Basics of image segmentation

- The segmentation objective

  The purpose of segmentation is the division of the image into regions corresponding to particular, visible objects on the image. Segmentation is connected with objects labeling process, that is, assigning to each object a different label (all pixels of the object receive the same value).

  The region in the image, blob - a set of pixels with similar characteristics (brightness, color, texture). Image segmentation into regions is the basis for objects detection and analysis.

- Image segmentation algorithms:
  - segmentation based on discontinuity in the image (such as points, lines, edges and contours of objects)
  - segmentation based on similarity in the image, partitioning an image into regions that are similar according to defined criteria (partitioning for homogeneity of neighborhood pixels)

Basics of image segmentation

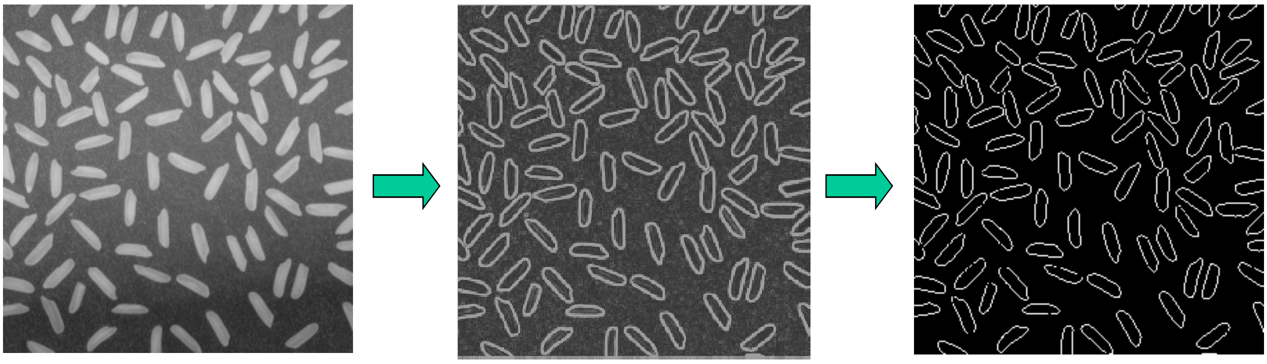
- The neighborhood of a pixel in the image
  - Direct neighbors, D-neighbors: pixels that are adjacent with a point (x,y) {0,2,4,6}
  - Nondirect neighbors, N-neighbors: Points that have adjacent corners with point (x,y) {1,3,5,7}

5	6	7
4	(x,y)	0
3	2	1

  - Conectivity of the region in the 4-neighbors sense - any of the D-neighbors has a uniform feature with the point (x, y)
  - Conectivity of the region in the 8-neighbors sense - any of the N-neighbors and D-neighbors has a uniform feature with the point (x, y)
  - Conectivity of the region in the 4-neighbors or 8-neighbors sense can refer to the lines, edges, contours of the regions as well to the pixels of regions

Basics of image segmentation

- Image segmentation by edge detection
  - The extraction of edge pixels is carried out by thresholding (binarization) of the gradient image obtained as a result of filtration with using gradient operators (gradient filters)



Sobel operators Thresholding

Basics of image segmentation

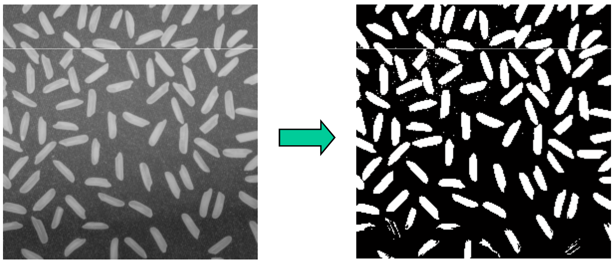
- The segmentation based on edge detection requires a contour tracking algorithm in order to define the continuous boundary of the regions
- Problems during tracking the edges:
  - edge discontinuity – the need for edge combining and gap filling,
  - thick edges - the necessity for edge thinning,
  - branching edges,
  - open contours.
- Boundary representations:
  - ordered list of edge pixels,
  - list of segments,
  - arc representation,
  - polynomial representation.

Basics of image segmentation

Image segmentation based on regions similarity

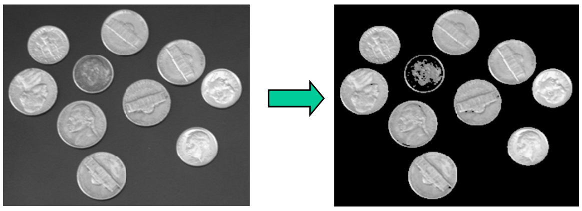
- Image segmentation by thresholding techniques
  - the advantage - simplicity of implementation
  - the problem with threshold (thresholds) selection
  - usually requires additional logical (non-linear) filtration to remove isolated pixels, smoothing the edges or to unify the interior of regions
- The basic method of thresholding – binarization with one threshold

