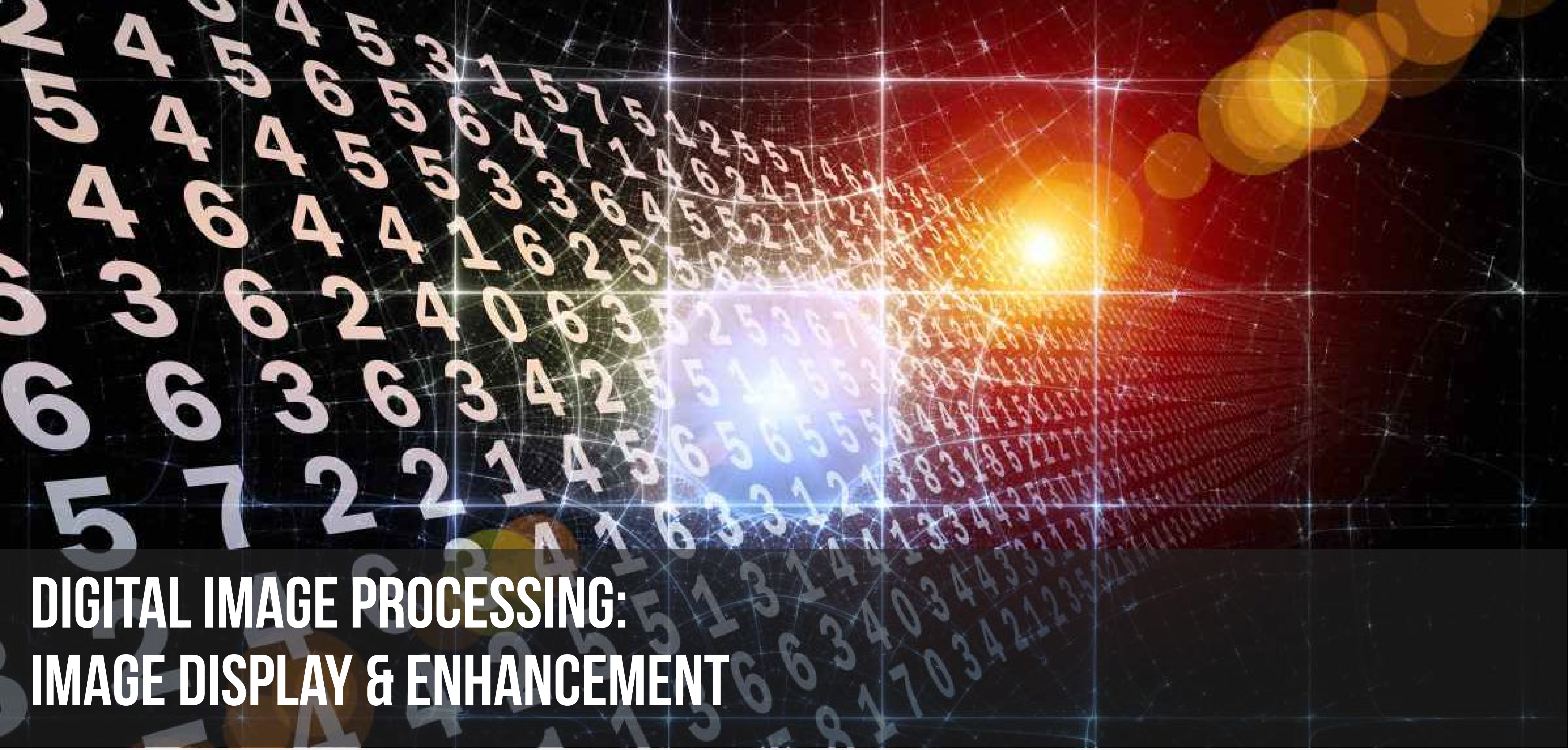
PENGOLAHAN SINYAL DIGITAL

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia



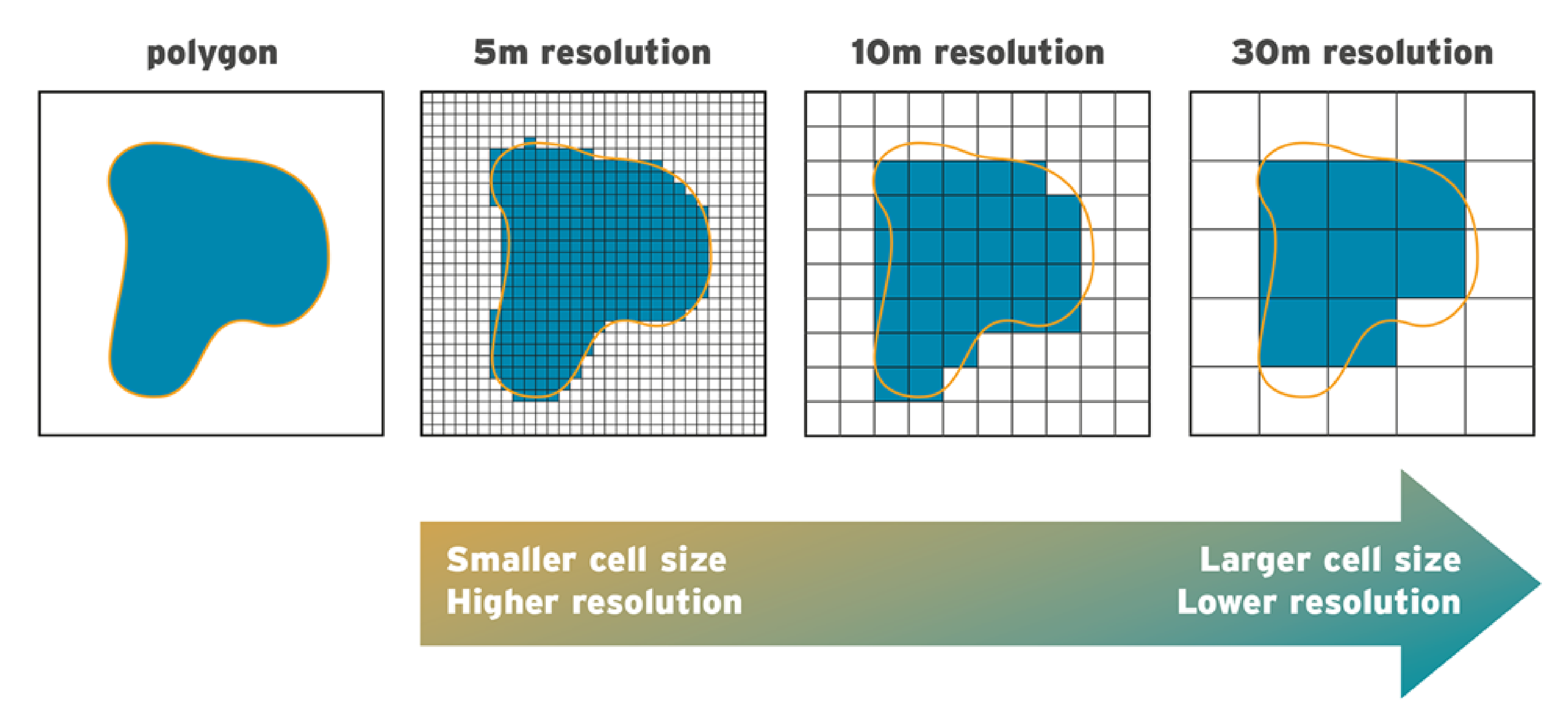
DIGITAL IMAGE PROCESSING:
IMAGE DISPLAY & ENHANCEMENT

