III. Intensity Transformations and Spatial Filtering

Dr. Mohammad Abu Yousuf
Associate Professor
IIT, JU

III. Intensity Transformations and Spatial Filtering

1. Preview
2. Some basic intensity transformation functions
3. Histogram processing
4. Fundamentals of spatial filtering
5. Smoothing spatial filters
6. Sharpening spatial filters
7. Combining spatial enhancement methods
8. Using fuzzy techniques for intensity transformations and spatial filtering

Preview

- The principal objective of enhancement
  - to process an image so that the result is **more suitable** than the original image for a **specific application**.

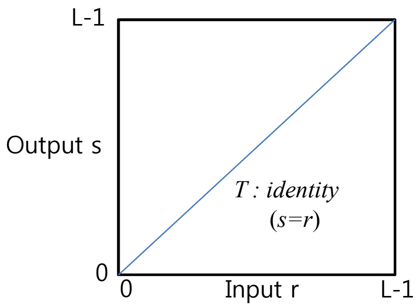
    ex) A method that is quite useful for enhancing X-ray images may not necessarily be the best approach for enhancing pictures of Mars transmitted by a space probe.

- The term **spatial domain** : the image plane itself
  - Based on direct manipulation of pixels in an image.
  - *Frequency domain* processing techniques : based on the Fourier transform of an image.

- There is **no general theory** of image enhancement.
  - When an image is processed for visual interpretation, the viewer is the ultimate judge of how well a particular method works.

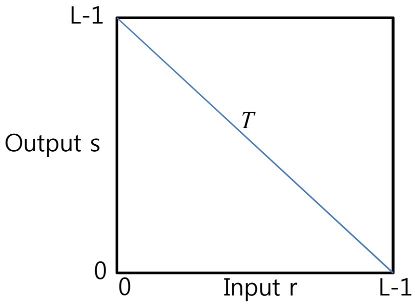
Intensity Transformation Functions

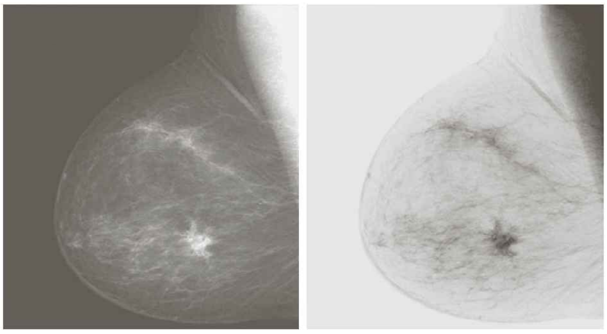
- Single point processing
- $s=T(r)$ r:input pixel value, s:output value
- Look-up table technique
  - L LUT (L : # of gray levels, 2^N)
  - s=LUT[r]
  - Filling LUT with T



Some Basic Intensity Transformation Functions
- Image Negatives

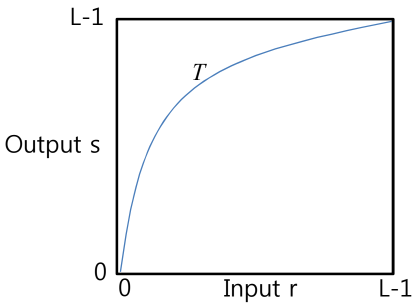
- s=L-1-r
- (0,1,2,…254,255)→(255,254,…,1,0)

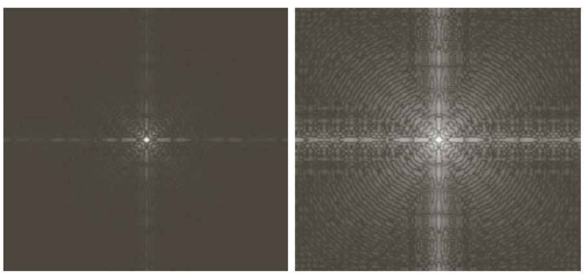




Log Transformations

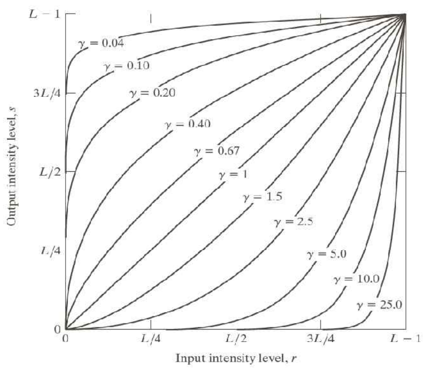
- s=c log(1+r)
- Compresses the dynamic range of the image
  ex) to visualize Fourier spectra
- to expand the values of dark pixels in an image while compressing the higher-level values. (and vice versa)





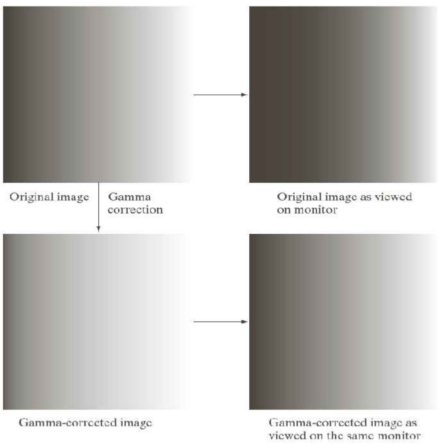
Power-Law (Gamma) Transformation

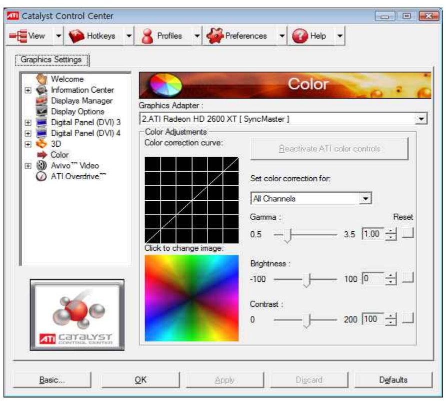
- $s = c\, r^{\gamma}$
- Compresses the dynamic range of the image
  - the exponent is called gamma (γ)
  - gamma correction
  - i.e. CRT monitors have an intensity to voltage response which is a power-law with gamma 1.8-2.5
- As in the case of the log transformation, power-law curves with fractional values of y map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels



Gamma Correction

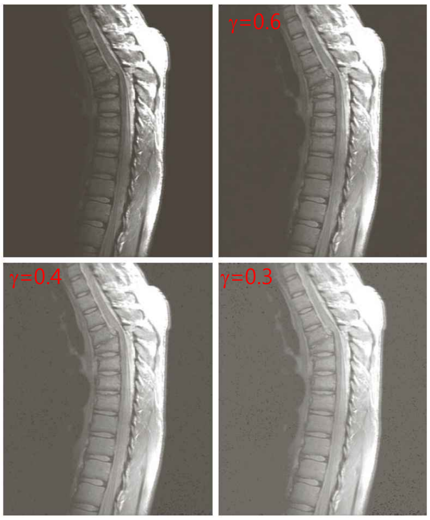
- Useful to compensate for the non-linear response of monitors



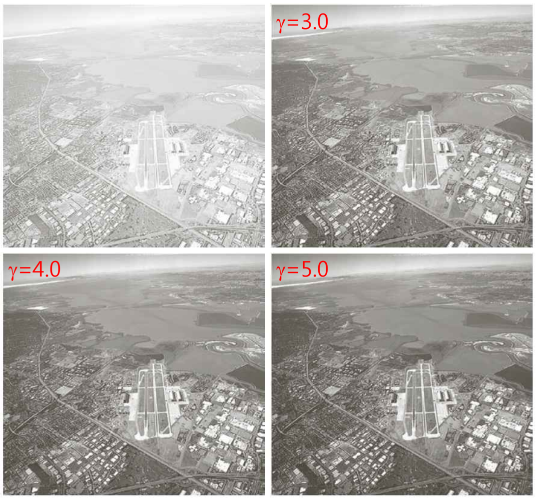


VGA control panel

ex) MRI image (surface coil) and aerial image (washed-out)



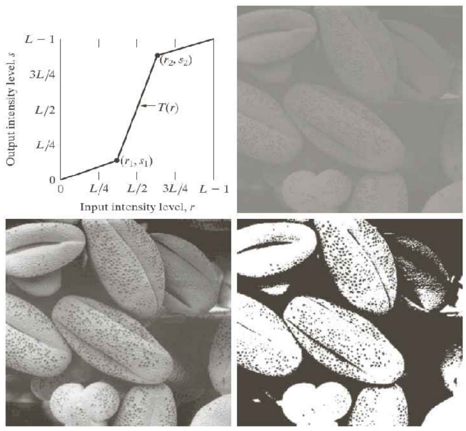
reduce contrast ($\gamma<1$)



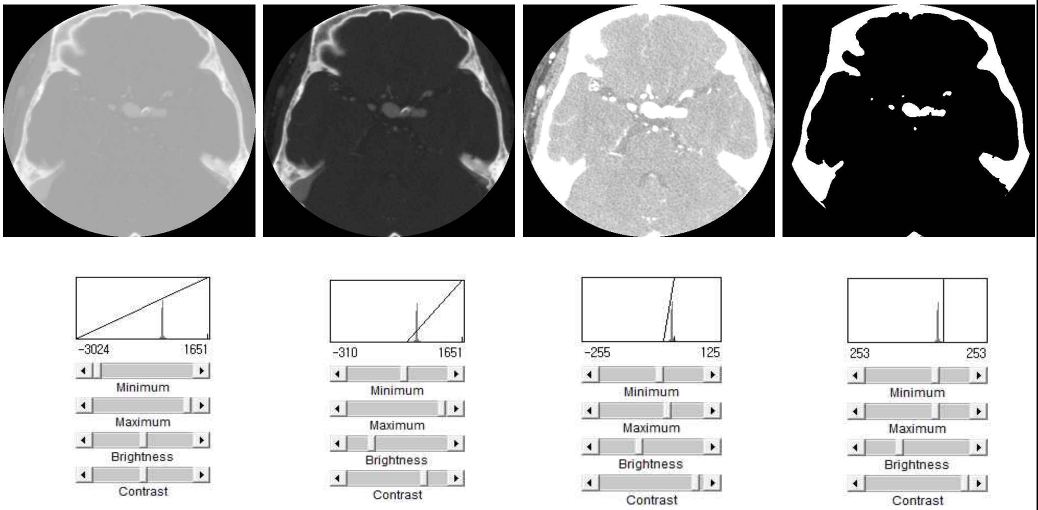
enhance contrast ($\gamma>1$)

Piecewise-Linear Transformations - Contrast Stretching

- Adjust brightness and contrast
- Expand the range of intensity levels

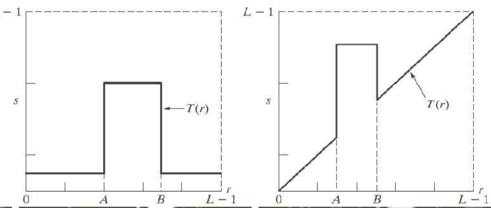


Example : CT image (Window/Level)



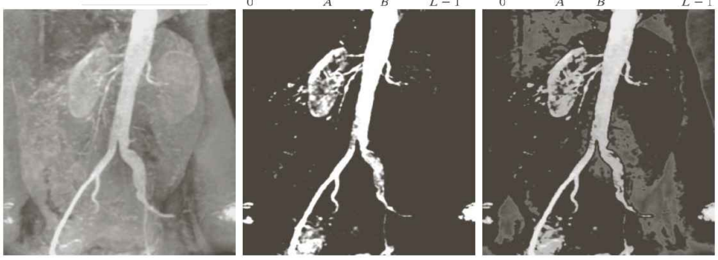
Intensity-Level Slicing

- Highlighting a specific range of intensities



a b

FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.



