Digital Image Processing (CSE/ECE 478)

Lecture # 14: Image segmentation

Avinash Sharma

Center for Visual Information Technology (CVIT),

IIIT Hyderabad

Today's Lecture

- Updates on project (TA Evaluation)
  - Get it done by this weekend.
  - Each team will get a score (out of 10)

- Topic: Image Segmentation
  - Gestalt Principles
  - Thresholding based approaches
  - Region based segmentation
  - Clustering based segmentation

Image Segmentation

- Organize the image into meaningful groups/regions

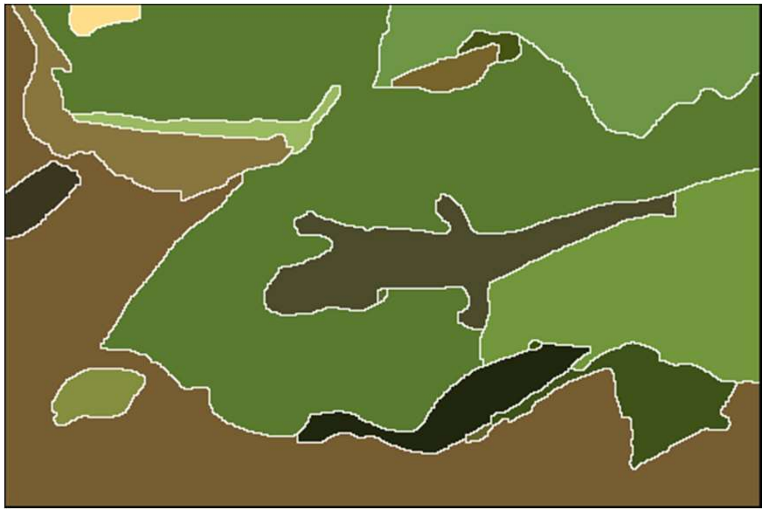
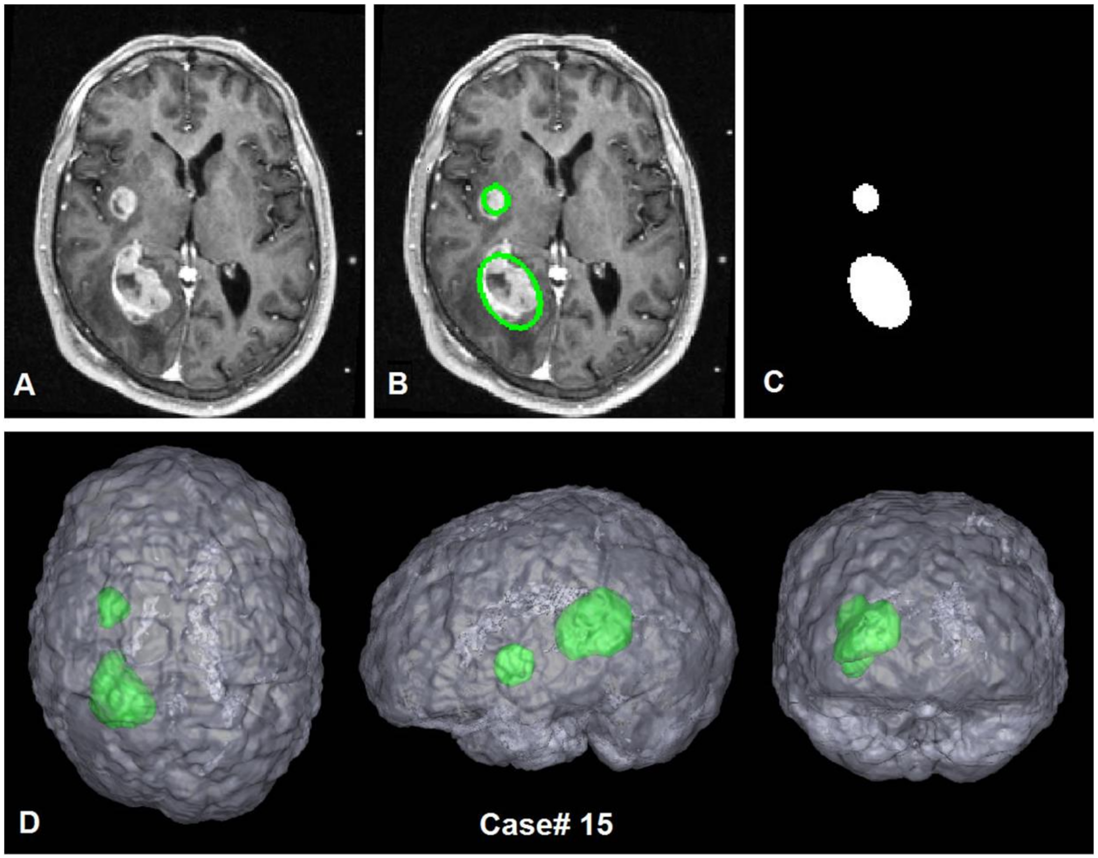
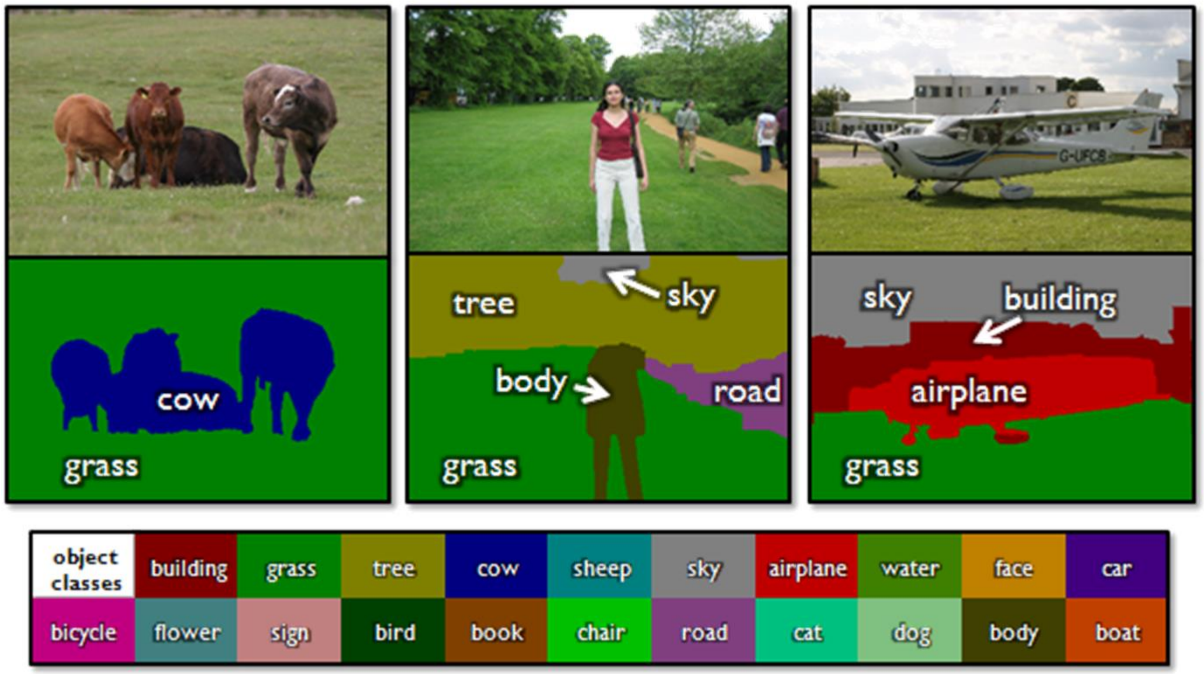


Image Segmentation: Applications



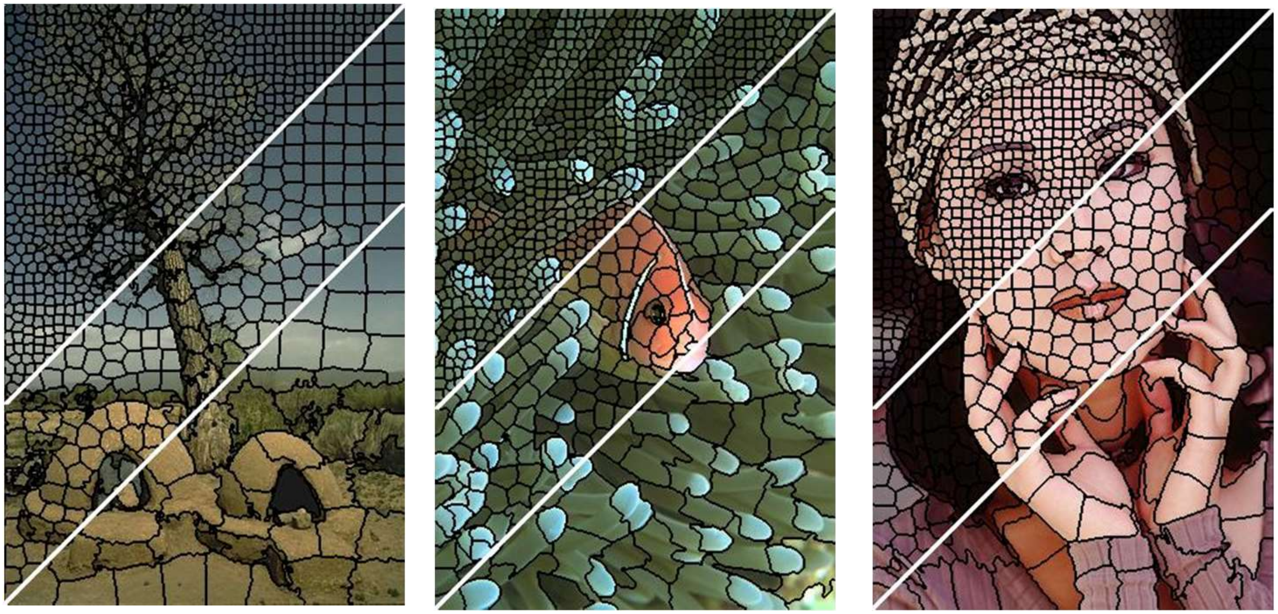
source: vision.cse.psu.edu

Image Segmentation: Applications



TextonBoost: Sheton et al. ECCV 2006

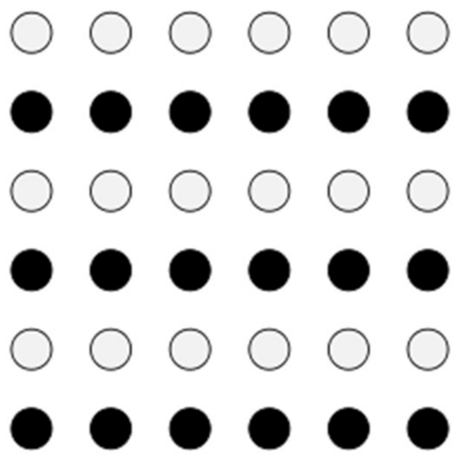
Image Segmentation: Applications

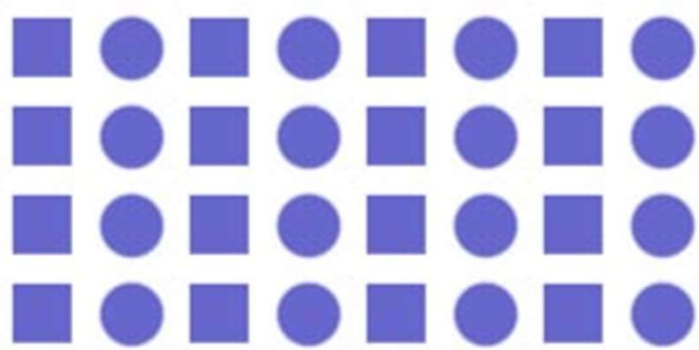


SLIC Superpixels: Achanta et al. TPAMI 2006

Image Segmentation: How humans do it?

- Gestalt Principles of grouping

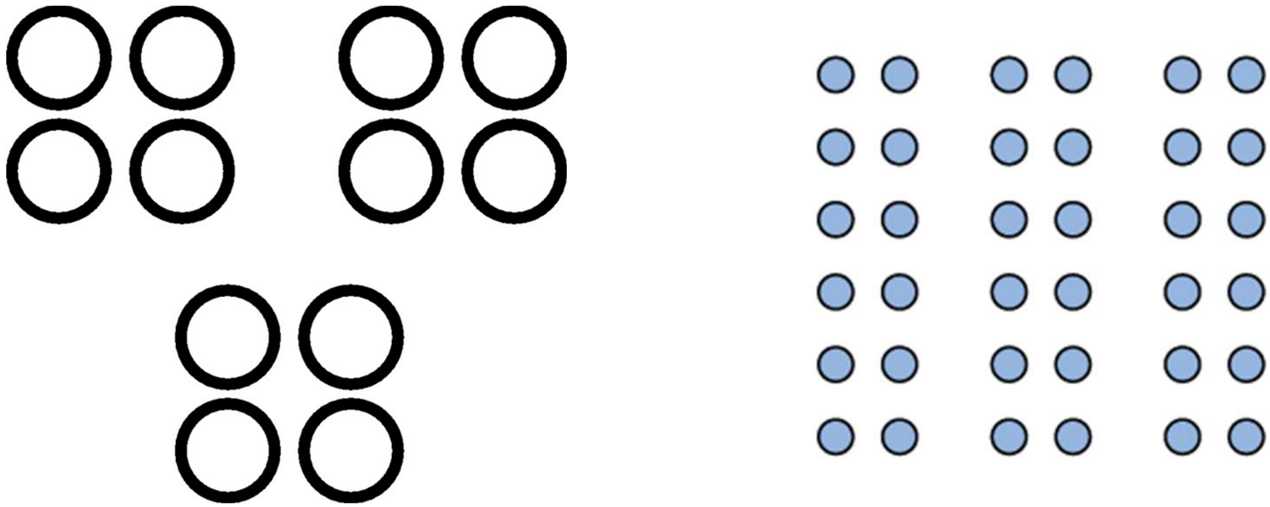




1. Similarity

Image Segmentation: How humans do it?

- Gestalt Principles of grouping



2. Proximity

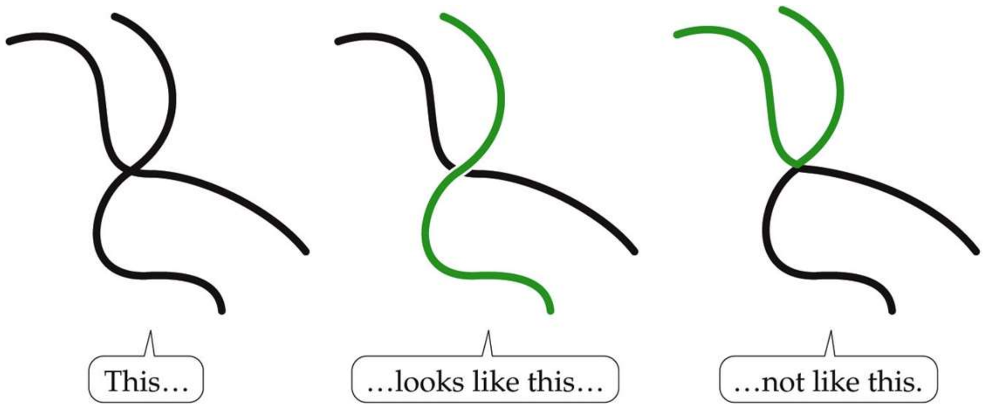
Image Segmentation: How humans do it?

- Gestalt Principles of grouping

3. Closure

Image Segmentation: How humans do it?

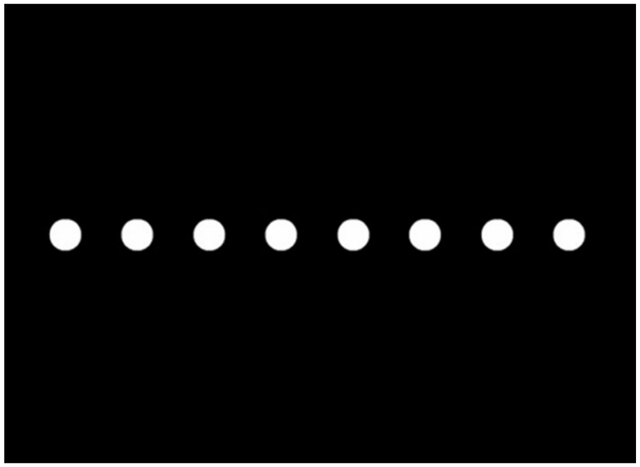
- Gestalt Principles of grouping



4. Good Continuation

Image Segmentation: How humans do it?

- Gestalt Principles of grouping



5. Common Fate

Image Segmentation: How humans do it?

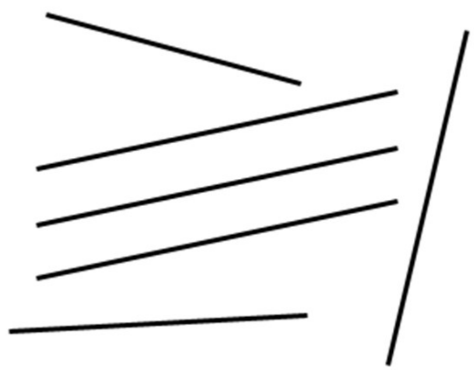
- Gestalt Principles of grouping

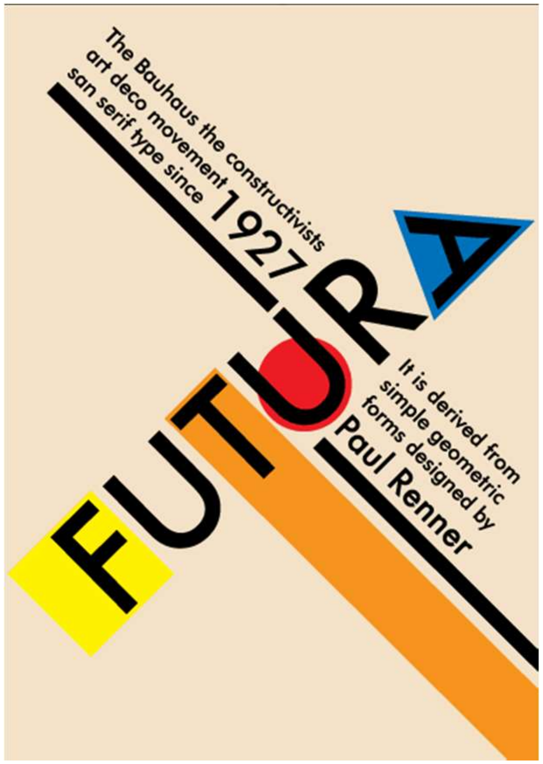
6. Symmetry

https://designschool.canva.com/blog/gestalt-theory/

Image Segmentation: How humans do it?

- Gestalt Principles of grouping

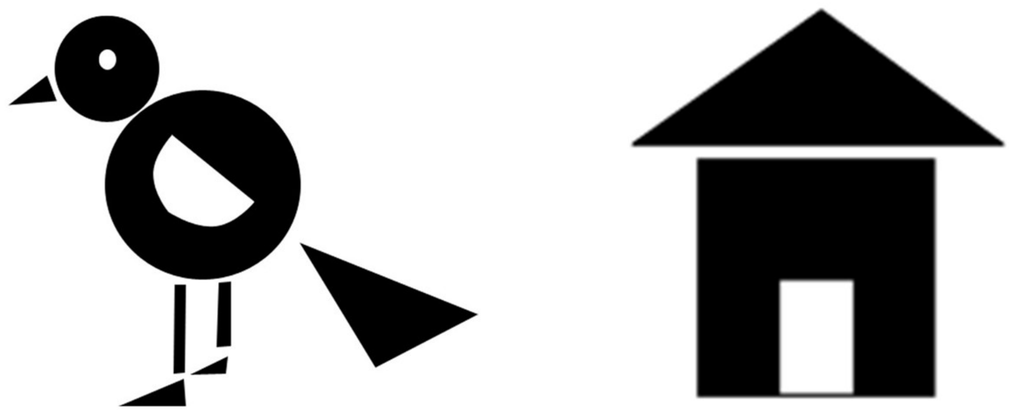




7. Parallelism

Image Segmentation: How humans do it?

- Gestalt Principles of grouping



8. Familiarity

Image Segmentation: How humans do it?

- Gestalt Principles of grouping
  1. Similarity
  2. Proximity
  3. Good continuity
  4. Closure
  5. Common Fate
  6. Symmetry
  7. Parallelism
  8. Familiarity

Image Segmentation: How computer does it?

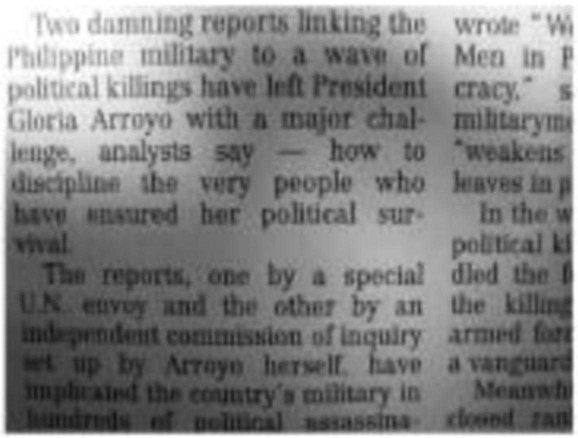
- Multiple ways to do it (edge based, contour based, color based, texture based, proximity based…)
  - Depends on the application requirement

- Lets start with simple two class segmentation problem
  - Separate pixels associated with object of interest from background
  - Often cast as a thresholding problem

Segmentation: Thresholding based approaches

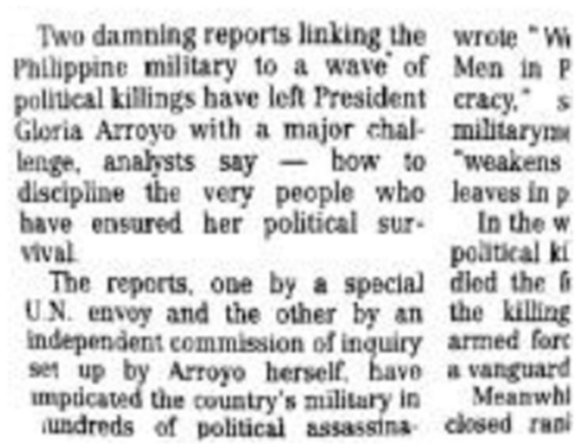
Two class Segmentation: Motivating example

- Separate pixels associated with object of interest from background



Two damning reports linking the Philippine military to a wave of political killings have left President Gloria Arroyo with a major challenge, analysts say — how to discipline the very people who have ensured her political survival.

The reports, one by a special U.N. envoy and the other by an independent commission of inquiry set up by Arroyo herself, have implicated the country's military in hundreds of political assassina-



Two damning reports linking the Philippine military to a wave of political killings have left President Gloria Arroyo with a major challenge, analysts say — how to discipline the very people who have ensured her political survival.

The reports, one by a special U.N. envoy and the other by an independent commission of inquiry set up by Arroyo herself, have implicated the country's military in hundreds of political assassina-

Thresholding

- Separate pixels associated with object of interest from background

- Given a image f(x,y), the segmented image g(x,y) is given by:

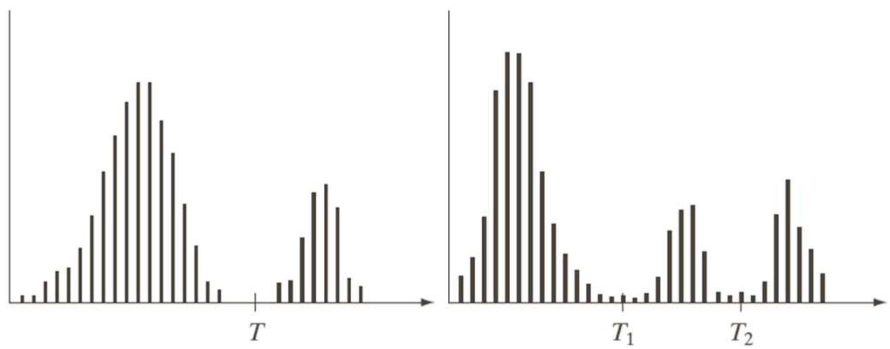
$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases}$$

If T is constant over the entire image → Global Thresholding
If T changes over the image → Variable Thresholding

The main question is: **How to find T?**

Thresholding

- How to find T?
- One Idea is to explore the intensity histograms (if there is clear separation)



a b

FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) > T_2 \\ b & \text{if } T_2 < f(x,y) \leq T_2 \\ c & \text{if } f(x,y) \leq T_1 \end{cases}$$

Thresholding: Role of Noise

- Clear separation?

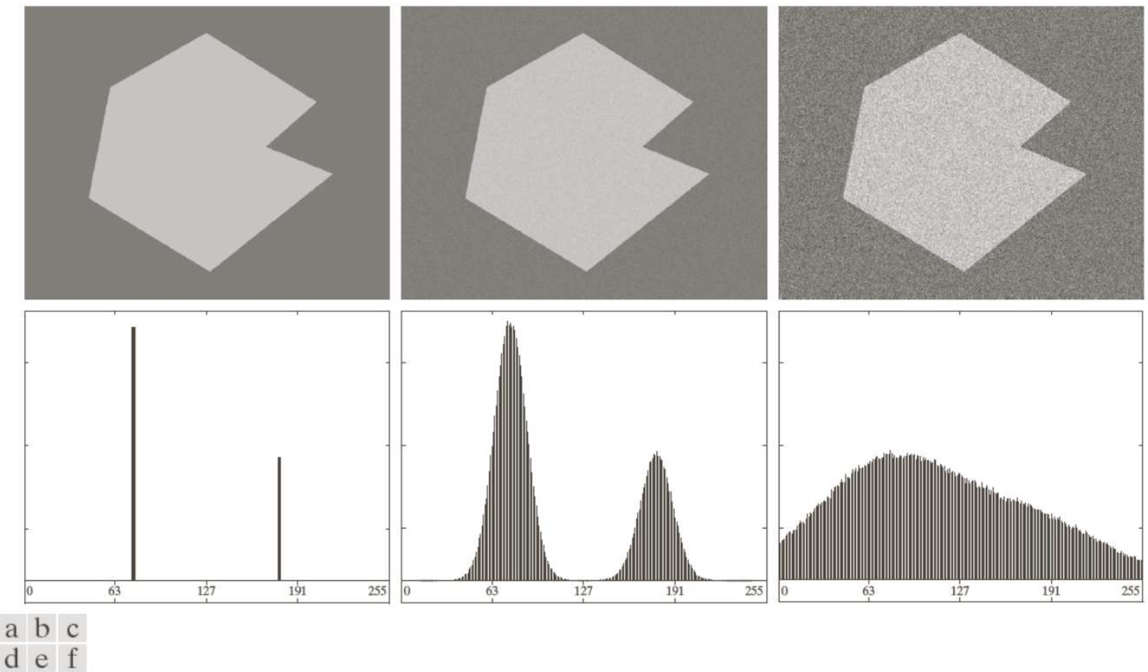


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Thresholding: Role of Illumination and Reflectance

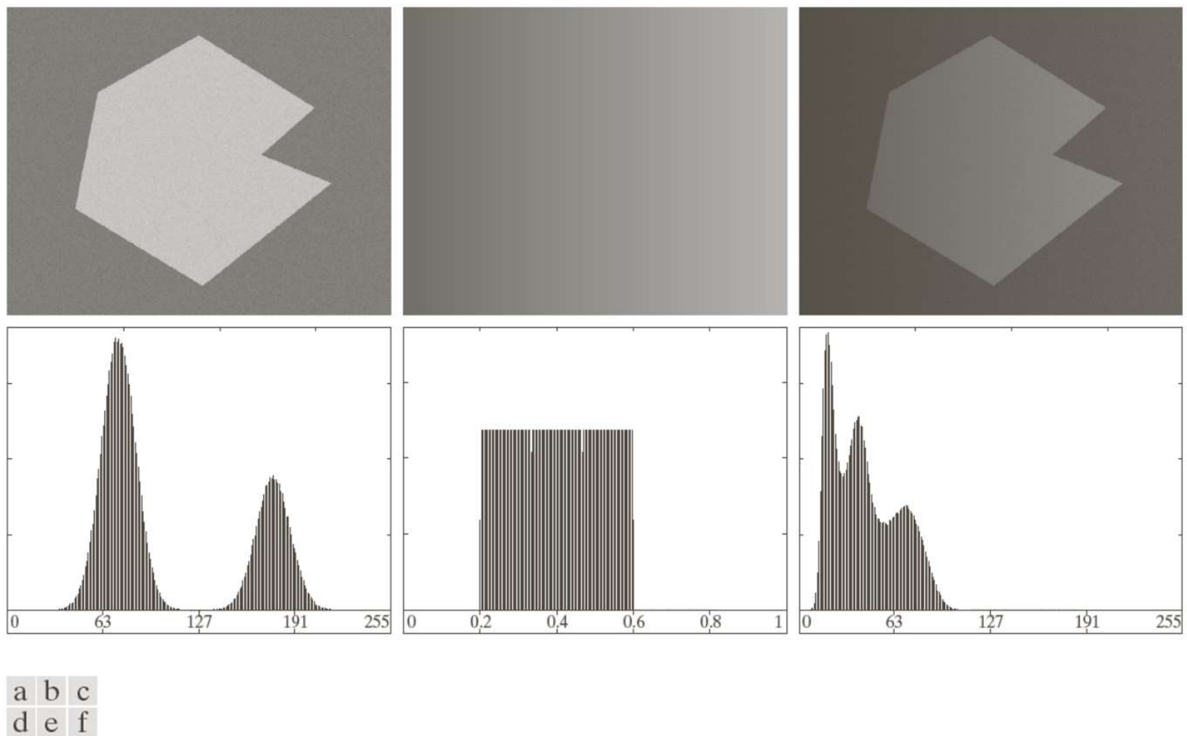


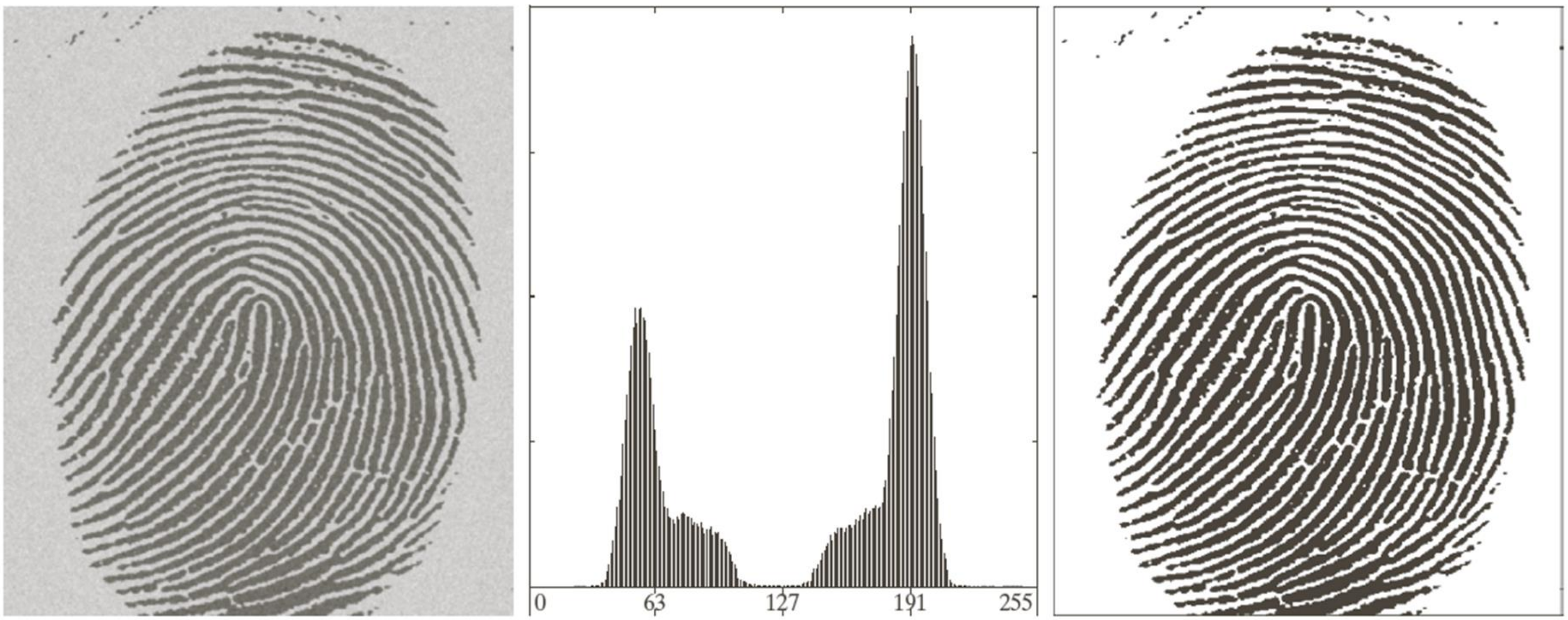
FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Finding T: Basic Global Thresholding

Iterative approach

1. Select an initial estimate of global threshold T
2. Segment the image using T, this will produce two groups of pixels (G1 and G2)
3. Compute the average (mean) intensity values m1 and m2 for the pixels in G1 and G2 respectively
4. Compute a new threshold value T= (m1+m2)/2
5. Repeat until convergence

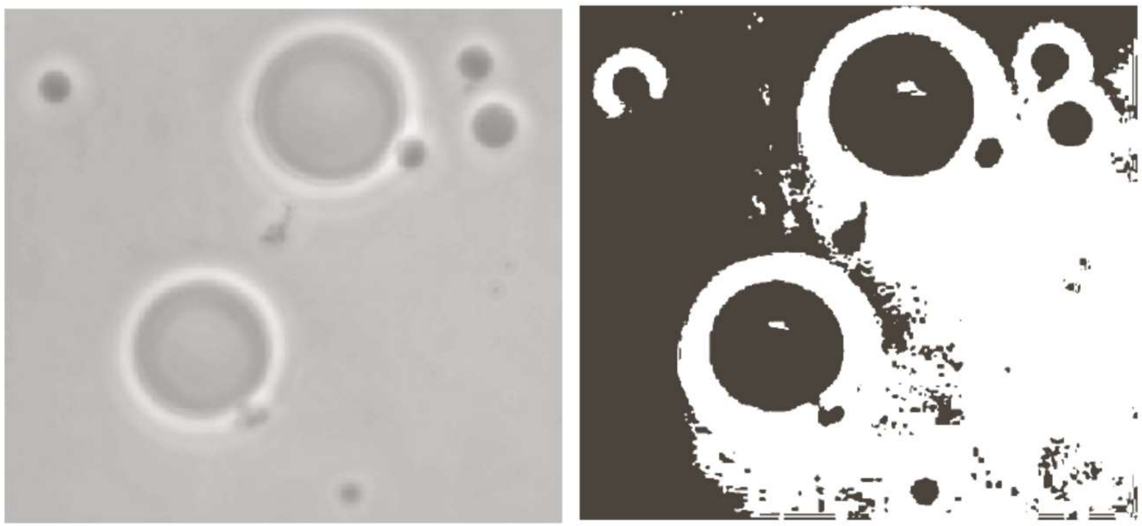
Basic Global Thresholding



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Basic Global Thresholding



Global Thresholding: Otsu's Method

- Based on histograms
- Automatically finds the optimal threshold maximizing the between class variance
- Proposed in 1975

Otsu's Method

- Compute the normalized histogram of the input image. Denote the components of the histogram by $p_i,\ i = 0, 1, 2, 3, \ldots, L-1$
- Suppose a threshold is selected $k, 0 < k < L-1$
- C_1 is the set of pixels with levels $[0, 1, 2, 3, \ldots, k]$
- C_2 is the set of pixels with levels $[k+1, k+2, k+3, \ldots, L-1]$

- Obtain the value of threshold which maximizes the between class variance

$$\sigma_B^2(k) = P_1(k)\,(m_1(k) - m_G)^2 + P_2(k)\,(m_2(k) - m_G)^2$$

Otsu's Method

- Compute the normalized histogram of the input image. Denote the components of the histogram by $p_i, \; i = 0, 1, 2, 3, \dots, L-1$
- Suppose a threshold is selected $k, 0 < k < L-1$
- C_1 is the set of pixels with levels $[0, 1, 2, 3, \dots, k]$
- C_2 is the set of pixels with levels $[k+1, k+2, k+3, \dots, L-1]$

- Obtain the value of threshold which maximizes the between class variance

$$\boxed{\sigma_B^2(k) = P_1(k)\,(m_1(k) - m_G)^2 + P_2(k)\,(m_2(k) - m_G)^2}$$

Otsu's Method

$$\boxed{\sigma_B^2(k) = P_1(k)\,(m_1(k) - m_G)^2 + P_2(k)\,(m_2(k) - m_G)^2}$$

- $P_1(k)$ is probability of C_1 occuring

$$P_1(k) = \sum_{i=0}^{k} p_i \;, k = 0,1,2, \dots , k$$

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i \; = 1 - P_1(k)\,, k = 0,1,2, \dots , k$$

- $m_1(k)$ and $m_2(k)$ are means of C_1 and C_2 and m_G is the global mean

$$m_1(k) = \frac{\sum_{i=0}^{k} i\, p_i}{P_1(k)} \qquad m_2(k) = \frac{\sum_{i=k+1}^{L-1} i\, p_i}{P_2(k)} \qquad m_G = \sum_{i=0}^{L-1} i\, p_i$$

Otsu's Method

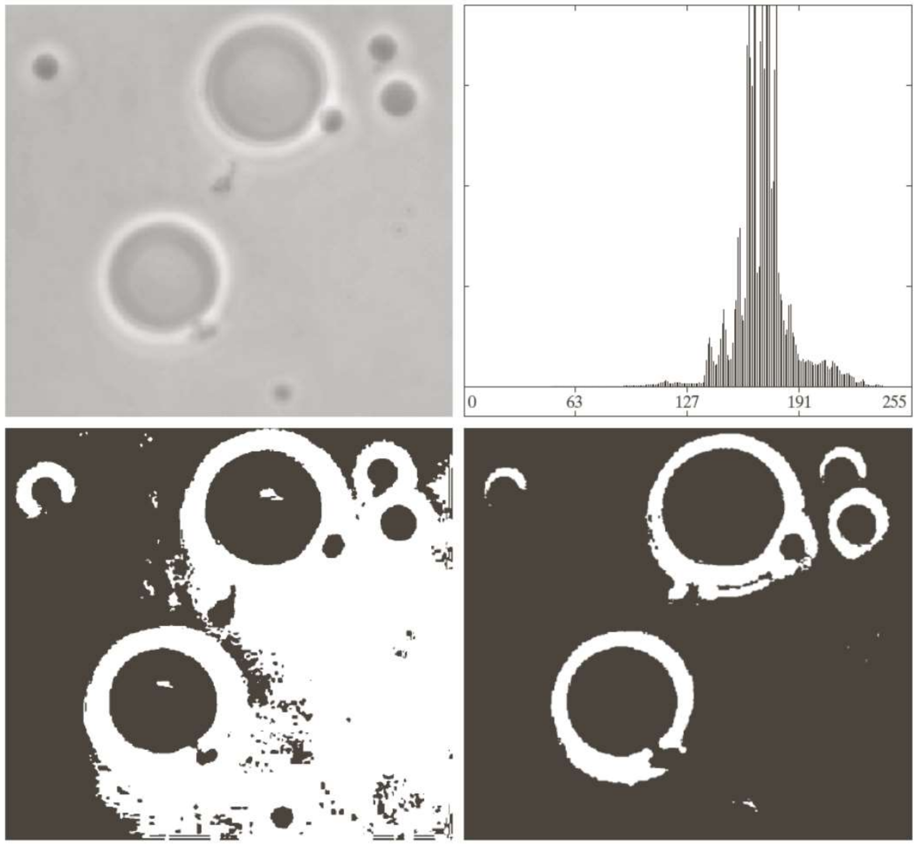
$$\boxed{\sigma_B^2(k) = P_1(k)\,(m_1(k) - m_G)^2 + P_2(k)\,(m_2(k) - m_G)^2}$$

$$\boxed{\sigma_B^2(k^*) = \max_{0 \le k \le L-1} \sigma_B^2(k)}$$

In simple words, we evaluate all values of k and select the value of k that yielded the maximum $\sigma_B^2(k)$

This idea can be easily extended to compute multiple thresholds!

Otsu's Method



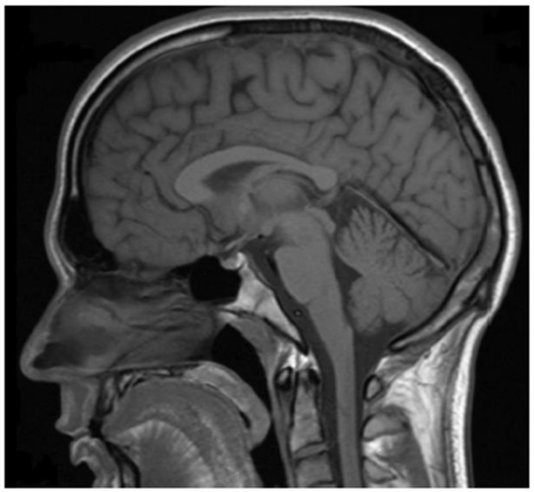
a b
c d

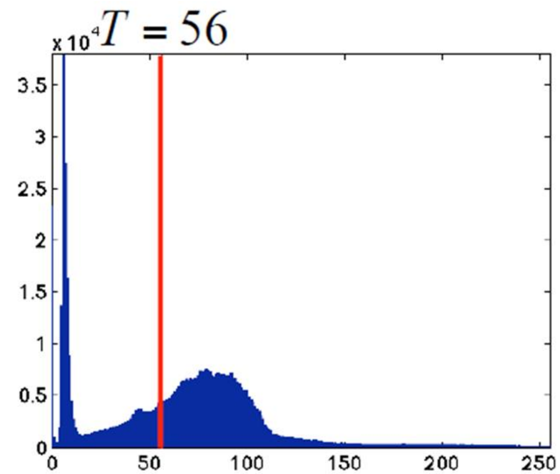
FIGURE 10.39 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

T = 169

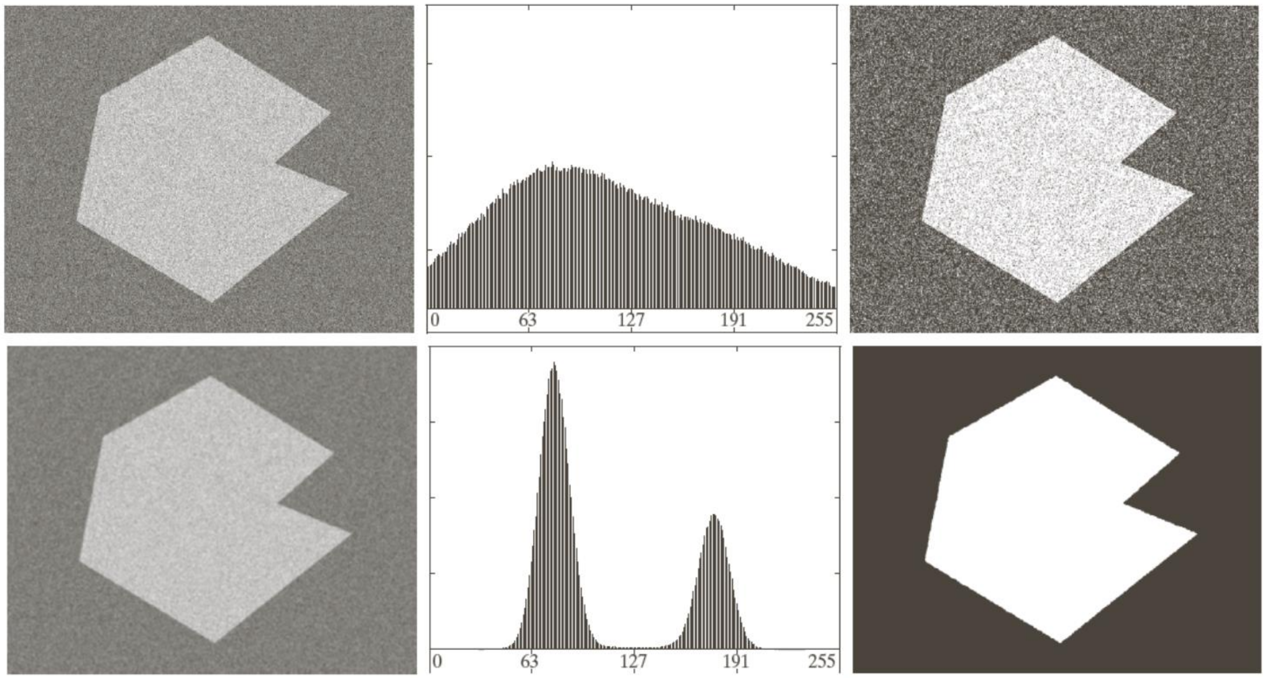
T = 181

Otsu's Method





Handling Noise



a b c
d e f

FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Otsu’s method: Main Limitation

Pai et al. PR 2010

No single threshold, may be ideal

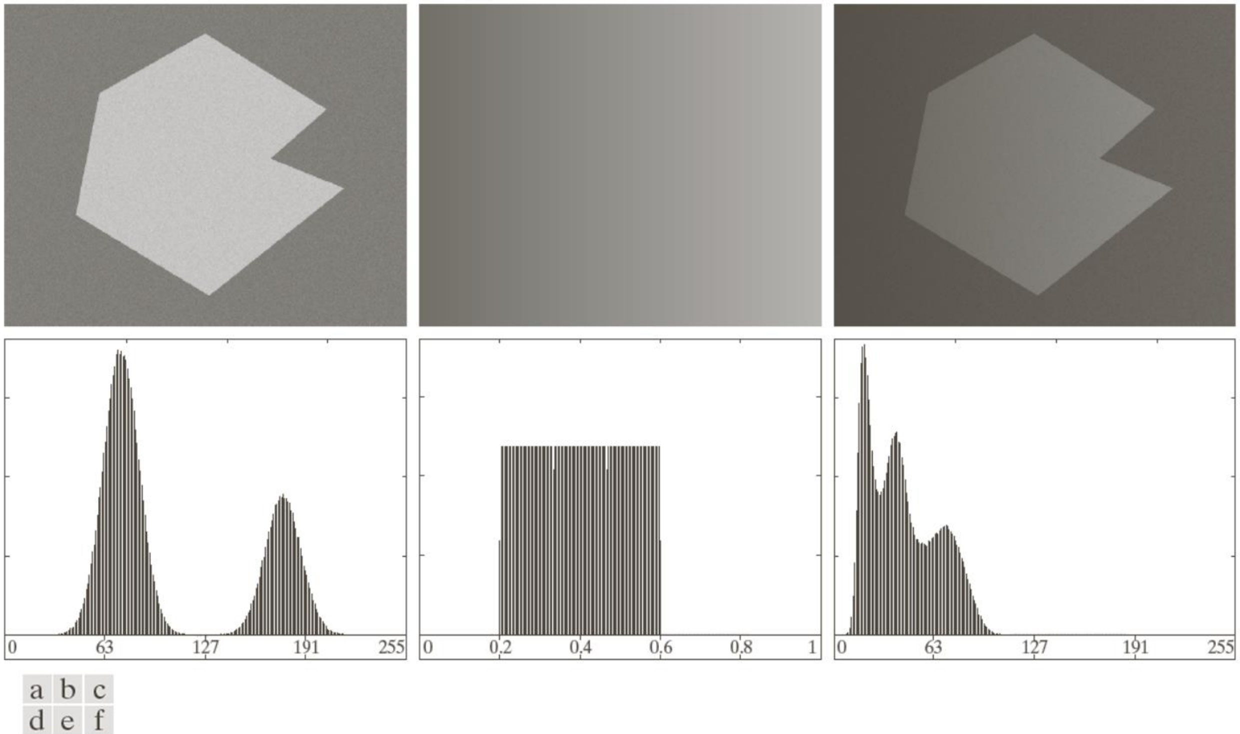
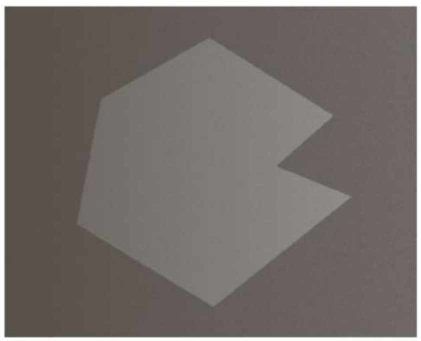
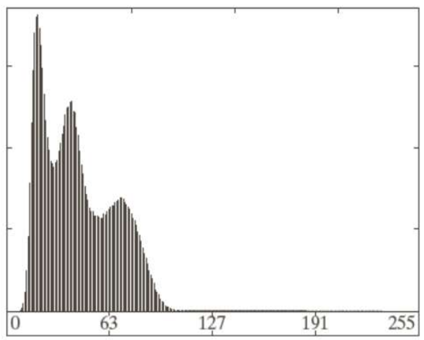
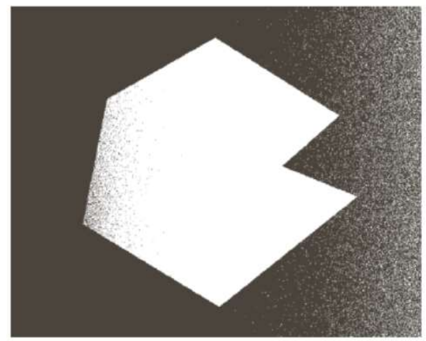


FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

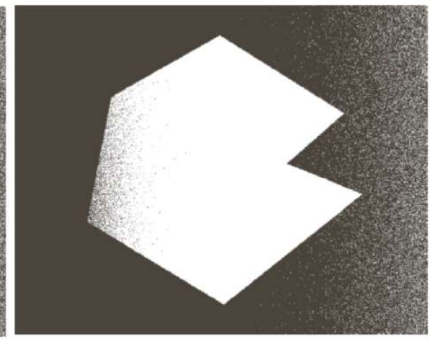
Global segmentation: main limitation





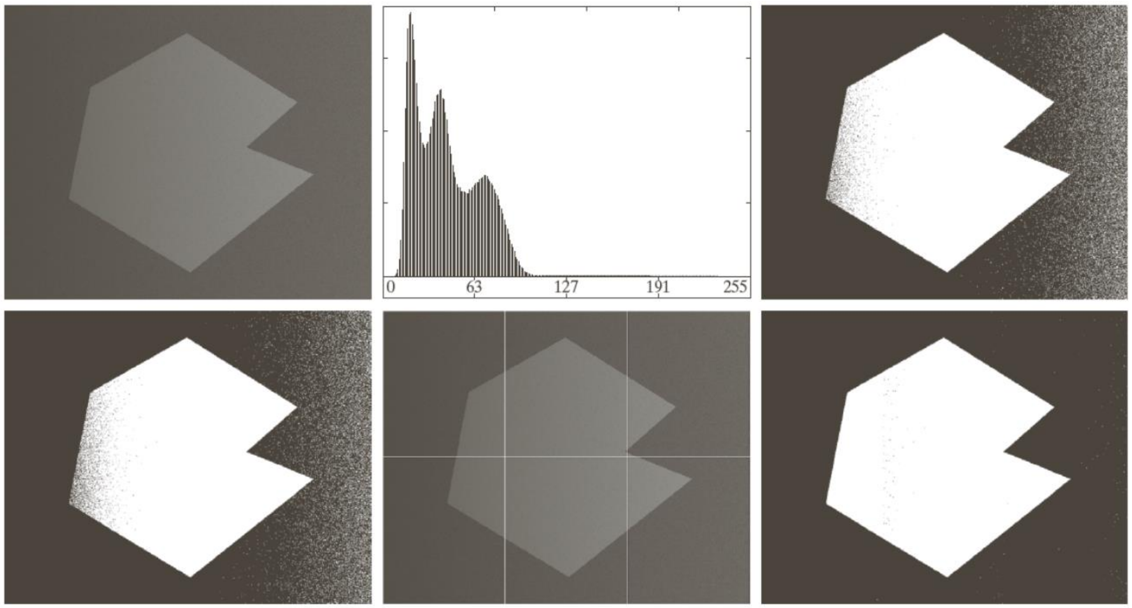


Iterative Global Threshold



Otsu's method

Image subdivision + variable Thresholding



a b c
d e f

FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

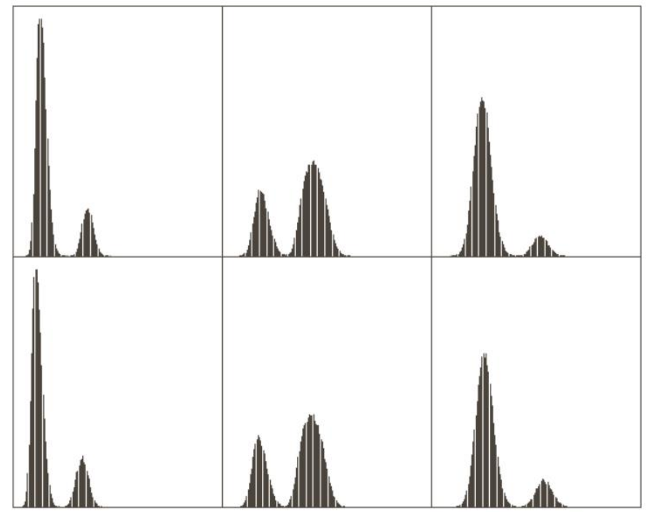


FIGURE 10.47 Histograms of the six subimages in Fig. 10.46(e).

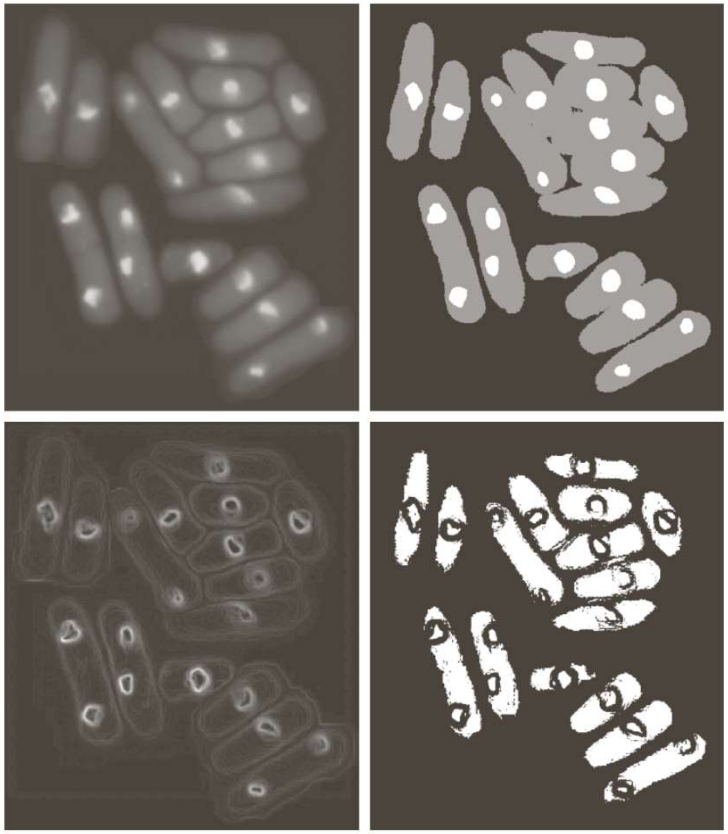
Per pixel variable Thresholding

- Compute standard deviation and mean of each pixel (around local neighbourhood)
- Let σ_{xy} , m_{xy} denote the standard deviation and mean value contained in neighbourhood S_{xy} centred around (x, y)
- Example threshold function:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T_{xy} \\ 0 & \text{if } f(x, y) \leq T_{xy} \end{cases}$$

$$T_{xy} = a\sigma_{xy} + bm_{xy} \quad \text{or} \quad T_{xy} = a\sigma_{xy} + bm_G$$

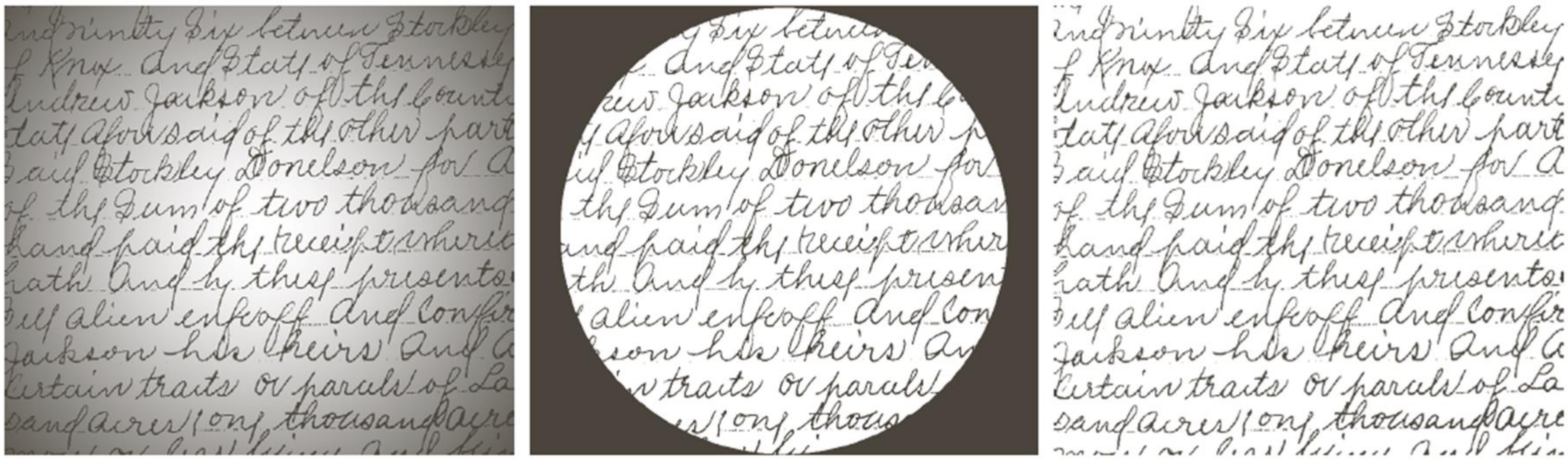
Per pixel variable Thresholding



a b
c d

FIGURE 10.48 (a) Image from Fig. 10.43. (b) Image segmented using the dual thresholding approach discussed in Section 10.3.6. (c) Image of local standard deviations. (d) Result obtained using local thresholding.

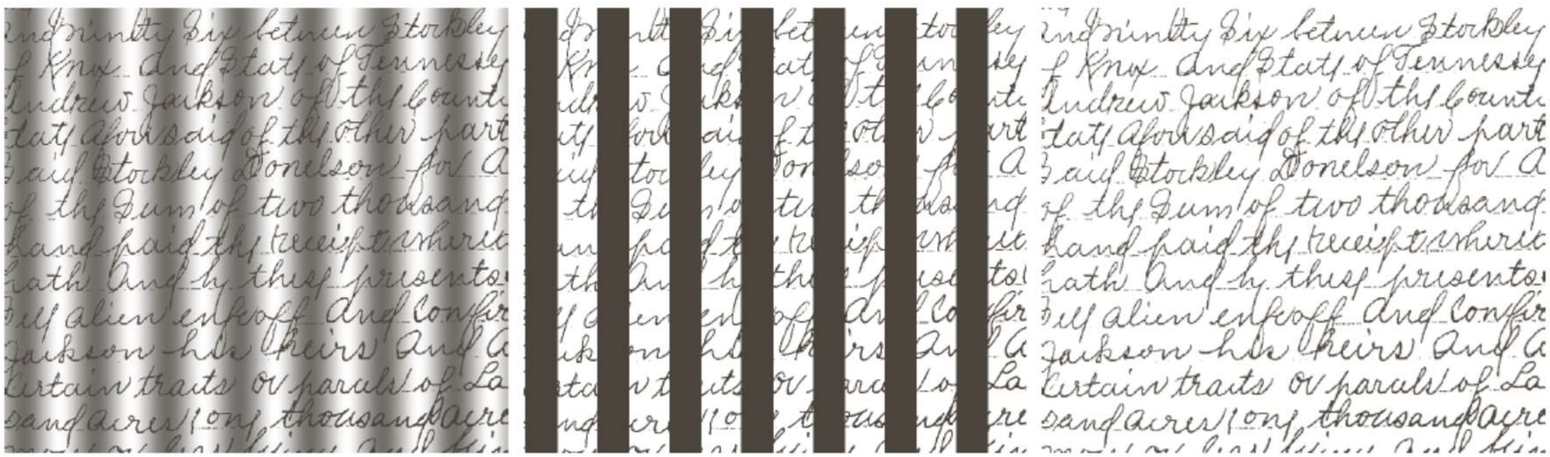
Per pixel: moving average



a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

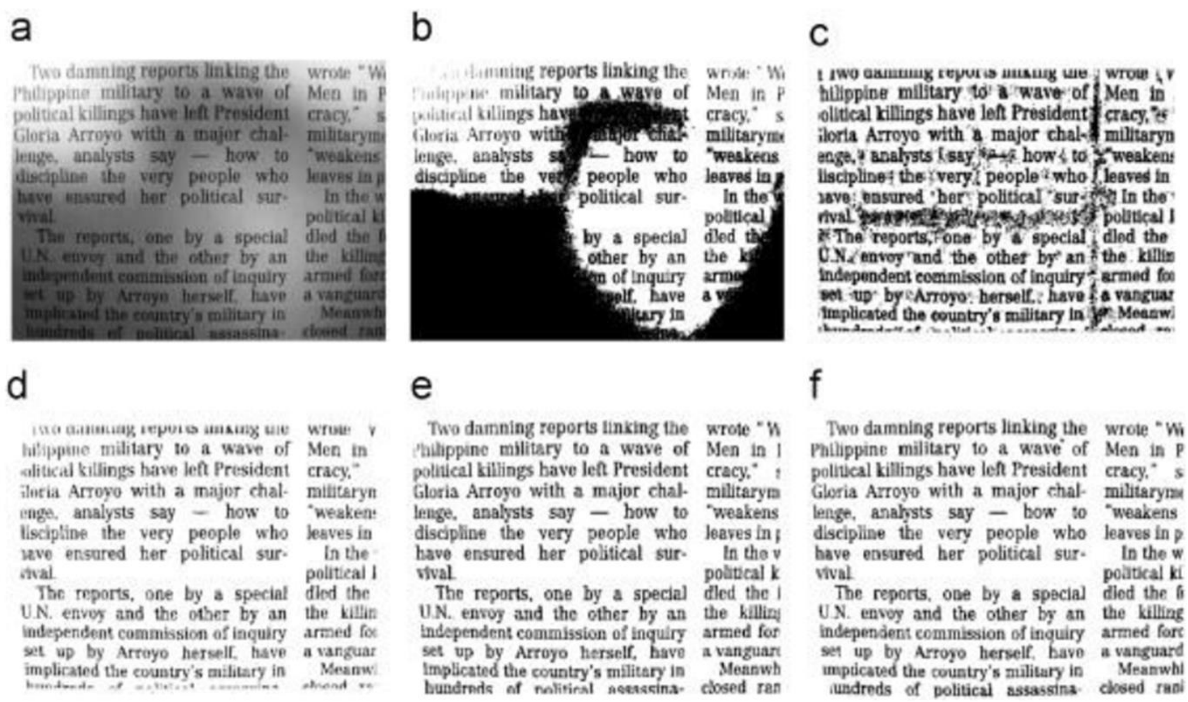
Per pixel: moving average



a b c

FIGURE 10.50 (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

Adaptive thresholding: advanced algorithms



Adaptive thresholding algorithm... Pai et al. PR 2010

Thresholding: Summary

- Many methods

- Survey

Sezgin, M and Sankur, B (2004), "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation", Journal of Electronic Imaging 13(1): 146-165

- Comparison

http://www.fmwconcepts.com/imagemagick/threshold_comparison/index.php

Segmentation: Region based approaches

Region Based Segmentation

- Basic Formulation: Let R represent the entire image region. Segmentation is the process of partitioning R into subregions $R_1, R_2, \ldots, R_n$, such that:

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

2. R_i is a connected region, for all i

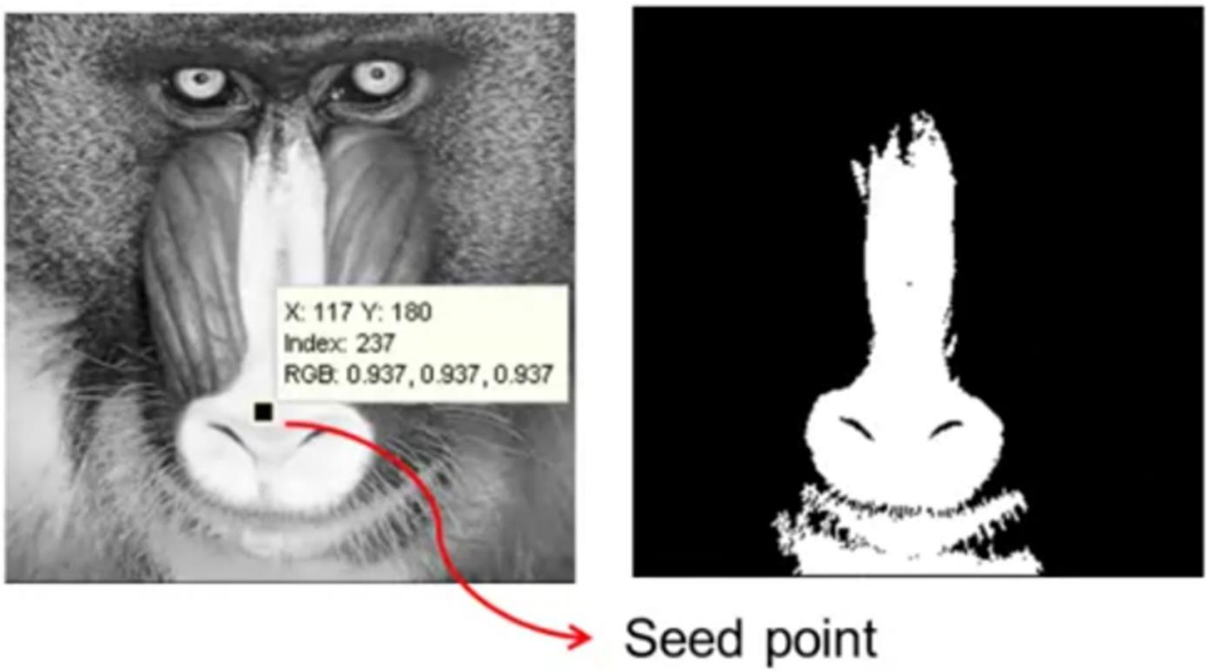
3. $R_i \cap R_j = \emptyset$ for all i and j, i≠j

Region Growing

- Start with a set of seed points
  - Based on prior information
  - Based on some properties calculated at each point

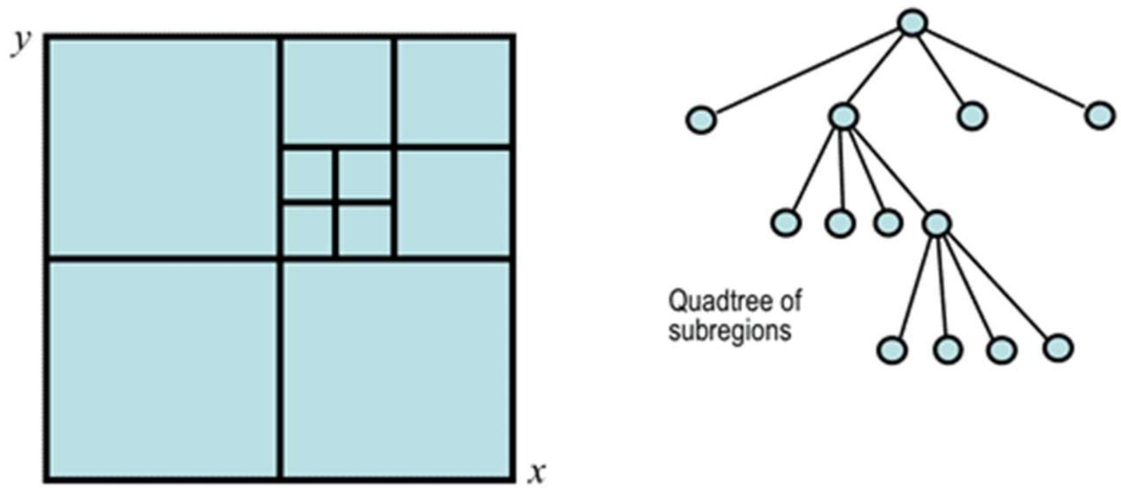
- Grow regions based on a predefined criteria
  - Similarities in color, texture etc.

Region Growing: Example



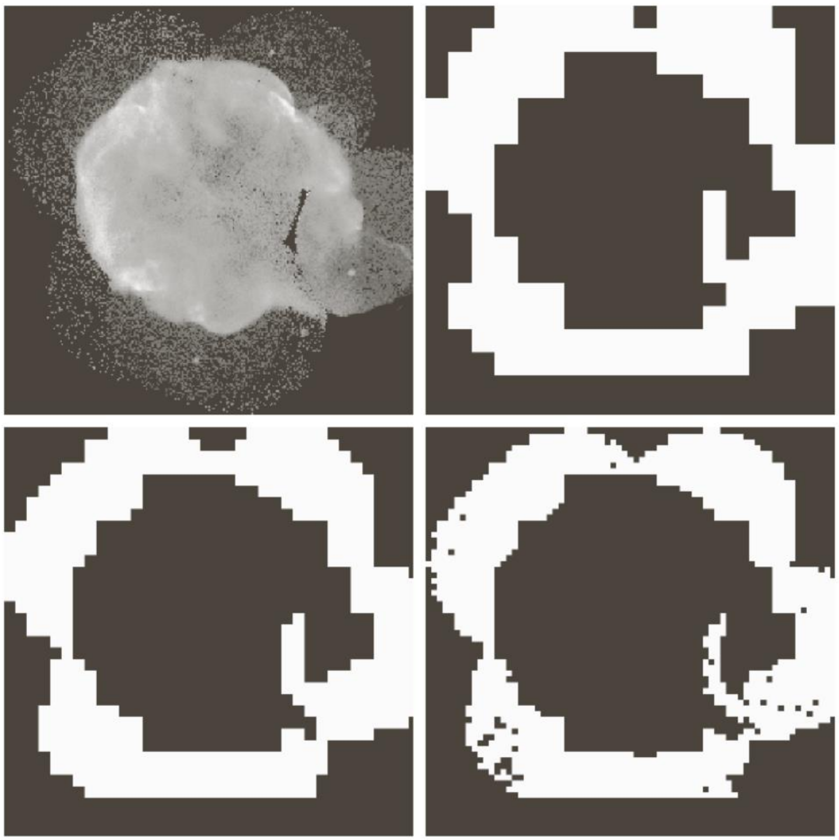
Source: Aggelos K. Katsaggelos

Region Splitting and Merging



Need to define a splitting function and size of the minimum quadrant

Region Splitting and Merging



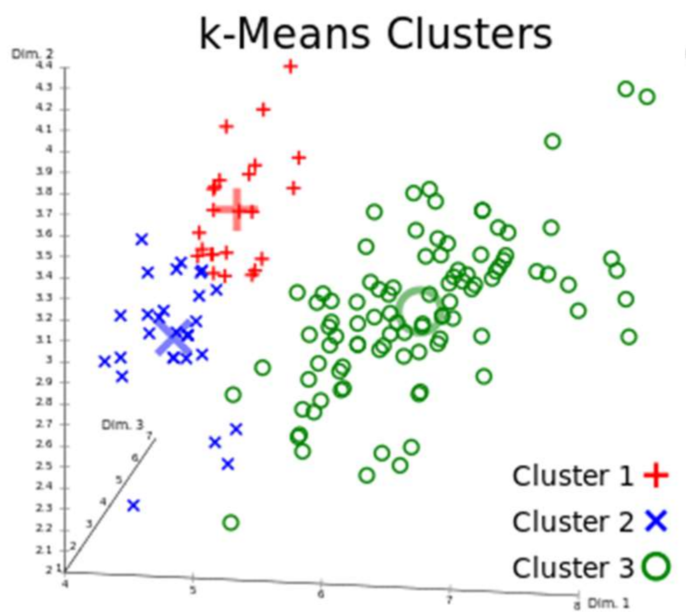
$$Q = \begin{cases} \text{TRUE} & \text{if } \sigma > a \text{ AND } 0 < m < b \\ \text{FALSE} & otherwise \end{cases}$$

a b
c d

FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

Clustering

- Clustering using extracted features + spatial coordinates
  - Features like color, texture etc.
  - Algorithms like k-means, agglomerative clustering, spectral clustering etc.

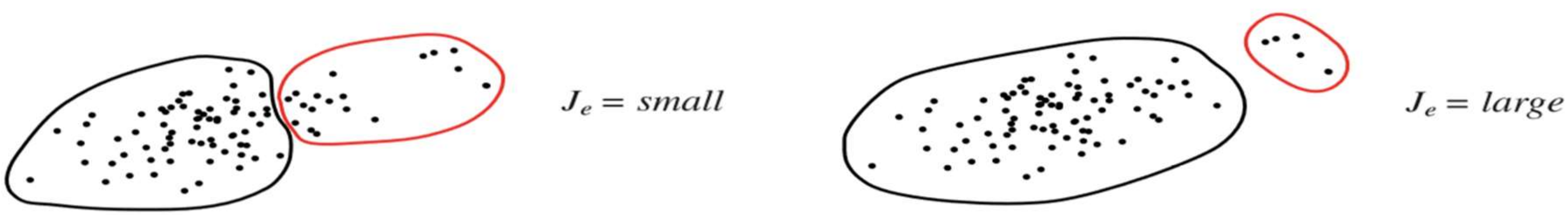


Criterion Functions for Clustering

- The Sum-of-Squared-Error Criterion:
  - Achieves minimum variance clustering

$$J_e = \sum_{i=1}^{c} \sum_{\mathbf{x} \in \mathcal{D}_i} \|\mathbf{x} - \mathbf{m}_i\|^2. \qquad \mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}.$$

  - Not always best criterion



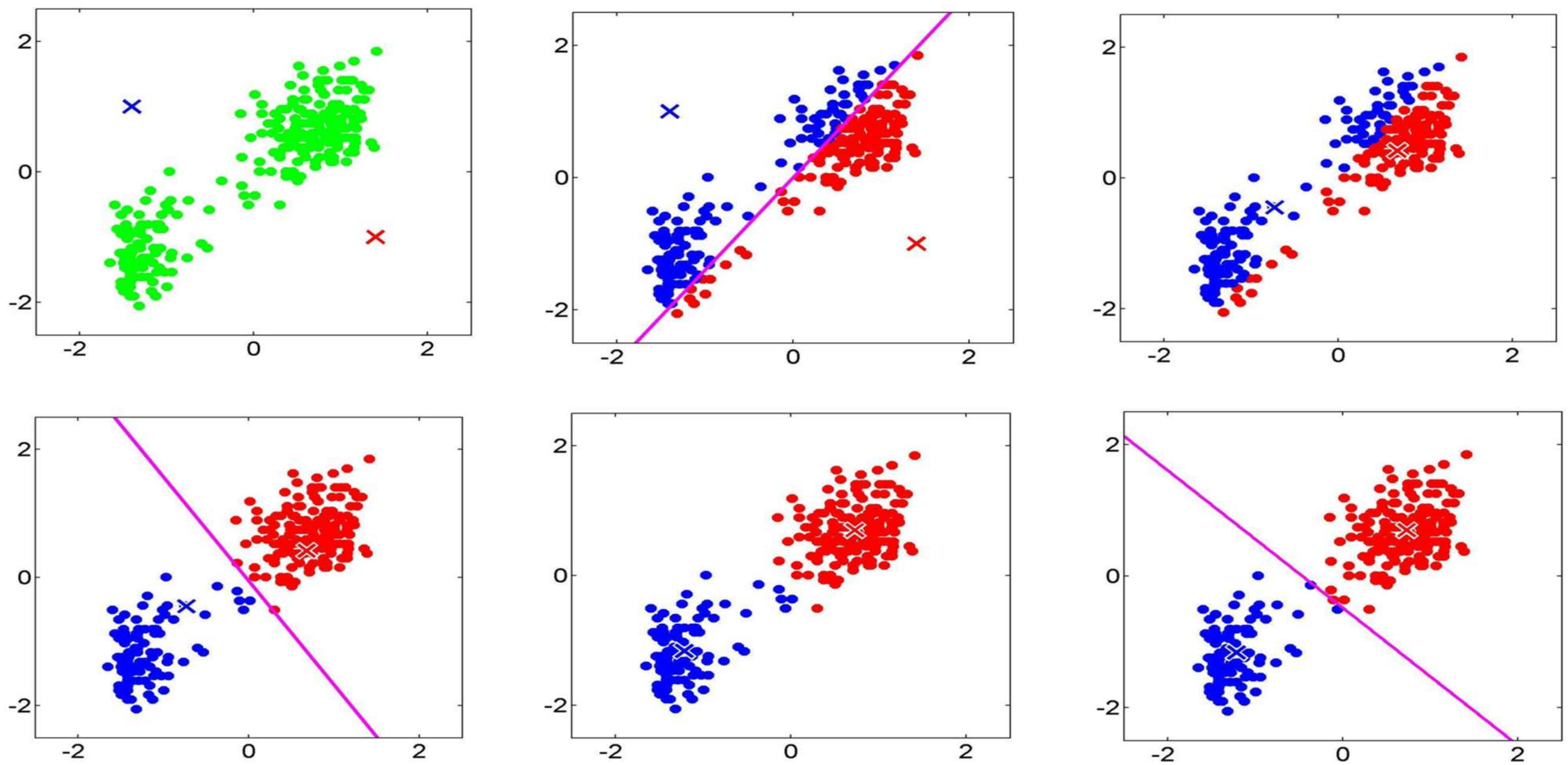
Kmeans Clustering

- Goal is to represent a data set in terms of K clusters using the respective cluster means

- Initialize means randomly
- Iterate between two phases:
  - Assign each data point to nearest mean
  - Update cluster means

- Simplest version is based on Euclidean distance

Kmeans Clustering



Kmeans Clustering

Minimize $J = \sum_{i=1}^{N} \sum_{j=1}^{K} a_{ij} \left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$

such that $a_{ij} \epsilon \{0,1\}$ and $\sum_{j=1}^{K} a_{ij} = 1$

- Initialize the K mean-vector $\boldsymbol{\mu}_j$ randomly (e.g., choosing any K data points as the mean vectors)

- E-step: minimize J w.r.t. a_{ij}
  - Set $a_{ij} = 1$ for cluster index j corresponding to the smallest $\left\| \mathbf{x}_i - \boldsymbol{\mu}_j \right\|^2$ i.e., closes cluster mean (centroid)

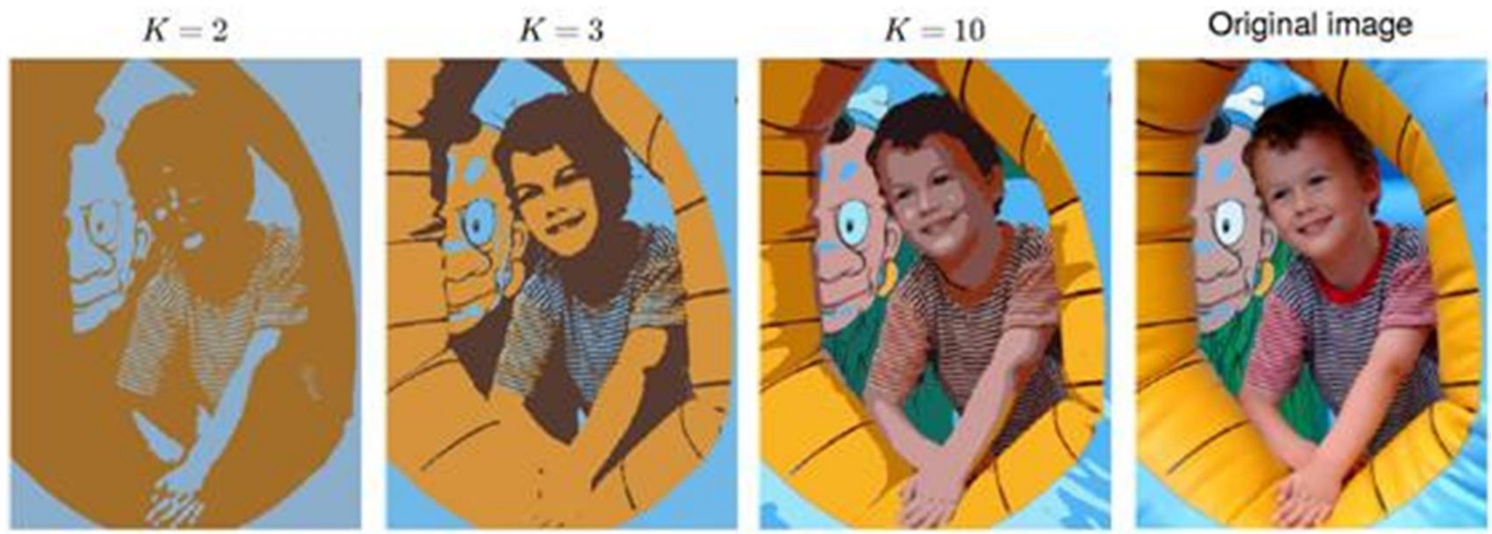
- M-step: minimize J w.r.t. $\boldsymbol{\mu}_j$
  - Set $\frac{\partial J}{\partial \boldsymbol{\mu}_j} = 0 \Rightarrow \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} a_{ij} \mathbf{x}_i}{\sum_{i=1}^{N} a_{ij}}$ i.e., re-computing the mean.

Limitations of Kmeans Clustering

- Convergence to local minima –sensitive to initialization.
- Applicable to data when mean is defined.
- Sensitive to outliers and data noise
- Suitable only to cases when clusters are convex shapes.
- Number of clusters (K) needs to be explicitly set.
- Computational complexity : $O(NKdL)$.

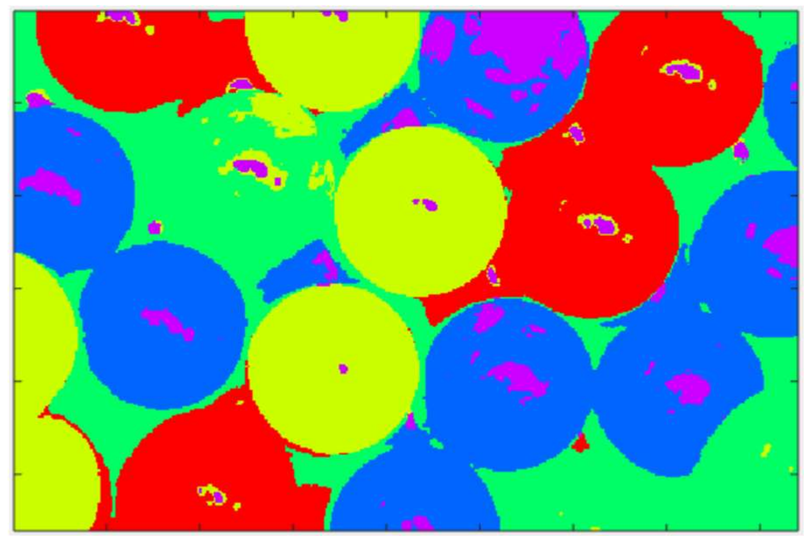
Clustering

- Clustering using extracted features + spatial coordinates
  - Features like color, texture etc.
  - Algorithms like k-means, agglomerative clustering, spectral clustering etc.



Clustering

- Example: kmeans (only color)



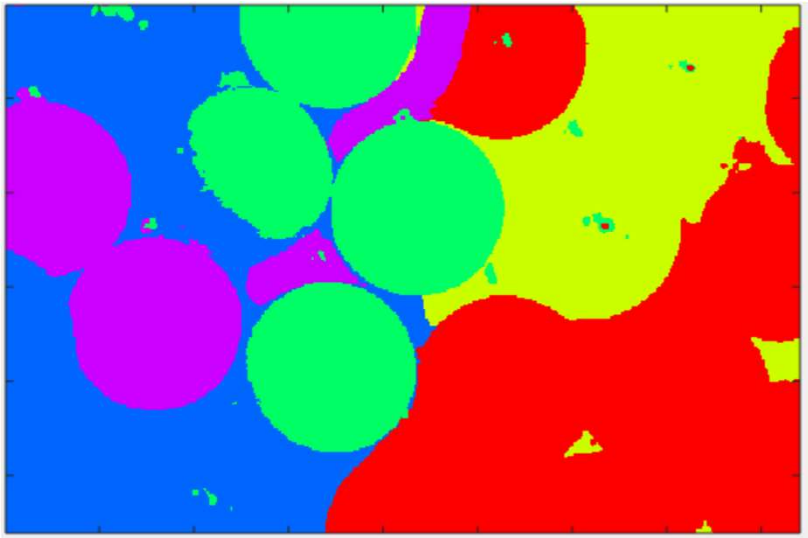
k=5

Clustering

```
im = imread('color_balls.jpg');
ab = double(im(:,:,1:3));
nrows = size(ab,1); ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,3);

nColors = 5;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', 'Replicates',3);

pixel_labels = reshape(cluster_idx,nrows,ncols);

imshow(pixel_labels,[]), title('image labeled by cluster index');
map = colormap(hsv(5)); image((pixel_labels)) colormap(map);
```

Clustering

- Example: kmeans (spatially constrained)

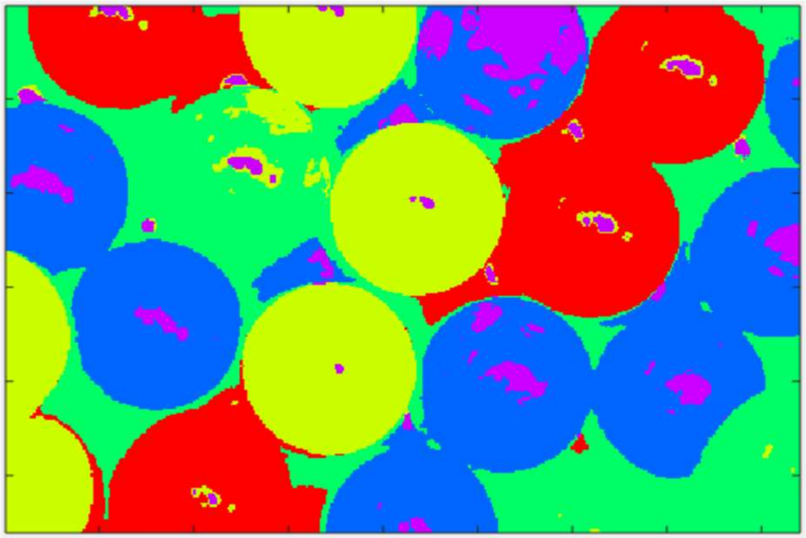


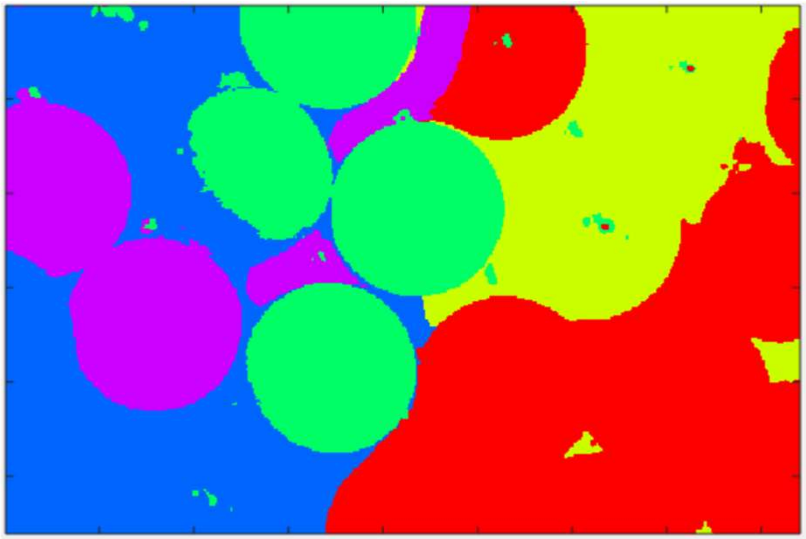
k=5

Clustering

```
im = imread('color_balls.jpg');
 ab = double(im(:,:,1:3));
ab(:,:,4) = repmat([1:size(im,2)],size(im,1),1);
ab(:,:,5) = repmat([1:size(im,1)]',1,size(im,2));
nrows = size(ab,1); ncols = size(ab,5);
ab = reshape(ab,nrows*ncols,5);

nColors = 5;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', 'Replicates',3);

pixel_labels = reshape(cluster_idx,nrows,ncols);

imshow(pixel_labels,[]), title('image labeled by cluster index');
map = colormap(hsv(5)); image((pixel_labels)) colormap(map);
```

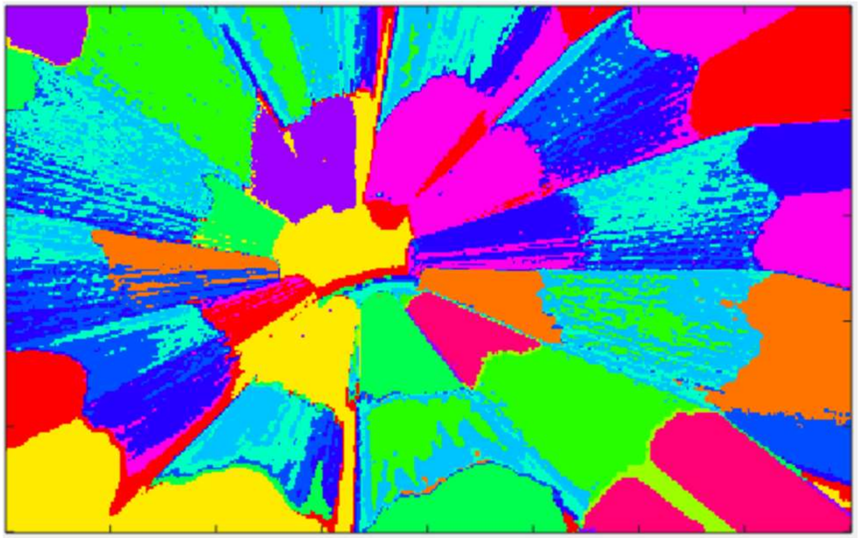
Clustering





Clustering

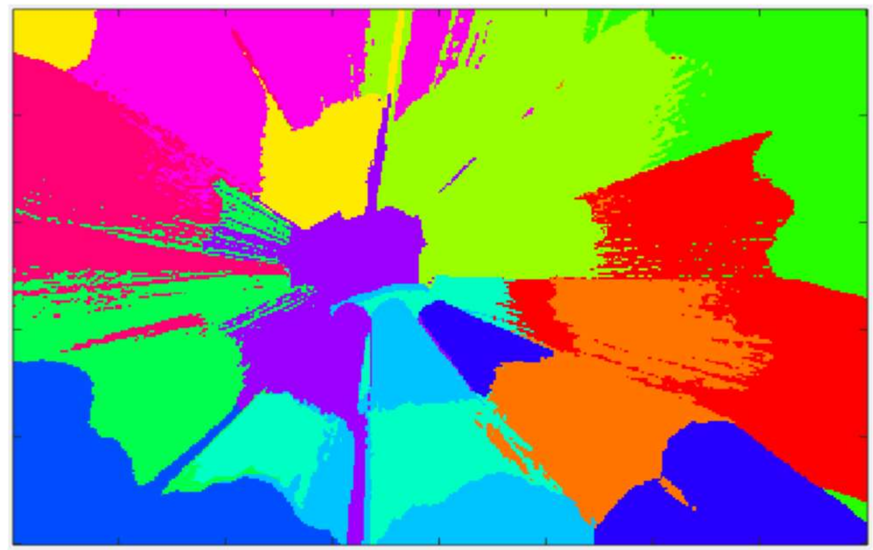
- Examples: kmeans



k=13

Clustering

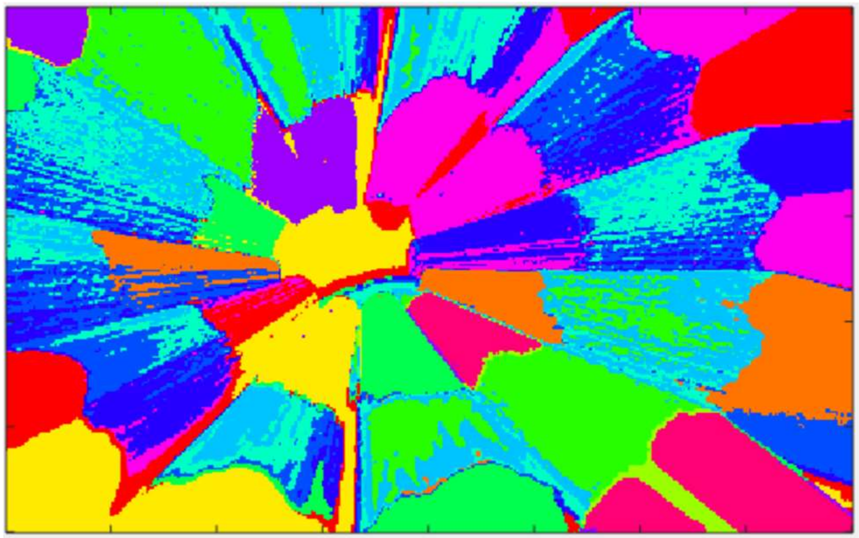
- Example: kmeans (spatially constrained)

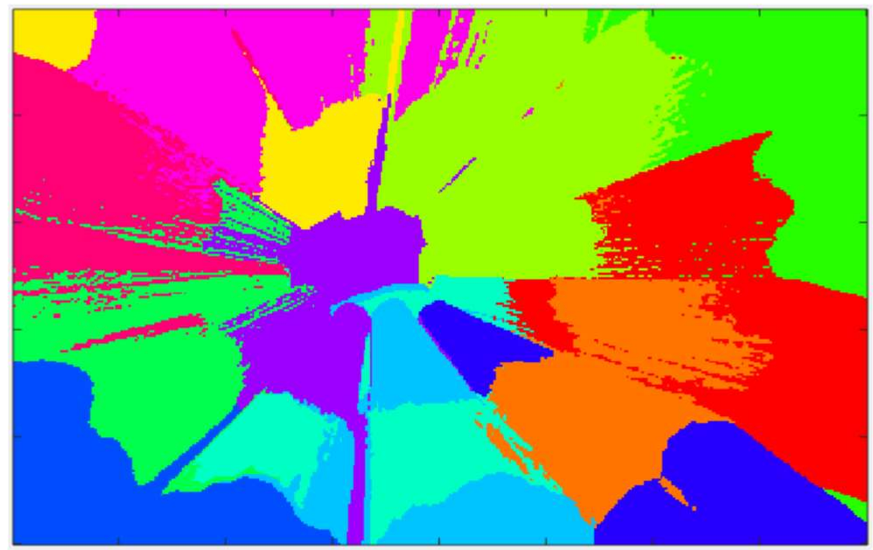


k=13

Clustering

- Example: kmeans (spatially constrained)

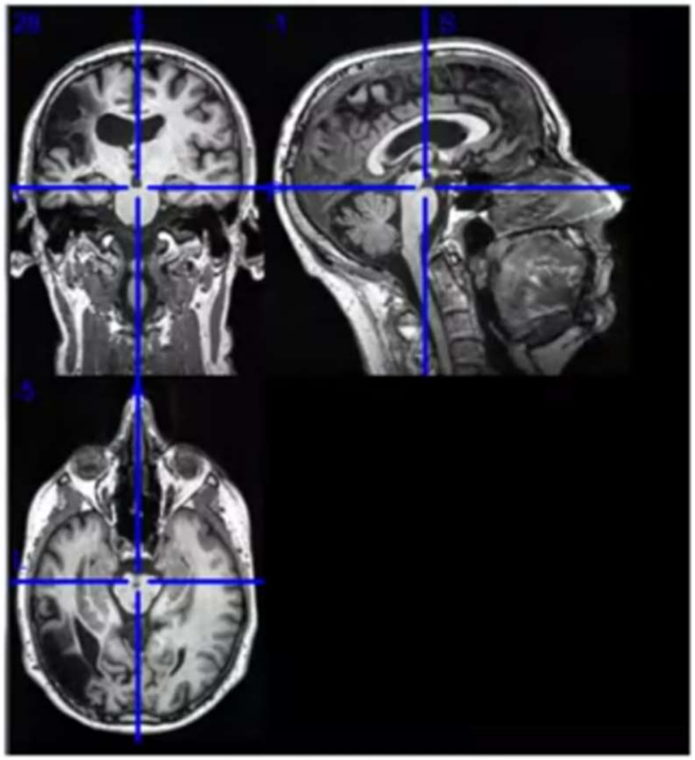




Clustering

- Clustering using extracted features + spatial coordinates
  - Features like color, texture etc.
  - Algorithms like k-means, agglomerative clustering, spectral clustering etc.

Clustering-Medical Images



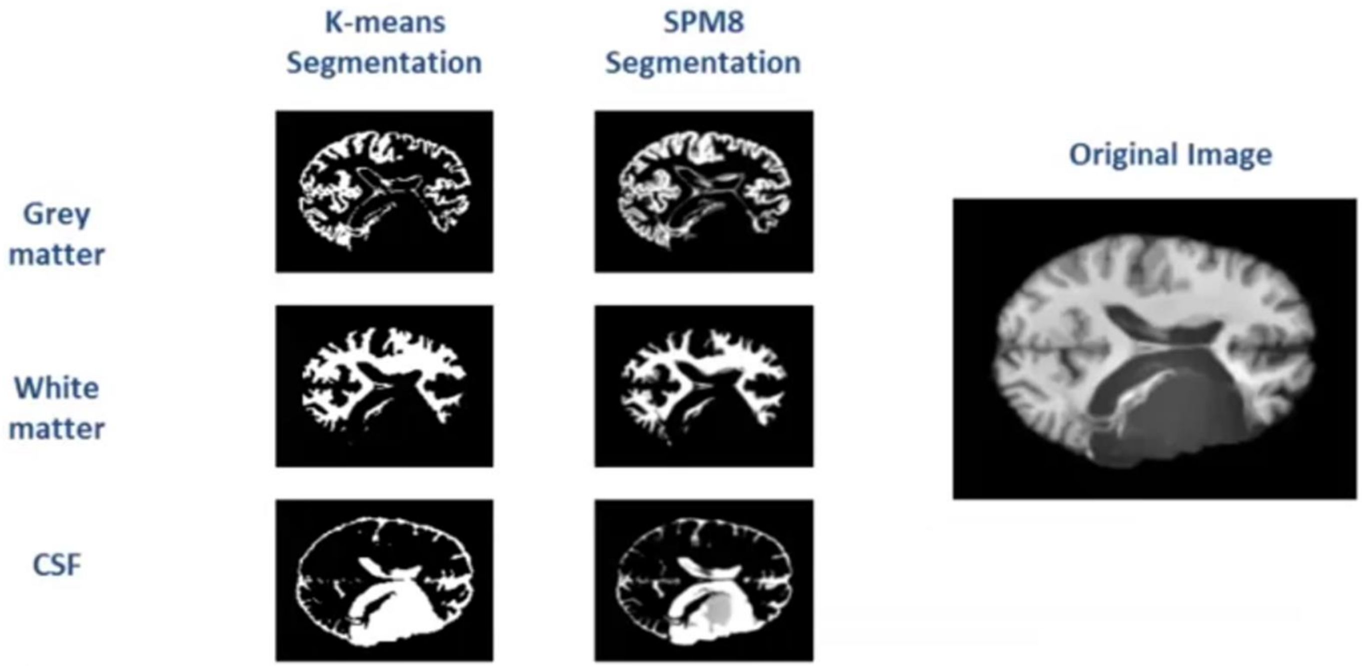
Voxels i.e., 3D pixels

Clustering in 4 dimensional space i.e
[x, y, z, I(x,y,z)]

There are three groups: White matter, Gray matter and CSF

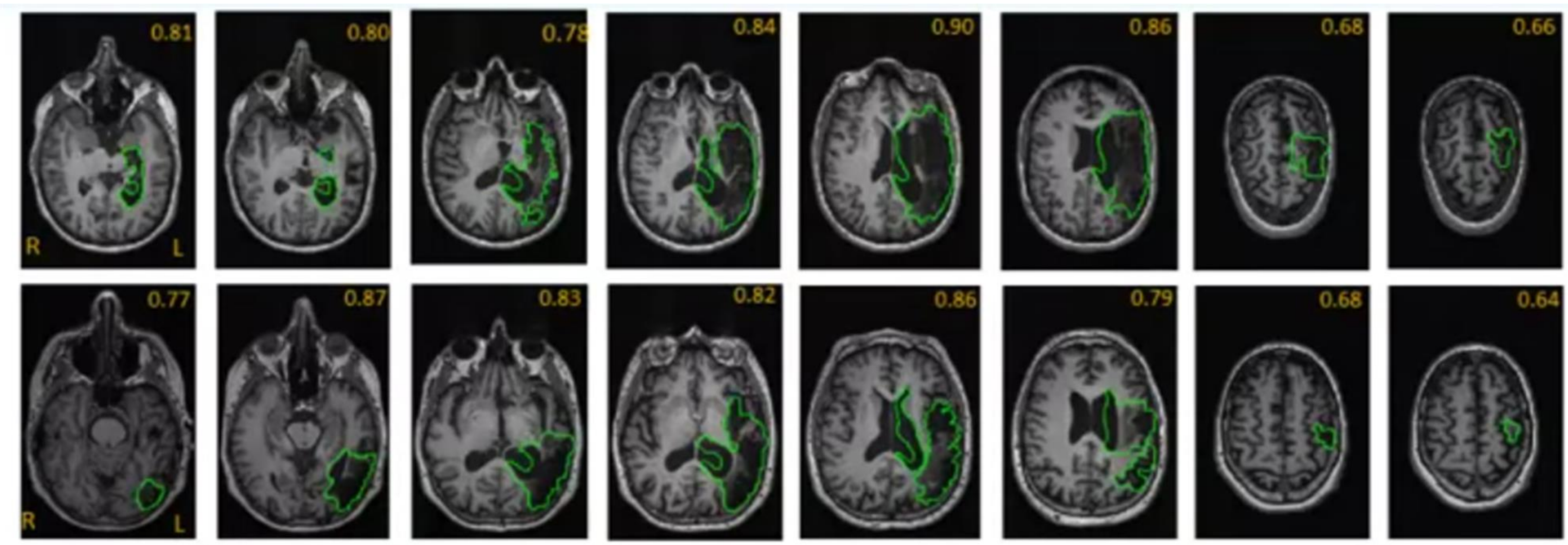
Source: Aggelos K. Katsaggelos

Clustering-Medical Images



Source: Aggelos K. Katsaggelos

Clustering-Medical Images



Important both for diagnosis + analysis of recovery

Source: Aggelos K. Katsaggelos

Advanced algorithms

- Mean Shift, Graph cut etc.
- Will be covered in computer vision/SMAI course