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

SPATIAL RESOLUTION

- The spatial resolution of an image is the physical size of a pixel in an image
  - Basically it is the smallest discern able detail in an image



SPATIAL RESOLUTION (CON'T)

- The greater the spatial resolution, the more pixels are used to display the image.
- How to alter the resolution using Matlab?
  - Imresize(x,n)
  - when n=1/2,

$$\begin{array}{|c|c|c|c|c|c|c|} \hline x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & \cdots \\ \hline x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & \cdots \\ \hline x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & \cdots \\ \hline x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & \cdots \\ \hline x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & \cdots \\ \hline x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & \cdots \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \hline \end{array} \rightarrow \text{imresize(x,1/2)} \rightarrow \begin{array}{|c|c|c|c|} \hline x_{22} & x_{24} & x_{26} & \cdots \\ \hline x_{42} & x_{44} & x_{46} & \cdots \\ \hline x_{62} & x_{64} & x_{66} & \cdots \\ \hline \vdots & \vdots & \vdots & \\ \hline \end{array}$$

SPATIAL RESOLUTION (CON'T)

- when n=2,

x_{22}	x_{24}	x_{26}	$\cdots$
x_{42}	x_{44}	x_{46}	$\cdots$
x_{62}	x_{64}	x_{66}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	

→ imresize(imresize(x,1/2),2); →

x_{22}	x_{22}	x_{24}	x_{24}	x_{26}	x_{26}	$\cdots$
x_{22}	x_{22}	x_{24}	x_{24}	x_{26}	x_{26}	$\cdots$
x_{42}	x_{42}	x_{44}	x_{44}	x_{46}	x_{46}	$\cdots$
x_{42}	x_{42}	x_{44}	x_{44}	x_{46}	x_{46}	$\cdots$
x_{62}	x_{62}	x_{64}	x_{64}	x_{66}	x_{66}	$\cdots$
x_{62}	x_{62}	x_{64}	x_{64}	x_{66}	x_{66}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

- Commands that generate the images of slide2
  - imresize(imresize(x,1/2),2);
  - imresize(imresize(x,1/4),4);
  - imresize(imresize(x,1/8),8);

QUANTIZATION

- Quantization refers to the number of greyscales used to represent the image.
  - A set of n quantization levels comprises the integers 0, 1, 2, …, n-1
  - 0 and n-1 are usually black and white respectively, with intermediate levels rendered in various shades of grey.
  - Quantization levels are commonly referred to as grey levels

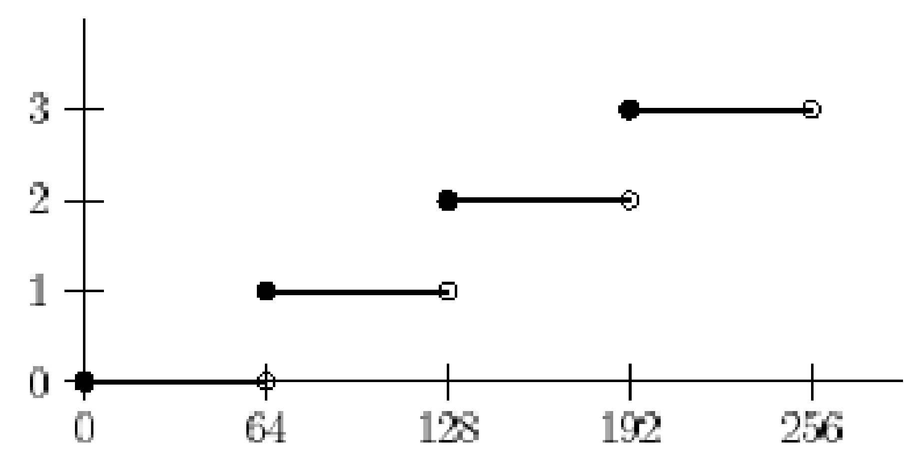
- Sometimes the range of values spanned by the gray levels is called the dynamic range of an image.

QUANTIZATION (CON'T)

- n is usually an integral power of 2, $n = 2^b$, where b is the number of bits used for quantization
  - Typically b=8 => 256 possible grey levels.
  - If b=1, then there are only two values: 0 and 1 (a binary image).

- Uniform quantization:

```
y=grayslice(x,4);
imshow(y, gray(4));
```



A mapping for uniform quantization

QUANTIZATION: USING "GRAYSLICE"

image quantized to 8 greyscales

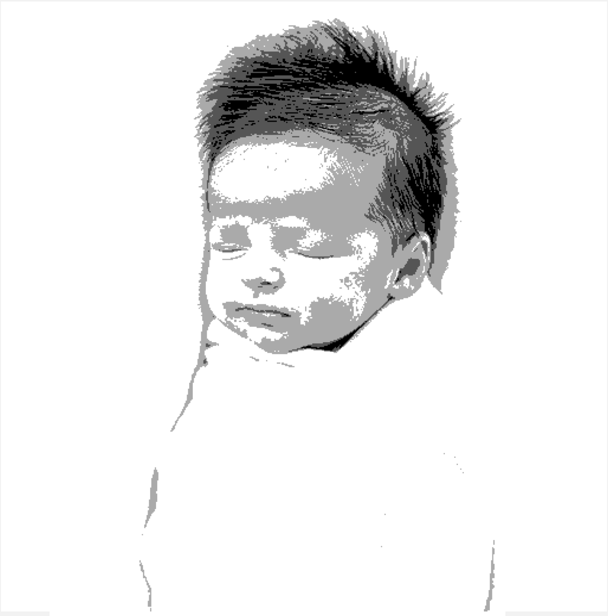


image quantized to 4 greyscales

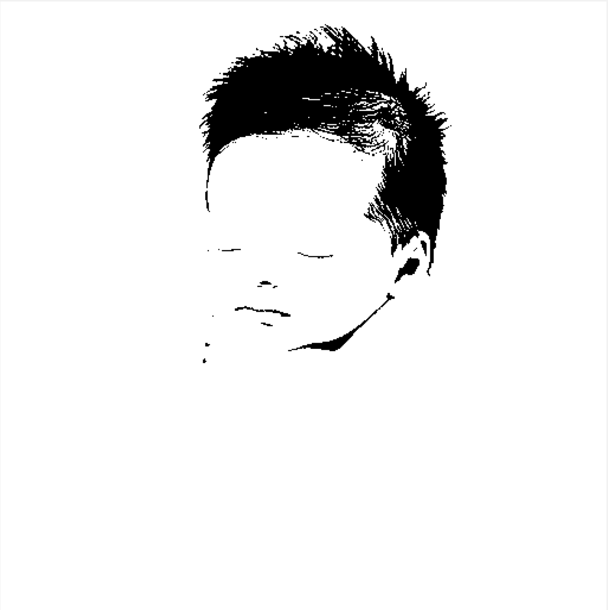


image quantized to 2 greyscales

METHOD 1: DITHERING

- Without dithering, visible contours can be detected between two levels. Our visual system is particularly sensitive to this.