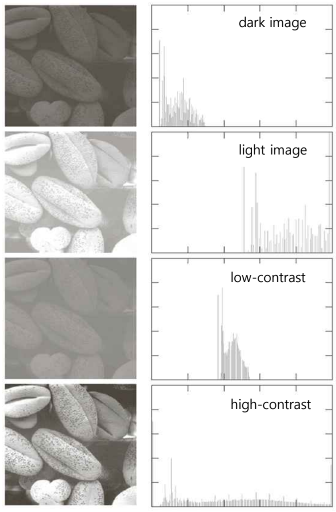
a b c

FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

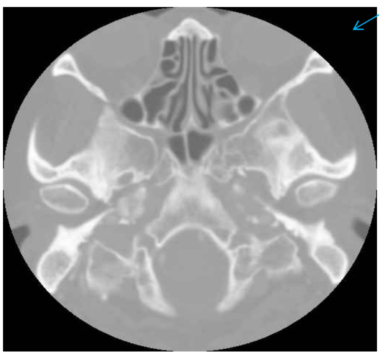
Histogram Processing

- The number of pixels having the same gray level
  → intensity distribution
- Useful for
  – gathering image statistics
  – spatial domain (real-time) processing
- How to get the histogram

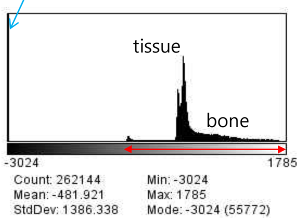
```
for (i=0; i<L; i++) hist[i]=0;
for (i=0; i<N; i++) hist[r]++;
```



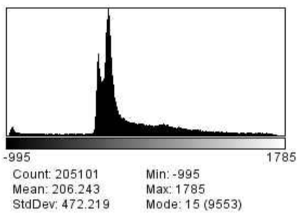
Example : CT image



The meaningless background should be excluded.

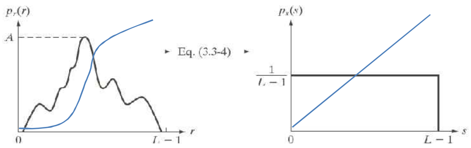


- Not evenly distributed
- The pixels of the meaningless background are also included.



Histogram Equalization

- $s=T(r)$
- PDF : $p_r(r)$, $p_s(s)$



a b

FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

- Cumulative distribution function

$$s=T(r)=(L-1)\int_0^r p_r(w)dw$$

$$\frac{ds}{dr}=\frac{dT(s)}{dr}=(L-1)\frac{d}{dr}[\int_0^r p_r(w)dw]=(L-1)p_r(r)$$

$$p_s(s)=p_r(r)\left|\frac{dr}{ds}\right|=p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right|=\frac{1}{L-1}$$

$p_s(s)$ is always a uniform PDF.

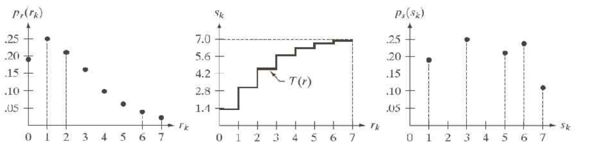
- For discrete values, PDF → normalized histogram (divided by the total number of pixels)

$$s=T(r)=(L-1)\sum_{j=0}^{r}p_r(j)$$ → **Histogram Equalization**

Example

Intensity distribution and histogram for a 3-bit 64x64 image

r	n	$p_r(r)$=n/(64x64)	s
0	790	0.19	1
1	1023	0.25	3
2	850	0.21	5
3	656	0.16	6
4	329	0.08	6
5	245	0.06	7
6	122	0.03	7
7	81	0.02	7



a b c

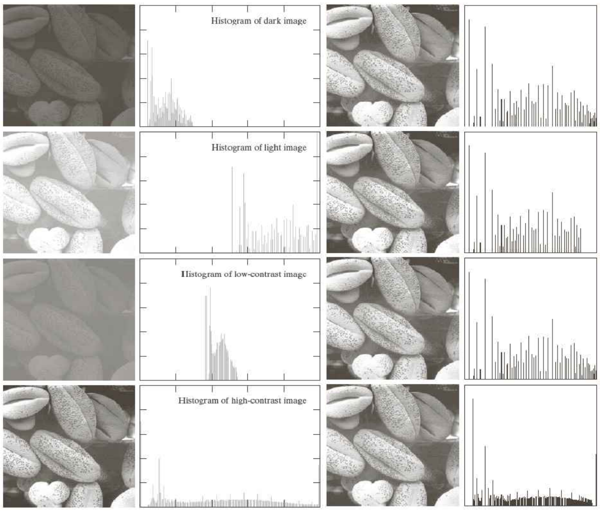
FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

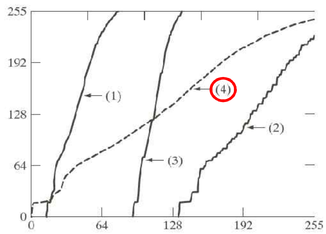
$$s_0=T(0)=7\sum_{j=0}^{0}p_r(j)=7p_r(0)=1.33\approx 1$$

