Intensity Transformations and Spatial Filtering

Digital Image Processing

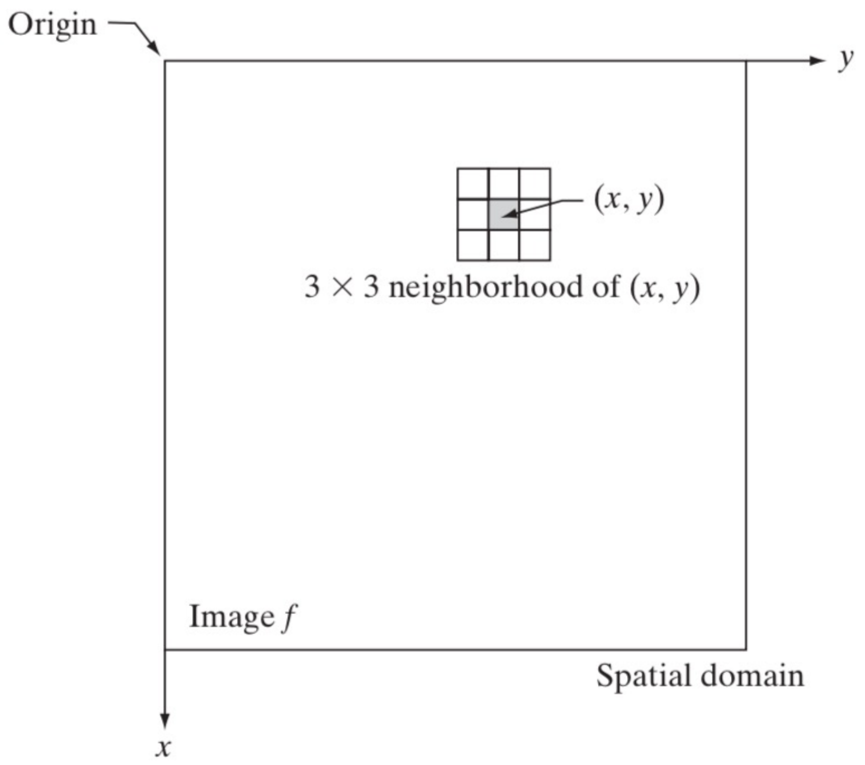
Contents

- Basics of Intensity Transformations and Spatial Filtering
- Basic Intensity Transformation Functions
- Histogram Processing
- Spatial Filtering Fundamentals
- Smoothing Spatial Filters
- Sharpening Spatial Filters
- Combining Spatial Enhancement
- Fuzzy Techniques

Basics of Intensity Transformations and Spatial Filtering

Intensity Transformations & Spatial Domain

- Basic intensity transform
  - $g(x, y) = T[f(x, y)]$
  - $f(x, y)$ is input image, and $g(x, y)$ is output
- Spatial domain operator
  - T is defined over neighborhood of point (x, y)



Spatial Domain Operator Example

- Averaging neighbor pixels
  - $g(x, y) \; = T[f(x, y)]$
- 4 neighbor
  - $g(x, y) = \frac{1}{5}\big(f(x, y) + f(x-1, y) + f(x+1, y) + f(x, y-1) + f(x, y+1)\big)$
- 8-neighbor
  - $g(x, y) = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} f(x+i, y+j)$
- To compute a pixel, $g(x, y)$
  - Locate the 3x3 window at $f(x, y)$
  - Collect the values in the window and compute the result

Filtering?

- Spatial filter
  - A set of coefficients (1/4’s and 1/9’s in the previous example)
  - Multiplied to the image window
  - Aka spatial mask, kernel, template, or window

Basic Intensity Transformation Functions

Intensity Transformation Functions

- Smallest neighborhood size: 1x1
  - $s = T(r)$
  - Many choices of $T(\cdot)$
  - Aka gray-le

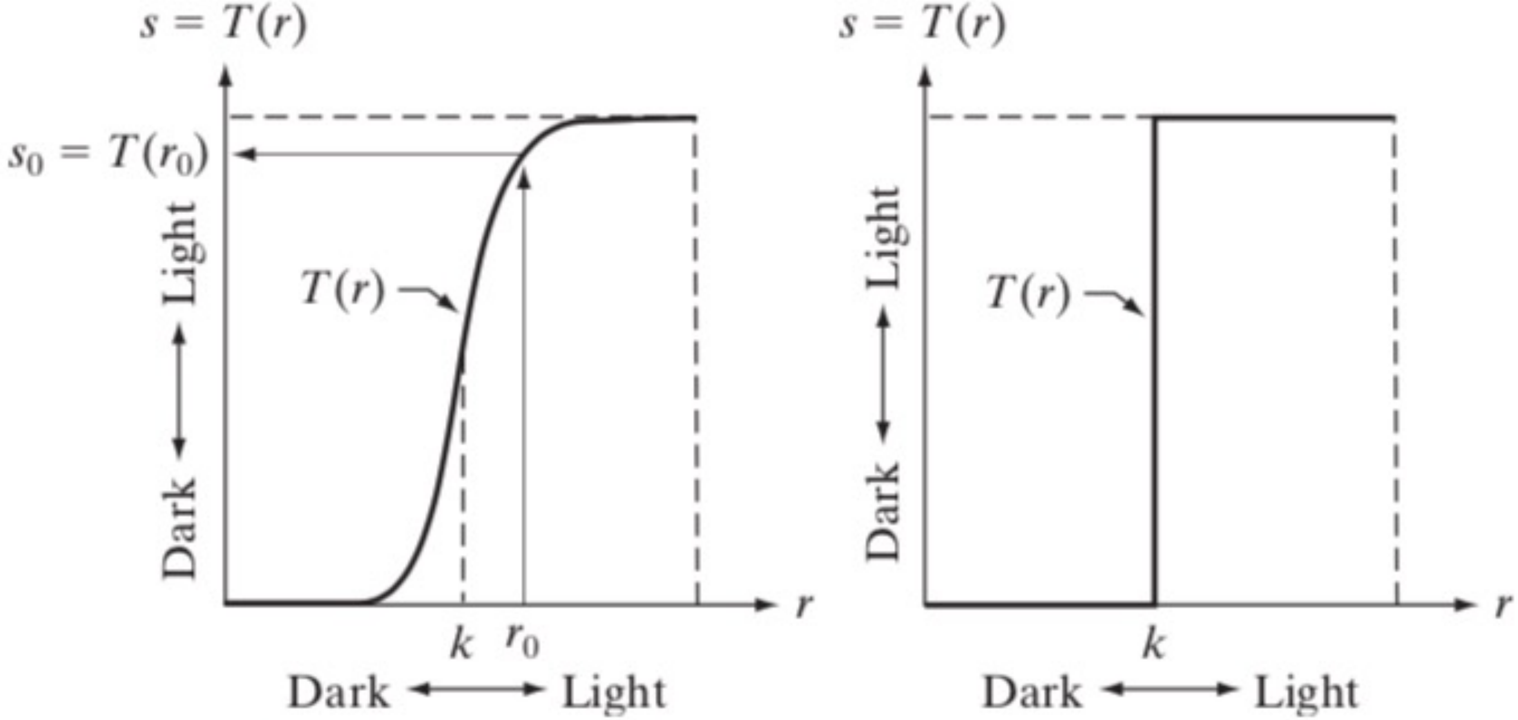
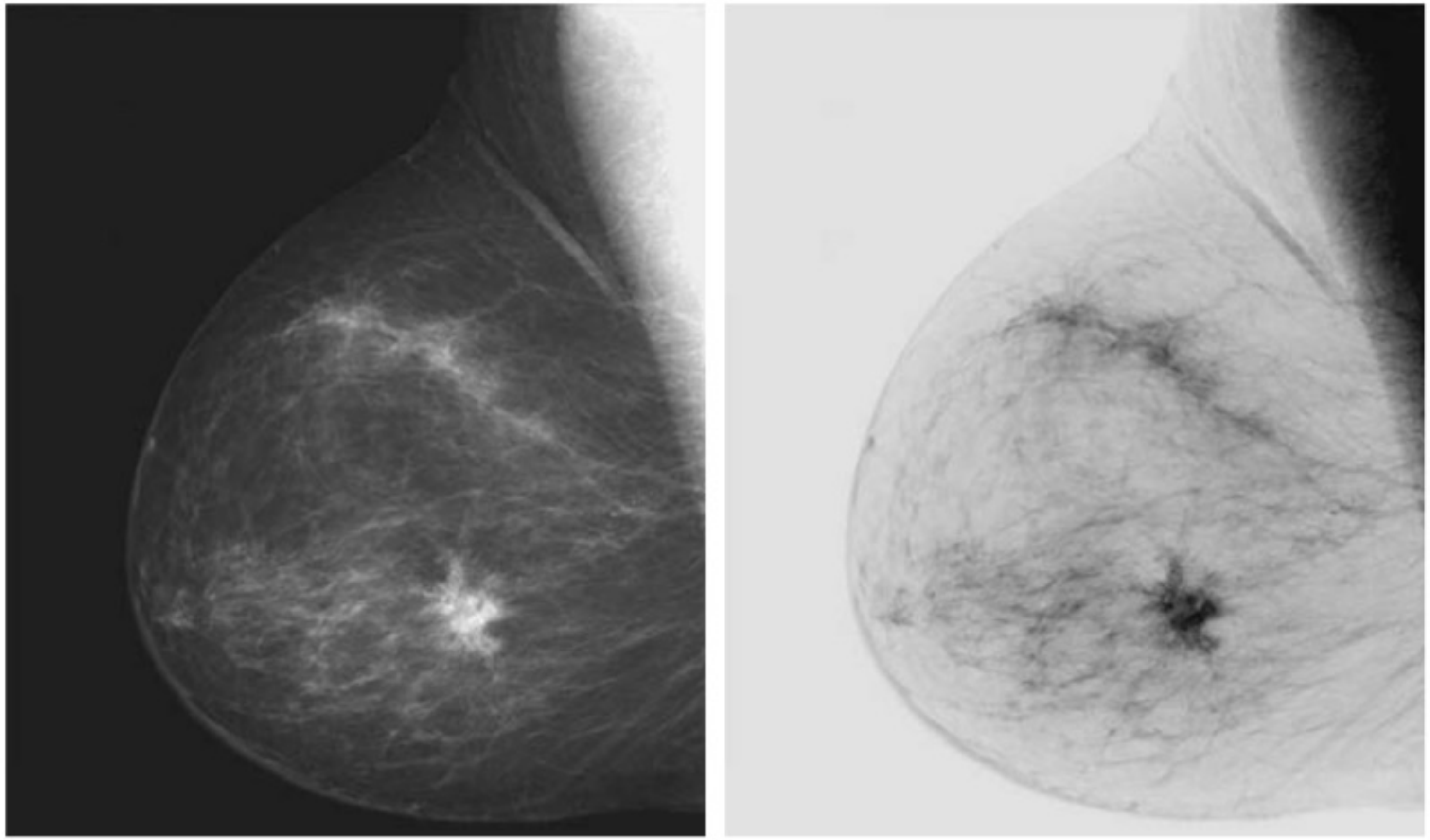


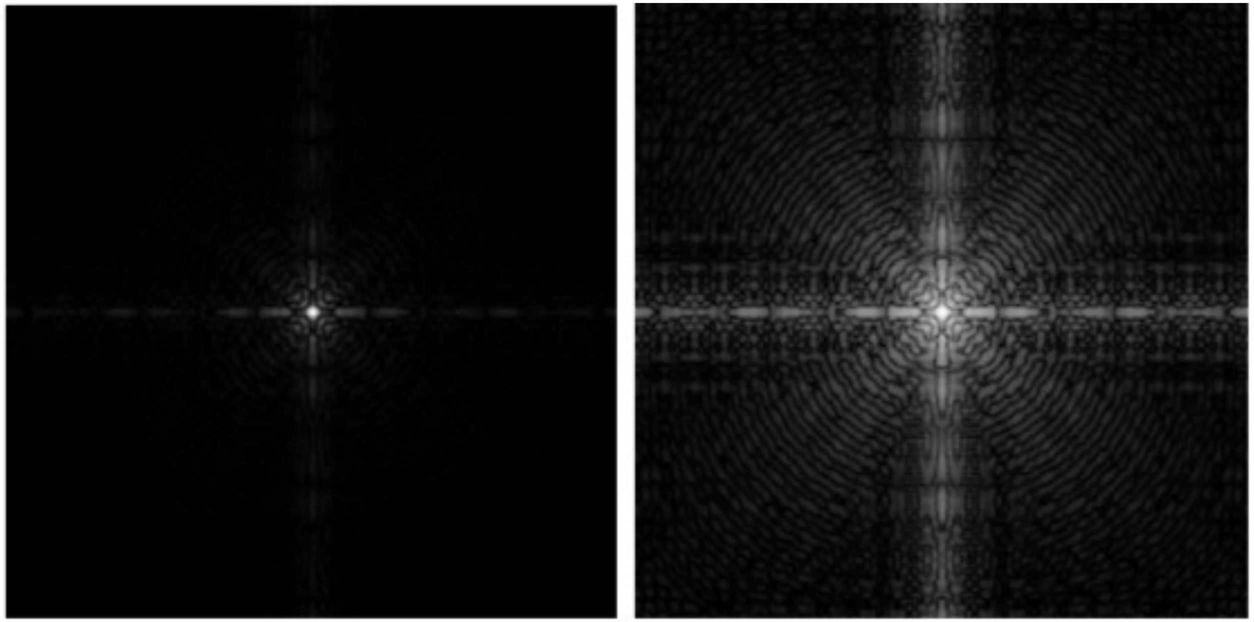
Image Negatives

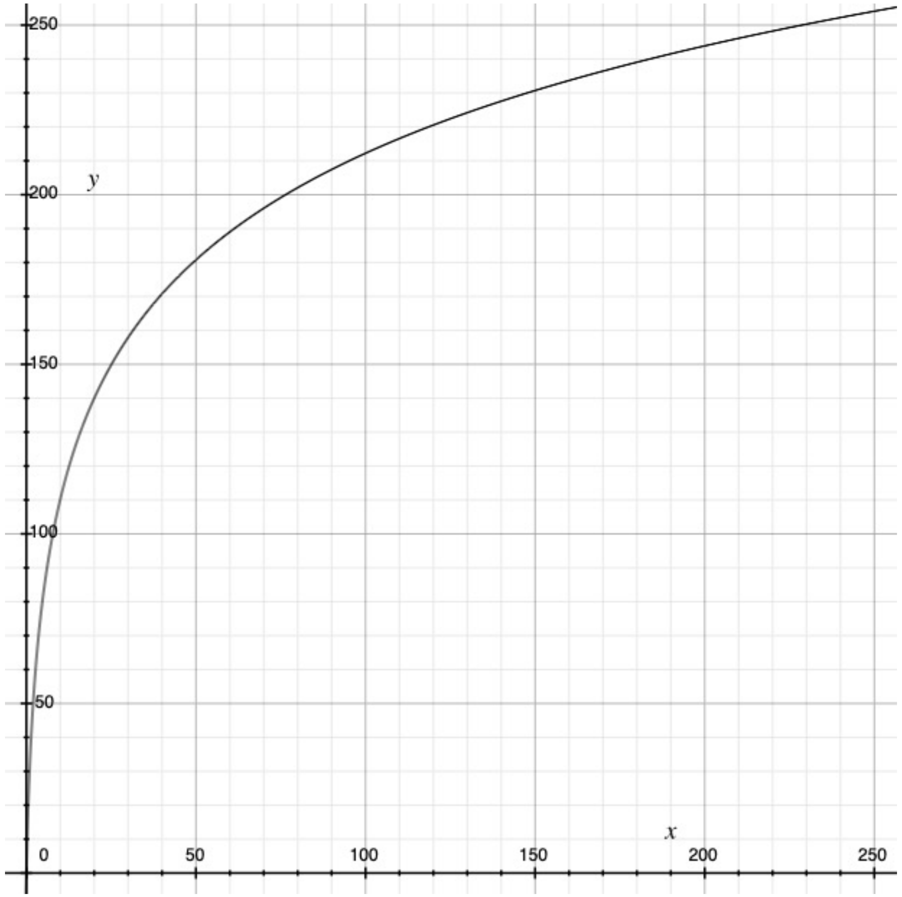
- Initial range
  - $[0, L-1]$
- Negatives
  - $s = L - 1 - r$



Log Transformation

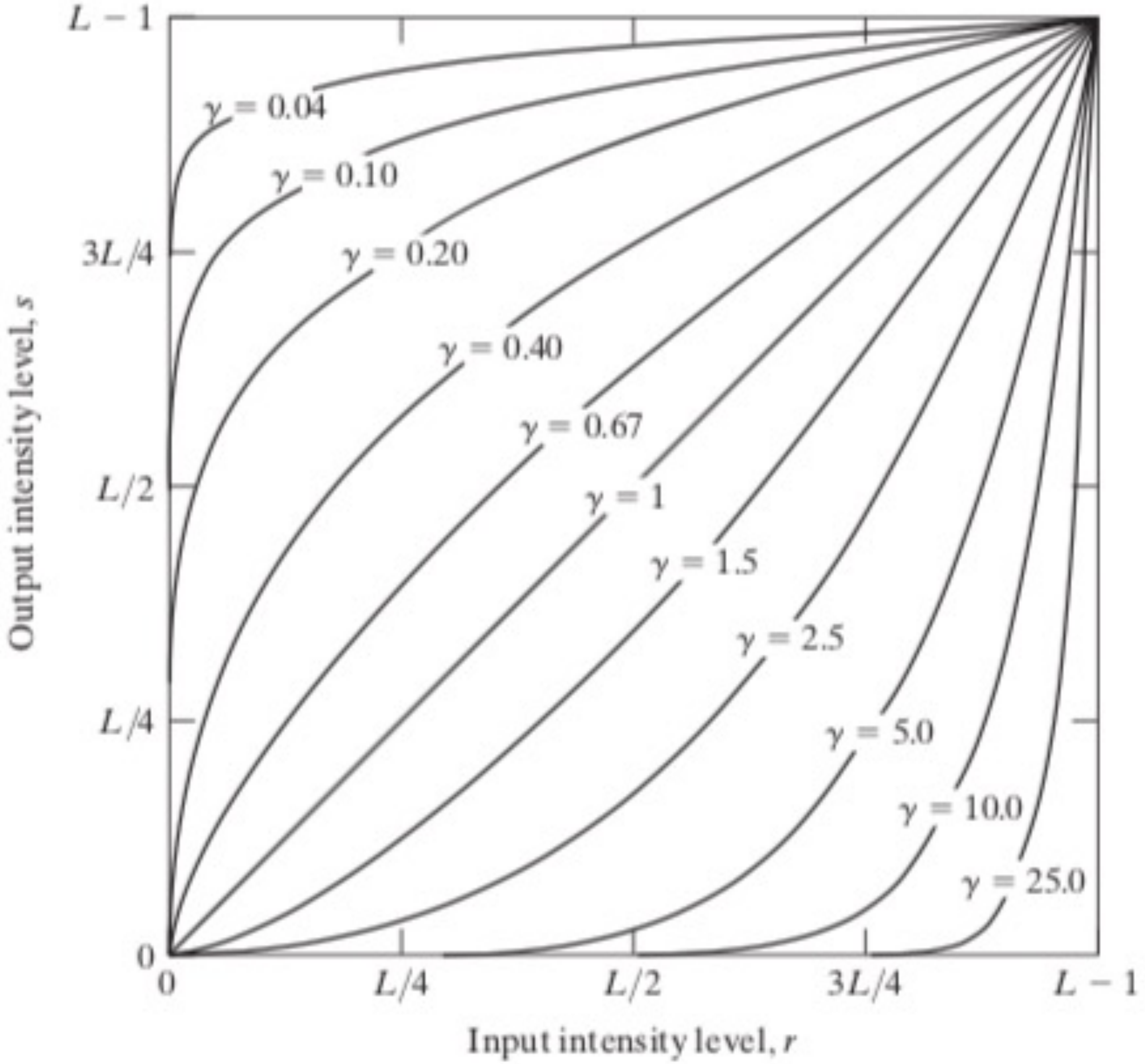
- Log transform
  - $s = c\log(1 + r)$





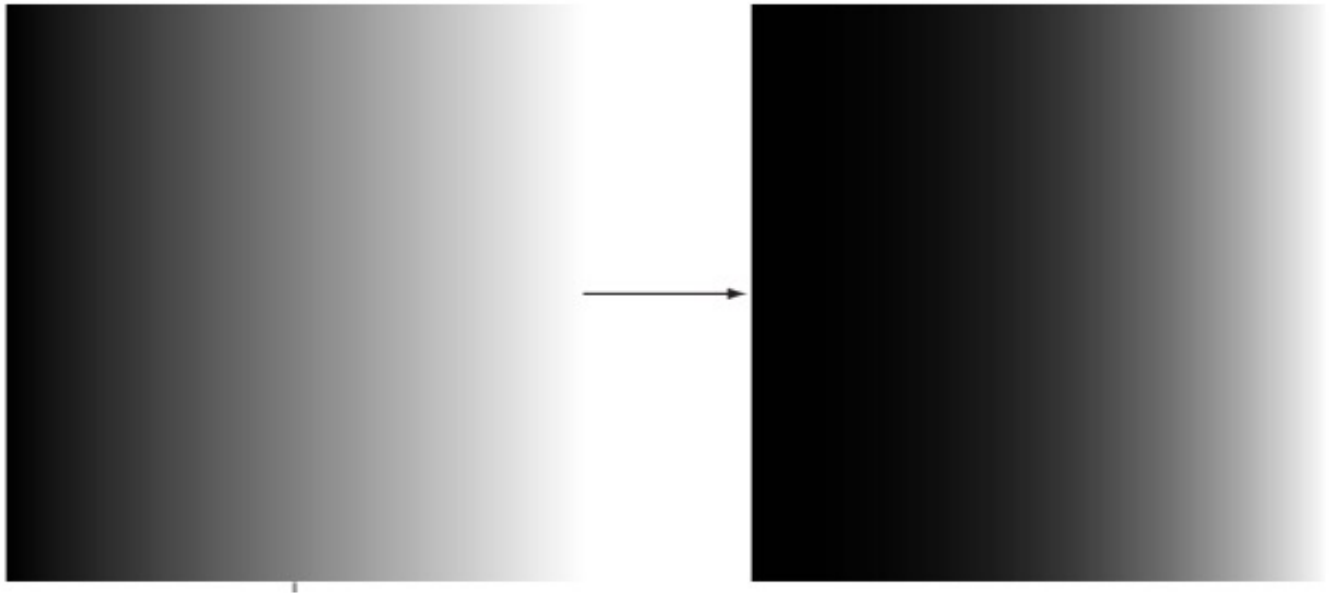
Power-Law (Gamma) Transformations

- General than Log transformation
- Power function
  - Aka Gamma correction
  - $s = cr^{\gamma}$
  - Usually $r \in [0,1], c = 1$
- Demo

