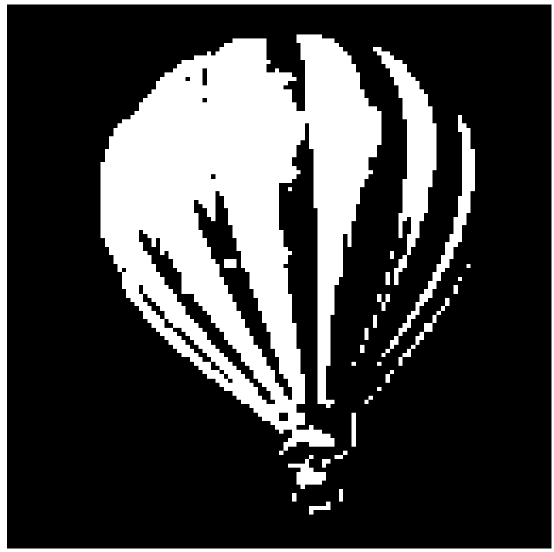
Binary Images (Part I)

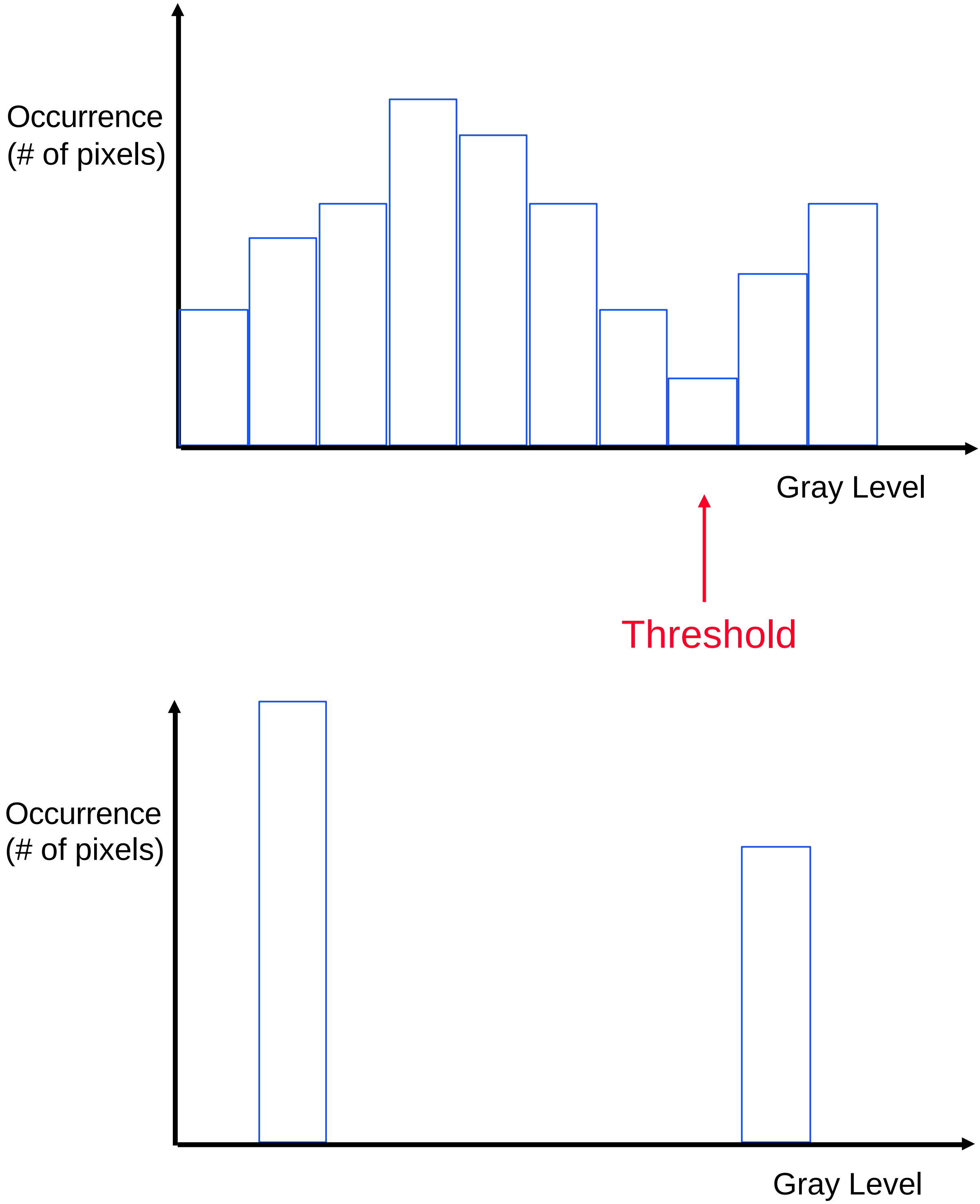
- Threshold
- Binary Image - Definition
- Connected Components
- Chain Code
- Edge Following

