Segmentation

- Threshold Segmentation
    - Local thresholding
    - Edge thresholding
    - Threshold using averaging
    - Gradient Detectors
- Region Growing
- Split & Merge

- Shape Matching
- Shape Representation

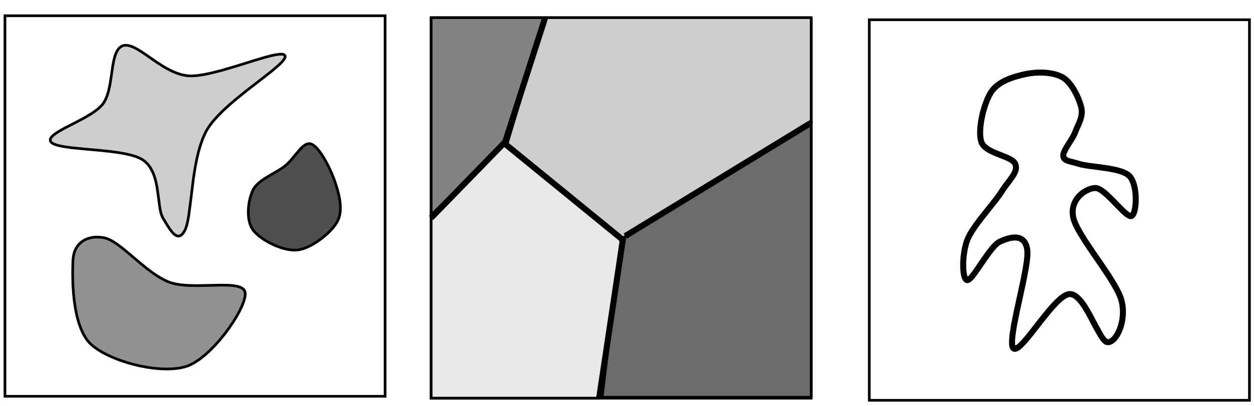
Segmentation

Image Segmentation = divide image into (continuous) regions or sets of pixels.

1) Region Based

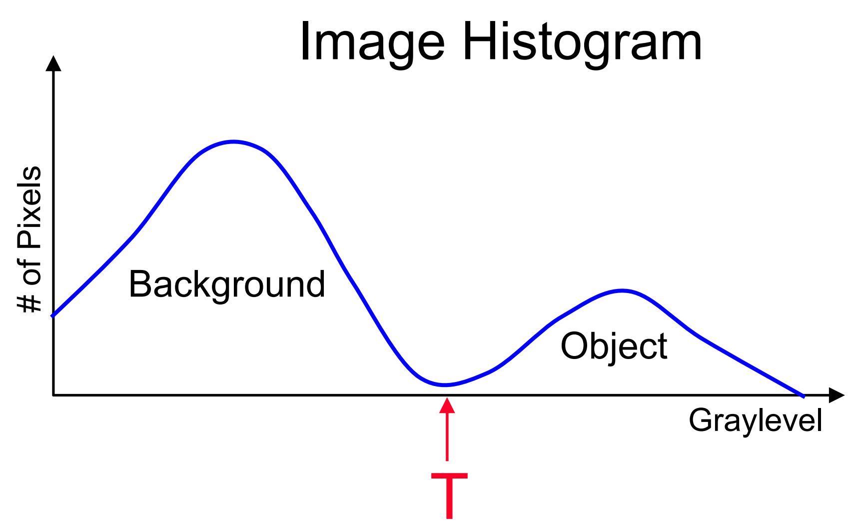
2) Boundary Based

3) Edge Based



Thresholding

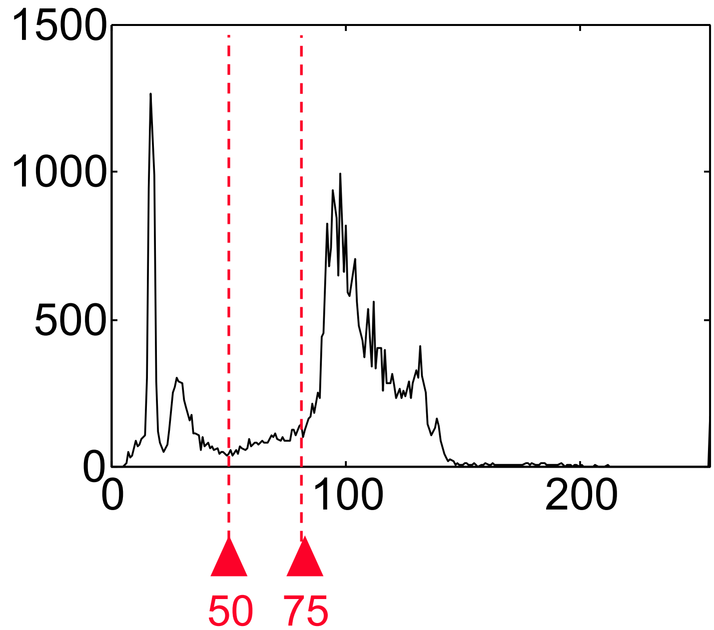
Global Thresholding = Choose threshold T that separates object from background.