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



- Pseudo binarization technique

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



Basics of image segmentation

- Dual thresholding method allows to extract the luminance areas within a certain range

$$J_w(x,y) = \begin{cases} 0, & J(x,y) \leqslant t_1 \\ 1, & t_1 < J(x,y) \leqslant t_2 \\ 0, & J(x,y) > t_2 \end{cases}$$

- Multi-criteria thresholding can generate images composed of multiple areas with different gray levels

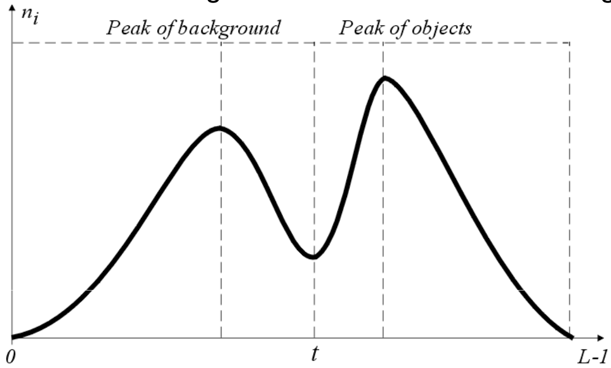
$$J_w(x,y) = \begin{cases} 1, & J(x,y) \in D_1 \\ 2, & J(x,y) \in D_2 \\ \cdots, & \cdots \\ n, & J(x,y) \in D_n \\ 0, & \text{other} \end{cases}$$

- Global thresholding – the threshold is the same for the entire image
- Local thresholding – the threshold is dynamic and takes values depending on the pixel coordinates in the image

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

Basics of image segmentation

– The choice of the threshold for the image based on histogram with a bimodal distribution (the method does not guarantee the coherence of regions in general)



– Determining the threshold based on the image statistics

- Determination of the maximum brightness gradient module $G(x,y)$ for each point

$$G_x(x,y) = J(x+1,y) - J(x-1,y),$$
$$G_y(x,y) = J(x,y+1) - J(x,y-1)$$
$$G(x,y) = max\{|G_x(x,y)|, |G_y(x,y)|\}$$

- The calculation of the threshold $t = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} J(x,y)G(x,y)}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} G(x,y)}$

Basics of image segmentation

- Adaptive thresholding algorithm [Bradley, Roth, 2007]
  – Integral image

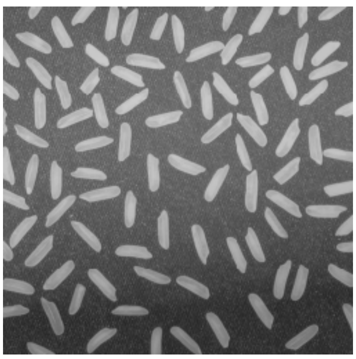
$$I(x,y) = J(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1)$$

Input image

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

Integral image

4	4	7	9	12
4	6	13	18	22
6	11	20	25	33
8	14	24	32	42



Basics of image segmentation

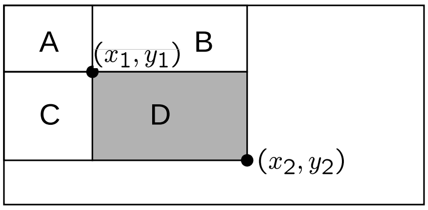
– The sum of the image function between the upper left and lower-right corner of the rectangular part of the image

Input image

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

Integral image

4	4	7	9	12
4	6	13	18	22
6	11	20	25	33
8	14	24	32	42



(A+B+C+D)-(A+B)-(A+C)+A=D

The sum of the image function between the upper left and lower-right corner based on the integral image

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} J(x,y) = I(x_2,y_2) - I(x_2,y_1-1) - I(x_1-1,y_2) + I(x_1-1,y_1-1)$$

Basics of image segmentation

– Adaptive thresholding algorithm based on the integral image

```
For i=0 to r
   For j=0 to c
      x1=i-s/2; x2=i+s/2; y1=j-s/2; y2=j+s/2;
      count=(x2-x1)*(y2-y1);
      sum=I(x2,y2)-I(x2,y1-1)-I(x1-1,y2)+I(x1-1,y1-1);
      If (in(i,j)*count)<=(sum*(100-t)/100) then
         out(i,j)=0;
      else
         out(i,j)=1;
      endif
   endfor
endfor
```

