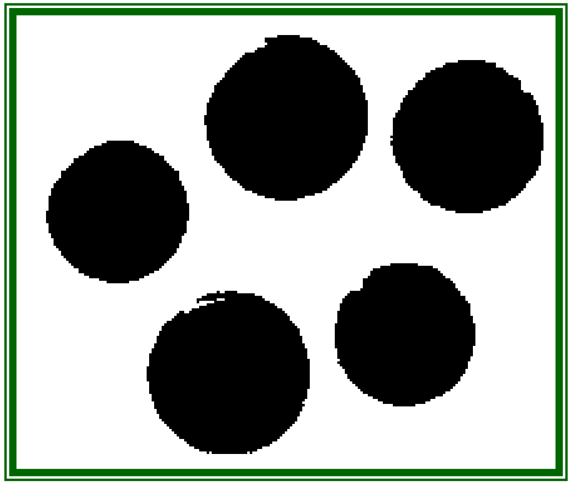
Region analysis

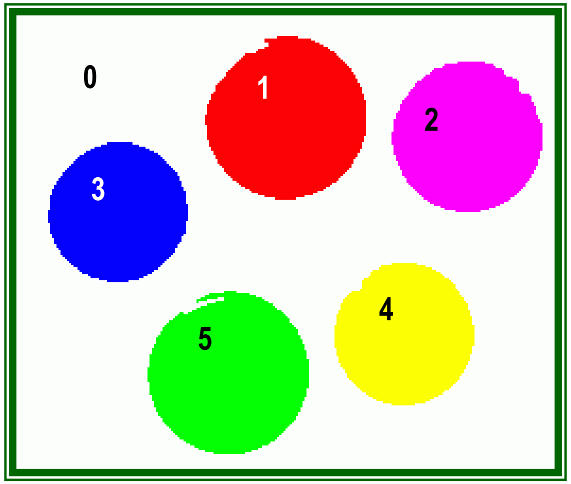
Connected Components

Moment analysis

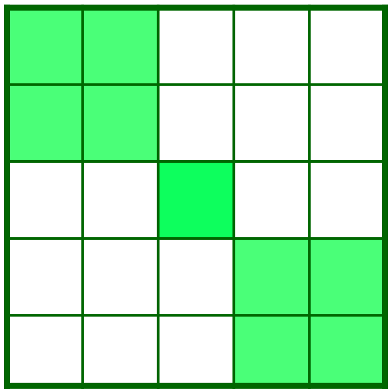
Connected components

- Given a binary image, you want to get a new image where each region has a different label (color)





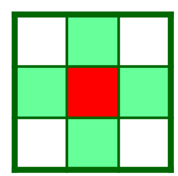
Connectivity



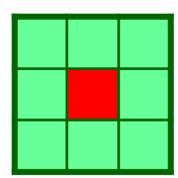
- 8 connection
  - 1 blob
  - background is connected
- 4 connection
  - 3 blobs
  - background is not connected

Remember the two definitions of neighbor:

