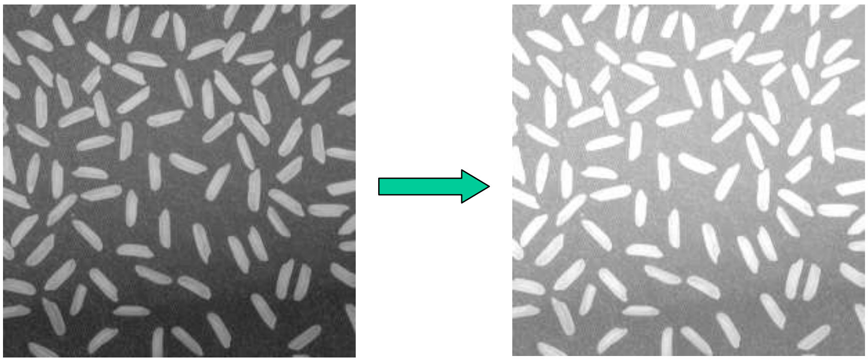
Basic image processing methods

Image processing based on simple point operations, pixels are processed without context (neighborhood pixels) analysis

- Enhancement using arithmetic operations
  - Adding (subtracting) a constant value to the image pixels

$$J_w(x,y) = J(x,y) \pm b$$

$b = 80$



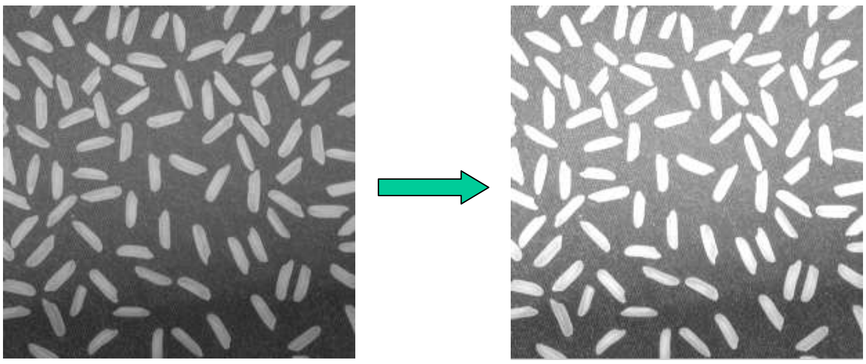
Depending on the hardware or software being used, this operations can be done sequentially one pixel at a time, or in parallel simultaneously

Basic image processing methods

- Multiplying (dividing) image pixels by a constant value

$$J_w(x,y) = aJ(x,y)$$

$a = 1,5$



Addition and multiplication operations are linear transformation of image pixels

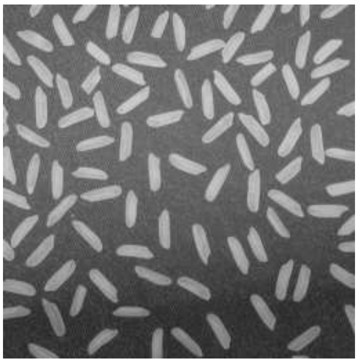
$$J_w(x,y) = aJ(x,y) + b$$

Be careful to not exceed the range of representation!

Basic image processing methods

– Adding (subtracting) images as appropriate matrix operations

$$J_w(x,y) = J_1(x,y) + J_2(x,y)$$

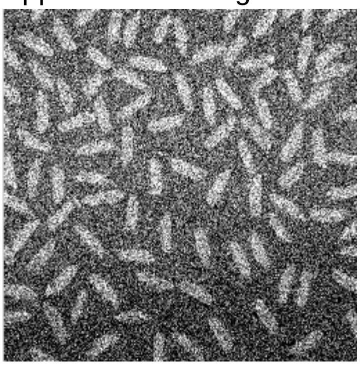


$+$

In the case of exceeding the range of representation, the arithmetic overflow or normalization should be used, or resulting image should be stored in buffer with higher bit depth

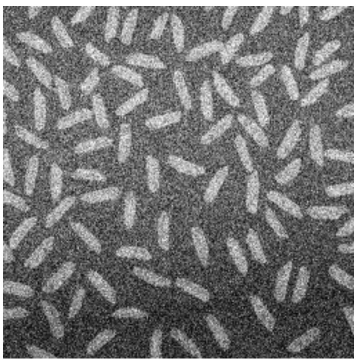
Basic image processing methods

– Application of images addition for noise reduction (image averaging)