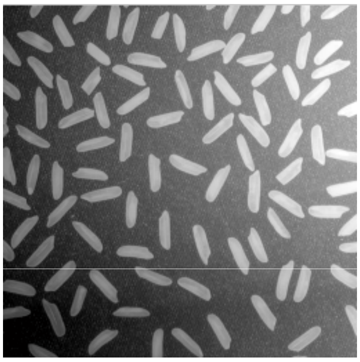
`r,c` - number of rows and columns

`s` - size of integral window

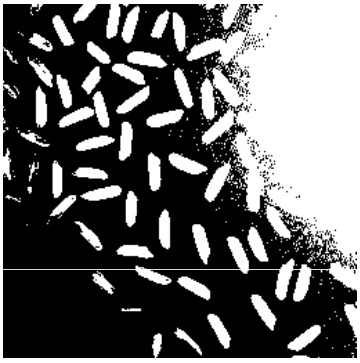
`t` - threshold in percent

Basics of image segmentation

– Adaptive thresholding algorithm - result



Input image

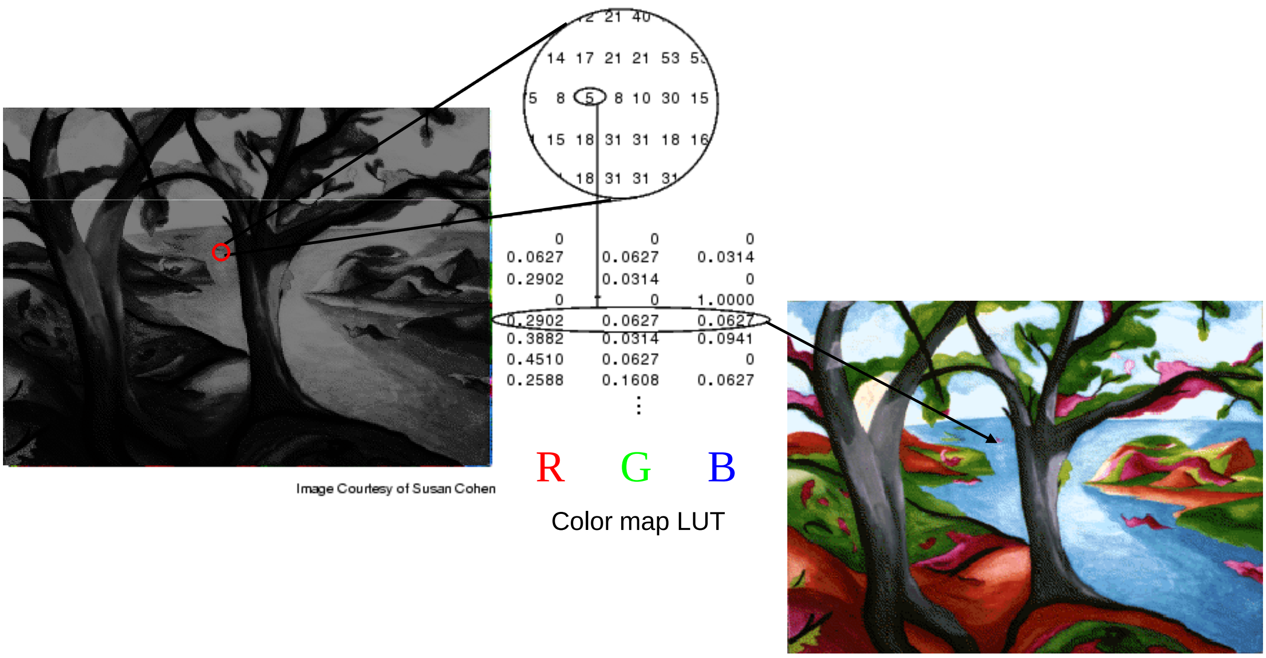


Binarization with fixed threshold

Adaptive thresholding

Basics of image segmentation

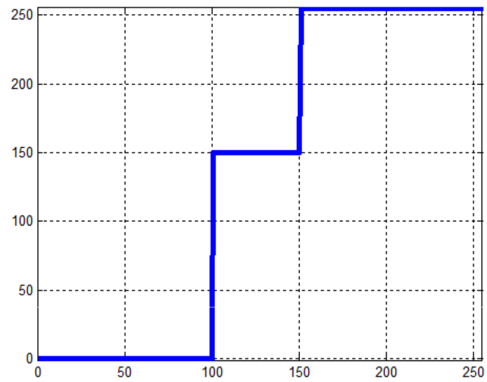
- Look Up Tables in image processing
  - LUT is an array in which values of the transformation function can be stored
  - Indexed image as an example of the use of LUT

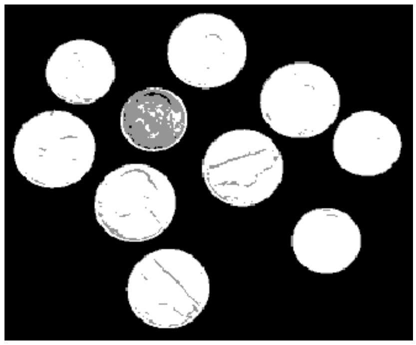


Basics of image segmentation

– An example of the use of LUT for multi-criteria thresholding

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





Basics of image segmentation

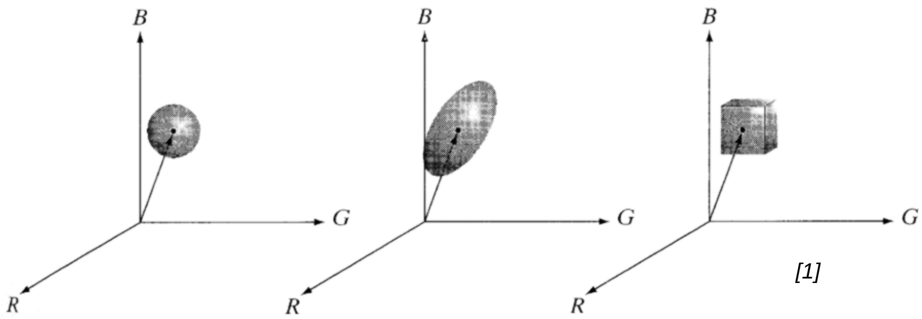
- Color image segmentation
  - Segmentation with using HSV color model
    - Hue component allows in a natural way to specify the color and search for homogeneous areas of color
    - Saturation component can be used to mask the image in order for further separation of the regions