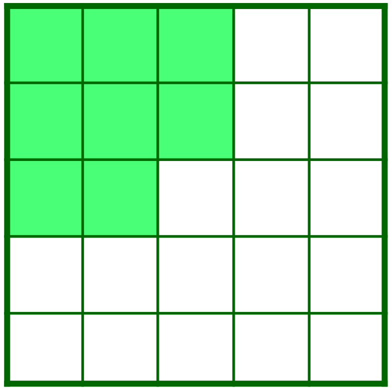
4 connection

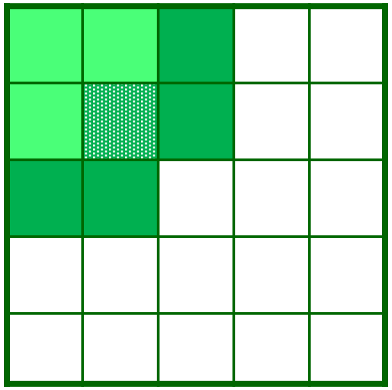


8 connection



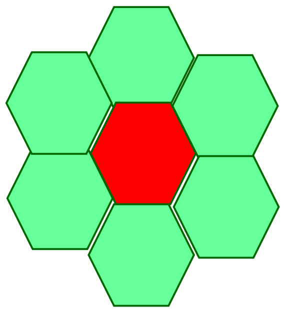
Connectivity and outline



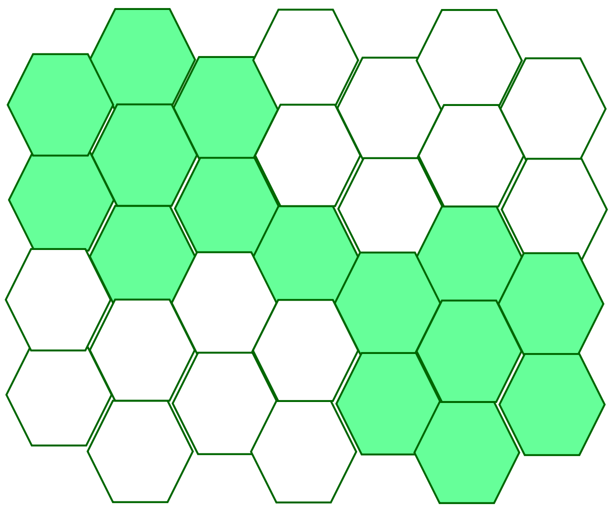


- 8 connection
  - the outline is 4 connected
- 4 connection
  - The outline is 8 connected

Notes on 6 Connectivity



- 6 connection
  - 1 blob
  - background is not connected
  - outline is 6 connected



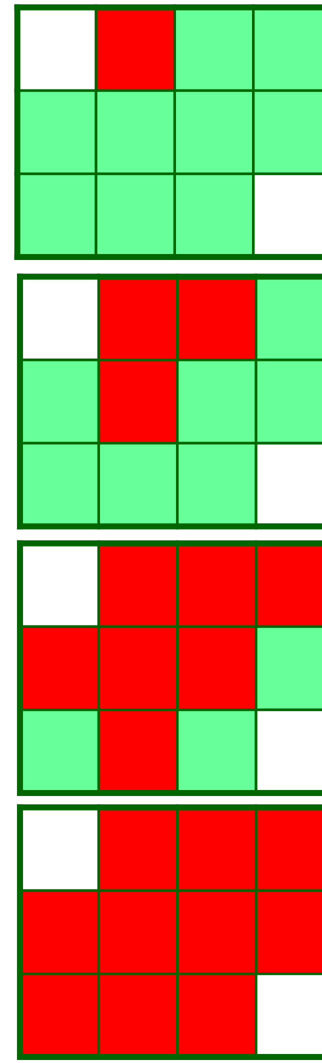
Intuitive algorithm

1. An unlabeled pixel is found
2. Assign a new label to the pixel
3. Repeat while it is possible

   Each nearby pixel, not yet labeled, is assigned the same label

Recursive Implementation

```
void labelling(matrix, i, j, label) {
     if(matrix[i][j]!=unlabeled) return;
     matrix[i][j] = label;
     labelling (matrix, i+1, j, label);
     labelling (matrix, i, j+1, label);
     labelling (matrix, i-1, j, label);
     labelling (matrix, i, j-1, label);
     // diagonal pixels for 8 connection
}
```



- **Possible problems:**
  - **Stack overflow**
  - **Border image pixels**

Classic algorithm

- Two scans of the image (raster scan: line by line, from left to right) are needed
- For each unclassified pixel, the neighboring pixels already classified are considered
  1. If none exist, a new label is introduced and assigned to the pixel
  2. Otherwise, the minimum label is assigned

17	0	15	0	0	0
17	1	1	0	1	0
1	0	0	1	1	0

Current pixel

0: background
1: foreground
Other values: labeled pixels

Classic algorithm

- Two scans of the image (raster scan: line by line, from left to right)
- For each unclassified pixel, the neighboring pixels already classified are considered
  1. If none exist, a new label is introduced and assigned to the pixel
  2. Otherwise, the minimum label is assigned

17	0	15	0	0	0
17	15	1	0	1	0
1	0	0	1	1	0

Rule 2 (adpting 8 connectivity)

Classic algorithm

- Two scans of the image (raster scan: line by line, from left to right)
- For each unclassified pixel, the neighboring pixels already classified are considered
  1. If none exist, a new label is introduced and assigned to the pixel
  2. Otherwise, the minimum label is assigned

17	0	15	0	0	0
17	15	15	0	1	0
1	0	0	1	1	0

Rule 2

Classic algorithm

- Two scans of the image (raster scan: line by line, from left to right)
- For each unclassified pixel, the neighboring pixels already classified are considered
  1. If none exist, a new label is introduced and assigned to the pixel
  2. Otherwise, the minimum label is assigned

