IE404
Digital Image Processing

Dr. Manish Khare

Image Segmentation

Image Segmentation

Preview

- If an image has been preprocessed appropriately to remove noise and artifacts, segmentation is often the key step in interpreting the image.
- Image segmentation is a process in which regions or features sharing similar characteristics are identified and grouped together.
- Segmentation is to subdivide an image into its constituent regions or objects.
- Segmentation should stop when the objects of interest in an application have been isolated.

- **Image segmentation may use statistical classification, thresholding, edge detection, region detection, or any combination of these techniques.**

- **The output of the segmentation step is usually a set of classified elements,**

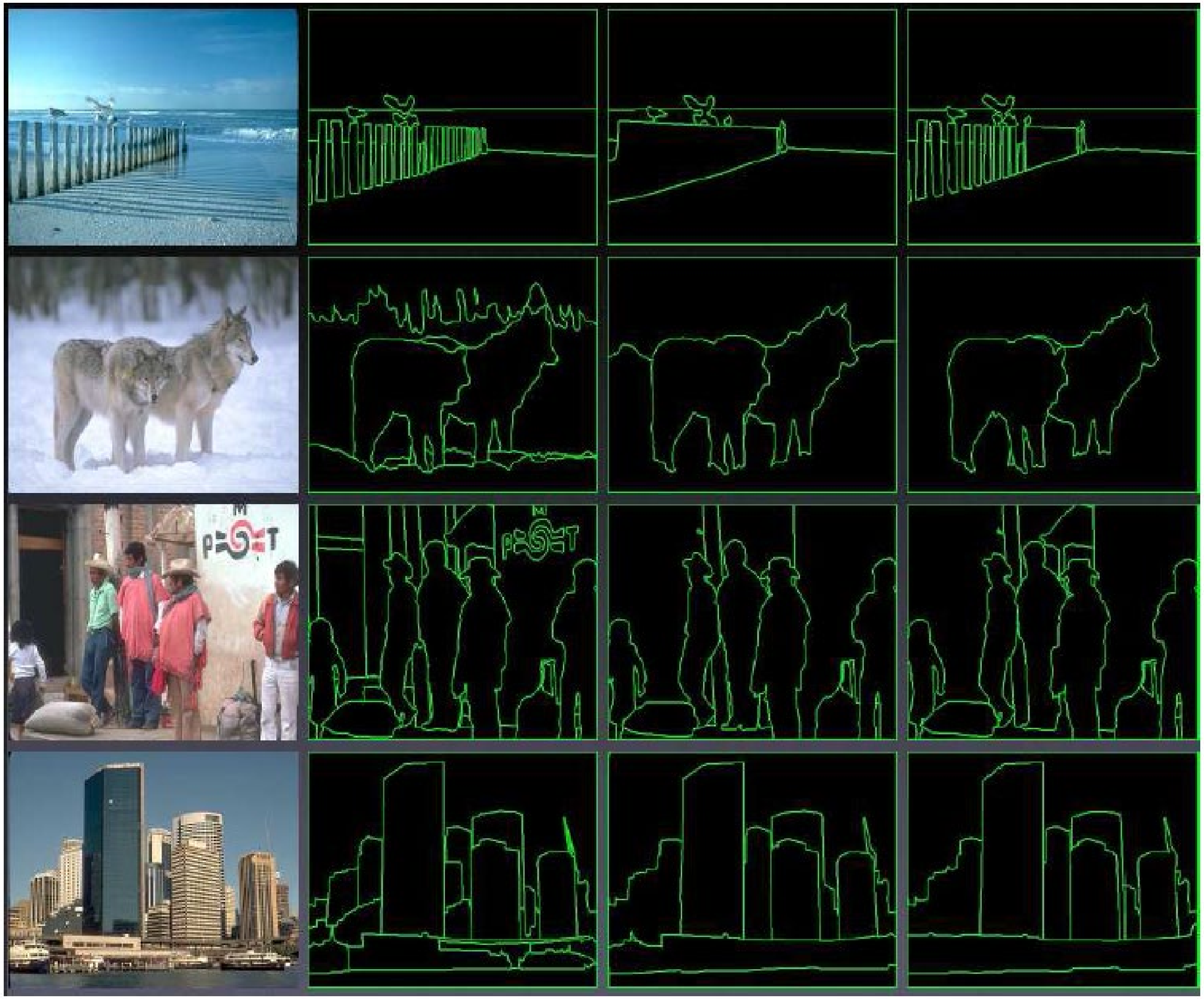
- **Segmentation techniques are either region-based or edge-based.**

- **Region-based techniques rely on common patterns in intensity values within a cluster of neighboring pixels. The cluster is referred to as the region, and the goal of the segmentation algorithm is to group regions according to their anatomical or functional roles.**

- **Edge-based techniques rely on discontinuities in image values between distinct regions, and the goal of the segmentation algorithm is to accurately demarcate the boundary separating these regions.**