Grayscale Image

Binary Image



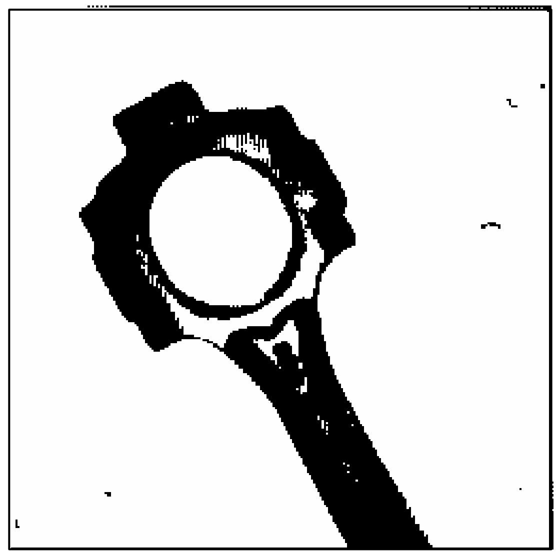
Thresholding



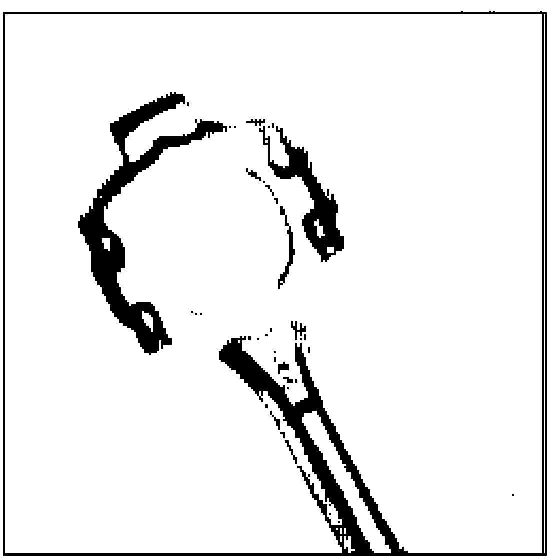
Thresholding a Grayscale Image

Original Image

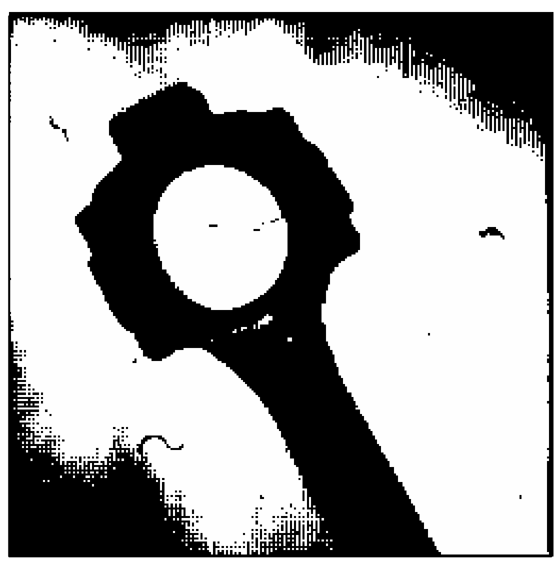
Binary Image



Threshold too low



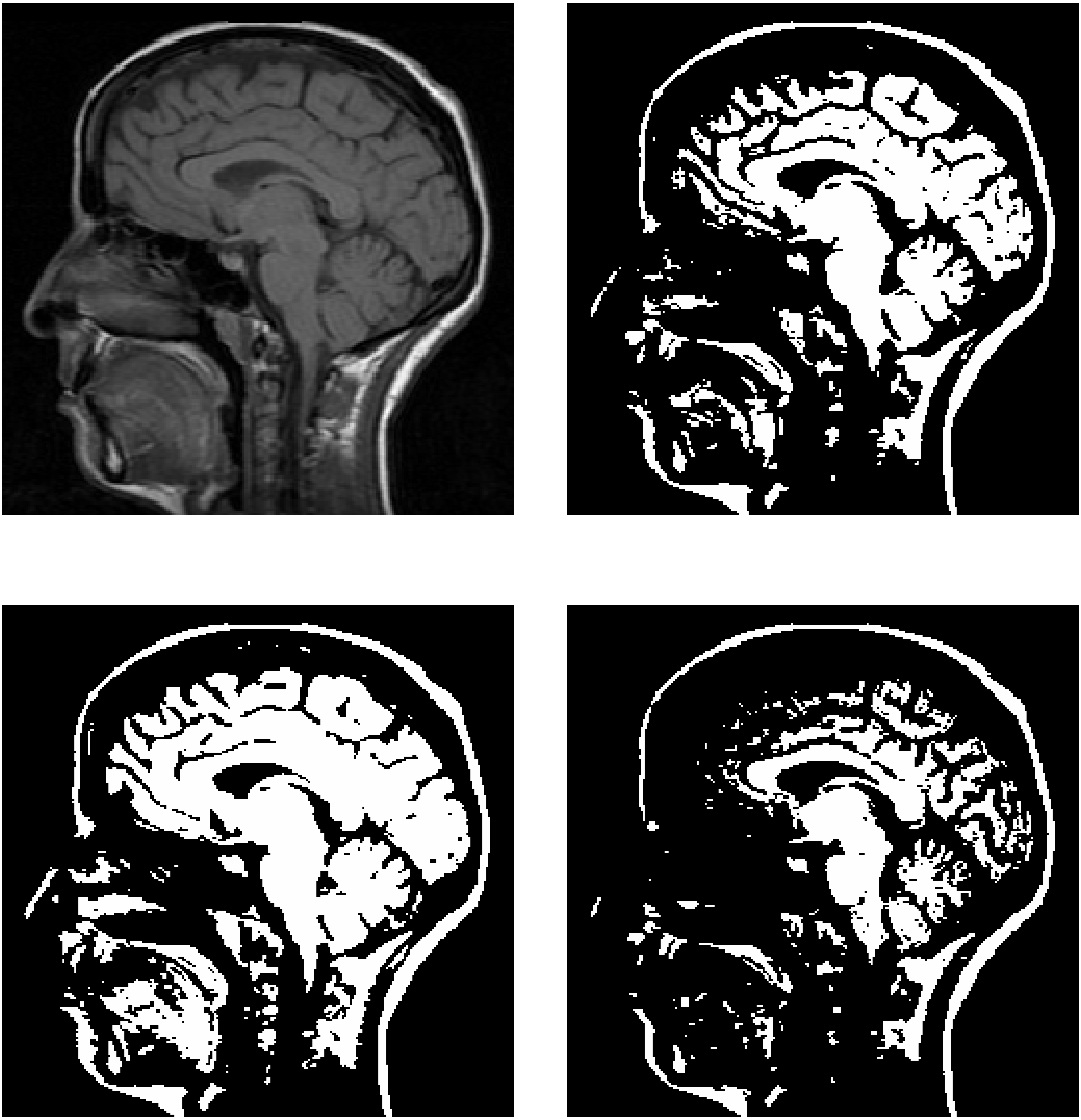
Threshold too high



FMRI - Example

Original Image

Threshold = 80



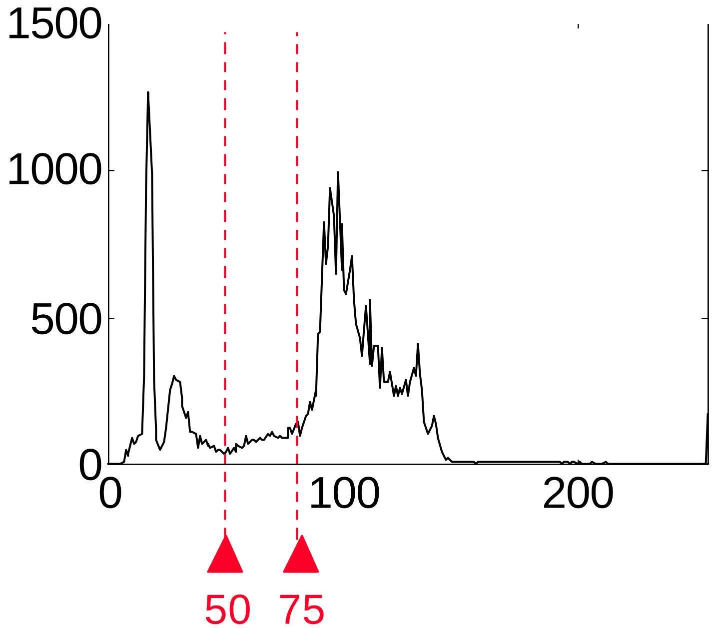
Threshold = 71

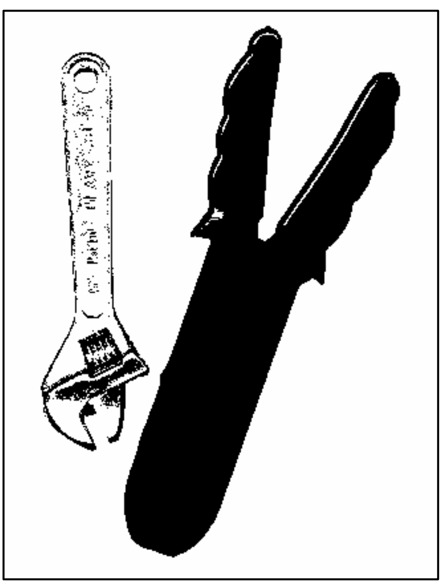