17	0	15	0	0	0
17	15	15	0	18	0
1	0	0	1	1	0

Rule 1

Classic algorithm

- Adopting 4 connection the labeling process is different

17	0	15	0	0	0
17	17	15	0	18	0
1	0	0	1	1	0

- If a pixel has multiple neighbors already classified I must keep this fact in mind, to merge different labels into a single one (equivalence classes)
  - For example 15 e 17
- The second step is used to reclassify all the pixels using the equivalences

Classic algorithm

											1	1	1	1
											1			1
				2	2	2	2	2						1
				2	2									1
		3	3	2				4						1
			3	2	2	2	2	2	2					1
									2					1
			5			6			2	2				1
			5	5	5	5								1
7	7	7	5								8			1
7														1
7	7	7	7	7	7	7	7	7	7	7	7	7	7	1

											1	1	1	1
											1			1
				2	2	2	2	2						1
				2	2									1
		2	2	2				2						1
			2	2	2	2	2	2	2					1
									2					1
			1			1			2	2				1
			1	1	1	1								1
1	1	1	1								3			1
1														1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Using 4 connection

Region properties

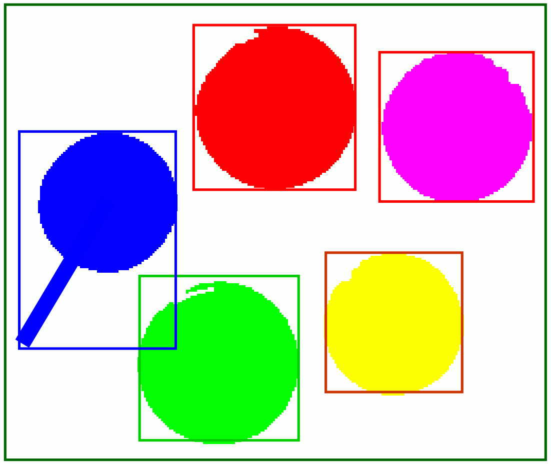
- Each region can be described using some parameters:
  - Area (number of pixels)
  - Center of gravity (r, c)
  - Bounding box (the rectangle in which the region is completely included
  - Perimeter (number of pixels that have pixels that do not belong to the region as neighbors)

$$A = \sum_{(i,j)\in R} 1$$

$$r = \frac{1}{A} \sum_{(i,j)\in R} i \qquad c = \frac{1}{A} \sum_{(i,j)\in R} j$$

Bounding box

Bounding boxes can be overlapped



A bounding box is defined by 4 parameters:

$i_{min}, i_{max}, j_{min}, j_{max}$

Moments

- A moment of order *k*, *l* is defined as:

$$m_{k,l} = \frac{1}{A} \sum_{(i,j)\in R} i^k j^l$$

- It can be shown that by knowing all the moments, the shape of the object can be reconstructed
- Moments of order (1,0) and (0,1) are the center of gravity

Moments

- **Higher order moments are generally referred to the center of gravity (invariance with respect to translation is obtained)**

$$\mu_{k,l} = \frac{1}{A} \sum_{(i,j)\in R} (i-r)^k (j-c)^l$$

Moments

- Usually only second order moments (moments of inertia of physics) are used (the second formulas can be calculated with a single image scan)

$$\mu_{rr} = \frac{1}{A}\sum_{(i,j)\in R}(i-r)^2 = \frac{1}{A}\sum_{(i,j)\in R} i^2 - r^2$$

$$\mu_{cc} = \frac{1}{A}\sum_{(i,j)\in R}(j-c)^2 = \frac{1}{A}\sum_{(i,j)\in R} j^2 - c^2$$

$$\mu_{rc} = \frac{1}{A}\sum_{(i,j)\in R}(i-r)(j-c) = \frac{1}{A}\sum_{(i,j)\in R} ij - rc$$

Moments

- The moments are not invariant under rotation

$$\mu_{cc}(\theta) = \mu_{cc}\cos(\theta)^2 + 2\mu_{rc}\sin(\theta)\cos(\theta) + \mu_{rr}\sin(\theta)^2$$

$$\mu_{rr}(\theta) = \mu_{rr}\cos(\theta)^2 - 2\mu_{rc}\sin(\theta)\cos(\theta) + \mu_{cc}\sin(\theta)^2$$

$$\mu_{rc}(\theta) = ½(\mu_{cc} - \mu_{rr})\sin(2\theta) - \mu_{rc}\cos(2\theta)$$

