KECE471 Computer Vision

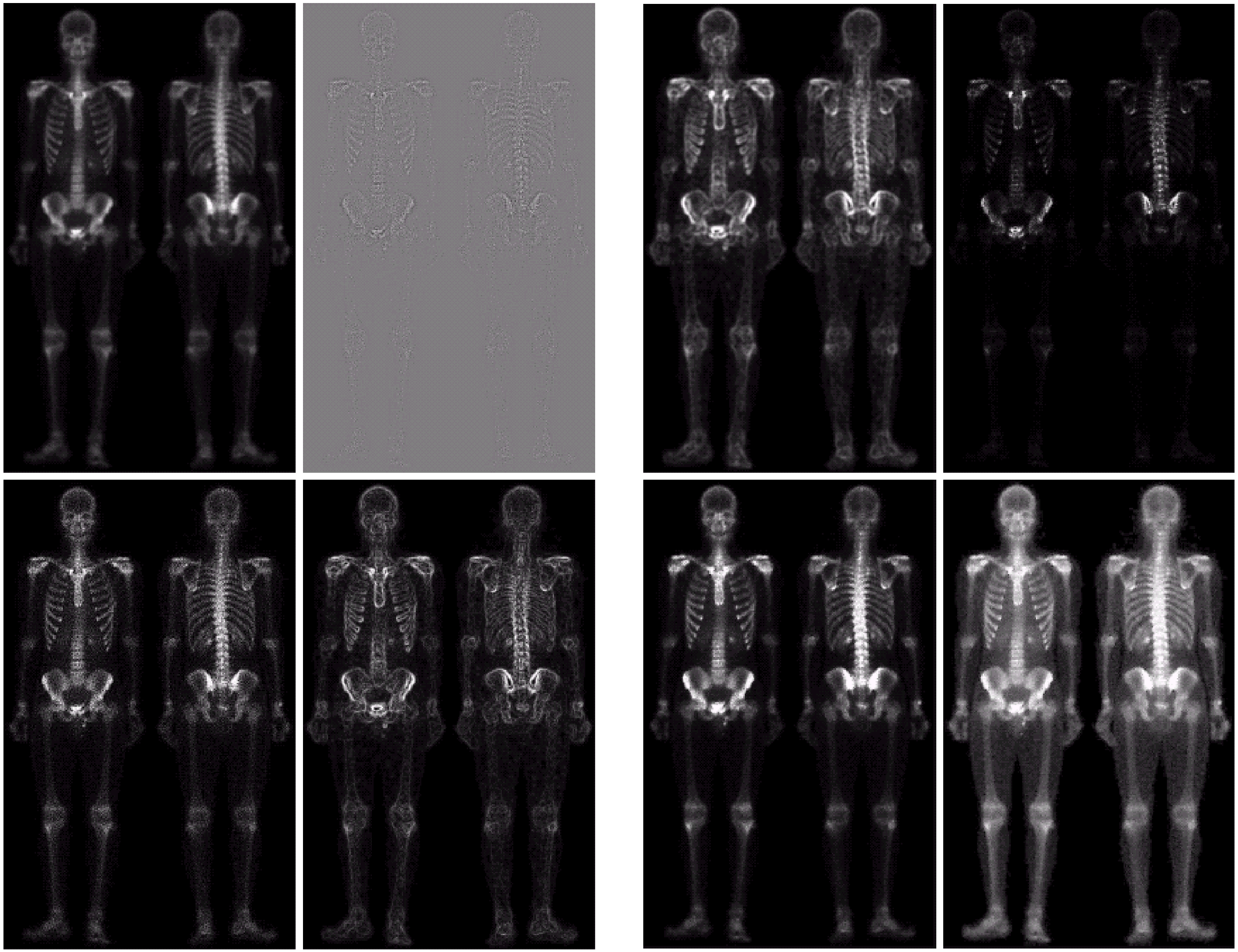
Filtering and Enhancing Images

Chang-Su Kim

Chapter 5, Computer Vision by Shapiro and Stockman
Note: Some figures and contents in the lecture notes of Dr. Stockman are used partly.

Make it better for human or machine vision

(a) original (b) Laplacian of (a) (e) smoothed (a) (f) = (c)x(e)



(c) = (a)+(b) (d) gradient of (a) (g) = (a)+(f) (h) power-law transform of (g)