Threshold = 88

Segmentation using Thresholding

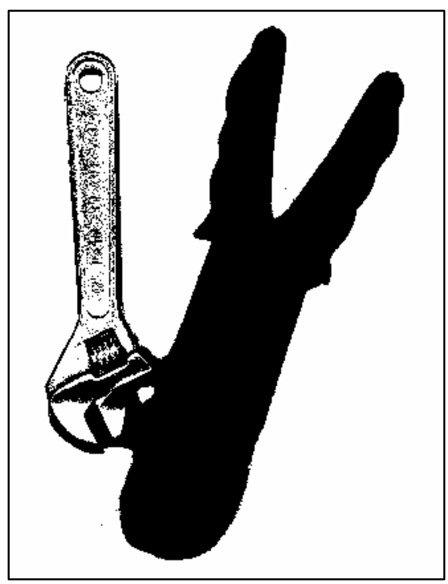
Original

Histogram





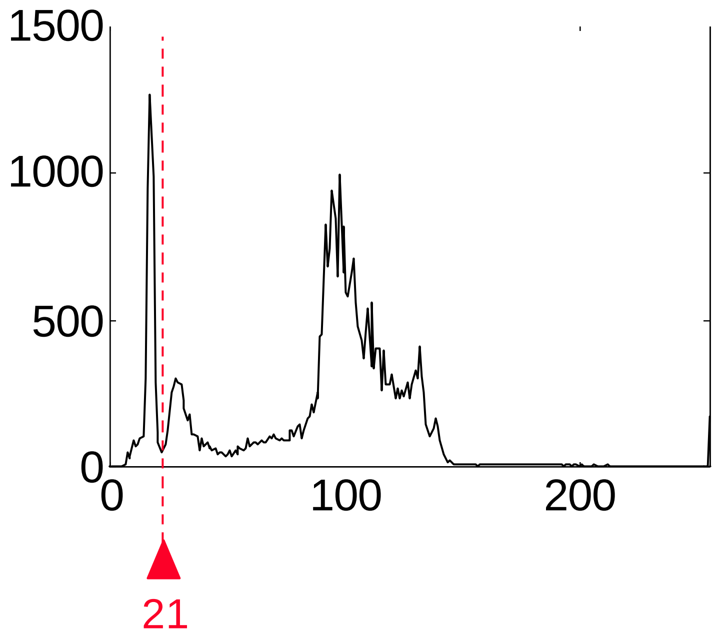
Threshold = 50

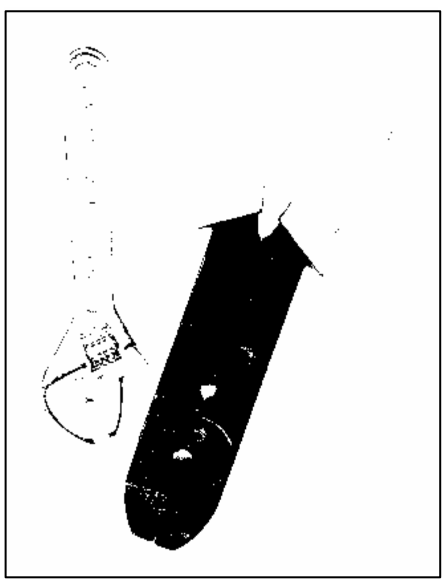


Threshold = 75

Original

Histogram



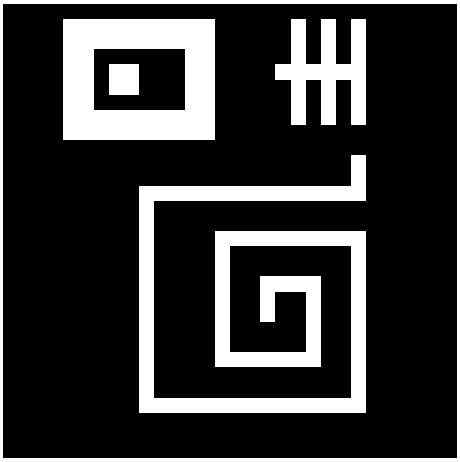


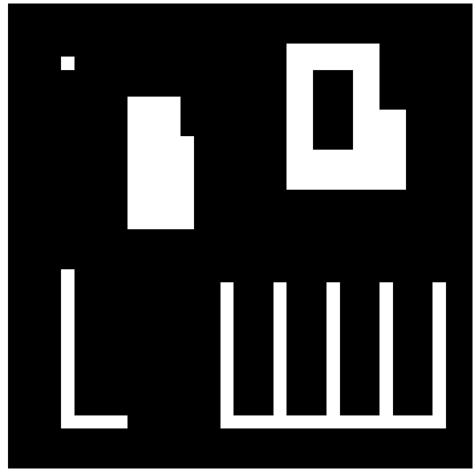
Threshold = 21

Binary Image = Figure + Ground

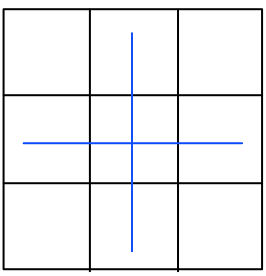
(Edgar Rubin 1915)

Connected Components