Basics of image segmentation

- Segmentation with using RGB color model
  - Segmentation uses three RGB components defining a vector in RGB space
  - The key problem is to define the ranges of variation of the RGB components and how to measure the similarity between the points of the color image in RGB color space
  - Euclidean distance can be used to measure the similarity between pixels

$$D(\mathbf{a},\mathbf{b}) = ||\mathbf{a}-\mathbf{b}|| = [(\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b})]^{\frac{1}{2}}$$

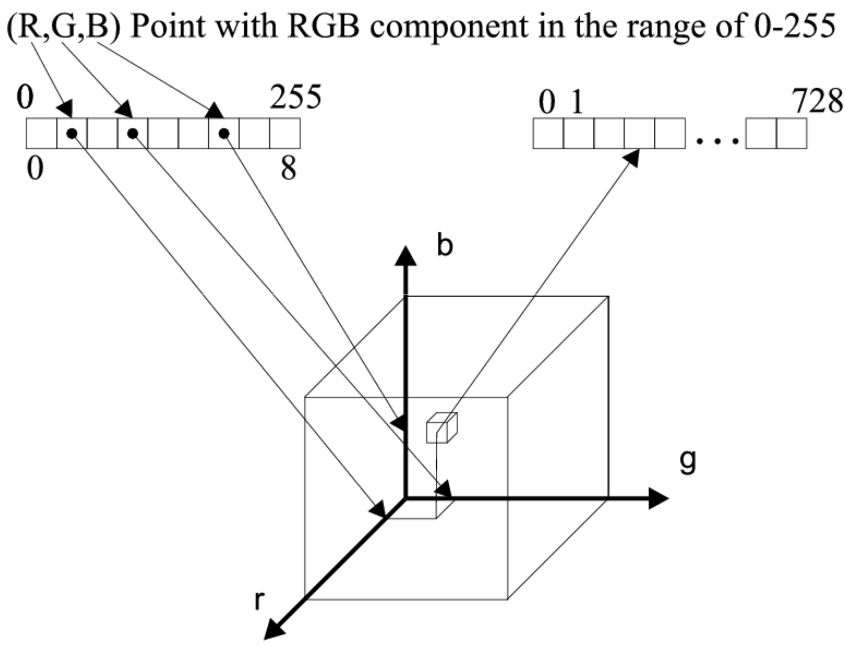
$$D(\mathbf{a},\mathbf{b}) = \sqrt{(a_R-b_R)^2+(a_G-b_G)^2+(a_B-b_B)^2} < D_0$$



[1]

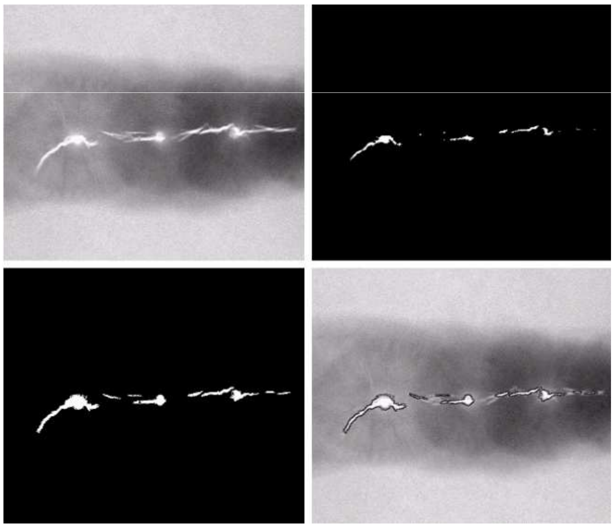
Basics of image segmentation

- Segmentation and color objects classification with RGB model and LUT



Basics of image segmentation

- Segmentation by region growing
  - It is based on a preliminary segmentation using binarization technique with conservative threshold to distinguish the seeds mark of the objects. Then the seeds regions are iteratively grown by comparing all unallocated neighboring pixels to the regions using local model of the regions.



