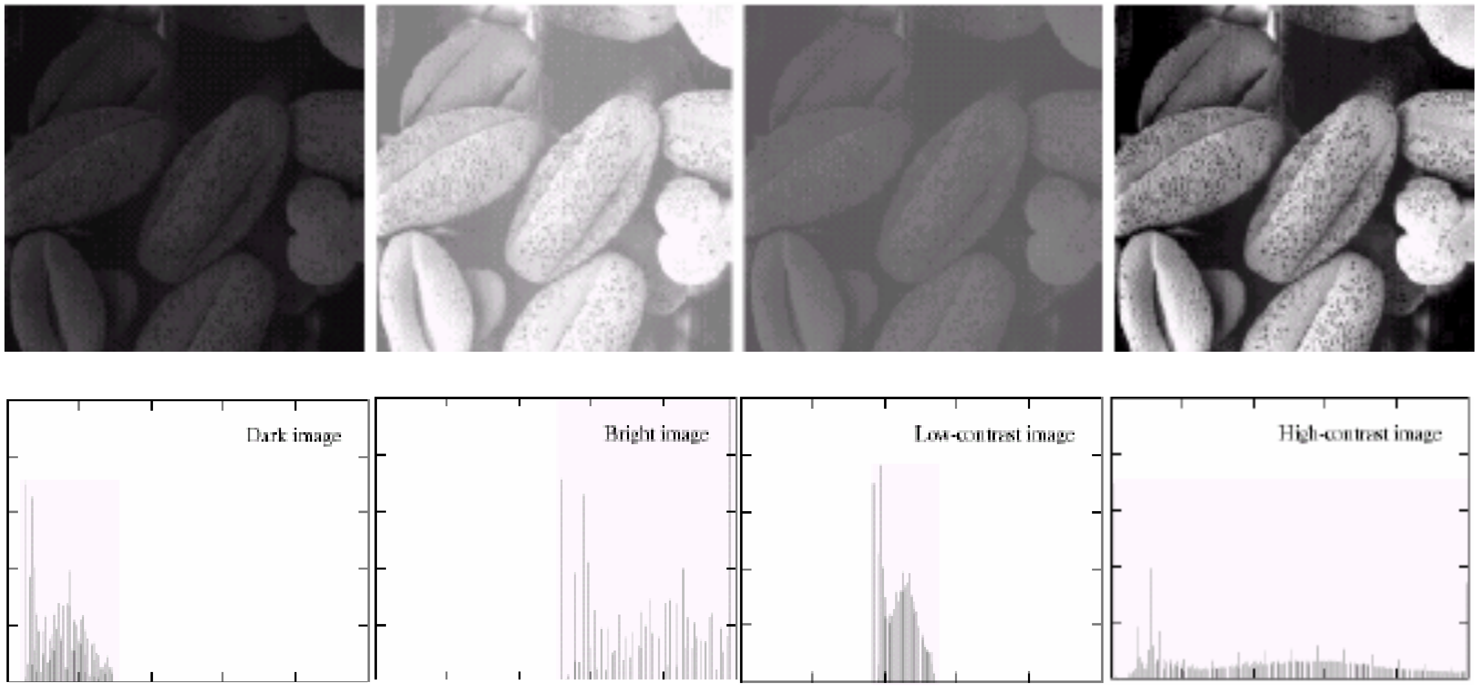






Histogram Modification

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

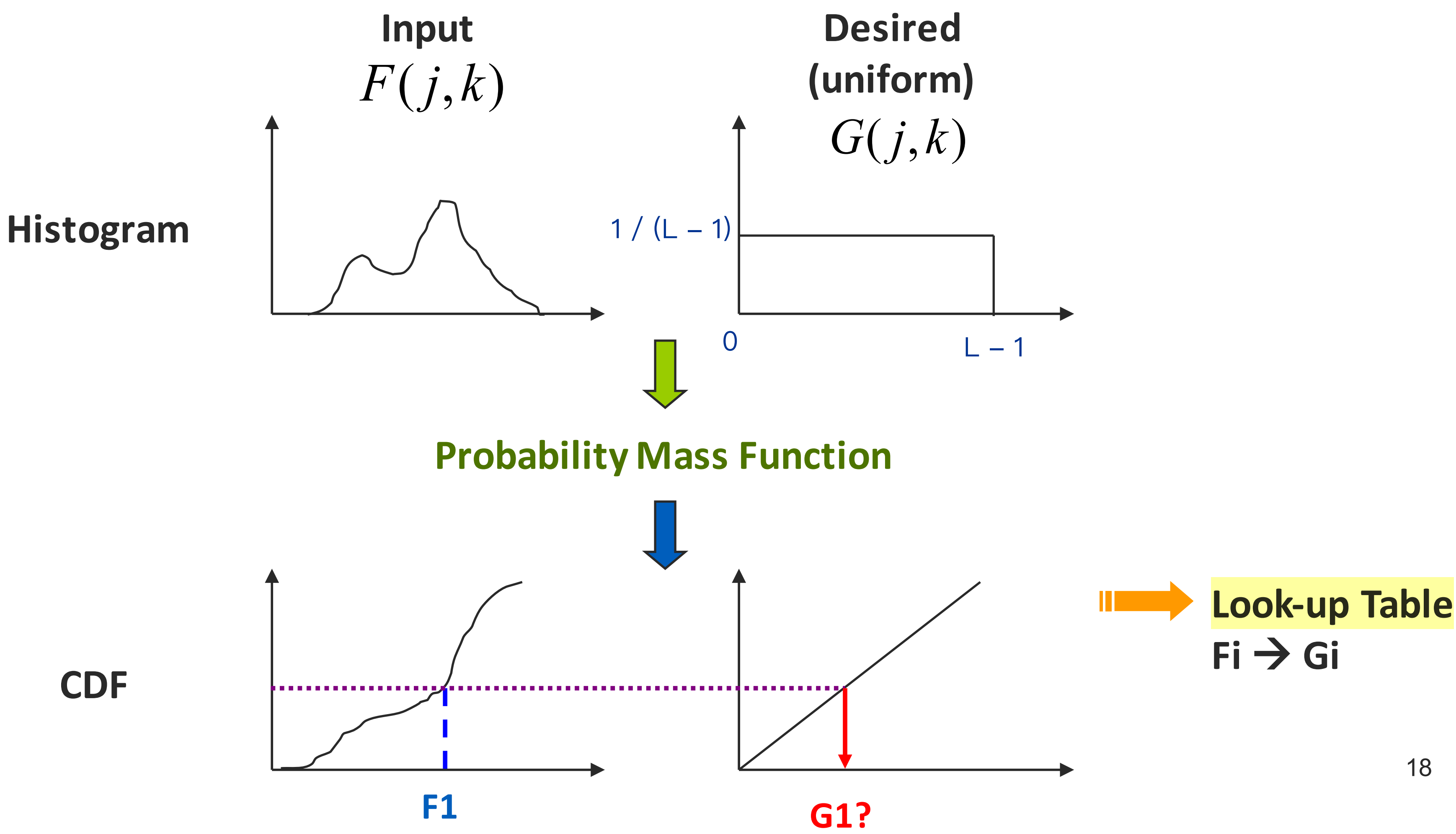


Histogram Modification

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