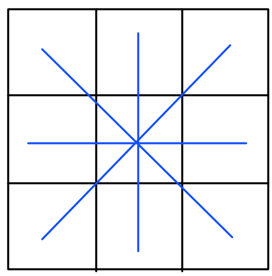




Neighborhoods:



4-neighbor metric



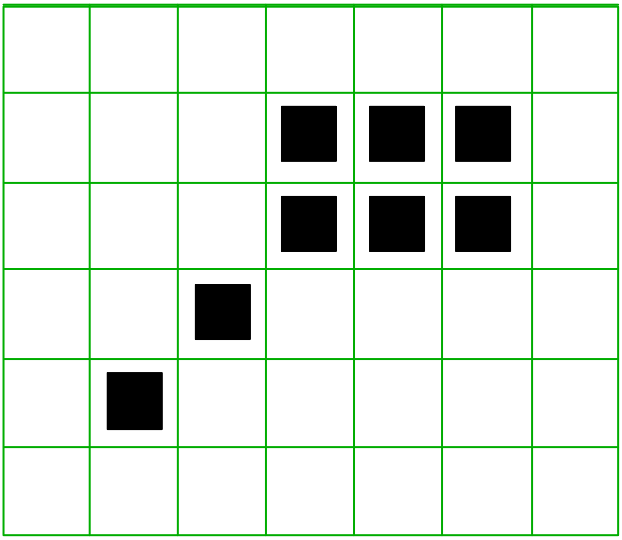
8-neighbor metric

Connected Components:

S = the set of object pixels

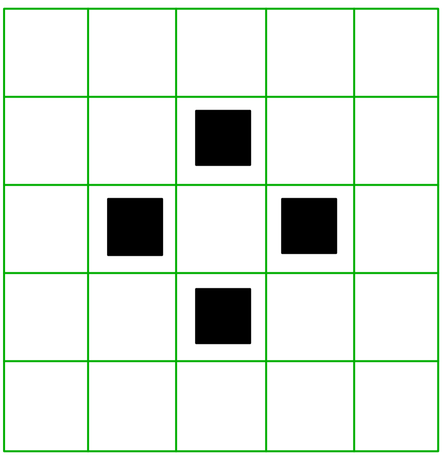
S is a Connected Component if for each pixel pair $(x_1, y_1) \in S$ and $(x_2, y_2) \in S$ there is a path passing through X-neighbors in S. (X = 4,8).

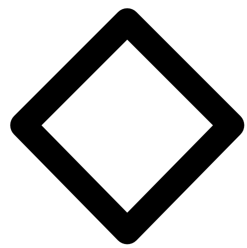
S may contain several connected components.