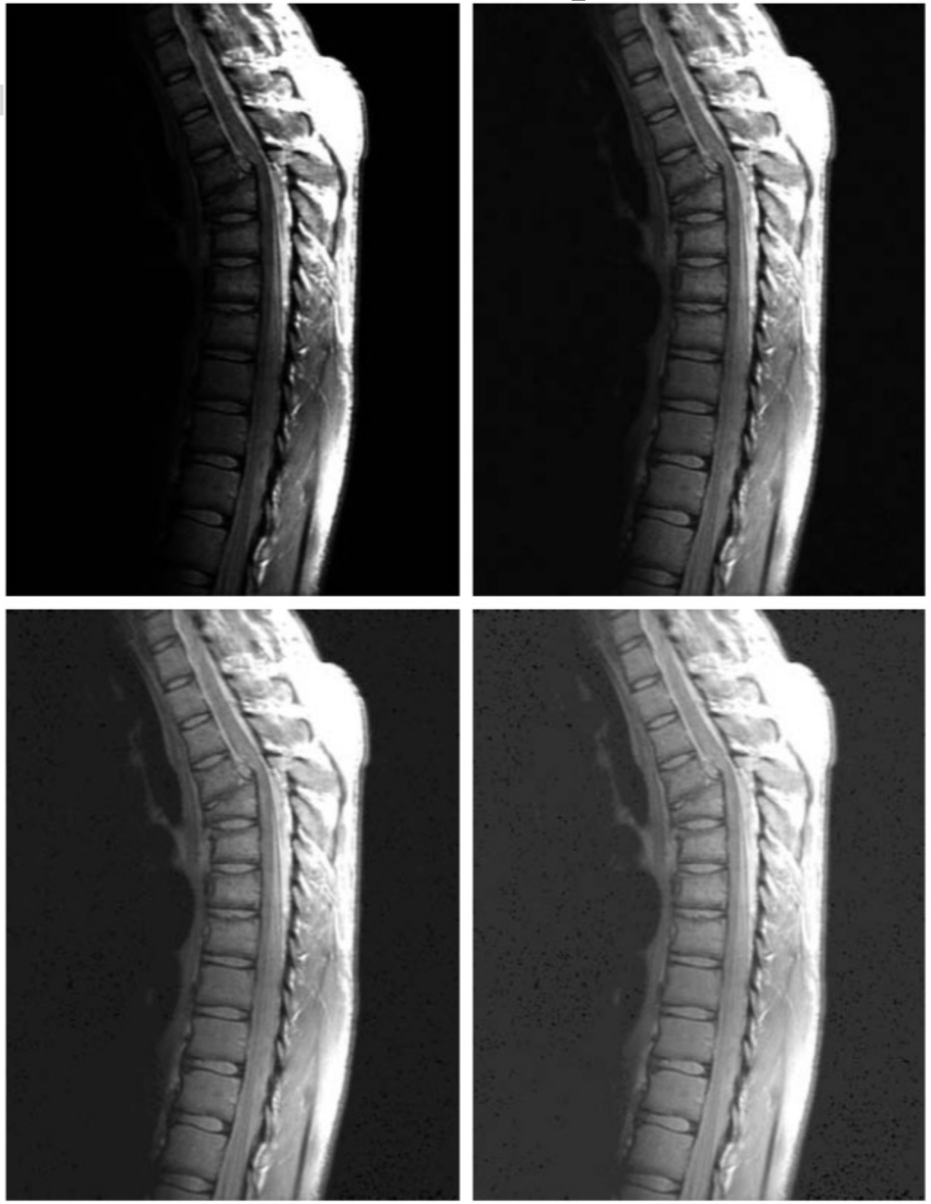


Power-Law (Gamma) Transformations

- Monitor response
  - Approximately $L = c^{2.2}$
  - Usually image file is stored transformed inversely: $f(x, y) = i(x, y)^{\frac{1}{2.2}}$
  - Linearization: Making $f'(x, y) = f(x, y)^{2.2}$

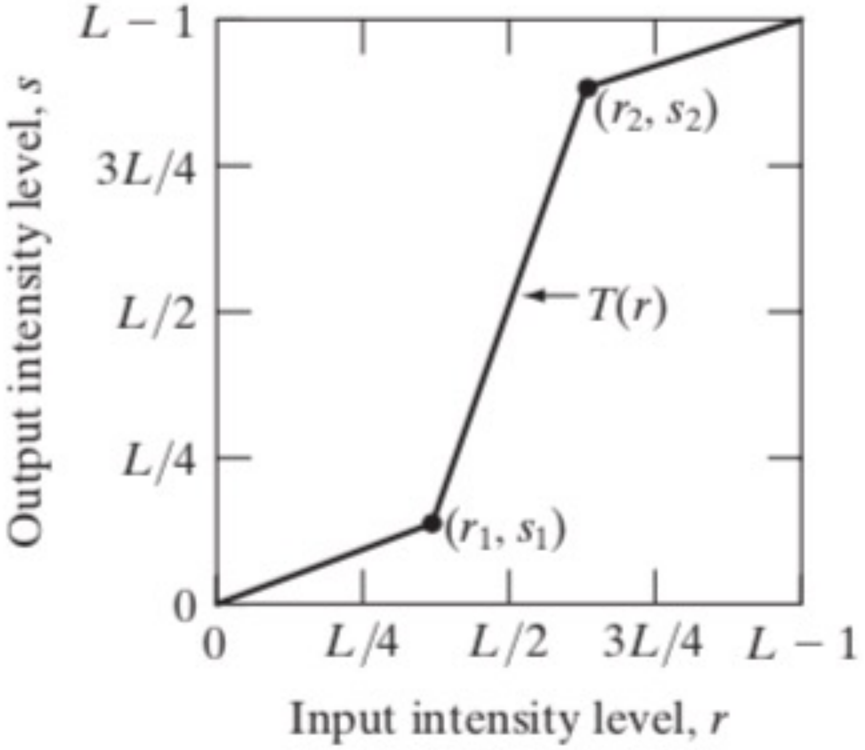


Power-Law (Gamma) Transformations



Piecewise-Linear Transformation

- Response curve is defined as line segments

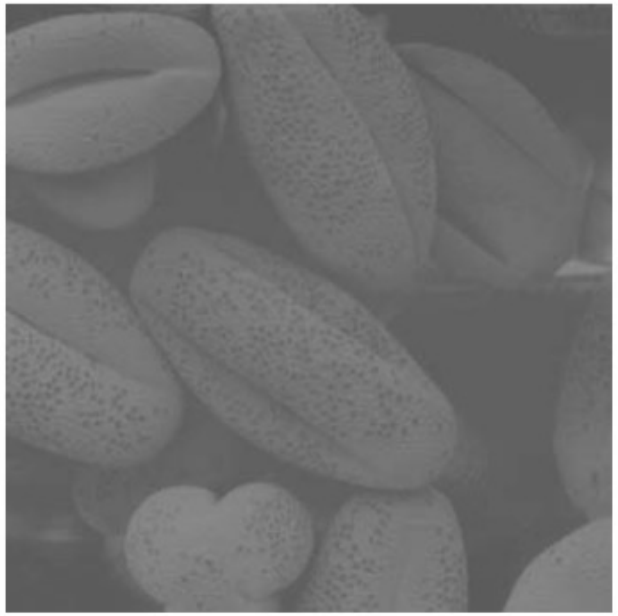


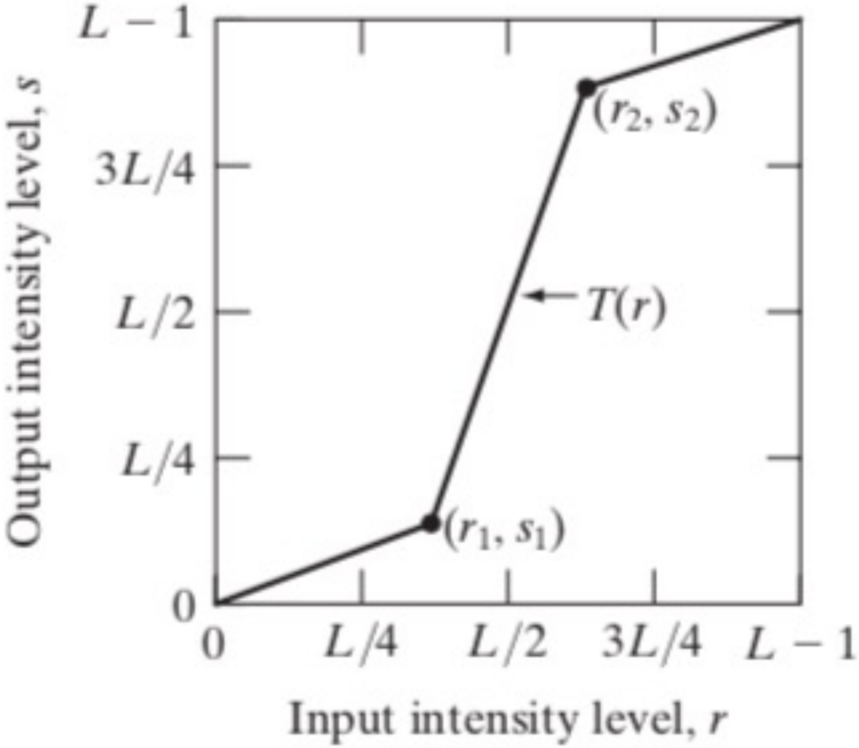
Piecewise-Linear Transformation

- Simple linear transform
  - $s = ar + b$
  - a: gain, b: bias
- $\boldsymbol{a}$ means
  - $a > 1$: Fast changing response than input
    - Small change in the input is exaggerated
  - $a < 1$: slower changing response
    - Big change in the input is suppressed
  - => Contrast
- $\boldsymbol{b}$ means
  - $b > 0$: s become larger than r => making image brighter
  - $b > 0$: the output gets darker
  - => Brightness

Piecewise-Linear Transformation

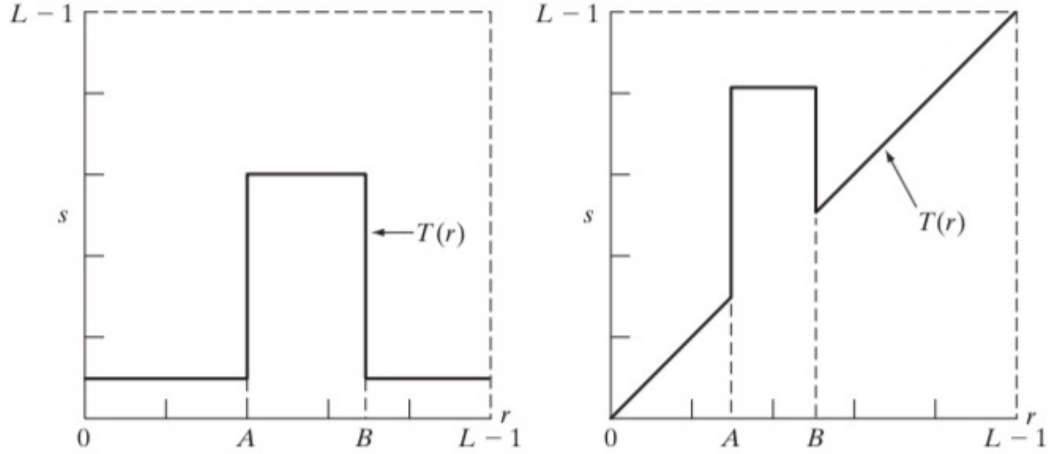
- Contrast stretching
  - Enhancing the midrange contrast

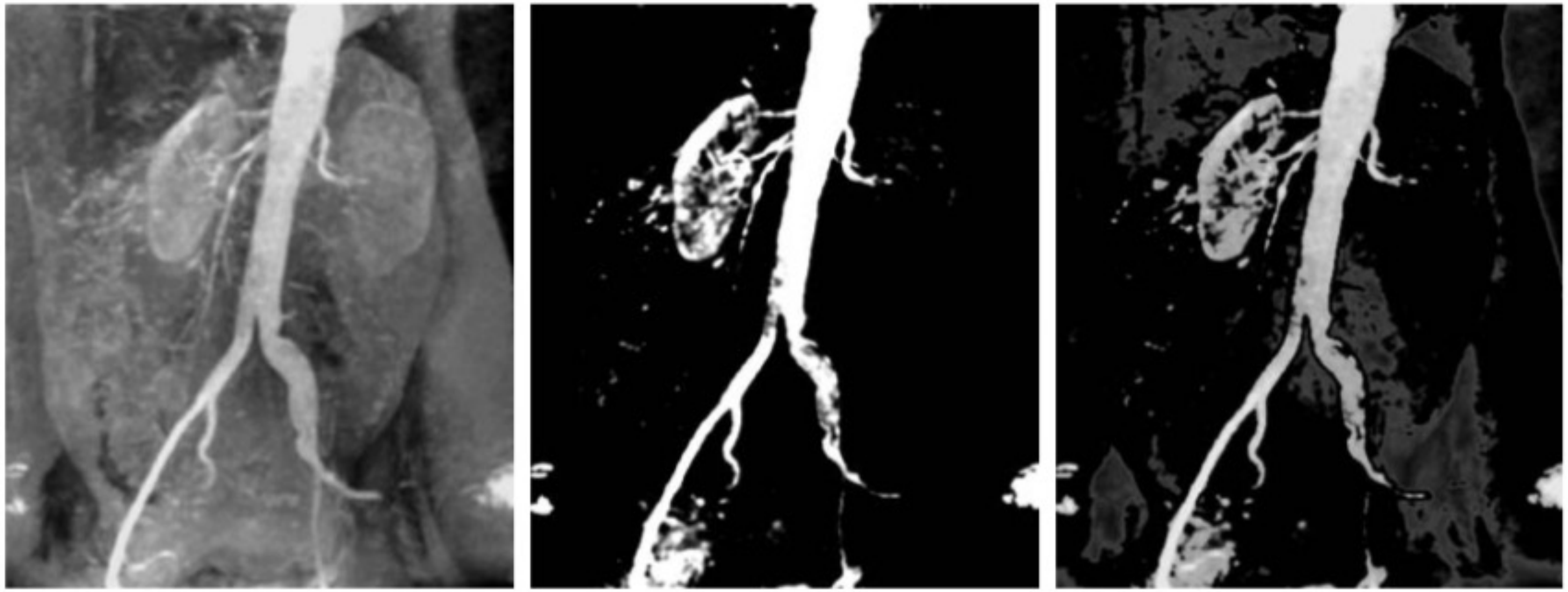




Piecewise-Linear Transformation

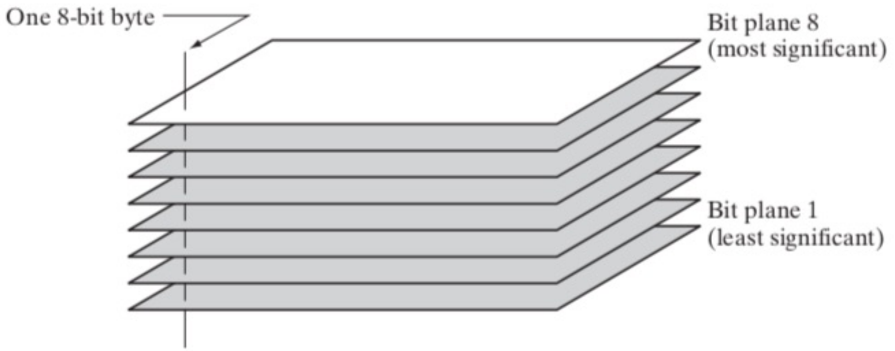
- Intensity slicing
  - Making an intensity range to a value
  - Leaving or suppressing others

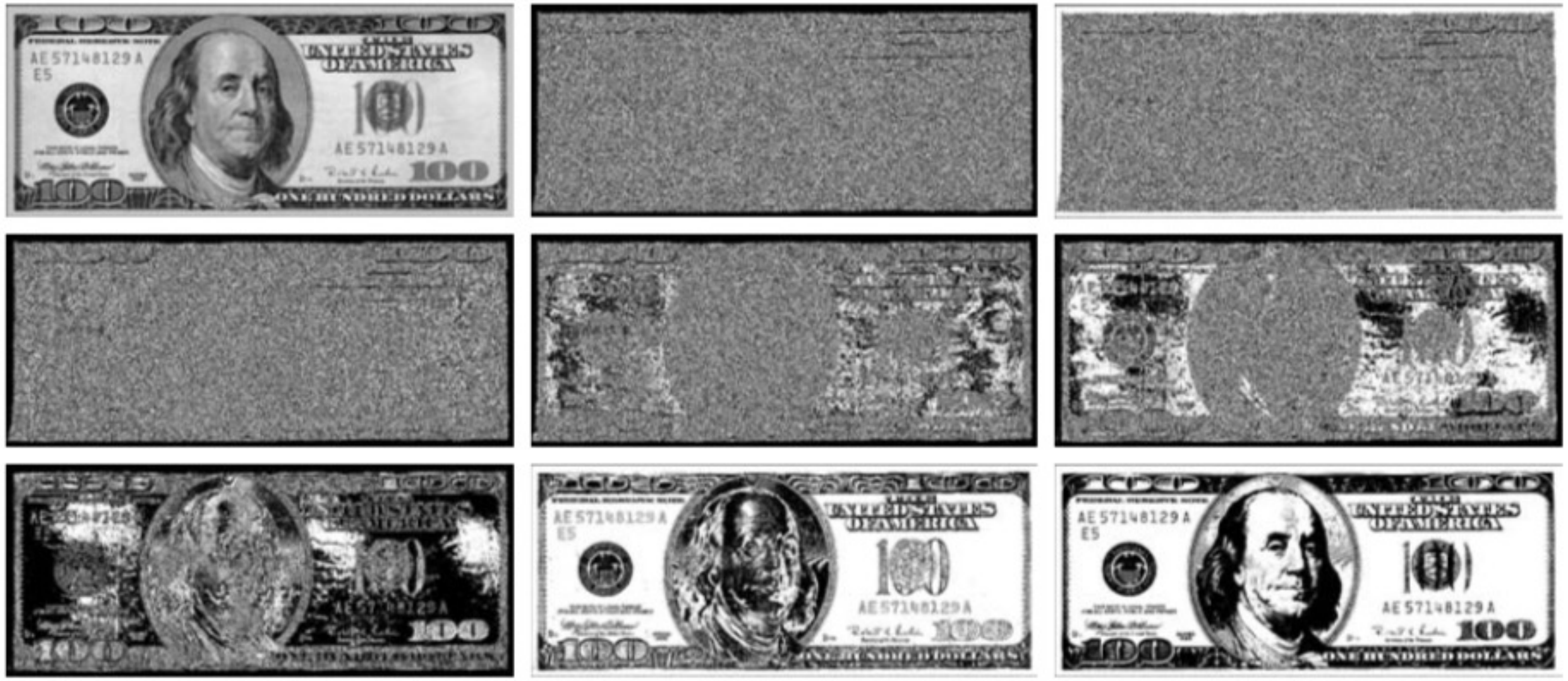




Piecewise-Linear Transformation

- Bit-plane slicing
  - Leaving only some bit-planes
  - Making the other zero
  - Usually most significant bits are left