$$s_1=T(1)=7\sum_{j=0}^{1}p_r(j)=7\{p_r(0)+p_r(1)\}=3.08\approx 3$$

s_2=4.55≈5, s_3=5.67≈6, s_4=6.23≈6, s_5=6.65≈7, s_6=6.86≈7, s_7=7.00≈7

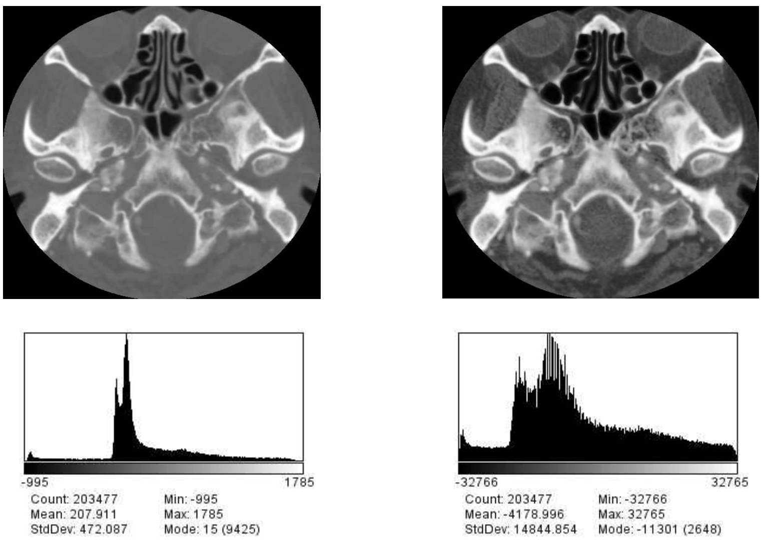
Examples





Each transformation function for histogram equalization

Example : CT image



Histogram Matching (Specification)

- Given histogram $p_z(z)$
- $s = T(r) = (L-1)\int_0^r p_r(w)dw$: histogram-equalized

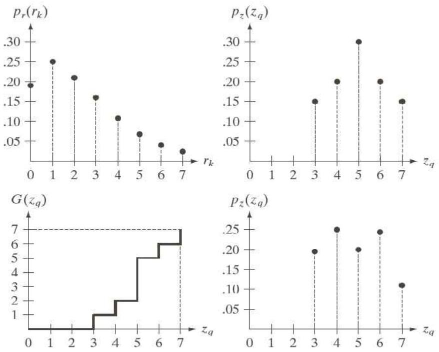
  $s = G(z) = (L-1)\int_0^z p_z(w)dw$

  $z = G^{-1}[T(r)] = G^{-1}(s)$

r	n	$p_r(r)$	s
0	790	0.19	1
1	1023	0.25	3
2	850	0.21	5
3	656	0.16	6
4	329	0.08	6
5	245	0.06	7
6	122	0.03	7
7	81	0.02	7

z	$p_z(z)$	$s=G(z)$
0	0.00	0
1	0.00	0
2	0.00	0
3	0.15	1
4	0.20	2
5	0.30	5
6	0.20	6
7	0.15	7

r	s	z
0	1	3
1	3	4
2	5	5
3	6	6
4	6	6
5	7	7
6	7	7
7	7	7



Using Histogram Statistics

- Mean : average gray level

  $$m = \sum_{i=0}^{L-1} r_i p(r_i)$$

  $$m = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

- Nth moment of r

  $$\mu_n(r) = \sum_{i=0}^{L-1}(r_i - m)^n p(r_i)$$

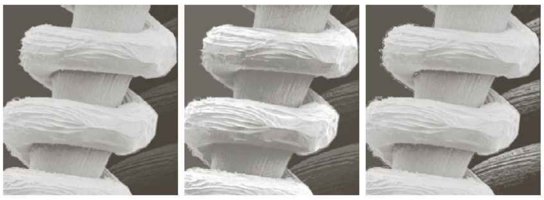
- Variance : average contrast, texturing, activity

  $$\mu_2(r) = \sigma_r^2 = \sum_{i=0}^{L-1}(r_i - m)^2 p(r_i)$$

  $$\sigma^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - m]^2$$

- Local statistics
  - Good measures for spatially-adaptive processing

Example of Local Processing



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

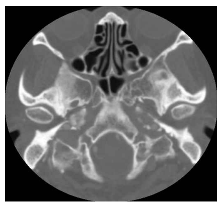
- Low and high contrast areas
- Enhance low contrast areas
- Local region ← 3x3 window

$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{if } m_{S_{xy}} \le k_0 m_G \text{ AND } k_1\sigma_G \le \sigma_{S_{xy}} \le k_2\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

Homework #2

- Enhance the contrast of the CT image
  1) Apply global histogram equalization.
  2) Apply the same algorithm as the example.
     - Choose appropriate E, k_0, k_1, k_2
  3) Compare both results.

☞ Do not include background pixels out of the circle



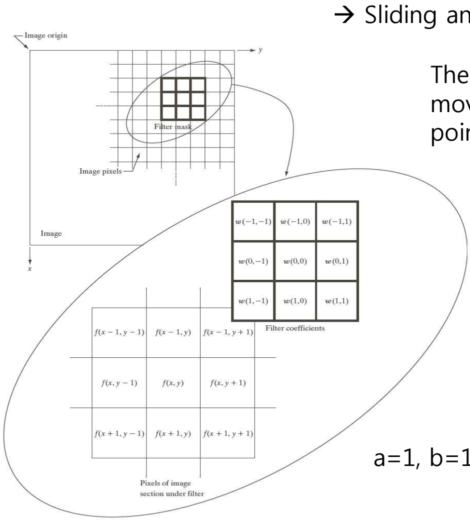
Fundamentals of Spatial Filtering

- Spatial filtering ⇔ Frequency domain filtering
- filtering operations that are performed directly on the pixels of an image
- Operation between a neighborhood
- Define a 2D filter with the following properties
  - linear
  - Space-invariant
  - finite impulse response (FIR) ← Convolution

Linear Spatial Filter

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

→ Sliding and sum-of-product



The process consists simply of moving the filter mask from point to point in an image.

