Computer Vision – TP10
Introduction to Segmentation

Miguel Coimbra, Hélder Oliveira

U. PORTO FC

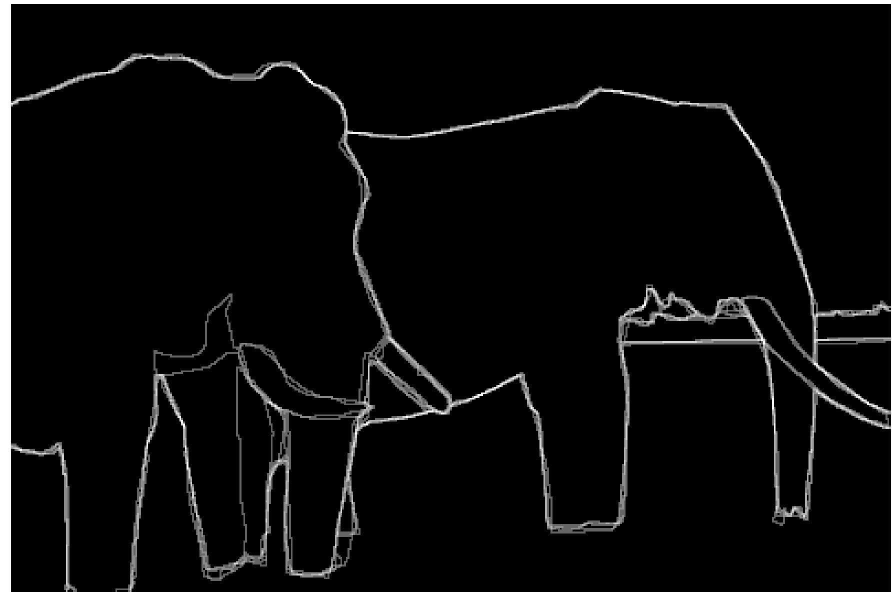
Outline

- Introduction to segmentation
- Thresholding
- Region based segmentation
- Segmentation by clustering

Topic: Introduction to segmentation

- Introduction to segmentation
- Thresholding
- Region based segmentation
- Segmentation by clustering

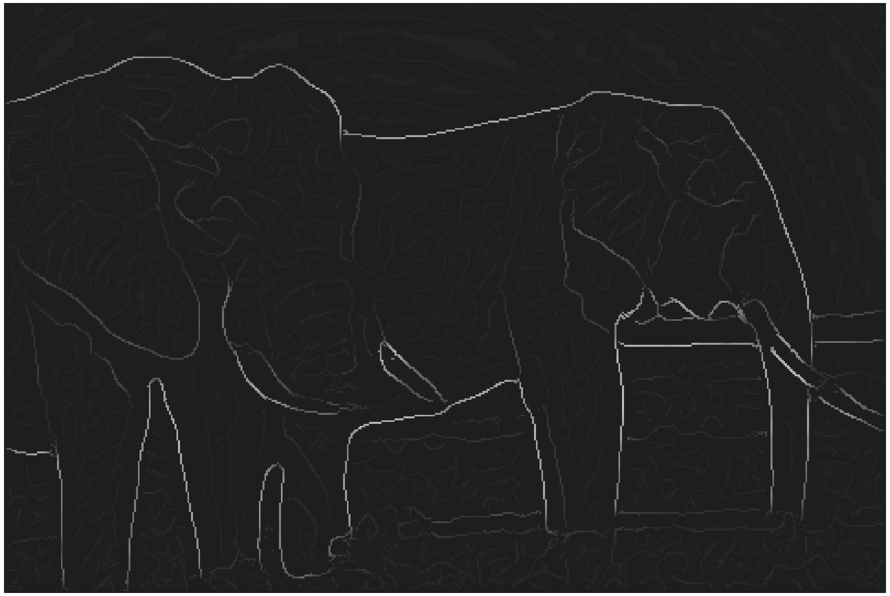
Boundaries of Objects



Marked by many users

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/bench/html/images.html

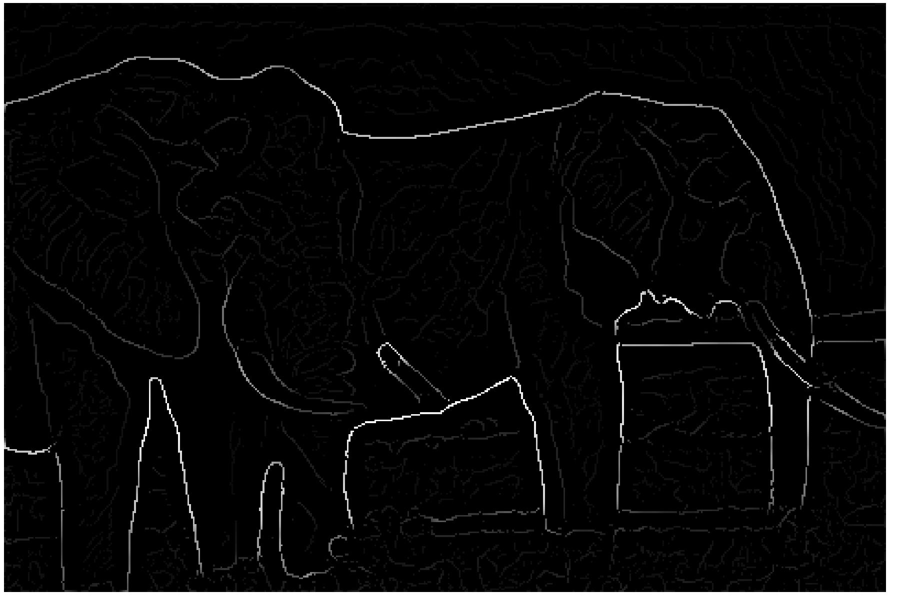
Boundaries of Objects from Edges



Brightness Gradient (Edge detection)

- Missing edge continuity, many spurious edges

Boundaries of Objects from Edges

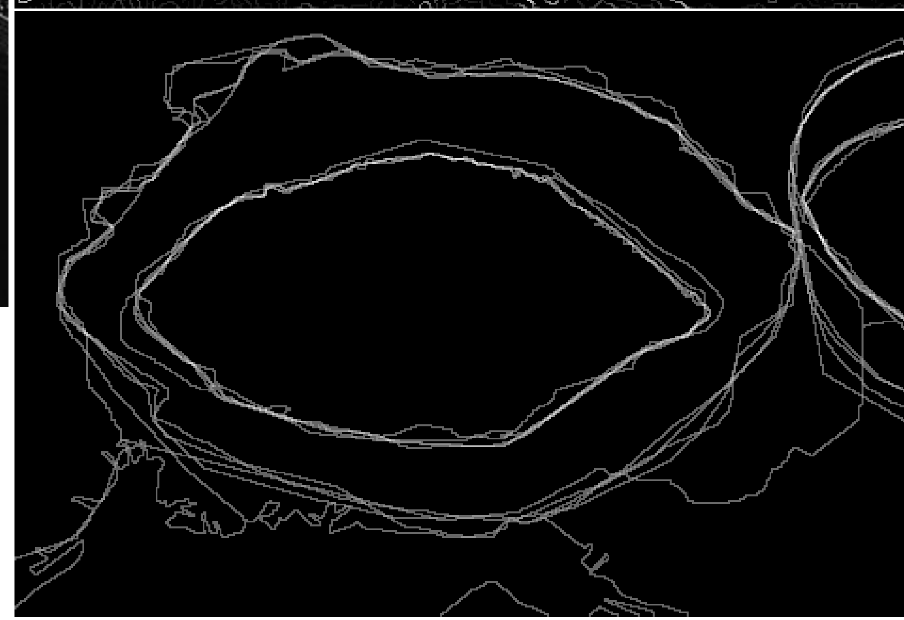


Multi-scale Brightness Gradient

- But, low strength edges may be very important

Machine Edge Detection

Image



Human Boundary Marking

Boundaries in Medical Imaging

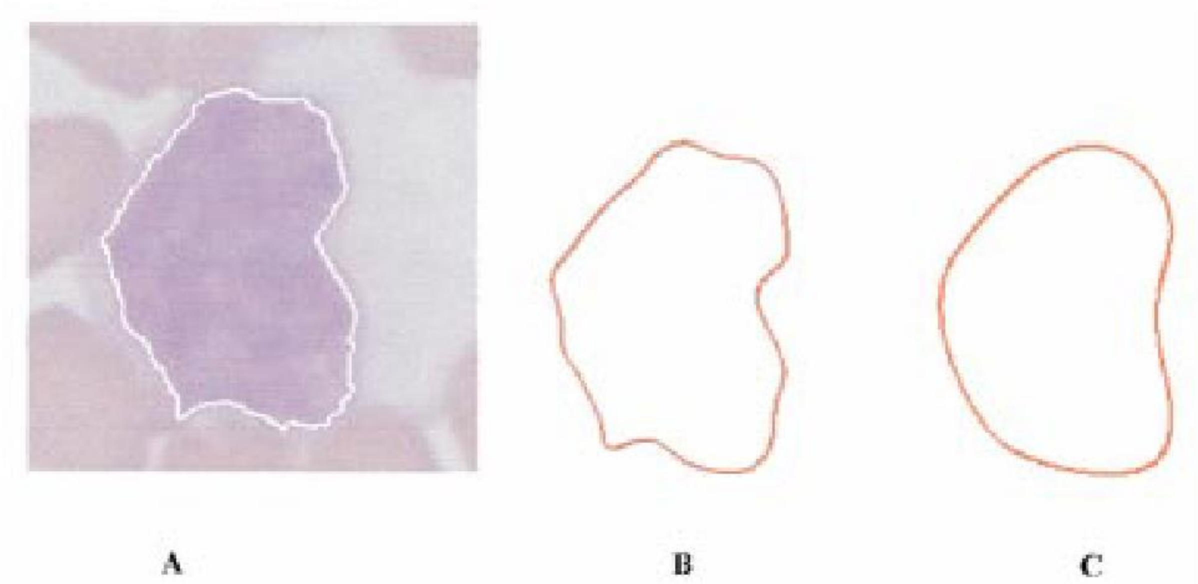
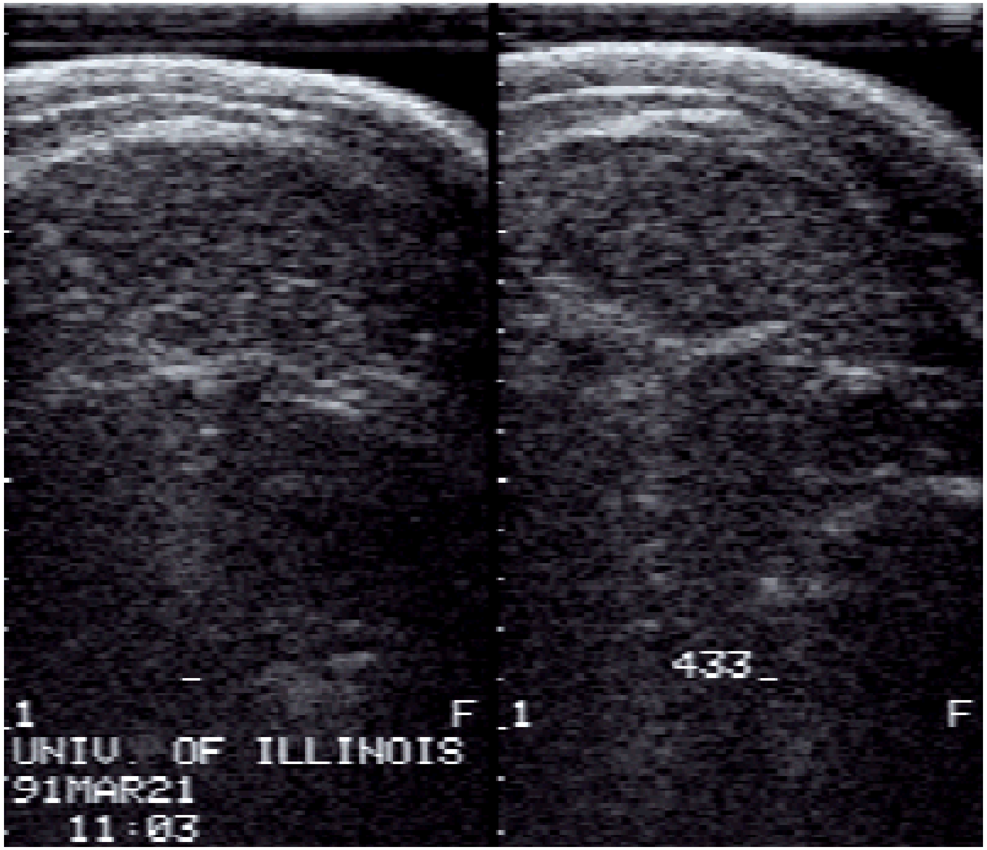


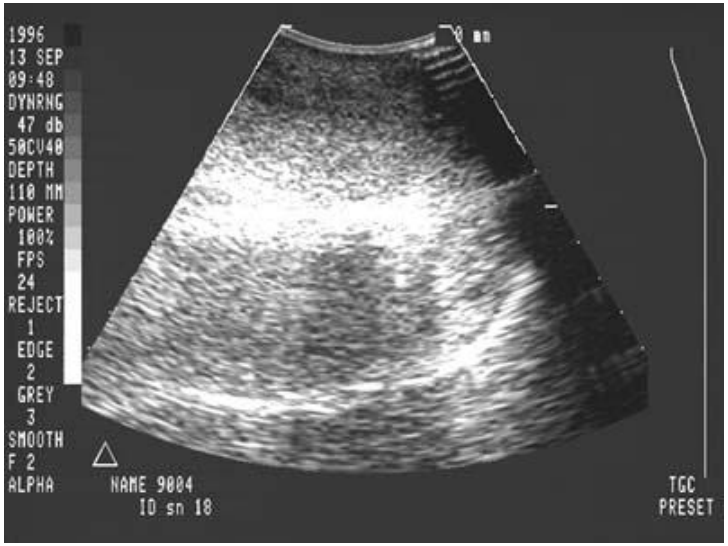
Fig. 2. Representation of a closed contour by elliptic Fourier descriptors. (a) Input. (b) Series truncated at 16 harmonics. (c) Series truncated to four harmonics.