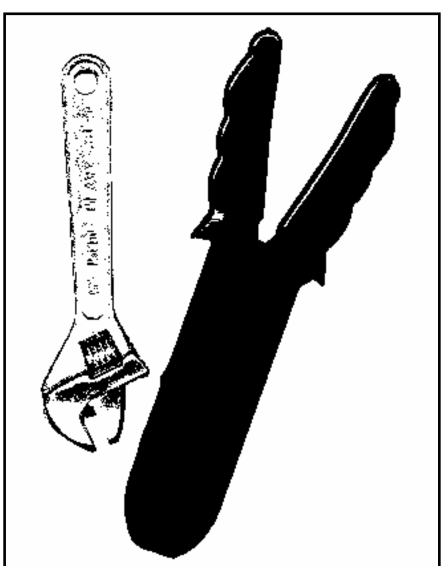


Segmentation using Thresholding

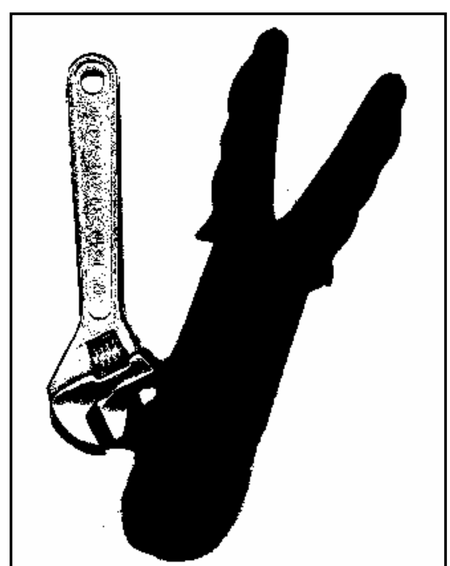
Original

Histogram





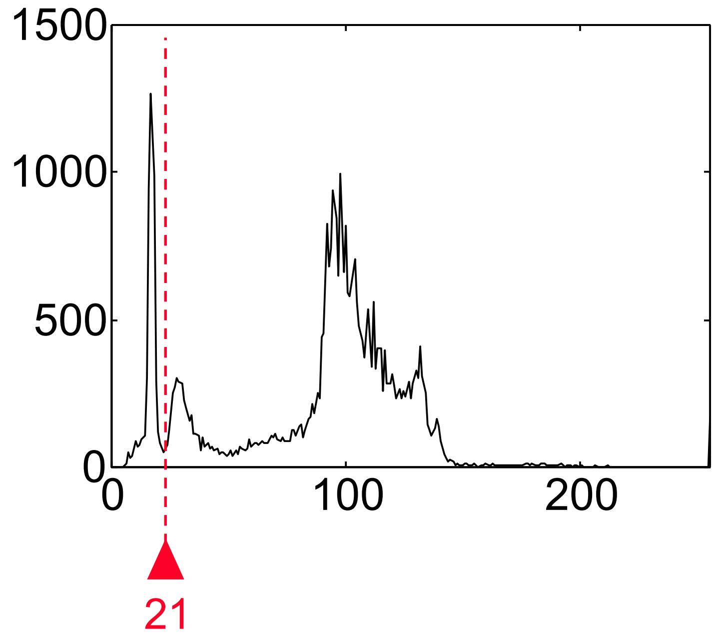
Threshold = 50

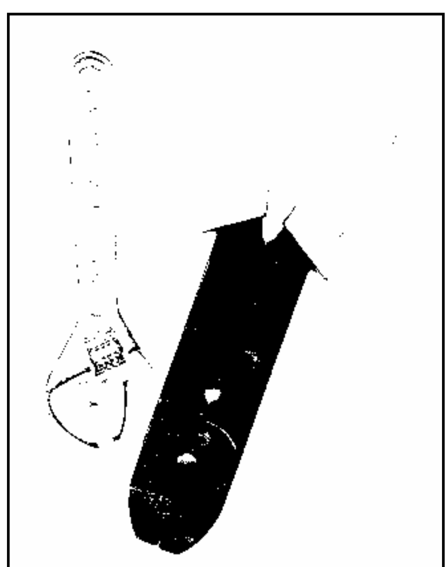


Threshold = 75

Original

Histogram



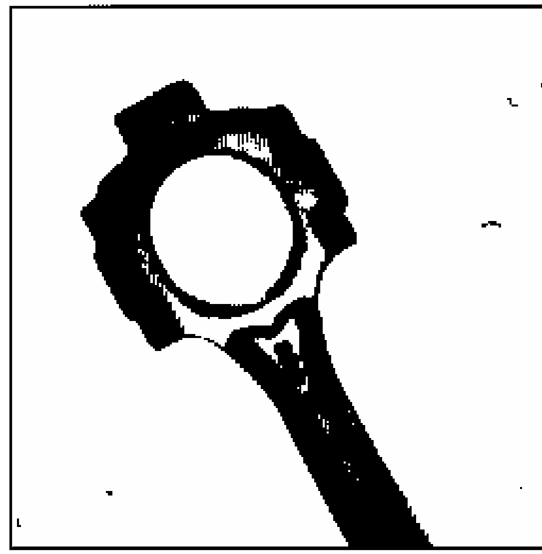


Threshold = 21

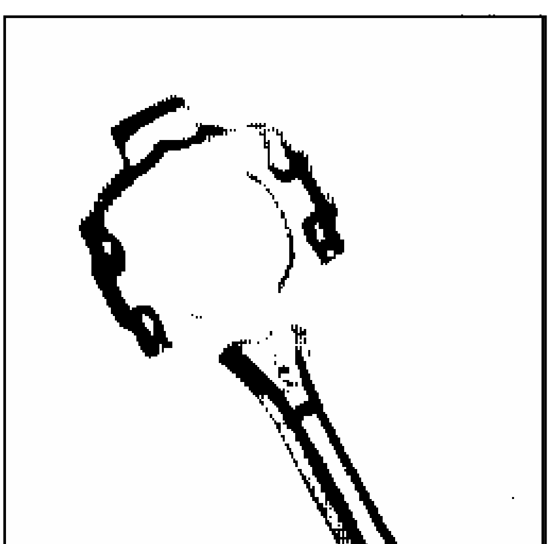
Thresholding a Grayscale Image

Original Image

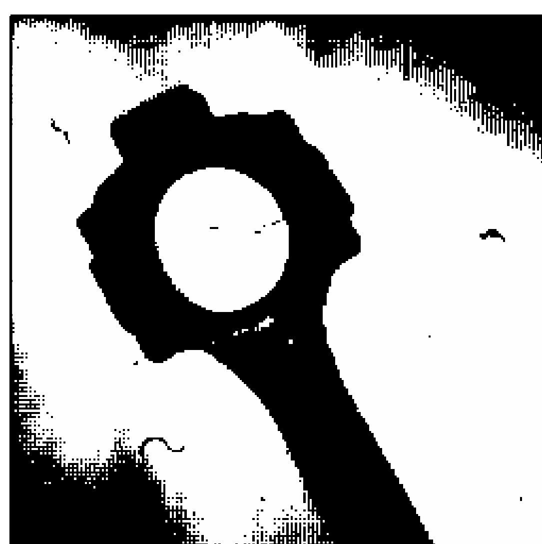
Thresholded Image



Threshold too low

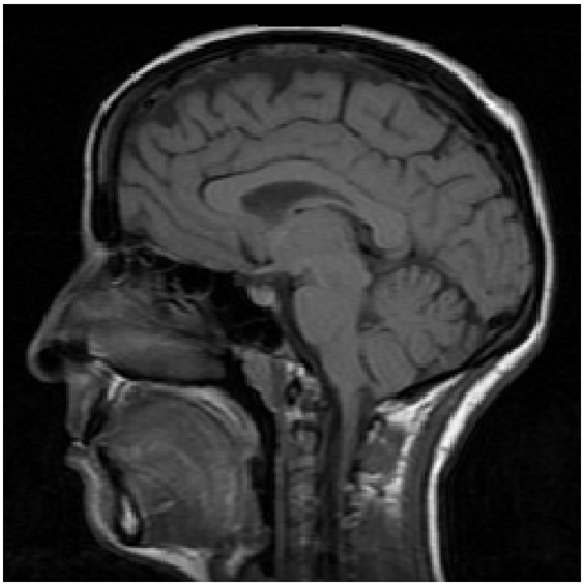


Threshold too high



FMRI - Example

Original Image

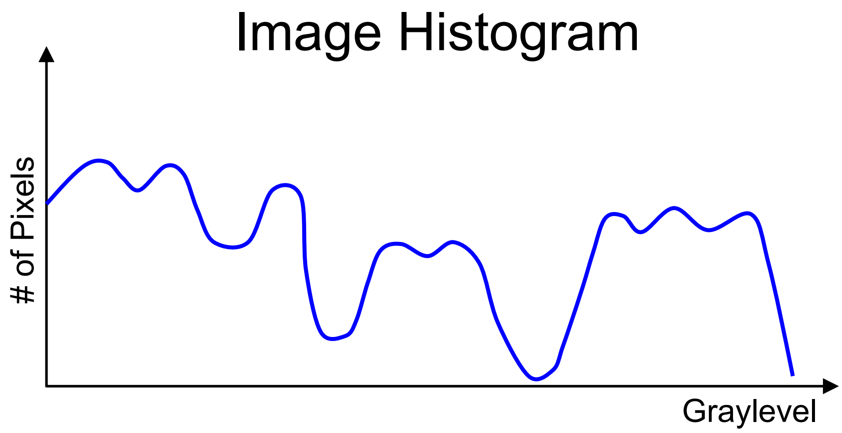


Threshold = 80

Threshold = 71

Threshold = 88

Simple thresholding is not always possible:



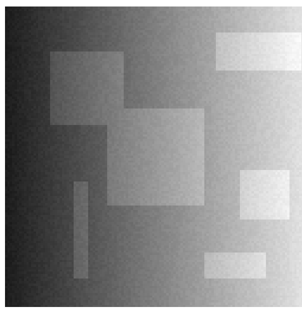
1) Many objects at different gray levels.

2) Variations in background gray level.

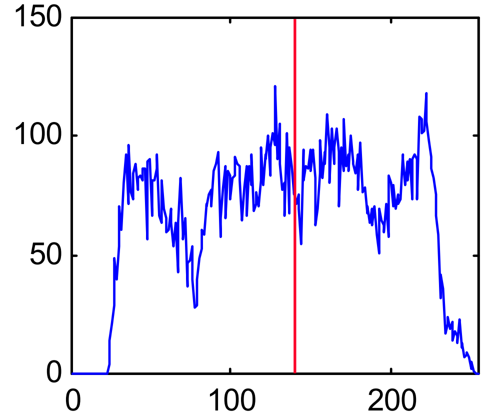
3) Noise in image.

Thresholding Example

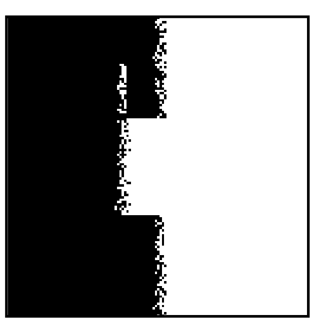
Original

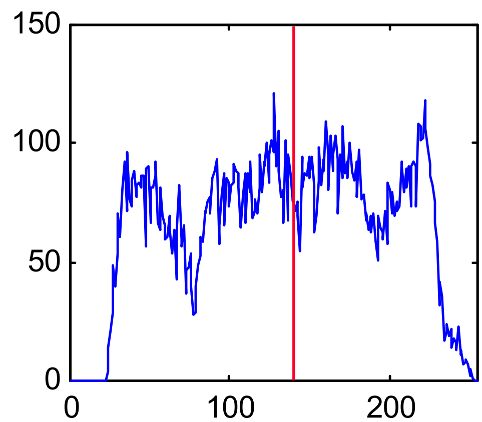


Histogram



Single Global Threshold

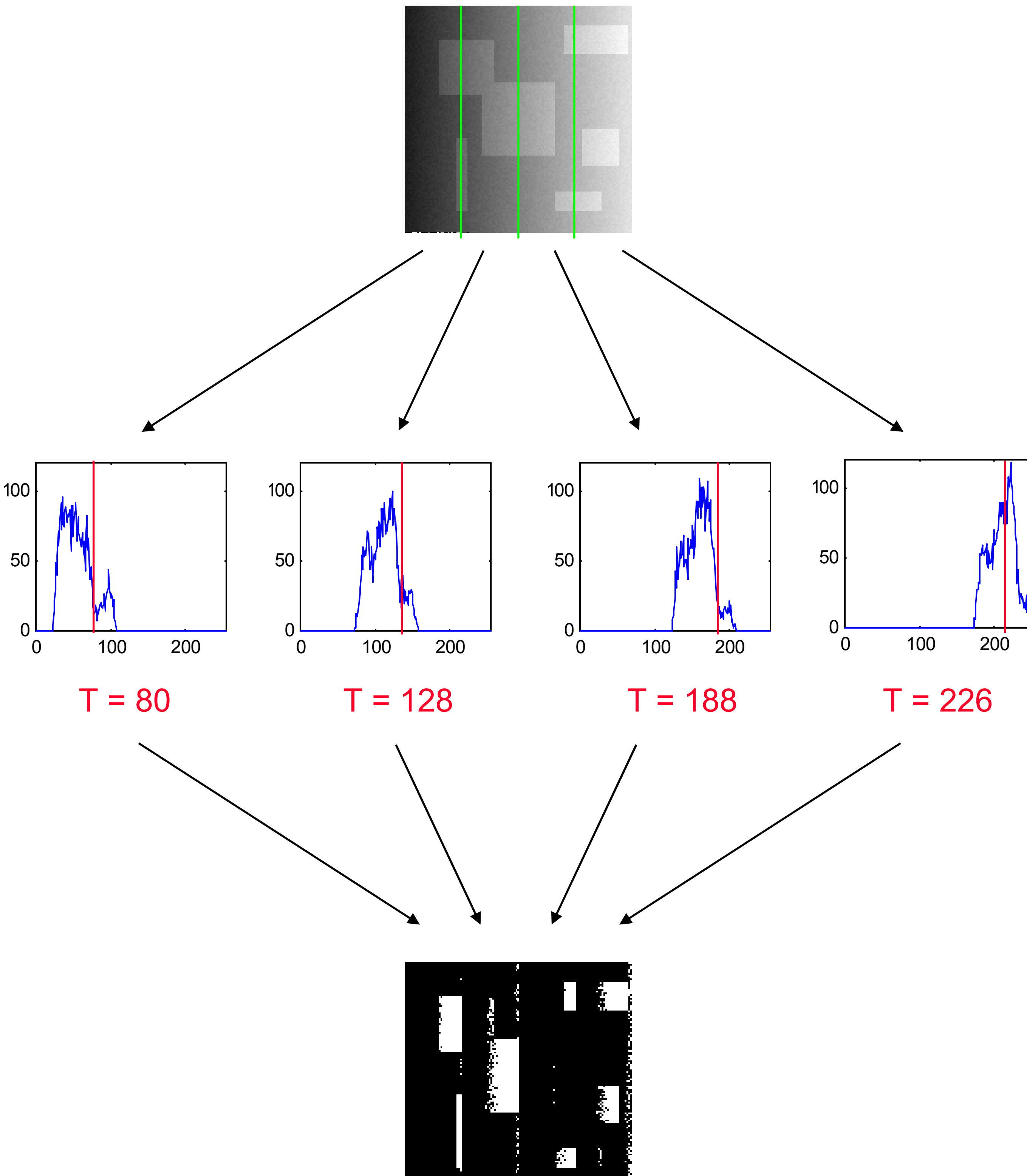




T = 128

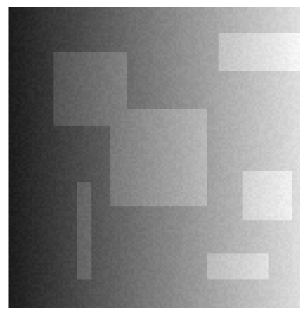
Local Thresholding - 4 Thresholds

Divide image in to regions. Perform thresholding independently in each region.

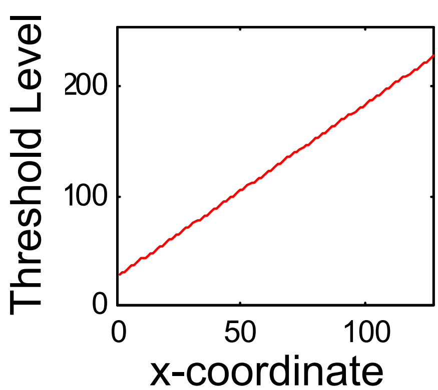


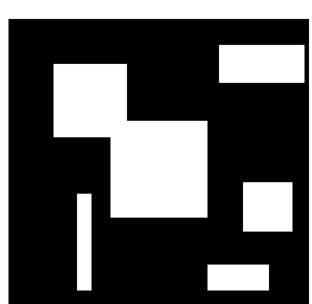
Adaptive Thresholding

Every pixel in image is thresholded according to the histogram of the pixel neighborhood.