a=1, b=1 for 3x3 filter mask

Linear Convolution

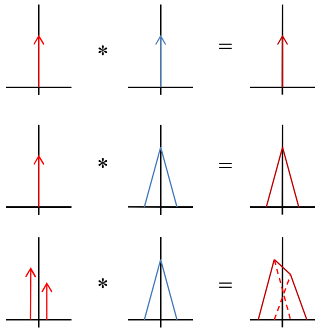
- Space-invariant linear system

input $f(x,y)$ → Impulse response $h(x,y)$ → output $g(x,y)=h(x,y)*f(x,y)$

- $h(x,y)$: impulse response, point spread function → **resolution of the system**
  ex) if $h(x,y) = \delta(x,y)$, $g(x,y)=f(x,y)$
- Convolution
  - Replace each pixel with r·$h(x,y)$
  - Sliding and sum-of-product
- m x n convolution (m=2a+1, n=2b+1)

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

$w(x,y)$: m × n convolution filter(mask, kernel)
→ filter design



Remarks

- Filtering is linear because each output pixel value is computed as a linear combination of a set of input pixel values.
- Filtering is *space invariant* because filter coefficients are independent on the point of application *(x, y).*
- The output has size (M+m-1)×(N+n-1) samples, but practically (M×N)
- The sum of filter coefficients?
  – Normalization
  – Output : overflow, underflow
- Computational complexity ∝ M×N×m×n
  – why 3×3 convolution is popular
- The borders of the image?
  – exclude borders → effective output size (M-a)×(N-b)
  – add padding to the input image with average gray level
  – bypass (replace with input pixels)
  – fill zeros

Smoothing Spatial Filters

- Blurring to remove small details and extract a large object
- Noise reduction
- Low pass filter – remove high frequency components
- The most simple smoothing filter : Averaging filter

1/9 ×

1	1	1
1	1	1
1	1	1

1/25 ×

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

- Weighted average

1/16 ×

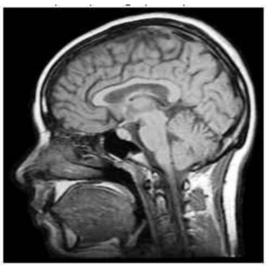
1	2	1
2	4	2
1	2	1

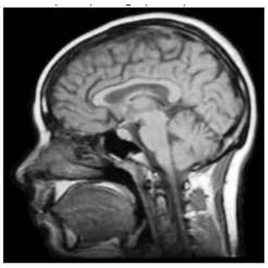
Weighted Average

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & K & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \frac{1}{K+8}$$

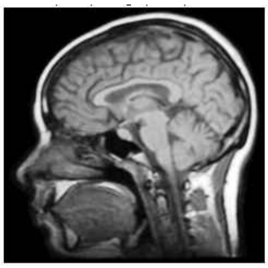
- 9 pixel averaging (K=1)

- K↑ → SNR↓, Resolution↑





K=4



K=1

Window Size Effect of Average Filter

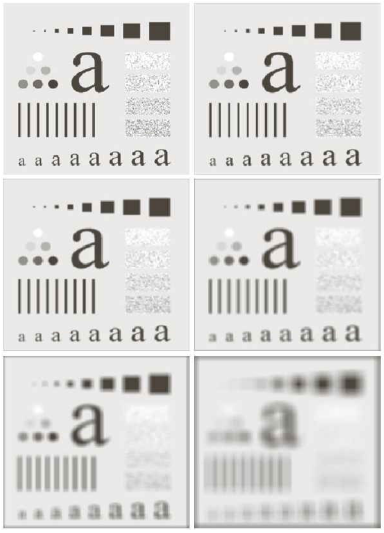
3x3

5x5

9x9

15x15

35x35



Removal of small details

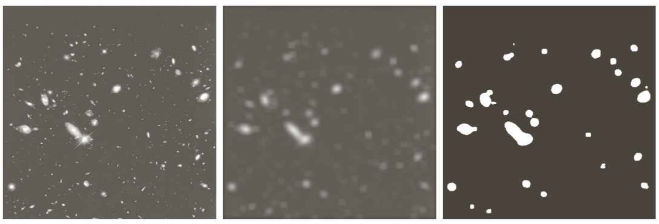


Image of size 528x485 (Hubble Space Telescope)

15 x 15 averaged

Result of thresholding

Order-Statistic (Nonlinear) Filters

- Based on ordering the pixels
- ex) maximum, minimum, median filters
- Nonlinear filter
- Median filter
  - Quite popular
  - Excellent noise-reduction capabilities with less blurring (*edge-preserving filter*)
  - In particular, effective to remove impulse noise (salt-and-pepper)
  - Remove defective pixels