Detection of cancerous regions

[Foran, Comaniciu, Meer, Goodell, 00]

Boundaries in Ultrasound Images





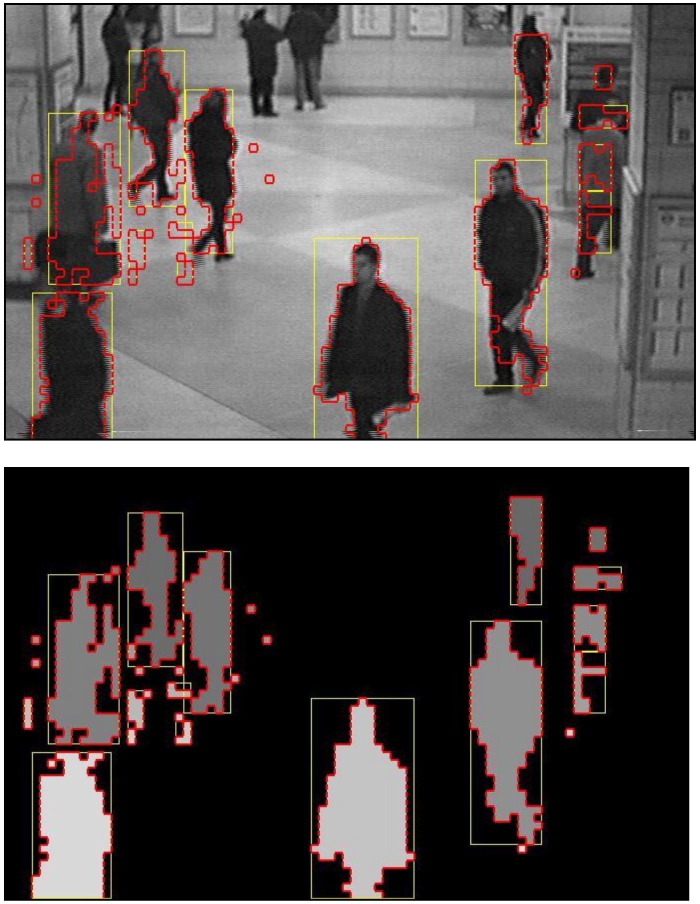
Hard to detect in the presence of large amount of speckle noise

Sometimes hard even for humans!

What is 'Segmentation'?

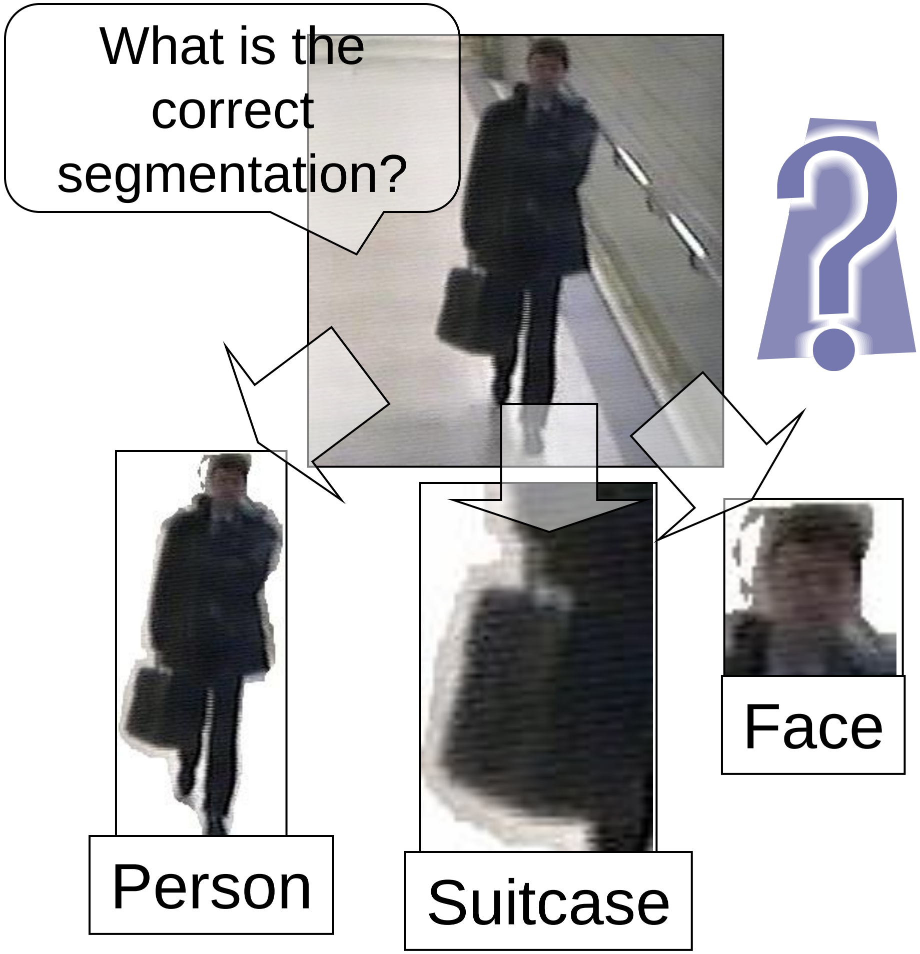
- Separation of the image in different areas
  - Objects
  - Areas with similar visual or semantic characteristics

Not trivial! It is the holy grail of most computer vision problems!



Subjectivity

- A ‘correct’ segmentation result is only valid for a specific context
  - Subjectivity!
  - Hard to implement
  - Hard to evaluate

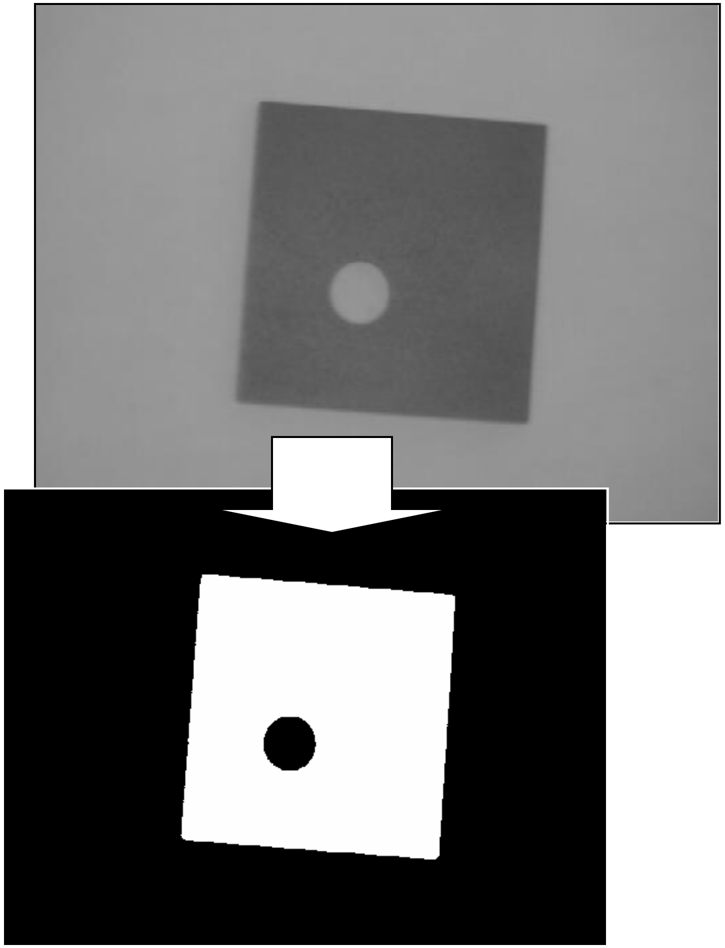


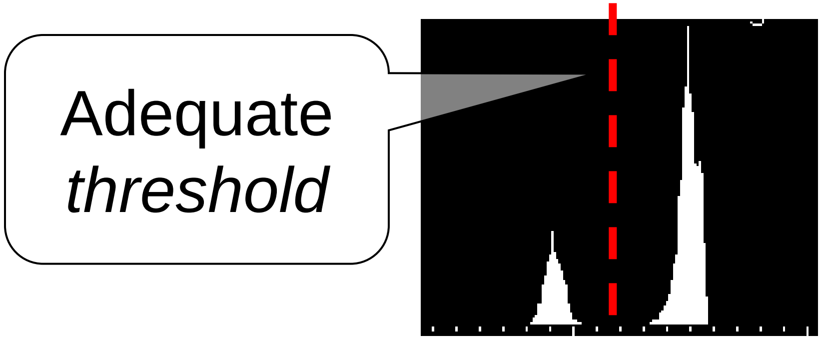
Topic: Thresholding

- Introduction to segmentation
- Thresholding
- Region based segmentation
- Segmentation by clustering

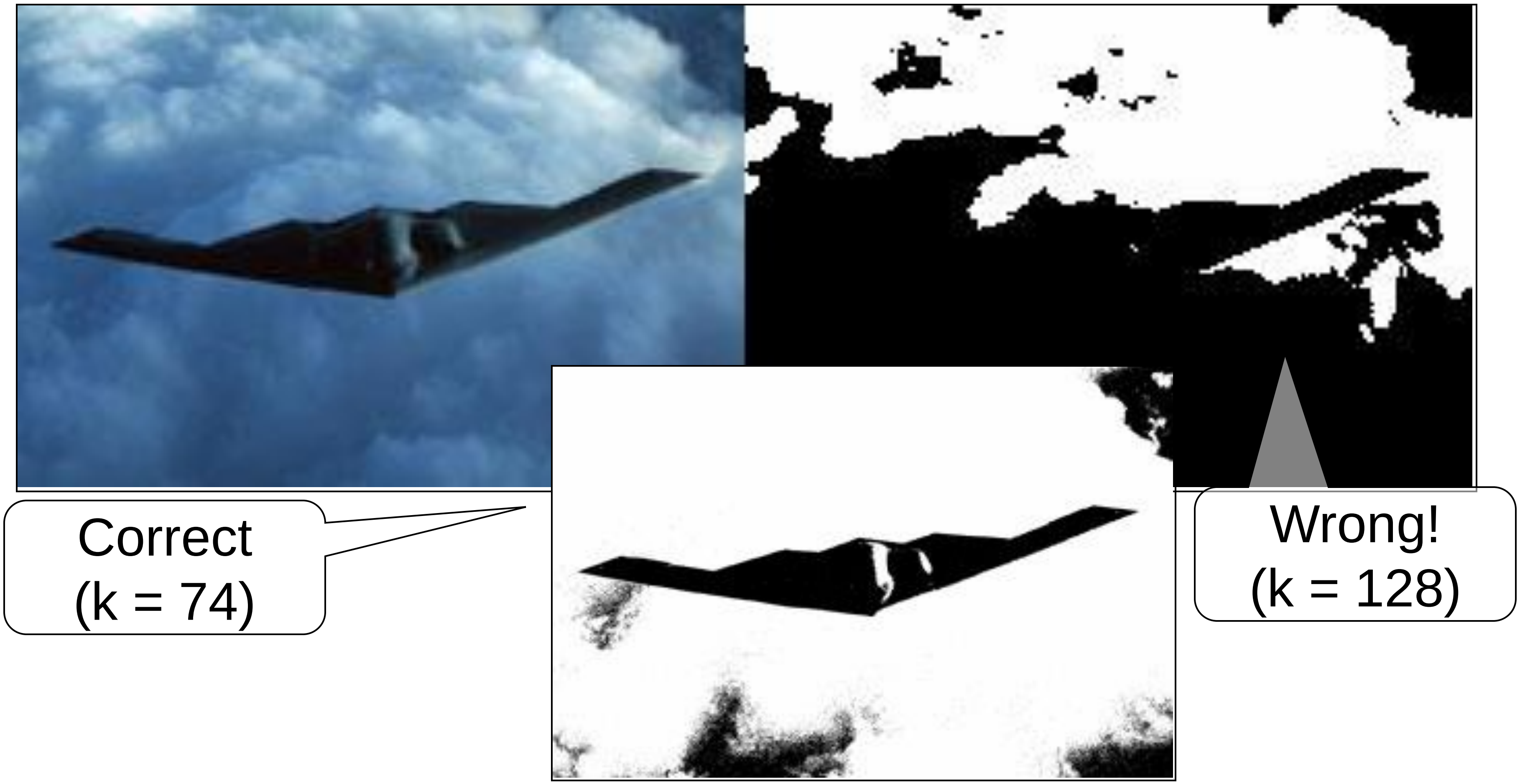
Core Technique: *Thresholding*

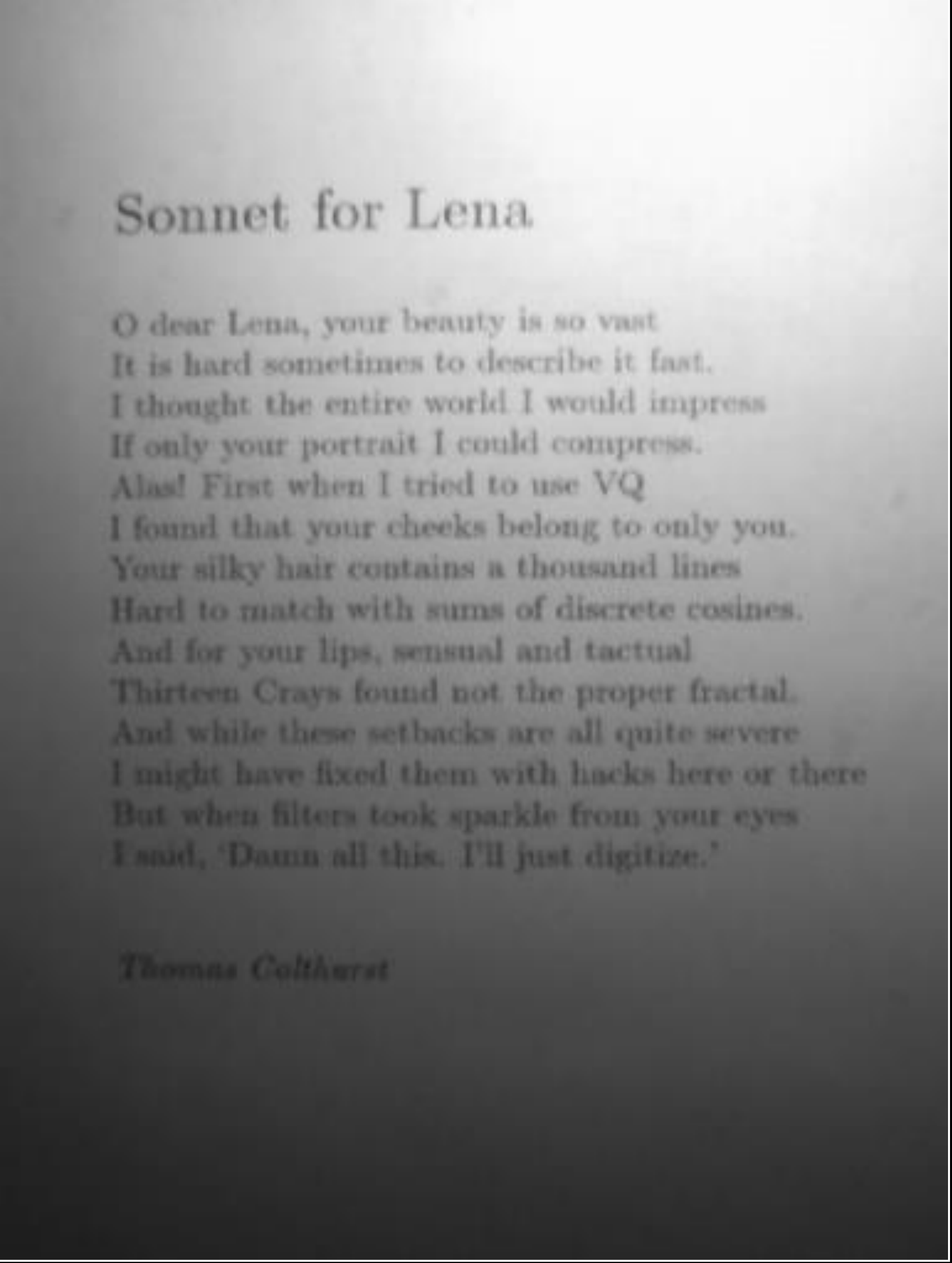
- Divide the image into two areas:
  - 1, if $f(x,y)>K$
  - 0, if $f(x,y)<=K$
- Not easy to find the ideal ***k*** *magic number*
- Core segmentation technique
  - Simple
  - Reasonably effective