X-ray image of weld defects [1]

Basics of image segmentation

- Segmentation by region splitting
  - The image or region is splitted successively into subregions as long as the specific condition of region homogenity is not met for each region

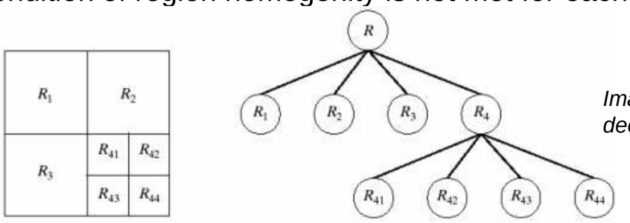
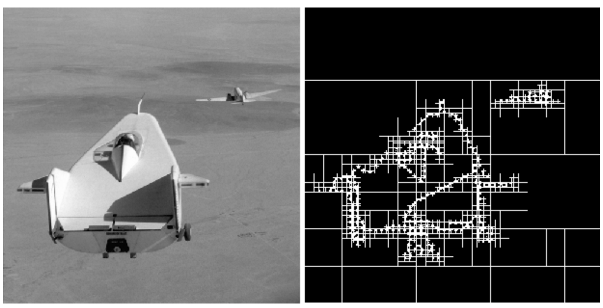


Image splitting using quadtree decomposition [1]

- Split and Merge techniques
  - During the splitting process some of the obtained neighboring regions may have the same properties - they can be merge to one region



Basics of image segmentation

- Region labeling on the image (connected component labeling)
  - Recursive algorithm

1. Search rows of image until meets a pixel with a value of "1" which has not been given the label. Give him a new label L.
2. Recursively assign L label to neighbors with a value of "1" using `Indexing(x,y,L)` procedure.
3. If the end of the image is reached then stop.
4. Go to 1.

```
Indexing(x,y,L)
If J(x,y)==1
{
       J(x,y)=L;
       Indexing(x-1,y,L); Indexing(x+1,y,L);
       Indexing(x,y-1,L); Indexing(x,y+1,L);
}
```


Basics of image segmentation

```
Indexing(x,y,L)
Push(x,y)
While !stack_empty do
{
       Pop(x,y)
       if J(x,y)==1
       {
              J(x,y)=L;
              push(x-1,y,L); push(x+1,y,L);
              push(x,y-1,L); push(x,y+1,L);
       }
}
```

Basics of region representation and description

Shape features

Topological

- **connected components**
- **numbers of holes**
- **Euler number**
- **skeleton**

Geometrical

- **length and diameter of boundary**
- **center of the gravity**
- **higher order moments**
- **major and minor axis**
- **eccentricity**
- **others**

Basics of region representation and description

- Distance measure on the image
  - Euclidean distance

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$\sqrt{2}$	1	$\sqrt{2}$
1	(x_1,y_1)	1
$\sqrt{2}$	1	$\sqrt{2}$

  - 4-neighbors distance

$$D = |x_2 - x_1| + |y_2 - y_1|$$

2	1	2
1	(x_1,y_1)	1
2	1	2

  - 8-neighbors distance

$$D = max(|x_2 - x_1|, |y_2 - y_1|)$$

2	2	2	2	2
2	1	1	1	2
2	1	(x_1,y_1)	1	2
2	1	1	1	2
2	2	2	2	2

Basics of region representation and description

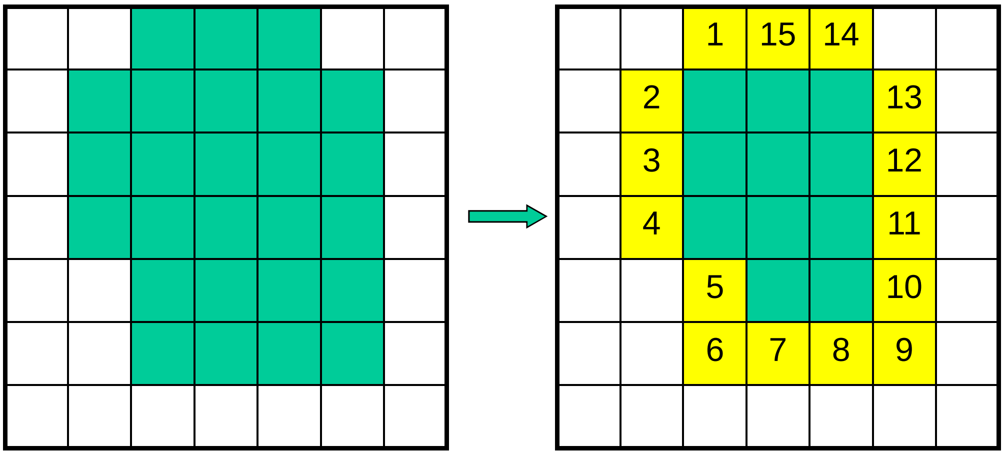
Geometrical descriptors

- Perimeter - length of the boundary

$$O = \int \sqrt{x^2(t) + y^2(t)}dt$$

  – For discrete grid, the real length of the edge is not the number of boundary points

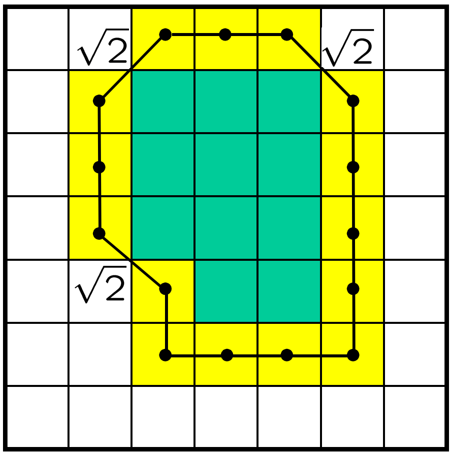
1. Perimeter of the object equals the number of elements of the boundary (the simplest method, but can produce a relatively large error of the estimator)



$O = 15$

Basics of region representation and description

2. Perimeter of the object is equal to the sum of the segments length connecting centers of contour pixels. It is assumed that the contour element is a square with side = 1.