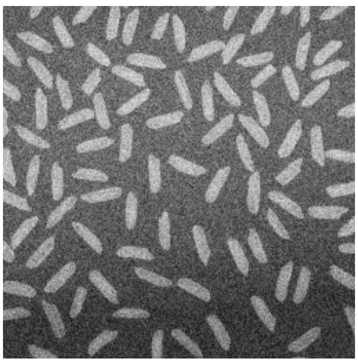


```
In=imnoise(i,'gaussian',0,.05)
```

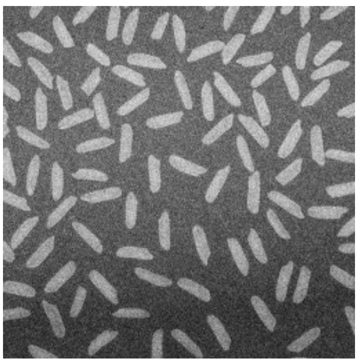
$$J_w(x,y) = \sum_{i=1}^{N} J_i(x,y)/N$$



$N = 3$



$N = 10$



$N = 20$

Basic image processing methods

– Mixing of the two images (a generalization of image addition)

$$J_w(x,y) = \alpha J_1(x,y) + (1-\alpha)J_2(x,y) \quad \alpha \in [0,1]$$

$\alpha = 0,5$

– Mixing of the *N* images

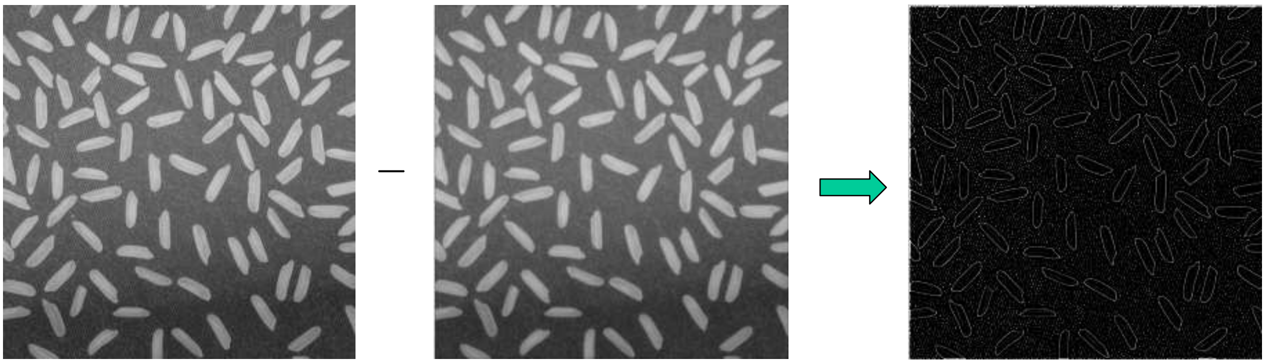
$$J_w(x,y) = \alpha_1 J_1(x,y) + \alpha_2 J_2(x,y) + \ldots + \alpha_N J_N(x,y)$$

where $\sum_{i=1}^{N} \alpha_i = 1$

Basic image processing methods

– Image subtraction for enhancement of differences between images (difference image)

$$J_w(x,y) = J_1(x,y) - J_2(x,y)$$

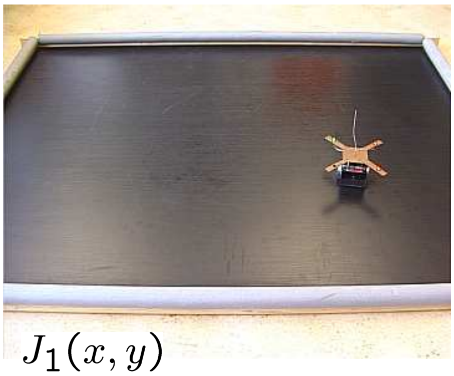


$J_1(x,y)$ $J_2(x,y)$ $J_w(x,y) * 10$

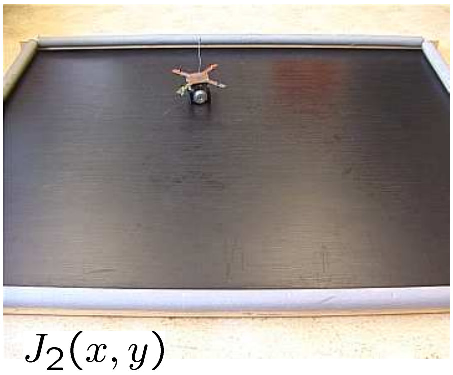
```
J2=imfilter(J1,fspecial('gaussian'));
```