Finding the ‘magic number’

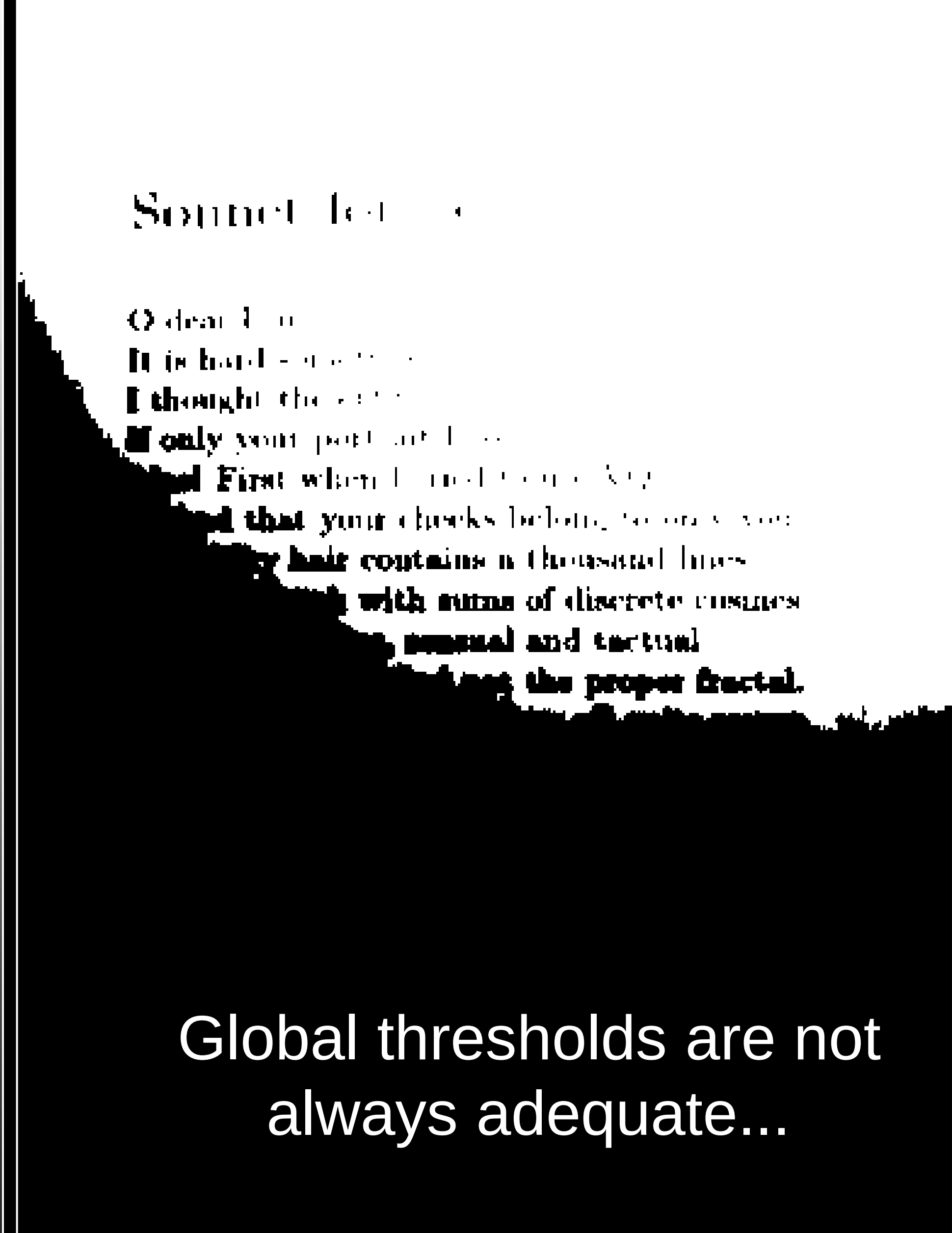




Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

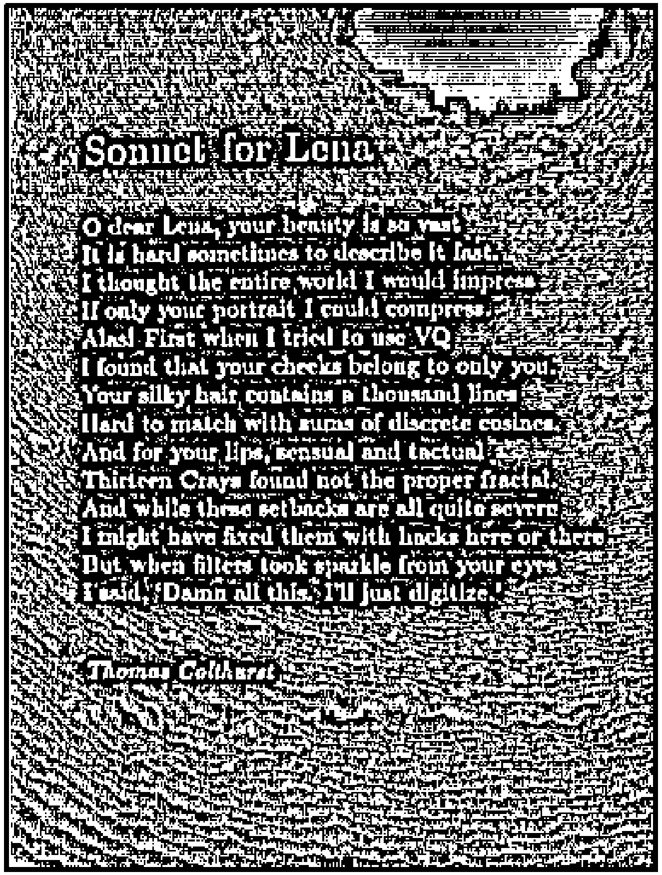
Thomas Colthurst



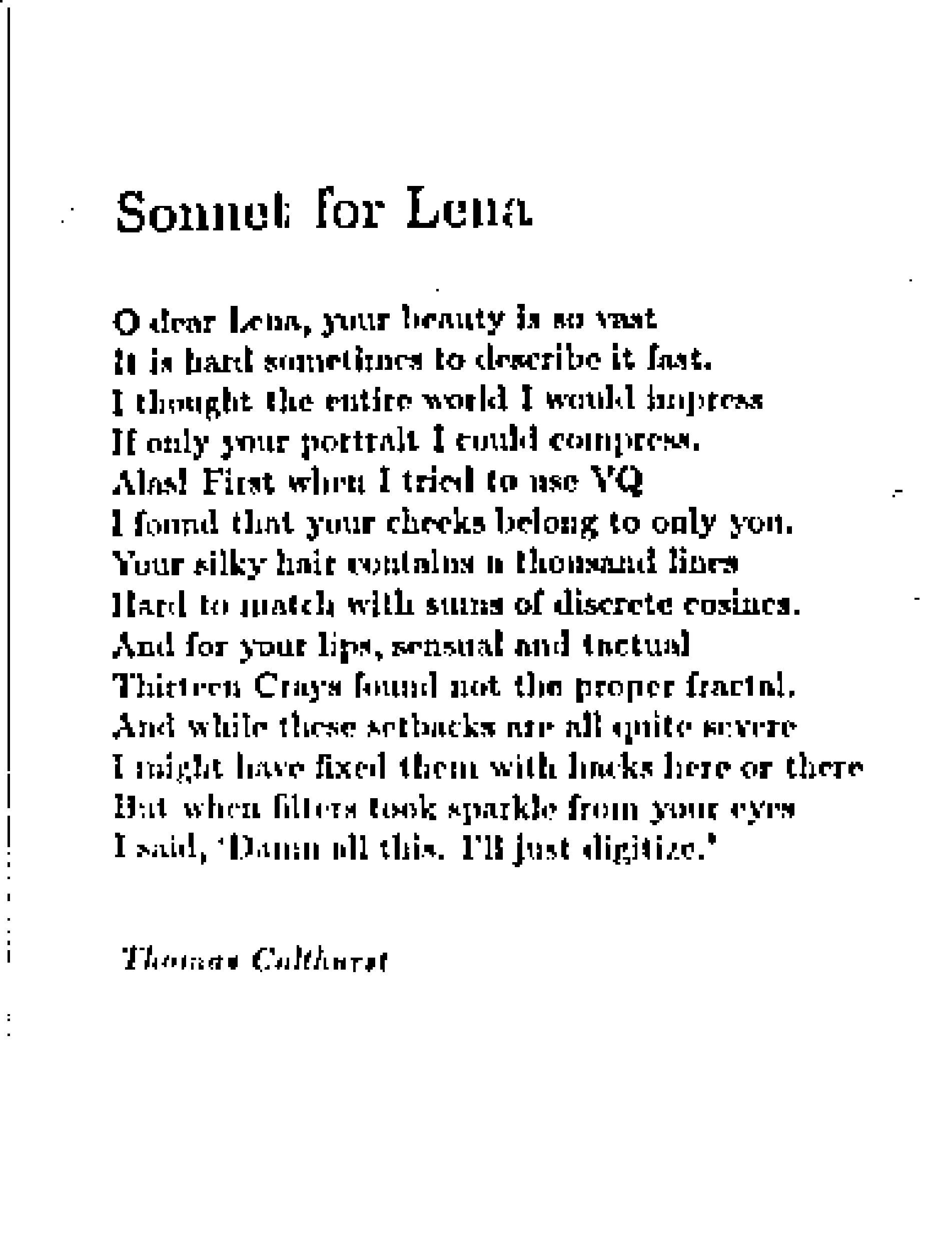
Global thresholds are not always adequate...

Adaptive Thresholding

- Adapt the threshold value for each pixel
- Use characteristics of nearby pixels
- How?
  - Mean
  - Median
  - Mean + K
  - ...



Mean of 7x7 neighborhood

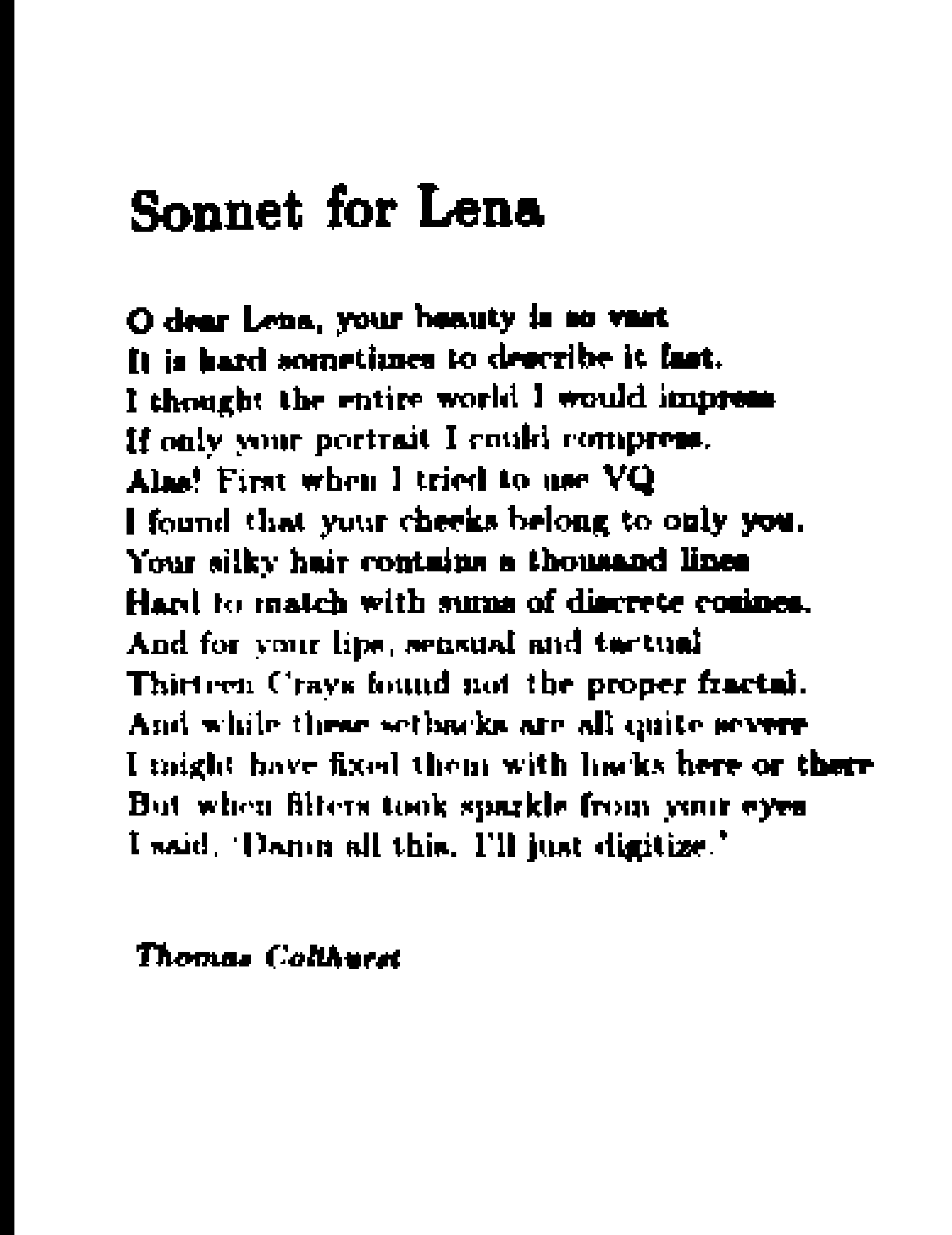


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

7x7 window; K = 7



Sonnet for Lena

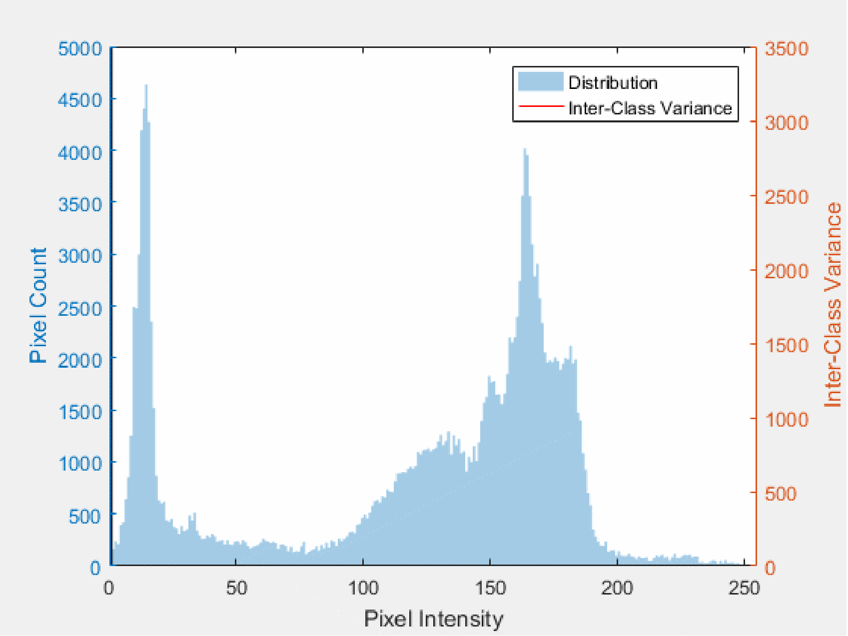
O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