Piecewise-Linear Transformation

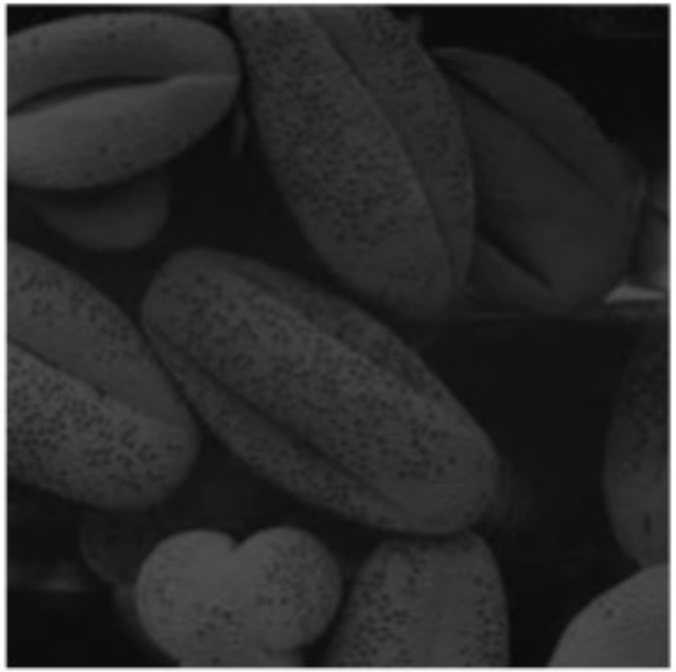
- Bit-plane slicing
  - Demo



Histogram Processing

Image Intensity Histogram

Image Intensity Histogram



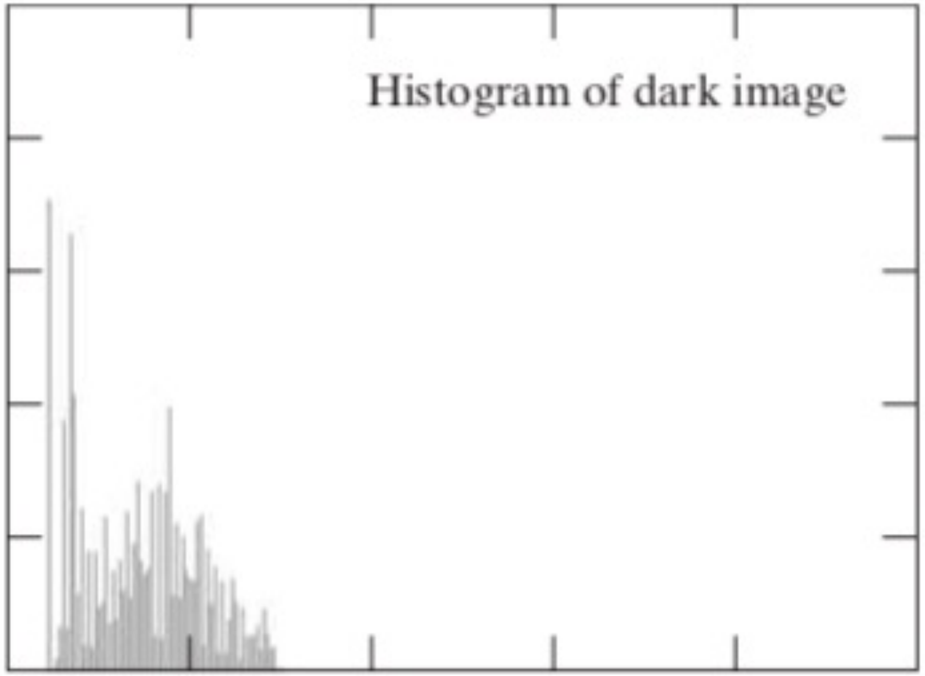
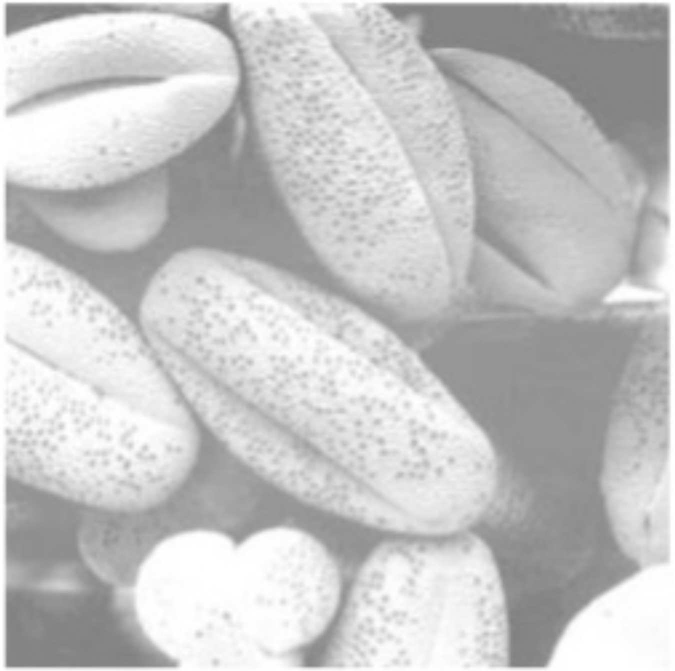


Image Intensity Histogram



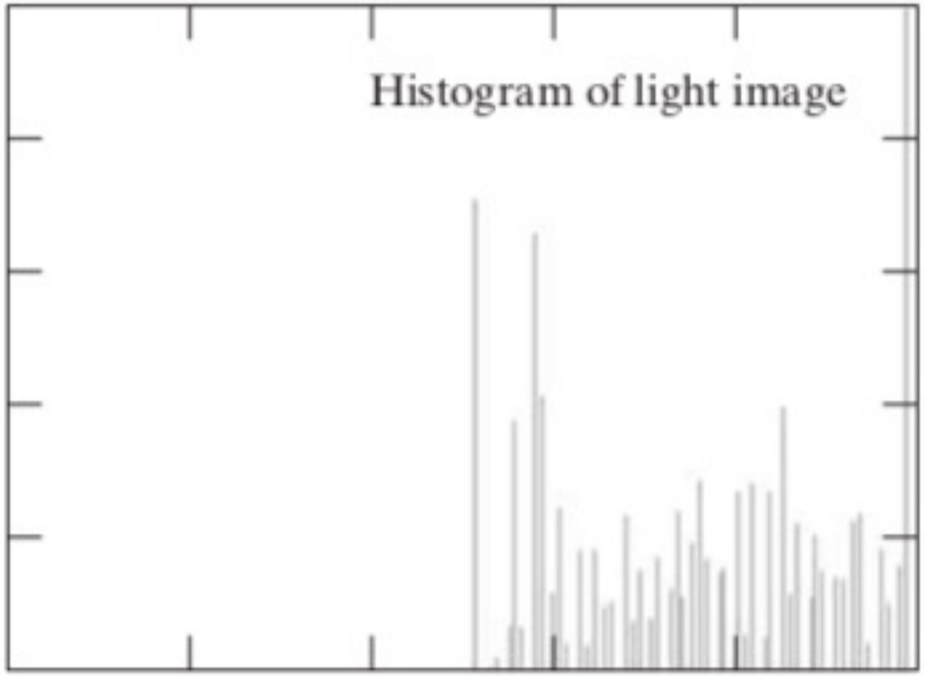
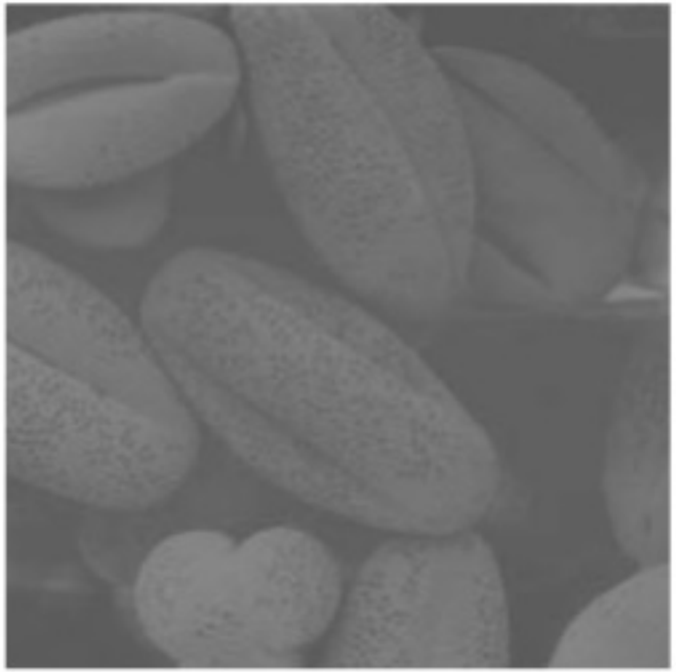


Image Intensity Histogram



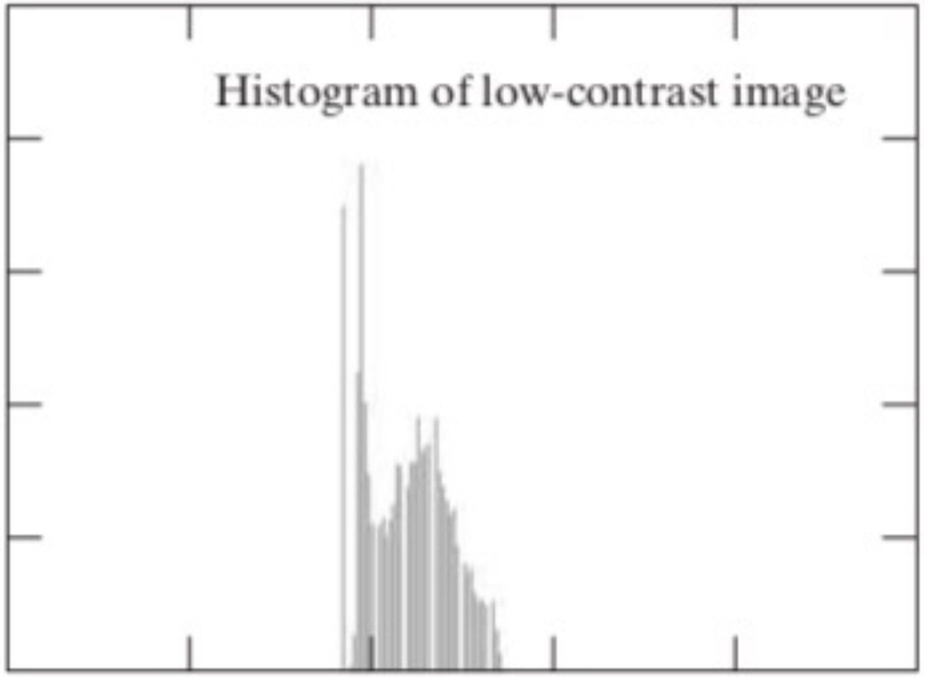
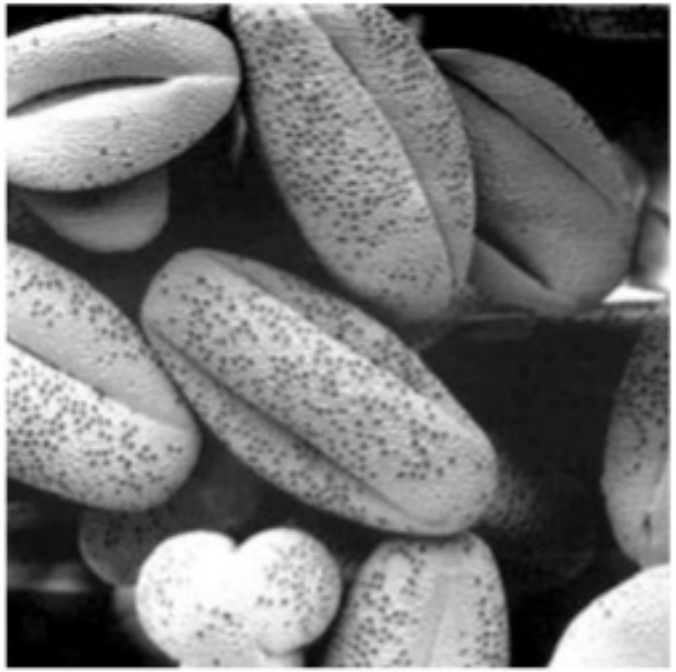
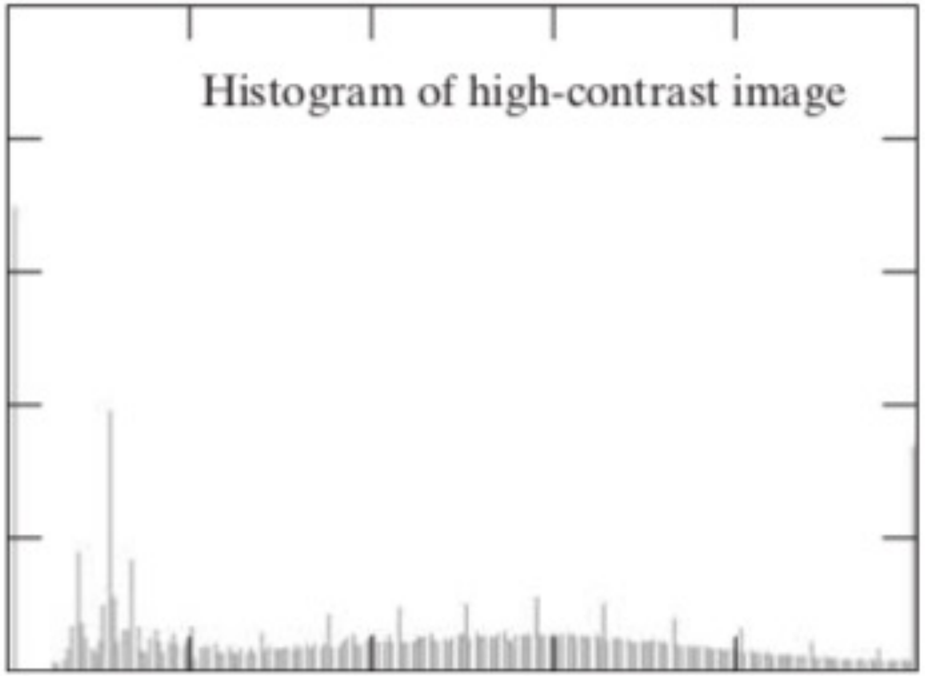


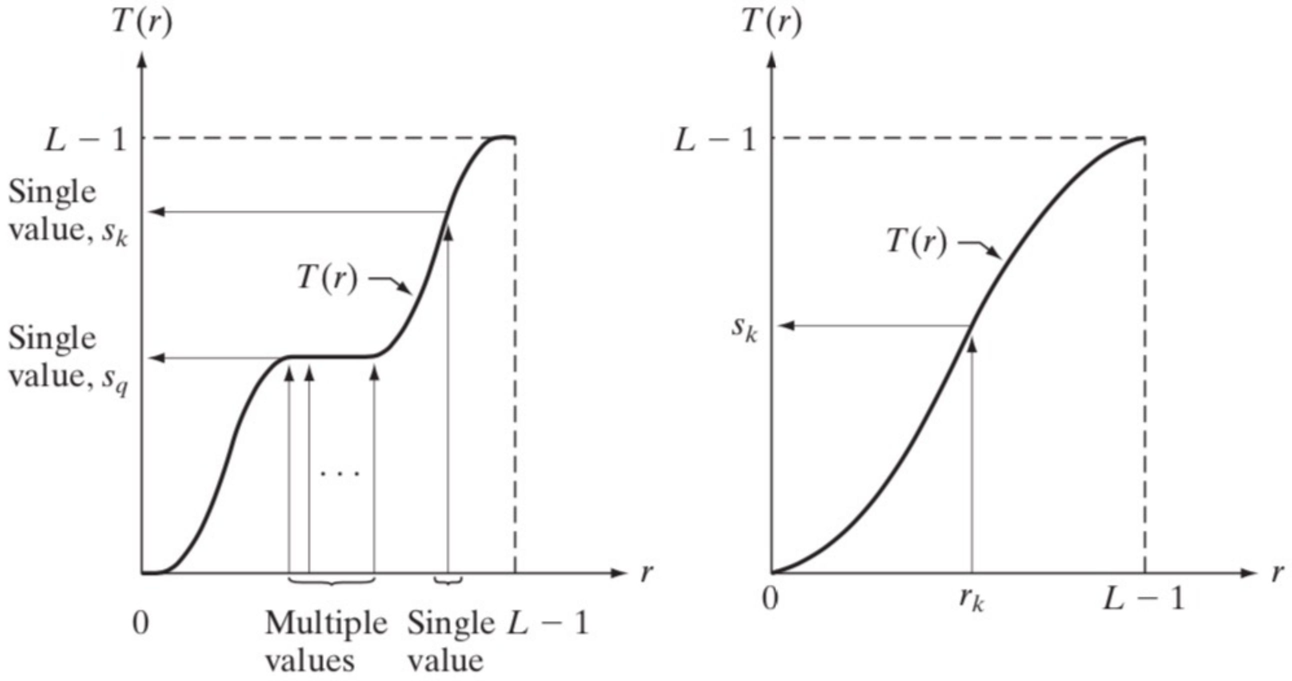
Image Intensity Histogram





Histogram Equalization

- An intensity transformation
  - $s = T(r) \quad 0 \leq r \leq L-1$
- Assumptions
  - $T(r)$ is strictly monotonically increasing function
  - $0 \leq T(r) \leq L-1$



Histogram as PDF

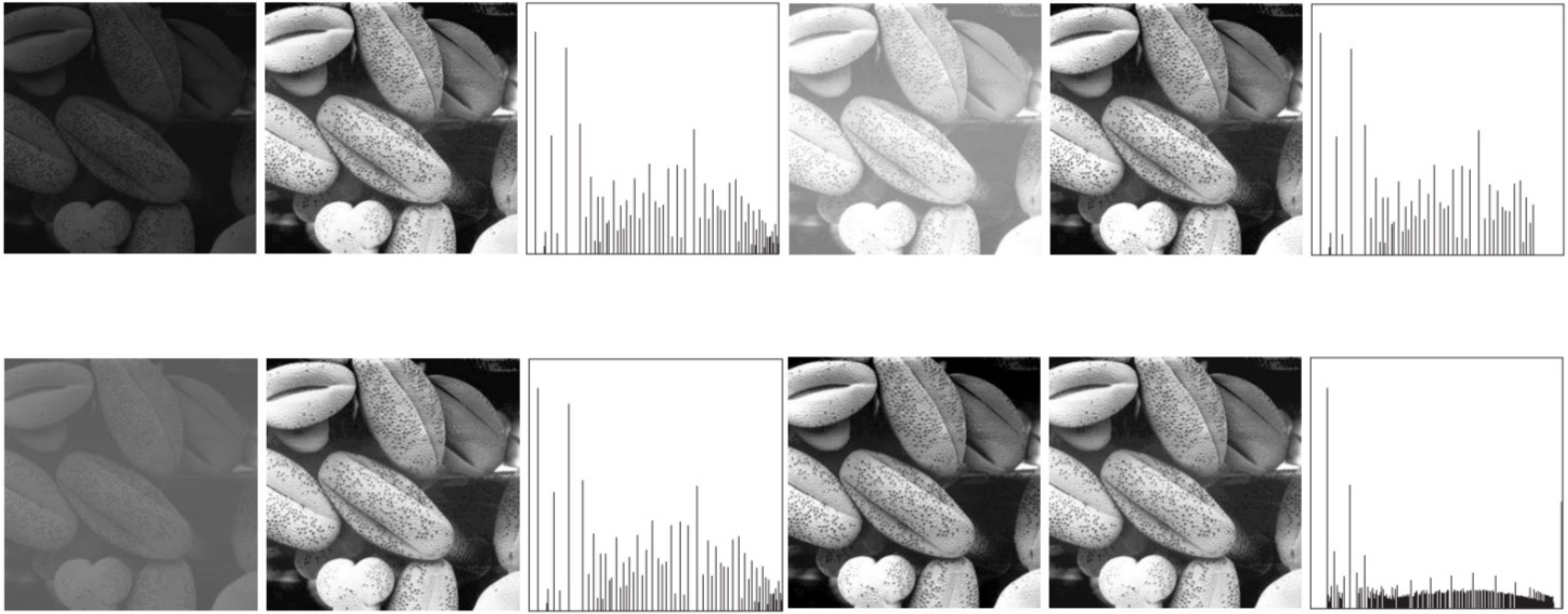
- Probability distribution function
  - $p_r(r) = \frac{H(r)}{\sum_j H(j)}$
  - $cdf_r(r) = \sum_{j=0}^{r} p_r(r)$

- Better contrasted image
  - Even histogram
  - => $p_s(s) = c$ (c is a constant)
  - => $p_s(s) = \frac{1}{L-1}$
  - => $cdf_s(s) = \frac{s}{L-1}$

Histogram Equalization

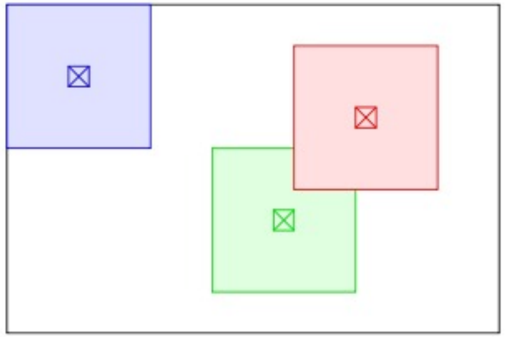
- Implementation
  - $cdf_r(r) = cdf_r\big(T(s)\big) = cdf_s(s)$
  - $cdf_s(s) = \frac{s}{L-1} = \sum_{j=0}^{r} p_r(r)$
  - $s = T(r) = (L-1)\sum_{j=0}^{r} p_r(r) = (L-1)\sum_{j=0}^{r} \frac{H(r)}{\sum_j H(j)} = \frac{(L-1)}{NM}\sum_{j=0}^{r} H(r)$

Histogram Equalization Results



Other Histogram Transformation

- Histogram matching
  - Making the histogram of an image as the given profile
- Local (adaptive) histogram equalization (AHE)
  - Global histogram equalization cannot improve local contrast
  - Histogram equalization on a window
  - Blending results at overlapping window

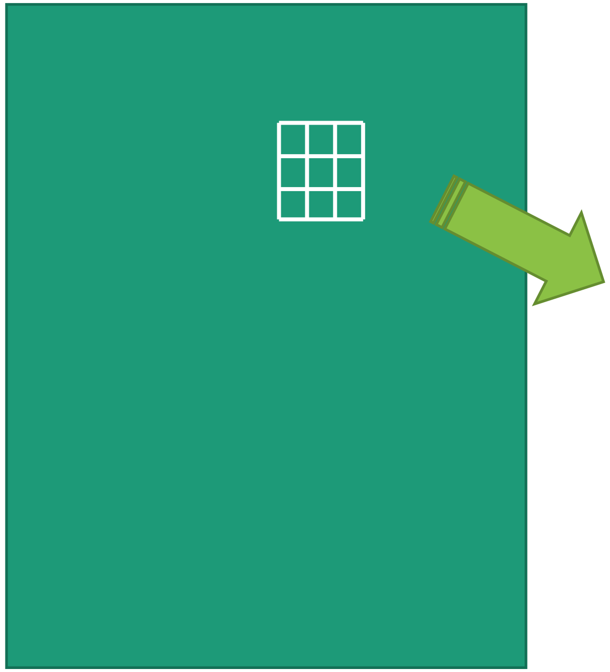


Spatial Filtering Fundamentals

Spatial Filtering

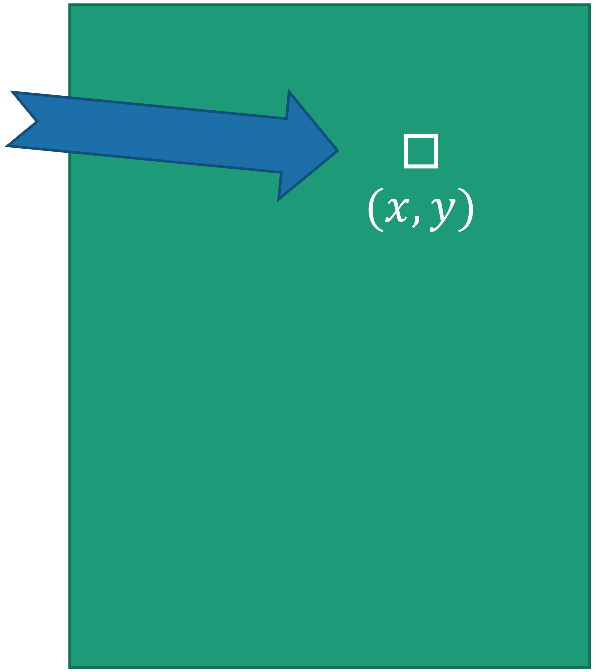
- Main components
  - Neighborhood
  - Predefined operation (kernel, mask, …)
- Definition
  - $g(x, y) = \underbrace{\sum_{s=-a}^{a} \sum_{t=-b}^{b}}_{\text{Neighborhood}} \underbrace{w(s, t)}_{\text{Operation}} f(x + s, y + t)$

Spatial Filtering



$f(x-1, y-1)$	$f(x, y-1)$	$f(x+1, y-1)$
$f(x-1, y)$	$f(x,y)$	$f(x+1, y)$
$f(x-1, y+1)$	$f(x, y+1)$	$f(x+1, y+1)$

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



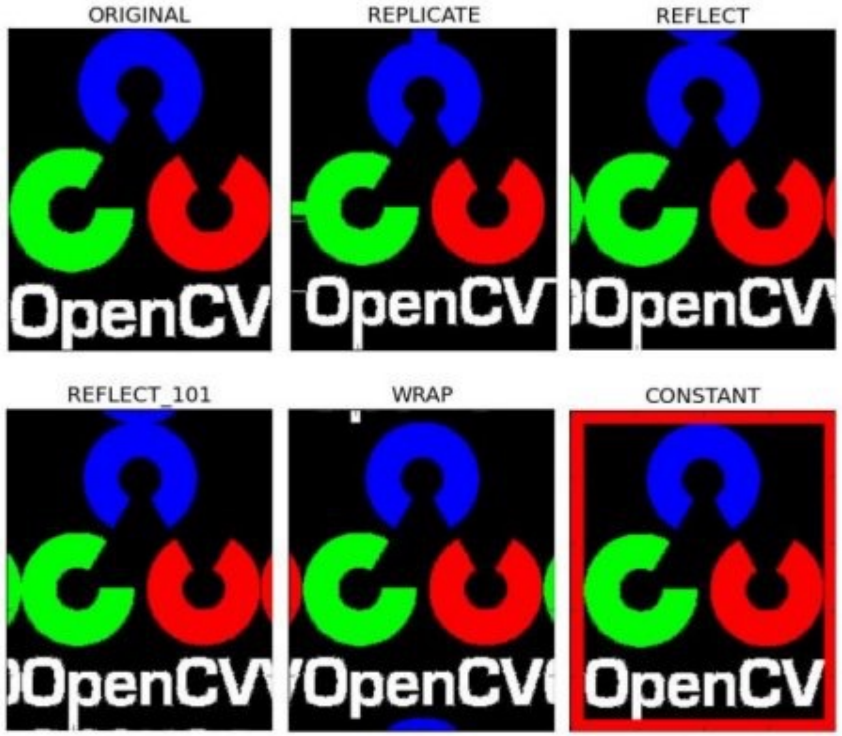
Similar Concepts

- Correlation
  - Exactly same as Spatial filtering
- Convolution
  - $g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t) = w(x, y) \otimes f(x, y)$

Padding

- No sufficient neighbors at the corner

- Adding (virtual) pixels at the four edges
  - Constant padding (zero padding)
  - Replicate padding
  - Mirror (reflection) padding



Smoothing Spatial Filters

Smoothing

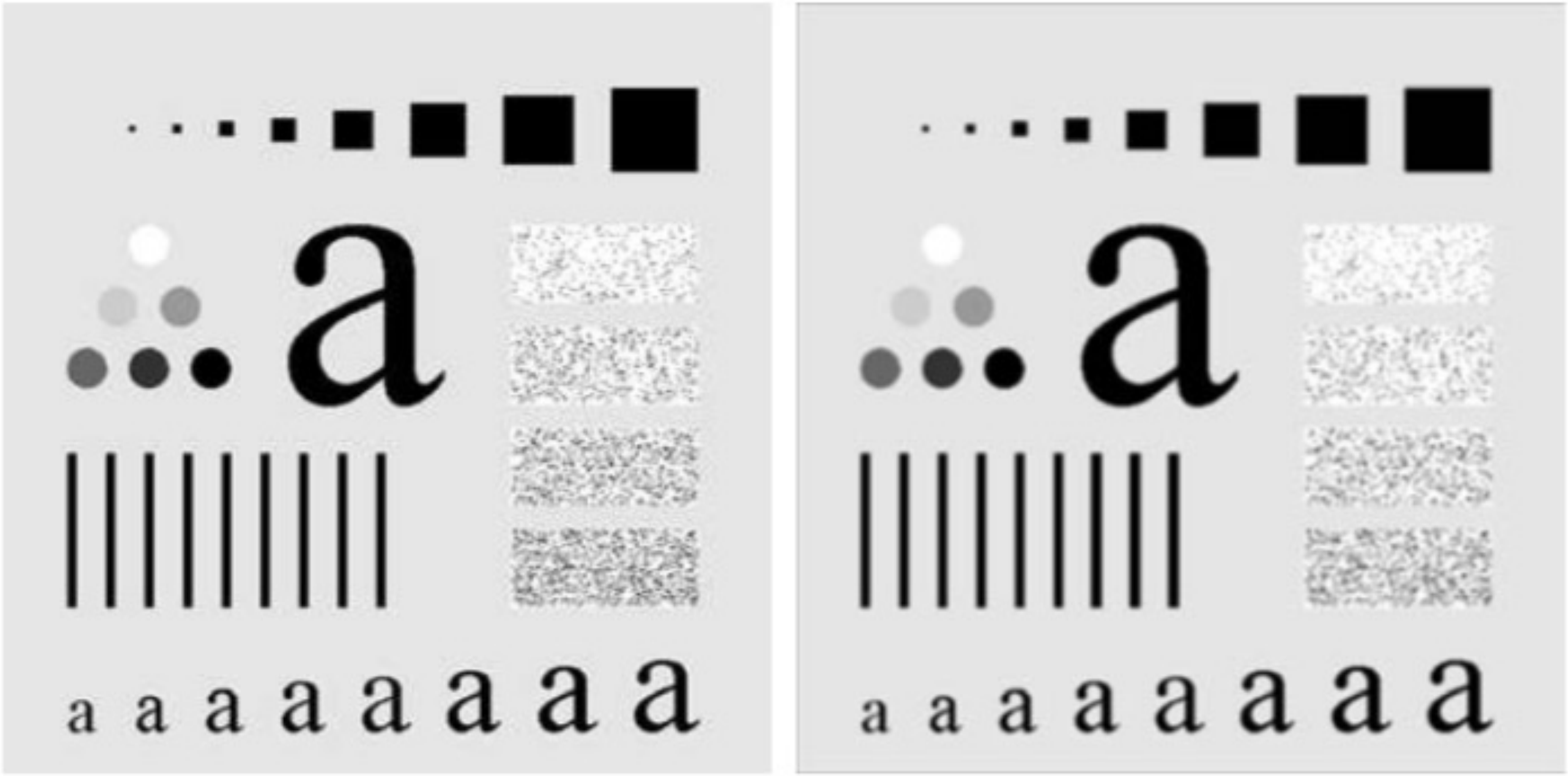
- Purpose
  - Making image smooth
  - Suppress noise
  - Removing high frequency

Averaging Filter

- Averaging neighborhood to produce the result
- 3x3 case
  - $r(x, y) = \frac{1}{9} \sum_{s=-1}^{1} \sum_{t=-1}^{1} f(x+s, y+t)$
  - Operation: $\frac{1}{9}$
- Also known as box filter

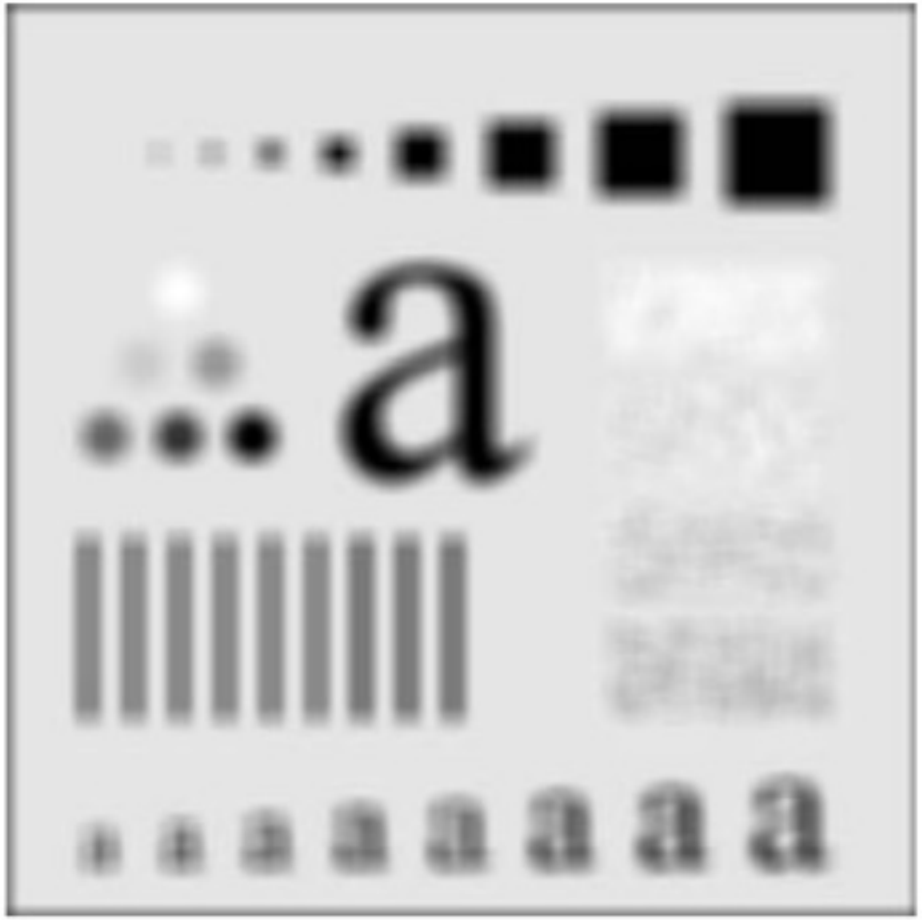
$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Box Filter



Box Filter

Box Filter



Weighted Averaging Filter

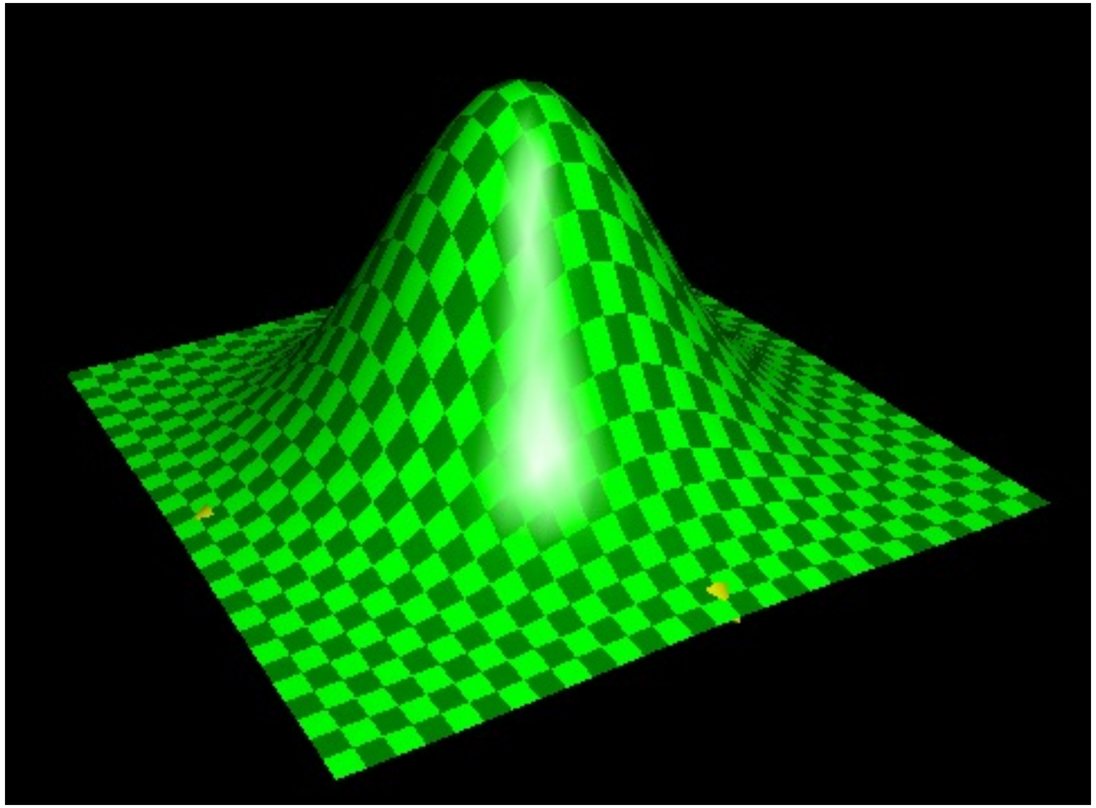
$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Weighted Averaging Filter

- More sophisticated kernel: Gaussian
  - $w(s,t) = \frac{1}{2\pi\sigma^2} e^{-\frac{s^2+t^2}{\sigma^2}}$

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



Weighted Averaging Filter

Noisy input: PSNR = 39.1 dB

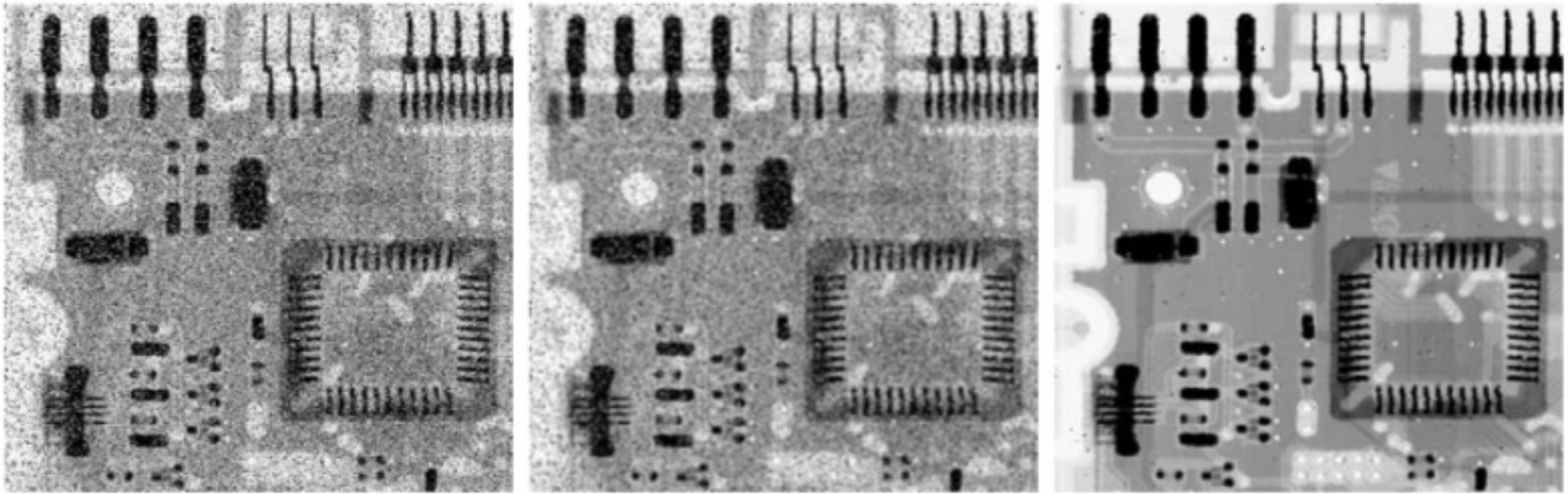
Gaussian filtered: PSNR = 67.9 dB

Separable Filter

- Operation on large kernel takes a lot of time ($m \times n$)
- Some operator is "separable"
  - Filtering x-direction (1D) first and then filtering y-direction yield the same result as 2D filtering
  - Making operation complexity to $m + n$
- Example
  - $\frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
  - $\frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Order-Statistic (Non-linear) Filters