1 connected component-8

3 connected components-4





8-neighborhood:

1 object connected component

1 background connected component

4-neighborhood:

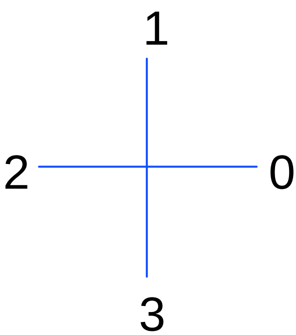
2 background connected components

4 object connected components

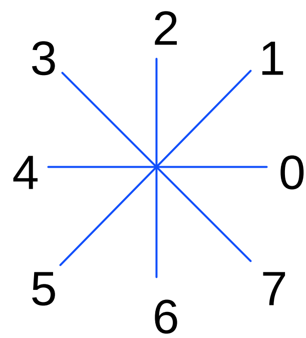
Always choose different neighborhood metrics for objects and backgrounds.

Chain Code

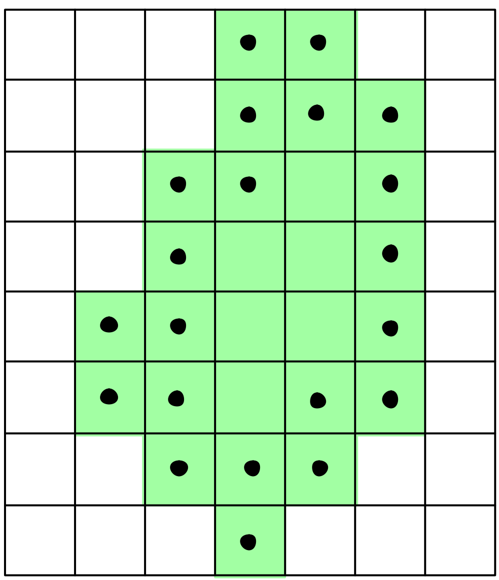
Each direction is assigned a code:

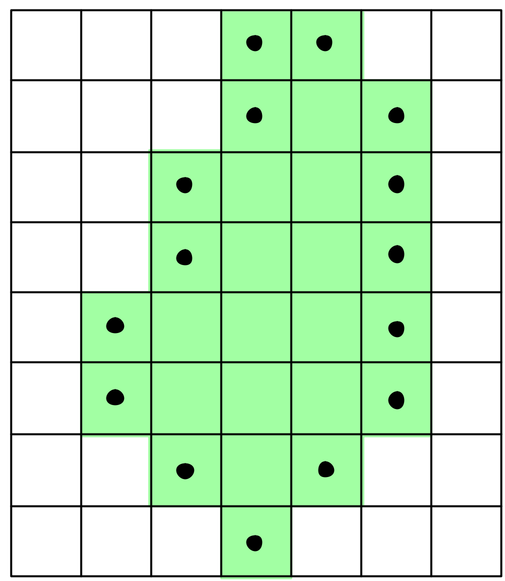


4-neighbor

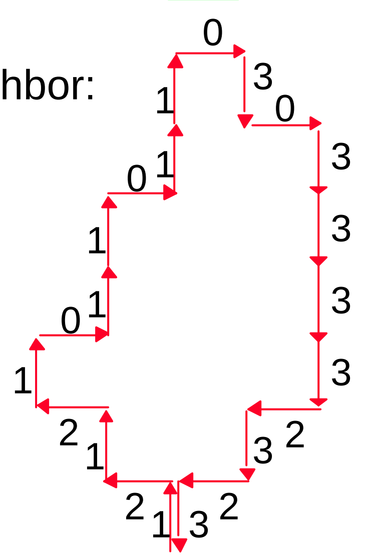


8-neighbor

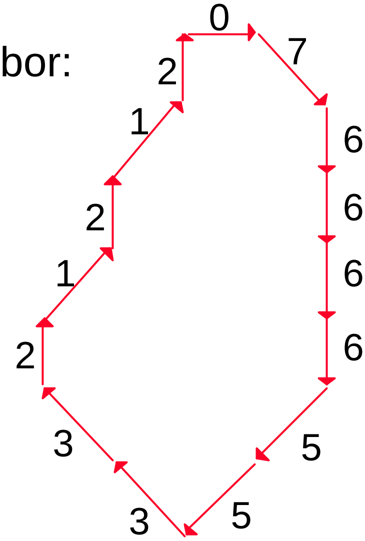




4-neighbor:



8-neighbor:



0303333232312121011011

076666553321212

Marking the Connected Components

Connected Component Algorithm: Two passes over the image.

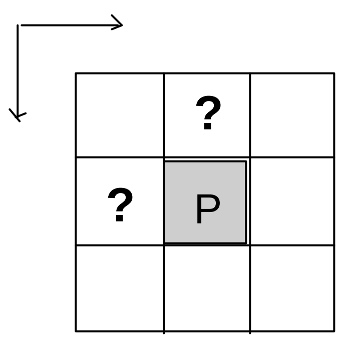
Pass 1:

Scan the image pixels from left to right and from top to bottom.
For every pixel P of value 1 (an object pixel), test top and left neighbors (4-neighbor metric).

- If 2 of the neighbors are 0: assign a new mark to P.
- If 1 of the neighbors isn't 0: assign the neighbor's mark to P.
- If 2 of the neighbors are not 0: assign the left neighbor's mark to P

Pass 2:

Divide all marks to equivalence classes (marks of neighboring pixels are considered equivalent).
Replace each mark with the number of its equivalence class.