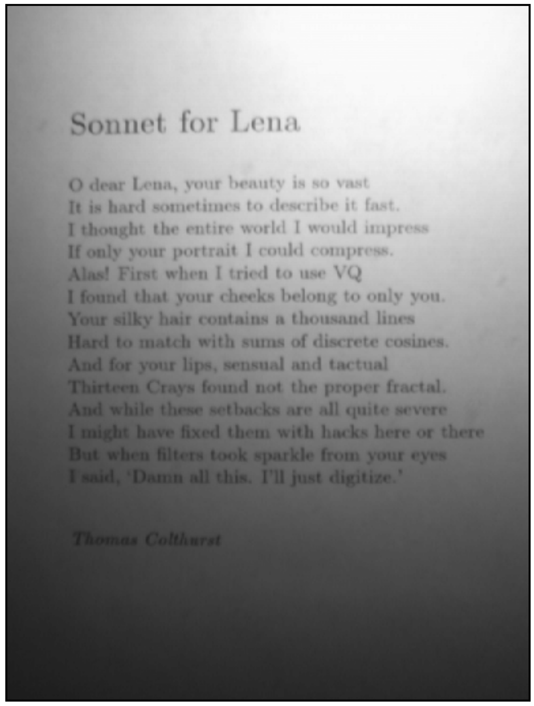
T =



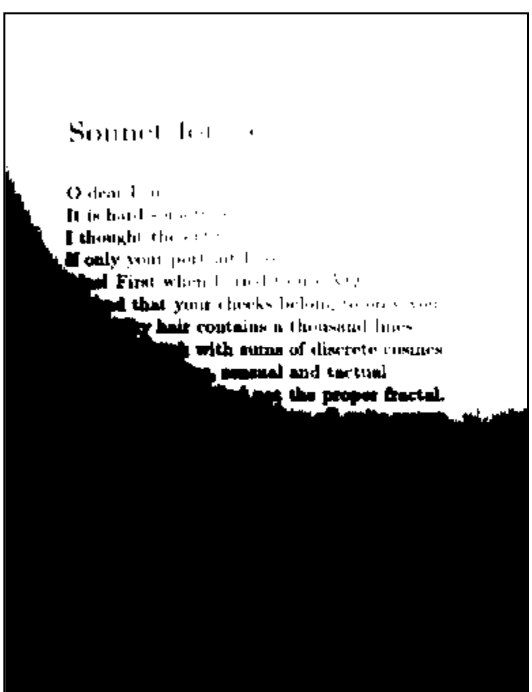


Adaptive Thresholding - Example

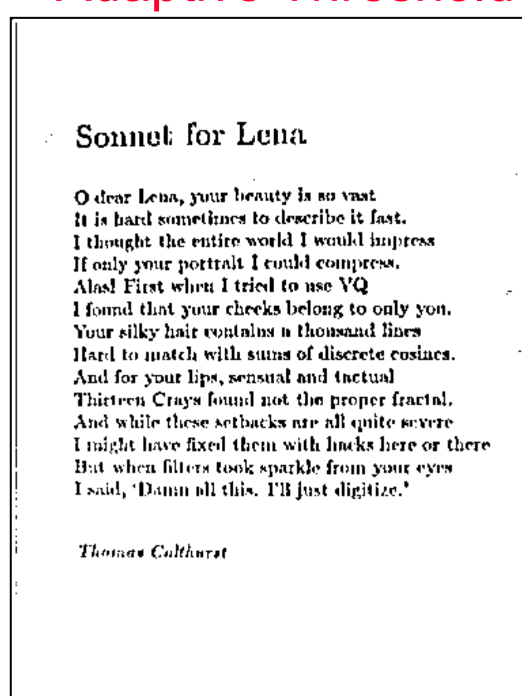
Original



Global Threshold



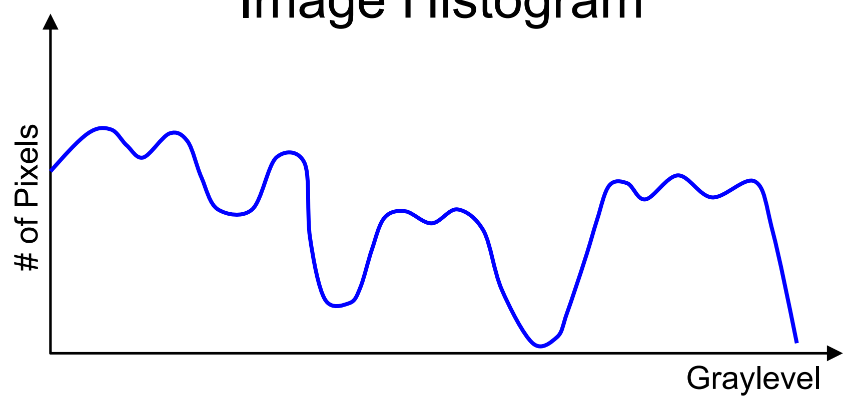
Adaptive Threshold



Threshold Segmentation of Noisy Images

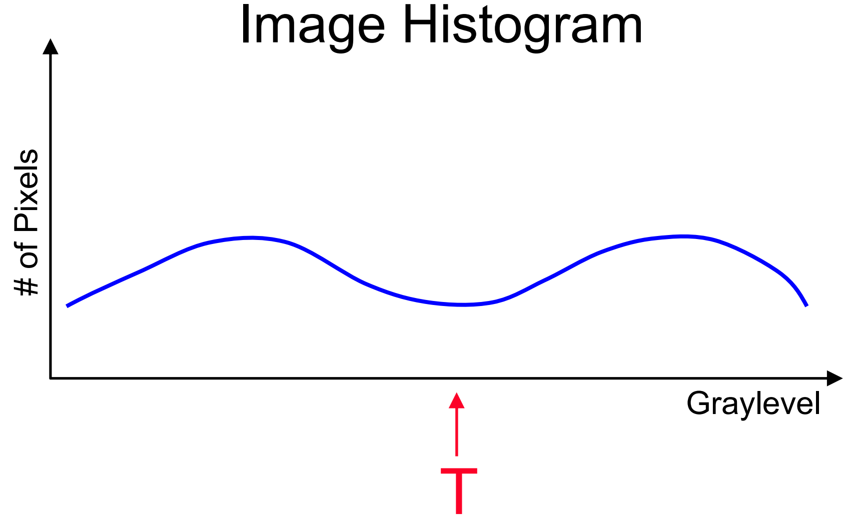
Noise inhibits localization of threshold.

Image Histogram

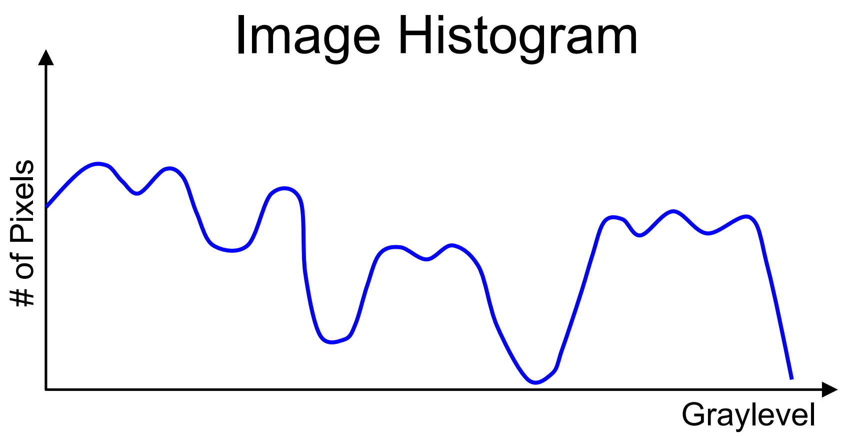


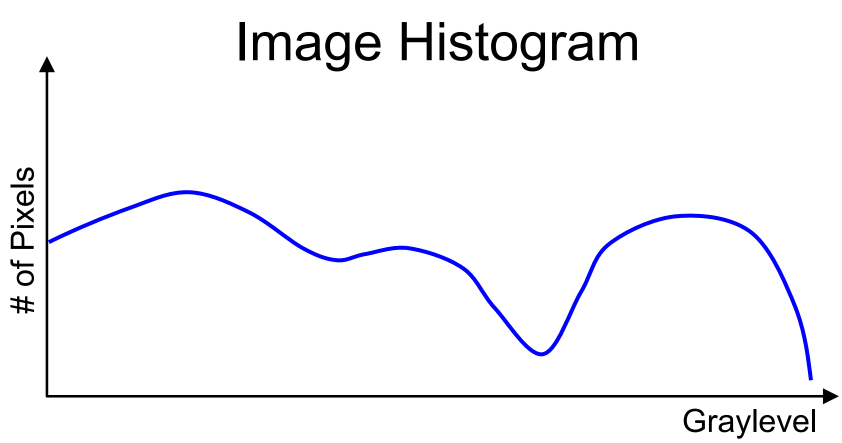
Smooth image and obtain a histogram for which threshold is easily determined.

Image Histogram

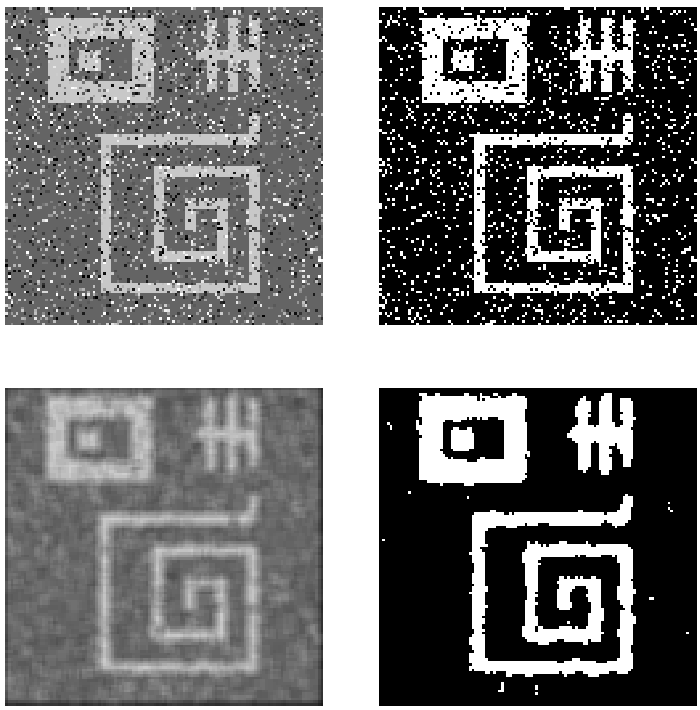


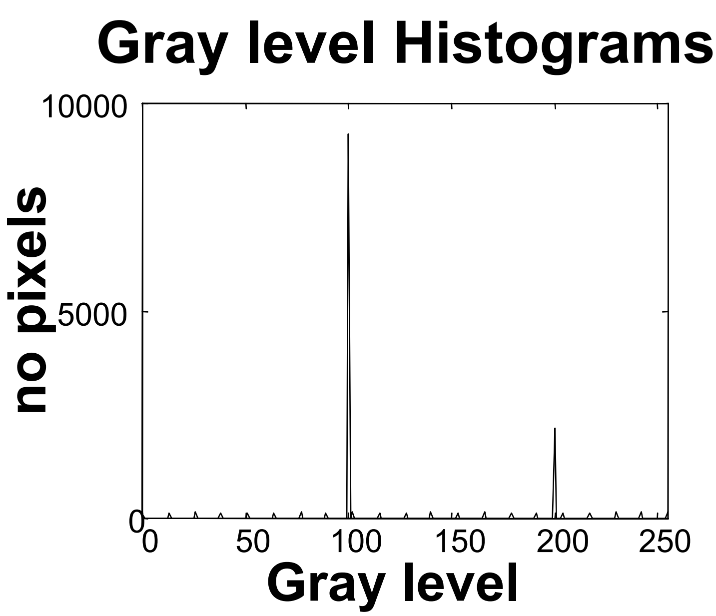
Note: Smooth the image, not the histogram...