75x75 window; K = 10

Otsu's Thresholding

- Is there an **optimal** threshold for a bimodal distribution?
  - Yes
  - Gist: Minimize **Within-Class Variance**
  - Alternatively: **Maximize Between-Class Variance**



Within Class Variance

- Class Variance
  - The lower the variance, the less dispersed the data is for each class

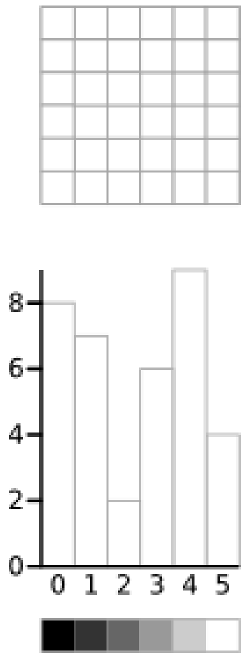
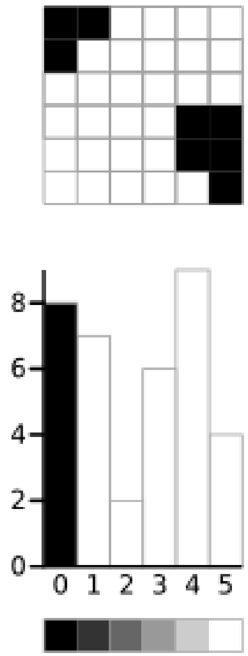
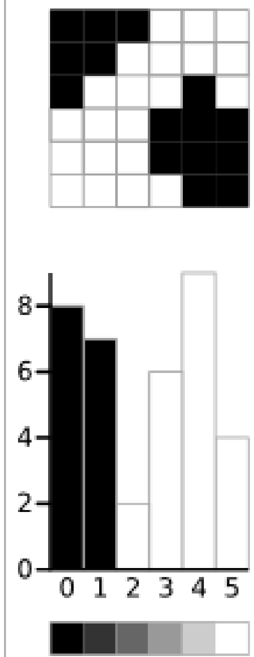
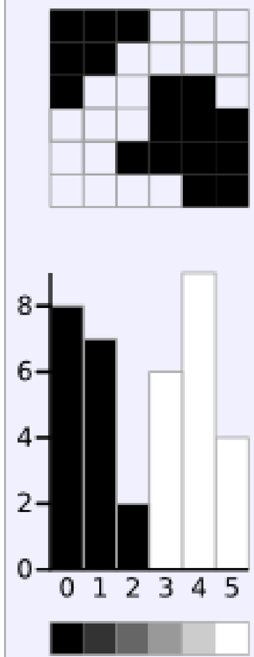
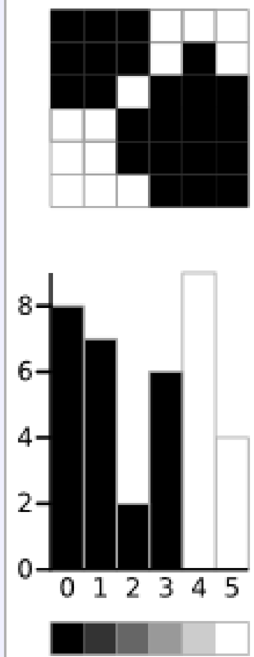
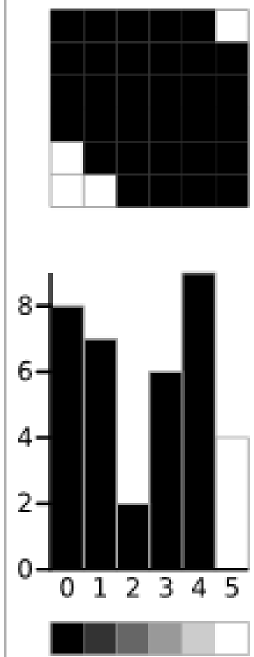
$$\sigma^2 = \frac{\sum_{i=0}^{N}(Xi - \mu)^2}{N}$$

Xi is the pixel value, μ is the mean, and N is the number of pixels in one image

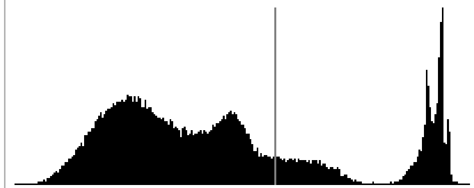
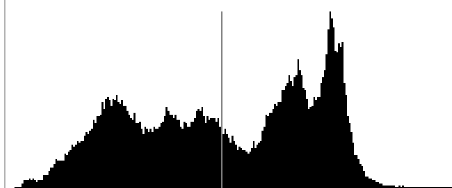
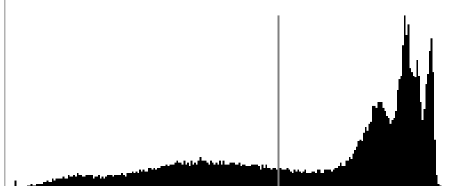
- Within Class Variance
  - Weighted sum of each class variance:
    - Background (b);
    - Foreground (f)

$$\sigma_w^2 = W_b\sigma_b^2 + W_f\sigma_f^2$$

Wj is the percentage of image pixels belonging to class j

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2_b = 0$	$\sigma^2_b = 0$	$\sigma^2_b = 0.2489$	$\sigma^2_b = 0.4637$	$\sigma^2_b = 1.4102$	$\sigma^2_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2_f = 3.1196$	$\sigma^2_f = 1.9639$	$\sigma^2_f = 0.7755$	$\sigma^2_f = 0.5152$	$\sigma^2_f = 0.2130$	$\sigma^2_f = 0$
Within Class Variance	$\sigma^2_W = 3.1196$	$\sigma^2_W = 1.5268$	$\sigma^2_W = 0.5561$	$\boldsymbol{\sigma^2_W = 0.4909}$	$\sigma^2_W = 0.9779$	$\sigma^2_W = 2.2491$

Link: http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

Greyscale Image	Binary Image	Histogram
		
		
		

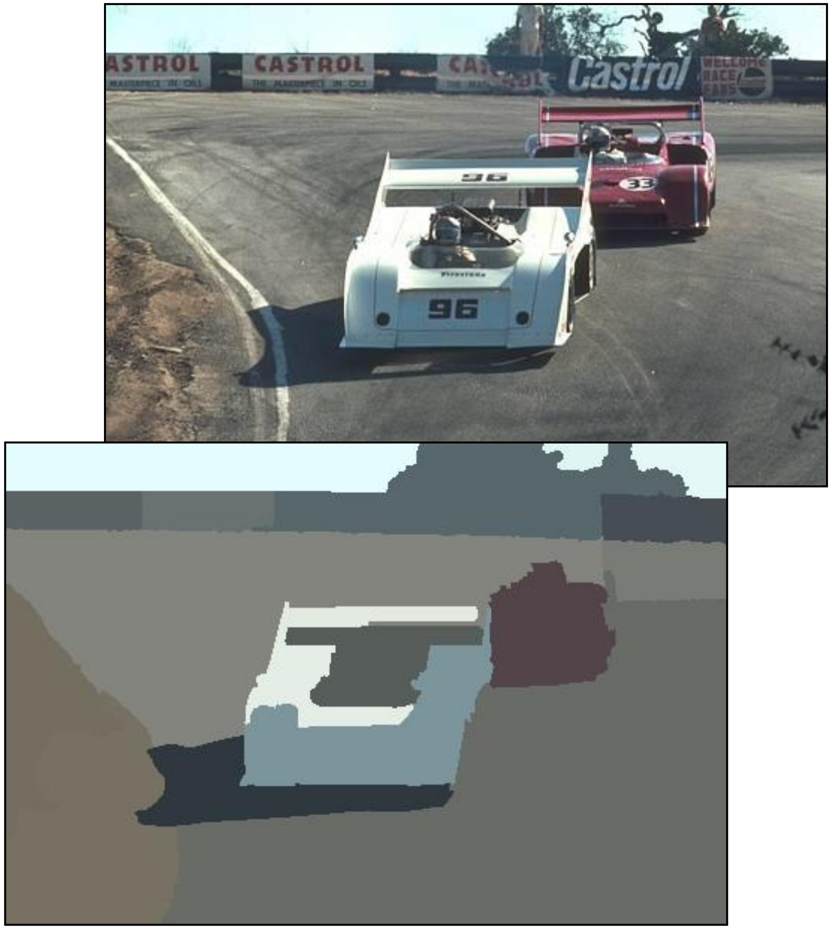
Link: http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

Topic: Region based segmentation

- Introduction to segmentation
- Thresholding
- Region based segmentation
- Segmentation by clustering

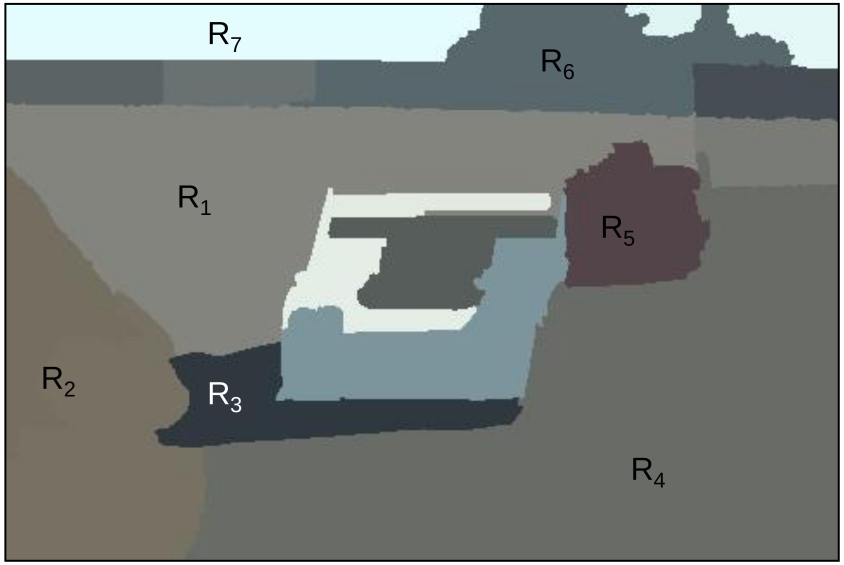
Why Region Based Segmentation?

- Segmentation
  - Edge detection and Thresholding not always effective
- Homogenous regions
  - *Region-based segmentation*
  - Effective in noisy images



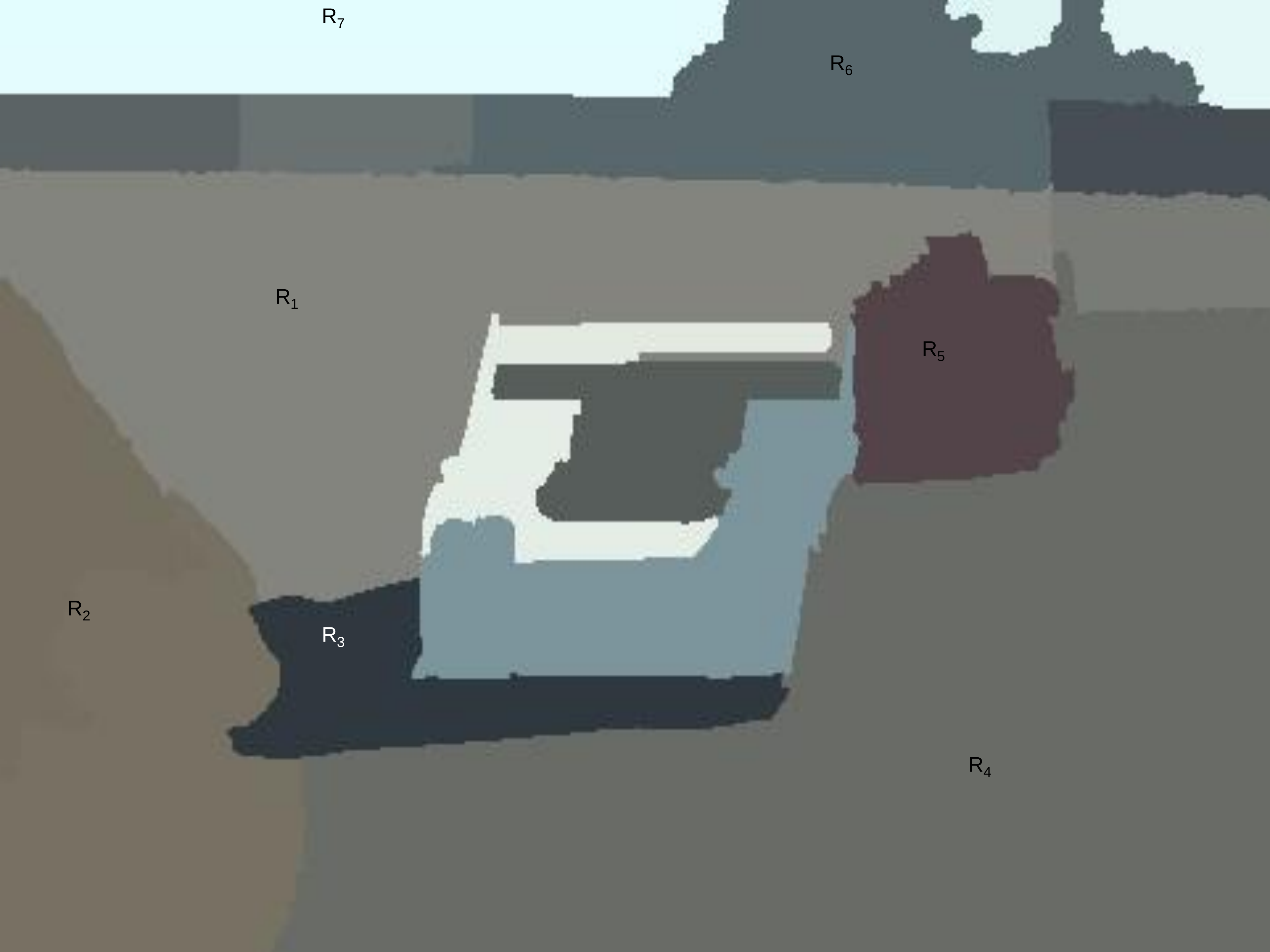
Definitions

- Based on *sets*
- Each image R is a set of regions R_i
  - Every pixel belongs to one region
  - One pixel can only belong to a single region



$$R = \bigcup_{i=1}^{S} R_i \qquad R_i \bigcap R_j = \varnothing$$


ASTROL
CASTROL
Castrol
96
96
33


R7
R6
R1
R5
R2
R3
R4

Basic Formulation

Let R represent the entire image region. Segmentation partitions R into n subregions, R_1, R_2, ..., R_n, such that:

a) $\bigcup_{i=1}^{n} R_i = R$

b) R_i is a connected region, $i = 1, 2, ..., n$.