$\mu(\theta)$ rotated image moment

$\mu = \mu(0)$ moment of original image

Moments

- The formula

$$\mu_{cc}(\theta)+\mu_{rr}(\theta)=\mu_{cc}+\mu_{rr}$$

Is invariant under rotation

$$\mu_{cr}(\theta)=\frac{1}{2}(\mu_{cc}-\mu_{rr})\sin(2\theta)-\mu_{cr}\cos(2\theta)=0$$

- There is an angle for which the moment cr is zero

$$\tan(2\theta)=\frac{2\mu_{cr}}{\mu_{cc}-\mu_{rr}}$$

Moments

- If we calculate the moments for this angle we obtain the principal axes of inertia
- If the region has circular symmetry the principal moments are equal and the mixed moment is zero

Gray images

- For each region two parameters can be evaluated:
  - average
  - variance

$$m = \frac{1}{A} \sum_{(i,j) \in R} I(i,j)$$

$$\sigma^2 = \frac{1}{A} \sum_{(i,j) \in R} \left(I(i,j) - m \right)^2 = \frac{1}{A} \sum_{(i,j) \in R} I(i,j)^2 - m^2$$

Variance minimization can be one of the criteria for binarizing an image

Gray images

- You can introduce many other parameters, for example
  - the weighted center of gravity
  - … …

$$w = \sum_{(i,j)\in R} I(i,j)$$

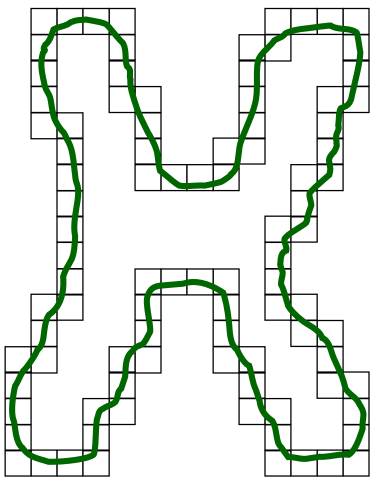
$$r = \frac{1}{w} \sum_{(i,j)\in R} iI(i,j)$$

$$c = \frac{1}{w} \sum_{(i,j)\in R} jI(i,j)$$

If $I(i,j)=1\ \forall i,j$ **we fall back into previous definitions**

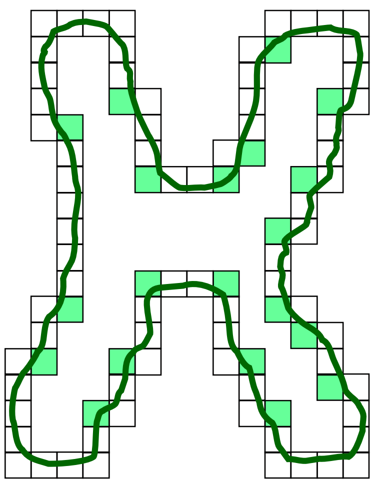
Contour representation

- A contour can be described by the sequence of corresponding pixels
- $X(k) = x_k$, $Y(k) = y_k$ $0 \leq k < K$ (K number of border pixels)



Contour representation

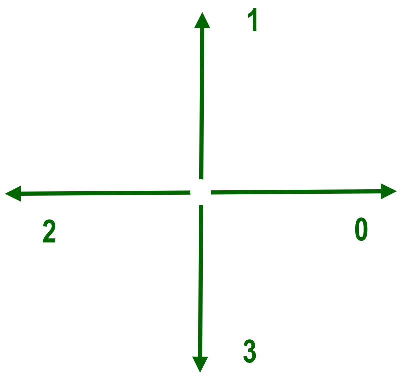
- A contour can be described by the sequence of corresponding pixels
  - $X(k) = x_k$, $Y(k) = y_k$ $0 \leq k < K$