- By adding random noise to the original image, we break up the contouring. The quantization noise itself remains evenly distributed about the entire image.

- The purpose:
  - The darker area will contain more black than white
  - The light area will contain more white than black

- Method: Compare the image to a random matrix d

DITHERING: HALFTONE

- Using one standard matrix

$$D = \begin{bmatrix} 0 & 128 \\ 192 & 64 \end{bmatrix}$$

- Generate matrix d, which is as big as the image matrix, by repeating D.
- The halftone image is

$$p(i,j) = \begin{cases} 1 & \text{if } x(i,j) > d(i,j) \\ 0 & \text{if } x(i,j) \le d(i,j) \end{cases}$$

```
>> r=repmat(D,256,256)
>> x2=x>r; imshow(x2)
```

DITHERING: GENERAL

- Example: 4 levels
- Step1: First quantize by dividing grey value x(i,j) by 85 (=255/3)

$$q(i,j) = \lfloor x(i,j)/85 \rfloor \qquad (0 \le q(i,j) \le 3)$$

- Suppose now that our replicated dither matrix d(i,j) is scaled so that its values are in the range 0 – 85
- The final quantization level p(i,j) is

$$p(i,j) = q(i,j) + \begin{cases} 1 & \text{if } x(i,j) - 85q(i,j) > d(i,j) \\ 0 & \text{if } x(i,j) - 85q(i,j) \le d(i,j) \end{cases}$$

DITHERING: GENERAL

```
x = imread('newborn.jpg');
x = rgb2gray(x);
% Dither to 4 greylevels
D=[0 56;84 28];
r=repmat(D,256,256);
x=double(x);
q=floor(x/85);
x4=q+(x-85*q>r);
subplot(211); imshow(uint8(85*x4));

% Dither to 8 grey levels
clear r x4
D=[0 24;36 12];
r=repmat(D,256,256);
x=double(x);
q=floor(x/37);
x4=q+(x-37*q>r);
subplot(212); imshow(uint8(37*x4));
```

4 output greyscales

8 output greyscales

METHOD2: ERROR DIFFUSION

- Quantization error: The difference between the original gray value and the quantized value.

- If quantized to 0 and 255, Intensity value closer to the center (128) will have higher error.

METHOD2: ERROR DIFFUSION

- Goal: spread this error over neighboring pixels.

- Method: sweeping from left->right, top->down, for each pixel
  - Perform quantization
  - Calculate the quantization error
  - Spread the error to the right and below neighbors

$$E = \begin{cases} p(i,j) & \text{if } p(i,j) < 128 \\ p(i,j) - 255 & \text{if } p(i,j) \geq 128 \end{cases}$$

ERROR DIFFUSION FORMULAE

- To spread the error, there are several formulae.
- Weights must be normalized in actual implementation.

		$+\frac{7}{16}E$
$+\frac{3}{16}E$	$+\frac{5}{16}E$	$+\frac{1}{16}E$

Floyd-Steinberg

ERROR DIFFUSION FORMULAE

- Matlab dither function implements the Floyd-Steinberg error diffusion

		current pixel	7	5
3	5	7	5	3
1	3	5	3	1

Jarvis-Judice-Ninke

		current pixel	8	4
2	4	8	4	2
1	2	4	2	1

Stucki

SUMMARY

- Quantization is necessary when the display device can handle fewer grayscales.
- The simplest method (least pleasing) is uniform quantization.
- The 2nd method is dithering by comparing the image to a random matrix.
- The best method is error diffusion by spreading the error over neighboring pixels.
- The chapter is from McAndraw. A copy will be placed online.

IMAGE ENHANCEMENT

- Two main categories:
  - Point operations: pixel’s gray value is changed without the knowledge of its surroundings. E.g. thresholding.
  - Neighborhood processing: The new gray value of a pixel is affected by its small neighborhood. E.g. smoothing.

- We start with point operations.

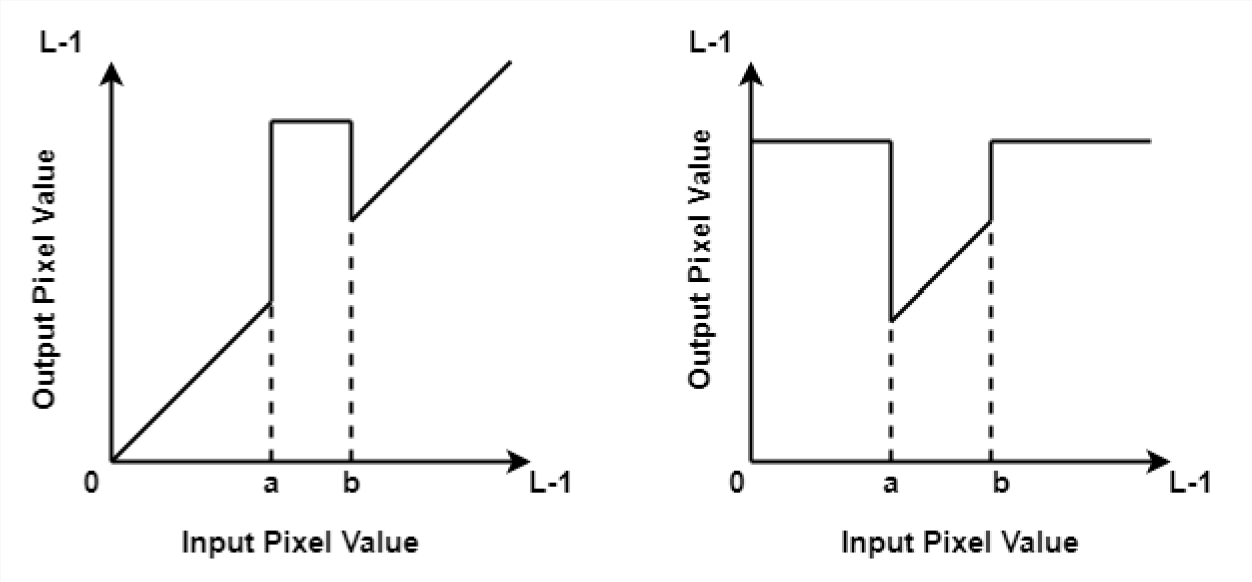
Gray-Level Slicing

- Highlighting a specific range of gray levels.

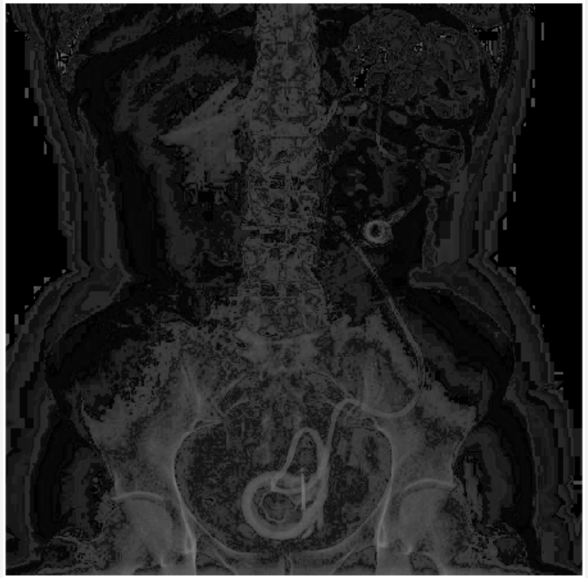
- Applications:
  - Enhancing specific features, s.t. masses of water in satellite images
  - Enhancing flows in X-ray images

- Two basic approaches:
  - High value for all gray levels in the range of interest, and low value outside the range.
  - High value for all gray levels in the range of interest, but background unchanged.