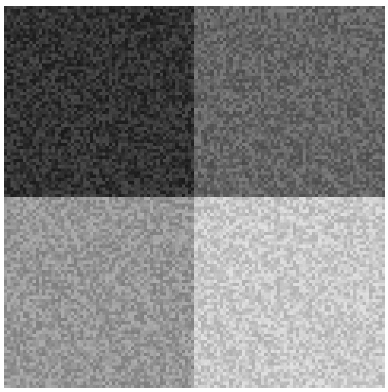


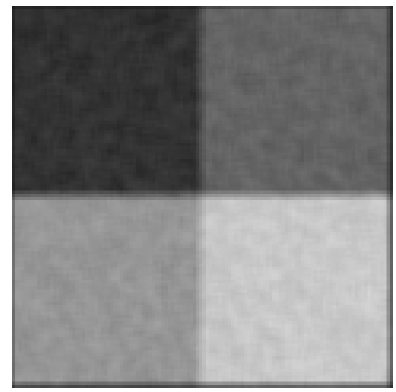
Threshold using Average

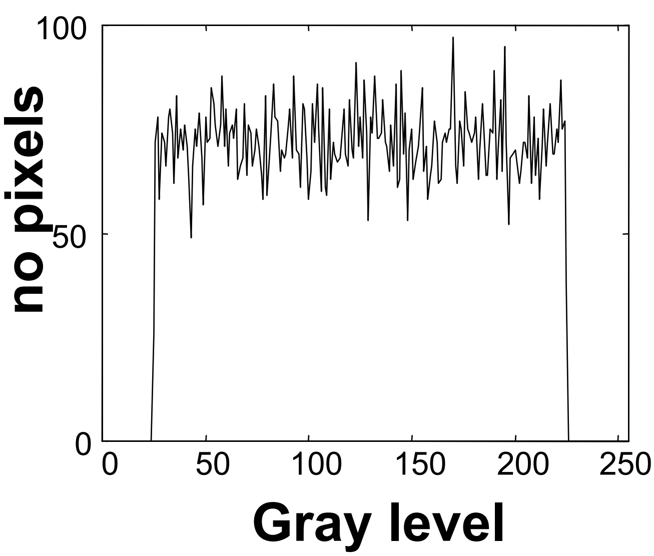


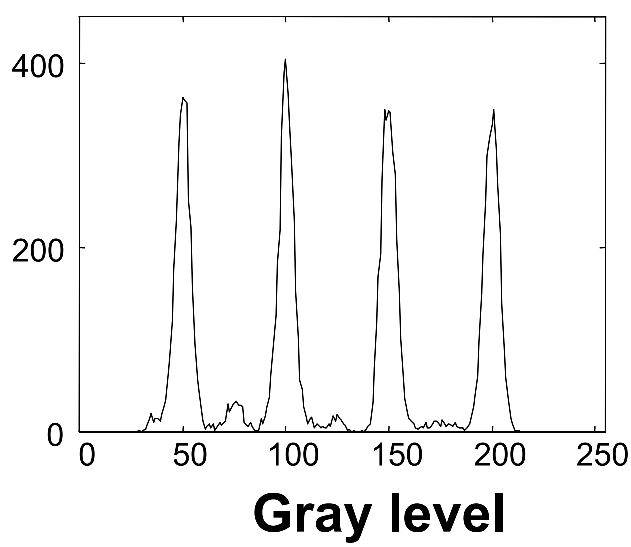


Threshold using Average



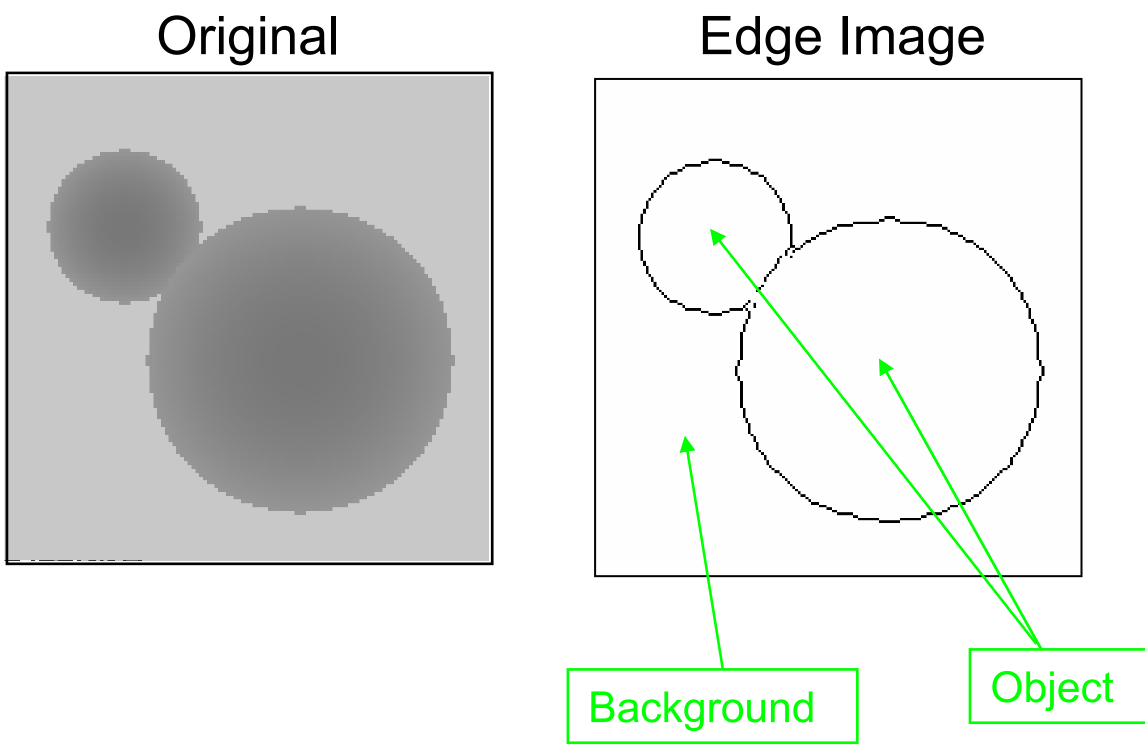




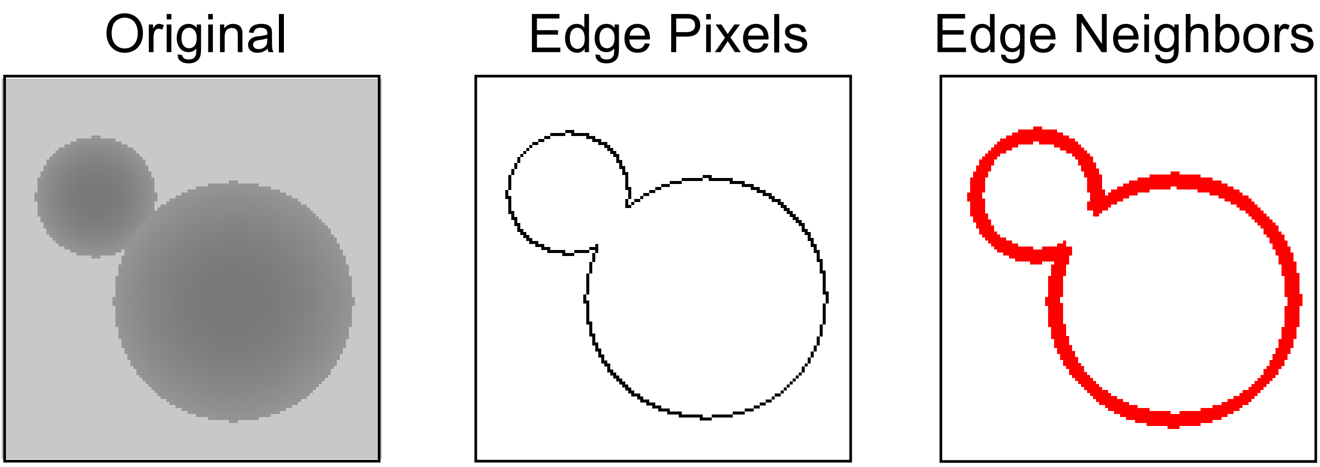


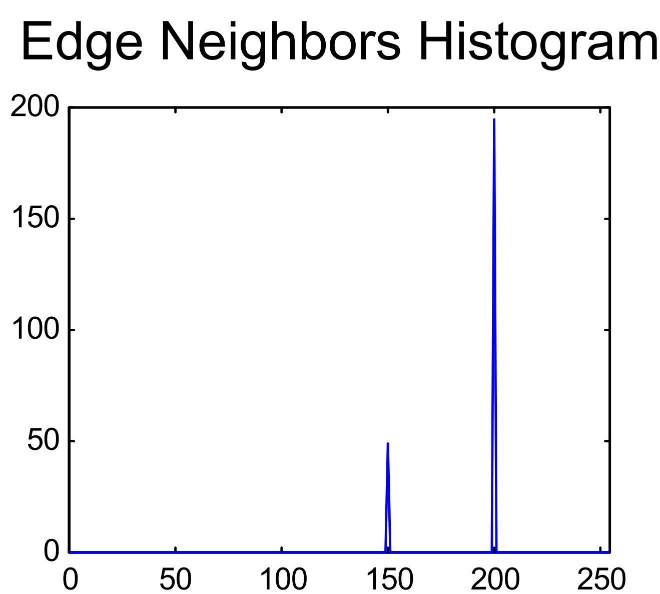
Gray level Histograms

Edge Based Segmentation



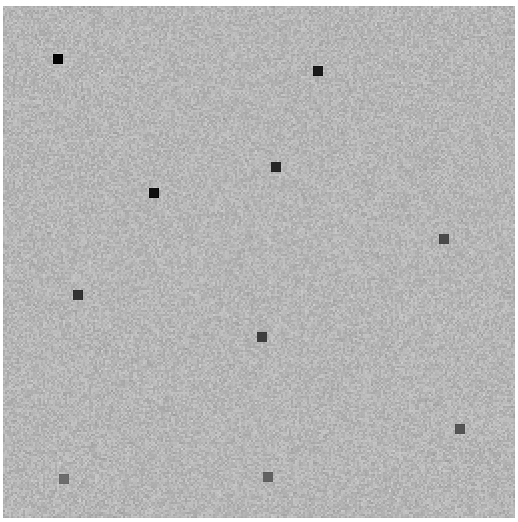
Edge Based Thresholding



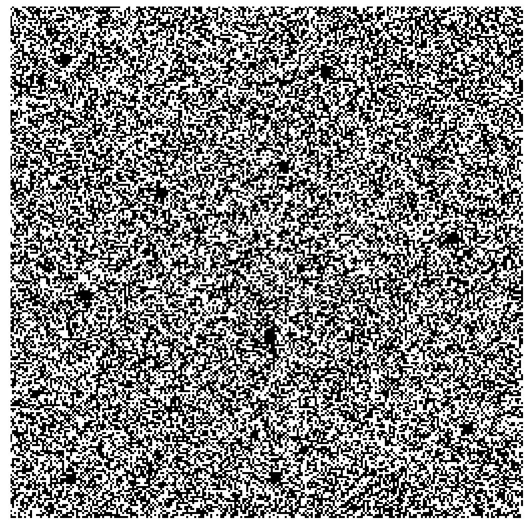


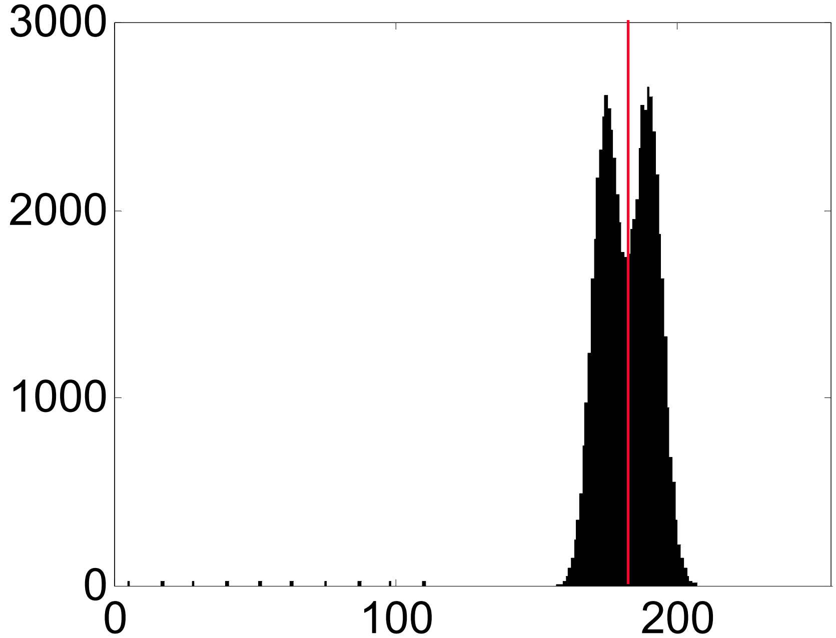
Thresholding Based on Boundary Characteristics

Original



Threshold (T=182)I

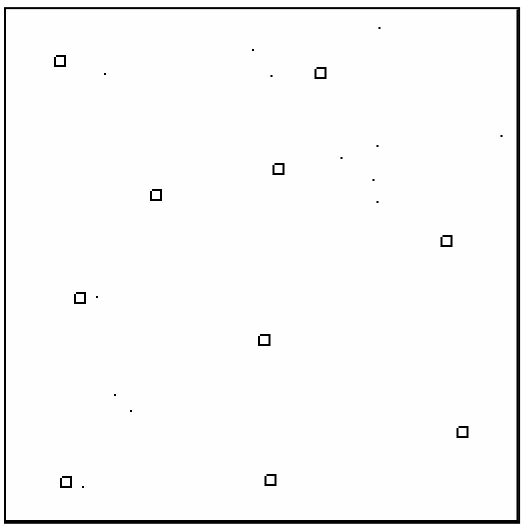




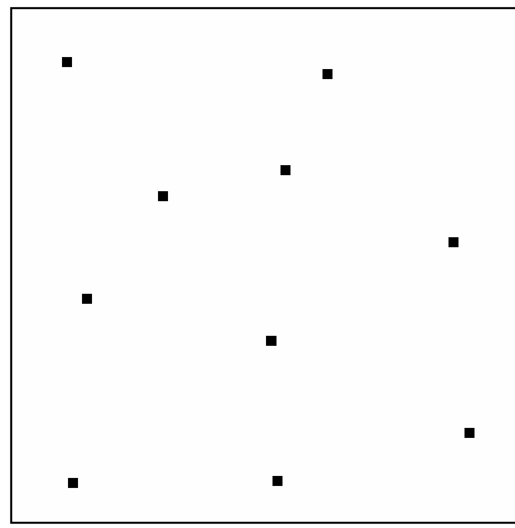
Global Histogram

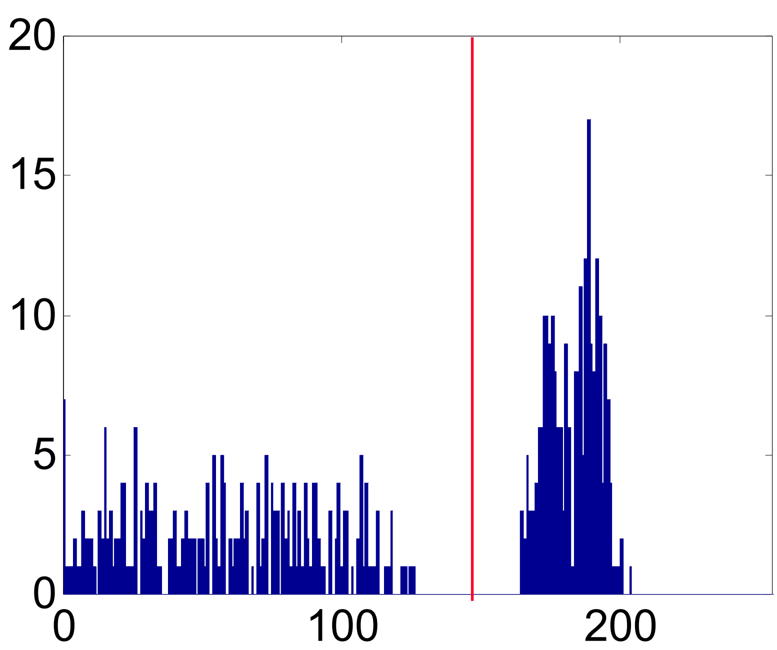
Thresholding Based on Boundary Characteristics

Original



Threshold (T=143)I





Edge Neighborhood Histogram

Region Growing

Define:

S = the set of pixels inside the region.