Principal approaches

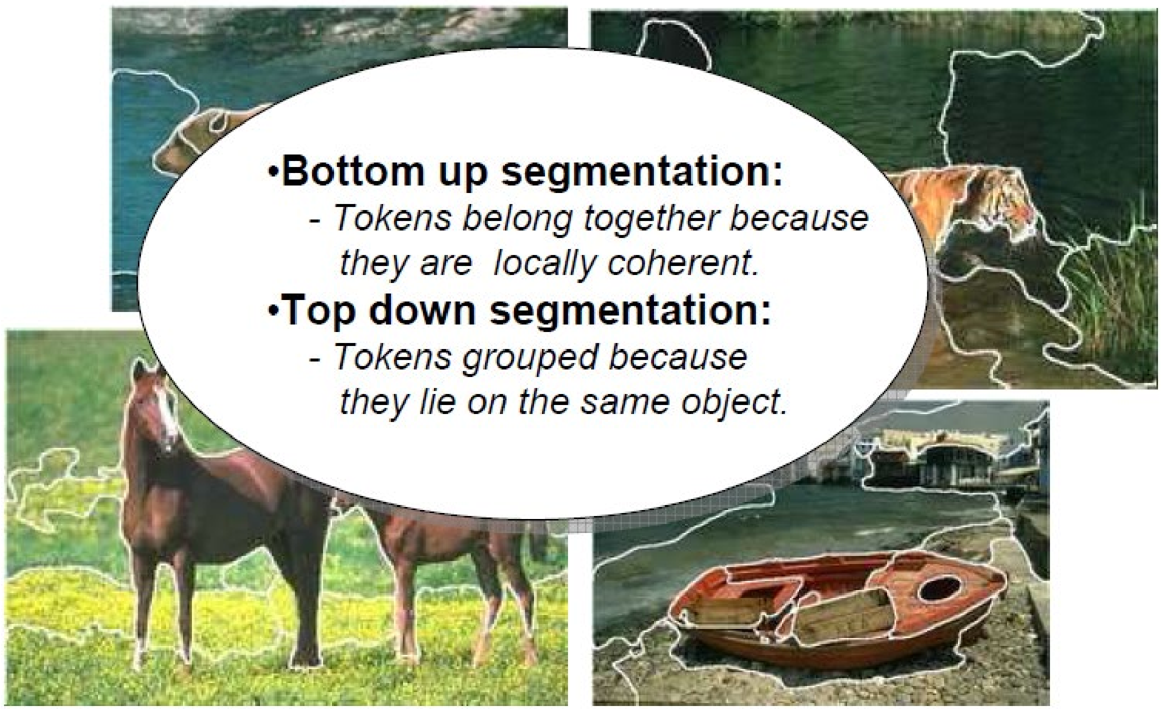
- Segmentation algorithms generally are based on one of 2 basis properties of intensity values
  - discontinuity : to partition an image based on abrupt changes in intensity (such as edges)
  - similarity : to partition an image into regions that are similar according to a set of predefined criteria.



Introduction to image segmentation

- 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 *greylevel*, *colour*, *texture*, *depth* or *motion*

➢ How do you pick the right segmentation?

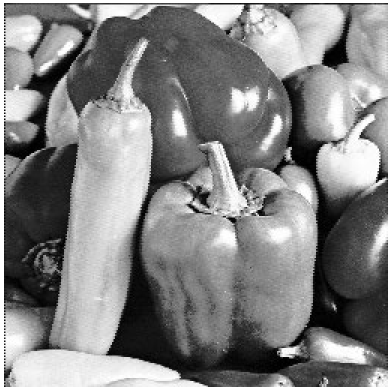


Introduction to image segmentation

- 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

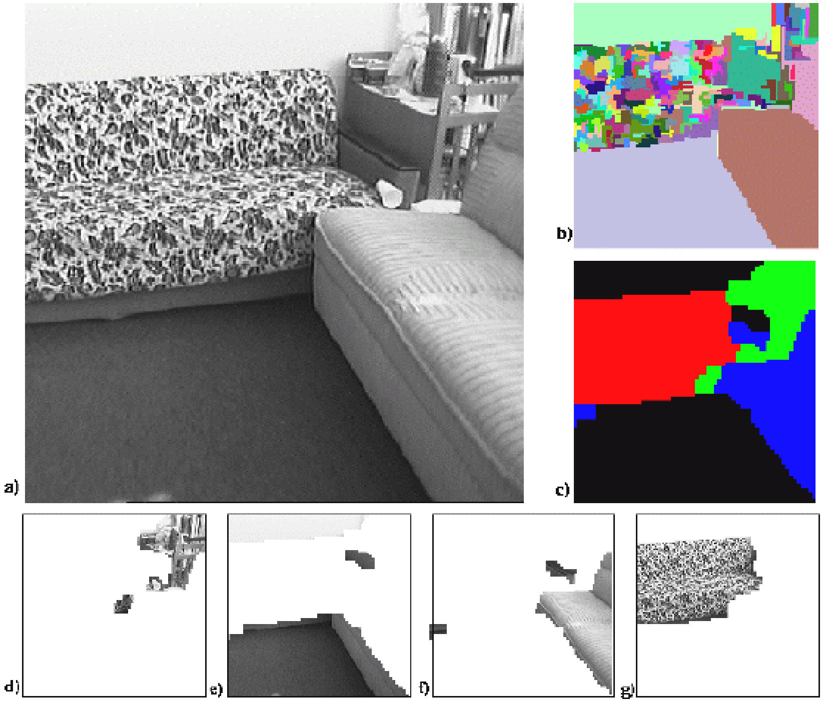
Introduction to image segmentation

- Example 1
  - Segmentation based on greyscale
  - Very simple 'model' of greyscale leads to inaccuracies in object labelling