$O = 12 + 3\sqrt{2} \approx 16,24$

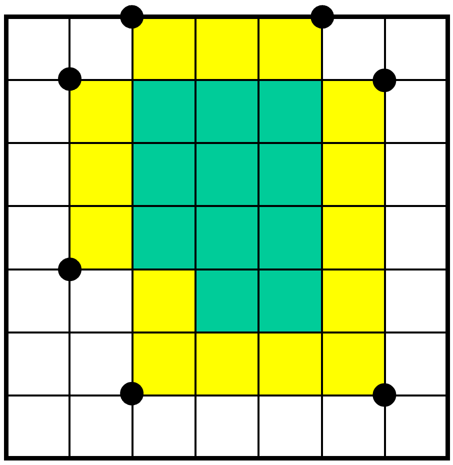
This method is more accurate than the method with the number of boundary points

Basics of region representation and description

3. Perimeter of the object is determined based on the relationship

$$O = aN_b - bN_w$$

N_b – the number of external sides of the contour points
N_w – the number of vertices of the contour



$$a = \frac{\pi(1+\sqrt{2})}{8} \qquad b = \frac{\pi}{8\sqrt{2}}$$

$$O = 22a - 7b \approx 18,91$$

This method provides a length estimator with zero mean and a minimum variance for segments inclined at different angles

Basics of region representation and description

- Determination (tracking) of the internal boundary of the region

1. Search the image rows in sequence until you find the first point of region. Mark this pixel as P_0 (the starting point of the contour). Define the variable *dir* storing the previous direction of movement along the boundary (from the previous to the next edge point).

   Assign *dir*=3 for 4-neighborhood
   dir=7 for 8-neighborhood
2. Search 3x3 neighborhood of the current pixel rotating counter clockwise direction starting from

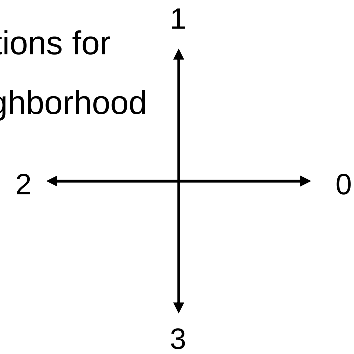
   a) for 4- neighborhood: (*dir*+3) mod 4
   b) for 8- neighborhood: (*dir*+7) mod 8 when *dir* is even number
   (*dir*+6) mod 8 when *dir* is odd number

   The first found point of the region is a new element P_n of the boundary. Update the value of *dir*.
3. If $P_n = P_1$ *i* $P_{n-1} = P_0$, then stop, else go to 2.
4. Internal boundary is determined by points $P_0, \ldots, P_{n-2}$