Make it better for human or machine vision

a b c

FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

Make it better for human or machine vision

Strong noise

Medium noise

Weak noise

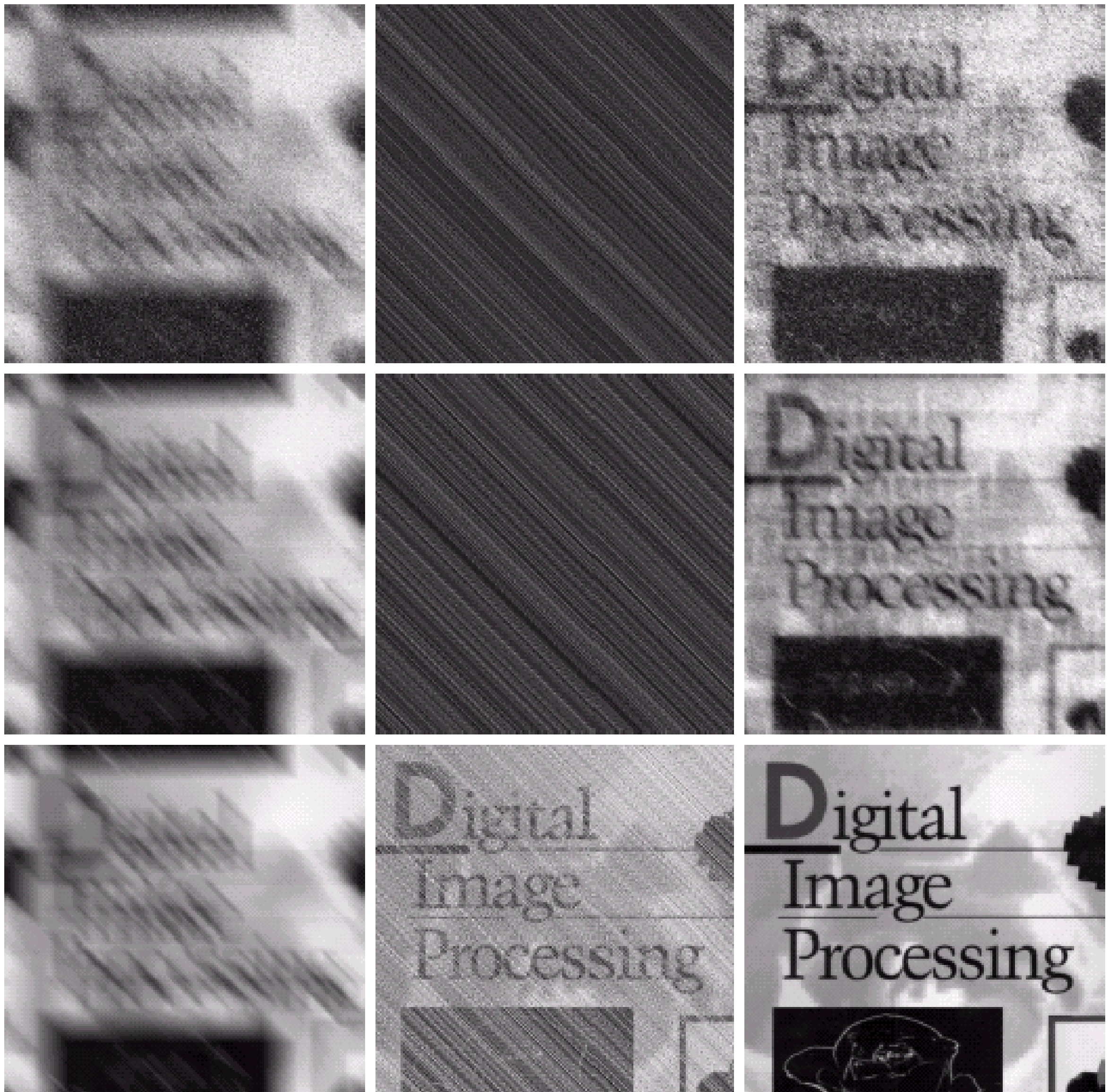


Image Enhancement and Restoration

- Enhancement
  - Subjective improvement of image quality to increase the detectability of important image details or objects by human or machine

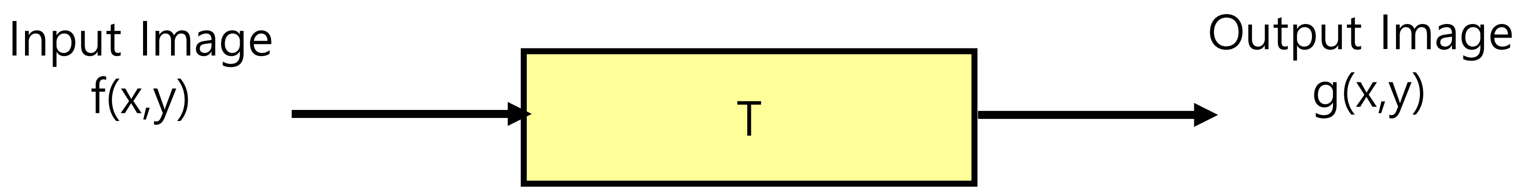
- Restoration
  - Object recovery of original image from degraded image
  - Knowledge on the image degradation process is required

Deraining

TIP
Sept. 2015

Personalized Enhancement

Point Operator



- Point processing

$$g(x,y) = T[\, f(x,y) \,]$$

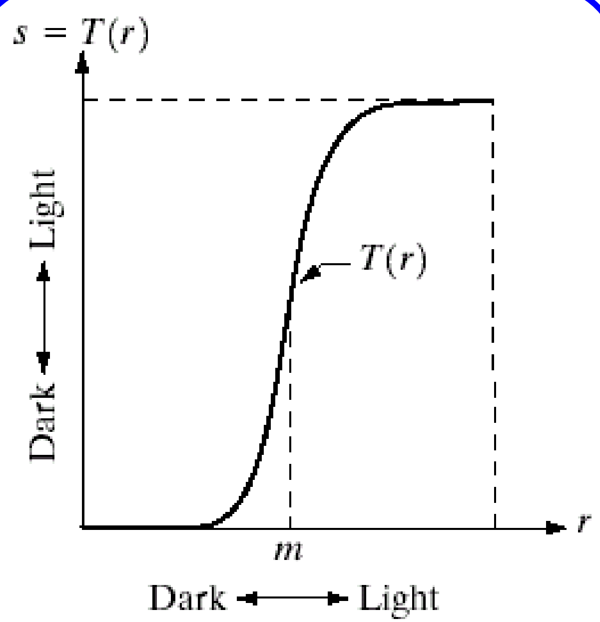
  - Output pixel value depends only on the input pixel value at the same location