Q = queue of pixels to be checked.

(x_0,y_0) = a pixel inside the region.

Algorithm:

Initialize: $S = \emptyset$

$Q = \{ (x_0,y_0) \}$

1) Extract pixel P from queue Q

2) Add P to S.

3) For each neighbor P' of P:

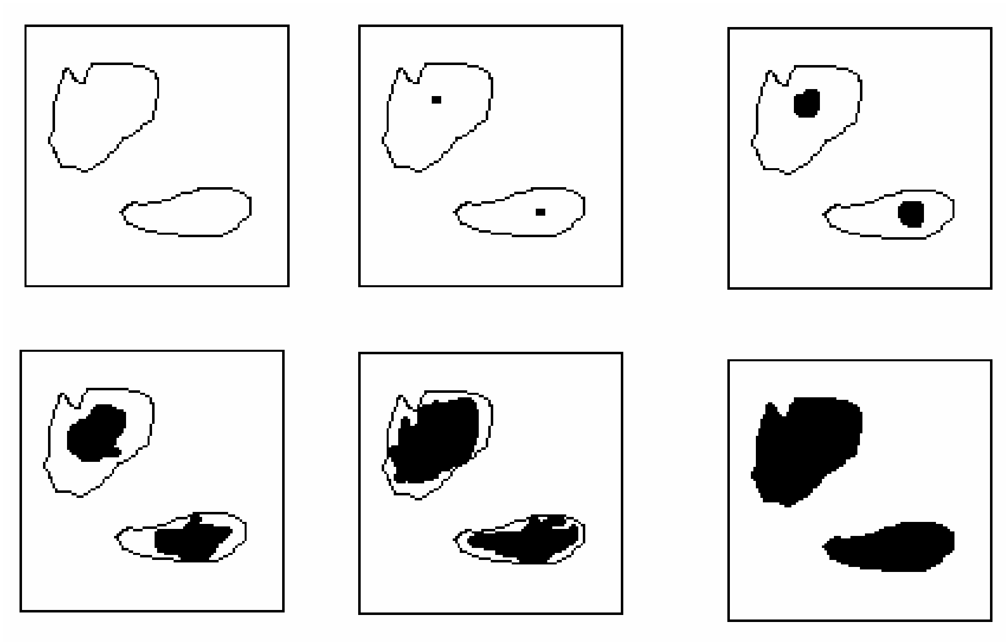
if P' is "similar" to P and $P' \notin S$ then

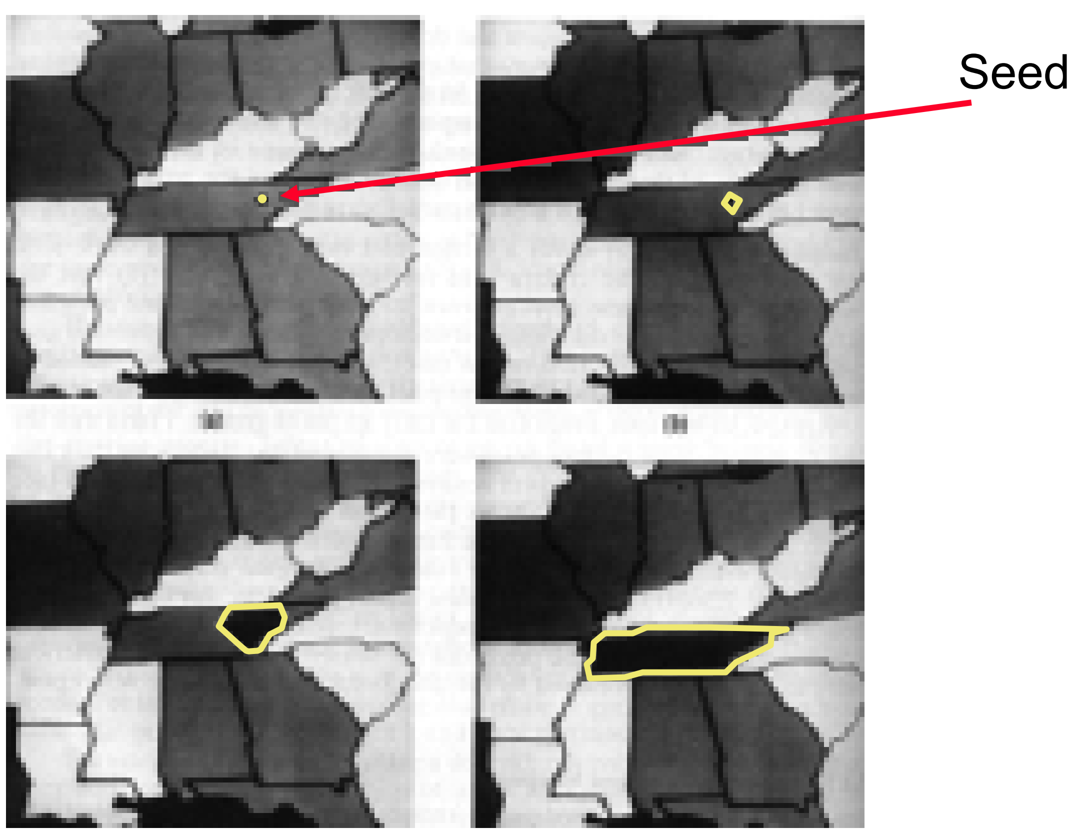
add P' to Q.

4) If $Q = \emptyset$ then end, else return to 1.

S = the extracted pixels of the region.
Define what "similar" means.
Problematic in small gradient regions.