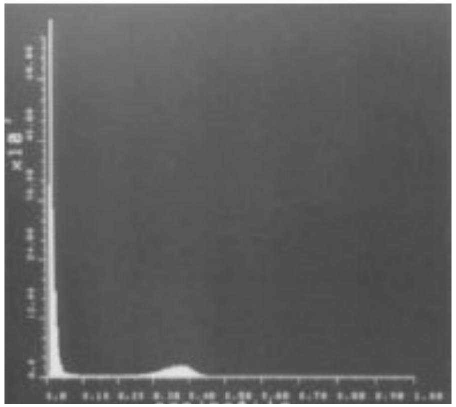
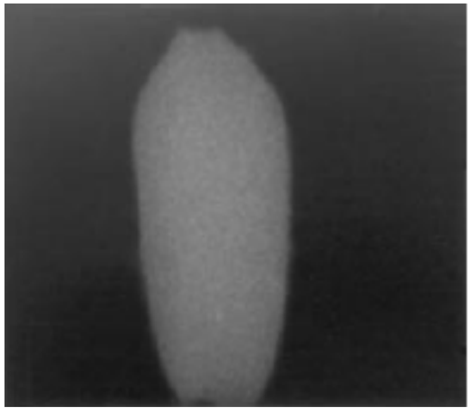
Histogram Equalization

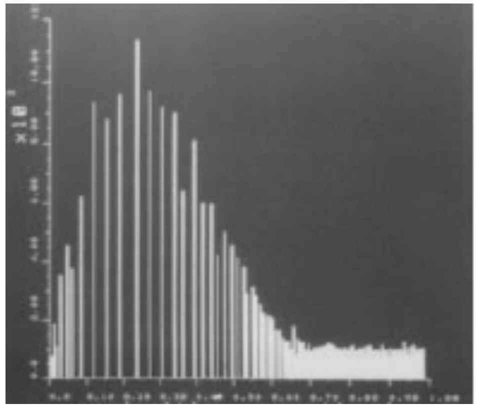
- **Transfer Function**



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