- Based on ordering (ranking) pixels in a window
- Median filter
  - Finding median in a window (neighborhood)
  - Removing impulse (slat and pepper) noise

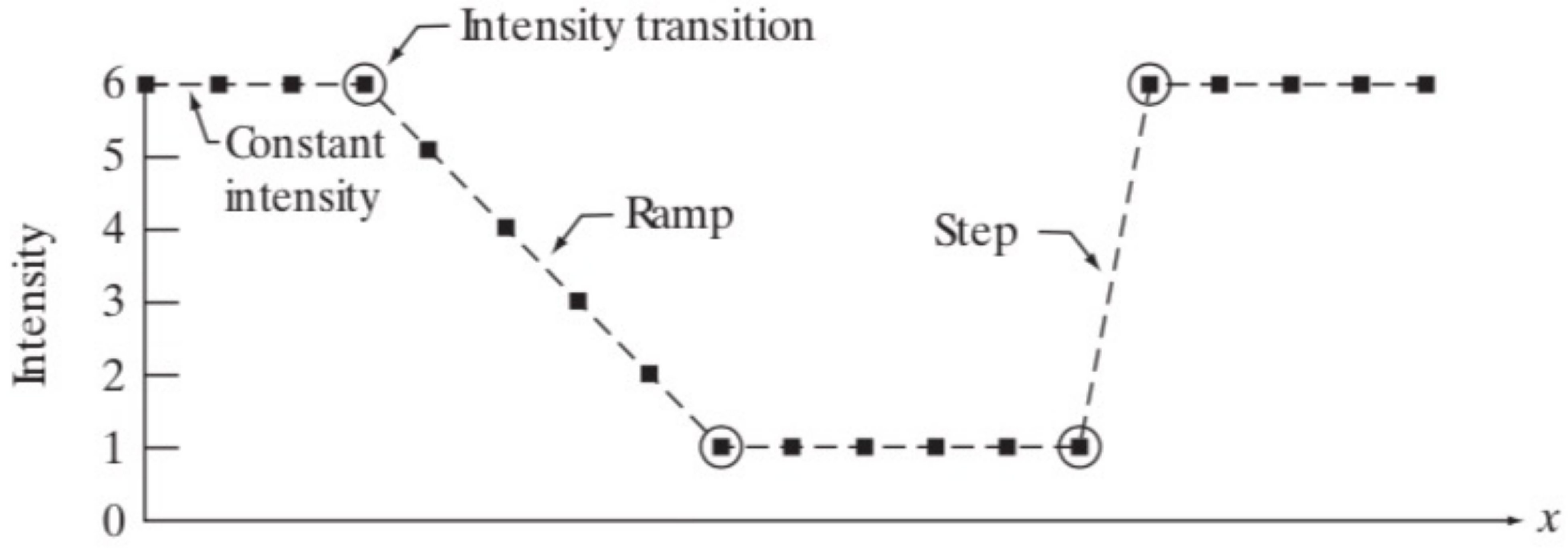


Sharpening Spatial Filters

Derivatives

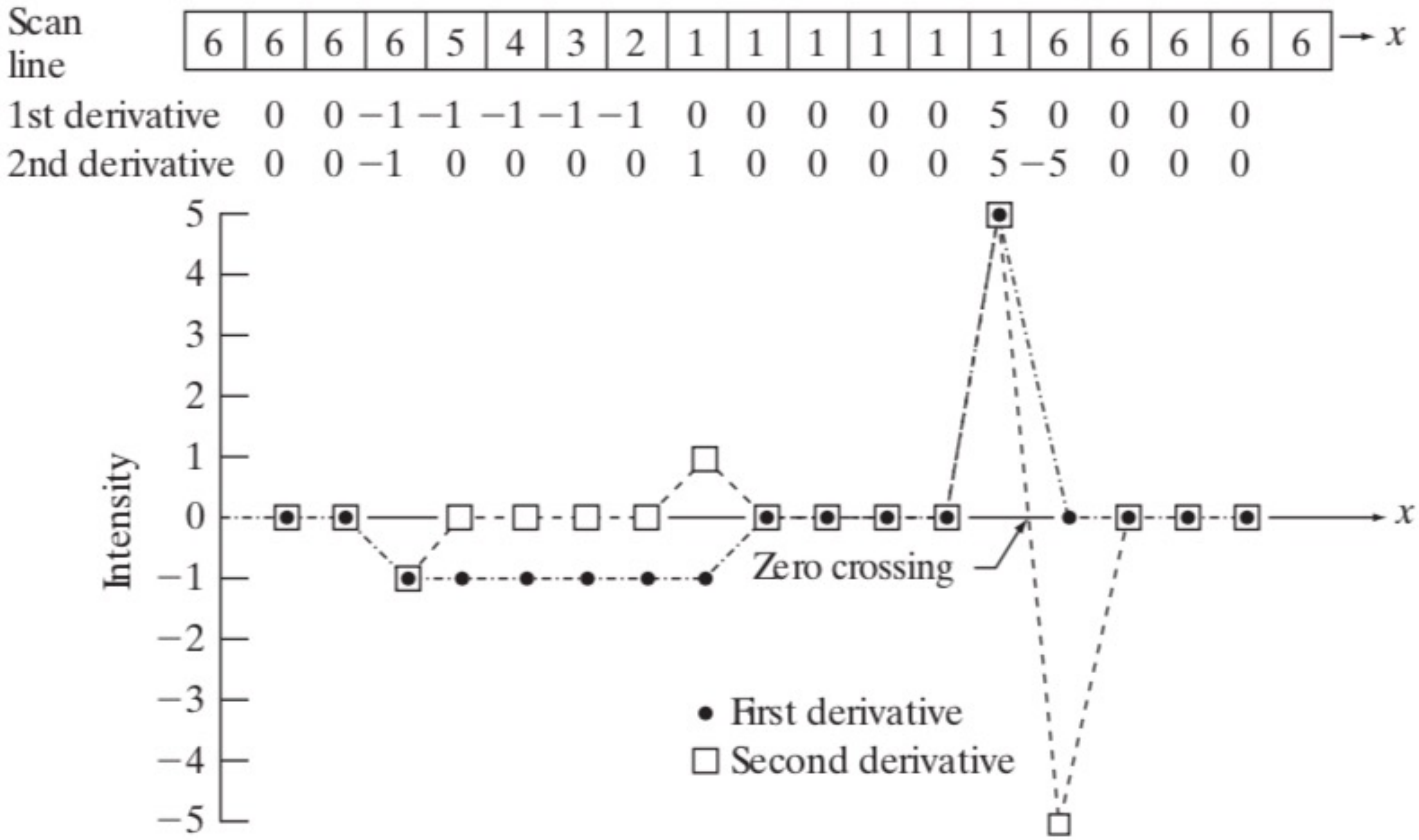
- Definition
  - $\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$
- Discretized version
  - Finite difference
  - $\frac{\partial f}{\partial x} = f(x+1) - f(x)$ (right difference)$\approx f(x) - f(x-1)$ (left difference)
- Second derivative
  - $\frac{\partial f}{\partial x}(x+1) = f(x+1) - f(x)$
  - $\frac{\partial f}{\partial x}(x) = f(x) - f(x-1)$
  - $\frac{\partial^2 f}{\partial x^2}(x) = \frac{\partial f}{\partial x}(x+1) - \frac{\partial f}{\partial x}(x) = f(x+1) - f(x) - \big(f(x) - f(x-1)\big)$
    $= f(x+1) + f(x-1) - 2f(x)$

Derivatives



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	

Derivatives



Derivatives

- Properties of a first derivative
  - Zero in area of constant intensity
  - Nonzero at the onset of an intensity step (aka ramp)
  - Nonzero along ramps
- Properties of a second derivative
  - Zero in constant areas
  - Nonzero at the the onset
  - Zero along ramps of constant slope
- Zero crossing
  - The second derivative cross x axis (sign changing)

The Laplacian

- Isotropic
  - Independent to the image orientation (direction)
  - Aka rotation invariant
- Laplacian
  - $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
  - $\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$
  - $\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$
  - $\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4(x, y)$

The Laplacian

- Kernel and its variation

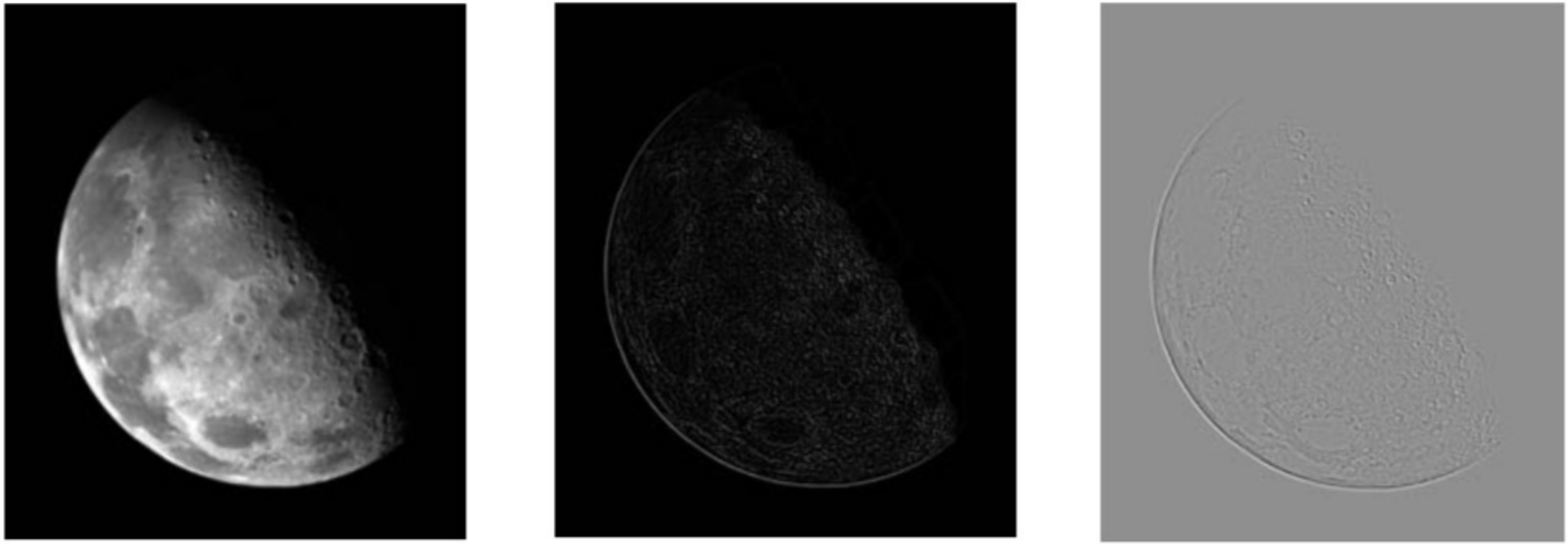
0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

The Laplacian

- Properties of a second derivative
  - Zero in constant areas
  - Nonzero at the the onset
  - Zero along ramps of constant slope
- Properties of the Laplacian
  - Highlighting intensity discontinuities
  - Deemphasizing regions with slowly varying intensity

The Laplacian



- Note: The Laplacian can be “negative”

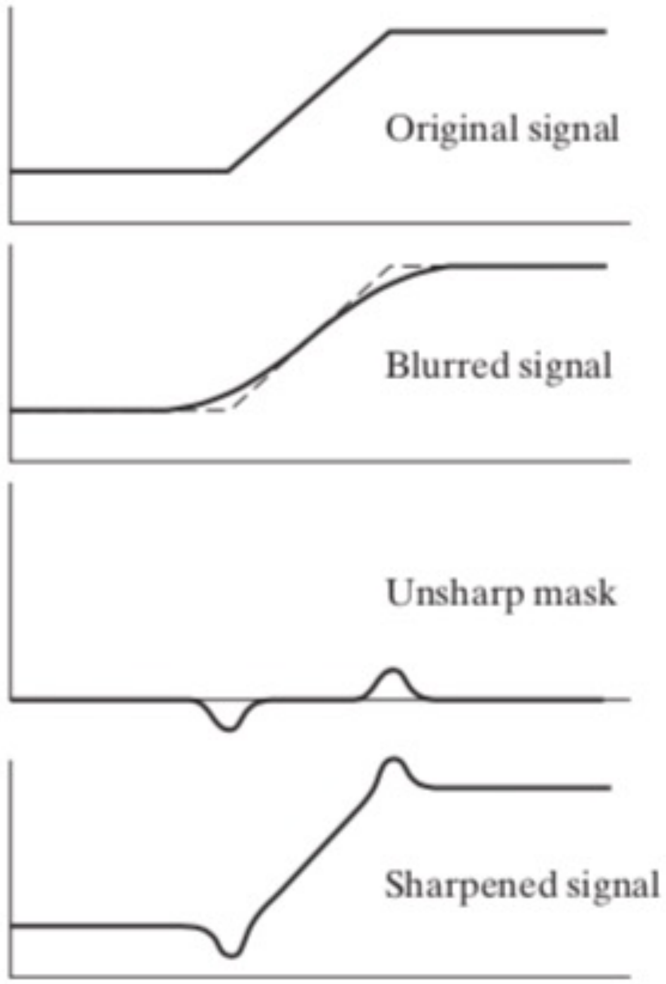
Image Sharpening with Laplacian

- Simple addition
  - $g(x, y) = f(x, y) + c[\nabla^2 f(x, y)]$



Unsharp Masking and Highboost Filtering

- Steps
  - Blur the original image
    - Leaving only global information
  - Subtract the blurred image from the original image
    - Taking fine details
    - Mask
    - $g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$
  - Add the mask to the original image
    - Exaggerating fine details
    - $g(x, y) = f(x, y) + k * g_{mask}(x, y)$



Unsharp Masking and Highboost Filtering

- $\boldsymbol{k = 1}$
  - Unsharp masking
- $\boldsymbol{k > 1}$
  - Highboost filtering
- $\boldsymbol{k < 1}$
  - De-emphasizing the contribution of the unsharp mask

Gradient

- Gradient
  - $\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$
  - The direction of the greatest rate of change of f at location (x, y)
- Magnitude
  - $M(x, y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$
  - Note: magnitude of gradient is not linear

Gradient & Sobel Operator

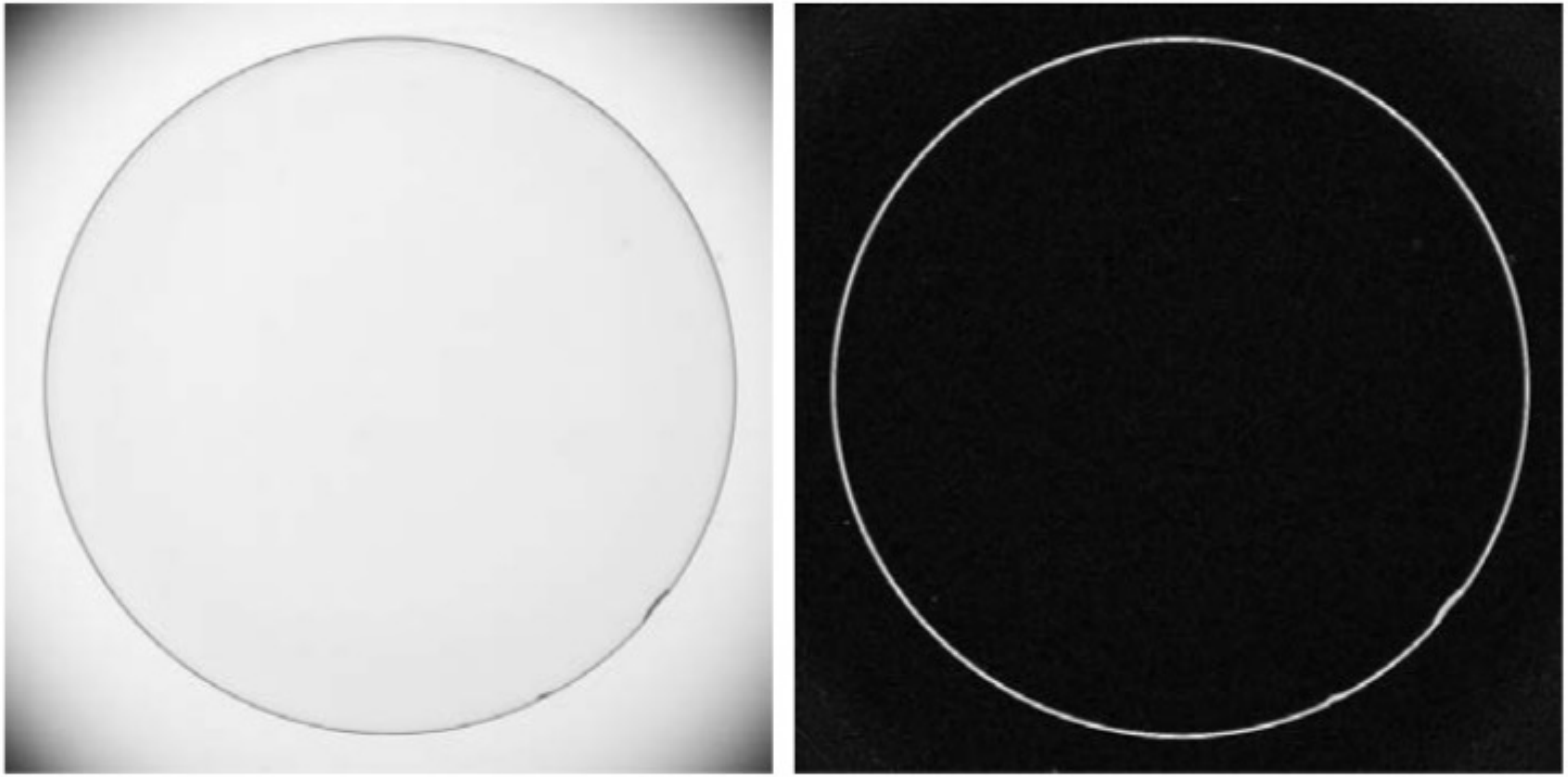
- Basic gradient
  - $g_x = z_8 - z_5,\ g_y = z_6 - z_5$
  - Not center symmetric
- Symmetric gradient
  - $g_x = z_8 - z_2, g_y = z_6 - z_4$
- 3x3 version
  - $g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$
  - $g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$
  - Called "Sobel" operator

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

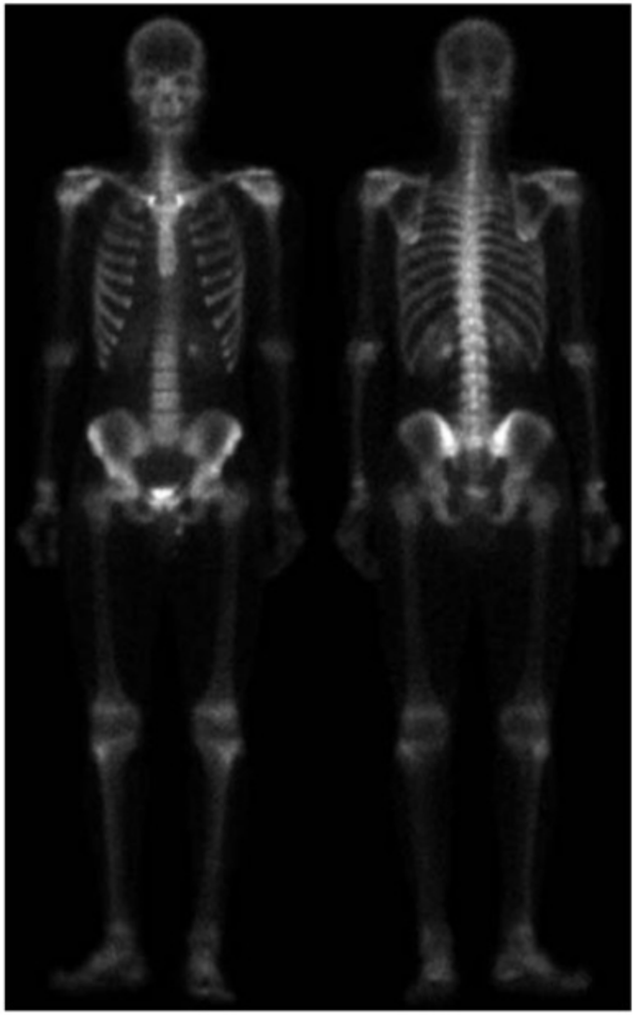
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