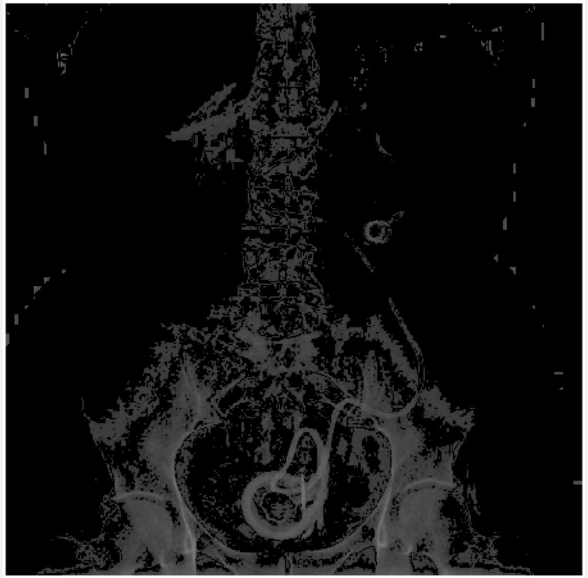
GRAY-LEVEL SLICING



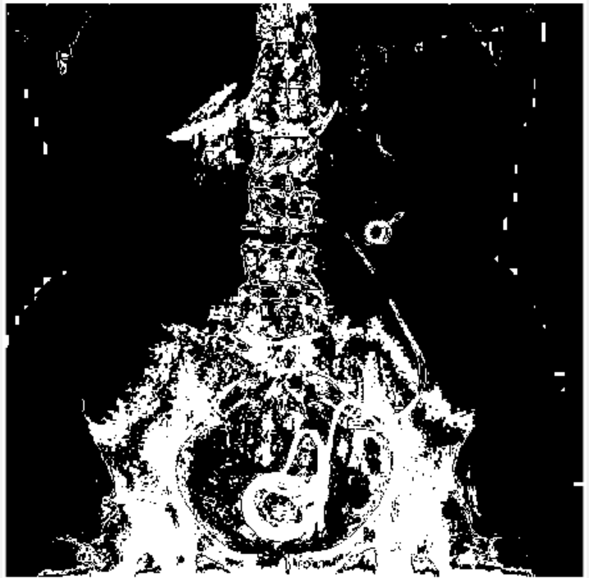
Original image



Graylevel slicing with background



Graylevel slicing without background



BIT-PLANE SLICING

- Highlighting the contribution made by specific bits.
- Bit-plane 0 is the least significant, containing all the lowest order bits.
- Bit-plane 7 is the most significant, containing all the highest order bits.
- Higher-order bits contain visually significant data
- Lower-order bits contain subtle details.
- One application is image compression

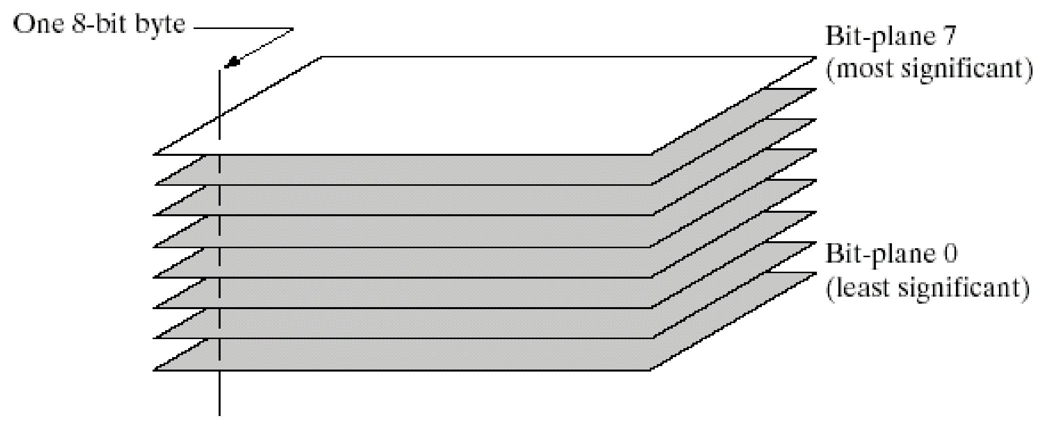


FIGURE 3.12 Bit-plane representation of an 8-bit image.