Connected Components - Example

Original Binary image

		1	1				
1	1	1	1		1	1	1

Pass 1:

		1	1				
2	2	2	2		3	3	3

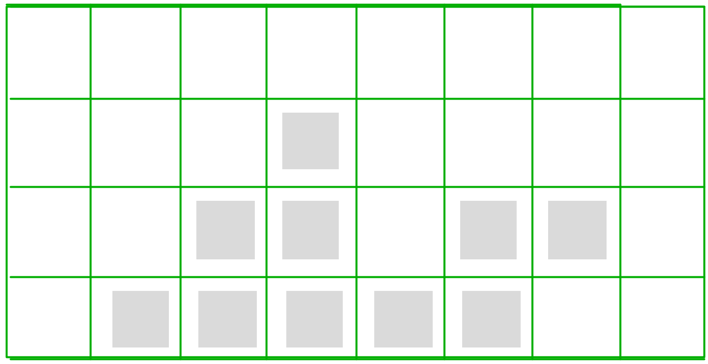
Pass 2:

		1	1				
1	1	1	1		2	2	2

Equivalence Class number	Original mark
1	1,2
2	3

Connected Components - Example II

Original Binary image



Pass 1:

			1				
		2	2		3	3	
	4	4	4	4	4		

Pass 2:

			1				
		1	1		1	1	
	1	1	1	1	1		

Equivalence Class number	Original mark
1	1,2,3,4

Edges

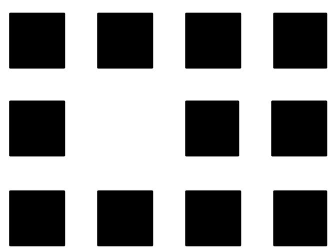
C = connected component of object S.

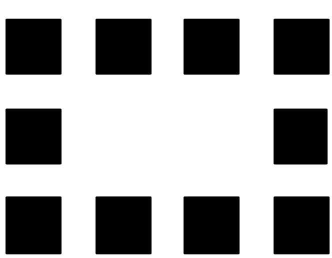
D = connected component of $\bar{S}$.

The D-Edge of C = the set of all pixels in C that have a neighboring pixel in D.
(neighboring-8 if C is 4-connected
neighboring-4 if C is 8-connected).

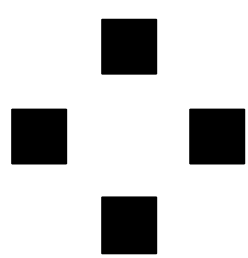
Example:

Object C

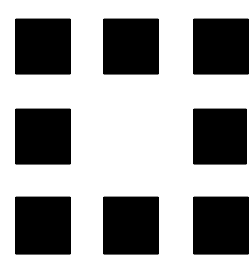




The Edge of C for background D.



(4-neighbor)



(8-neighbor)

The Edge of C for hole D.

Distances

Two grid point: P = (x,y) and Q = (u,v)

Euclidean Distance

$$d_e(P,Q) = \sqrt{(x-u)^2 + (y-v)^2}$$

City Block Distance

$$d_4(P,Q) = |x-u| + |y-v|$$

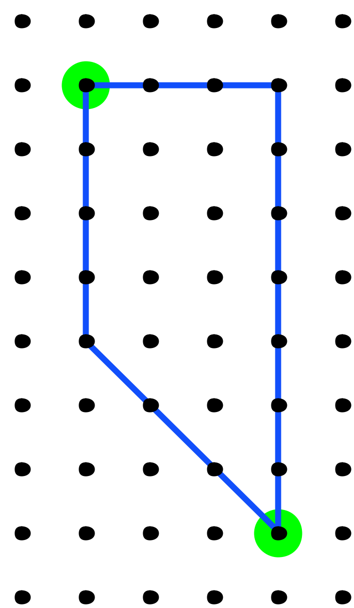
Chessboard Distance

$$d_8(P,Q) = \max(\ |x-u|\ ,\ |y-v|\ )$$

$d_e = 7.6$

$d_8 = 7$

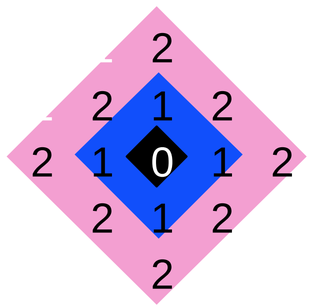
$d_4 = 10$



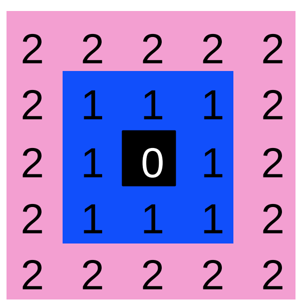
d_e d_8 d_4 are all **metrics**:

1. Distance metric: $d(P,Q) \geq 0$
2. Positive: $d(P,Q) = 0$ iff $P=Q$
3. Symmetric: $d(P,Q) = d(Q,P)$
4. Triangular inequality: $d(P,Q) \leq d(P,R) + d(R,Q)$

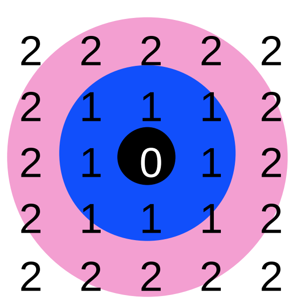
All pixels at equal d_4 distance form a “diamond” :



All pixels at equal d_8 distance form a “square” :



All pixels at equal d_e distance form a “circle” :



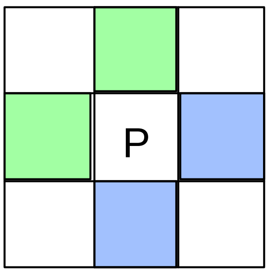
2-Pass Distance Algorithm

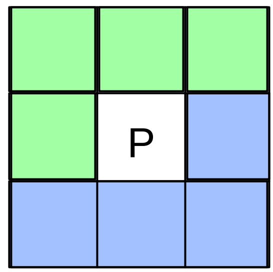
For each pixel calculate the d_4 or d_8 distance from a pixel in set S.

2 passes:

Pass 1: scan image left-to-right and top-to-bottom
Pass 2: scan image right-to-left and bottom-to-top.