c) $R_i \cap R_j = \phi$ for all i and $j, i \neq j$

d) $P(R_i) = TRUE$ for $i = 1,2,...,n$.

e) $P(R_i \cup R_j) = FALSE$ for $i \neq j$.

a) Every pixel must be in a region

b) Points in a region must be connected

c) Regions must be disjoint

d) All pixels in a region satisfy specific properties

e) Different regions have different properties

How do we form regions?

- Region Growing
- Region Merging
- Region Splitting
- Split and Merge
- Watershed
- ...

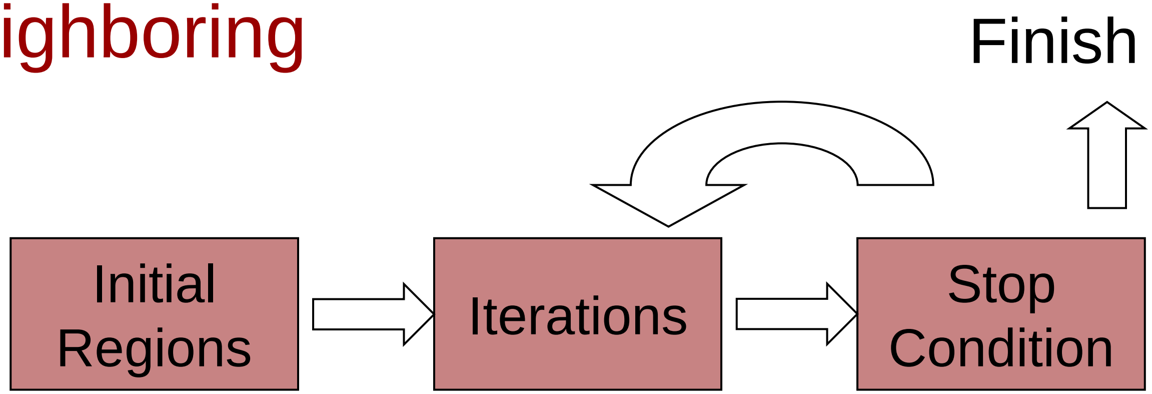
0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

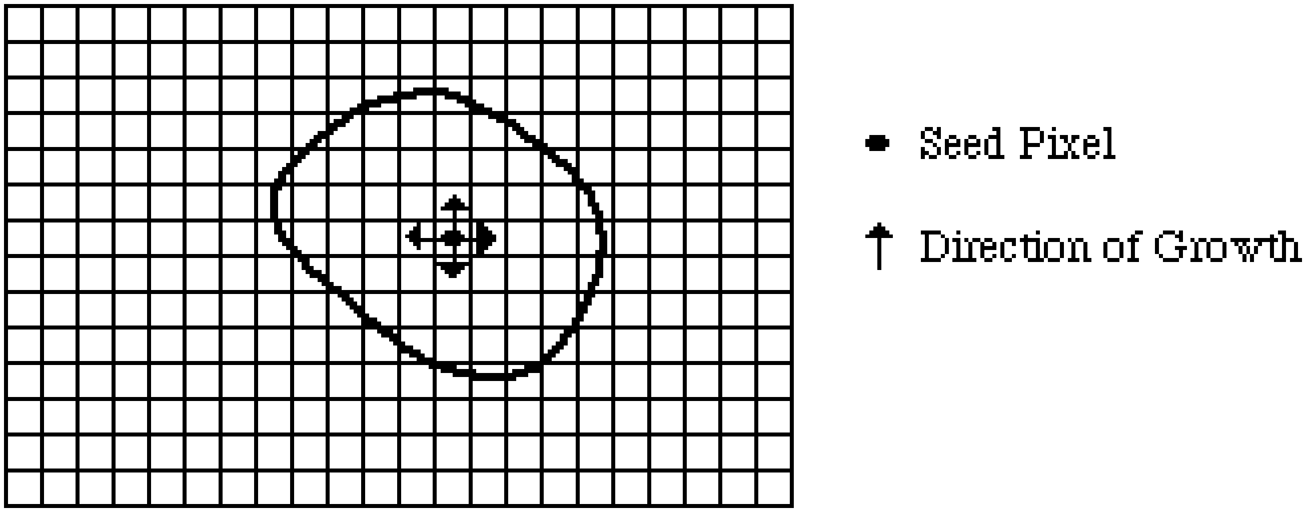
What a computer sees

Region growing

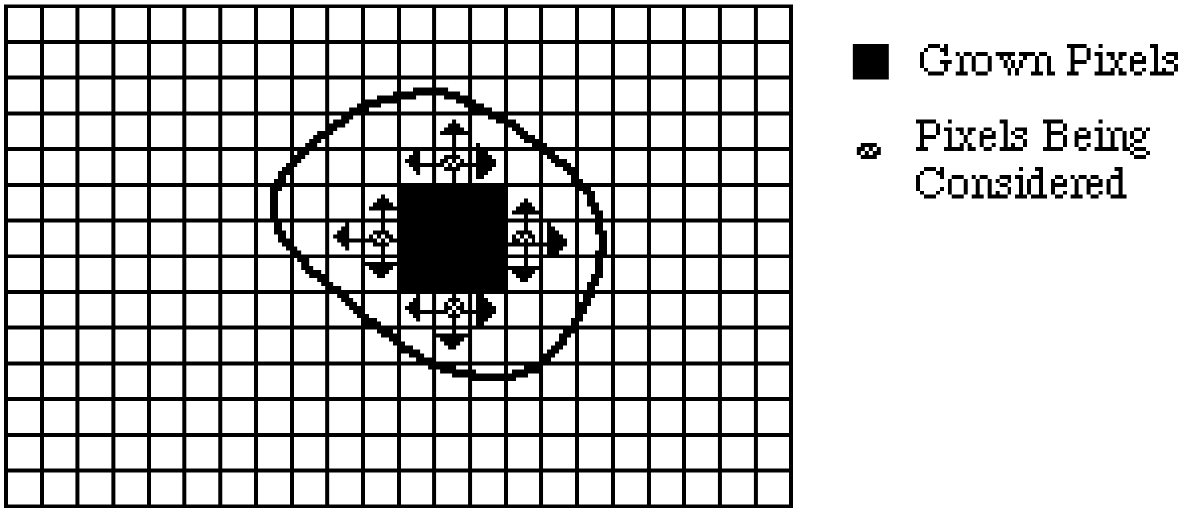
- Groups pixels into larger regions.
- Starts with a **seed** region.
- **Grows** region by **merging** neighboring pixels.

- Iterative process
  - How to start?
  - How to iterate?
  - When to stop?





(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

Region merging

- Algorithm
  - Divide image into an initial set of regions
    - One region per pixel
  - Define a **similarity criteria** for merging regions
  - **Merge** similar regions
  - Repeat previous step until no more merge operations are possible

Similarity Criteria

- Homogeneity of regions is used as the main segmentation criterion in region growing
  - gray level
  - color, texture
  - shape
  - model
  - etc.

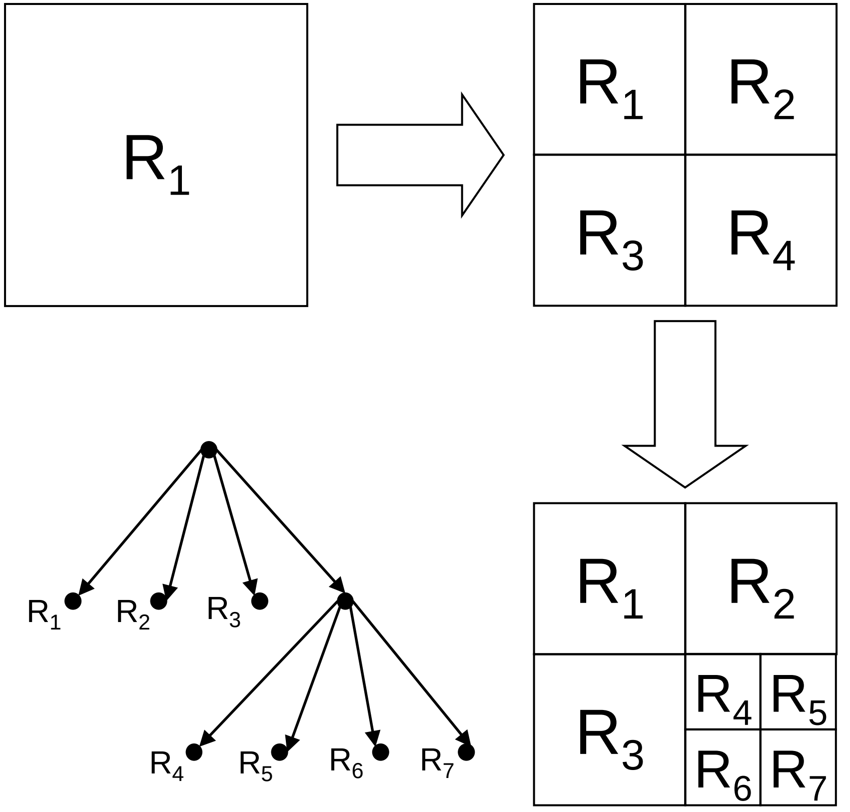
Choice of criteria affects segmentation results dramatically!

Gray-Level Criteria

- Comparing to Original Seed Pixel
  - Very sensitive to choice of **seed point**
- Comparing to Neighbor in Region
  - Allows gradual changes in the region
  - Can cause significant drift
- Comparing to Region Statistics
  - Acts as a **drift dampener**
- Other possibilities!

Region splitting

- Algorithm
  - One initial set that includes the **whole image**
  - **Similarity criteria**
  - Iteratively **split** regions into sub-regions
  - Stop when no more splittings are possible



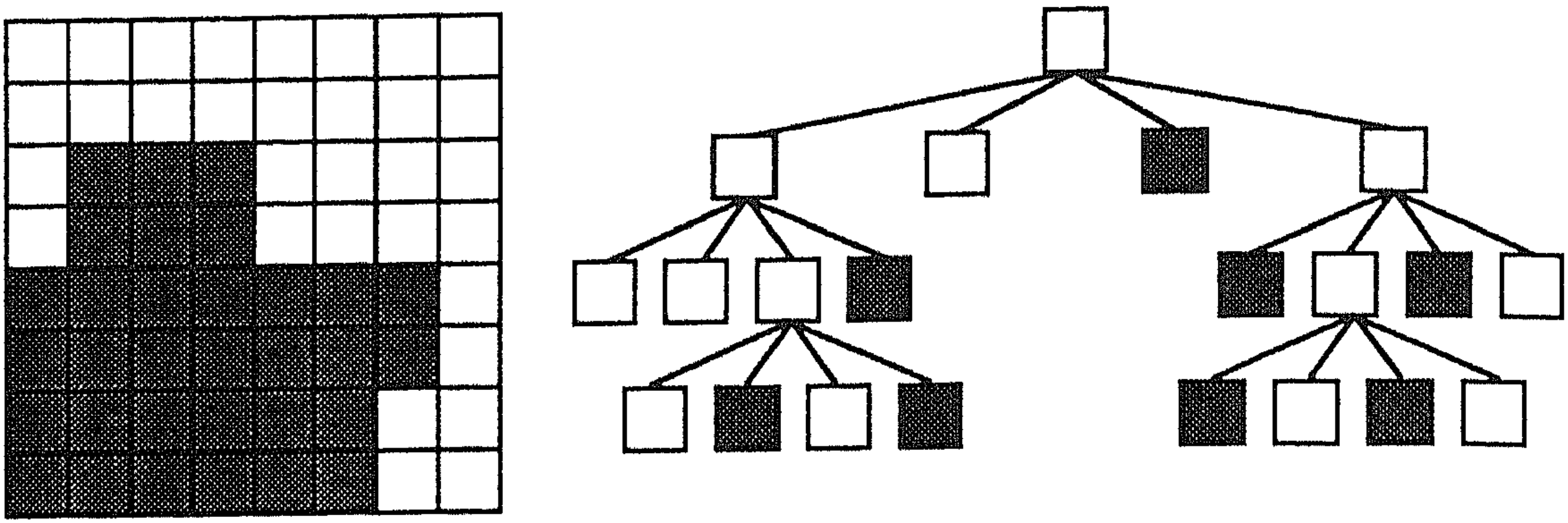
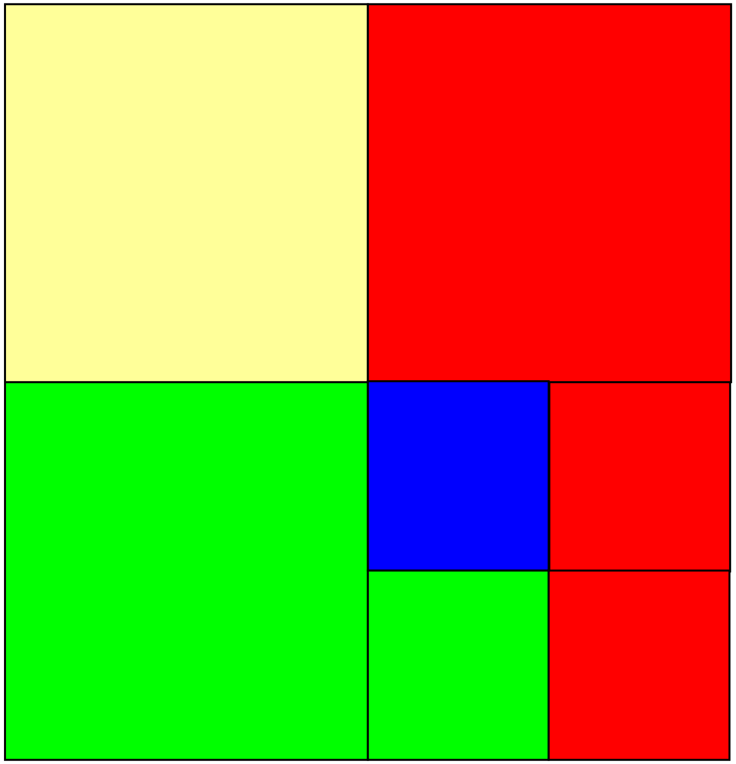


Figure 5.23 A quad-tree representation of an 8×8 binary image.