  - The enhancement system is fully described by

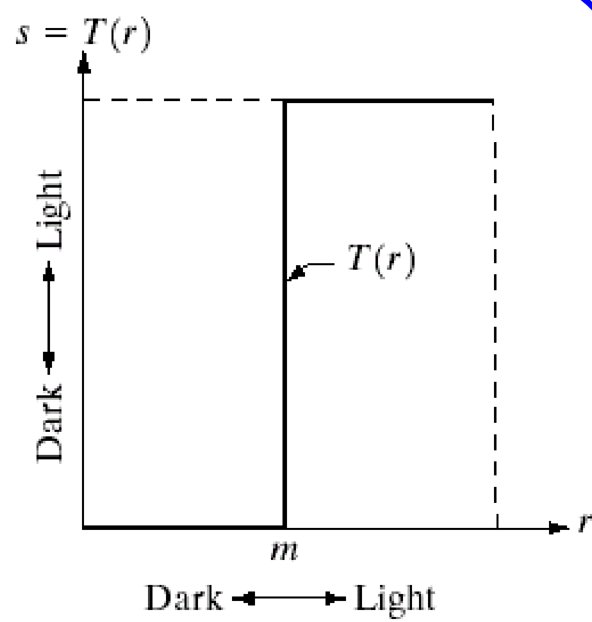
$$s = T(r)$$

where $s = g(x,y)$ and $r = f(x,y)$

Point Operator



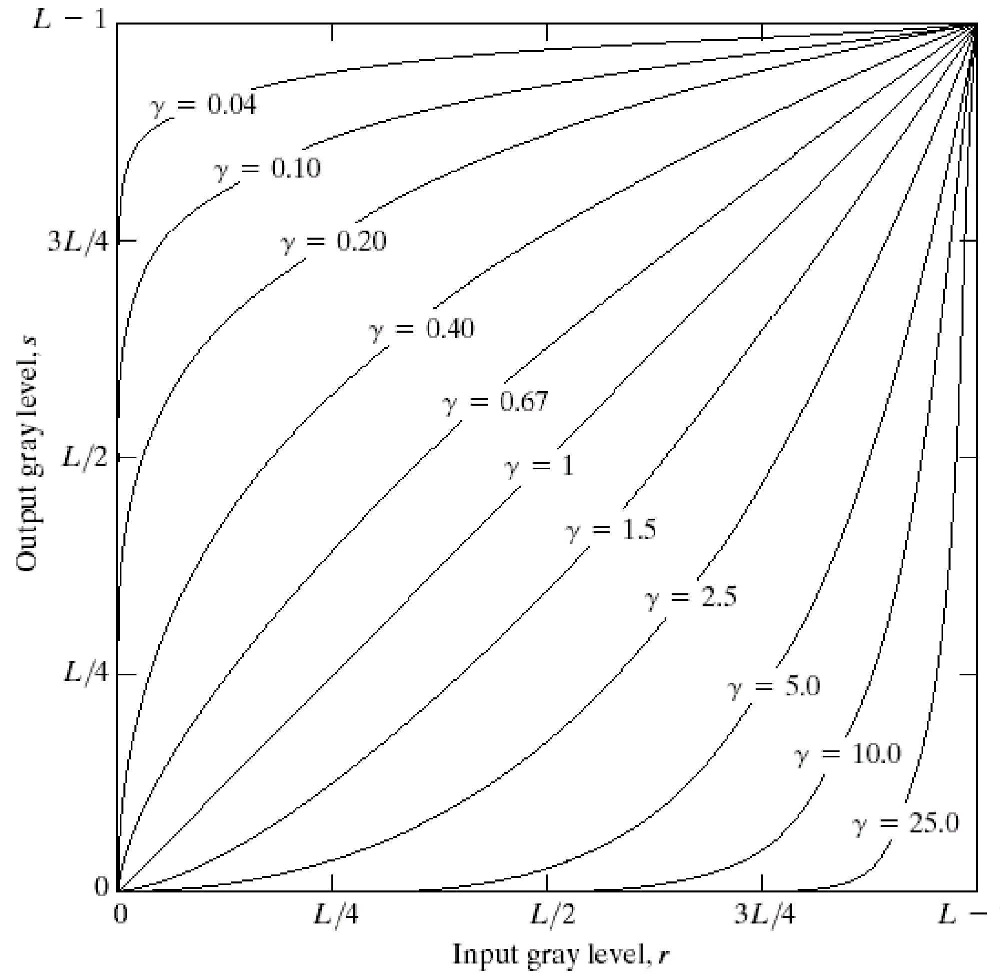
Contrast stretching for medium gray levels



Gray-to-binary image converter

Point Operator – Gamma Correction

- $s = c\, r^{\gamma}$
  - $c = 255^{1-\gamma}$:
    $[0,255] \rightarrow [0,255]$
- $\gamma < 1$:
  - expand dark levels and compress bright levels
- $\gamma > 1$:
  - expand bright levels and compress dark levels
- Varying γ controls the amount of expansion and compression



Point Operator – Histogram Equalization

- Histograms are the basis for numerous spatial domain image processing techniques
  - Rough estimate of probability distribution of gray levels
  - Simple to compute

- Histogram

$$h(r_k) = n_k$$

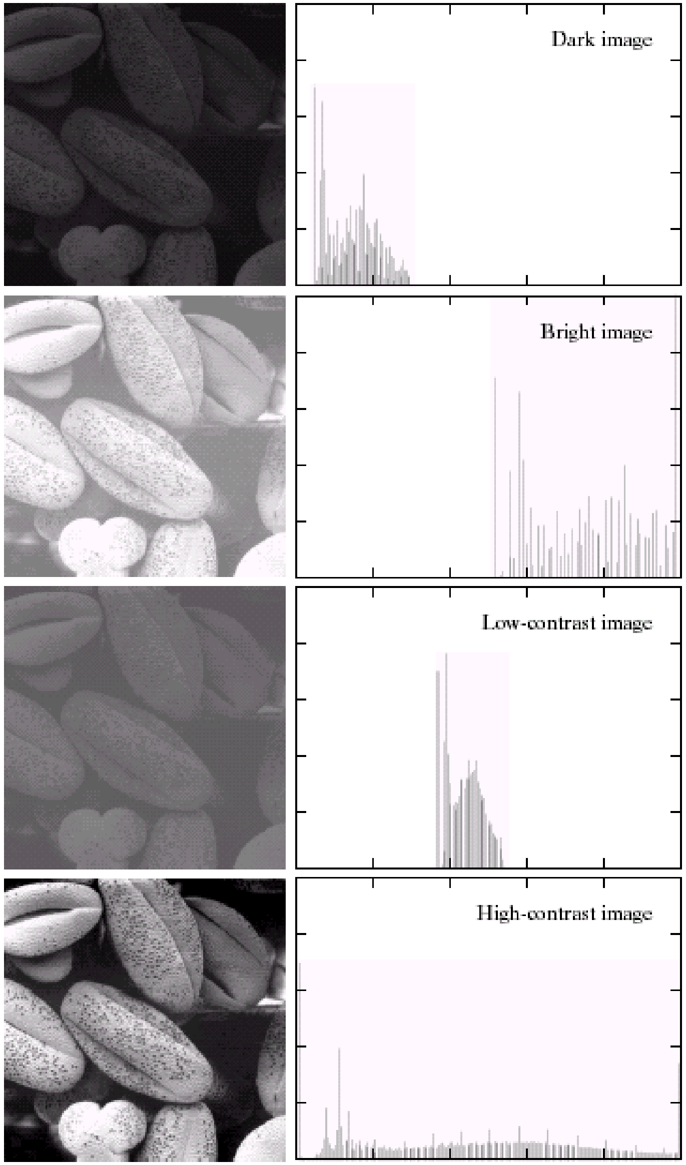
  - r_k: k-th gray level
  - n_k: the number of pixels in the image having gray level r_k

- Normalized histogram

$$p(r_k) = n_k/n$$

  - n: the total number of pixels
  - $\sum_k p(r_k) = 1$

Point Operator – Histogram Equalization



- In general, the uniform distribution of gray levels is desirable
  - high contrast
  - a great deal of details
  - high dynamic range

Point Operator – Histogram Equalization

- Example: An image of 128 pixels. There are 8 gray levels only.
  - Note that each gray level should have 16 pixels in the output histogram

r_k	0	1	2	3	4	5	6	7
n_k	1	7	21	35	35	21	7	1
$\sum n_k$	1	8	29	64	99	120	127	128
$T(r_k)$	0	0	1	3	6	7	7	7