Region Growing - Example

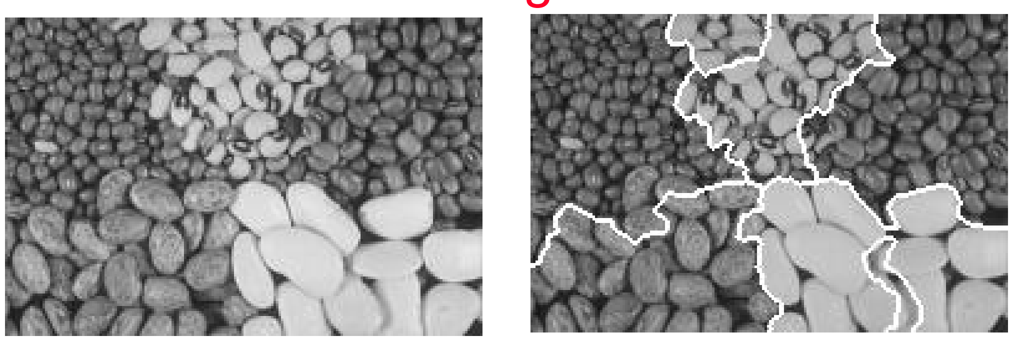




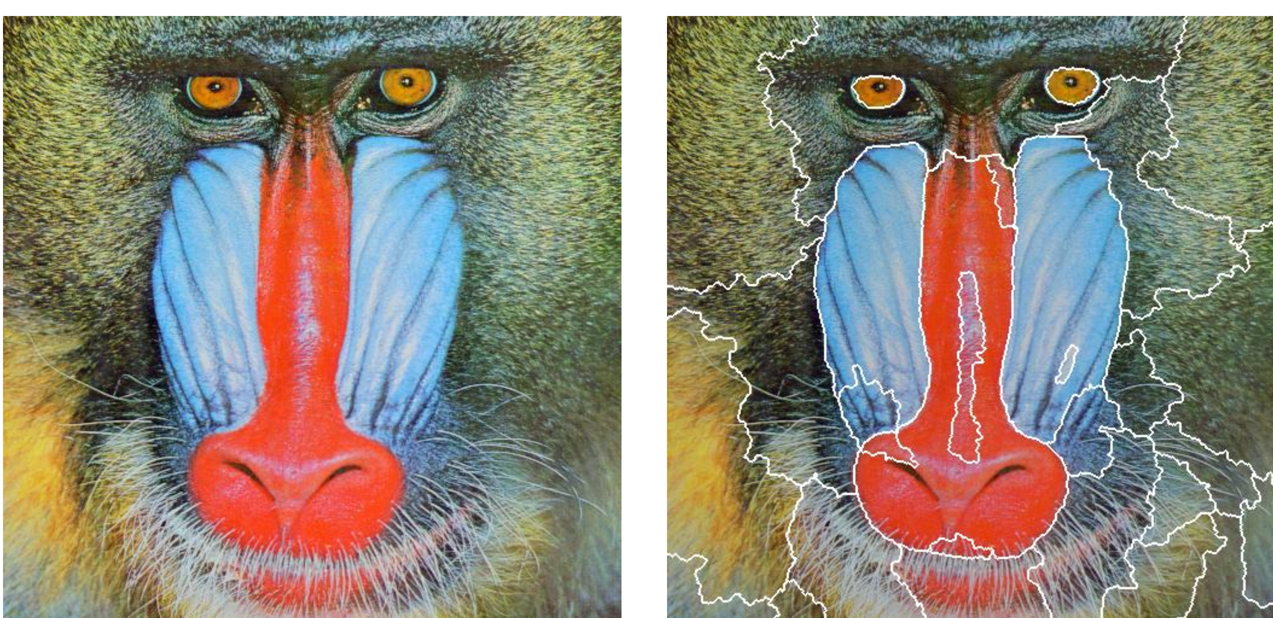
Region Growing - Examples

Color Segmentation

Texture Segmentation

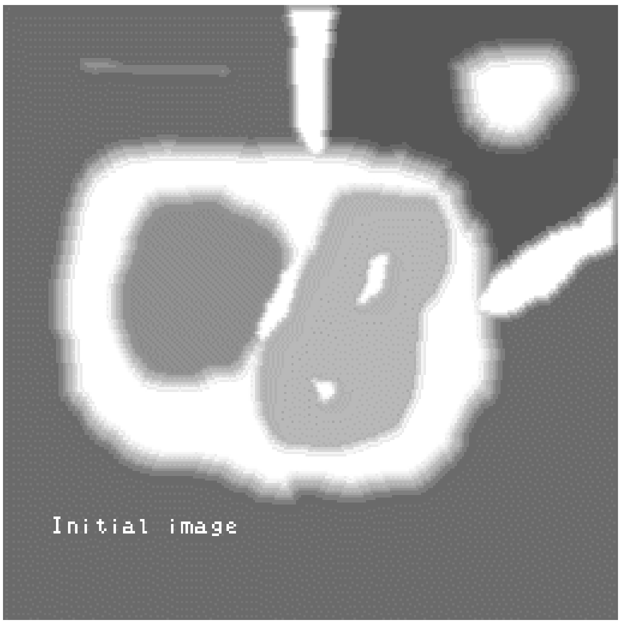


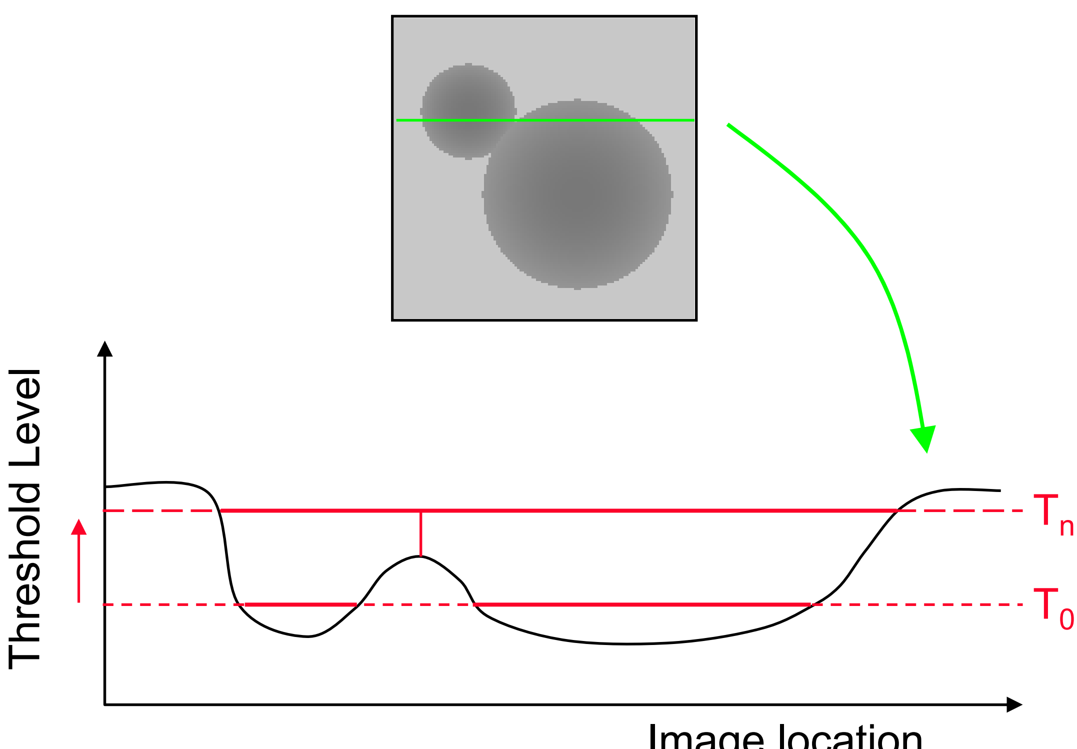
Color + Texture Segmentation



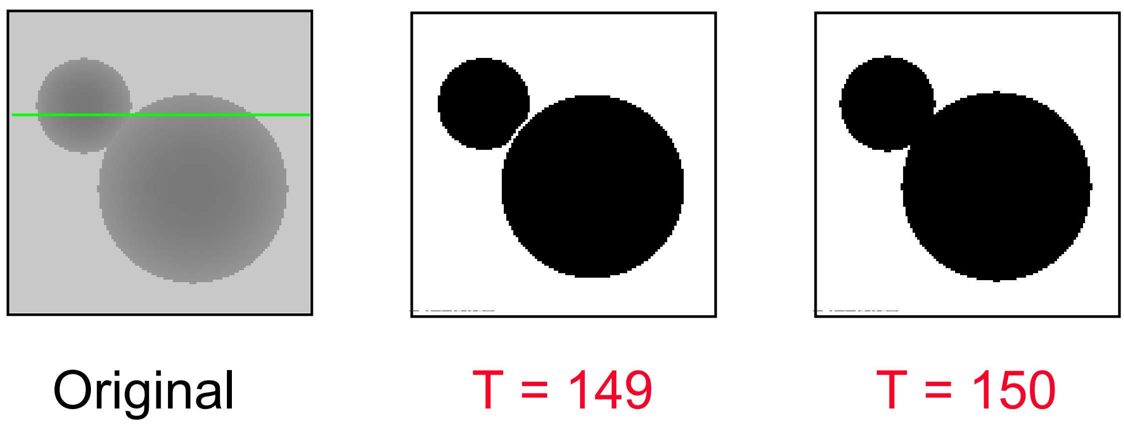
Watershed Threshold Algorithm

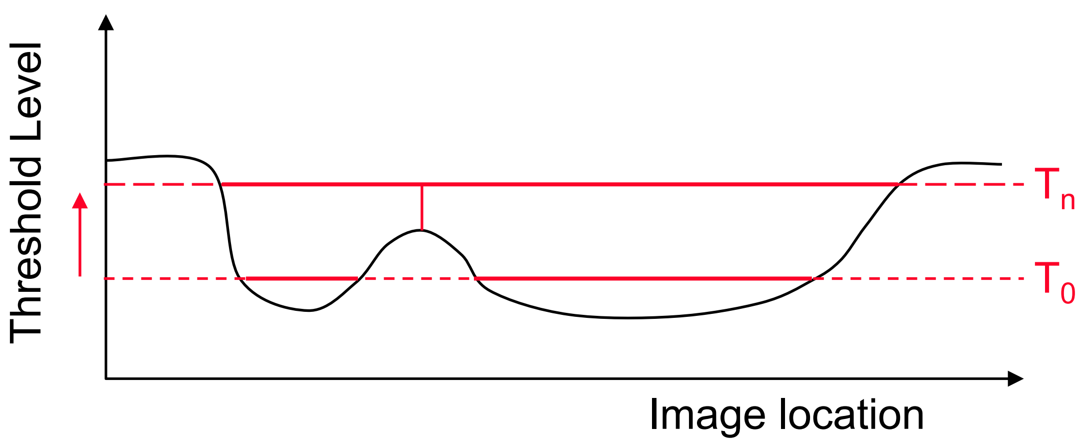
An Image can be viewed as a topographic map





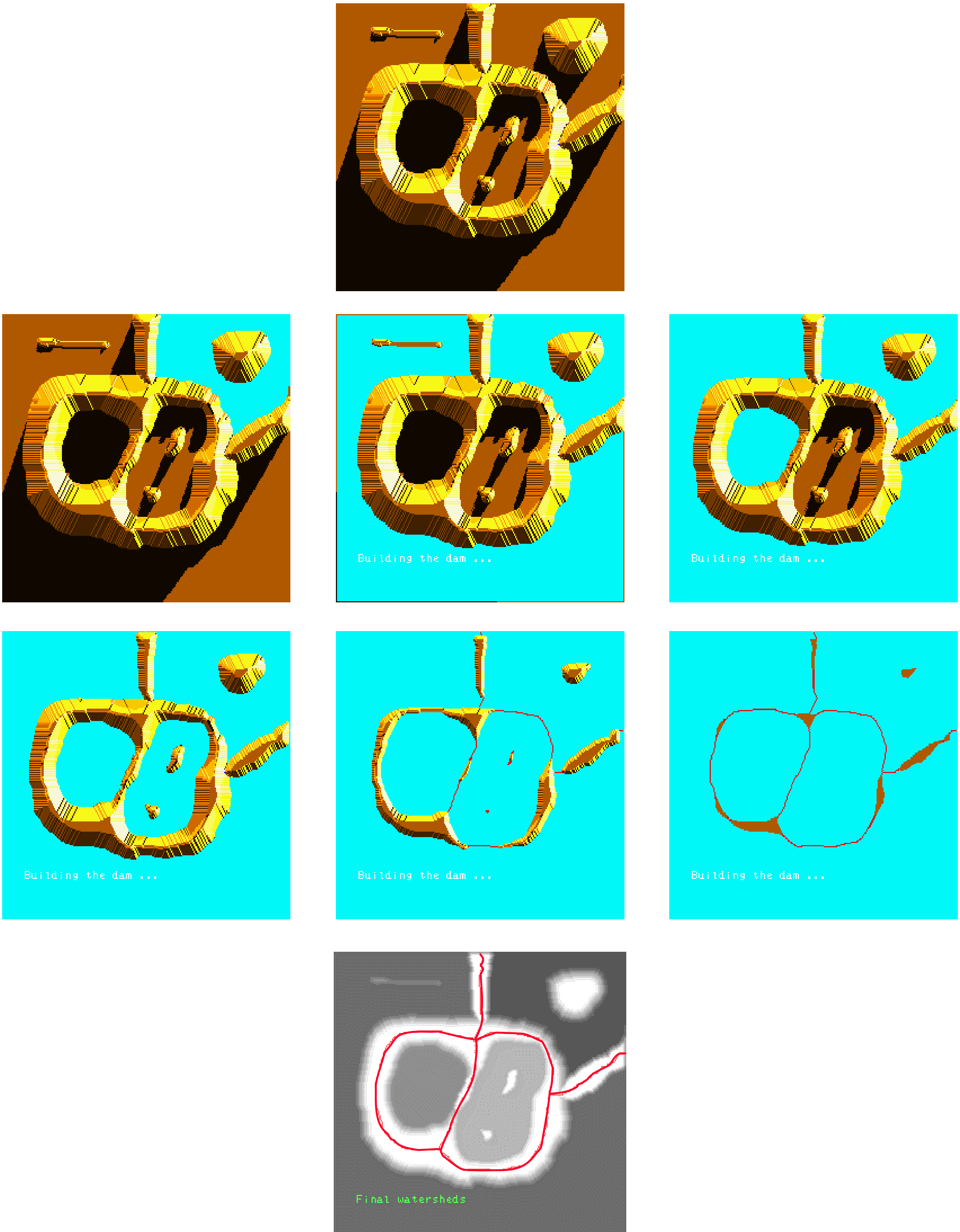
Watershed Threshold Algorithm





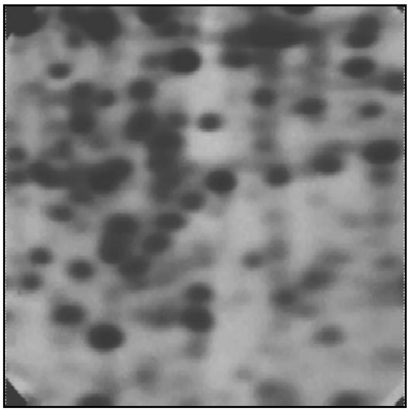
Initialize threshold at T_0 that separates objects well.
Determine connected components.
Raise threshold, and detect pixels that pass threshold and belong to more than one connected component.
Do not let objects merge.
Set these pixels as object boundaries.

Watershed Threshold Algorithm

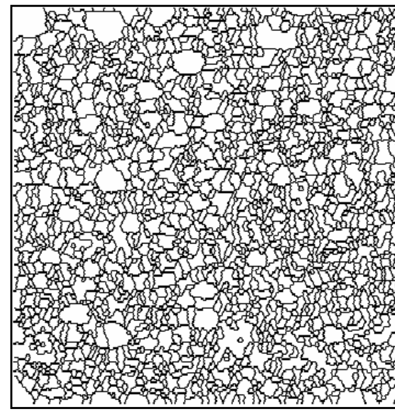


Watershed Threshold Algorithm

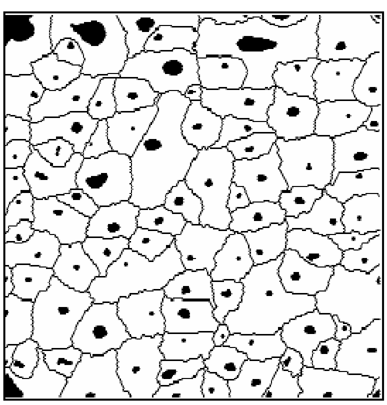
Original



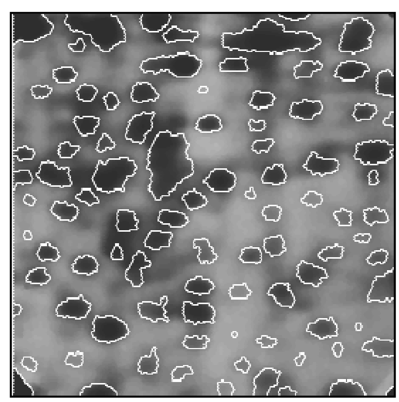
Watershed Boundaries



Watershed Markers



Watershed Boundaries



Watershed Markers may be chosen manually or local global maximas (as above)

Split & Merge Segmentation

2 Stage Algorithm:

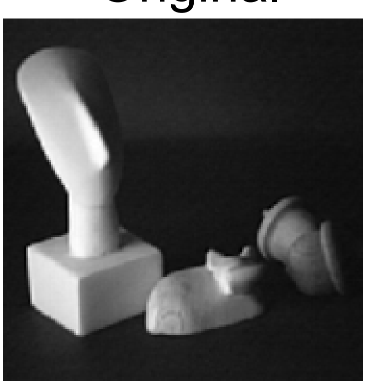
Stage 1: Split

Split image into regions using a Quad Tree representation.

Stage 2: Merge

Merge "leaves" of the Quad Tree which are neighboring and "similar".