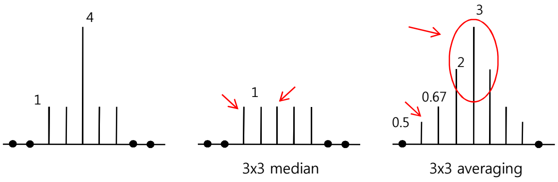
5, 4, 5.5, 100, 6, 7, 8, 7, 7.5

↓

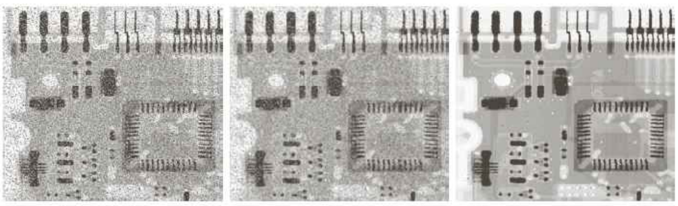
4, 5, 5.5, 6, 7, 7, 7.5, 8, 100

Median = 7

Average = 16.7

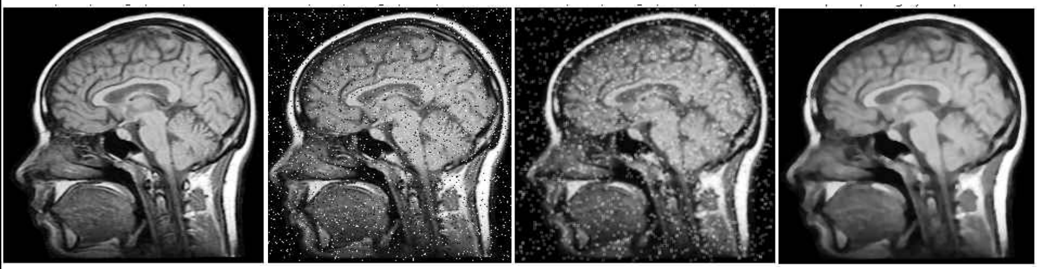


Examples



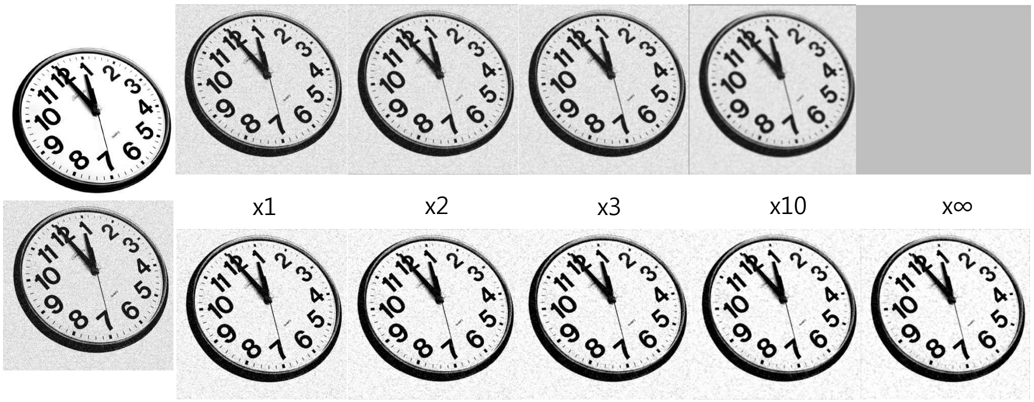
a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



Salt and pepper noise | averaging | median

Blurring Filter and Median Filter



Sharpening Spatial Filters

- Highlighting intensity transitions
  - Enhance intensity discontinuity (edges and noise)
  - Deemphasize areas with slowly varying intensities
- Edge detection

- Sharpening ← spatial differentiation

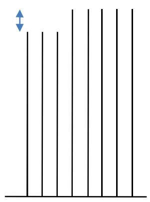
  (Averaging ← integration)

- Digital derivatives → differences
  - The 1st-order derivative

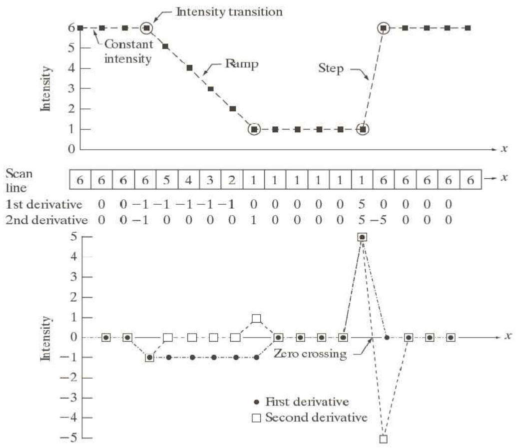
    $\partial f(x,y)/\partial x = f(x+1,y) - f(x,y)$

  - The 2nd-order derivative

    $\partial^2 f(x,y)/\partial x^2 = \{f(x+1,y)-f(x,y)\}-\{f(x,y)-f(x-1,y)\} = f(x+1,y)+f(x-1,y)-2f(x,y)$



The 1st and 2nd Order Derivatives



- 1st-order derivatives → thicker edges

- 2nd-order derivatives → a stronger response to fine detail → better for enhancement

-1st-order derivatives → a stronger response to a gray-level step

-2nd-order derivatives → a double response at step changes in gray level

The Laplacian – The 2nd Derivative

- $\nabla^2 f = \partial^2 f(x,y)/\partial x^2 + \partial^2 f(x,y)/\partial y^2$
  $\partial^2 f(x,y)/\partial x^2 = f(x+1,y)+f(x-1,y)-2f(x,y)$
  $\partial^2 f(x,y)/\partial y^2 = f(x,y+1)+f(x,y-1)-2f(x,y)$
  $\nabla^2 f = f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)-4f(x,y)$

0	1	0
1	−4	1
0	1	0

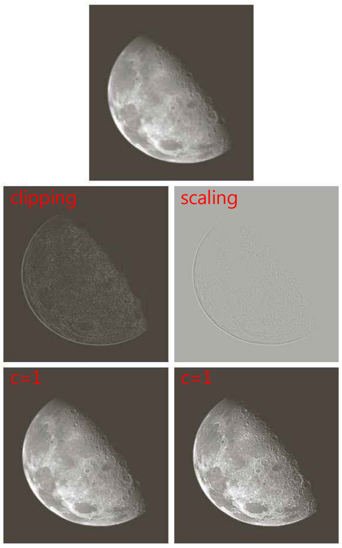
1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

- Laplacian Sharpening : $g(x,y)=f(x,y)+c[\nabla^2 f]$
- Remark : Negative value → clipping or scaling

Examples of the Laplacian Operator



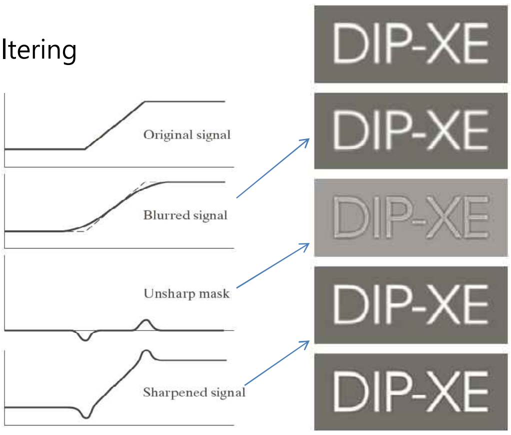
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$g(x,y)=f(x,y)+\nabla^2 f=(1+\nabla^2)f(x,y)$

Unsharp Masking and Highboost Filtering

- Unsharp masking
  1) Blur
  2) Subtract the blurred image from the original → unsharp mask
     $$g_{mask}(x,y)=f(x,y)-\bar{f}(x,y)$$
  3) Add the mask to the original
     $$g(x,y)=f(x,y)+k\cdot g_{mask}(x,y)$$
- $k>1$: Highboost filtering



5x5 Gaussian smoothing filter (σ=3)

k=1

Highboost (k=4.5)

Gradient – The 1st Derivative

- Gradient

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Gradient image

$$M(x,y) = |\nabla f| = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

- The simplest approximation

$g_x = z_6 - z_5$, $g_y = z_8 - z_5$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Roberts cross-gradient operator

$g_x = z_9 - z_5$, $g_y = z_8 - z_6$

$M(x,y) = [g_x^2 + g_y^2]^{1/2} = |g_x| + |g_y|$

−1	0
0	1

0	−1
1	0

Sobel Operators

- $g_x = f(x,y)*$

−1	0	1
−2	0	2
−1	0	1

$g_y = f(x,y)*$

−1	−2	−1
0	0	0
1	2	1

−1	0	1

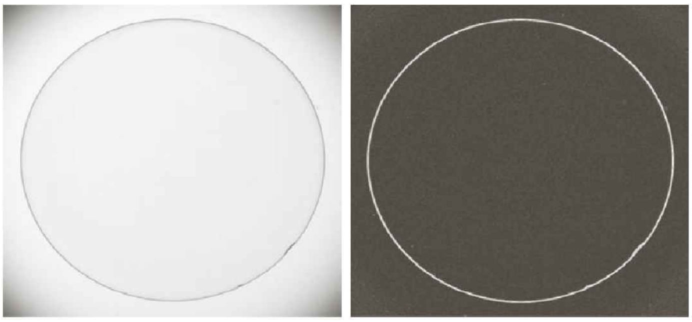
$\longrightarrow$

−1	0	1
−1	0	1
−1	0	1

- $M(x,y) = |g_x| + |g_y|$

- Some smoothing by giving more importance to the center point
- Sum of coefficients = 0 → 0 in flat areas of constant intensity
- The most popular edge-detector
- Enhance defects and eliminate slowly changing background features → preprocessing for automatic inspection

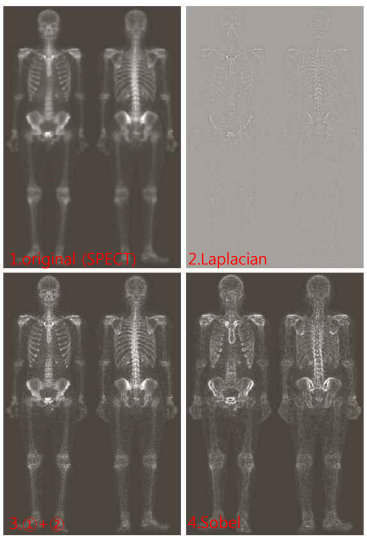
Example

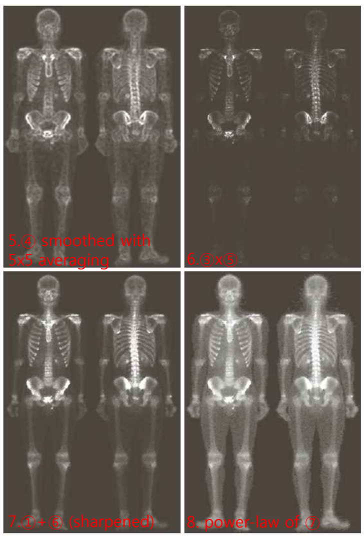


Optical image of contact lens and Sobel gradient
(note defects)

The edge defects are quite visible, but with the added advantage that constant or slowly varying shades of gray have been eliminated thus simplifying considerably the computational task required for automated inspection. → preprocessing step

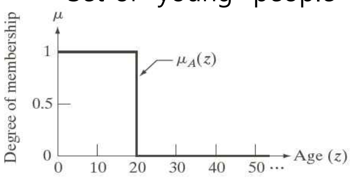
Combining Spatial Enhancement Methods



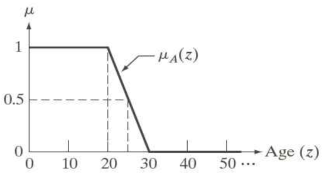


Fuzzy Set

- "Crisp" set : 1 or 0, true or false (bi-valued Boolean logic)
- Membership functions
  - Set of "young" people

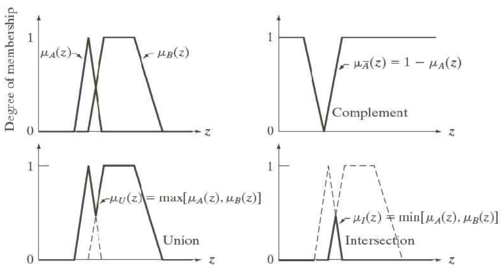


young or not.
How about 20 years and 1 sec?