  - Ideally, starting from the smallest gray level,
    - the first 16 pixels should be assigned gray level 0 (0, 1 => 0)
    - 32 pixels => gray level 0 or 1 (0, 1, 2 => 0, 1)
    - 48 pixels => gray level 0, 1, or 2 Skip
    - 64 pixels => gray level 0, 1, 2, 3 (0, 1, 2, 3 => 0, 1, 2, 3)
    - 80 pixels => gray level 0, 1, 2, 3, 4 Skip
    - 96 pixels => gray level 0, 1, 2, 3, 4, 5 Skip
    - 112 pixels => gray level 0, 1, 2, 3, 4, 5, 6 (0, 1, 2, 3, 4 => 0, 1, 2, 3, 4, 5, 6)
    - 128 pixels => gray level 0, 1, 2, 3, 4, 5, 6, 7 (0, 1, 2, 3, 4, 5, 6, 7 => 0, 1, 2, 3, 4, 5, 6, 7)

Point Operator – Histogram Equalization

- Example: An image of 128 pixels. There are 8 gray levels only.
  - Note that each gray level should have 16 pixels in the output histogram
  - More sophisticated equalization

r_k	0	1	2	3	4	5	6	7
n_k	1	7	21	35	35	21	7	1
$\sum n_k$	1	8	29	64	99	120	127	128
$T(r_k)$	0	0	0: 8 pixels 1: 13 pixels	1: 3 pixels 2: 16 pixels 3: 16 pixels				

0	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	→	0
2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	→	1
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	→	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	→	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	→	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	→	5
4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	→	6
5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	7	→	7

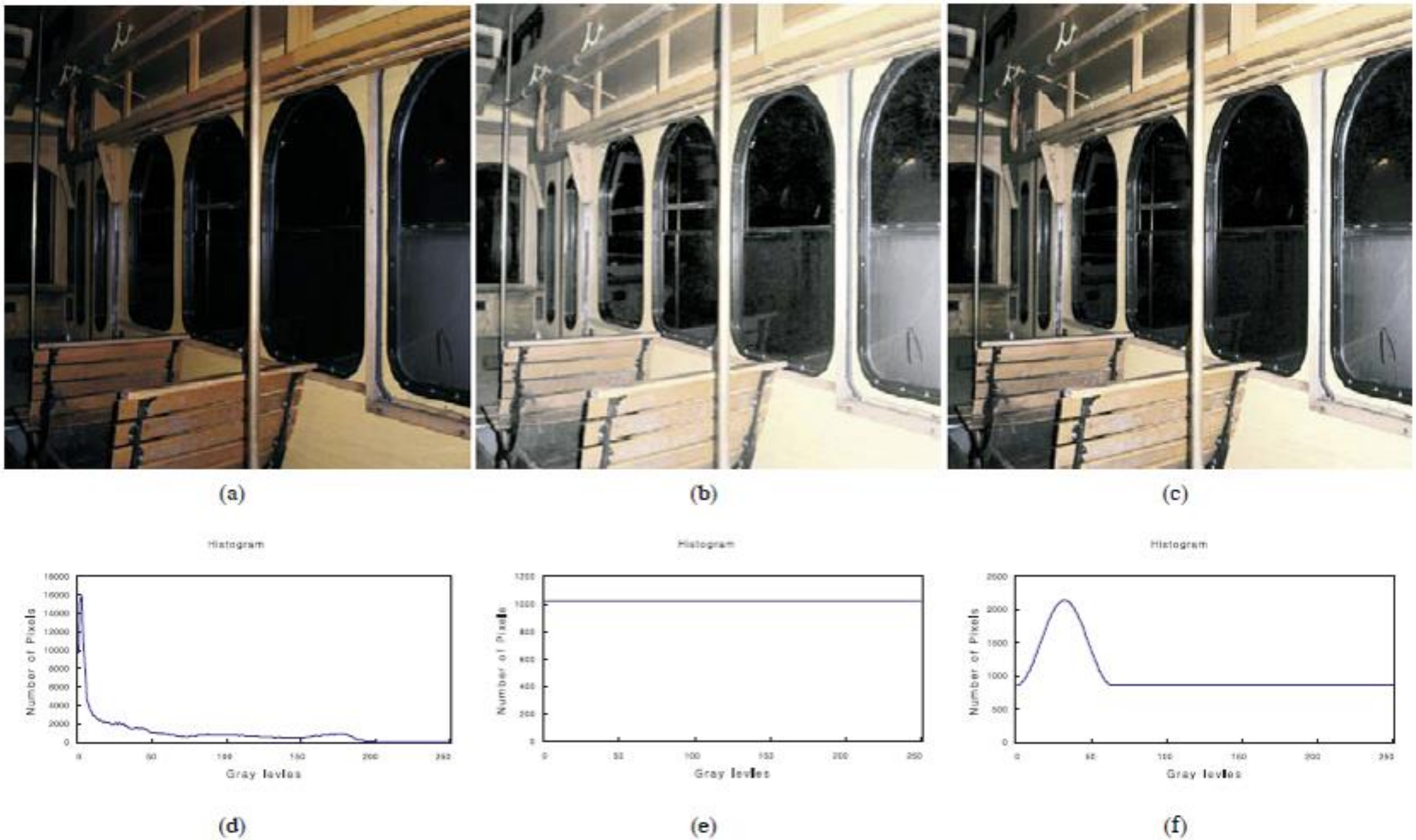


Fig. 2. SHE and SHS: (a) original image, (b) output of SHE, and (c) output of SHS. (d), (e), and (f) are the histograms of (a), (b), and (c), respectively.