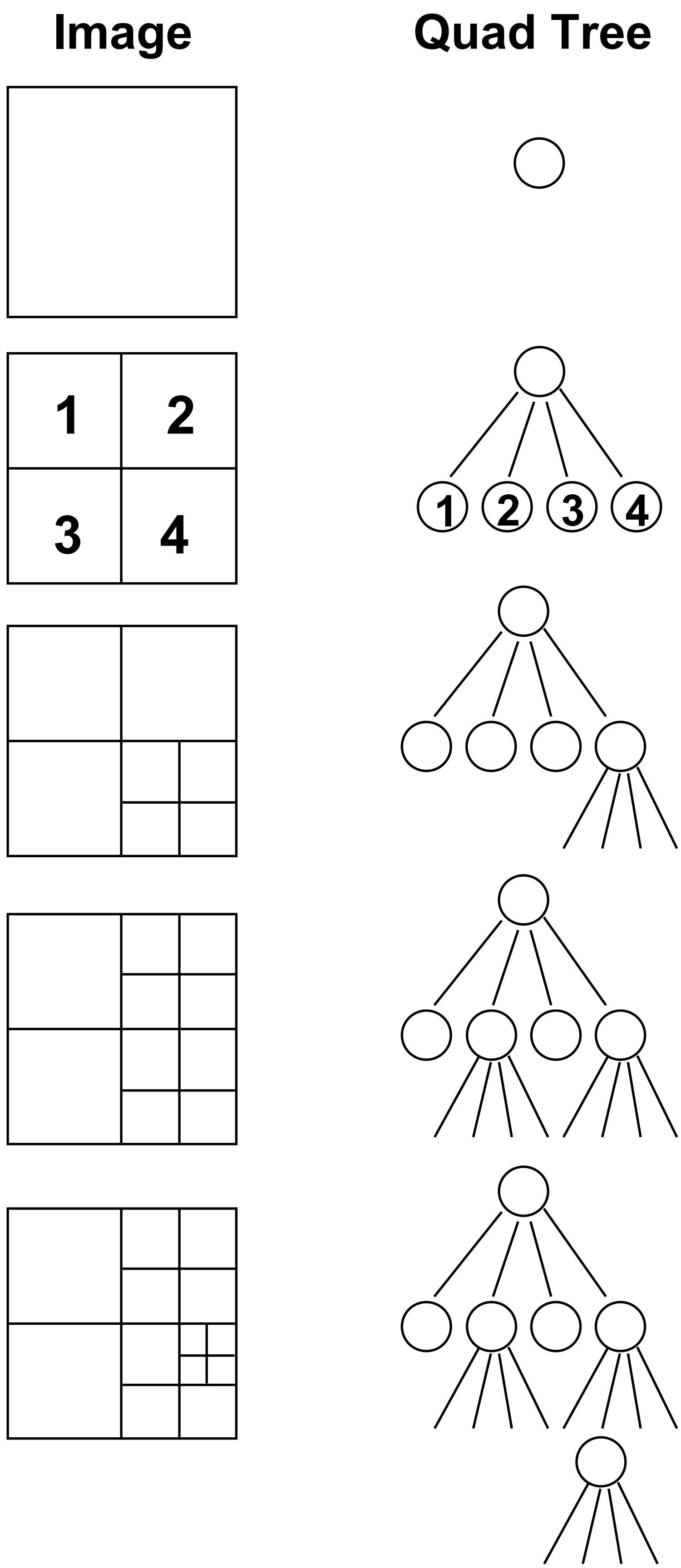
Original



Split

Split + Merge

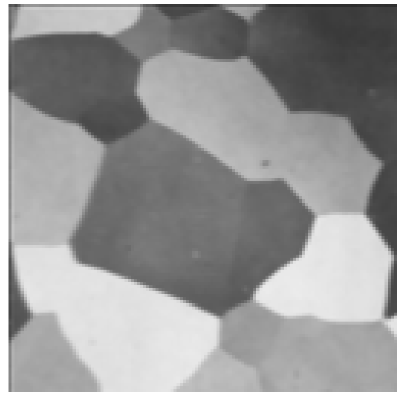
Quad Tree - Representation



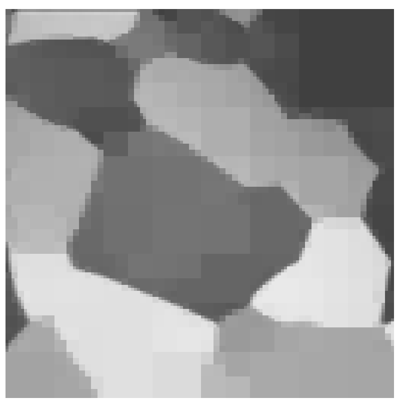
Demo

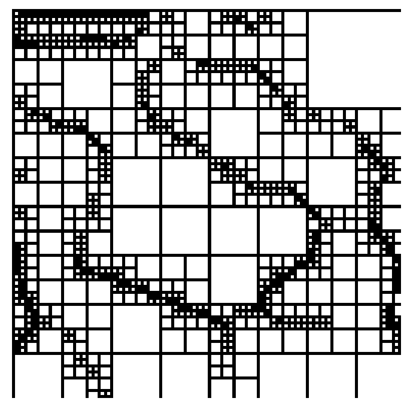
Quad Tree Representation

Original

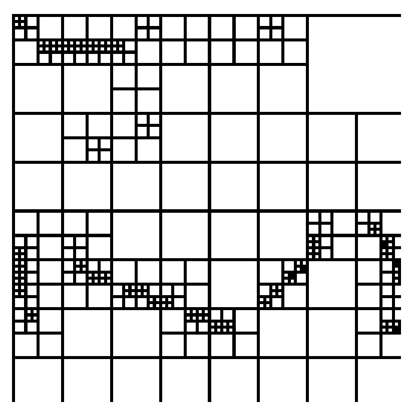


Thresh = 0.20

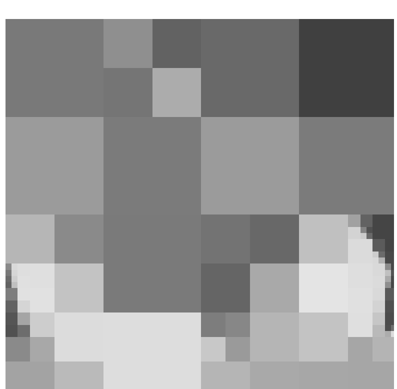


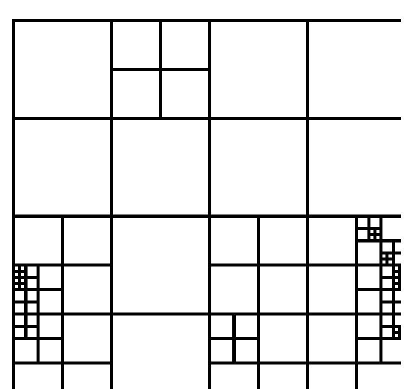


Thresh = 0.40



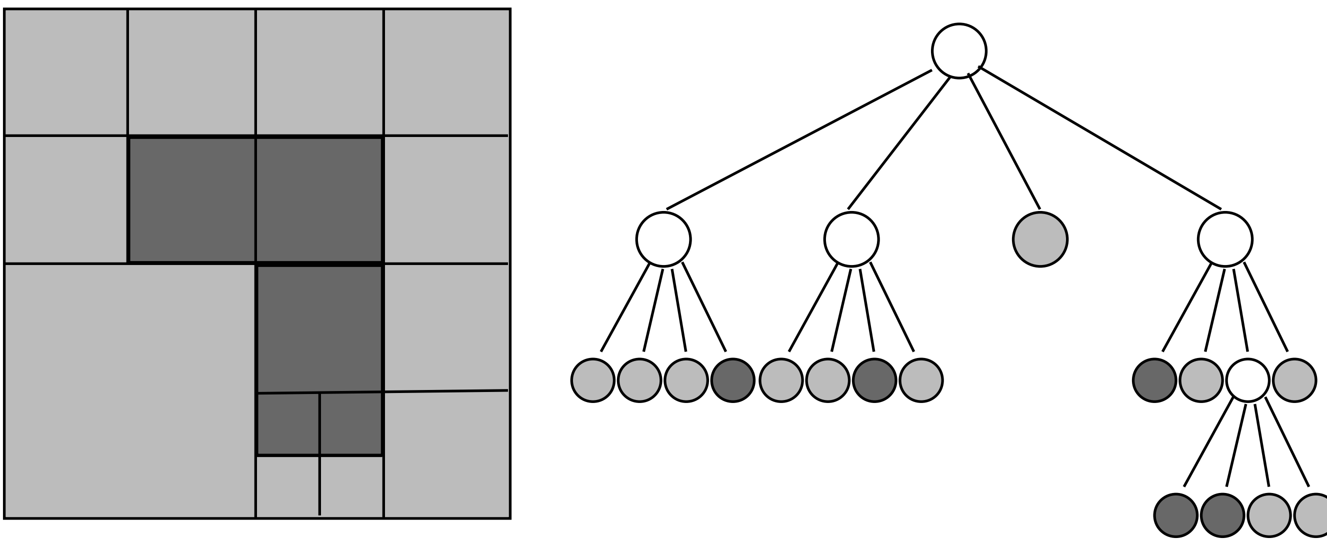
Thresh = 0.55



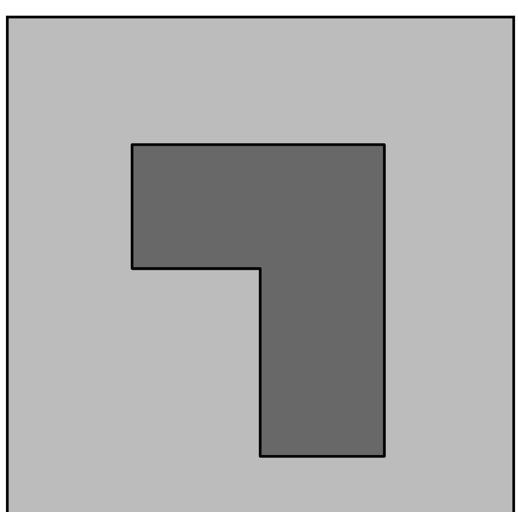


Split & Merge Example

Stage 1: Split

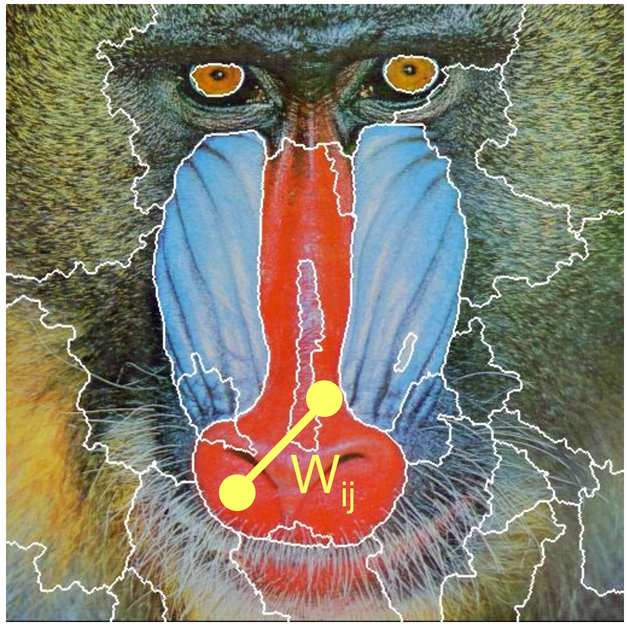


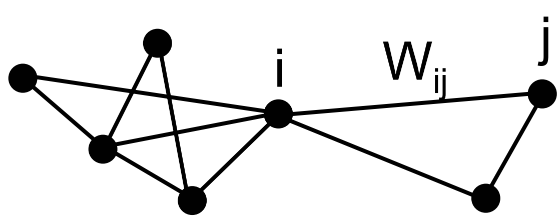
Stage 2: Merge



Split & Merge Example

Graph-Cut Segmentation





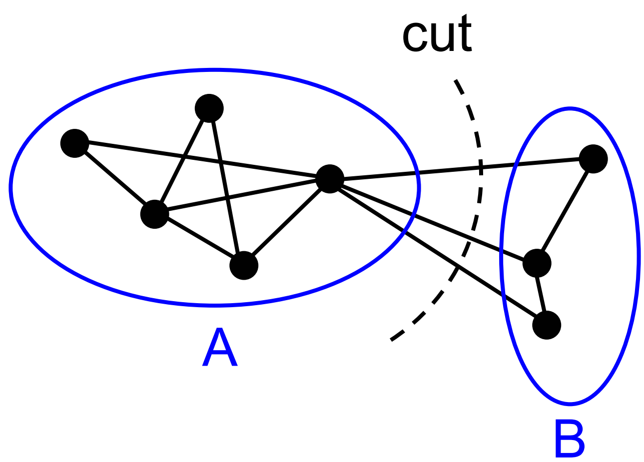
G = { V , E }

V = vertices
E = Edges

V = image pixels
E = pixel similarity

Segmentation = Graph Partitioning

Min-Cut Segmentation



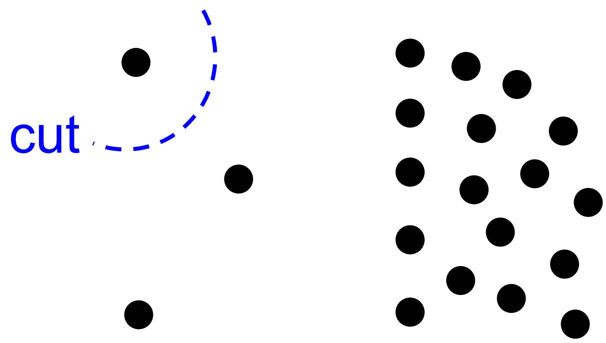
$$\text{cut}(A,B) = \sum_{i \in A, j \in B} W_{ij}$$

Segmentation by min-cut:

Find A,B such that cut(A,B) is *minimal*.