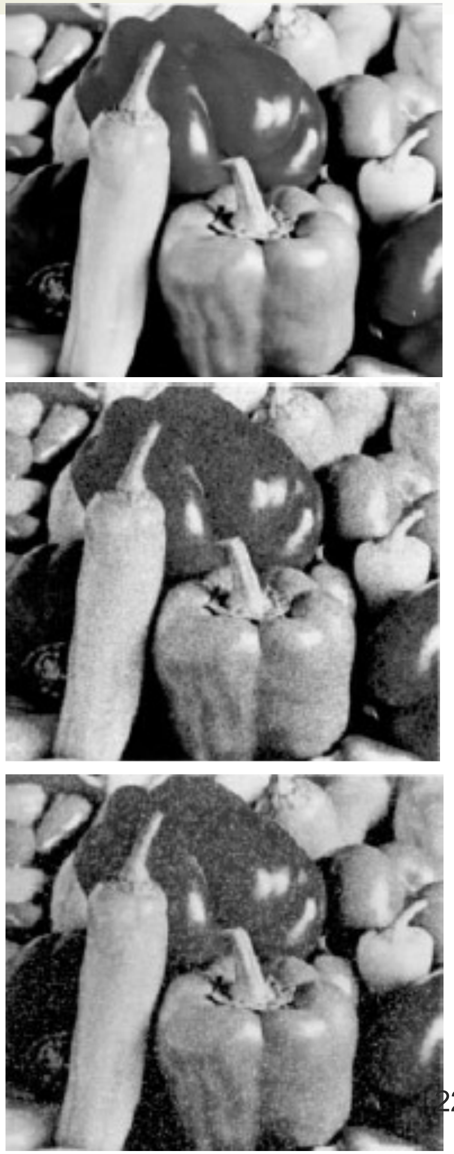
N: # of total pixels

- 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 point
  - not spatially correlated point <–> point
  - freq：Hz = 次 / sec — higher spatial frequency
    改變次數 / 單位空間(space)