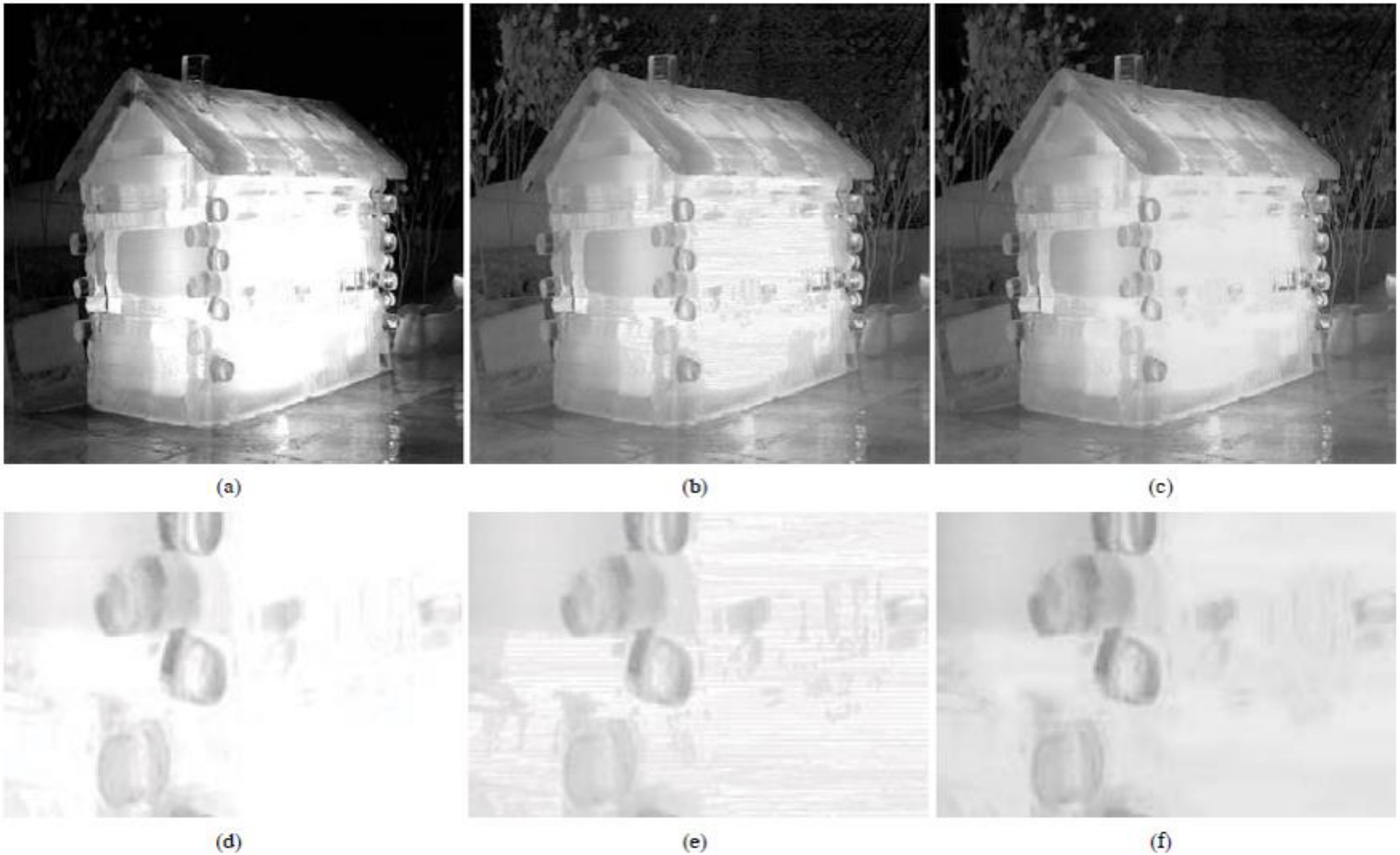
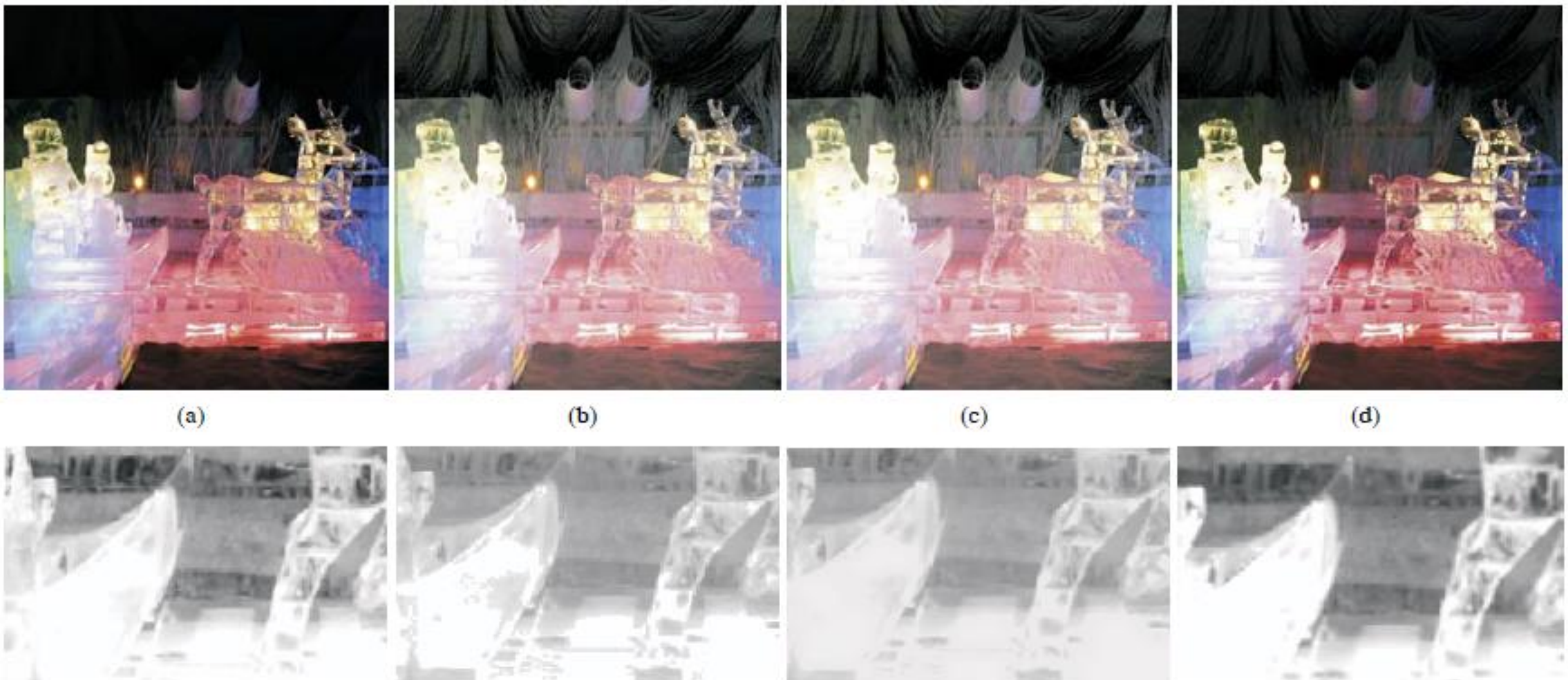


Fig. 3. (a) Original image, (b) output of SHE, and (c) output of SHE + POCS. (d), (e), and (f) are enlarged parts of (a), (b), and (c), respectively.