(Wu and Leahy 1993)

Normalized-Cut Segmentation



Min-cut segmentation favors small segments.

Segmentation by normalized-cut:

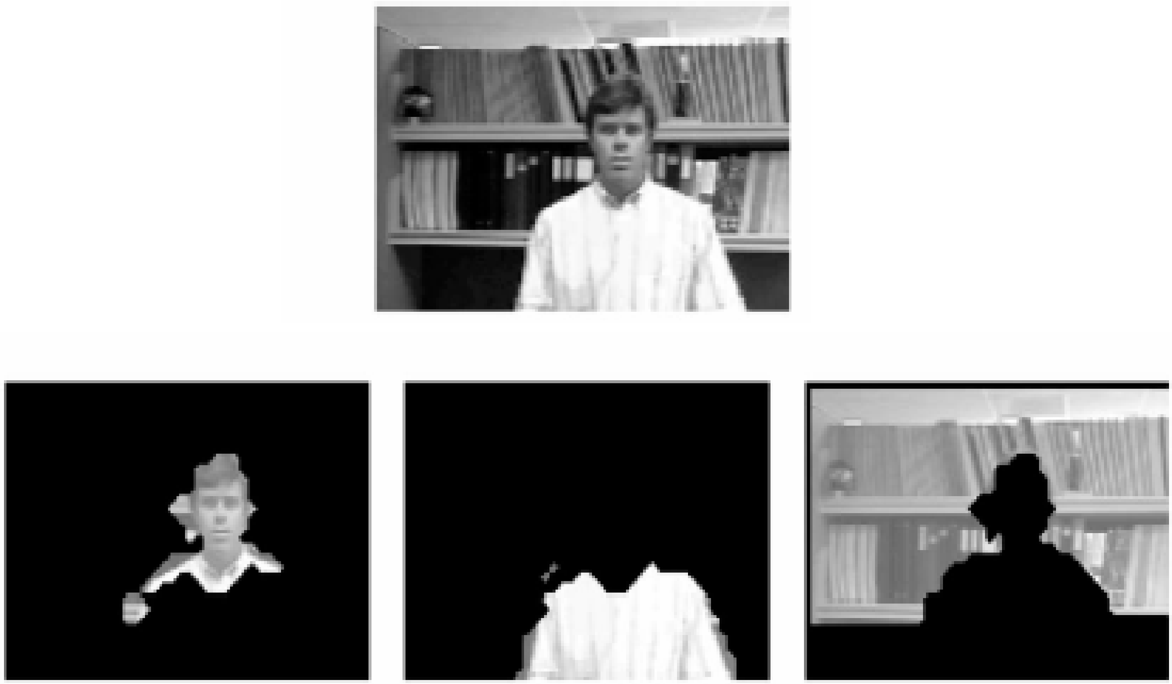
Find A,B such that Ncut(A,B) is *minimal*.

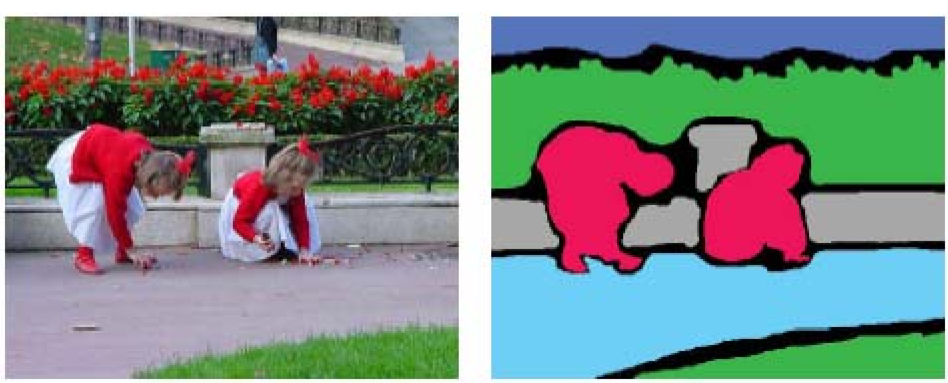
$$\mathrm{Ncut}(A,B) = \sum_{i \in A, j \in B} W_{ij} \left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)} \right)$$

where $$\mathrm{vol}(A) = \sum_{i \in A, j \in A} W_{ij}$$

(Shi and Malik 2000)

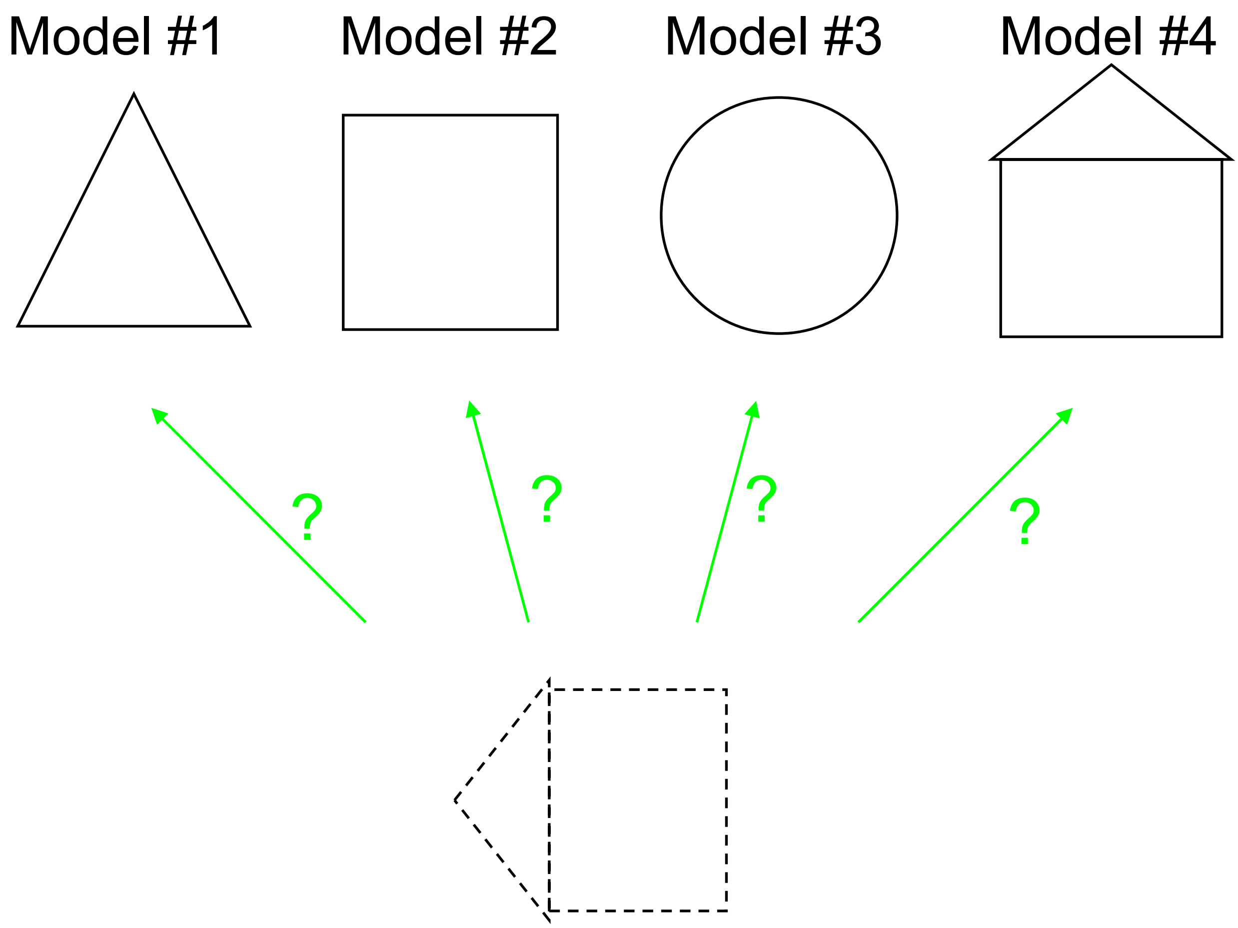
Normalized-Cut Segmentation - Examples





(from Cohen-Or 2005)

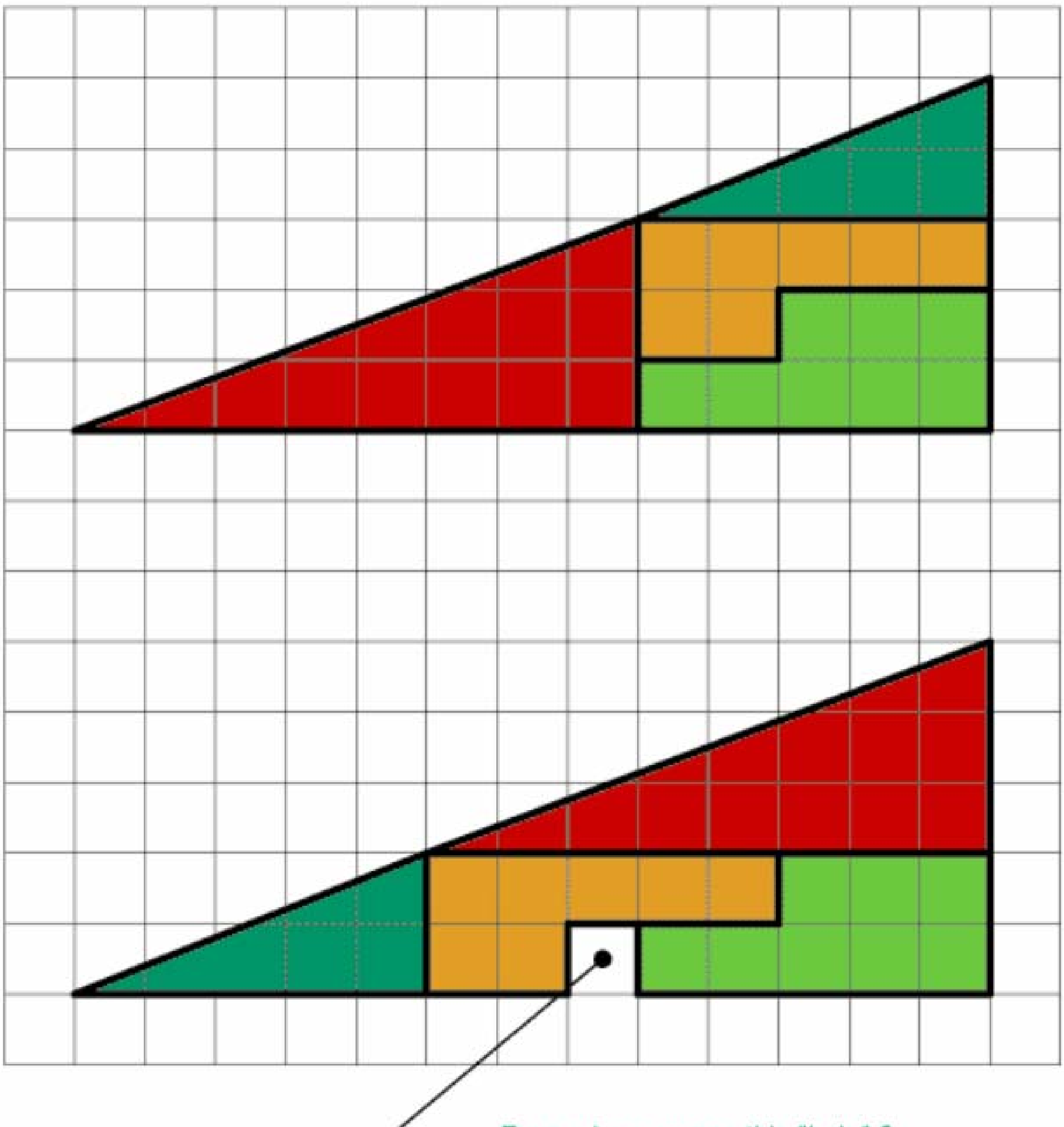
Shape Matching / Object Recognition



Which Model matches the Measurement?

- Which Model
- What is the transformation from Model to Measurement (translation, rotation, scale,…)

HOW CAN THIS BE TRUE ?



Below the four parts are moved around

The partitions are exactly the same, as those used above

From where comes this "hole" ?