Introduction to image segmentation

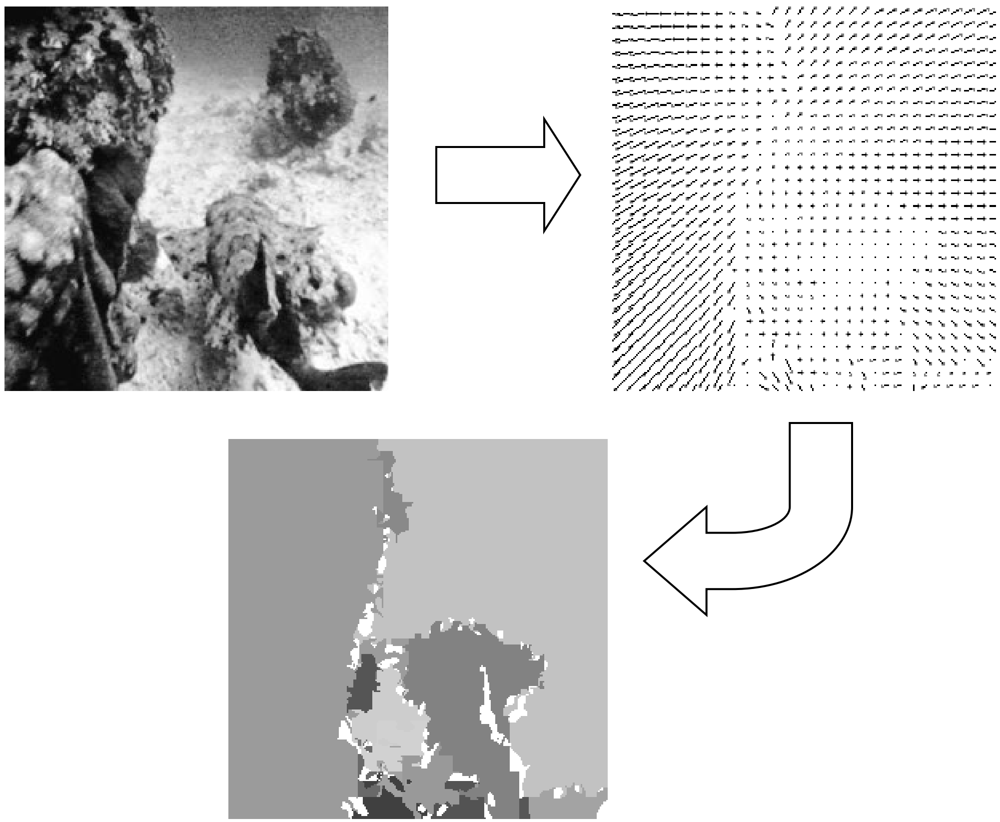
- Example 2
  - Segmentation based on texture
  - Enables object surfaces with varying patterns of grey to be segmented



Introduction to image segmentation

- Example 3
  - 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*
  - The segmentation must be based on this estimate and not, in general, the true flow

Introduction to image segmentation

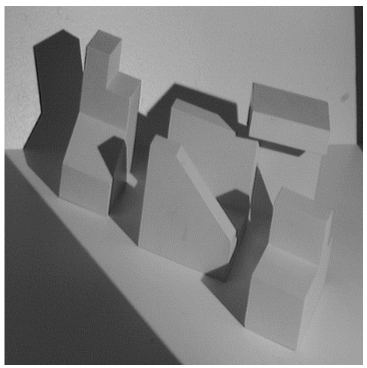


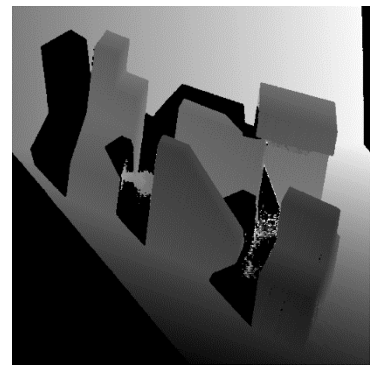
Introduction to image segmentation

- Example 3
  - Segmentation based on depth
  - This example shows a range image, obtained with a laser range finder
  - A segmentation based on the range (the object distance from the sensor) is useful in guiding mobile robots

Introduction to image segmentation

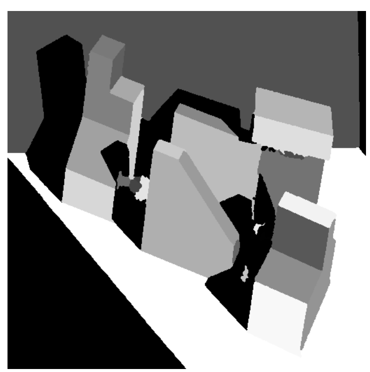
Original image





Range image

Segmented image



“Correct” segmentation

- There may not be a single correct answer.
- Gestalt Laws seek to formalize what is an object
  - proximity, similarity, continuation, closure, common fate
- Segmentation can be thought of a partition problem
- Many approaches proposed
  - cues:
    - color, regions, contours, texture, motions
  - automatic vs. user-guided
  - no clear winner

Overview of Segmentation Techniques

- Edge-based
- Color-based
- Texture-based
- Disparity-based
- Motion-based

- Document images
- Medical images
- Range images
- Biometric images
- Texture images

Image Segmentation

Segmentation: Dividing into the regions/segments of similar properties

- Discontinuity – boundary
- Similarity – region

Discontinuity Detection: Point, Line and Edge

Mask Operation (Review)

FIGURE 10.1 A general 3 × 3 mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Detection of Discontinuities

- detect the three basic types of gray-level discontinuities
  - points , lines , edges
- the common way is to run a mask through the image

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Point Detection

−1	−1	−1
−1	8	−1
−1	−1	−1

- a point has been detected at the location on which the mark is centered if

$$|R| \geq T$$

- where
  - T is a nonnegative threshold
  - R is the sum of products of the coefficients with the gray levels contained in the region encompassed by the mark.