[Machine Vision; David Vernon]

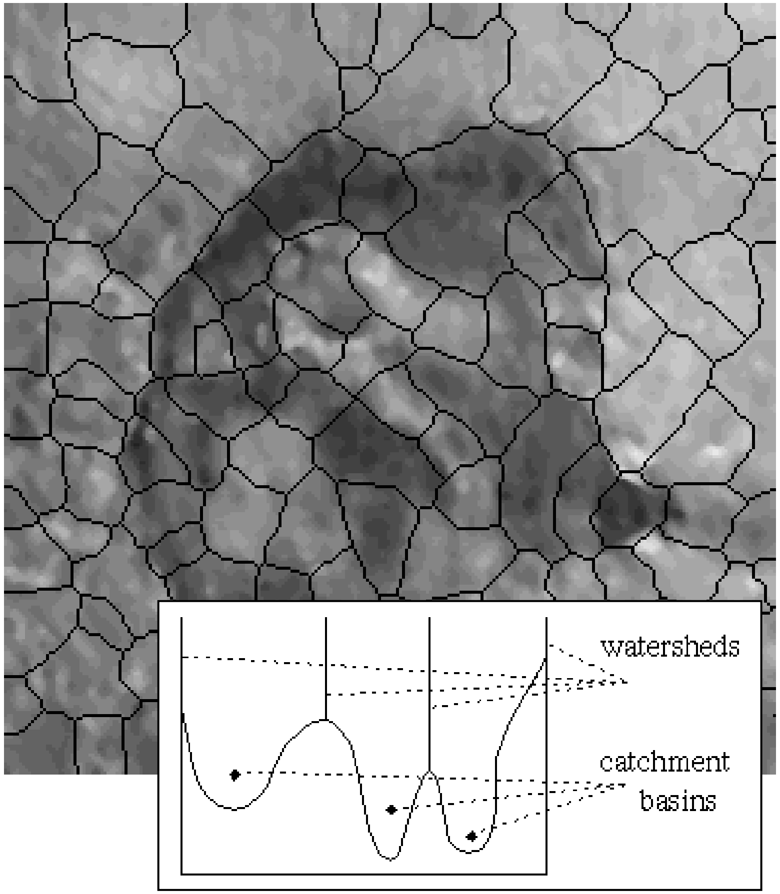
Split and Merge

- Combination of both algorithms
- Can handle a larger variety of shapes
  - Simply apply previous algorithms consecutively



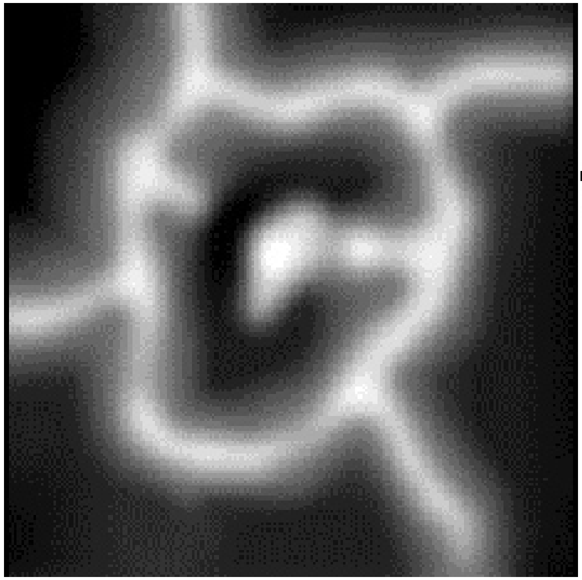
The *Watershed* Transform

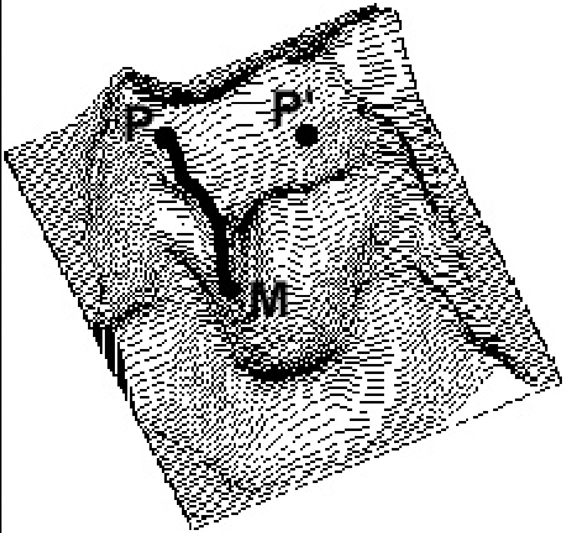
- Geographical inspiration
  - Shed water over rugged terrain
  - Each lake corresponds to a region
- Characteristics
  - Computationally complex
  - Great flexibility in segmentation
  - Risk of over-segmentation

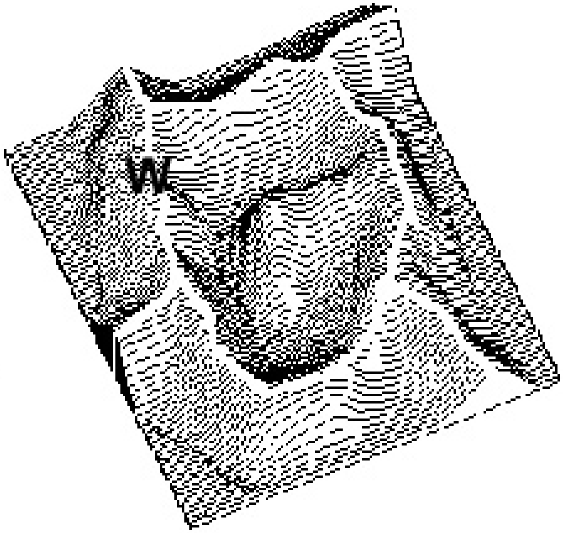


The Drainage Analogy

- Two points are in the same region if they drain to the same point

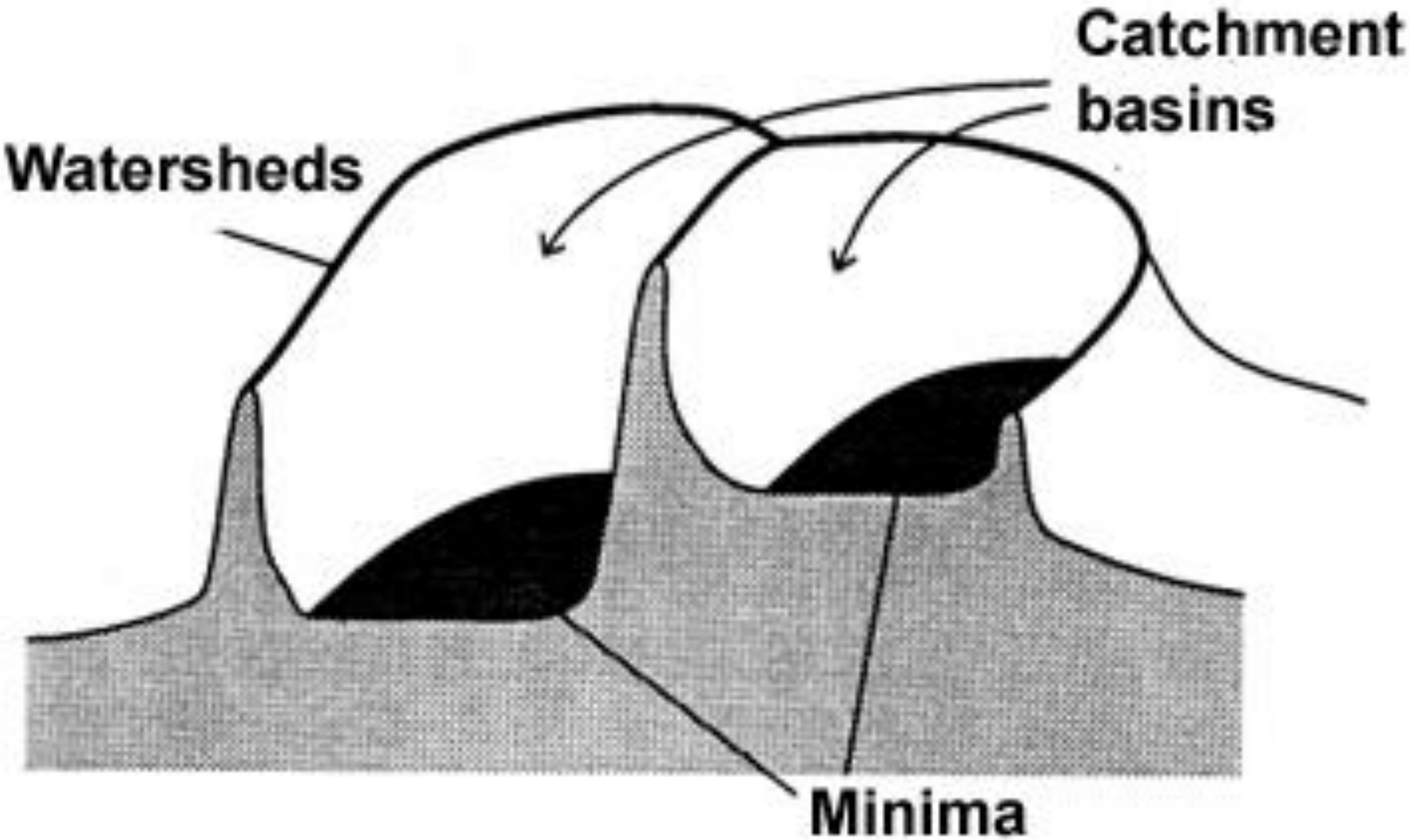


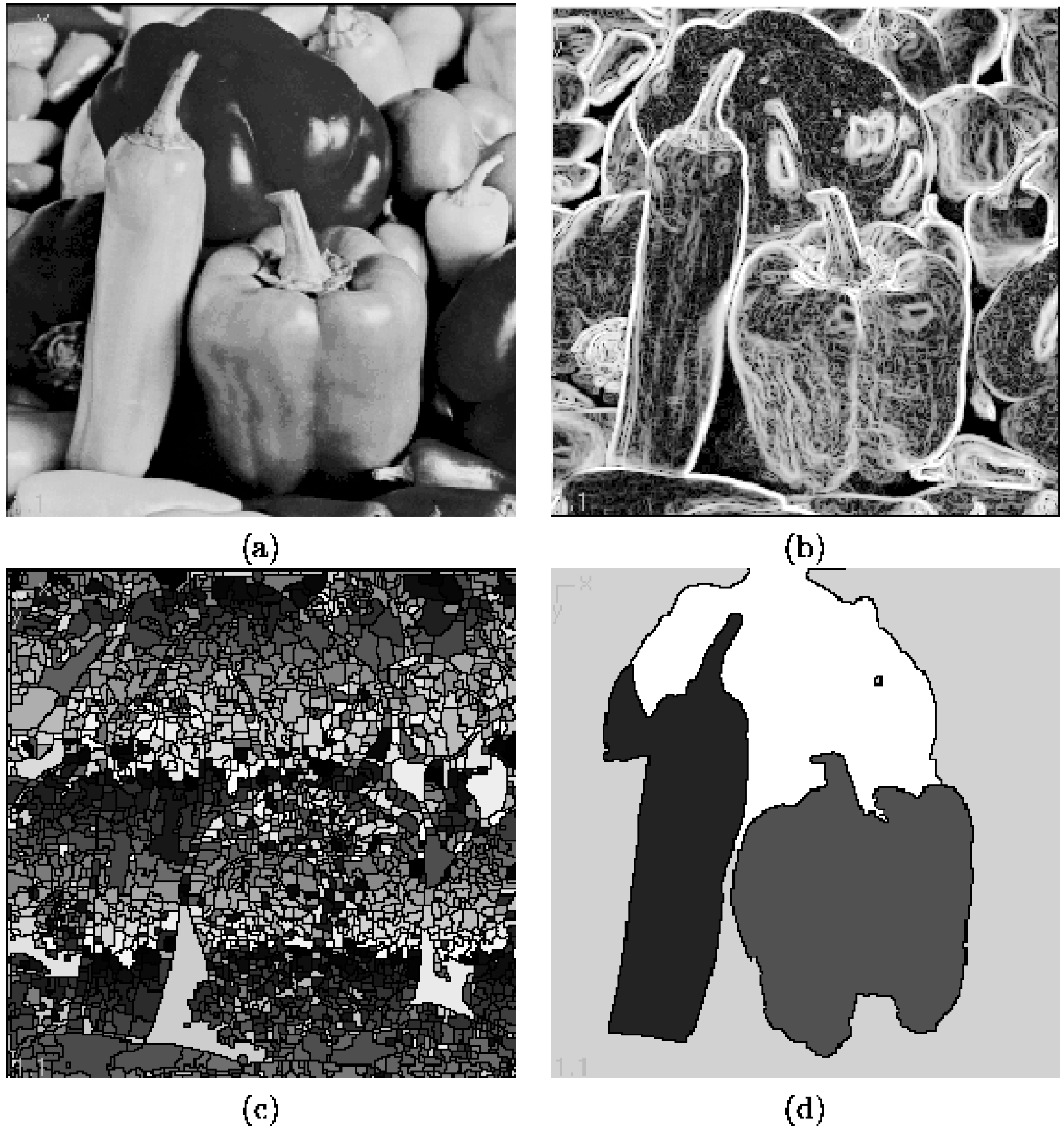




Courtesy of Dr. Peter Yim at National Institutes of Health, Bethesda, MD

The Immersion Analogy





[Milan Sonka, Vaclav Hlavac, and Roger Boyle]

Figure 5.51: *Watershed segmentation: (a) original; (b) gradient image, 3×3 Sobel edge detection, histogram equalized; (c) raw watershed segmentation; (d) watershed segmentation using region markers to control oversegmentation. Courtesy W. Higgins, Penn State University.*

Over-Segmentation

- Over-segmentation
  - Raw watershed segmentation produces a severely oversegmented image with hundreds or thousands of catchment basins
- Post-Processing
  - Region merging
  - Edge information
  - Etc.