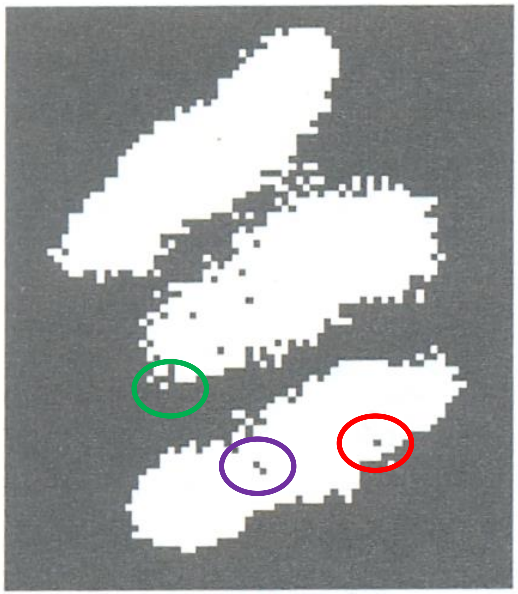
(a) (b) (c) (d)

(e) (f) (g) (h)

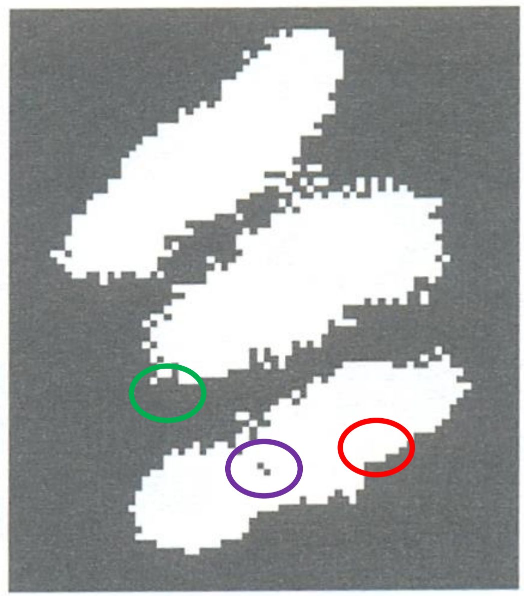
Fig. 4. Comparison of the proposed algorithm with the conventional histogram equalization method in [1]: (a) the original image SANTA, (b) the conventional histogram equalization method, (c) the proposed SHE + POCS algorithm, and (d) the proposed SHS + POCS algorithm. (e), (f), (g), and (h) are enlarged parts of (a), (b), (c), and (d), respectively.