For each pixel P mark as follows:





Pass 1: consider all neighbors of P that have been scanned N_1 =

$$d'(P,S) = \begin{cases} 0 & \text{if } P \in S \\ \min\limits_{Q \in N_1} \{d'(Q,S)\}+1 & \text{if } P \notin S \end{cases}$$

Pass 2: consider all neighbors of P that have been scanned N_2 =

$$d''(P,S) = \min_{Q \in N_2} \{d'(P,S)\ ,\ d''(Q,S)+1\}$$

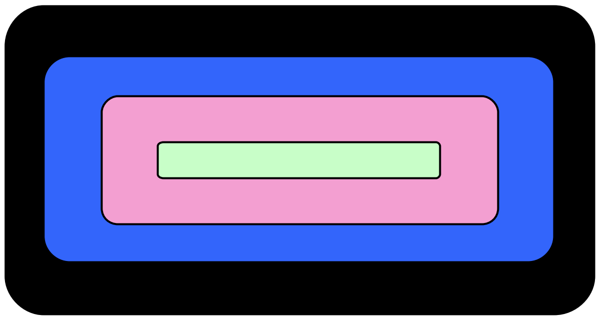
Example measuring d_4:

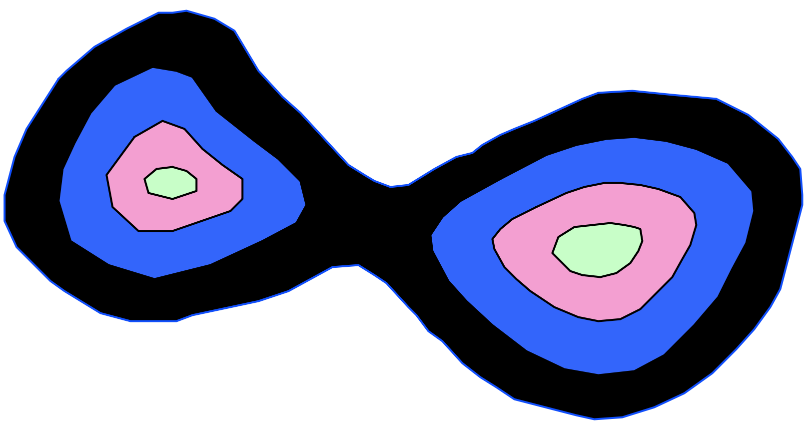
1	0	0	0	0	1	2	3	0	1	2	1
0	0	0	1	1	2	3	0	1	2	1	0
0	0	0	0	2	3	4	1	2	3	2	1

S is marked as 1 — Pass 1: d'(P,S) — Pass 2: d''(P,S)

Skeletons

Consider all edge pixels of an object as the group S.