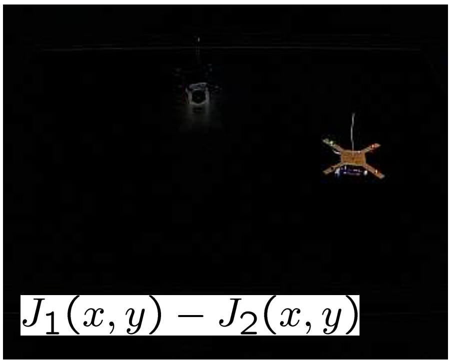
Basic image processing methods

– Difference image for object detection



–





Basic image processing methods

- Nonlinear point operations
  - Power-law transformation

$$J_w(x,y) = J(x,y)^{\alpha}, \quad \alpha > 0$$

Power transformation with normalization

$$J_w(x,y) = 255 \cdot \left(\frac{J(x,y)}{J_{max}}\right)^{\alpha}$$

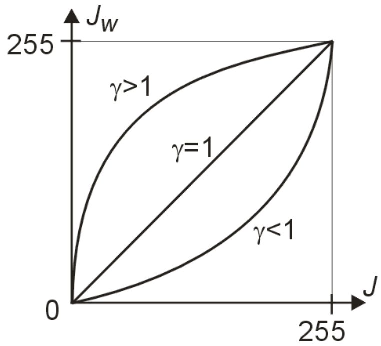
- For the integer value of the exponent it can be done by multiplying of corresponding image pixels
- For $\alpha > 1$ power transformation results with image darkening and greater variation of light pixels (map narrow range of light input values into wider range of output values)
- For $\alpha < 1$ power transformation results with image brightening and grater variation of dark pixels (map narrow range of dark input values into wider range of output values)

Gamma correction – it derive from correction of nonlinearity of cathode ray tube (CRT) devices used in analog television (e.g. for PAL system $\gamma = 2,8$)

- Gamma correction is done by power transformation where $\alpha = 1/\gamma$
- The aim of gamma correction is usually to brighten dark areas of the image

Basic image processing methods

Correction curves, depending on the γ value



$\gamma = 2$

Remark: in Matlab $\alpha = \gamma$

Basic image processing methods

– Log transformation

$$J_w(x,y) = \log(1 + J(x,y))$$

Log with normalization

$$J_w(x,y) = 255 \cdot \frac{\log(1+J(x,y))}{\log(1+J_{max})}$$

Logarithm transformation maps a narrow range of low gray level values in input image into a wider range of output levels.

– Image masking by multiplication operation (logic AND operation)

$$J_w(x,y) = J(x,y) \cdot M(x,y)$$

Masking sometimes is referred to as region of interest (ROI) processing. In terms of enhancement, masking is used to isolate an area for processing.