Noise Cleaning

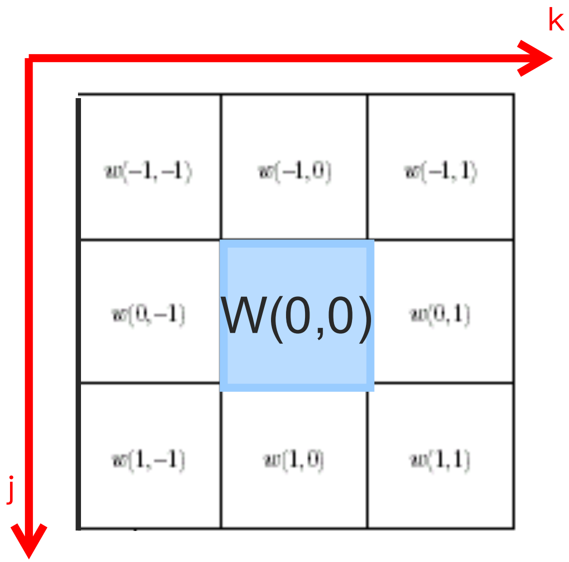
- Two types of noise
  - Uniform Noise
    - Additive uniform noise, Gaussian noise
  - Impulse Noise
    - Salt and pepper noise