Basics of region representation and description

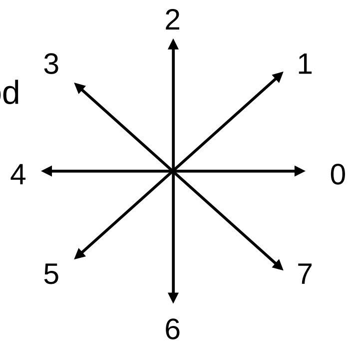
Directions for

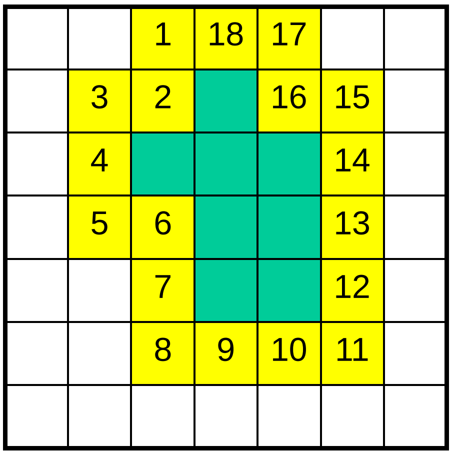
4-neighborhood

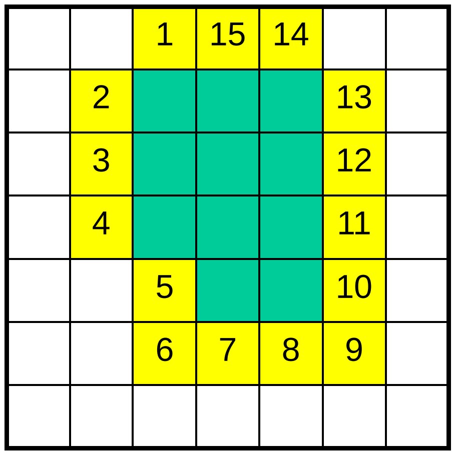


Directions for

8-neighborhood

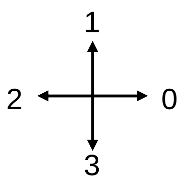


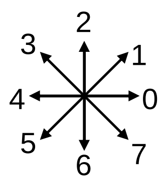


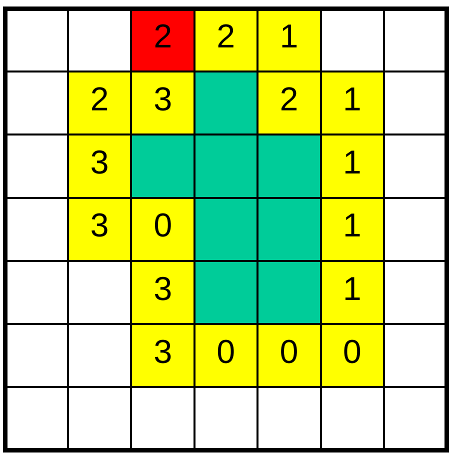


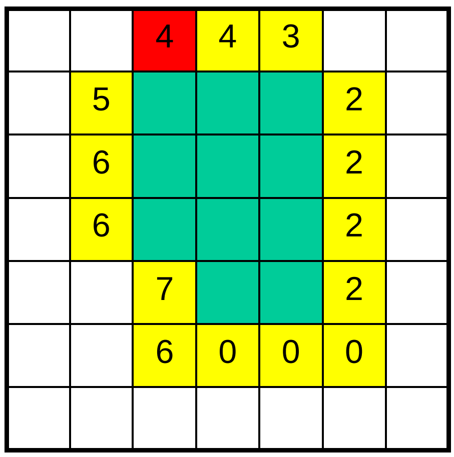
Basics of region representation and description

- Chain codes









323303300011112122

566760002222344

Basics of region representation and description

- Statistical moments for digital images
  - Geometrical moments of p,q order

$$m_{pq} = \sum_{i=1}^{k} \sum_{j=1}^{k} i^p j^q f(i,j)$$

$$m_{00} = \sum_{i=1}^{k} \sum_{j=1}^{k} f(i,j)$$ - the number of region pixels (area of the object)

$$\bar{i} = m_{10}/m_{00}, \quad \bar{j} = m_{01}/m_{00}$$ - center of gravity coordinates (position of the object)

  - Central moments

$$\mu_{pq} = \sum_{i=1}^{k} \sum_{j=1}^{k} (i-\bar{i})^p (j-\bar{j})^q f(i,j)$$

Basics of region representation and description

  - Central moments of order up to 3 with using normal moments

$$\mu_{00} = m_{00}$$
$$\mu_{10} = \mu_{01} = 0$$
$$\mu_{11} = m_{11} - \bar{j}m_{10} = m_{11} - \bar{i}m_{01}$$
$$\mu_{20} = m_{20} - \bar{i}m_{10}$$
$$\mu_{02} = m_{02} - \bar{j}m_{01}$$
$$\mu_{30} = m_{30} - 3\bar{i}m_{20} + 2\bar{i}^2 m_{10}$$
$$\mu_{03} = m_{03} - 3\bar{j}m_{02} + 2\bar{j}^2 m_{01}$$
$$\mu_{21} = m_{21} - 2\bar{i}m_{11} - \bar{j}m_{20} + 2\bar{i}^2 m_{01}$$
$$\mu_{12} = m_{12} - 2\bar{j}m_{11} - \bar{i}m_{02} + 2\bar{j}^2 m_{10}$$

These moments are the basis for determining the 7 moments invariant to shift, rotation, and scale

Basics of region representation and description

- Hu moments (invariant to shift, rotation and scale transformation)
  - Normalized central moments η_{pq}

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \gamma = \frac{p+q}{2} + 1$$

  - Hu moments

$$\begin{aligned}
N_1 &= \eta_{20} + \eta_{02} \\
N_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
N_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
N_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
N_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
N_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
N_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}$$

Basics of region representation and description

- Determination of the position and orientation (direction) of objects
  - Location of the object on the image can be determined by the coordinates of the center of gravity of the area

$$\bar{i} = m_{10}/m_{00} = \frac{1}{P}\sum_{i=1}^{P} i_P, \quad \bar{j} = m_{01}/m_{00} = \frac{1}{P}\sum_{j=1}^{P} j_P$$

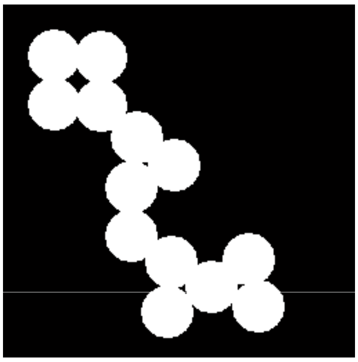
P - number of region pixels

  - The orientation is defined for the slender objects. As the orientation is assumed direction of the axis with the smallest inertia (which corresponds to the direction of the longer side of the circumscribing rectangle)