Removal of Small Image Regions

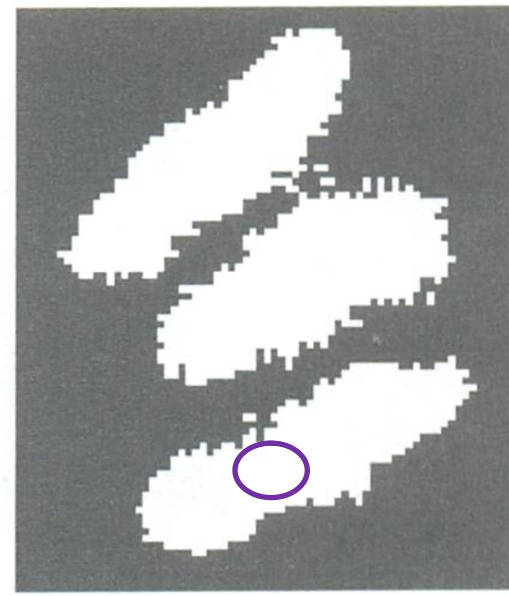
- Removal of Salt-and-Pepper Noise



Input



8-neighbor decision



4-neighbor decision

X	X	X
X	L	X
X	X	X

⇒

X	X	X
X	X	X
X	X	X

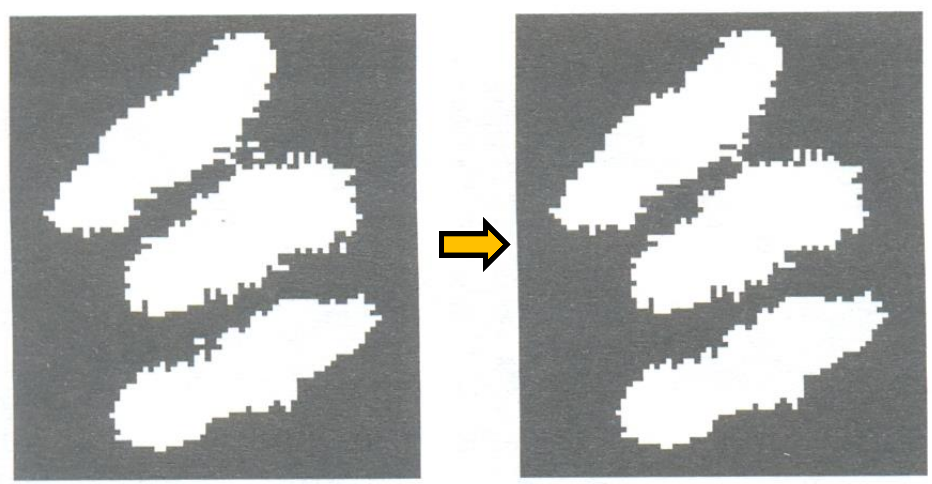
	X	
X	L	X
	X	

⇒

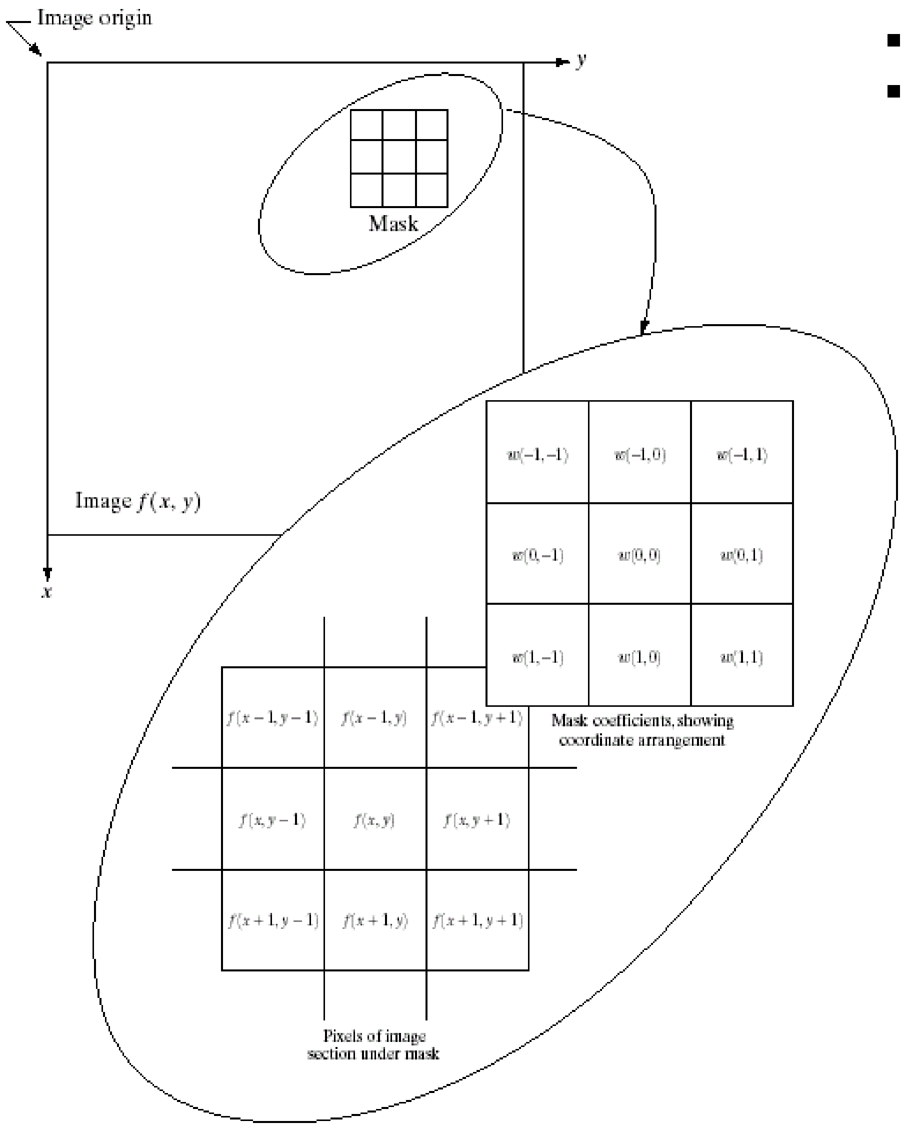
	X	
X	X	X
	X	

Removal of Small Image Regions

- Removal of Small Components
  - Count the number of pixels in a component. If it is less than a threshold, remove the component.
  - ex) Threshold 12



Masking (Linear Filtering)



- Mask is moved from pixel to pixel
- At each location, the mask coefficients are multiplied by the corresponding pixel values, and then summed up

$$
\begin{aligned}
g(x,y) = {} & w(-1,-1)f(x-1,y-1) \\
& + w(-1,0)f(x-1,y) + \ldots \\
& + w(1,1)f(x+1,y+1)
\end{aligned}
$$

Masking with

a	b	c
d	e	f
g	h	i

=

Convolving with

i	h	g
f	e	d
c	b	a

Masking (Linear Filtering)

- Masking with a mask w of size $(2a+1)\times(2b+1)$

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

- Convolving with a filter h of size $(2a+1)\times(2b+1)$

$$g'(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} h(s,t)f(x-s,y-t)$$

- Note that $g(x,y)=g'(x,y)$ if $w(s,t)=h(-s,-t)$
- For masking, we use the following notation also

$$R=\sum_{i=1}^{k} w_i z_i = w_1 z_1 + w_2 z_2 + \ldots + w_k z_k$$

where w_i's are masking coefficients and z_i's are pixel values.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Masking (Linear Filtering)

- Boundary problem
  1. Limit the excursion of the center of the mask, so that the mask is fully contained within the image
     - Output image is smaller than input image
  2. Extrapolate the input image sufficiently, so that the mask can be applied near the boundaries also.
     - Zero padding
     - Repetition
     - Mirroring
     - etc

0	0	0	0	0
0	0	0	0	0
0	0	a	b	c
0	0	d	e	f
0	0	g	h	i

a	a	a	b	c
a	a	a	b	c
a	a	a	b	c
d	d	d	e	f
g	g	g	h	i

a	a	d	e	f
a	a	a	b	c
b	a	a	b	c
e	d	d	e	f
h	g	g	h	i

Smoothing Filters

- Averaging filter (**box filter**) and weighted averaging filter

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Blends with adjacent pixel values
- Blurring
  - Removal of small details before large object extraction
  - Bridging of small gaps in lines or curves
  - Reduction of sharp transitions in gray levels
    - Advantage: noise reduction
    - Disadvantage: edge blurring