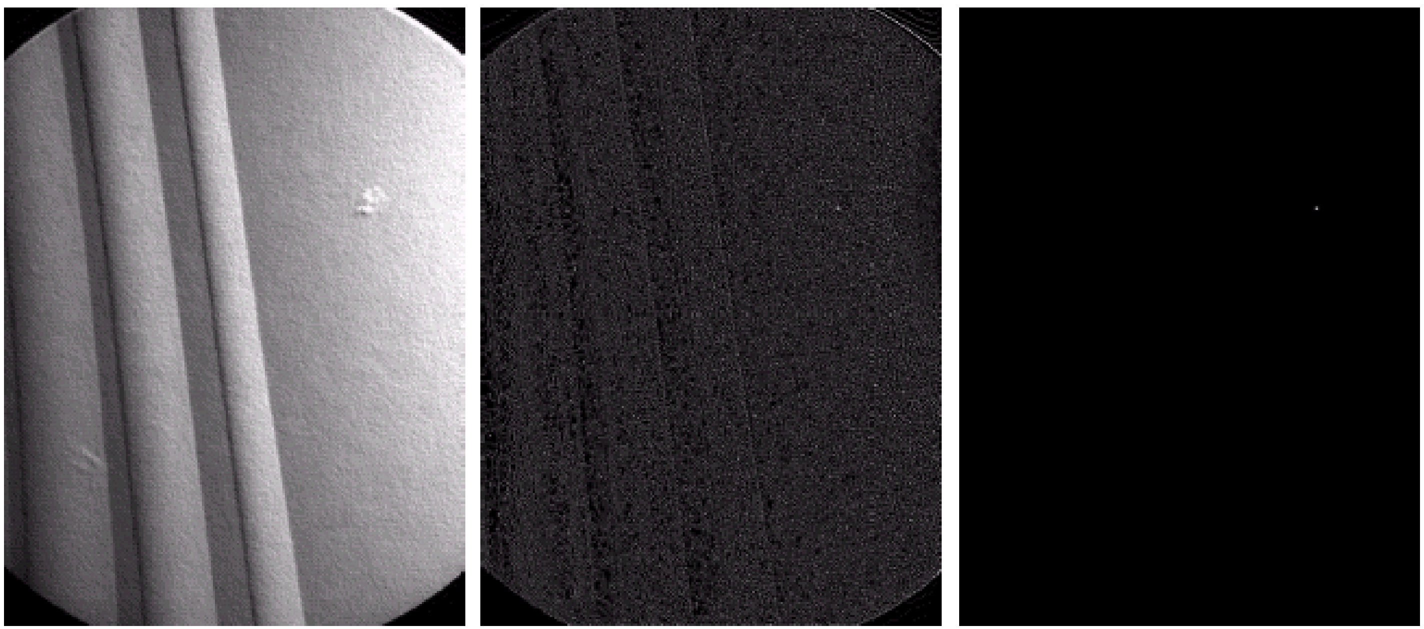
Point Detection

- Note that the mark is the same as the mask of Laplacian Operation (in chapter 3)
- The only differences that are considered of interest are those large enough (as determined by T) to be considered isolated points.

$$|R| \geq T$$

Example

−1	−1	−1
−1	8	−1
−1	−1	−1



a
b c d

FIGURE 10.2
(a) Point detection mask.
(b) X-ray image of a turbine blade with a porosity.
(c) Result of point detection.
(d) Result of using Eq. (10.1-2).
(Original image courtesy of X-TEK Systems Ltd.)

Line Detection

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

$+45°$

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

$-45°$

- Horizontal mask will result with max response when a line passed through the middle row of the mask with a constant background.
- the similar idea is used with other masks.
- note: the preferred direction of each mask is weighted with a larger coefficient (i.e.,2) than other possible directions.

Line Detection

- Apply every masks on the image
- let R1, R2, R3, R4 denotes the response of the horizontal, +45 degree, vertical and -45 degree masks, respectively.
- if, at a certain point in the image

$$|R_i| > |R_j|,$$

- for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i.

Line Detection

- Alternatively, if we are interested in detecting all lines in an image in the direction defined by a given mask, we simply run the mask through the image and threshold the absolute value of the result.
- The points that are left are the strongest responses, which, for lines one pixel thick, correspond closest to the direction defined by the mask.