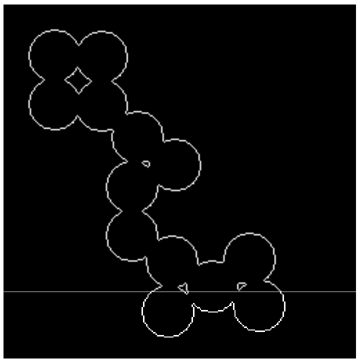
$$\theta = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

Basics of region representation and description

- Segmentation and counting of objects in the image

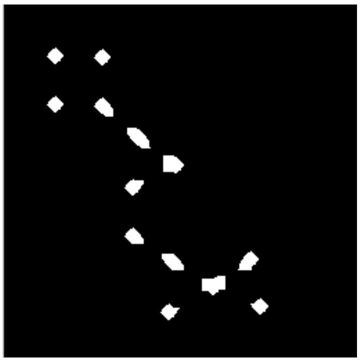


Edge detection ?



With using erosion

```
i=imread('circles.png');
in=bwmorph(i,'erode',10);
[il,num]=bwlabel(in);
figure,imshow(in)
disp(‘Number of
objects:');num
```



Basics of region representation and description

– Counting of objects using distance transform

Distance transform is based on addition of intermediate results of the subsequent erosion of the image

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

→

0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	2	2	1	0	0
0	1	2	2	2	1	0
0	0	1	2	2	1	0
0	0	0	1	1	0	0
0	0	0	0	1	0	0

→

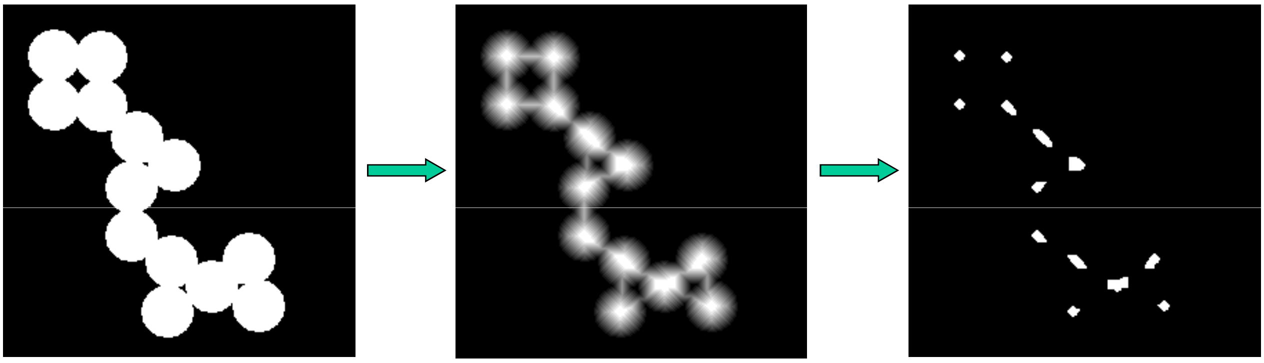
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	2	2	1	0	0
0	1	2	3	2	1	0
0	0	1	2	2	1	0
0	0	0	1	1	0	0
0	0	0	0	1	0	0

Example of Distance transform implementation in Matlab

```
s=[0 1 0; 1 1 1; 0 1 0];
in=uint8(ib);ib=uint8(ib);
for j=1:20
    ib=imerode(ib,s);
    in=in+uint8(ib);
end
```

Basics of region representation and description

– Example of distance transform for objects counting

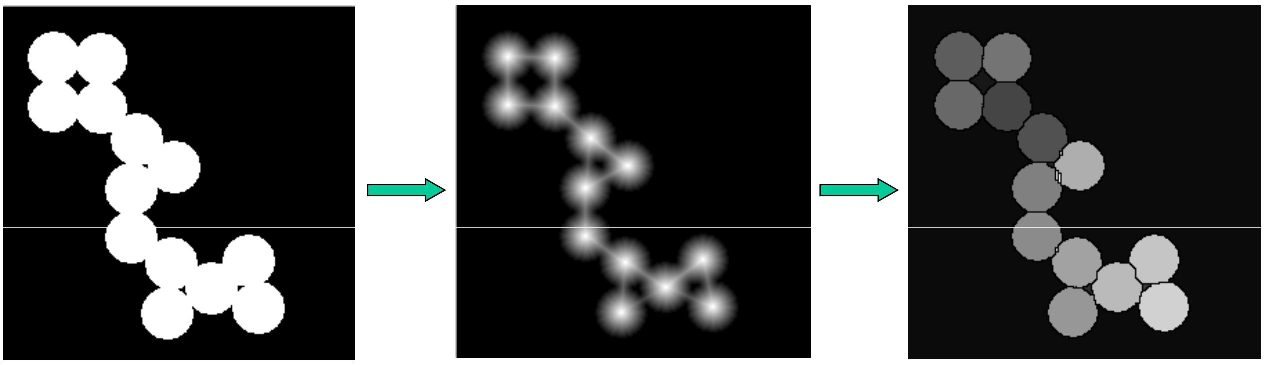


Result of distance transform after histogram stretching

Binarization

Basics of region representation and description

– Distance transform and watershed segmentation example



Distance transform

Watershed segmentation