DEMO: A FRACTAL IMAGE

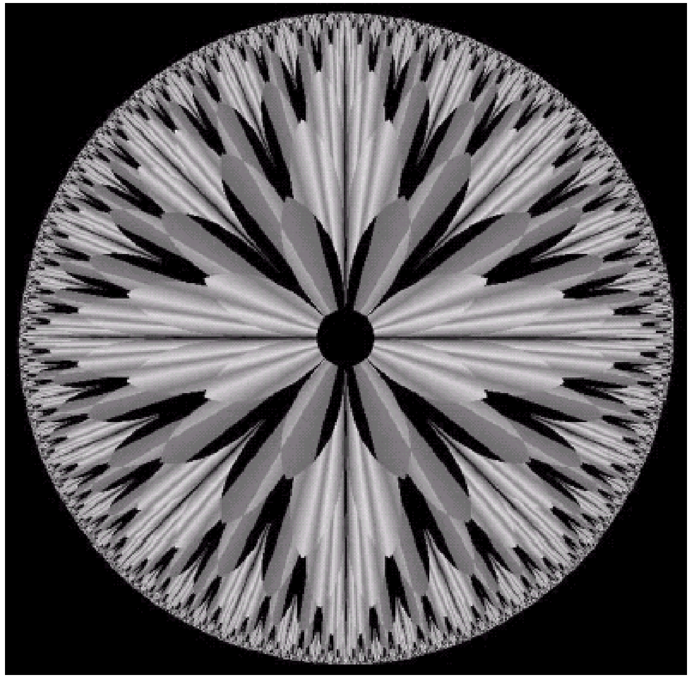


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

EIGHT BIT PLANES OF THE IMAGE

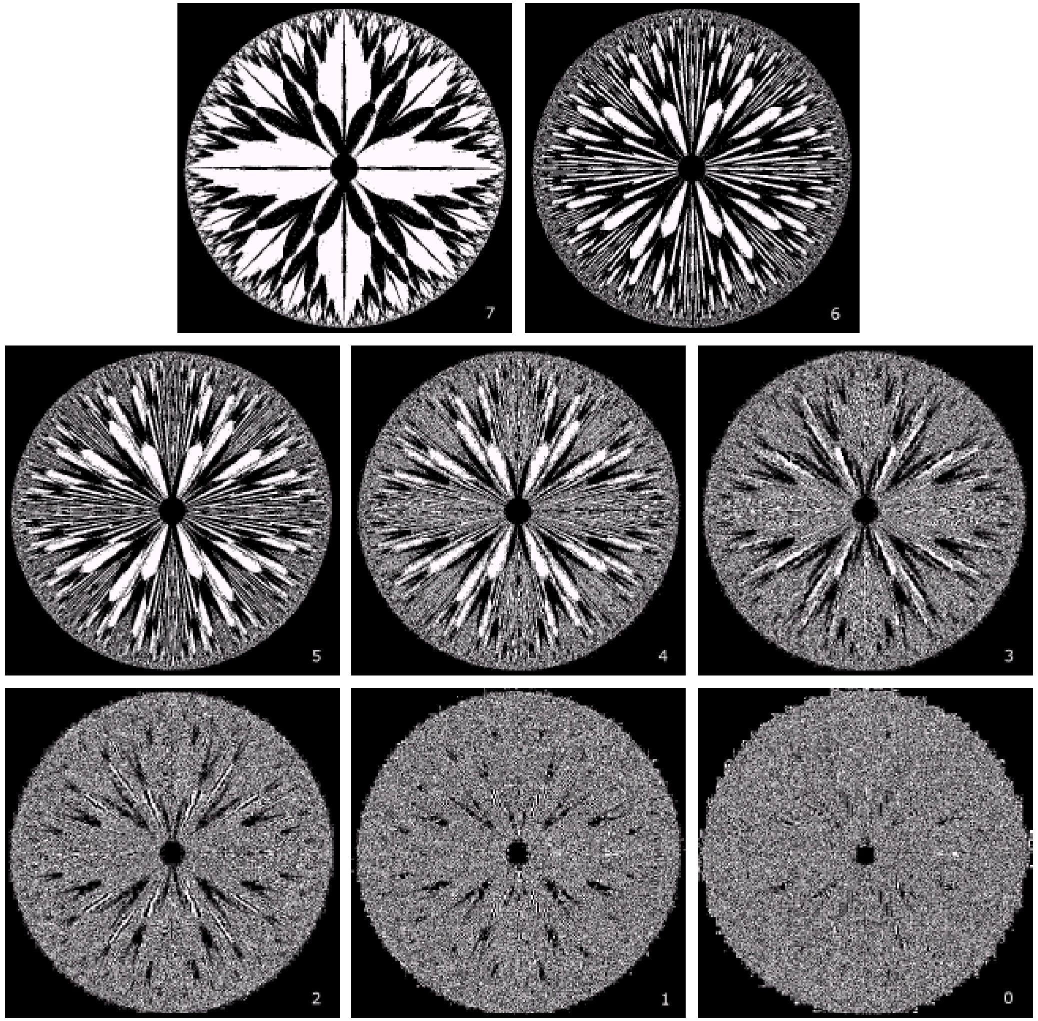


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

ARITHMETIC OPERATIONS

- May perform arithmetic operations (+,-,*,/,neg) on an image.
- Always make sure the new value, as well as the intermediate value, is within the intensity range.
- Operations are straightforward.
- Matlab commands: *imadd*, *imsubtract*, etc..

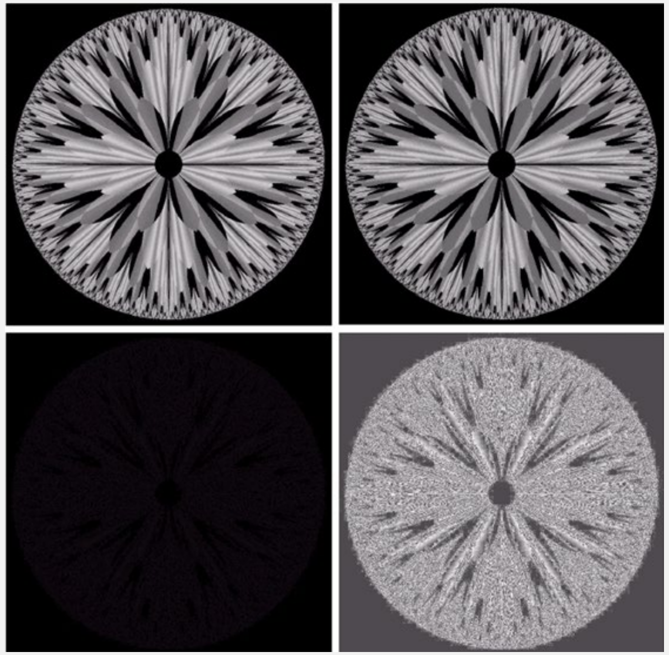
IMAGE SUBTRACTION

- A more interesting arithmetic operation is pixel-wise subtraction of two images.

- Refer to the fractal image again.

$$g(x, y) = f(x, y) - h(x, y)$$

original



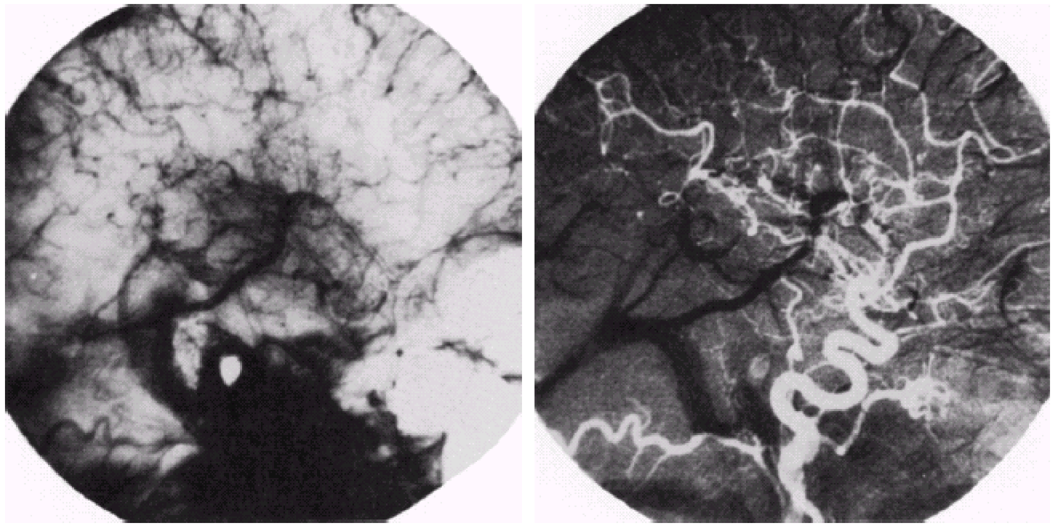
4 lower-order bit planes zeroed out

difference

contrast enhanced

IMAGE SUBTRACTION

- Mask mode radiography
  - The mask image is taken before injection of the contrast medium into patient's bloodstream
  - Areas different between 2 images appear enhanced.