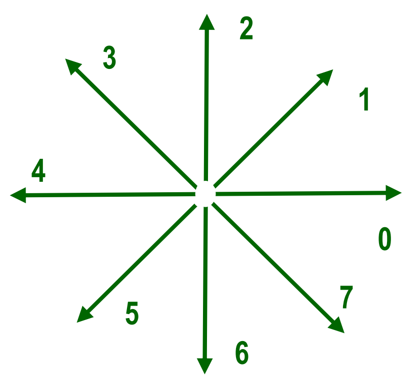


The size of the vectors (K) depends on the type of connection chosen

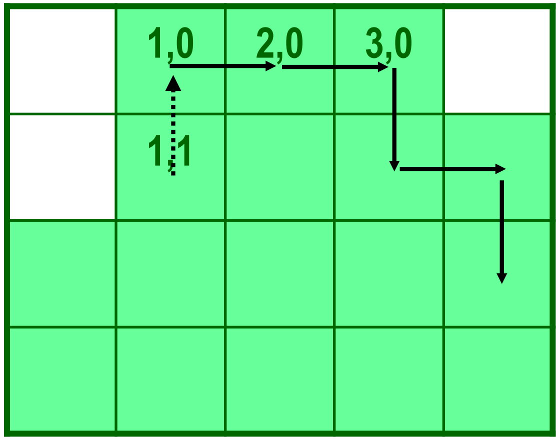
Freeman Code

- A more compact representation can be obtained using the Freeman code (chain code)
  - The first pixel is stored and for each subsequent pixel the spatial relationship between it and the previous one

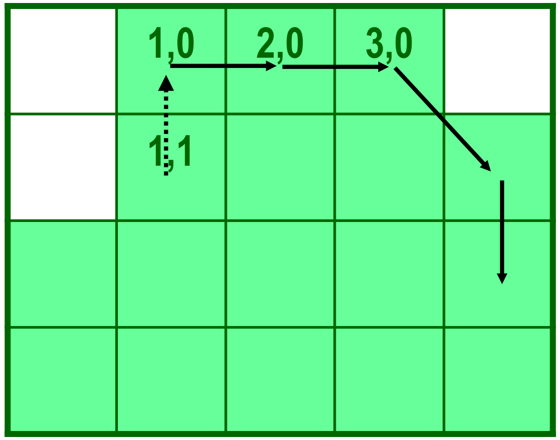




Example



(1,0), 0, 0, 3, 0, 3, …, 1
2 bits per pixel are enough



(1,0), 0, 0, 7, 6, …, 2
3 bits per pixel are enough

If we neglect the initial pixel the representations are translation invariant

Fourier Descriptors

- A contour can also be represented as a sequence of complex numbers:

  $s(k) = x_k + jy_k \quad 0 \leq k < K$

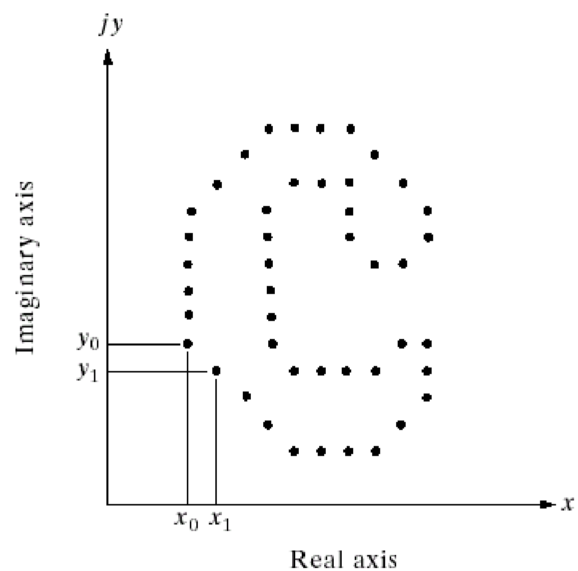


FIGURE 11.13 A digital boundary and its representation as a complex sequence. The points (x_0, y_0) and (x_1, y_1) shown are (arbitrarily) the first two points in the sequence.

Discrete Fourier transform

$$a(u) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/K}$$

Direct transform

$$s(k) = \frac{1}{K}\sum_{u=0}^{K-1} a(u)e^{j2\pi uk/K}$$

Inverse transform

Transformation	**Boundary**	**Fourier Descriptor**
Identity	$s(k)$	$a(u)$
Rotation	$s_r(k) = s(k)e^{j\theta}$	$a_r(u) = a(u)e^{j\theta}$
Translation	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
Scaling	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Starting point	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

TABLE 11.1 Some basic properties of Fourier descriptors.

Example of reconstruction