Example

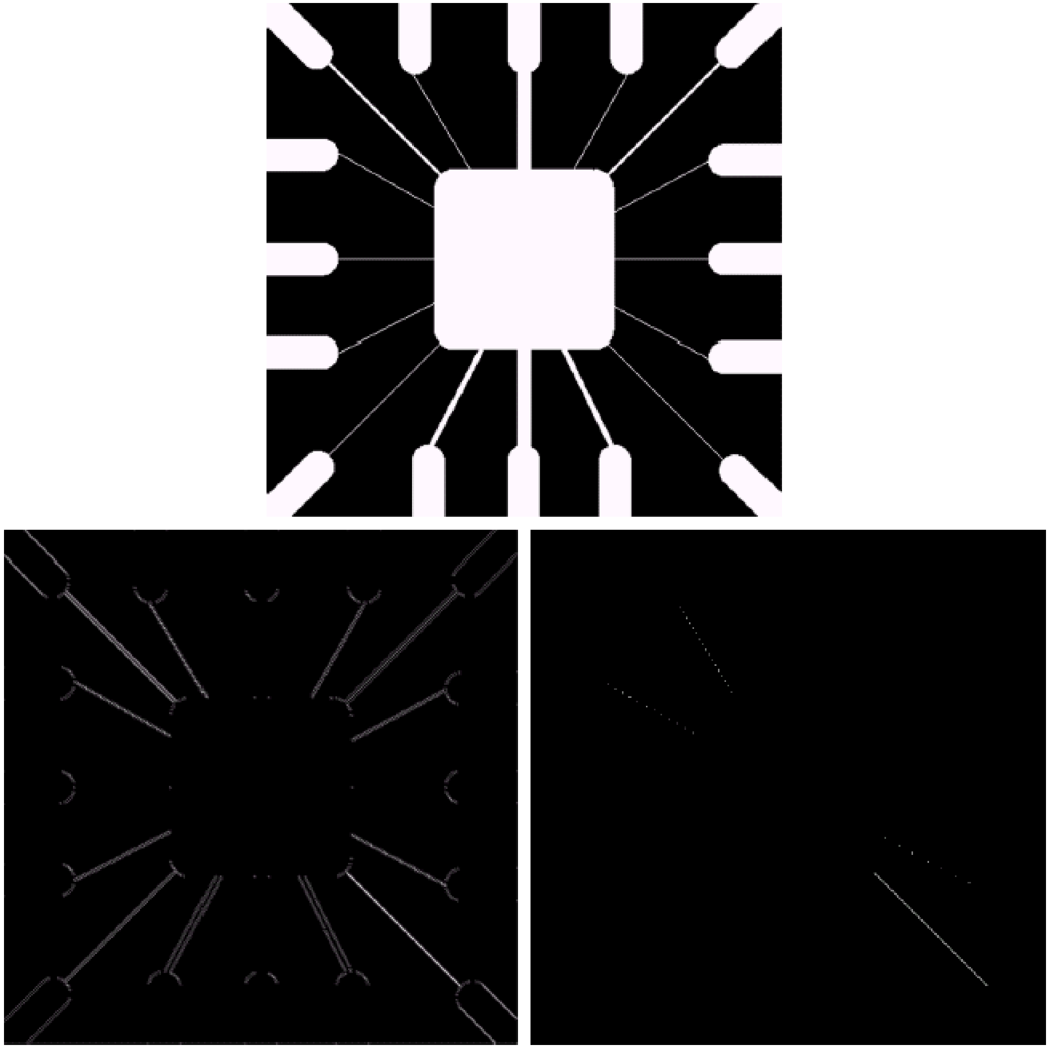


FIGURE 10.4 Illustration of line detection. (a) Binary wire-bond mask. (b) Absolute value of result after processing with $-45°$ line detector. (c) Result of thresholding image (b).

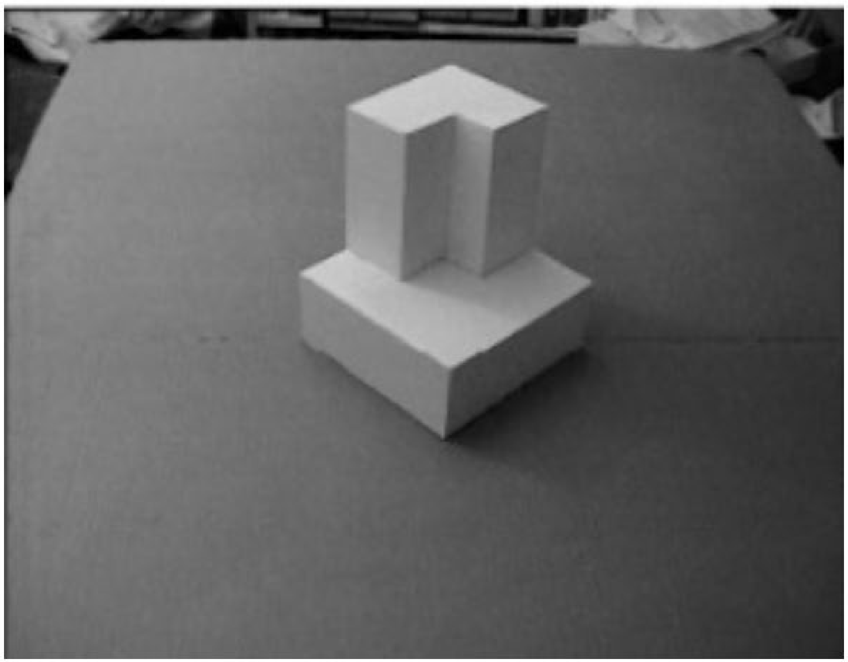
Edge Detection

- the most common approach for detecting meaningful discontinuities in gray level.
- we discuss approaches for implementing
  - first-order derivative (Gradient operator)
  - second-order derivative (Laplacian operator)
- Here, we will talk only about their properties for edge detection.
- we have introduced both derivatives in chapter 3