3	4	4	3
5	5	4	3
3	3	0	2
1	2	1	1

•

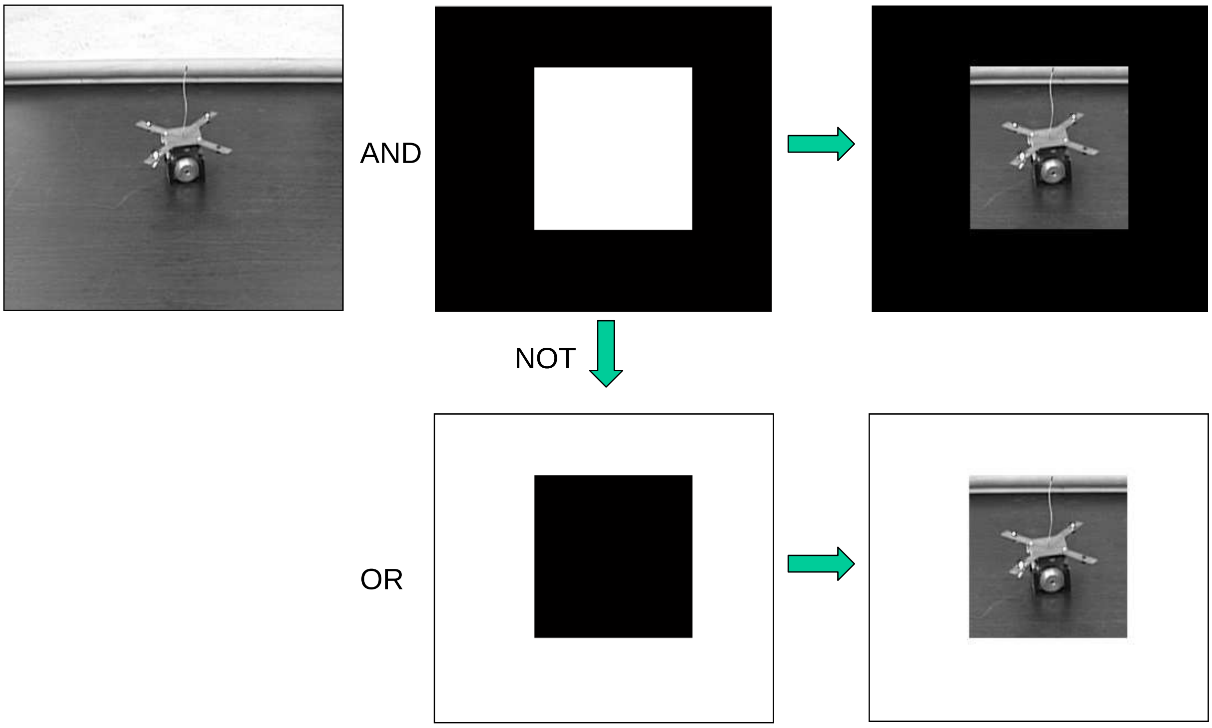
0	0	0	1
0	1	1	1
0	1	1	1
0	0	0	1

=

0	0	0	3
0	5	4	3
0	3	0	2
0	0	0	1

Basic image processing methods

– Examples of masking using logical operation AND and OR



Basic image processing methods

- Geometric transformations
  - Shifting of image pixels

$$x' = x_o + x_i$$
$$y' = y_o + y_i$$

(x_o, y_o) - initial coordinates of the pixels
(x_i, y_i) - offset values
(x', y') - new coordinates of pixels after movement

  - Image resampling (scaling)

$$x' = x_o \cdot x_s$$
$$y' = y_o \cdot y_s$$

x_s, y_s - the scaling factors

  - Image rotation

$$x' = x_o \cdot \cos(\theta) - y_o \cdot \sin(\theta)$$
$$y' = x_o \cdot \sin(\theta) + y_o \cdot \cos(\theta)$$

θ - rotation angle

For scaling and rotation typically image resampling is required using interpolation methods for non-existent pixels (*nearest neighbor*, *bilinear*, or *bicubic method*)

Basic image processing methods

- Nearest neighbor interpolation
  - each pixel of the output image takes unmodified value of closest pixel from input image in a distance sense
  - in the case of increasing spatial resolution a few points of the resulting image can be set to the same value of one point from the input image, in the case of decreasing image resolution some points, which don't have corresponding pixels in the output image are lost

$$x_s = M_{src}/M_{dst}$$
$$y_s = N_{src}/N_{dst}$$

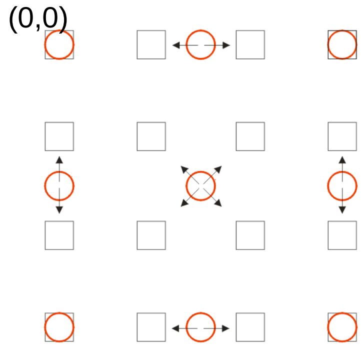
For the pixel of output image with coordinates (i,j) the coordinates of the input image:

$$x = i \cdot x_s$$
$$y = j \cdot y_s$$

Example scaling from 3x3 to 4x4 resulting pixel (1,3):

$$x = i \cdot x_s = 1 \cdot 3/4 = 0,75 \approx 1$$
$$y = j \cdot y_s = 3 \cdot 3/4 = 2,25 \approx 2$$