Topic: Segmentation by clustering

- Introduction to segmentation
- Thresholding
- Region based segmentation
- Segmentation by clustering

What is ‘Segmentation’? (again?)

- Traditional definition:

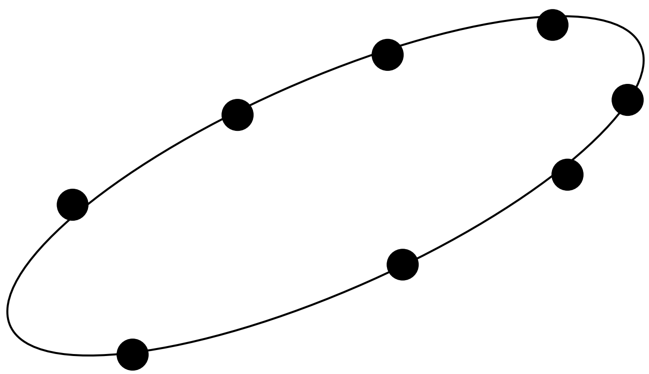
  “Separation of the image in different areas“

  - Decompose an image into “superpixels”
  - Colour and texture coherence.

- Aren’t there other ways to look at the ‘Segmentation’ concept?

Other '*Segmentation*' problems

- Fitting lines to edge points



We can't see this as 'separating an image in different areas'!

- Fitting a fundamental matrix to a set of feature points

This one is complicated!
Check Forsyth and Ponce, chap.14

Segmentation as Clustering

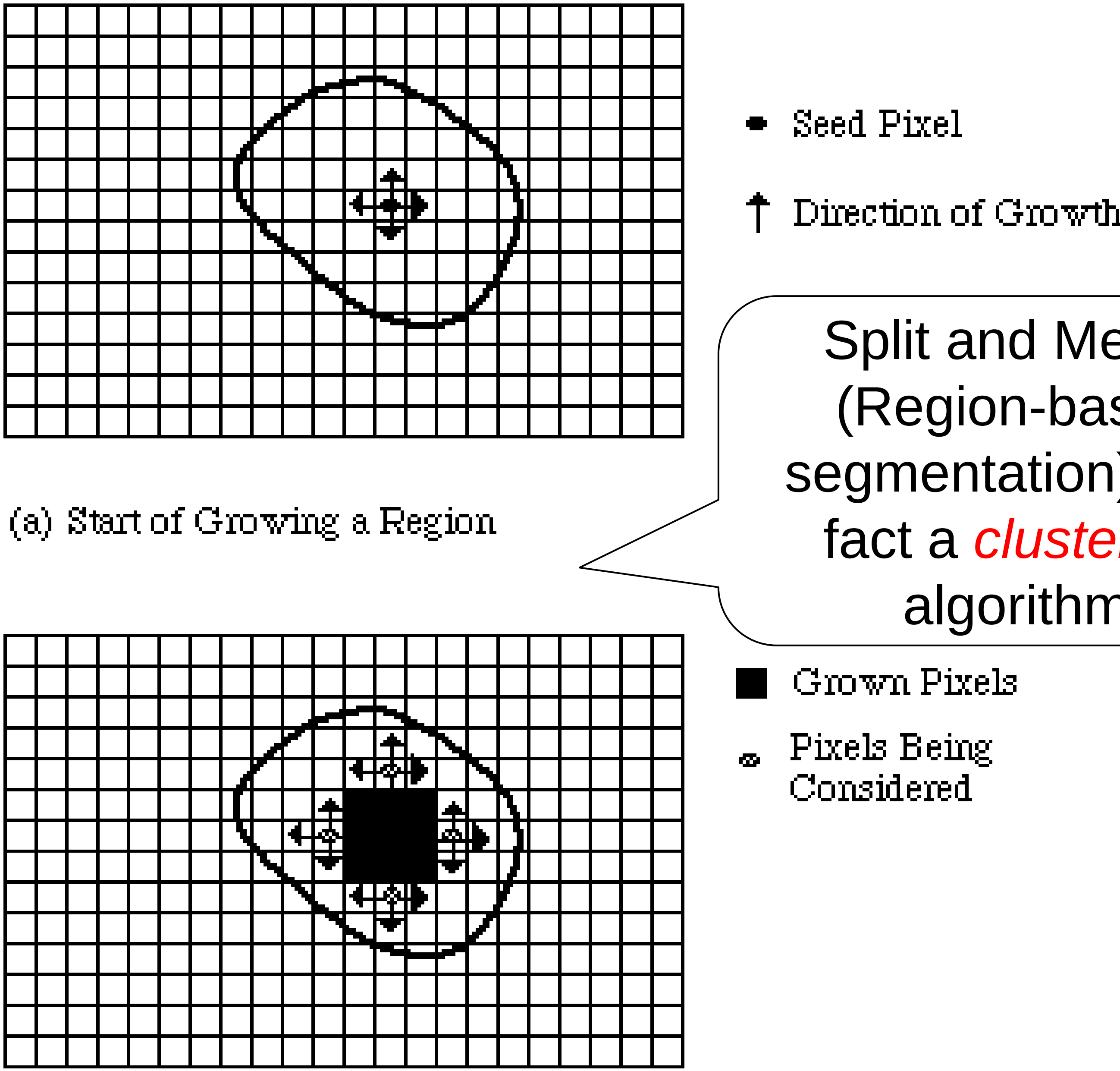
- Tries to answer the question:

  "Which components of the data set naturally belong together?"

- Two approaches:
  - Partitioning
    - Decompose a large data set into pieces that are 'good' according to our model
  - Grouping
    - Collect sets of data items that 'make sense' according to our model

Simple clustering

- Two natural types of clustering:
  - Divisive clustering
    - Entire data set is regarded as a cluster
    - Clusters are recursively split
  - Agglomerative clustering
    - Each data item is a cluster
    - Clusters are recursively merged
- Where have I seen this before?



(b) Growing Process After a Few Iterations

Split and Merge (Region-based segmentation) is in fact a *clustering* algorithm

Generic simple clustering algorithms

- Divisive Clustering
  - Construct a single cluster containing all points
  - While the clustering is not satisfactory
    - Split the cluster that yields the two components with the largest inter-cluster distance
  - end
- Agglomerative Clustering
  - Make each point a separate cluster
  - Until the clustering is satisfactory
    - Merge the two clusters with smallest inter-cluster distance
  - end

Which inter-cluster distance?

What does this mean?

Simple clustering with images

- Some specific problems arise:
  - Lots of pixels! Graphical representations are harder to read
  - Segmentation: It is desirable that certain objects are connected. How to enforce this?
  - When do we stop splitting/merging process?
- Complex situations require more complex clustering solutions!

K-means Clustering

- What if we know that there are *k* clusters in the image?
- We can define an *objective function*!
  - Expresses how good my representation is
- We can now build an algorithm to obtain the *best* representation

Caution! "*Best*" given my objective function!

K-means Clustering

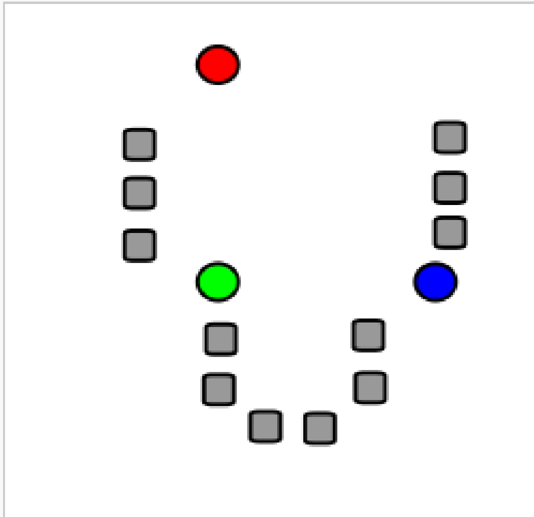
- Assume:
  - We have k clusters
  - Each cluster i has a centre c_i
  - Element j to be clustered is described by a feature vector x_j
- Our objective function is thus:

What does this mean?

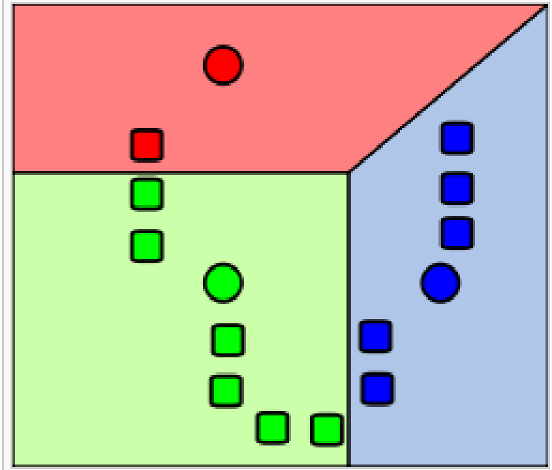
$$\Phi(clusters, data) = \sum_{i \in clusters} \left\{ \sum_{j \in cluster(i)} (x_j - c_i)^T (x_j - c_i) \right\}$$

Iteration step

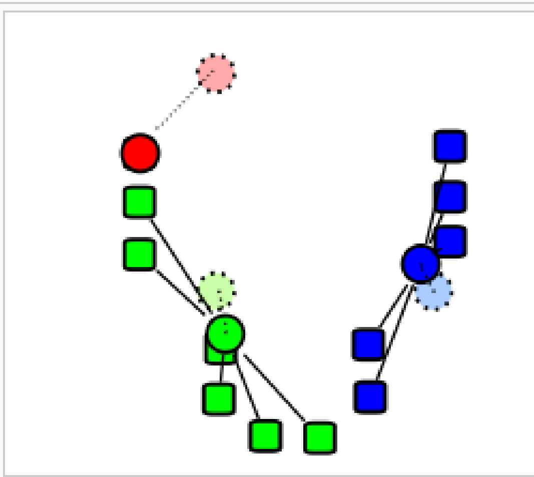
- Too many possible allocations of points to clusters to search this space for a minimum
- Iterate!
  - Assume cluster centres are known and allocate each point to the closest cluster centre
  - Assume the allocation is known and choose a new set of cluster centres. Each centre is the mean of the points allocated to that cluster



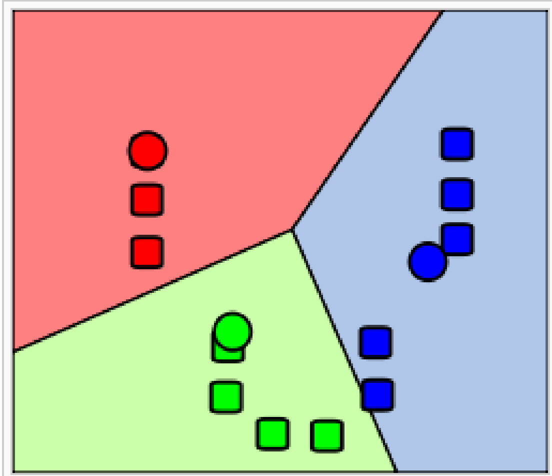
Shows the initial randomized centroids and a number of points.



Points are associated with the nearest centroid.



Now the centroids are moved to the center of their respective clusters.



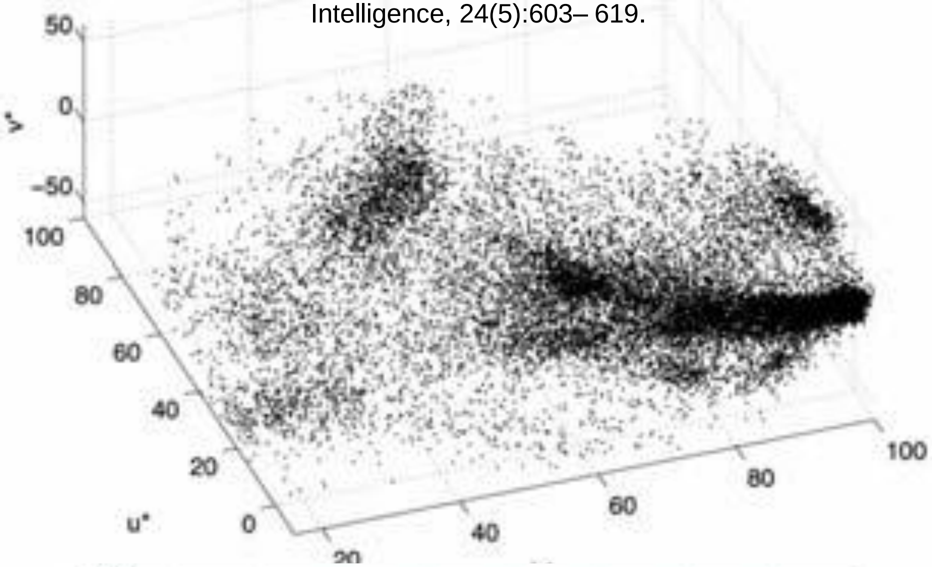
Steps 2 & 3 are repeated until a suitable level of convergence has been reached.