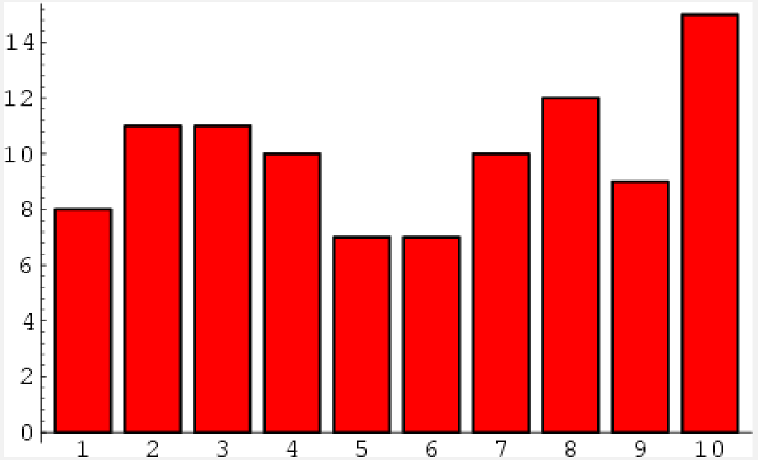
a b

FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

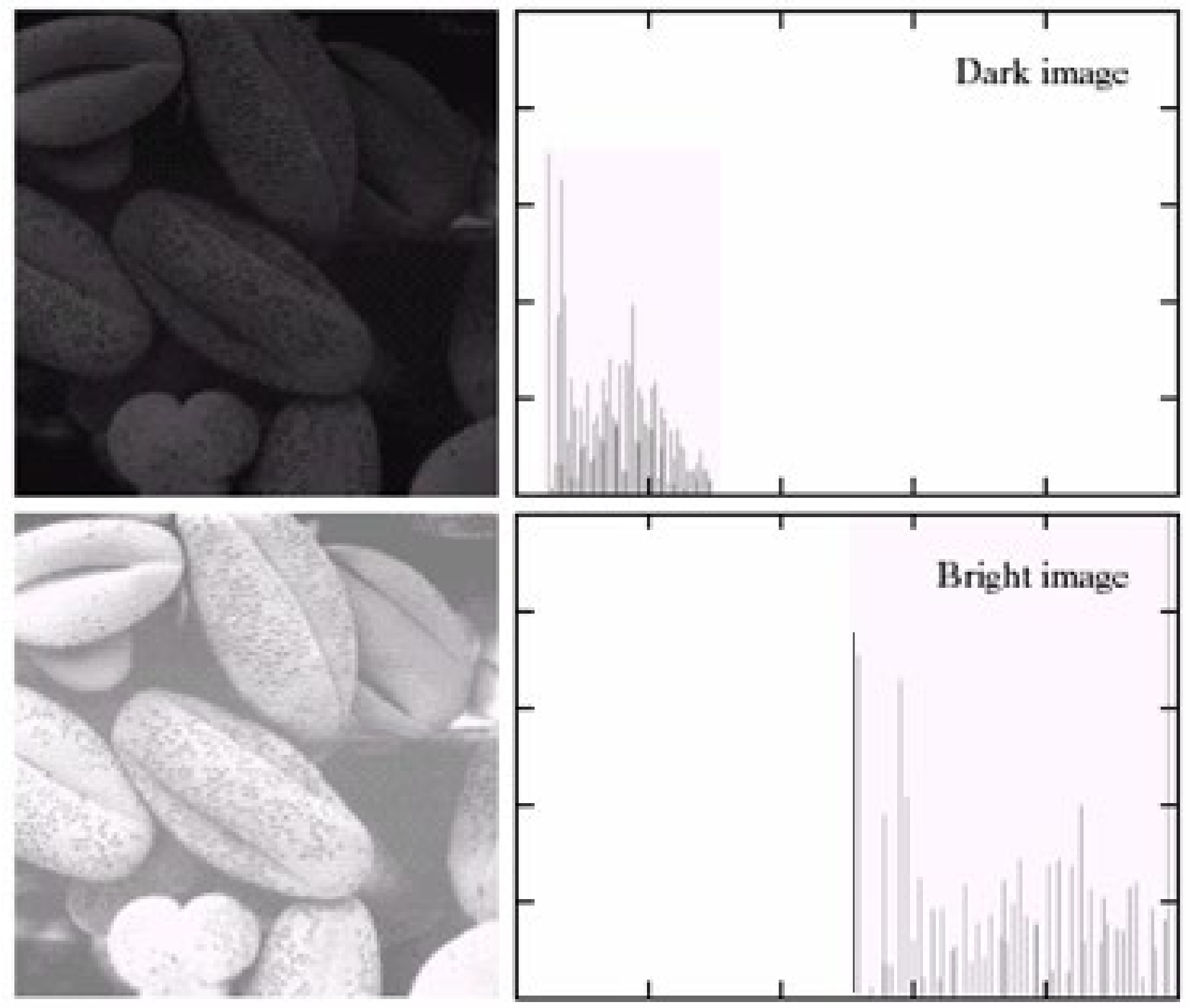
Mask: h(x,y) Result: g(x,y)

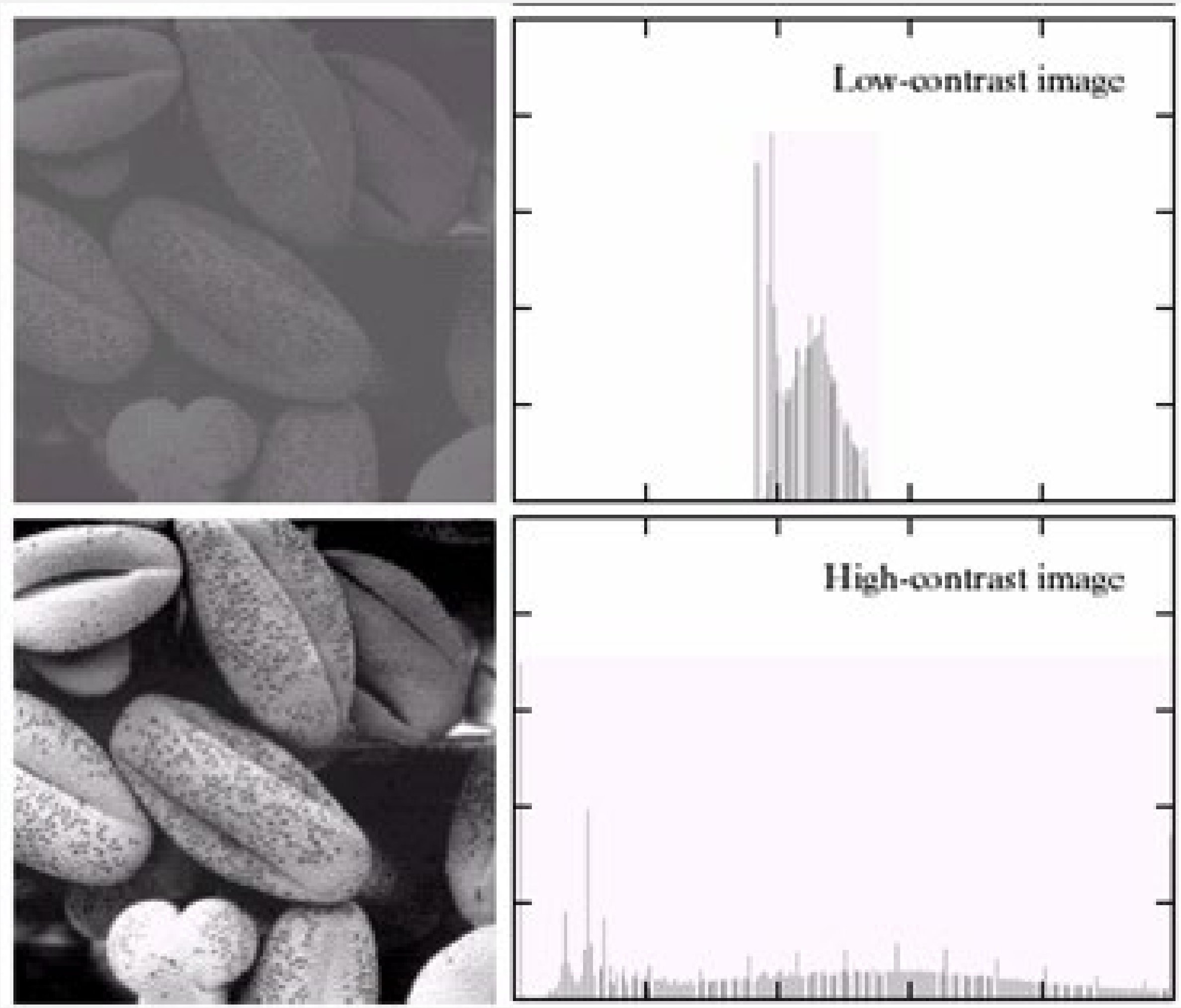
WHAT IS A HISTOGRAM?