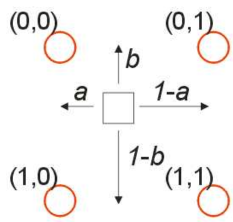
From 3x3 up to 4x4

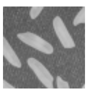


Basic image processing methods

- Bilinear interpolation
  - each pixel of the resulting image has a value based on calculation with using the four neighboring points from the input image
  - in the first step, the coordinates of the pixel in the input image are determined (as in the nearest neighbor method)
  - in the next step, linear interpolation is done in the two directions

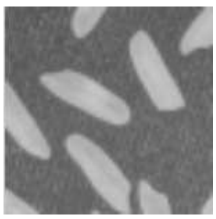
$$f(i,j) = (1-b)[(1-a)f(x,y) + af(x+1,y)] + \\ +b[(1-a)f(x,y+1) + af(x+1,y+1)]$$





x2.5

Nearest neighbor



Bilinear interpolation

Basic image processing methods

- Image histogram
  - The histogram is a diagram which shows a graphical representation of the statistical distribution of gray level (color components) in the image.

$J_0, J_1, J_2, \ldots, J_{L-1}$ - indicate the possible pixel gray levels values, L is the number of available levels

$$h(J_i) = n_i, \quad i = 0, 1, 2, \ldots, L-1$$

Calculation of the histogram

$$n_i = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g_i(x,y), \quad i = 0, 1, 2, \ldots, L-1$$

$$g_i(x,y) = 1 \quad for \quad J(x,y) = i$$
$$g_i(x,y) = 0 \quad for \quad J(x,y) \neq i$$

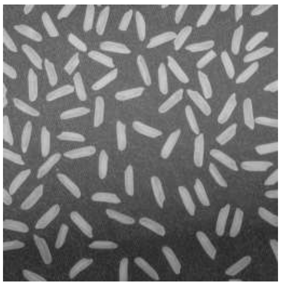
n_i - is the number of pixels with gray level value J_i

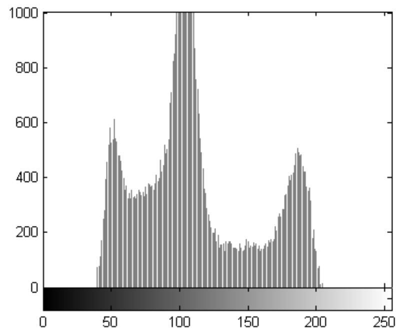
Histogram normalization

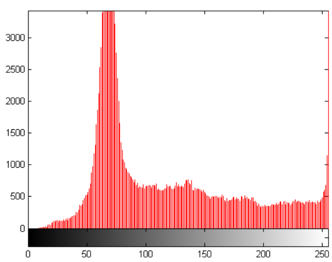
$$h(J_i) = \frac{n_i}{n}, \quad n = M \cdot N$$

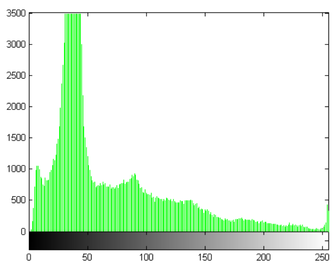
Basic image processing methods

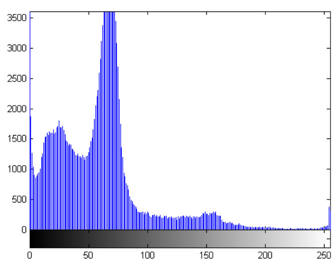
- Examples of images and histograms











Basic image processing methods

- *Histogram equalization*
  - It is based on transformation of $h(J_i)$ function in order to uniform the resulting histogram of output image