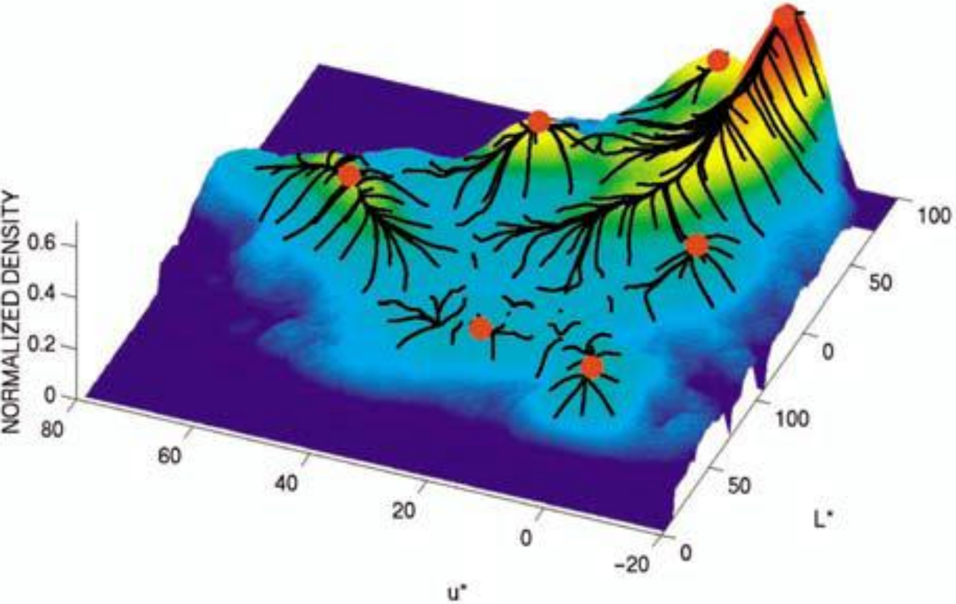
[Wikipedia]

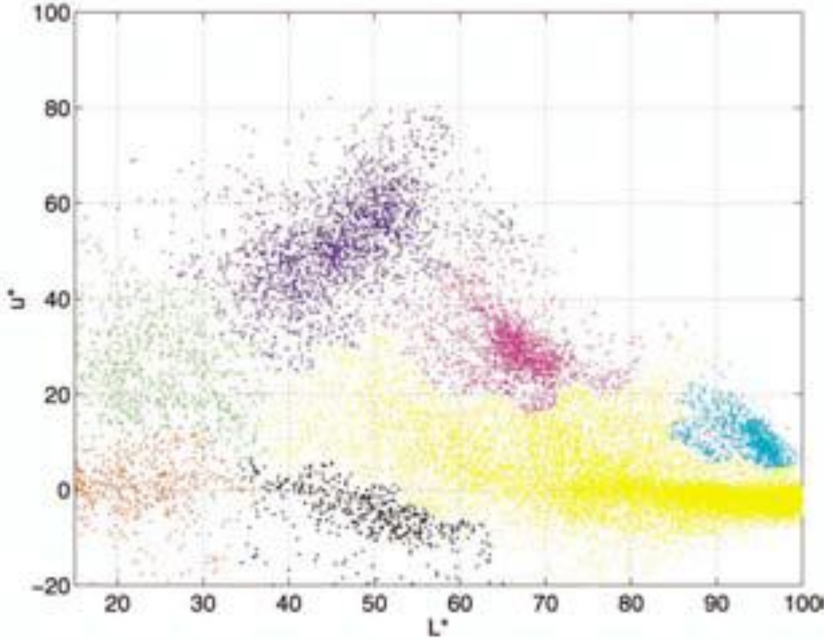
Mean Shift

- K-means:
  - Segments our feature space (and not the image!) into clusters (and not regions)
  - Uses a parametric model for its distributions (e.g. Gaussians), whose locations (centers) and shape (covariance) can be estimated
- Mean shift:
  - Segments an image (and not the feature space) into regions
  - Uses a non-parametric model (simply tries to find distribution peaks)

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603– 619.







How does it work?

- Each pixel ‘finds’ its nearest distribution peak by ‘climbing uphill’ the image’s kernel density function f(x)
- The gradient of f(x) defines the direction for this ‘climb’, by defining mean-shift vectors
- Pixels that ‘climb’ to the same peak form a region

One-dimensional example

Szeliski, “Computer Vision: Algorithms and Applications”, Springer, 2011

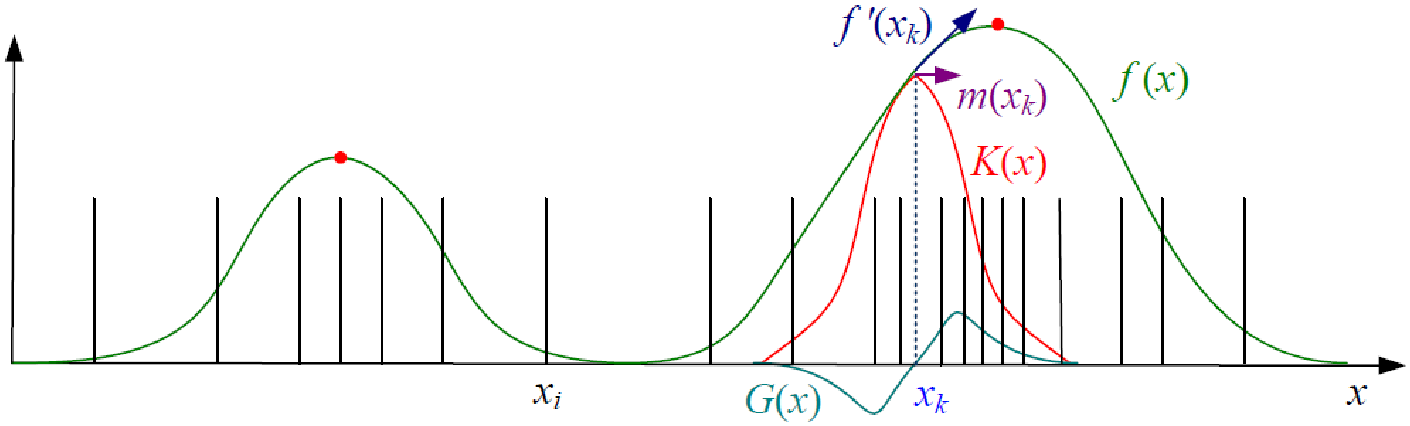
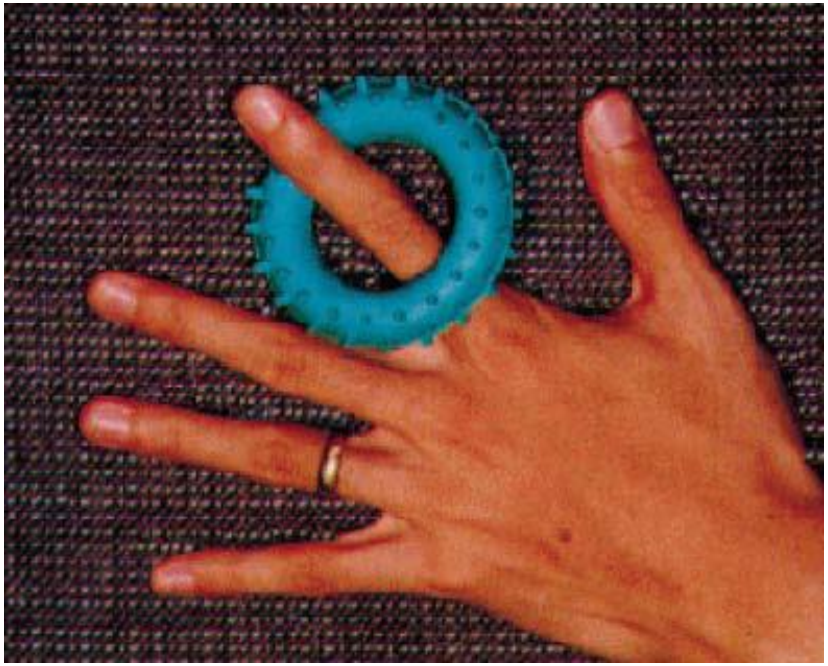
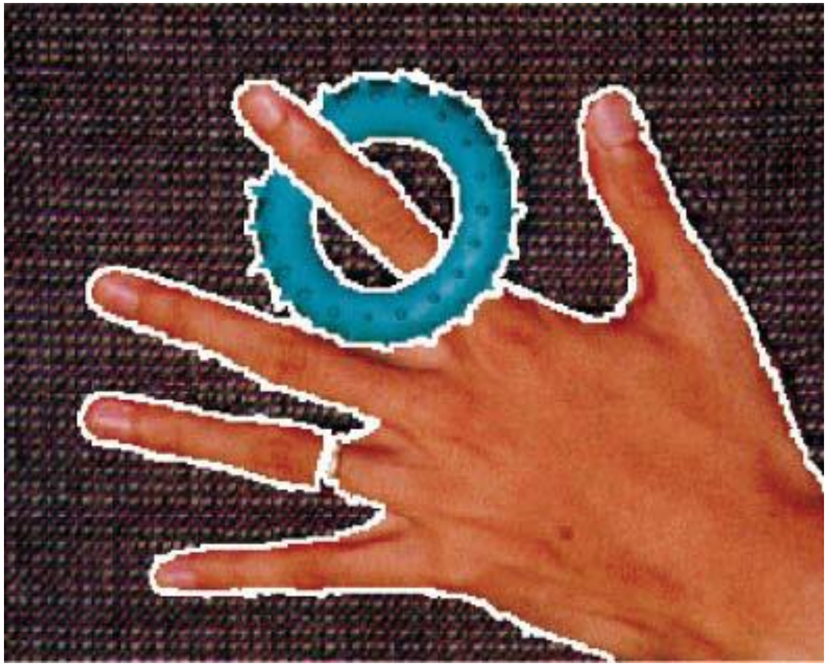


Figure 5.17 One-dimensional visualization of the kernel density estimate, its derivative, and a mean shift. The kernel density estimate $f(x)$ is obtained by convolving the sparse set of input samples x_i with the kernel function $K(x)$. The derivative of this function, $f'(x)$, can be obtained by convolving the inputs with the derivative kernel $G(x)$. Estimating the local displacement vectors around a current estimate x_k results in the mean-shift vector $m(x_k)$, which, in a multi-dimensional setting, point in the same direction as the function gradient $\nabla f(\boldsymbol{x}_k)$. The red dots indicate local maxima in $f(x)$ to which the mean shifts converge.

Mean Shift

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603– 619.





Normalized Cuts

- Clustering can be seen as a problem of "*cutting graphs into good pieces*"
- Data Items
  - Vertex in a weighted graph
  - Weights are large if elements are similar
- Cut edges
  - Cut edges with small weights
  - Keep connected components with large interior weights

Regions!

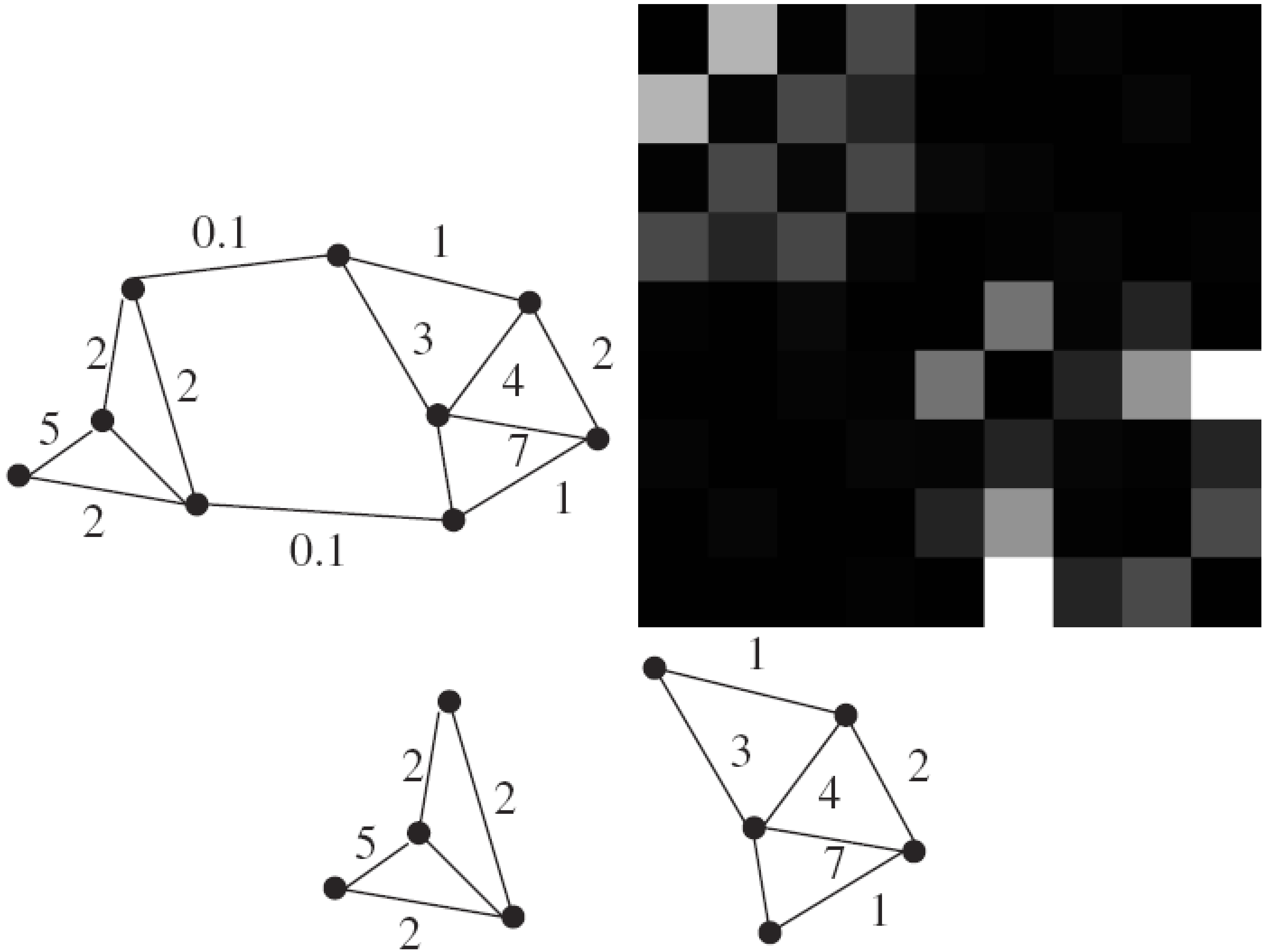


Figure 16.16. On the **top left**, a drawing of an undirected weighted graph; on the **top right**, the weight matrix associated with that graph. Larger values are lighter. By associating the vertices with rows (and columns) in a different order, the matrix can be shuffled. We have chosen the ordering to show the matrix in a form that emphasizes the fact that it is very largely block-diagonal. The figure on the **bottom** shows a cut of that graph that decomposes the graph into two tightly linked components. This cut decomposes the graph's matrix into the two main blocks on the diagonal.

Graphs and Clustering

- Associate each element to be clustered with a vertex on a graph
- Construct an edge from every element to every other
- Associate a weight with each edge based on a similarity measure
- Cut the edges in the graph to form a good set of connected components

Weight Matrices

- Typically look like block diagonal matrices
- Why?
  - Interclusters similarities are strong
  - Intracluster similarities are weak
- Split a matrix into smaller matrices, each of which is a block
- Define *Affinity Measures*

Resources

- Szeliski, “Computer Vision: Algorithms and Applications”, Springer, 2022
  – Chapter 7 – “Feature Detection and Matching”