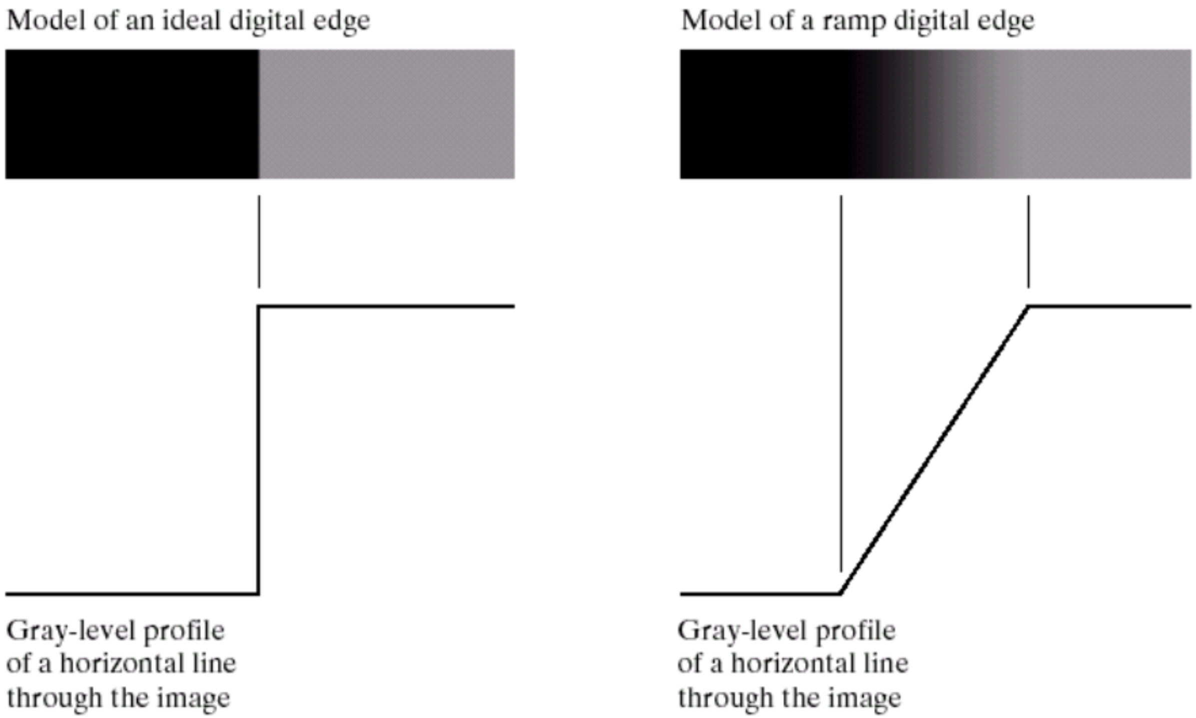
Basic Formulation

- an edge is a set of connected pixels that lie on the boundary between two regions.
- an edge is a "local" concept whereas a region boundary, owing to the way it is defined, is a more global idea.

Edge: Sharp change in intensity (discontinuity)



Edge: Sharp change in intensity (discontinuity)



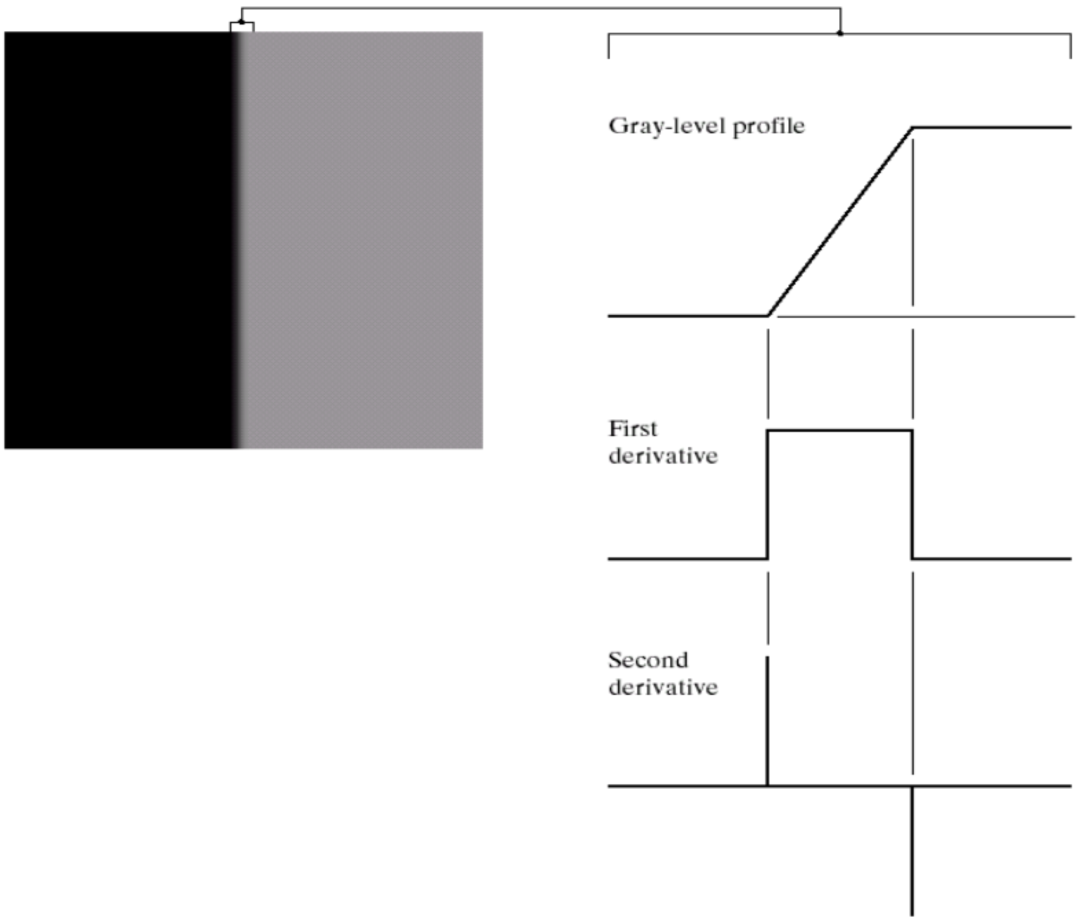
a b

FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

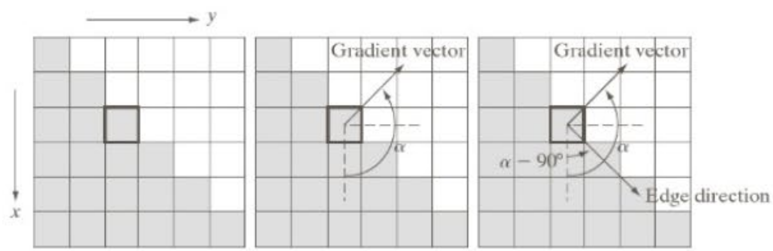
Edge Detection

a b

FIGURE 10.6
(a) Two regions separated by a vertical edge.
(b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.