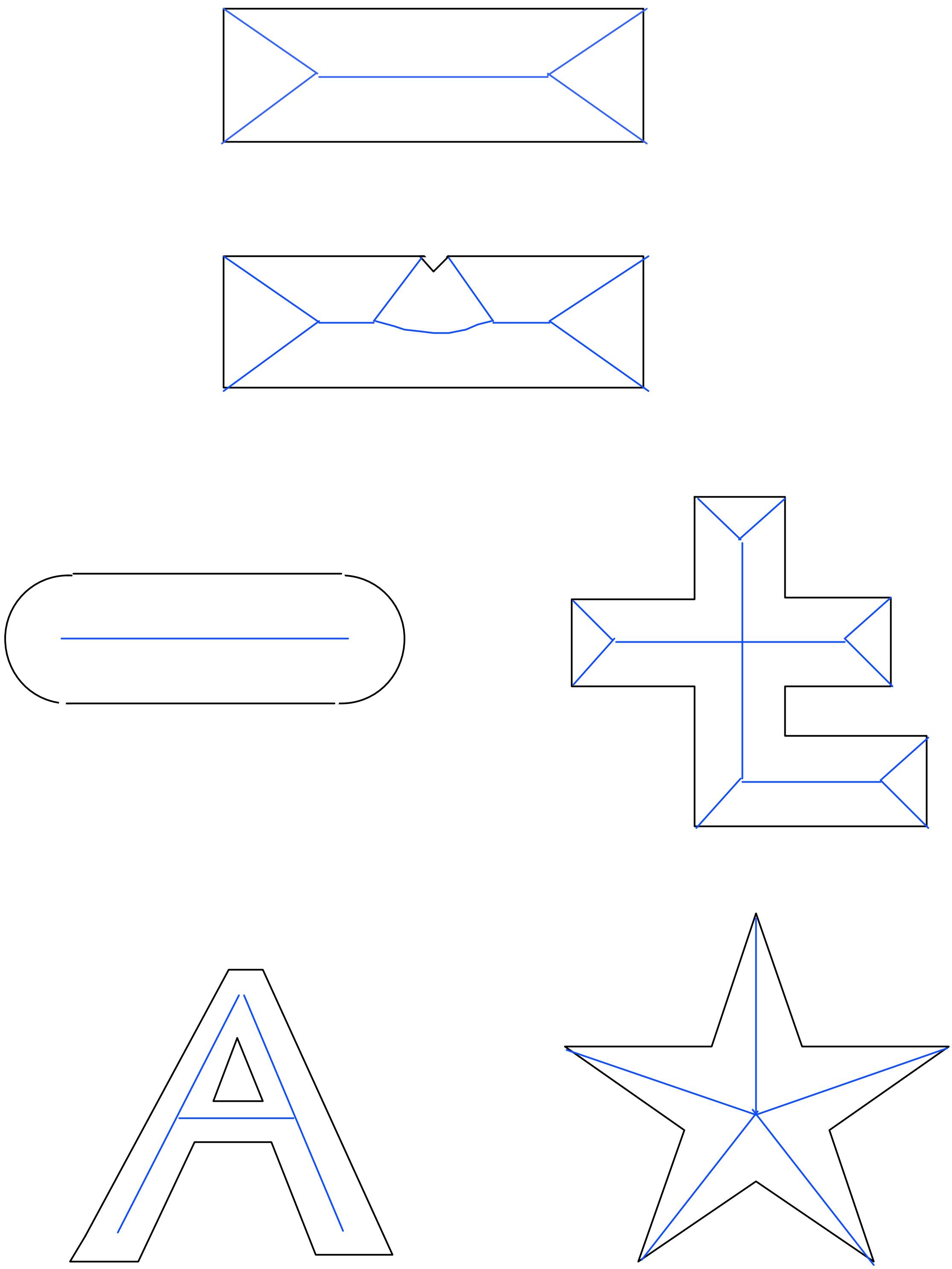


The pixels whose distance is a local maxima are the Skeleton of the object.

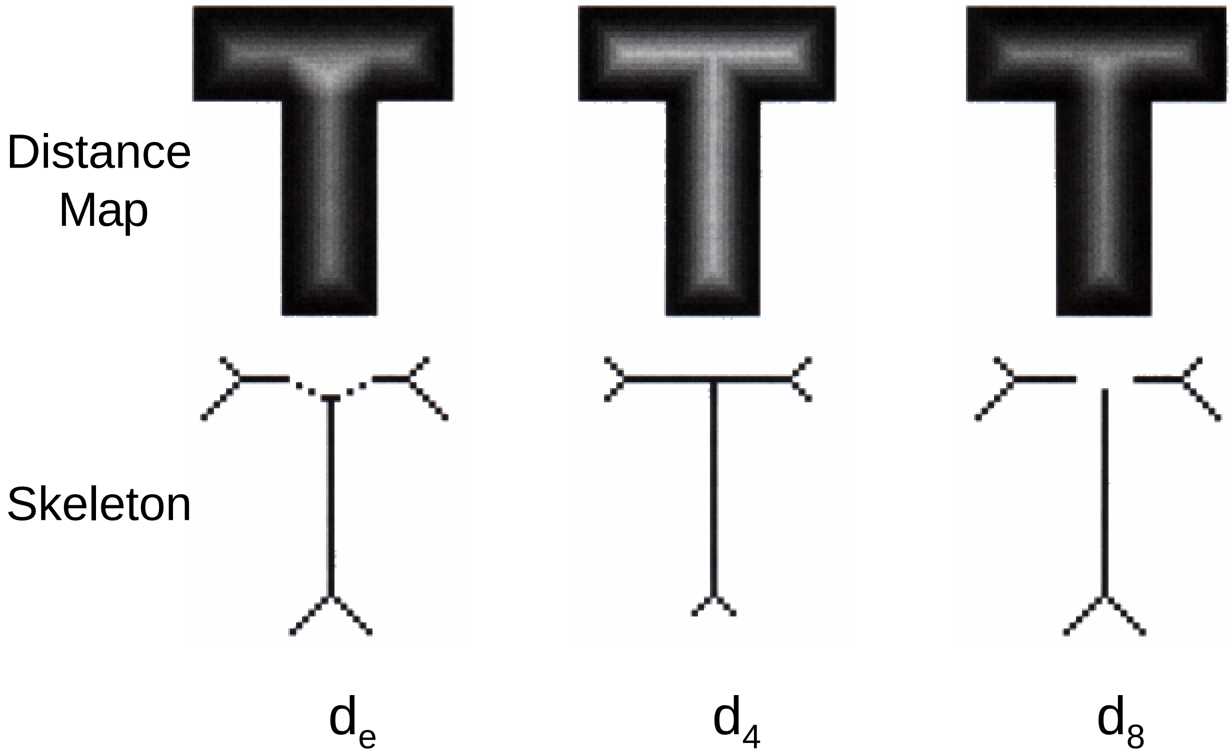
The Skeleton can be used as a shape descriptor.

MAT = Medial Axis Transform

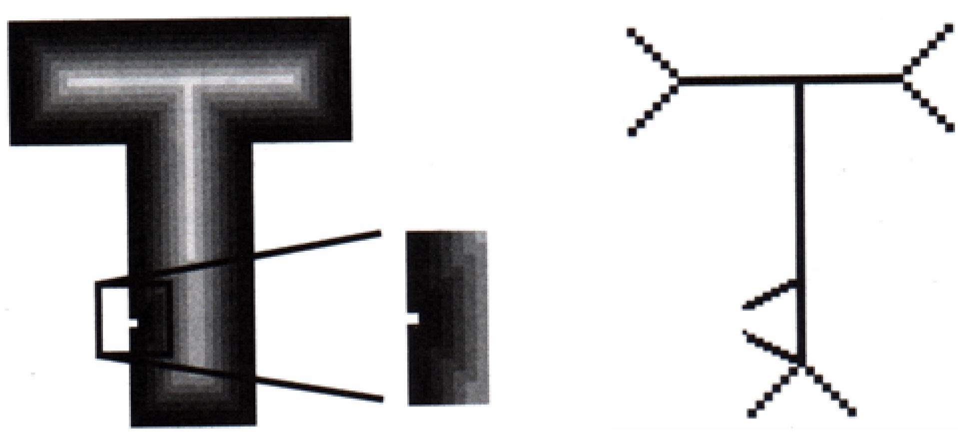
Grass fire technique (Blum, 1993)



Skeletons - Example



Sensitivity to contour changes:



Distance Map

Skeleton