- A graph indicating the number of times each gray level occurs in the image, i.e. frequency of the brightness value in the image
- The histogram of an image with L gray levels is represented by a one-dimensional array with L elements.
- Algorithm:
  - Assign zero values to all elements of the array h_f;
  - For all pixels (x,y) of the image f, increment $h_f[f(x,y)]$ by 1.



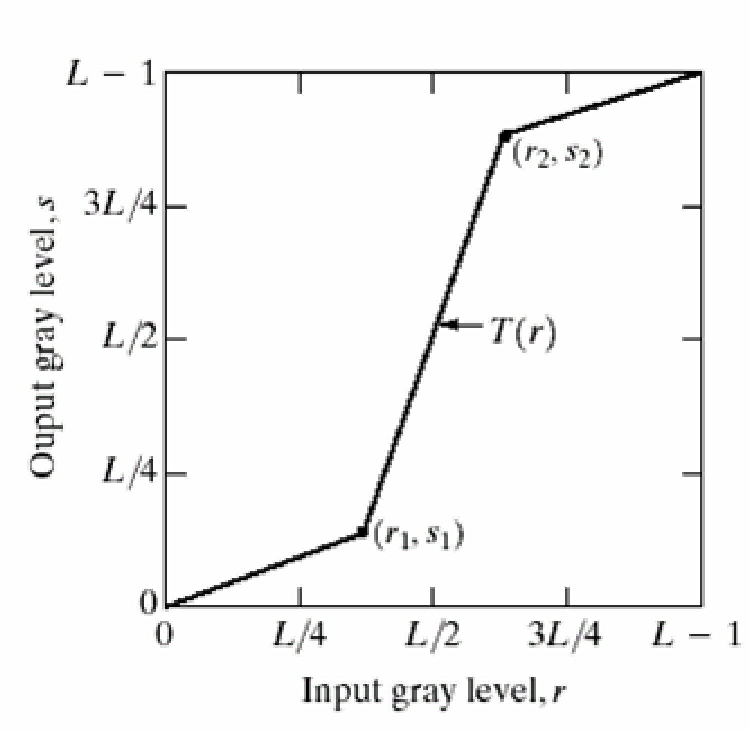
HISTOGRAM EXAMPLES





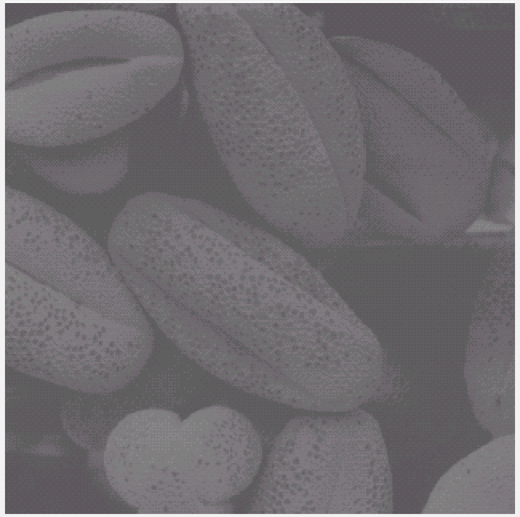
HISTOGRAM/CONTRAST STRETCHING

- To increase the dynamic range of the gray levels in the image.
- Piecewise linear function
- (r_1, s_1) & (r_2, s_2) control the shape of the mapping, where $r_1<r_2$ & $s_1<s_2$.
- What if
  - $r_1=s_1$ & $r_2=s_2$?
  - $r_1=r_2$, $s_1=0$ & $s_2=L-1$?

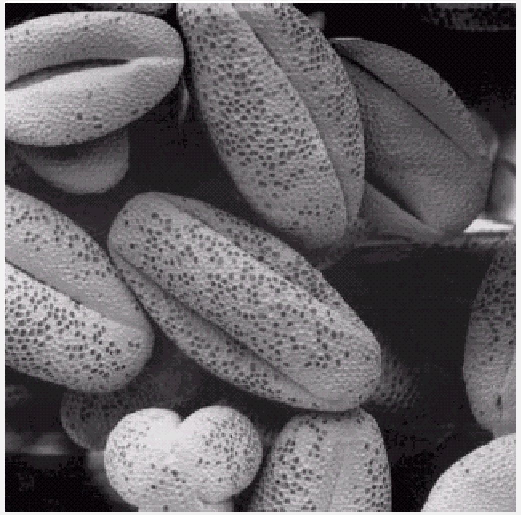


EXAMPLES OF HISTOGRAM STRETCHING

- What is the problem here?



Original
($r_1=s_1$&$r_2=s_2$)