Smoothing Filters

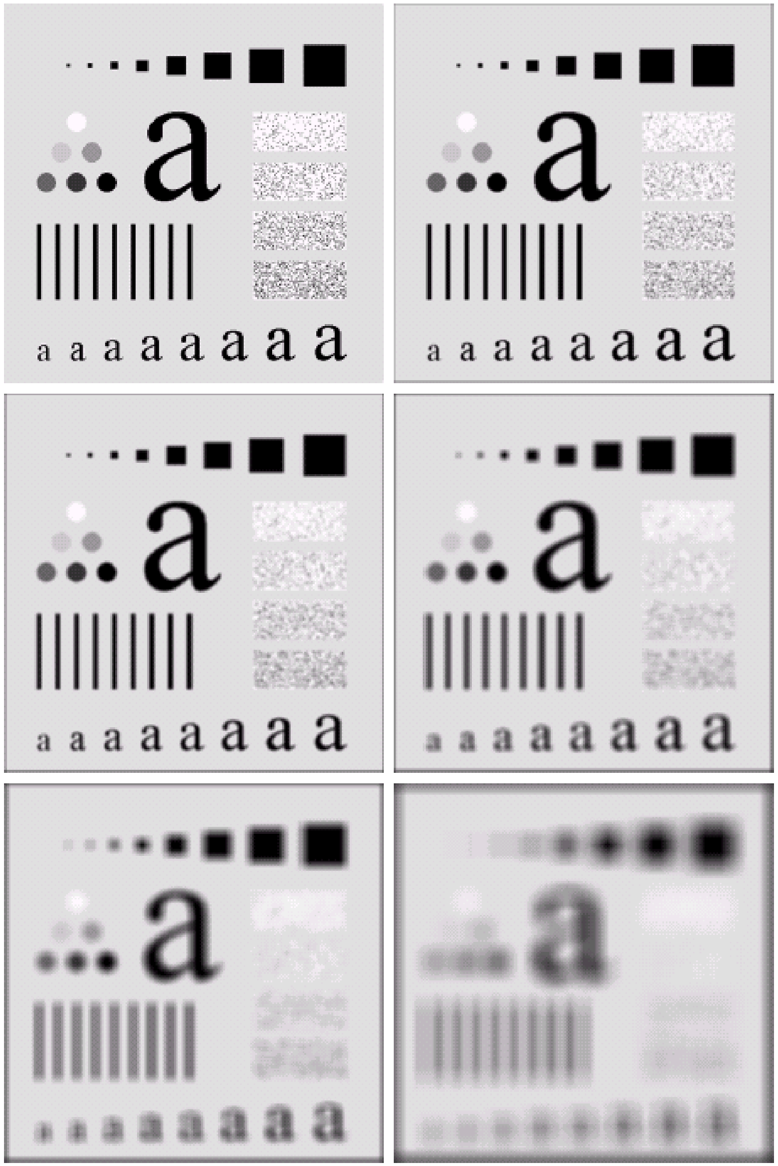
- Gaussian filter

$$g(x,y) = c\sum_{s}\sum_{t} w(s,t)\, f(x+s, y+t)$$

where

$$w(s,t) = e^{-\frac{(s^2+t^2)}{2\sigma^2}}$$

Smoothing Filters



3

5

9

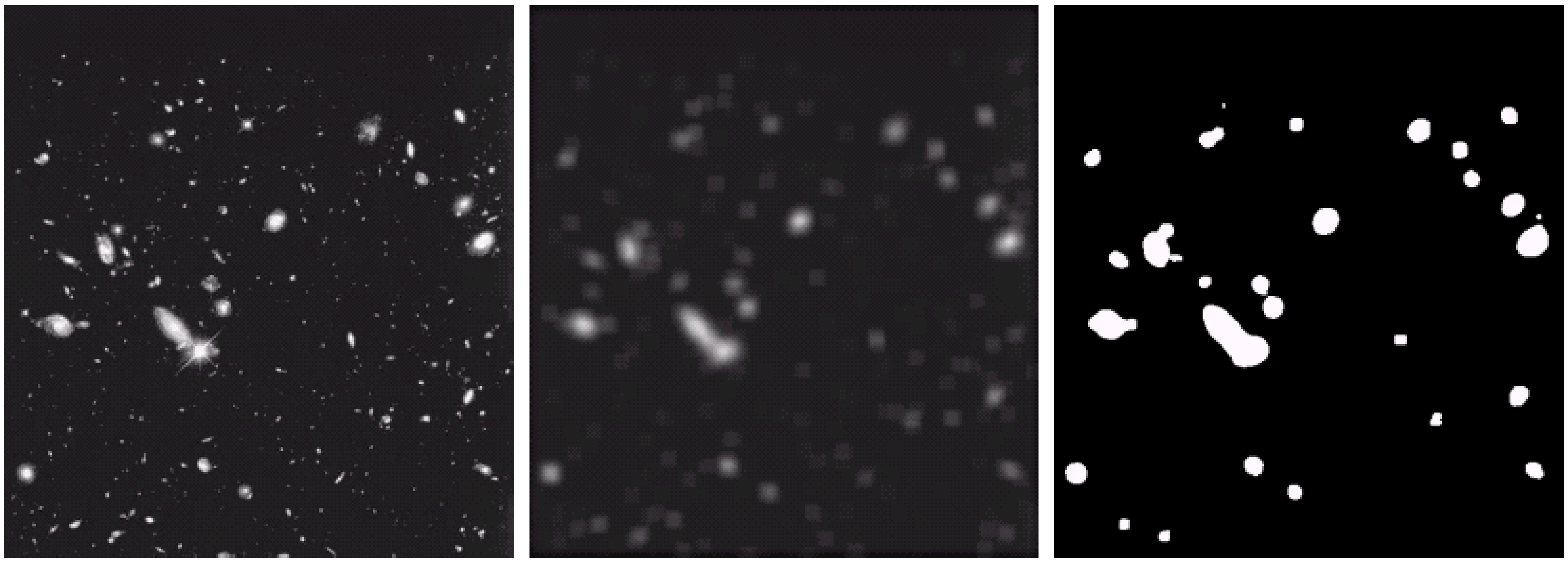
15

35

- Losing edges
- Reducing noises
- Removing small objects

Smoothing Filters

- Finding objects of interest



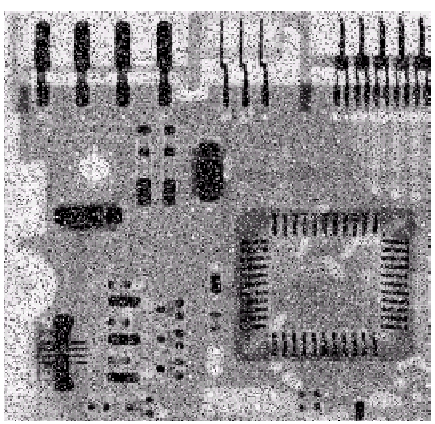
a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

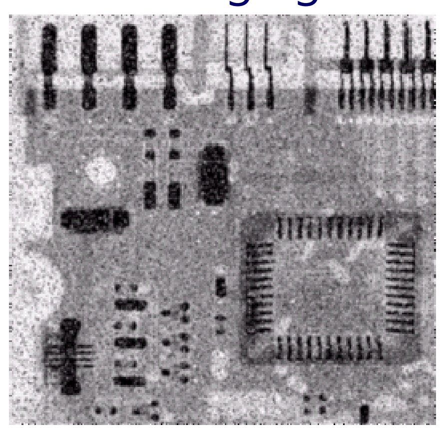
Order-Statistics Filter

- Sort the gray levels of the neighborhood
  - (0, 1, 2, 2, 3, 4, 5, 6, 6)
    min median max
- Min filter
  - Replace the center pixel with the minimum gray level (0)
- Max filter
  - Replace the center pixel with the maximum gray level (6)
- Median filter
  - Replace the center pixel with the median (3)
  - Excellent suppression of salt-and-pepper noises without blurring

6	4	6
2	1	3
2	5	0



3x3 averaging filter



3x3 median filter