- 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



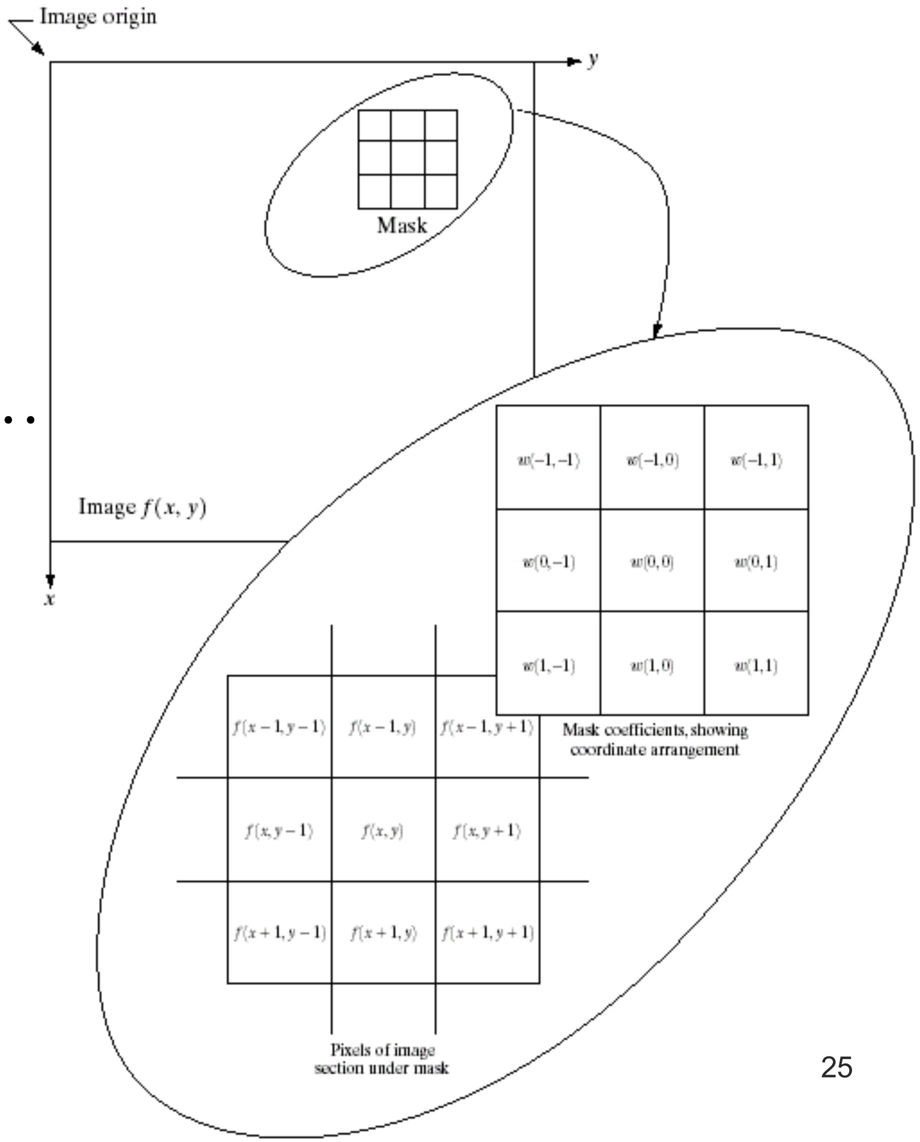
- **Spatial Filtering/Convolution**

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

output: a value

Basics of Spatial Filtering

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

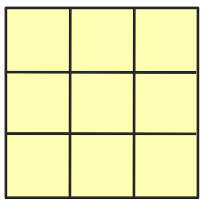


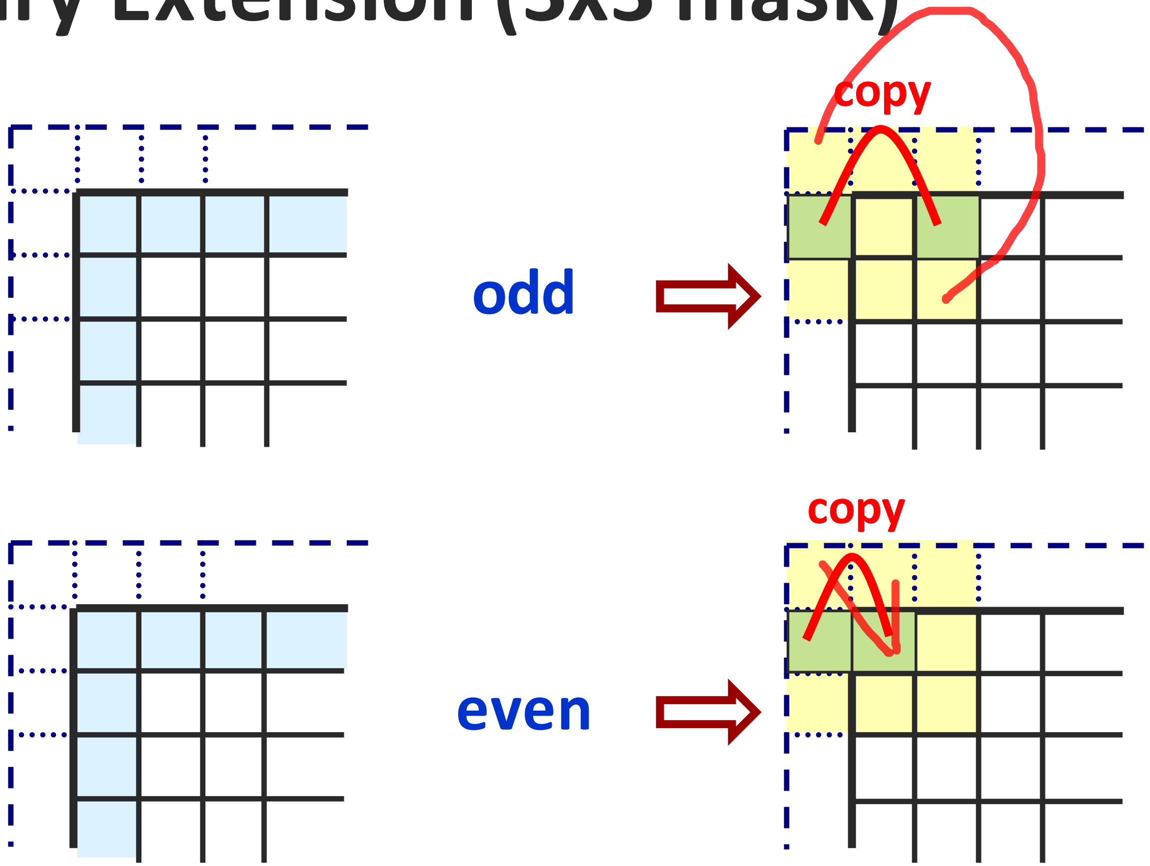
Q: Boundary pixels?

Basics of Spatial Filtering

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

e.g.
3x3 mask, w





Q: 5x5 mask?

Noise Cleaning

- **Uniform noise** let low freq component pass(i.e., high freq is discarded)
  - **Perform low-pass filtering** symmetric, square, all sum up to 1

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

  - **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

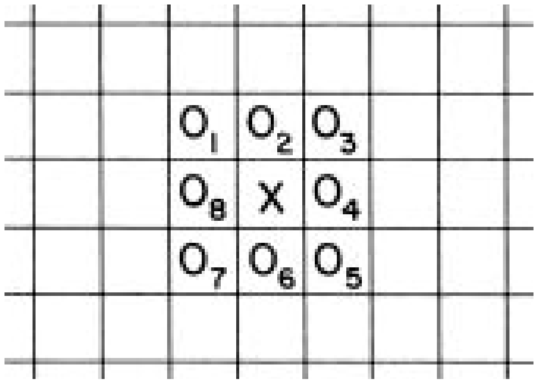
Noise Cleaning

- 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

0 or 255

- **Outlier detection**



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

How to choose ε **?** eg. standard deviation

Larger window?

Impulse Noise Removal

- **Median filtering**

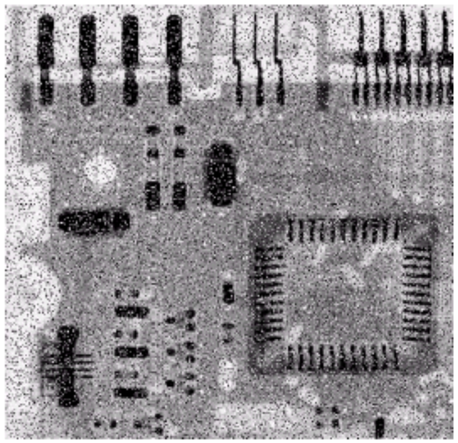
  $$a_1, \ldots, a_N \quad where \quad N \quad is \quad odd$$

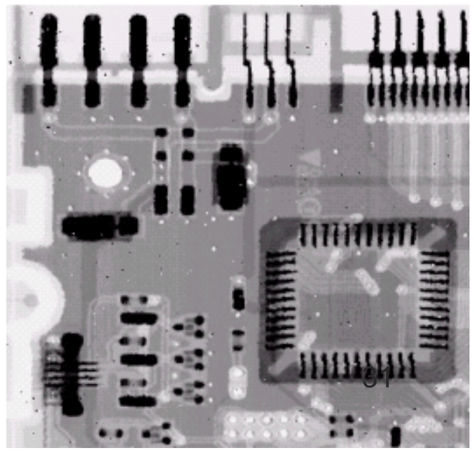
  - **sort those values in order**
  - **pick the middle one in the sorted list**
  - **e.g. 3x3 mask:**

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

→ **Median is 3**

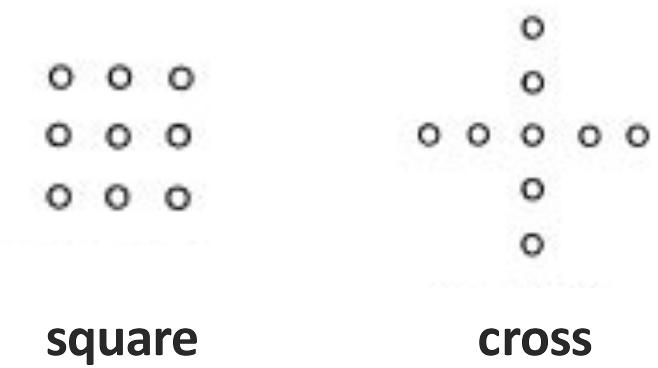
1, 1, 2, 3, 3, 5, 6, 7, 8





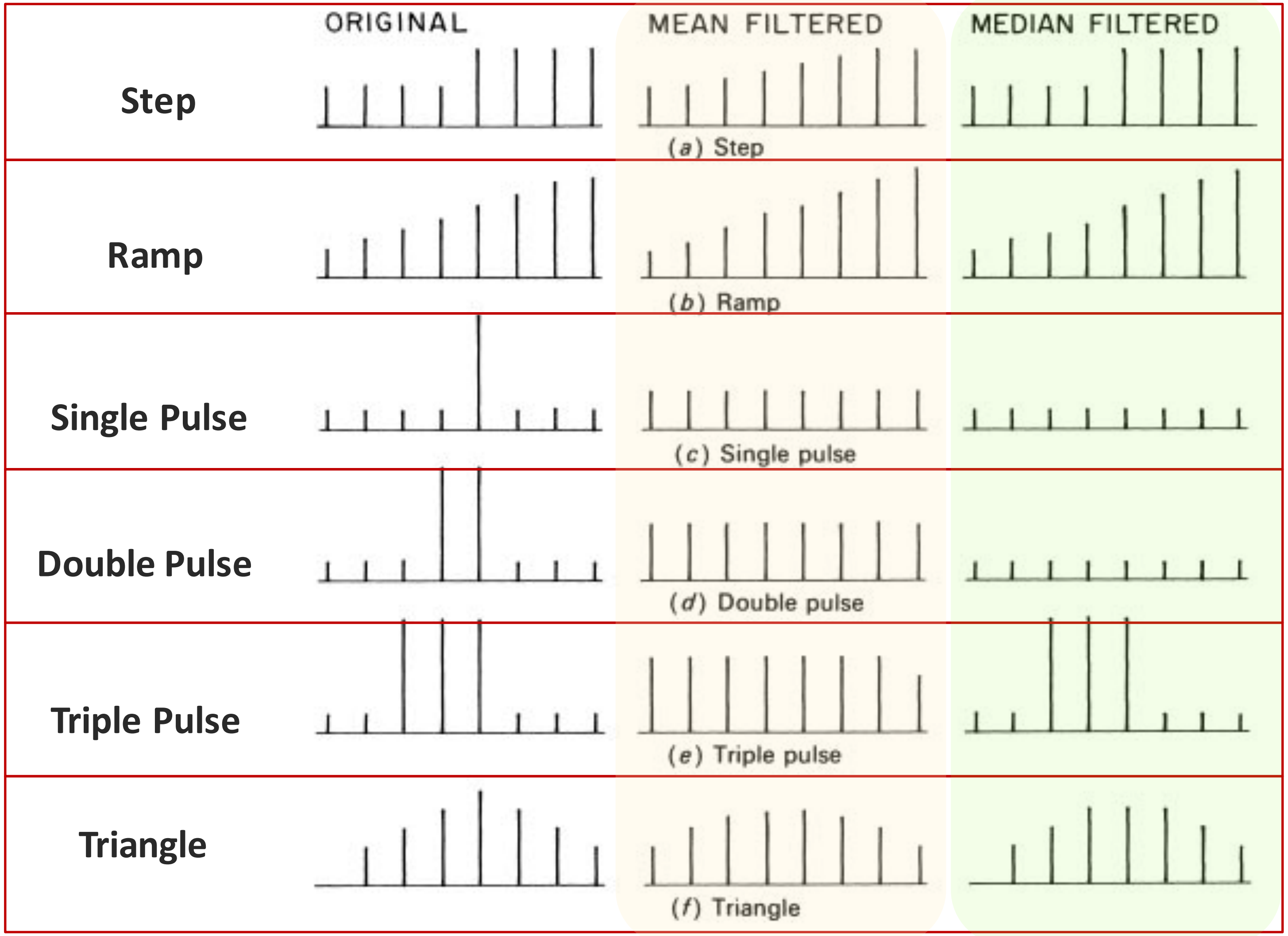
Impulse Noise Removal

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



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)

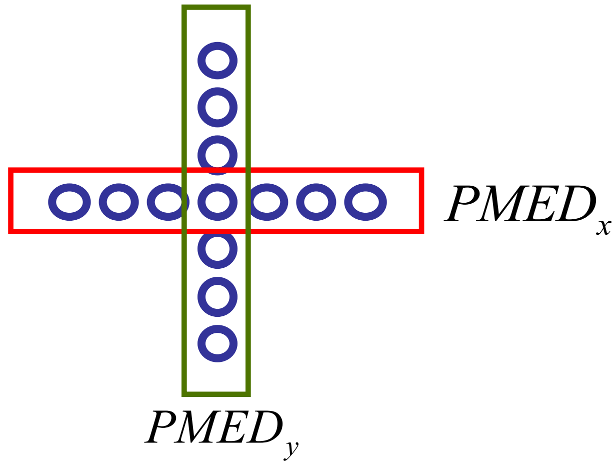
    = 0.5 * (A + B) = 0.5 * (MAXMIN + MINMAX)

    ~ 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 (white)
  - MINMAX
    - Remove pepper noise (black)
  - 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)**

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

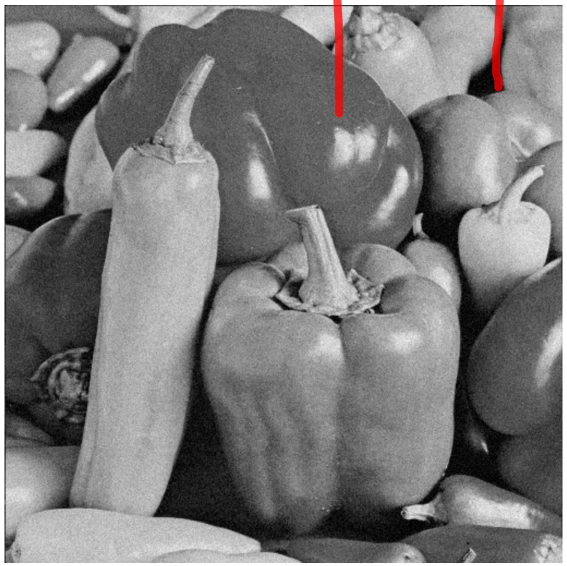
  - **The PSNR is defined as**

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

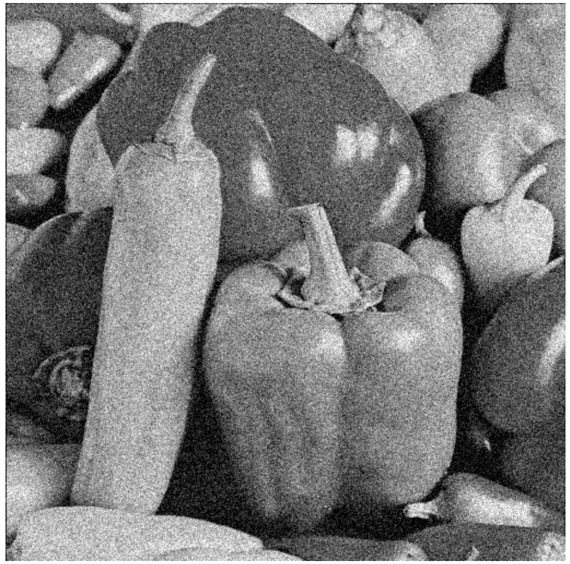
Example

σ大 σ小

Original image



Gaussian noise (σ=10)
PSNR : 28.18dB



Gaussian noise (σ=30)
PSNR : 18.81dB

Q: Represent perceived visual quality?