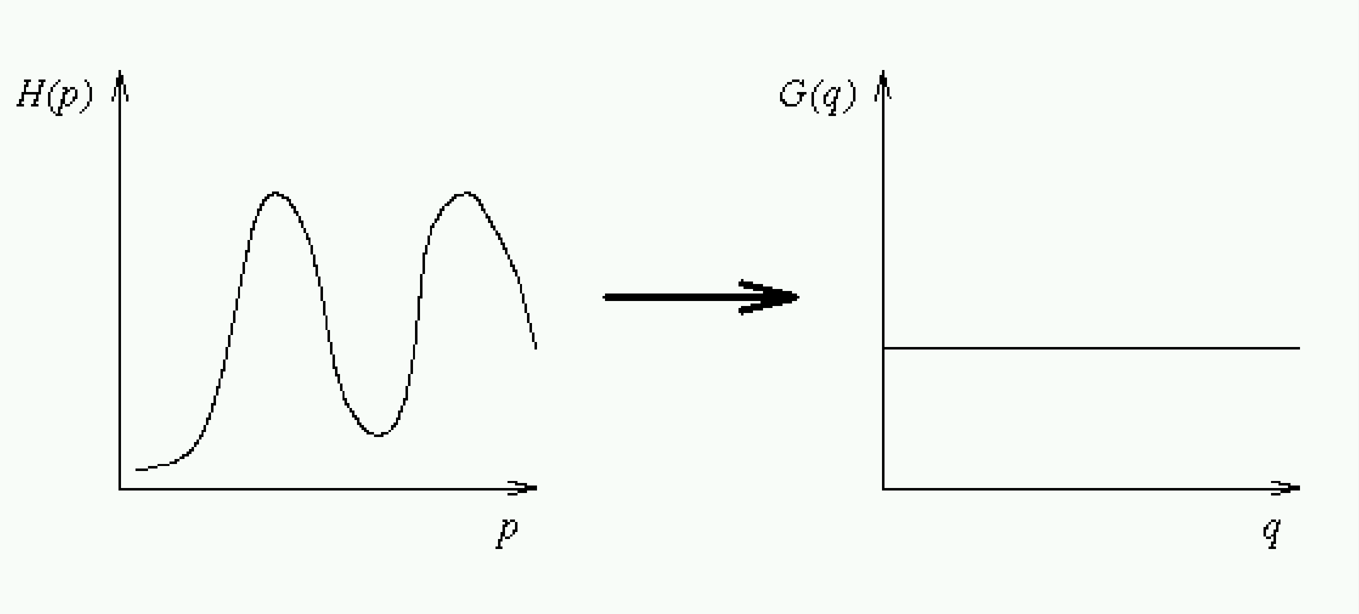


Contrast stretched
($r_1=r_{min}$, $r_2=r_{max}$, $s_1=0$,$s_2=L-1$)

Thresholding
($r_1=r_2$,$s_1=0$& $s_2=L-1$)

HISTOGRAM EQUALIZATION: INTUITION

- Goal: to produce an image with equally distributed brightness levels over the whole brightness scale.
- Effect: enhancing contrast for brightness values close to histogram maxima, and decreasing contrast near minima.
- Result is better than just stretching, and method fully automatic.



HISTOGRAM EQUALIZATION: HOW IT WORKS?

- Let $H(p)$ be the input histogram and that the input gray-scale is $[p_o, p_k]$.
- The intention is to find a monotonic pixel brightness mapping $q = T(p)$ such that the output histogram $G(p)$ is uniform over the whole output brightness scale
- The monotonic property of the transform T implies

$$\sum_{i=0}^{j} G(q_i) = \sum_{i=0}^{j} H(p_i)$$

HISTOGRAM EQUALIZATION: HOW IT WORKS?

- The histogram is a discrete probability density function.
- The sum in the above equation can be interpreted as discrete distribution function.
- Assume that the image has M rows and N columns. The equalized histogram corresponding to a uniform probability density function *is*:

$$G(q) = \frac{MN}{q_k - q_0}$$

HISTOGRAM EQUALIZATION: HOW IT WORKS?

- The equalized histogram can be obtained precisely only for the ``idealized'' continuous probability density, in which case the sum equation becomes

$$\int_{q_0}^{q} G(s)ds = \int_{q_0}^{q} \frac{MN}{q_k - q_0} ds = \boxed{\frac{MN(q - q_0)}{q_k - q_0} = \int_{p_0}^{p} H(s)ds}$$

- The desired pixel brightness transformation T can then be derived as

$$q = T(p) = \frac{q_k - q_0}{MN} \underbrace{\int_{p_0}^{p} H(s)ds}_{\text{Cumulative histogram}} + q_0$$

HISTOGRAM EQUALIZATION: HOW IT WORKS?

- The discrete approximation of the continuous pixel brightness transformation from the above equation is

$$q = T(p) = \frac{q_k - q_0}{MN} \sum_{i=p_0}^{p} H(i) + q_0$$

HISTOGRAM EQUALIZATION - ALGORITHM

- For an NxM image of G gray-levels (often 256), compute the image histogram H.
- For the cumulative histogram H_c:

$$H_c[0] = \quad H[0]$$
$$H_c[p] = \quad H_c[p-1] + H[p] \quad , \mathrm{p} = 1,\ldots,\mathrm{G\text{-}1}$$

- Set

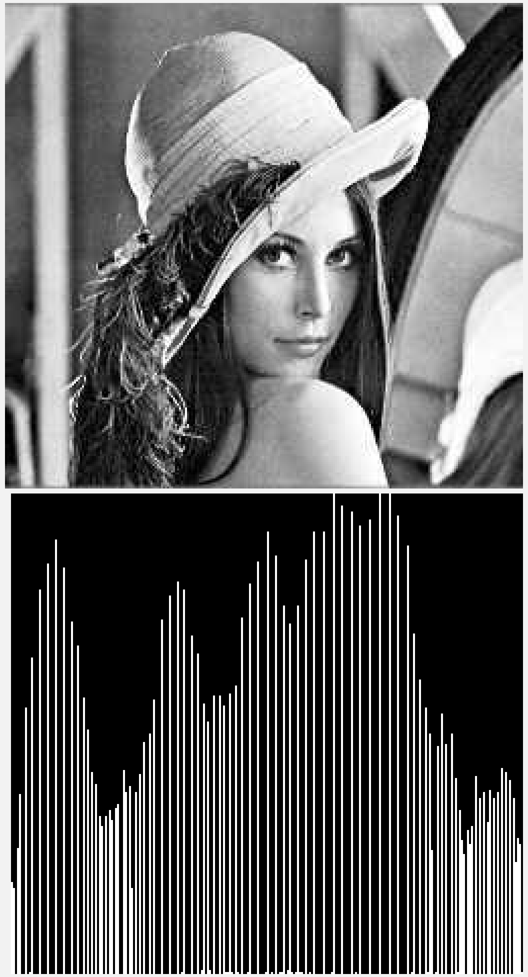
$$T[p] = round\left(\frac{G-1}{NM} H_c[p]\right)$$

- Rescan the image and write an output image with gray-levels

$$g_q = T[g_p]$$

HISTOGRAM EQUALIZATION: EXAMPLE

Original



Equalized

HISTOGRAM MATCHING/SPECIFICATION

- The generalization of histogram equalization is called histogram matching.
- The aim is to produce an image with desired distributed brightness levels over the whole brightness scale, as opposed to uniform distribution.
- Assume the histogram of the desired probability density function is $G(q)$.
- Let the desired pixel brightness transform be $q=T[p]$.

HISTOGRAM MATCHING/SPECIFICATION

- Similarly, the cumulative histogram of the input image is

$$H_c[p] = \int_{q_0}^{q=T[p]} G[s]ds$$

- From the above eqn, it is possible to find T.
- E.g. if G is the exponential distribution,

$$G[q] = \alpha e^{-\alpha(q-q_0)}, \quad \text{for } q \geq q_0$$

- We have

$$H_c[p] = 1 - e^{-\alpha(T[p]-q_0)}$$

- Then,

$$T[p] = q_0 - \frac{1}{\alpha}\log\left(1 - H_c[p]\right)$$

SUMMARY

- Enhancement using point operations
  - Arithmetic operations
  - Histogram stretching
  - Histogram equalization
  - Histogram matching/specification

TERIMA KASIH

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia