Lecture 03: Image Segmentation

Instructor: Dr. Hossam Zawbaa

Introduction

- The purpose of image segmentation is to partition an image into **meaningful regions** with respect to a particular application.
- The segmentation is based on measurements taken from the image and might be *grey-level*, *colour*, *texture*, *depth,* and *motion.*

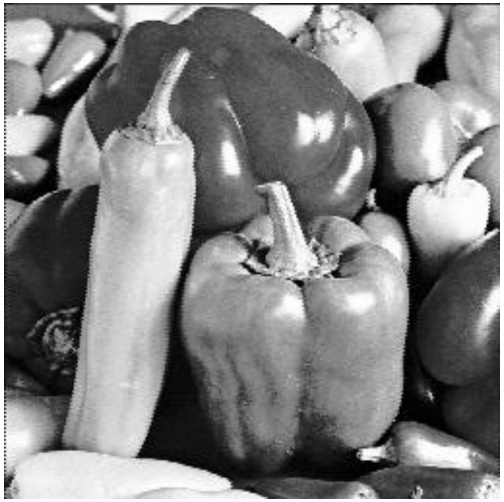
Introduction

- Usually image segmentation is an **initial and vital step** in a series of processes aimed at overall image understanding.
- Applications of image segmentation include
  - Identifying objects in a scene for object-based measurements such as size and shape.
  - Identifying objects in a moving scene for *object-based video compression (MPEG4).*
  - Identifying objects which are at different distances from a sensor using depth measurements from a laser range finder enabling path planning for a mobile robots.

Introduction

- **Segmentation based on greyscale**

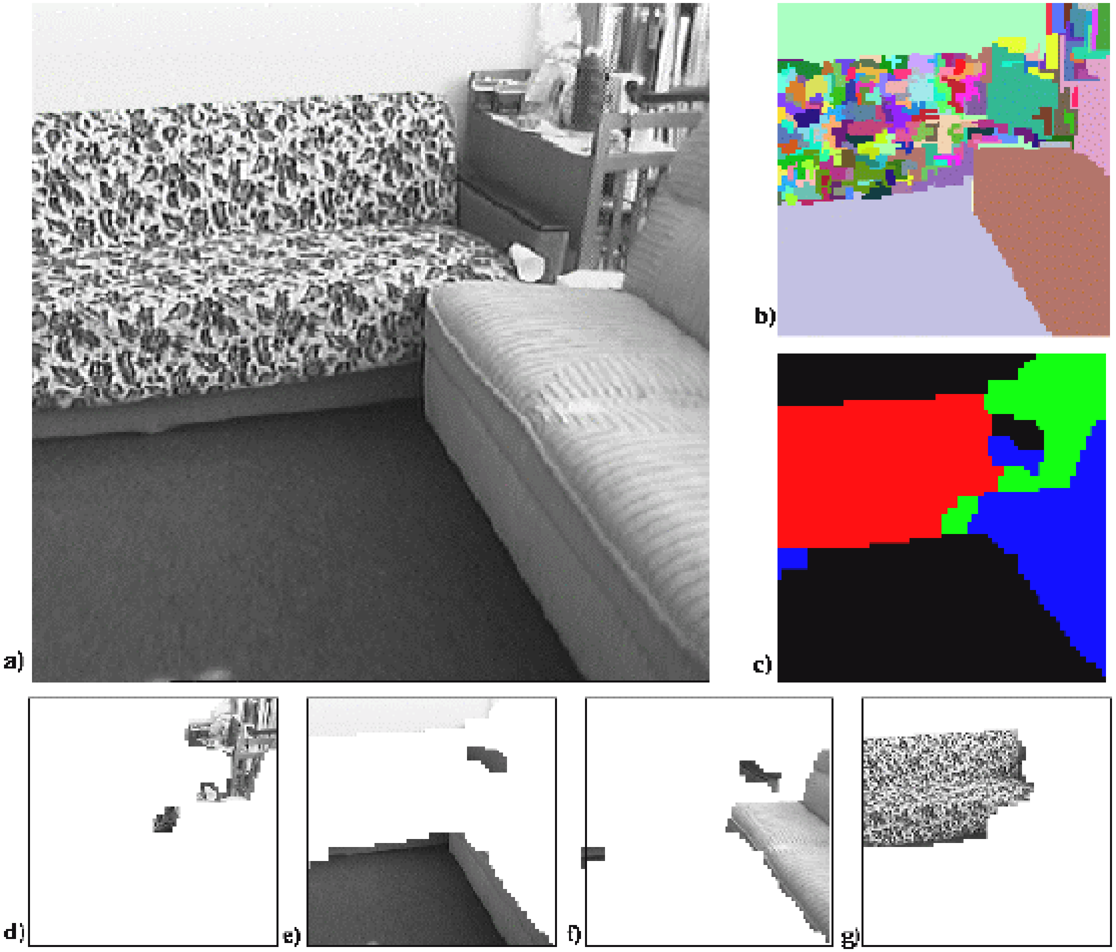
Very simple ‘model’ of greyscale leads to inaccuracies in object labelling



Introduction

- **Segmentation based on texture**

Enables object surfaces with varying patterns of grey to be segmented



Introduction

- **Segmentation based on motion**

The main difficulty of motion segmentation is that an intermediate step is required to (either implicitly or explicitly) estimate an *optical flow field.*

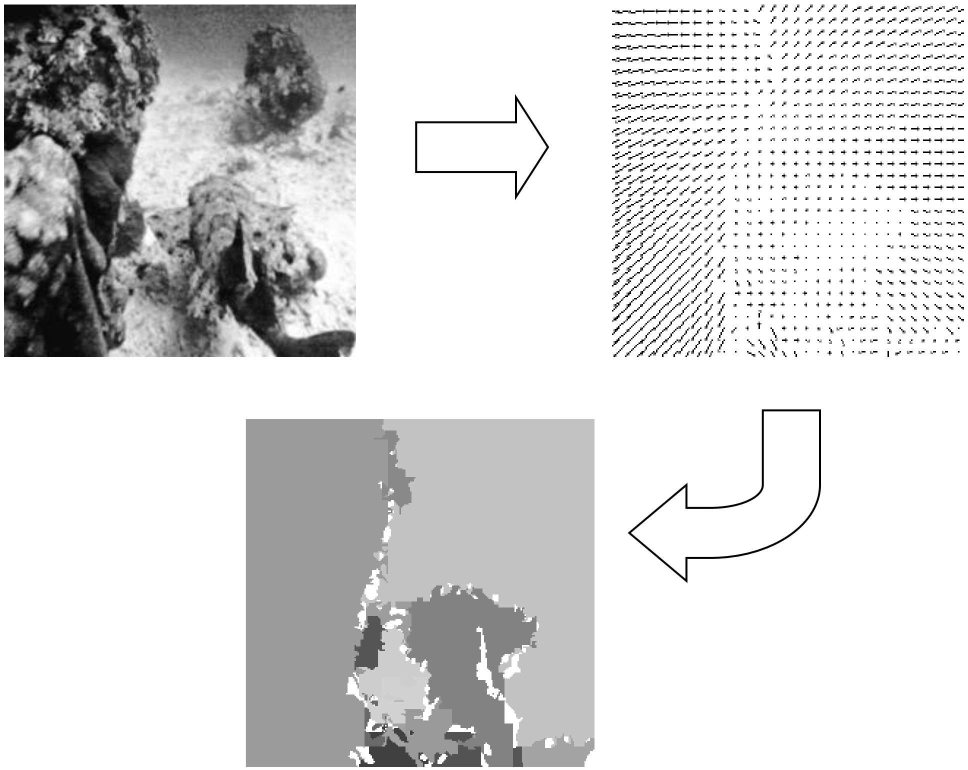


Image Segmentation

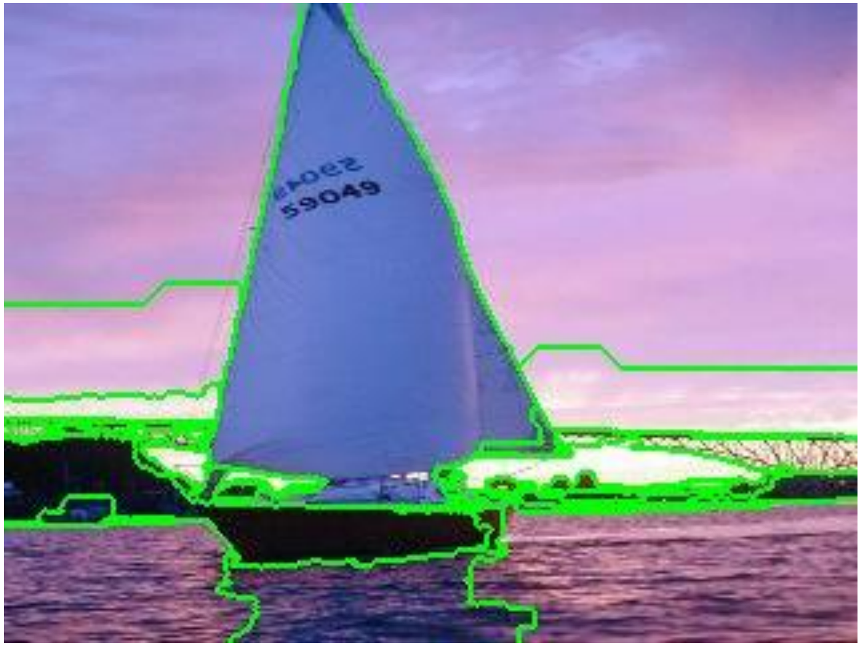
Image segmentation is the operation of **partitioning an image into a collection of connected sets of pixels**.

1. into regions, which usually cover the image

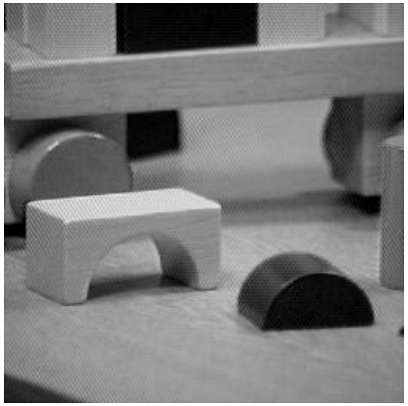
2. into linear structures, such as
   - line segments
   - curve segments

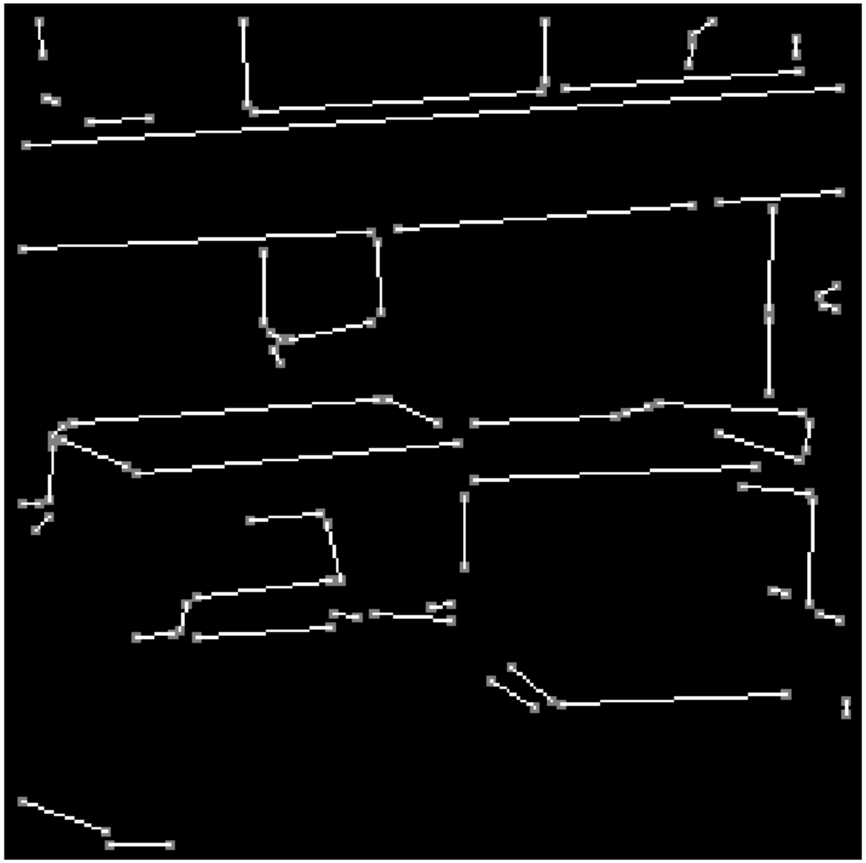
3. into 2D shapes, such as
   - circles
   - ellipses
   - ribbons (long, symmetric regions)

Example 1: Regions



Example 2: Straight Lines



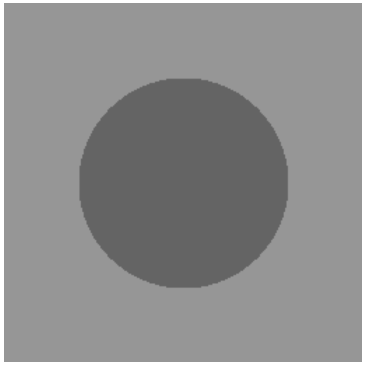


Example 3:
Lines and Circular Arcs

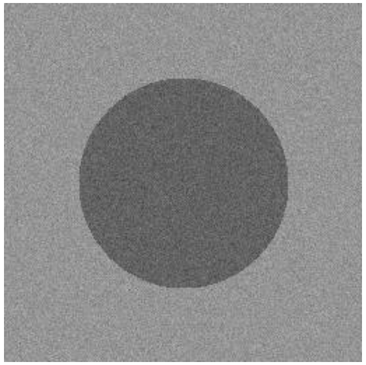
Grey-level histogram-based segmentation

- We will look at *two very simple image segmentation techniques* that are based on the grey-level histogram of an image
  1. **Thresholding**
  2. **Clustering**
- We will use a very simple object-background test image
  - We will consider a zero, low, and high noise images

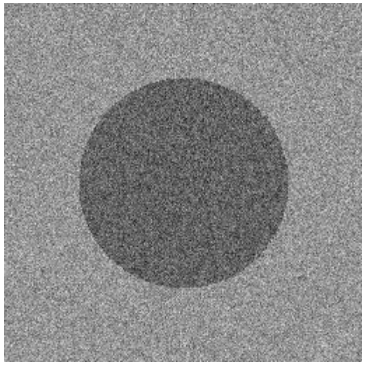
Grey-level histogram-based segmentation



Noise free



Low noise

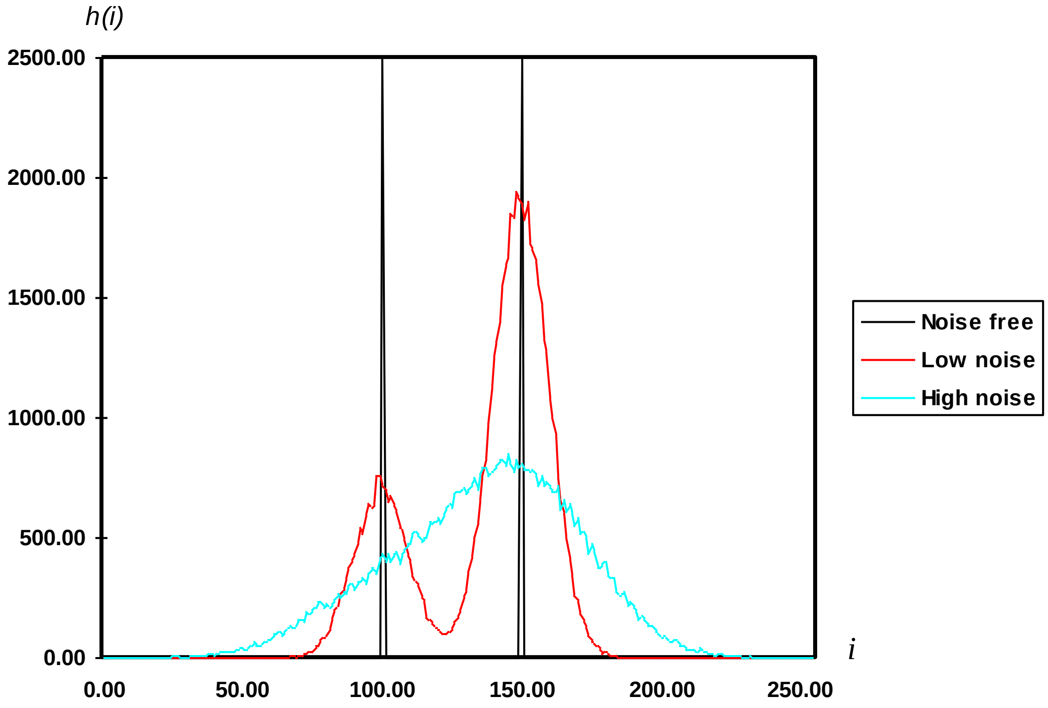


High noise

Grey-level histogram-based segmentation

- How do we characterise low noise and high noise?
- We can consider the histograms of our images
  - For the **noise free image**, its simply **two peaks** at i=100, $i=150$
  - For the **low noise image**, there are **two clear peaks** centred on i=100, $i=150$
  - For the **high noise image**, there is a **single peak** – two grey-level populations corresponding to object and background have merged

Grey-level histogram-based segmentation



Grey-level histogram-based segmentation

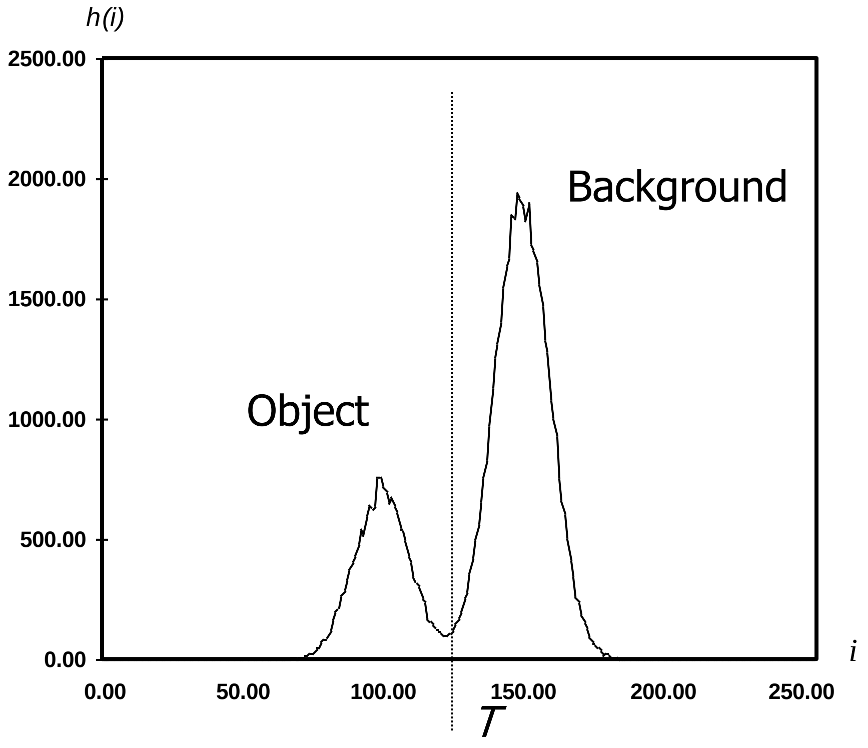
- We can define the input image *signal-to-noise ratio* in terms of the mean grey-level value of the object pixels and background pixels and the additive noise standard deviation.

$$S / N = \frac{|\mu_b - \mu_o|}{\sigma}$$

- For our test images :
  - *S/N* (noise free) = ∞
  - *S/N* (low noise) = 5
  - *S/N* (high noise) = 2

Grey-level thresholding

- We can easily understand segmentation based on thresholding by looking at the histogram of the low noise object/background image
  - **There is a clear 'valley' between to two peaks**



Grey-level thresholding

- We can define the grey-level thresholding algorithm as follows:
  - If the grey-level of pixel p <=T then pixel p is an object pixel

  Else
  - Pixel p is a background pixel

- This simple threshold test, the obvious question how do we determine the threshold ?

There are many approaches possible

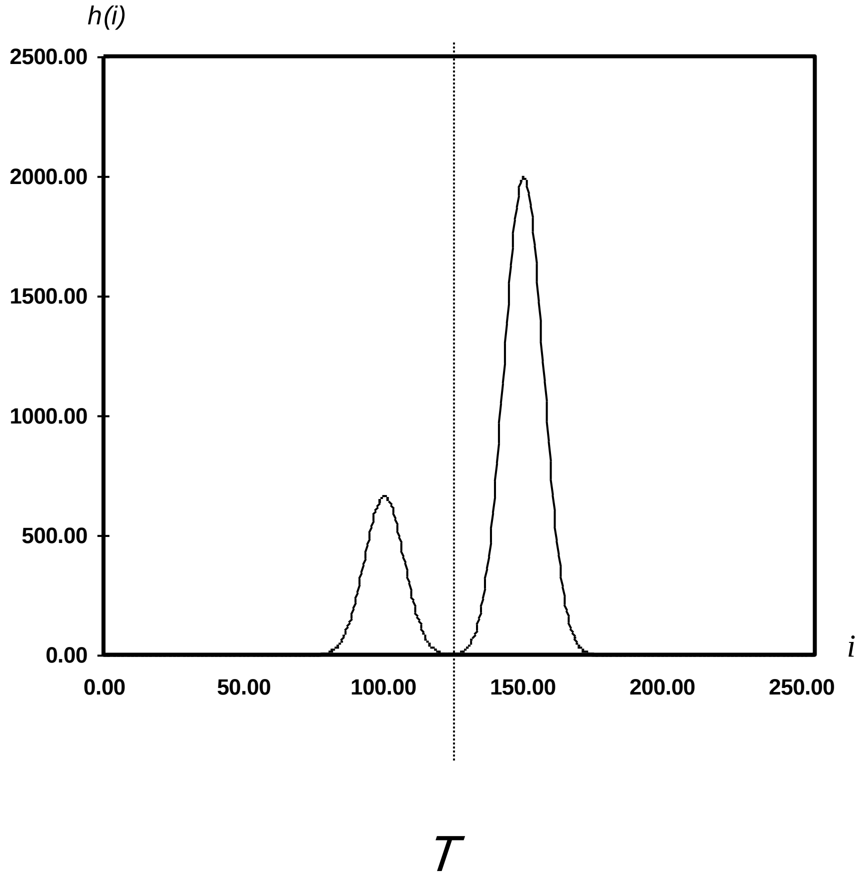
1. Interactive threshold
2. Adaptive threshold
3. Minimization method

Grey-level thresholding

- We will consider in detail a minimization method for determining the threshold
  - Minimization of the *within group variance*
  - Robot Vision, Haralick & Shapiro, volume 1, page 20

Grey-level thresholding

- Idealized object/background image histogram



Grey-level thresholding

- Any threshold separates the histogram into 2 groups with each group having its own statistics (mean, variance)
- The homogeneity of each group is measured by the ***within group variance***
- The optimum threshold is that threshold which **minimizes** the within group variance thus maximizing the homogeneity of each group

Grey-level thresholding

- The following expressions can easily be derived for prior probabilities of object and background

$$p_o(T) = \sum_{i=0}^{T} P(i)$$

$$p_b(T) = \sum_{i=T+1}^{255} P(i)$$

$$P(i) = h(i) / N$$

- where *h(i)* is the histogram of an N pixel image

Grey-level thresholding

- The mean and variance of each group are as follows :

$$\mu_o(T) = \sum_{i=0}^{T} iP(i) / p_o(T)$$

$$\mu_b(T) = \sum_{i=T+1}^{255} iP(i) / p_b(T)$$

$$\sigma_o^2(T) = \sum_{i=0}^{T} \left[i - \mu_o(T)\right]^2 P(i) / p_o(T)$$

$$\sigma_b^2(T) = \sum_{i=T+1}^{255} \left[i - \mu_b(T)\right]^2 P(i) / p_b(T)$$

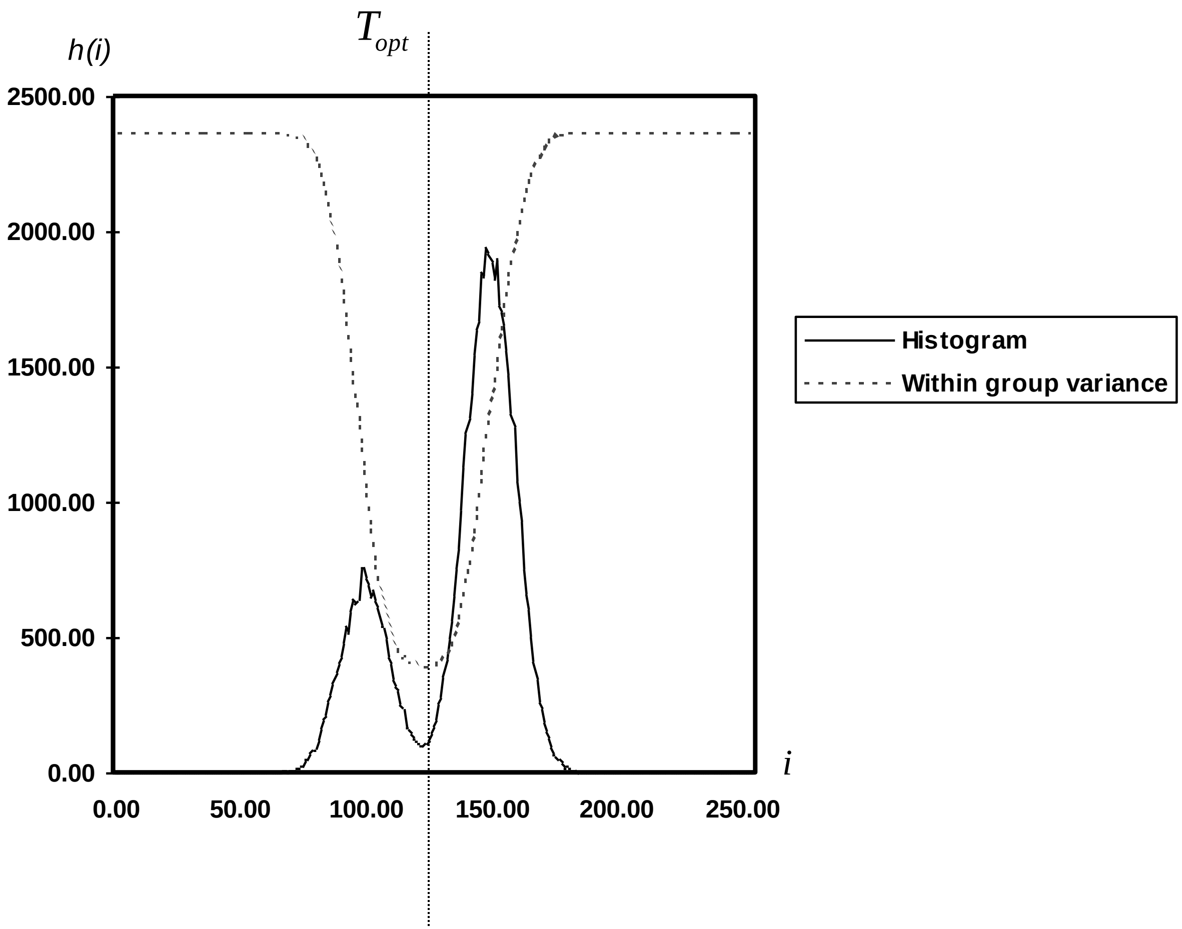
Grey-level thresholding

- The within group variance is defined as :

$$\sigma_W^2(T) = \sigma_o^2(T)p_o(T) + \sigma_b^2(T)p_b(T)$$

- We determine the optimum T by minimizing this expression with respect to T
  - Only requires 256 comparisons for an 8-bit grey-level image

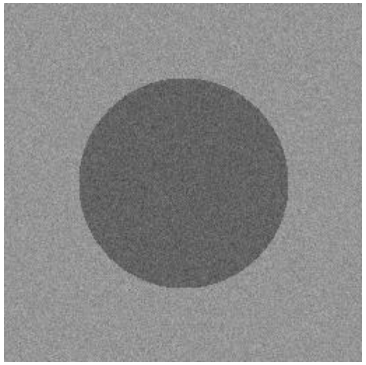
Grey-level thresholding



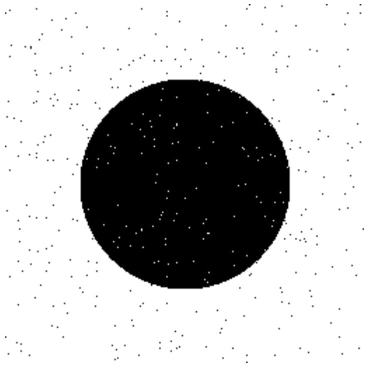
Grey-level thresholding

- We can examine the performance of this algorithm on our low and high noise image
  - For the low noise case, it gives an optimum threshold of T=124
  - Almost exactly halfway between the object and background peaks
  - We can apply this optimum threshold to both the low and high noise images

Grey-level thresholding

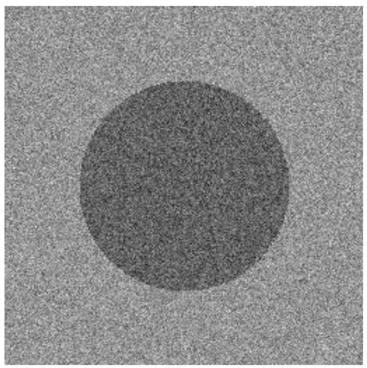


Low noise image

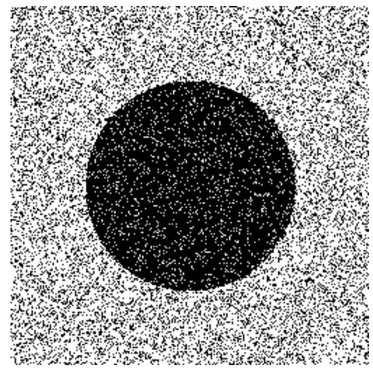


Thresholded at *T=124*

Grey-level thresholding



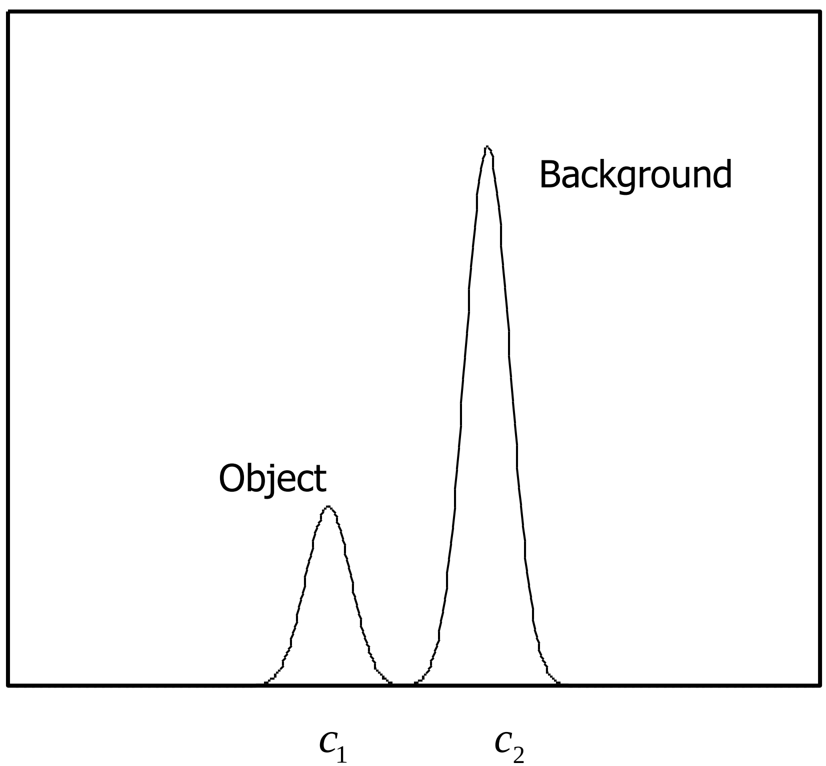
High noise image



Thresholded at *T=124*

Grey-level clustering

- Consider an idealized object/background histogram



Grey-level clustering

- Clustering tries to separate the histogram into 2 groups
- Defined by two cluster centres c_1 and c_2
  - Grey-levels classified according to the nearest cluster centre

Grey-level clustering

- A *nearest neighbour* clustering algorithm allows us perform a grey-level segmentation using clustering.
  - A simple case of a more general and widely used ***K-means* clustering.**
  - A simple iterative algorithm which has known convergence properties.

Grey-level clustering

- Given a set of grey-levels

$$\{g(1), g(2) \ldots\ldots g(N)\}$$

- We can partition this set into two groups

$$\{g_1(1), g_1(2) \ldots\ldots g_1(N_1)\}$$

$$\{g_2(1), g_2(2) \ldots\ldots g_2(N_2)\}$$

Grey-level clustering

- Compute the local means of each group

$$c_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} g_1(i)$$

$$c_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} g_2(i)$$

Grey-level clustering

- Re-define the new groupings

$$\left|g_1(k)-c_1\right| < \left|g_1(k)-c_2\right| \quad k = 1..N_1$$

$$\left|g_2(k)-c_2\right| < \left|g_2(k)-c_1\right| \quad k = 1..N_2$$

- In other words all grey levels in set 1 are nearer to cluster centre c_1 and all grey levels in set 2 are nearer to cluster centre c_2

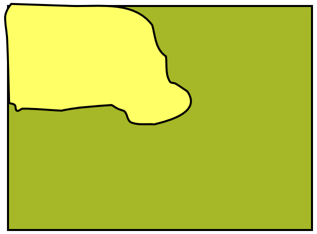
Main Methods of Region Segmentation

1. Region Growing

2. Clustering

3. Split and Merge

Region Growing



Region growing techniques start with one pixel of a potential region and try to grow it by adding adjacent pixels till the pixels being compared are too dissimilar.

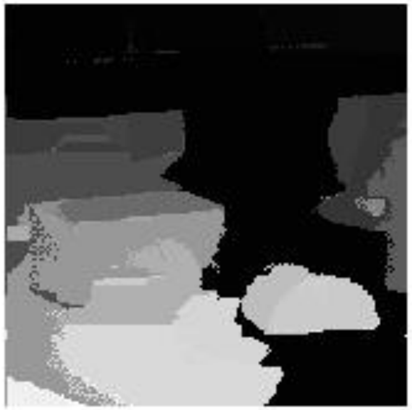
- The first pixel selected can be just the first unlabeled pixel in the image or a ***set of seed pixels*** *can be chosen from the image*.
- Usually **a statistical test** is used to decide which pixels can be added to a region.

RGGROW Example

image

Not great!

segmentation



Clustering

- There are **K** clusters $C_1, \ldots, C_K$ with means $m_1, \ldots, m_K$.

- The **least-squares error** is used as distance measure.

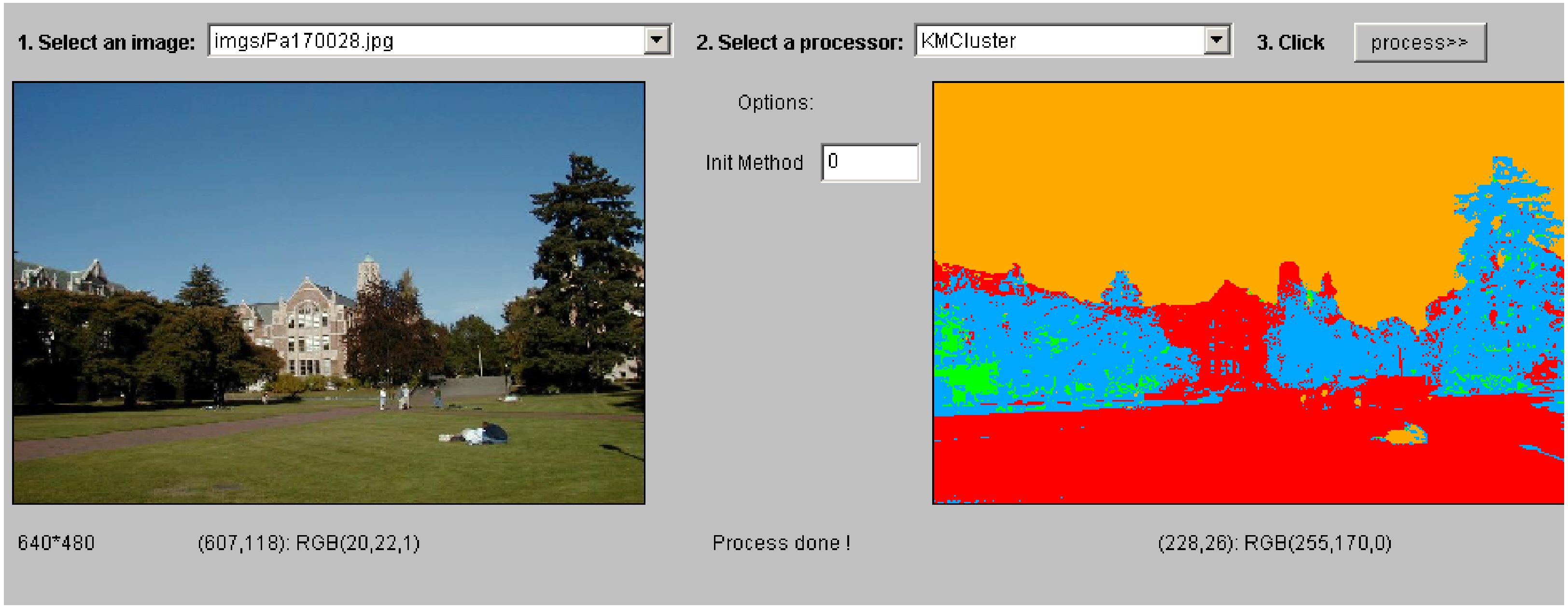
- Out of all possible partitions into **K** clusters, choose the one that minimizes **D**.

- **Some Clustering Methods**
  1. K-means Clustering and Variants.
  2. Iso-data Clustering.
  3. Histogram-Based Clustering and Recursive Variant.
  4. Graph-Theoretic Clustering.

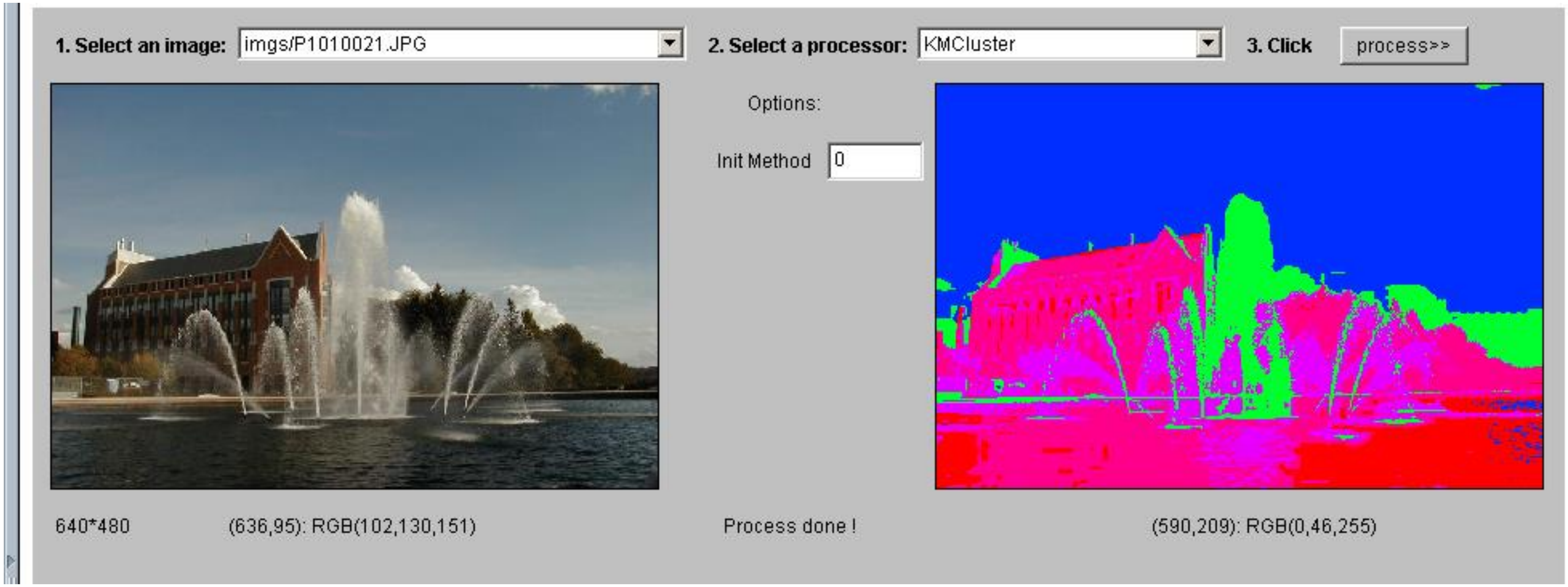
K-means Clustering

1. Randomly select *'K'* cluster centers.
2. Calculate the distance between each data point and cluster centers.
3. Assign *the data point to the cluster center* whose distance from the cluster center is **minimum** of all the cluster centers.
4. Recalculate *the new cluster center*.
5. Recalculate the distance between each data point and new obtained cluster centers.
6. If *no data point was reassigned* then stop, otherwise repeat from step 4).

K-means Clustering
Example 1



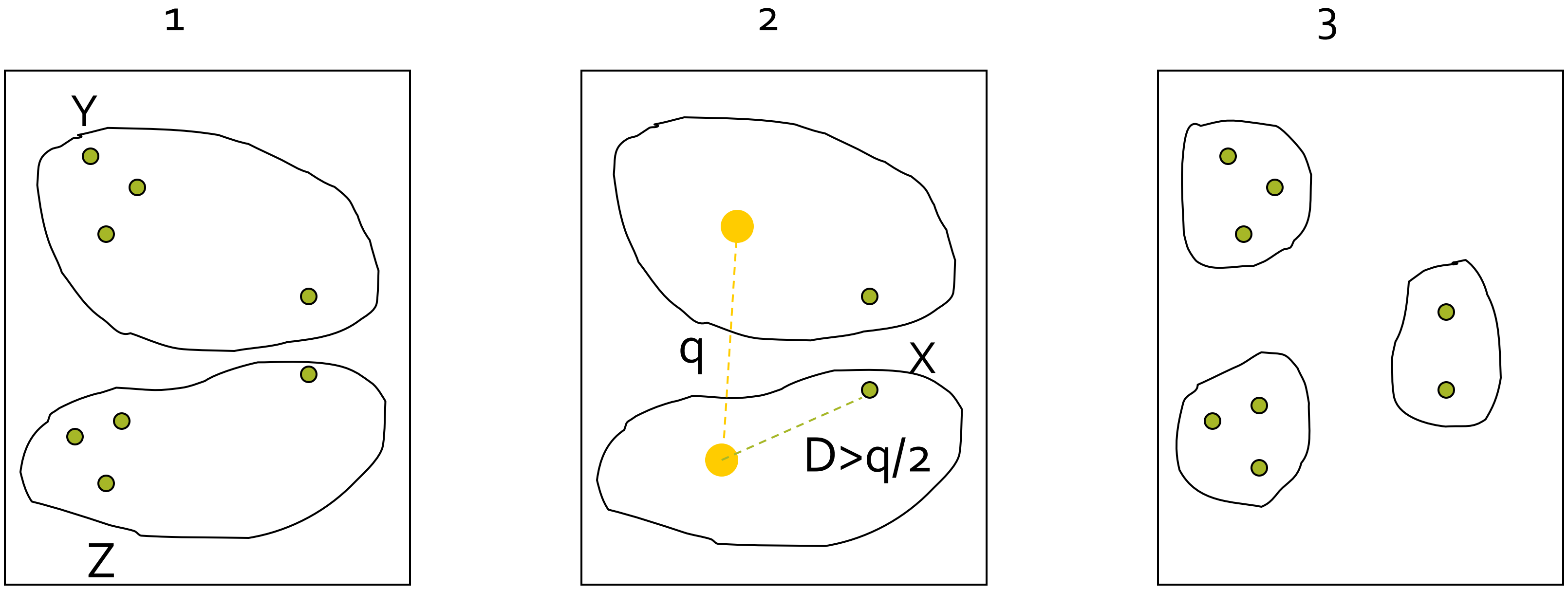
K-means Clustering
Example 2



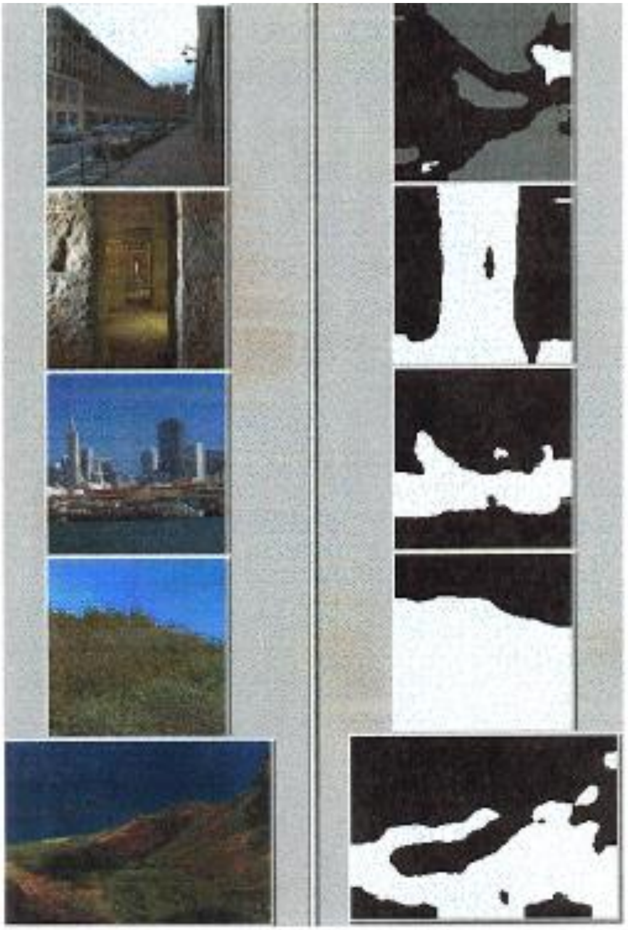
K-means Variant by Heng

1. **Pick 2 points Y and Z** that are *furthest apart in the measurement space* and make them initial cluster means.
2. *Assign all points to the cluster* whose mean they are closest to.
3. Recalculate *the new cluster means*.
4. Let **d** be the max distance from each point to its cluster mean and let **X** be the point with this distance.
5. Let **q** be the average distance between each pair of means.
6. If **d > q / 2**, make **X** a new cluster mean.
7. If *a new cluster was formed*, repeat from step 2.

Illustration of Heng Clustering



Heng Clustering with Texture Feature



Iso-data Clustering

1. **Select several cluster means** and form clusters.

2. **Split** any cluster whose *variance is too large*.

3. **Group together clusters** whose *variances are too small*.

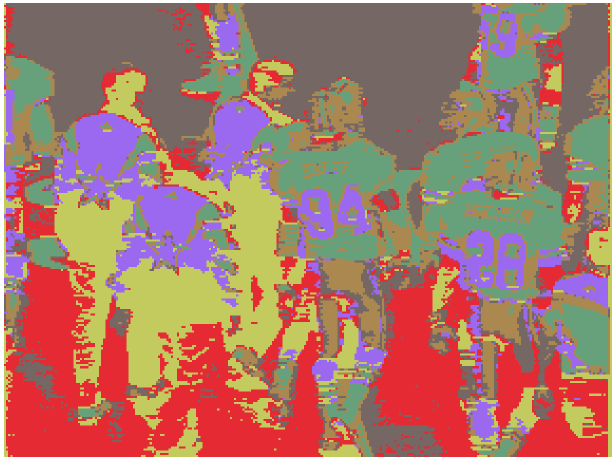
4. Recompute means.

5. Repeat till 2 and 3 cannot be applied.

We used this to cluster normal vectors in 3D data.

Comparison

K-means, K=6



Original

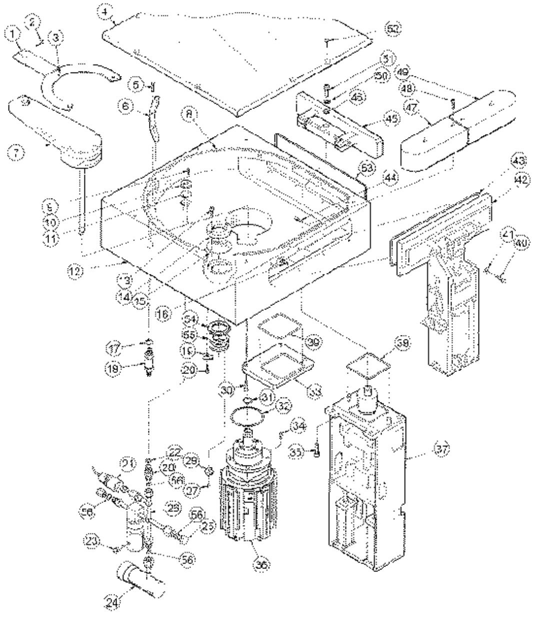
Iso-data, K became 5

Lines and Arcs Segmentation

In some image sets, lines, curves, and circular arcs are *more useful than regions* or helpful in addition to regions.

Lines and arcs are often used in

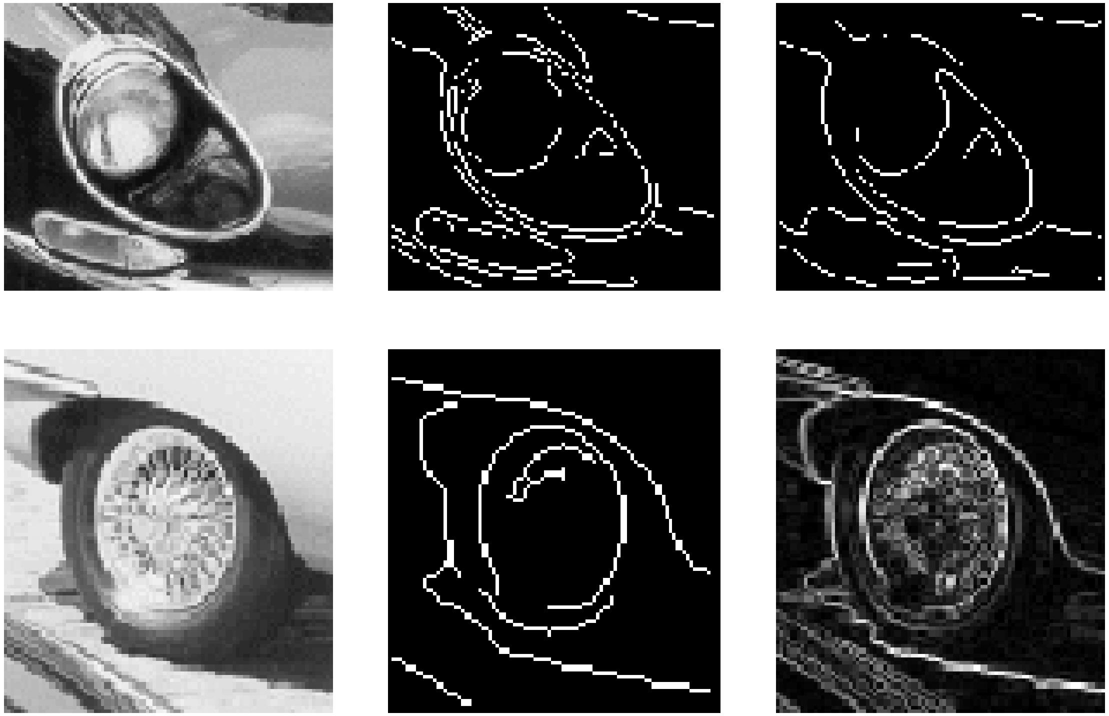
- Object recognition
- Stereo matching
- Document analysis



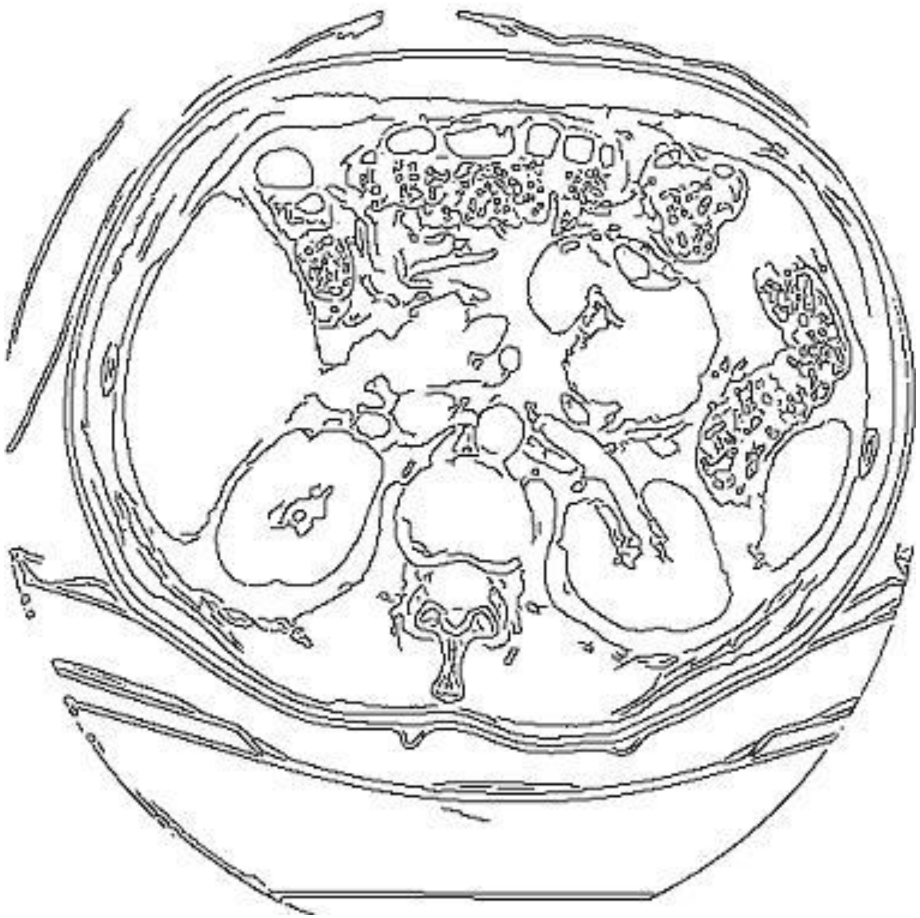
Canny Edge Detector

- Smooth the image with a Gaussian filter.
- Compute gradient magnitude and direction at each pixel of the smoothed image.
- Zero out any pixel response $\leq$ the two neighboring pixels on either side of it, along the direction of the gradient.
- Track high-magnitude contours.
- Keep only pixels along these contours, so weak little segments go away.

Canny
Example 1



Canny
Example 2 - **Kidney**



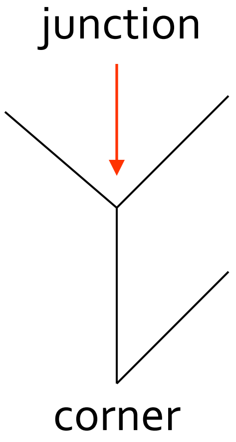
Finding Line and Curve Segments from Edge Images

Given an edge image, how do we find line and arc segments?

Method 1: Tracking

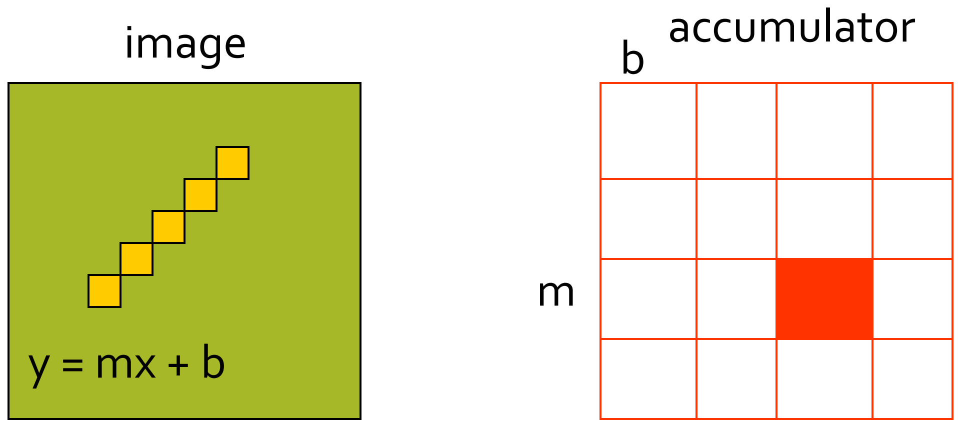
Use masks to identify the following events:

1. start of a new segment
2. interior point continuing a segment
3. end of a segment
4. junction between multiple segments
5. corner that breaks a segment into two



Hough Transform

- The Hough transform is a method for detecting lines or curves specified by a **parametric function**.
- If the parameters are p1, p2, … pn, then the Hough procedure uses an n-dimensional accumulator array in which it accumulates votes for the correct parameters of the lines or curves found on the image.



Parameters for analytic curves

Analytic Form	Parameters	Equation
Line	ρ, θ	$x\cos\theta+y\sin\theta=\rho$
Circle	x_0, y_0, ρ	$(x-x_o)^2+(y-y_0)^2=r^2$
Parabola	x_0, y_0, ρ, θ	$(y-y_0)^2=4\rho(x-x_o)$
Ellipse	x_0, y_0, a, b, θ	$(x-x_o)^2/a^2+(y-y_0)^2/b^2=1$

Generalized Hough Transform

- The Generalized Hough transform can be used to detect arbitrary shapes

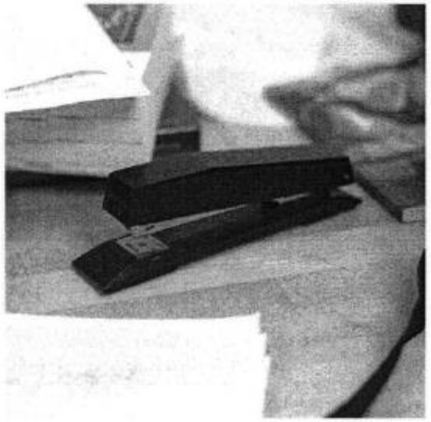
- Complete specification of the exact shape of the target object is required.

- Information that can be extracted are
  - **Location**
  - **Size**
  - **Orientation**
  - **Number of occurrences of that particular shape**

Generalized Hough Transform – Advantages and disadvantages

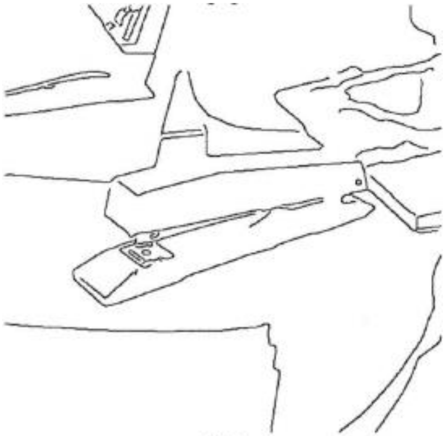
- Advantages
  - A method for object recognition
  - Robust to partial deformation in shape
  - Tolerant to noise
  - Can detect multiple occurrences of a shape in the same pass
- Disadvantages
  - Lot of memory and computation is required

Example



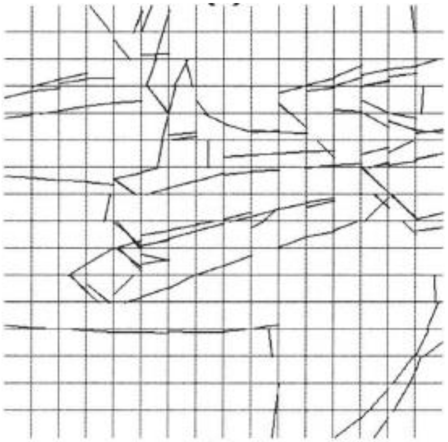
(a)

Original Image



(b)

Edge Detected Image



(c)

HT Results