    The probability of gray level r_k occurrence in the image

$$p(r_k) = \frac{n_k}{n}, \quad 0 \leqslant r_k \leqslant 1, \quad k = 0, 1, 2, \ldots, L-1$$

n_k - the number of pixels with a given gray level

n - the total number of pixels in an image

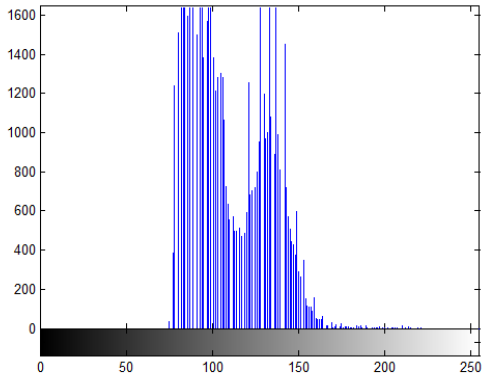
The transformation function for histogram equalization (cumulative histogram)

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j), \quad k = 0, 1, 2, \ldots, L-1$$

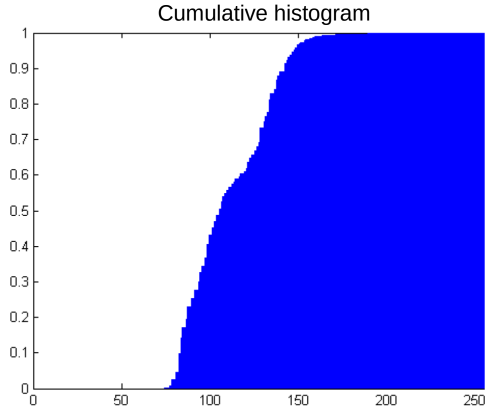
  - Histogram equalization enhances the contrast of the image (sometimes unnaturally) and provide occupation of the entire range of possible gray levels

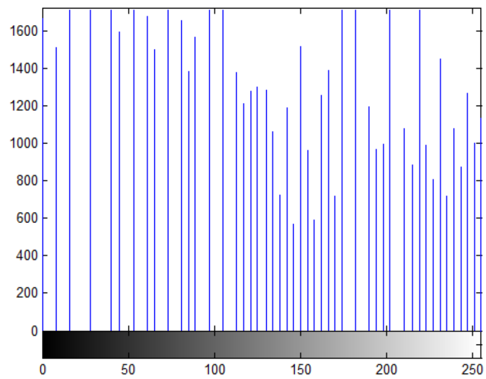
Basic image processing methods

- Example of histogram equalization



Cumulative histogram

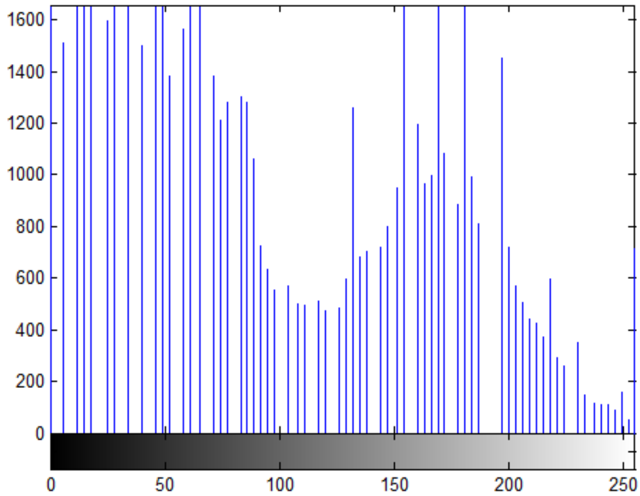




Basic image processing methods

- Stretching of the histogram (contrast *stretching*)
  - Let the gray level range is between J_{min} and J_{max}
    The operation of the linear stretching to the entire range of gray levels is given by

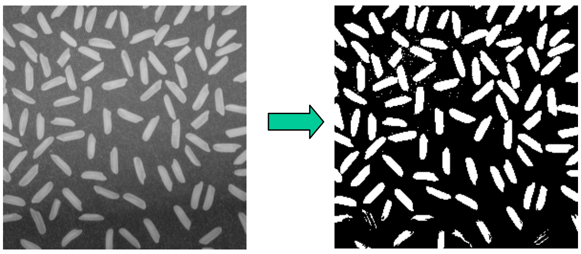
$$J_w(x,y) = \frac{255}{J_{max}-J_{min}}(J(x,y) - J_{min})$$



Basic image processing methods

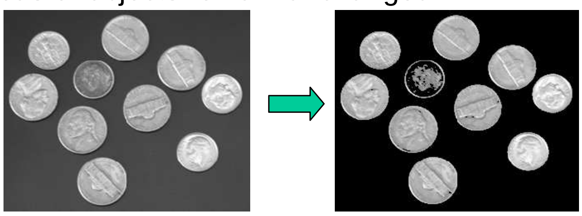
- Image thresholding
  - Thresholding is a point operation which can be used for image partitioning to distinguish similar areas in the image. The process of image dividing into areas that meet predetermined criterion is called segmentation. It is one of the basic tasks of image analysis. The concept of the area is used to define a coherent group of pixels having a common feature.
  - The basic method of thresholding – binarization with one threshold t

$$J_w(x,y) = \begin{cases} 1, & J(x,y) > t \\ 0, & J(x,y) \leqslant t \end{cases}$$



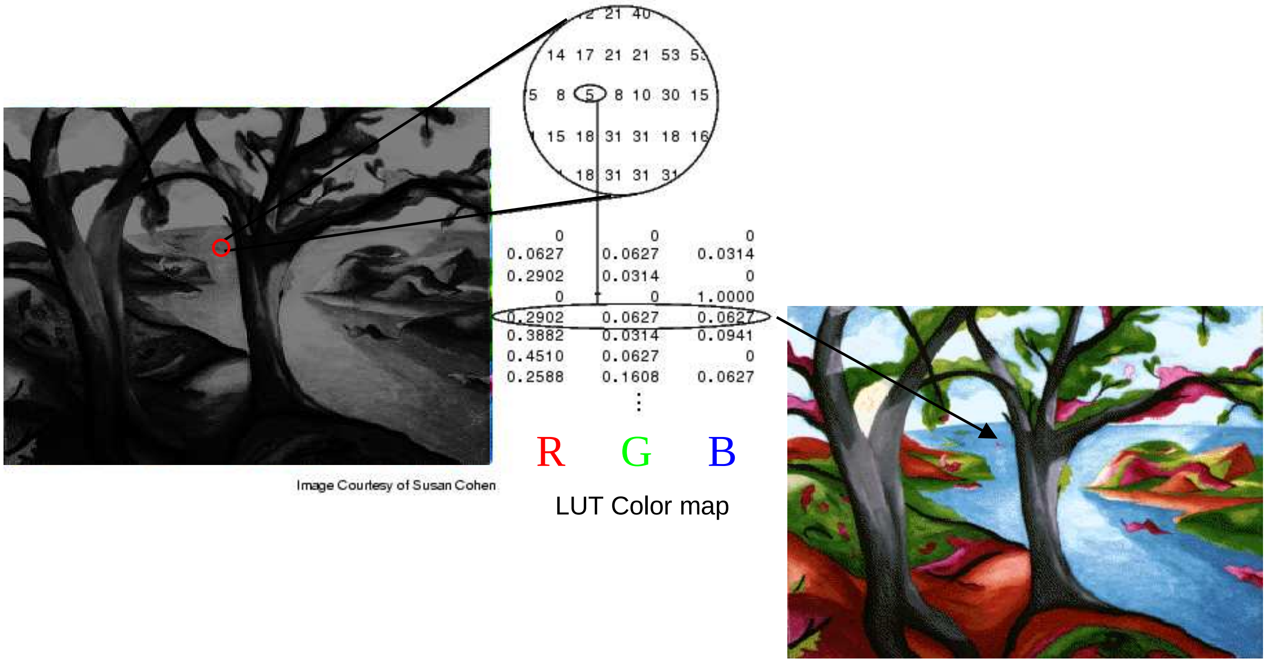
  - Pseudo binarization technique, detected areas of objects remain unchanged

$$J_w(x,y) = \begin{cases} J(x,y), & J(x,y) > t \\ 0, & J(x,y) \leqslant t \end{cases}$$



Basic image processing methods

- Look Up Table (LUT)
  - Look Up Table is an array in which the values of transformation function are stored
  - Indexed image as an example of using LUT



Basic image processing methods

- Example of using LUT to multicriteria binarization

0	0
...	...
99	0
100	150
...	...
149	150
150	255
...	...
255	255