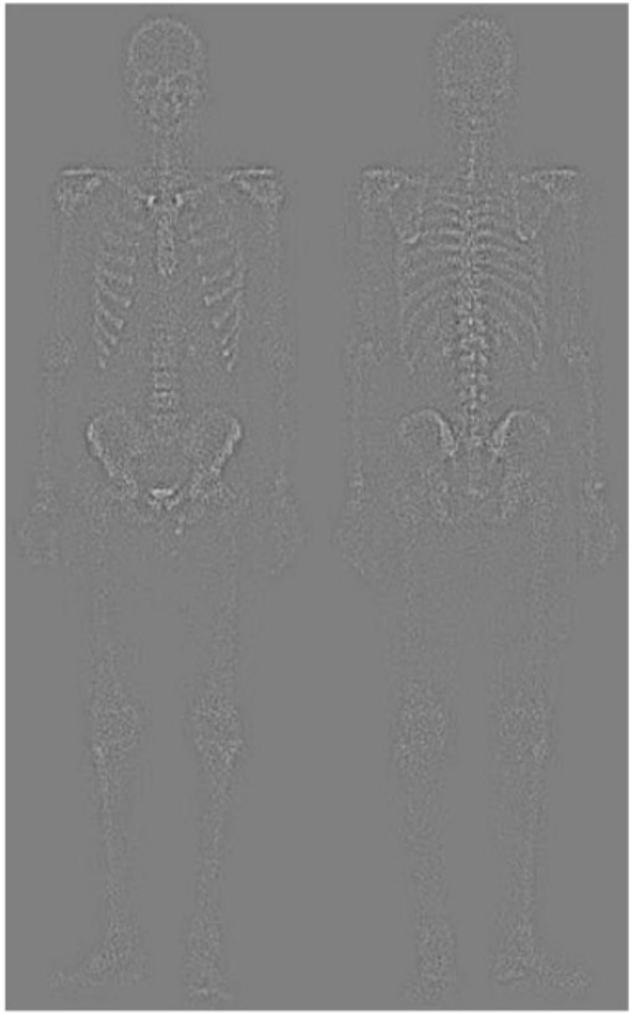
Sobel Operator



Combining Spatial Enhancement



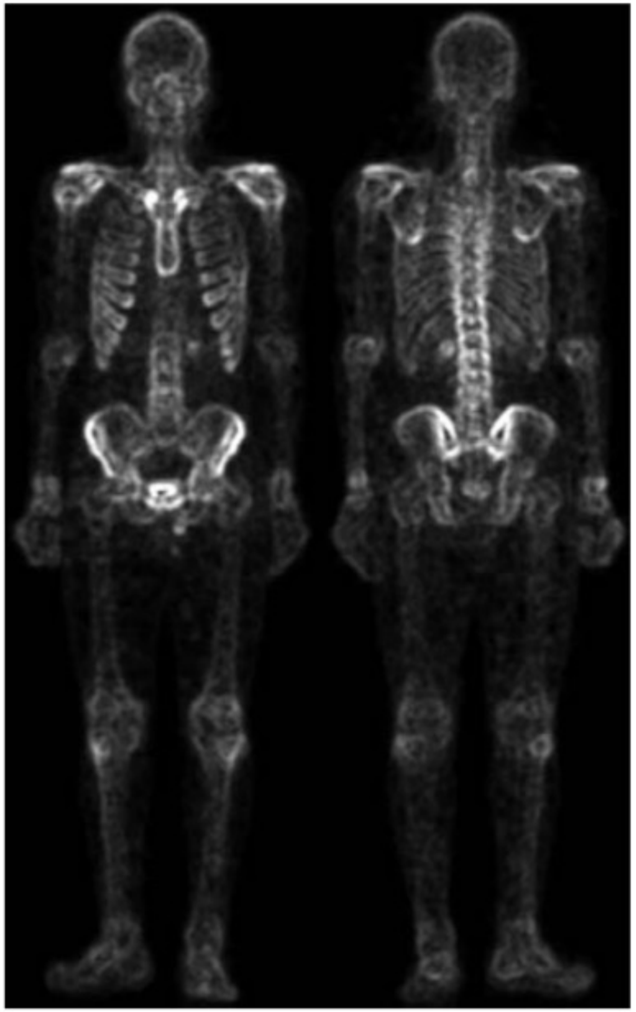
(a) Body scan



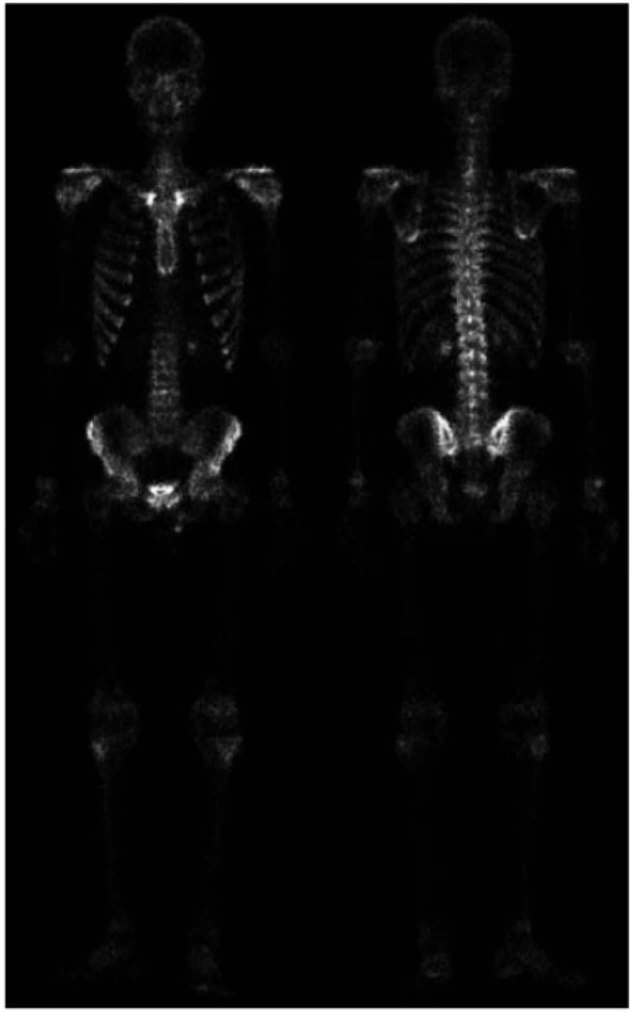
(b) Laplacian

(c) Sharpened (a)+(b)

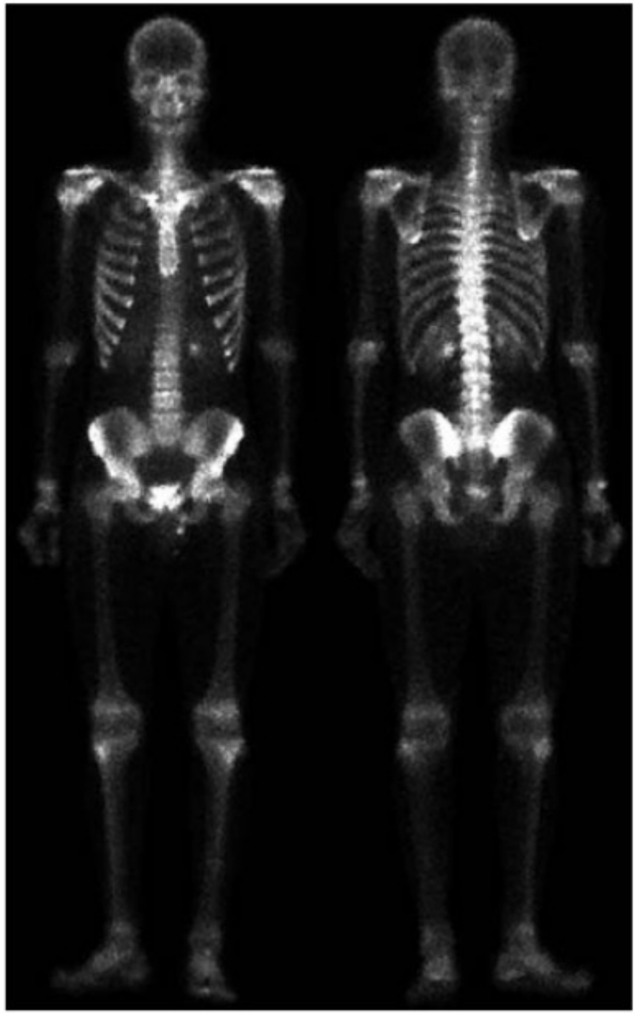
(d) Sobel of (a)



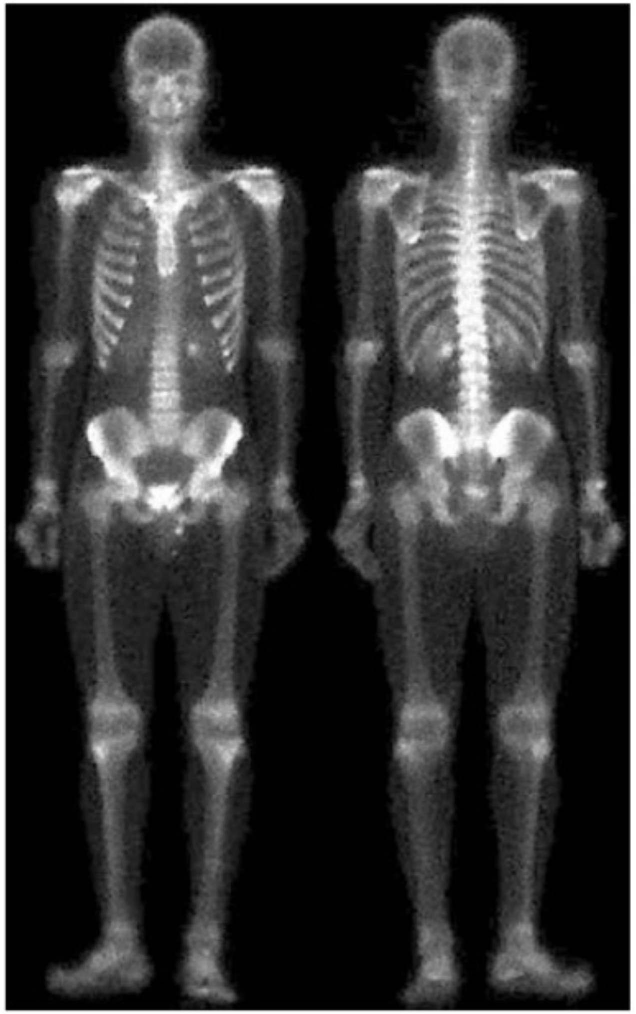
(e) Smoothed gradient



(f) Mask image (c)*(e)



(g) Sharpened (a)+(f)

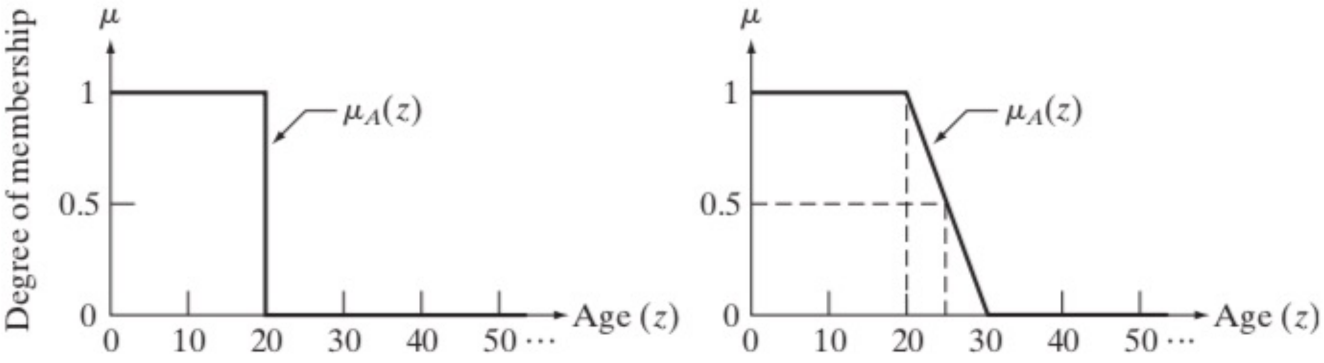


(h) Gamma correction of (g)

Fuzzy Techniques

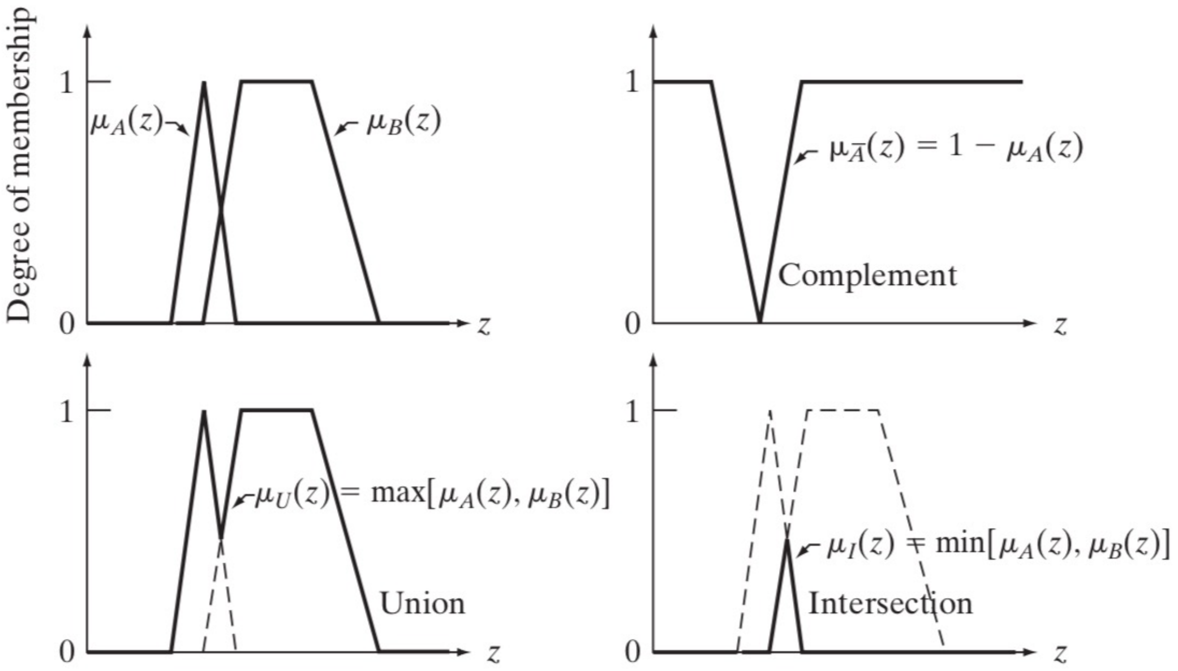
Fuzzy Set Theory

- Introduced by L. A. Zadeh (1965)
- Ordinary set theory
  - $z \in Z$
- Fuzzy set A in Z
  - $A = \{z, \mu_A(z) | z \in Z\}$
  - where $0 \leq \mu_A(z) \leq 1$ (membership function)
  - $\mu_A(z) = 0$ means z is not a member of Z
  - $\mu_A(z) = 1$ means z is a *full* member of Z



Definitions in Fuzzy Set Theory

- Empty set & Equality
  - $\mu_A(z) = 0$ for $\forall z \in Z$
  - $A = B$ iff $\mu_A(z) = \mu_B(z)$
- Complement
  - $\mu_{\bar{A}}(z) = 1 - \mu_A(z)$
- Subset
  - A is subset of B iff $\mu_{A(z)} \leq \mu_{B(z)}$
- Union & Intersection (and, or)
  - $\mu_U(z) = \max[\mu_A(z), \mu_B(z)]$ $(U = A \cup B)$
  - $\mu_I(z) = \min[\mu_A(z), \mu_B(z)]$ $(I = A \cap B)$



Common Membership Function

Triangular:

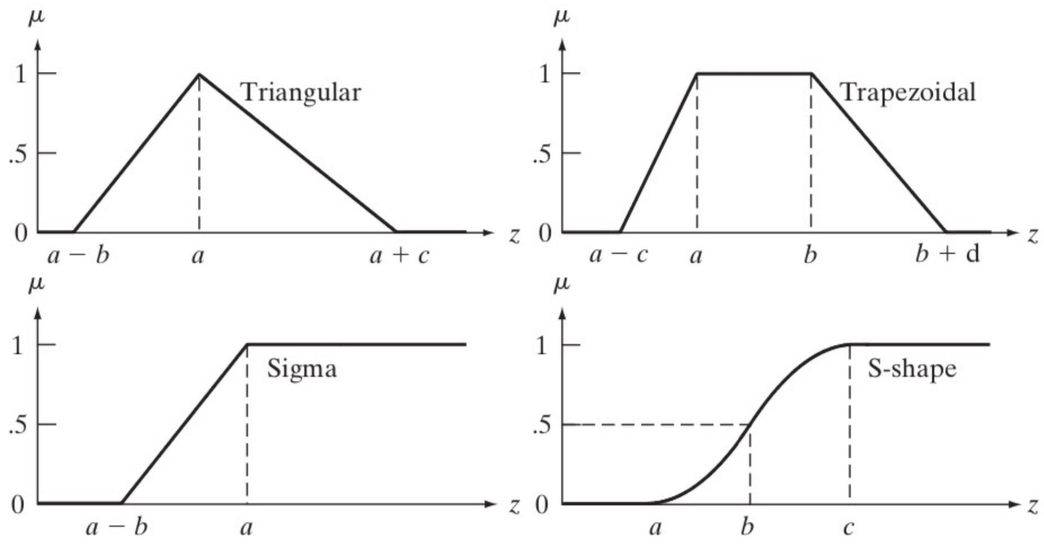
$$\mu(z) = \begin{cases} 1 - (a - z)/b & a - b \le z < a \\ 1 - (z - a)/c & a \le z \le a + c \\ 0 & \text{otherwise} \end{cases}$$

Trapezoidal:

$$\mu(z) = \begin{cases} 1 - (a - z)/c & a - c \le z < a \\ 1 & a \le z < b \\ 1 - (z - b)/d & b \le z \le b + d \\ 0 & \text{otherwise} \end{cases}$$

Sigma:

$$\mu(z) = \begin{cases} 1 - (a - z)/b & a - b \le z \le a \\ 1 & z > a \\ 0 & \text{otherwise} \end{cases}$$



Common Membership Function

S-shape:

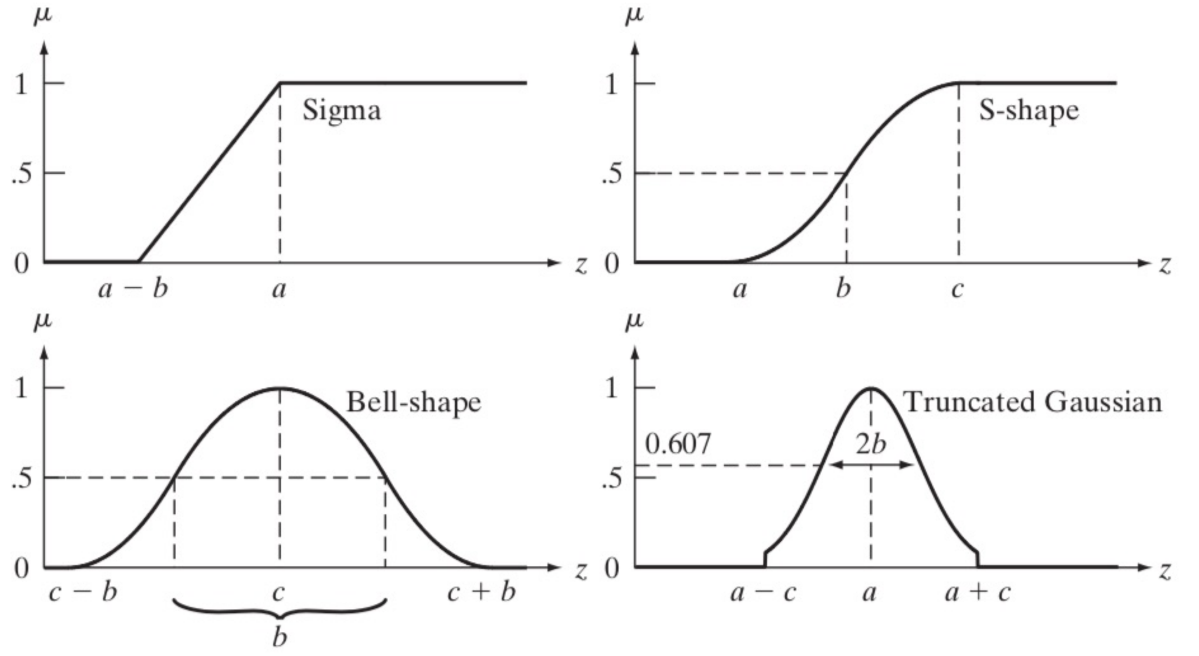
$$S(z; a, b, c) = \begin{cases} 0 & z < a \\ 2\left(\dfrac{z-a}{c-a}\right)^2 & a \le z \le b \\ 1 - 2\left(\dfrac{z-c}{c-a}\right)^2 & b < z \le c \\ 1 & z > c \end{cases}$$

Bell-shape:

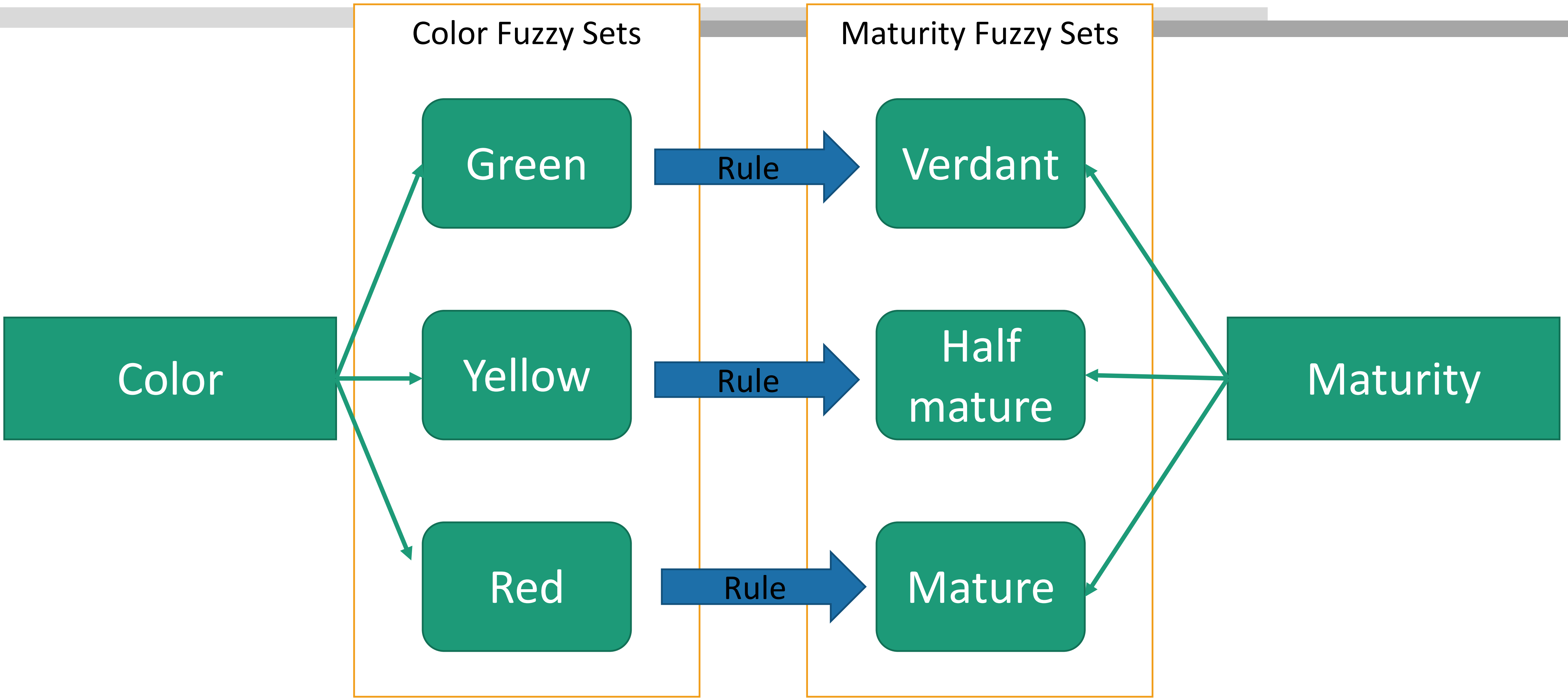
$$\mu(z) = \begin{cases} S(z; c-b, c-b/2, c) & z \le c \\ 1 - S(z; c, c+b/2, c+b) & z > c \end{cases}$$

Truncated Gaussian:

$$\mu(z) = \begin{cases} e^{-\frac{(z-a)^2}{2b^2}} & a - c \le z \le a + c \\ 0 & \text{otherwise} \end{cases}$$



Using Fuzzy Sets: Rule Design



Using Fuzzy Sets: General Steps

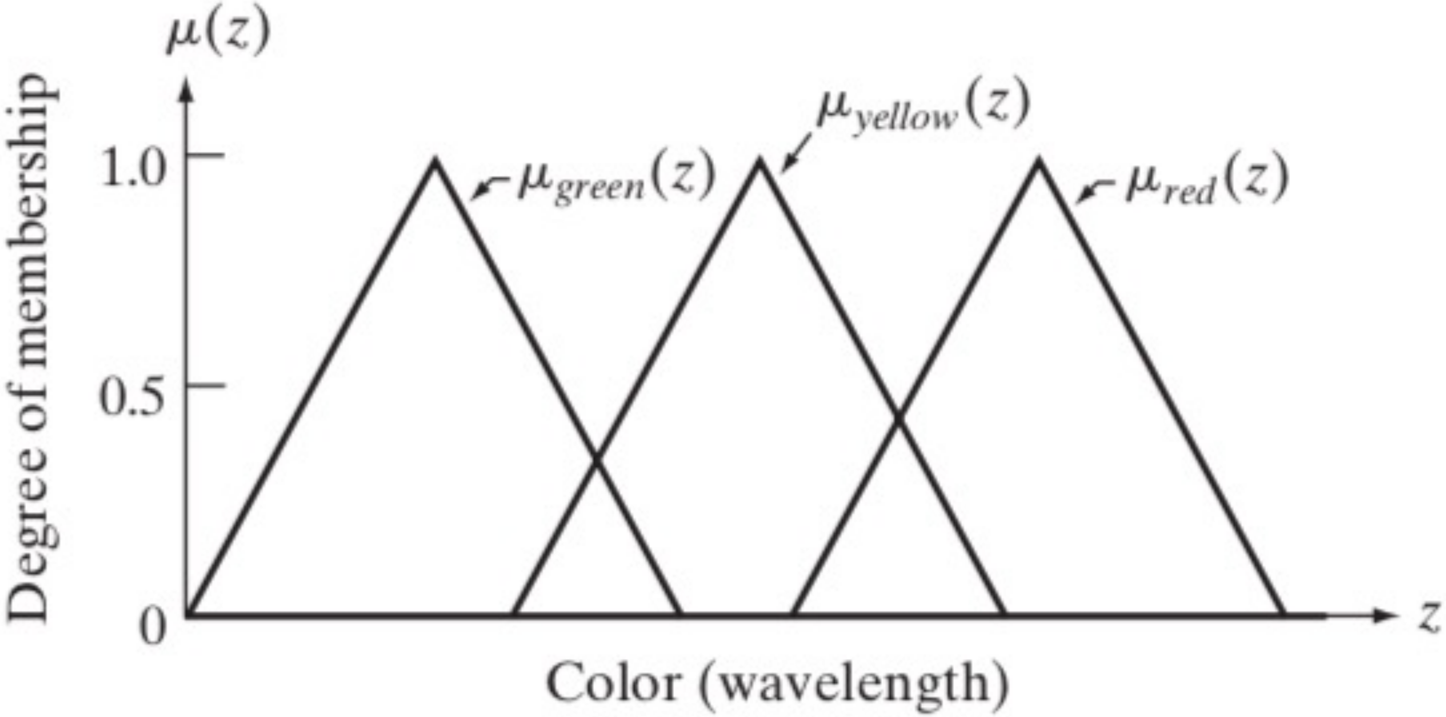
- Rule-based fuzzy logics
  - IF an input value is A and (or) another input value is B, THEN it is C
  - ...
  - Ex) IF the color of a fruit is red AND it is soft, THEN it is mature.
- Fuzzify the inputs
  - Mapping each scalar input to the interval [0,1] using an applicable member function for each rule
  - Ex) Color (spectrum value) -> How much it is the member of “red” set
- Perform any required fuzzy logical operations
  - Ex) Color is red AND it is soft

Using Fuzzy Sets: General Steps

- Apply an implication method for each rule
  - Output is also mapped to [0,1]
  - Ex) How maturity 80% is mature?
  - Use ”AND” rule for implication
  - Ex) How given color is “red” and how given maturity is “mature”.
- Aggregate all rules
  - Merging all rules with “OR” operation
- Defuzzification
  - Computing the center of gravity

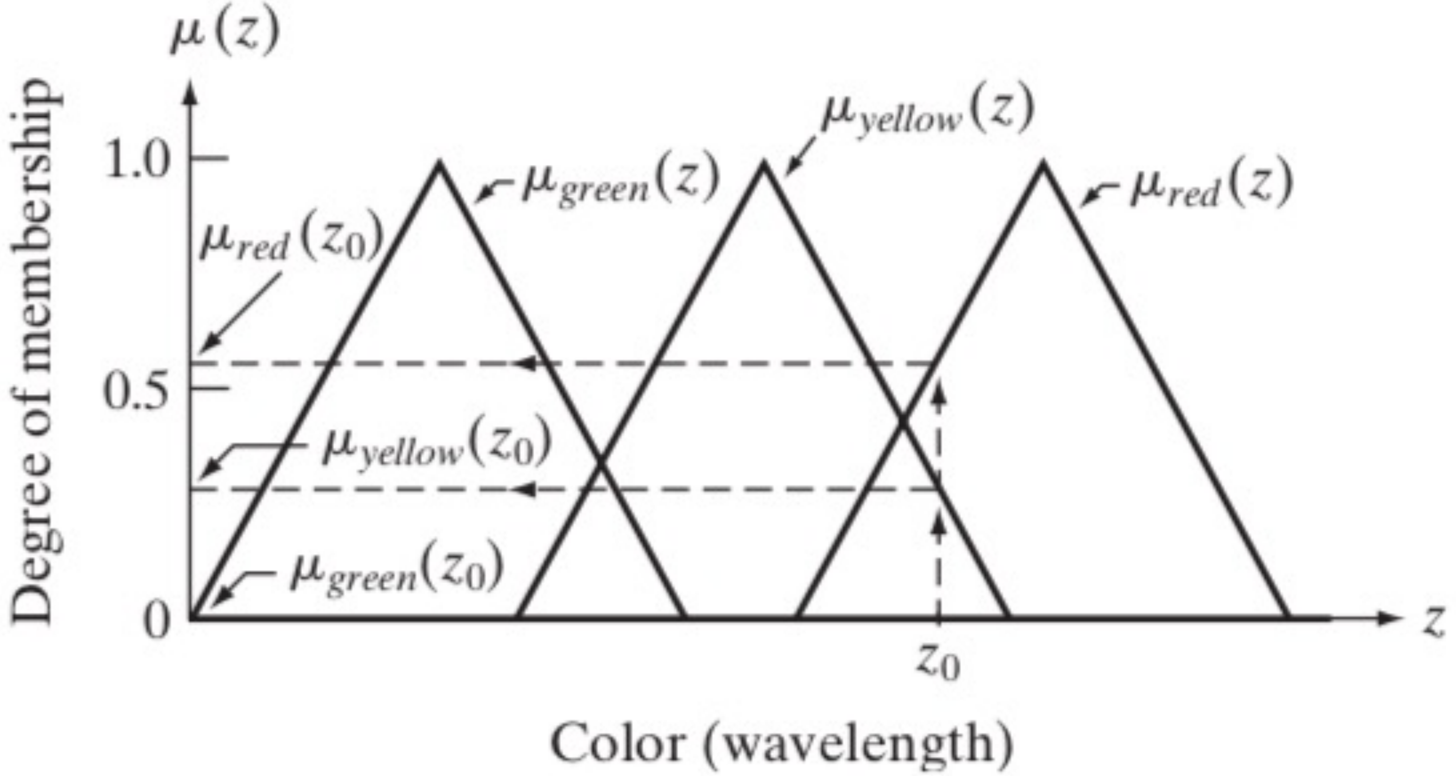
Using Fuzzy Sets: Example

- Fuzzifying fruit color



Using Fuzzy Sets: Example

- Membership of z_o

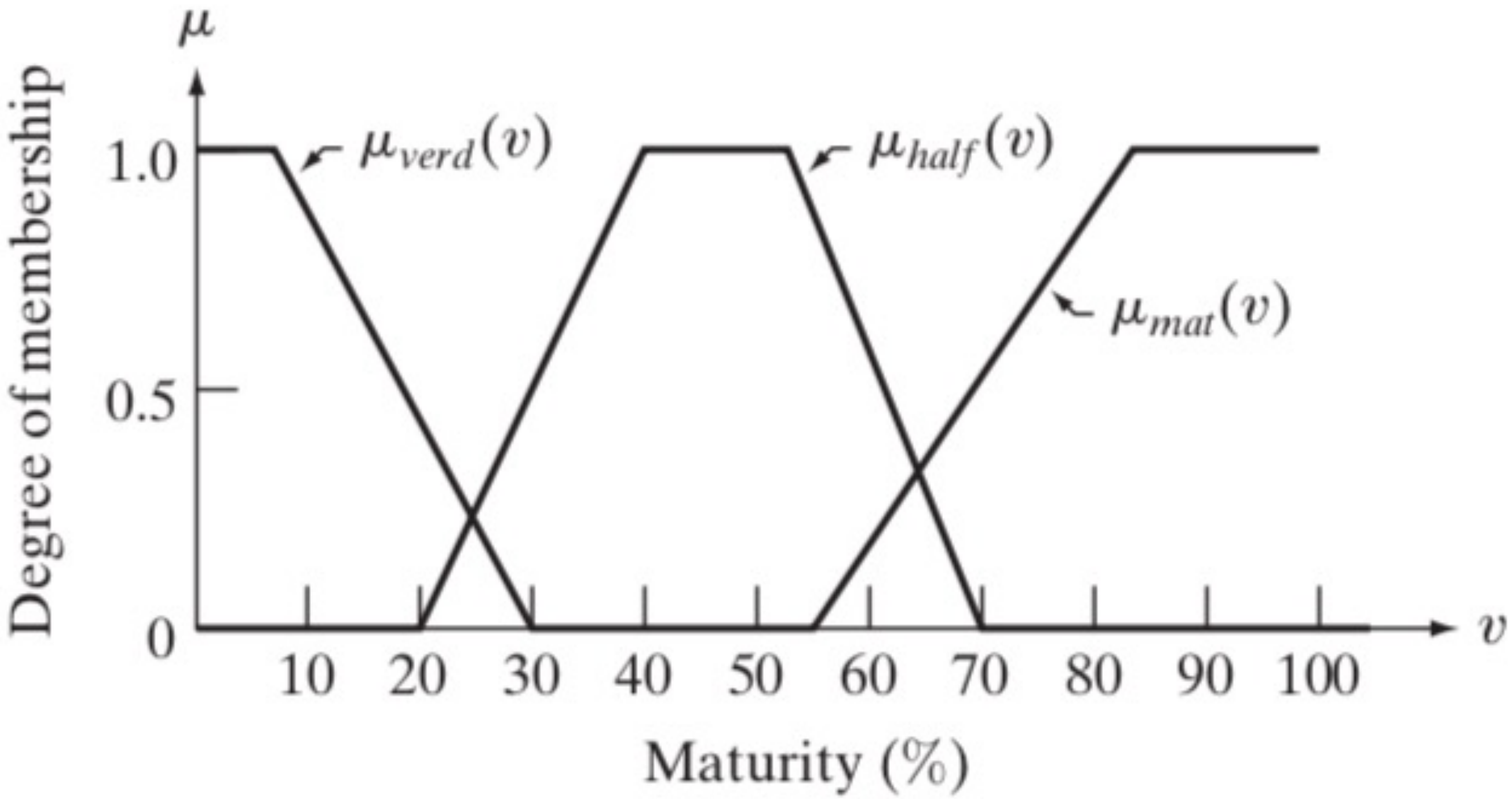


Using Fuzzy Sets: Example

- Problem-specific knowledge
  - R1: IF the color is green, THEN the fruit is verdant
  - R2: IF the color is yellow, THEN the fruit is half-mature
  - R3: IF the color is red, THEN the fruit is mature

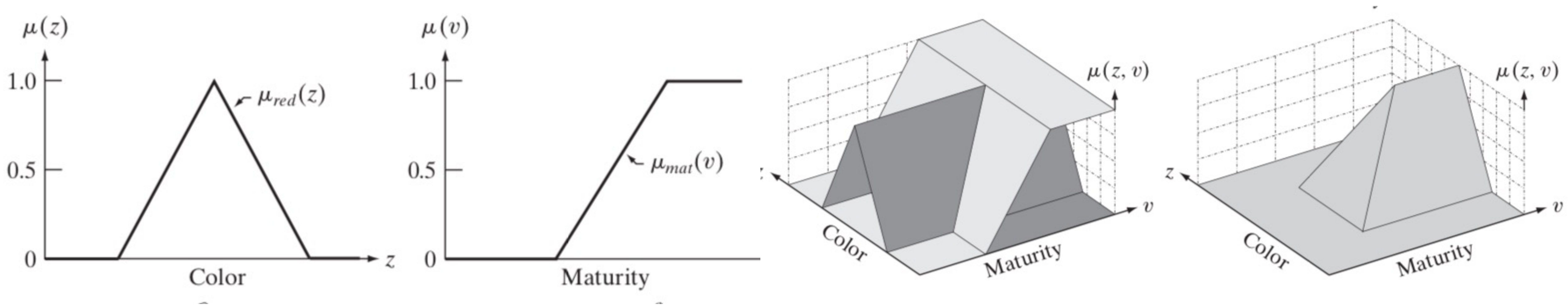
Using Fuzzy Sets: Example

- Output is also fuzzy



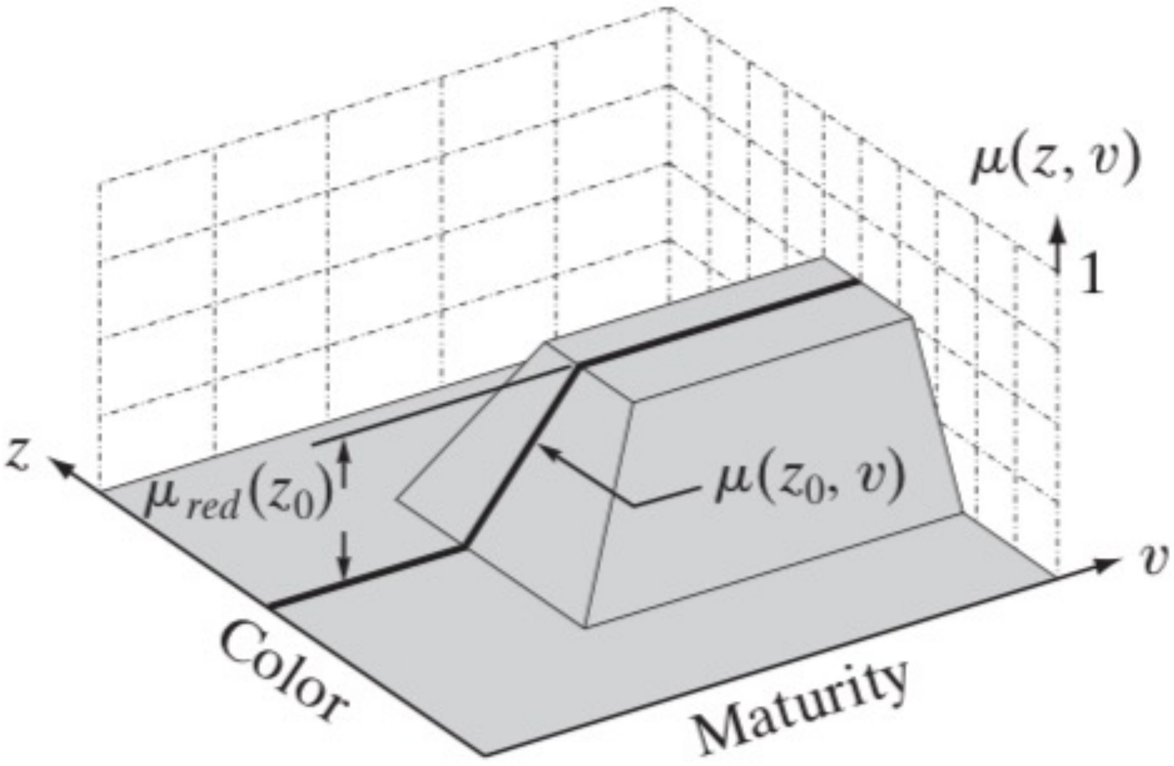
Using Fuzzy Sets: Example

- Implication
  - Logical AND
  - $\mu_3(z, v) = \min\{\mu_{red}(z), \mu_{mat}(v)\}$ (for Rule 3)



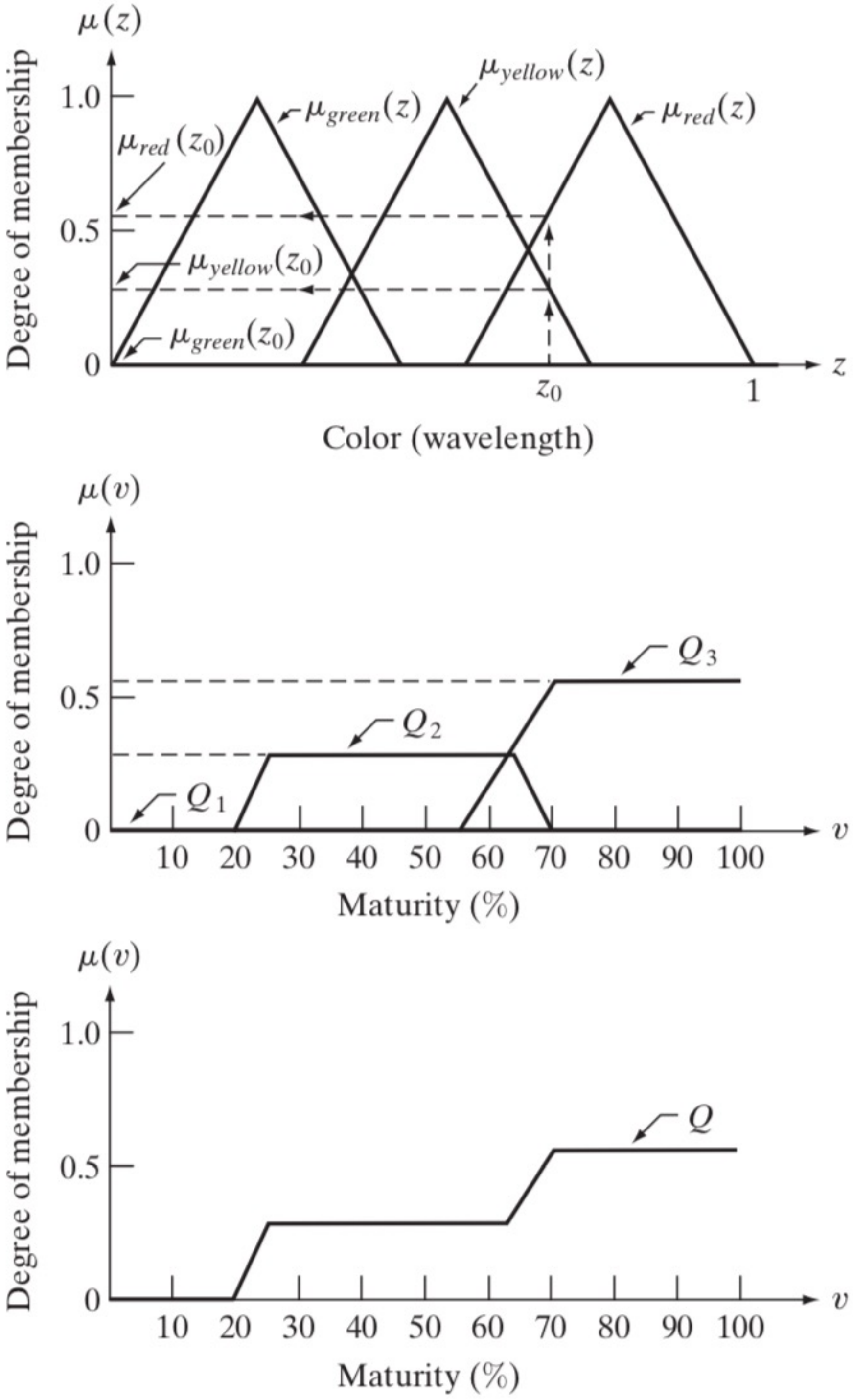
Using Fuzzy Sets: Example

- Fuzzy output of a value (z_0) due to rule R3:
  - $Q_3(v) = \min\{\mu_{red}(z_0), \mu_3(z_0, v)\}$
- Similarly
  - $Q_1(v) = \min\{\mu_{green}(z_0), \mu_1(z_0, v)\}$
  - $Q_2(v) = \min\{\mu_{yellow}(z_0), \mu_2(z_0, v)\}$
- Aggregation
  - $Q = Q_1 \ OR \ Q_2 \ OR \ Q_3$
  - $Q(v) = \max_r\{\min_s\{\mu_s(z_0), \mu_r(z_0, v)\}\}$



Using Fuzzy Sets

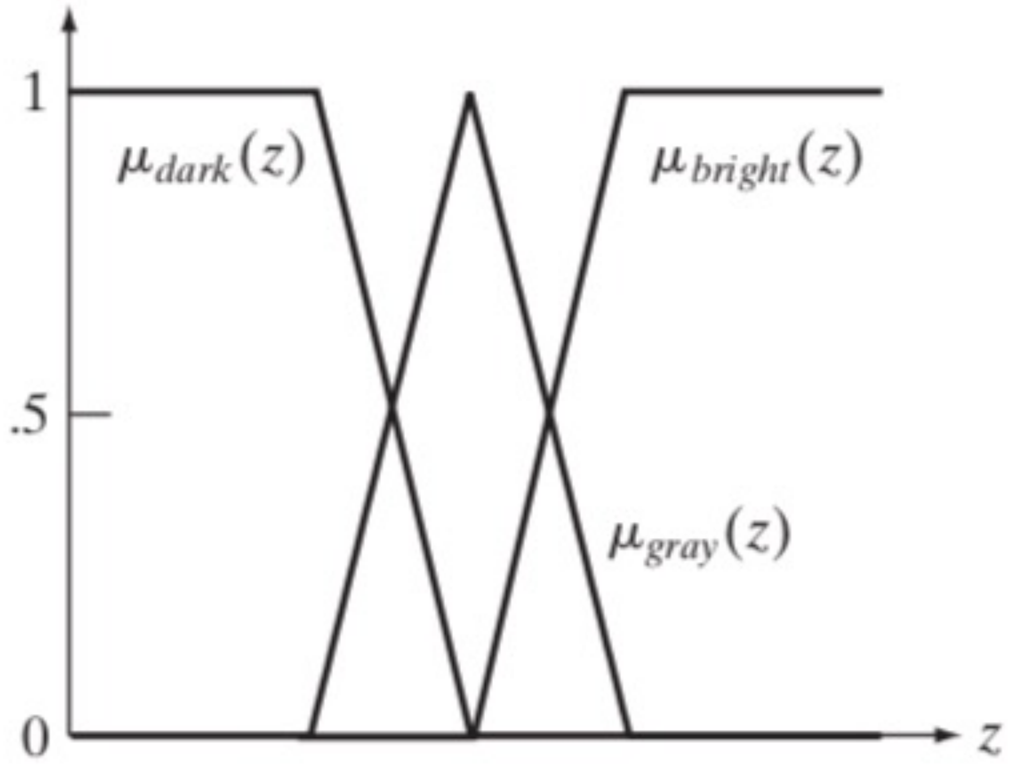
- Final ”Fuzzy” result of Z0
- Defuzzyfication
  - $v_0 = \frac{\sum_{v=1}^{K} vQ(v)}{\sum_{v=1}^{K} Q(v)}$

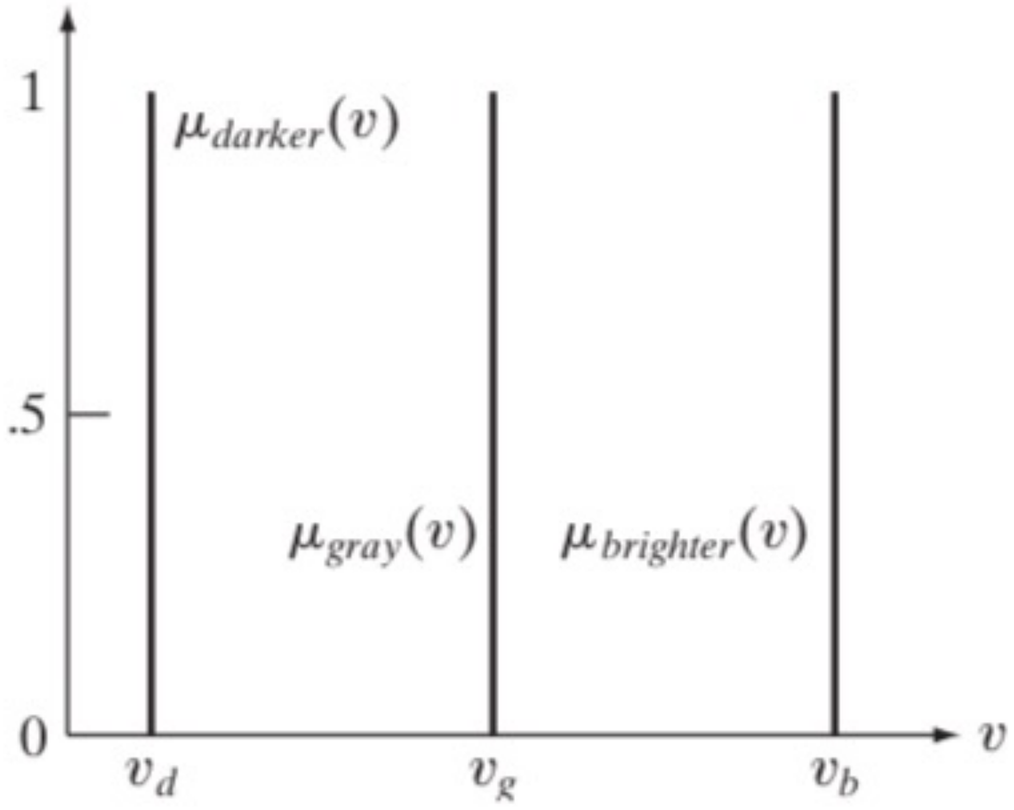


Contrast Enhancement with Fuzzy Sets

- Example Rule for 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
- Fuzziness
  - How input pixel intensity is a member of “Dark” set?
  - How output intensity is a member of “Darker” set?

Designing Membership Function





Implication

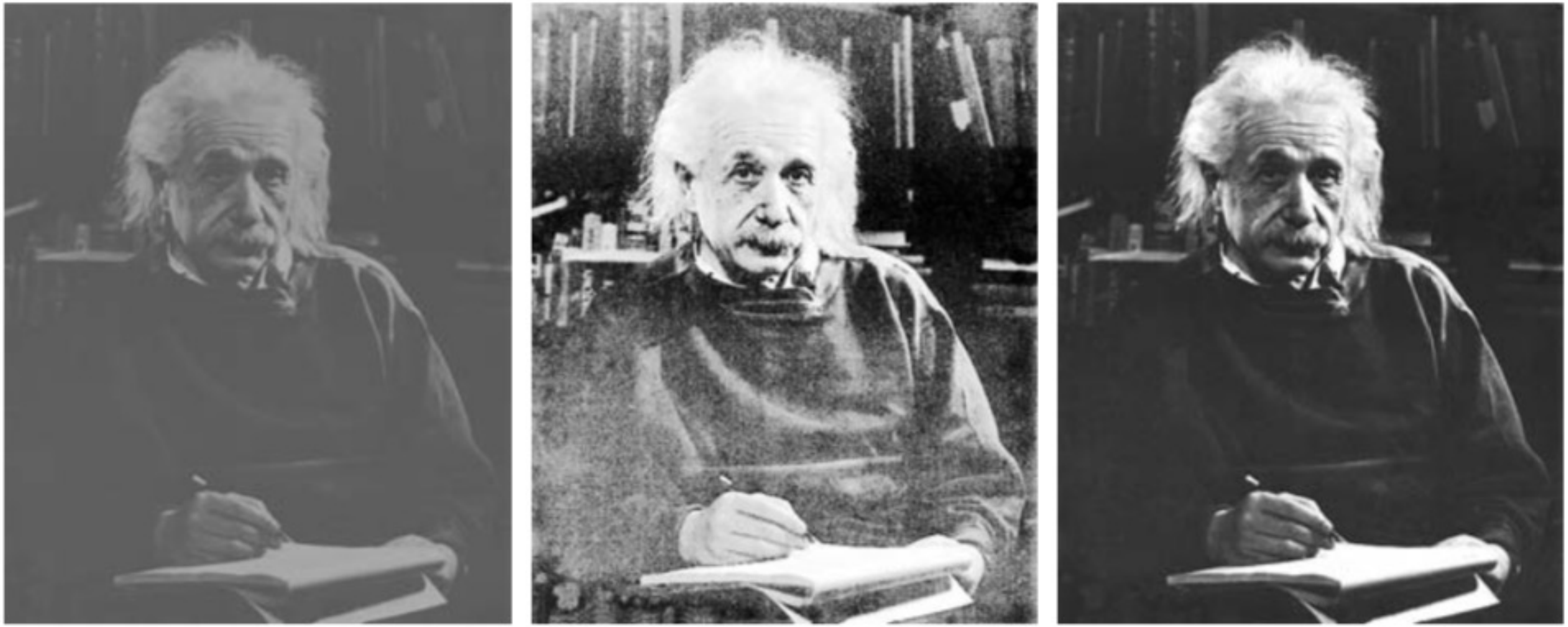
- Rule
  - $\mu_{darker}(v) = \begin{cases} 1 \; where \; v = v_d \\ 0 \; where \; v \neq v_d \end{cases} \cdots$
- Implication
  - $\mu_1(z, v) = \min\{\mu_{dark}(z), \mu_{darker}(v)\} \quad \Rightarrow Q_1(v) = \min\{\mu_{dark}(z_0), \mu_1(z_0, v)\}$
  - $\mu_2(z, v) = \min\{\mu_{gray}(z), \mu_{gray}(v)\} \quad \Rightarrow Q_2(v) = \min\{\mu_{gray}(z_0), \mu_2(z_0, v)\}$
  - $\mu_3(z, v) = \min\{\mu_{bright}(z), \mu_{brighter}(v)\} \Rightarrow Q_3(v) = \min\{\mu_{bright}(z_0), \mu_3(z_0, v)\}$
  - Note:
    - $\mu_1(z, v) = \mu_{dark}(z)$, when $v = v_d$ and 0, otherwise
    - Therefore, $Q_1(v) = \mu_{dark}(z_0)$ only when $v = v_d$

Aggregation and Defuzzyfication

- Aggregation
  - $Q(v) = \max\{Q_i(v)\}$
- Defuzzyfication
  - $v_0 = \frac{\sum_{v=1}^{K} vQ(v)}{\sum_{v=1}^{K} Q(v)}$
  - Note that $Q(v) = \begin{cases} \mu_{dark}(z_0) & if\ v = v_d \\ \mu_{gray}(z_0) & if\ v = v_g \\ \mu_{bright}(z_0) & if\ v = v_b \\ 0 & otherwise \end{cases}$
  - Therefore, $v_0 = \frac{v_d\mu_{dark}(z_0)+v_g\mu_{gray}(z_0)+v_b\mu_{bright}(z_0)}{\mu_{dark}(z_0)+\mu_{gray}(z_0)+\mu_{bright}(z_0)}$

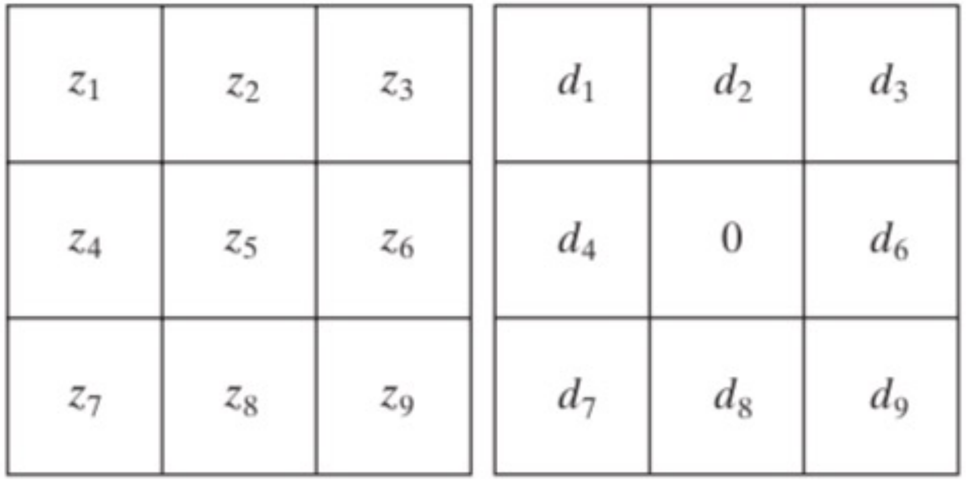
Result

- Setting output membership function
  - $v = 0$, $v = 127$, and $v = 255$



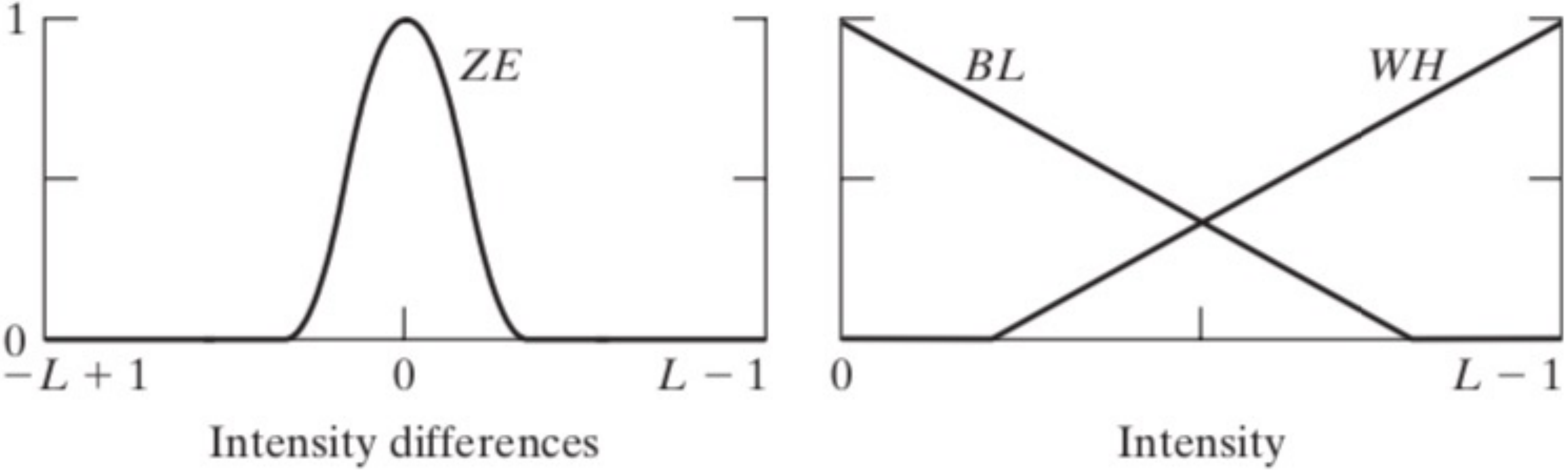
Boundary Extraction with Fuzzy Sets

- Spatial filtering with fuzzy sets
  - Use neighborhood pixels in the rules
- Boundary extraction
  - Making uniform region white and making their boundary black
- Rules
  - IF d2 is 0 and d6 is 0 THEN z5 is white
  - IF d6 is 0 and d8 is 0 THEN z5 is white
  - IF d8 is 0 and d4 is 0 THEN z5 is white
  - IF d4 is 0 and d2 is 0 THEN z5 is white



Pixel neighborhood   Intensity differences

Membership Functions



Result

Bottomline

- Use case
  - There are many problems that need to be solved based on simple rules
  - However, some logical decision can be not “crisp” (fuzzy.)
  - The output of the logical rules can be also fuzzy.
- Fuzzy Sets
  - By defining membership functions, we can deduce the output from simple steps.