Edge Detection



a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

$$\nabla f = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

Strength and direction of an edge can be determined using the gradient

Strength (magnitude)

$$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

Direction

$$\alpha(x,y) = Tan^{-1}\left(\frac{g_x}{g_y}\right)$$

Edge Detection

- Partial derivatives of images replaced by finite differences

$$\Delta_x f = f(x,y) - f(x-1,y) \qquad \begin{bmatrix} -1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\Delta_y f = f(x,y) - f(x,y-1)$$

- Alternatives are:

$$\Delta_{2x} f = f(x+1,y) - f(x-1,y) \qquad \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$\Delta_{2y} f = f(x,y+1) - f(x,y-1)$$

- Robert's gradient

$$\Delta_{+} f = f(x+1,y+1) - f(x,y) \rightarrow \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\Delta_{-} f = f(x,y+1) - f(x+1,y) \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Prewitt

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel

Edge Detection

a b
c d

FIGURE 10.10 (a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.

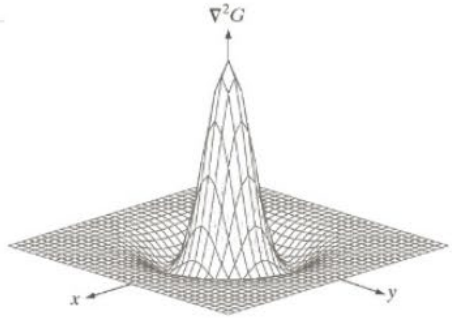
Edge Detection

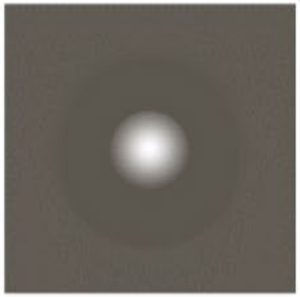
Laplacian of Gaussian (LoG)

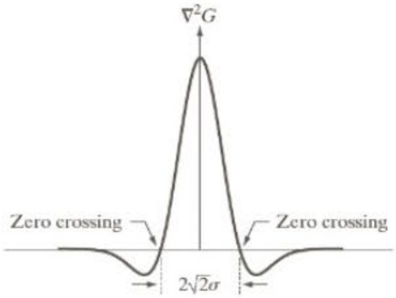
The LoG is sometimes called the Mexican hat operator

The Laplacian is NEVER used directly because of its strong noise sensitivity

Combining the Laplacian with a Gaussian gives the LoG







0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5 × 5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

$$\nabla^2 h(r) = -\left[\frac{r^2-\sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

Edge Detection

Laplacian of Gaussian (LoG)

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla^2 G(x,y) = [\frac{x^2+y^2-2\sigma^2}{\sigma^4}]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$g(x,y) = [\nabla^2 G(x,y)] * f(x,y)$$
$$g(x,y) = \nabla^2 [G(x,y) * f(x,y)]$$

Canny Edge Detection

- Low error rate
  All true edges should be found
- Edge points should be well localized
  Edges should be located as close as possible to the true edges
- Single edge point response
  One point for each true edge point
  No of local maxima around true should be minimum

Canny Edge Detection

1. Smooth image with a Gaussian filter Low error rate

2. Compute gradient magnitude M[x,y] and direction α[x,y]

3. Apply non-maximal suppression to the gradient magnitude

4. Use double thresholding (and subsequent connectivity analysis) to detect link edges

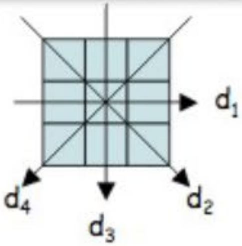
Canny Edge Detection

M[i,j] will have large values where the gradient is large. We need to find the local maxima in this array to locate the edges.

Must be thin so only points of the greatest local change remain.

Canny Edge Detection

For a 3x3 region quantize α to four directions $\zeta[i,j] = \{d_1, d_2, d_3, d_4\}$



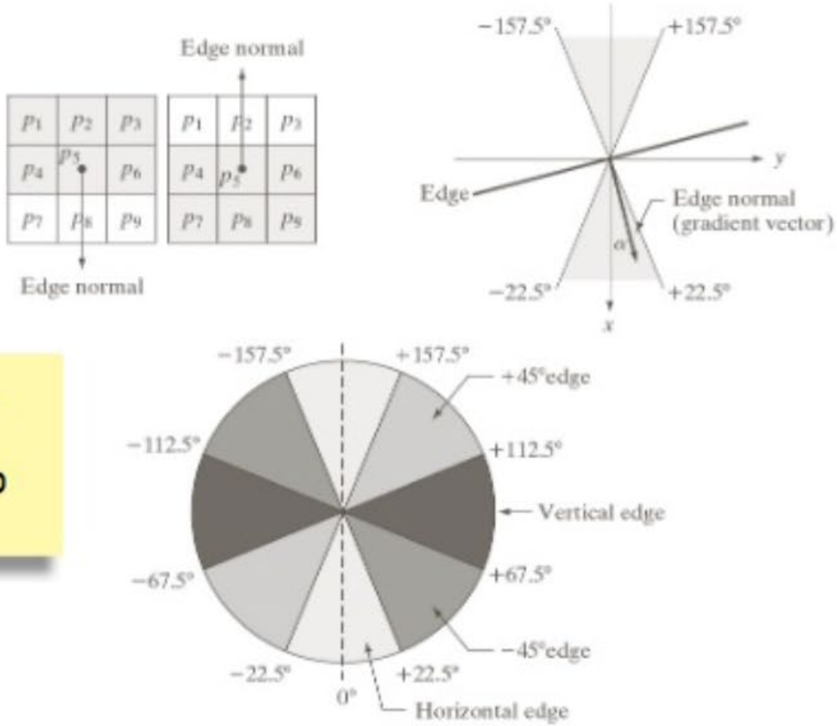
1. Pick the d_i which is closest to $\alpha[x,y]$
2. If $M[x,y]$ is less than one of its two neighbors along $\alpha[x,y]$ then $g_N[x,y]=0$ [suppress a non-maximum] else $g_N[x,y]=M[x,y]$