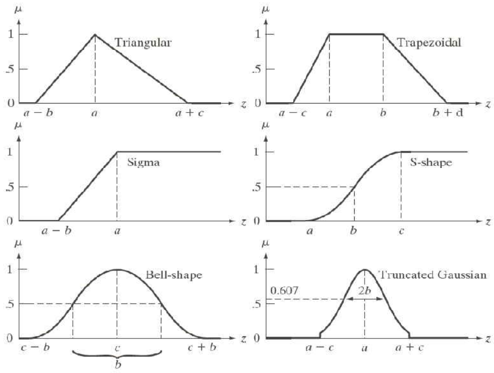
Degrees of "youngness"
young, relatively young, 50% young, not so young, not young
→ "Fuzzy" statements

Several Definitions



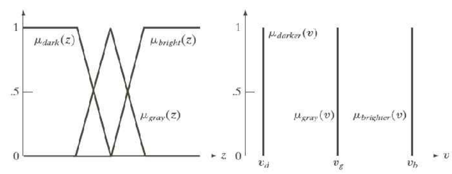
A∪B A∩B

Examples of Membership Functions



Using Fuzzy Sets for Intensity Transformations

- Contrast enhancement
  - IF a pixel is dark, THEN make it darker
  - IF a pixel is gray, THEN make it gray
  - IF a pixel is bright, THEN make it brighter



- Defuzzification (center of gravity) : obtain a crisp output from a fuzzy set

$$v_0 = \frac{\mu_{dark}(z_0) \times v_d + \mu_{gray}(z_0) \times v_g + \mu_{bright}(z_0) \times v_b}{\mu_{dark}(z_0) + \mu_{gray}(z_0) + \mu_{bright}(z_0)}$$

- v_d=0(black), v_g=127(mid gray), v_b=255(white)

Example

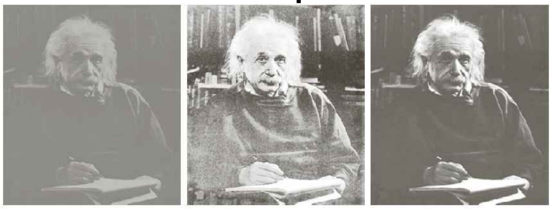


FIGURE 3.54 (a) Low-contrast image. (b) Result of histogram equalization. (c) Result of using fuzzy, rule-based contrast enhancement.

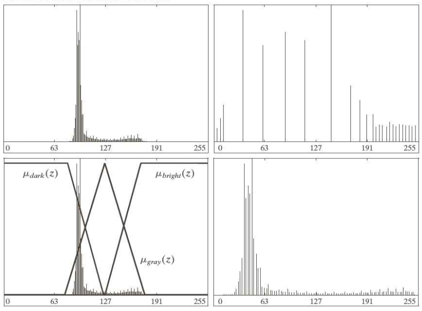


FIGURE 3.55 (a) and (b) Histograms of Figs. 3.54(a) and (b). (c) Input membership functions superimposed on (a). (d) Histogram of Fig. 3.54(c).

Using Fuzzy Sets for Spatial Filtering

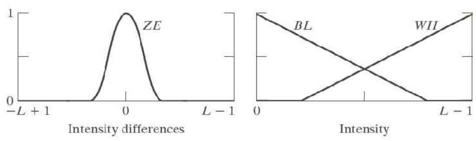
- Boundary extraction
  If a pixel belongs to a uniform region, then make it *white*; else make it *black*
- $d_i = z_i - z_5$
  IF d_2 is *zero* AND d_6 is *zero*, THEN z_5 is *white*
  IF d_6 is *zero* AND d_8 is *zero*, THEN z_5 is *white*
  IF d_8 is *zero* AND d_4 is *zero*, THEN z_5 is *white*
  IF d_4 is *zero* AND d_2 is *zero*, THEN z_5 is *white*
  ELSE z_5 is *black*

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixel neighborhood

d_1	d_2	d_3
d_4	0	d_6
d_7	d_8	d_9

Intensity differences



Example

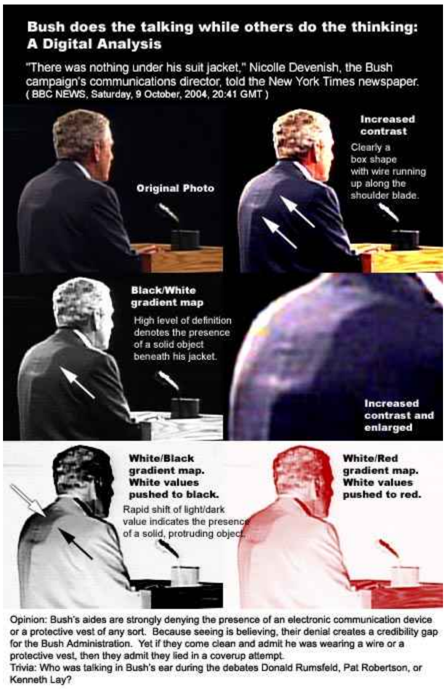
a b c

FIGURE 3.59 (a) CT scan of a human head. (b) Result of fuzzy spatial filtering using the membership functions in Fig. 3.57 and the rules in Fig. 3.58. (c) Result after intensity scaling. The thin black picture borders in (b) and (c) were added for clarity; they are not part of the data. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Does Bush have a device under his Jacket?

*See big bulge with **image enhancement**.*

Bush is wired! Secret earpiece tells him what to say!



Homework #3

- Improve the SNR of the noisy MR image below by use of a 5x5 averaging filter
- Preserve edges as much as possible
  - Design a simple adaptive filter (space-variant)
    - Get an edge image of the 2nd noisy image using the Sobel mask
    - Change the filter coefficient K according to the edge strength
      - K=1 for a flat area (no edge)
      - Larger K for a strong edge
- Compare the result with the original images with/without noise

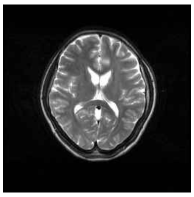


Image without noise (for comparison)

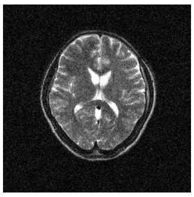
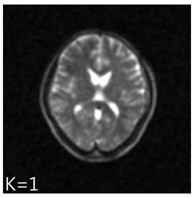


Image with noise



$$\frac{1}{(K+24)} \times \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & K & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$