NOTE: Resulting contours may still be be multiple-pixels thick requiring use of a thinning algorithm

Denote the entire process $N[i,j]=Non\_maximal\_suppression\{M[I,j], \zeta[i,j]\}$

Canny Edge Detection

Each direction $\{d_1, d_2, d_3, d_4\}$ actually corresponds to two edge directions



a b
c

FIGURE 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a 3 × 3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3 × 3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

Canny Edge Detection

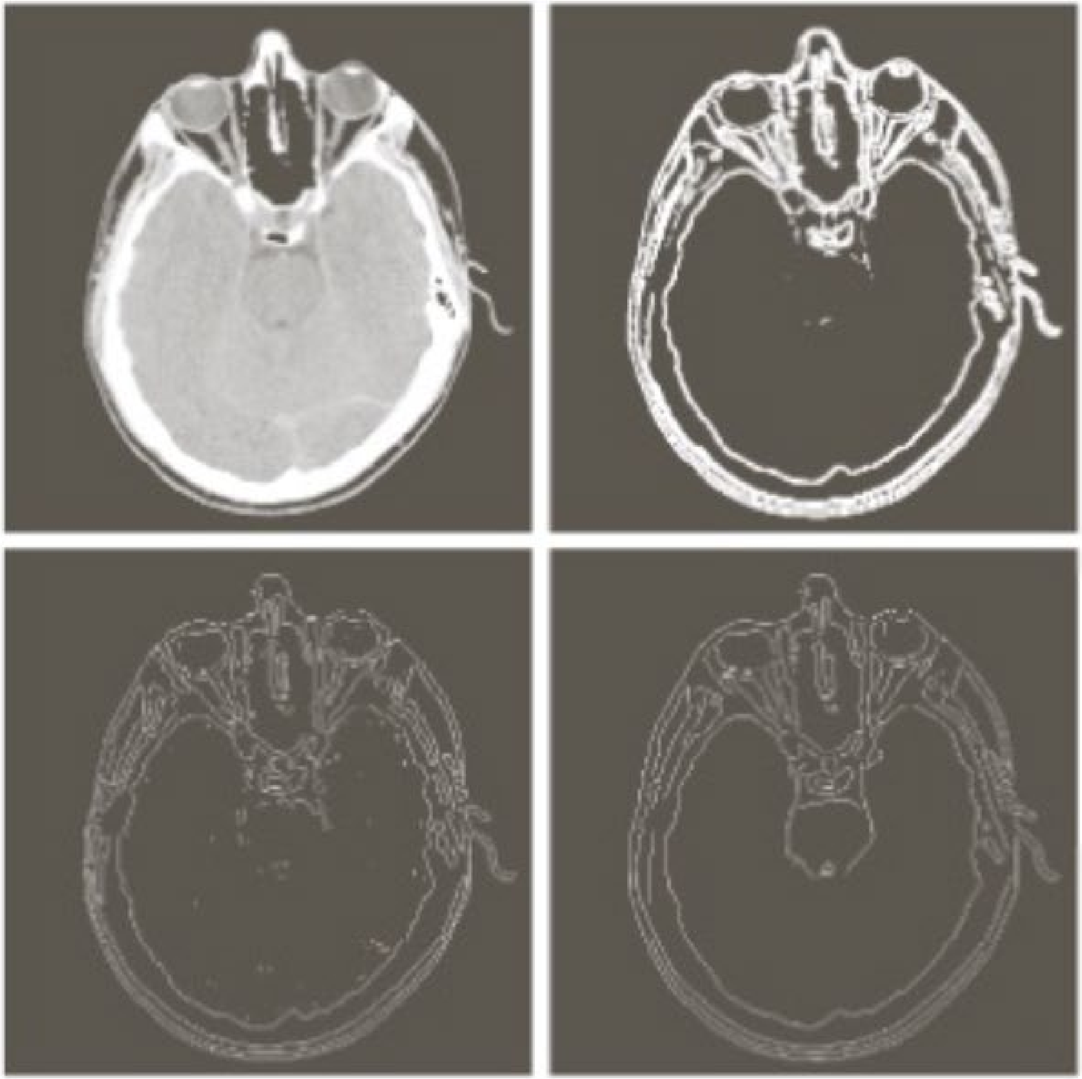
- After non-maximal suppression image contains many false edge fragments caused by noise and fine texture
- You can threshold N[i,j], but good results are difficult to achieve with a single threshold T.
- Use two thresholds T_1 and T_2. Initially link contours using threshold T_1. If a gap is encountered drop to threshold T_2 until you rejoin a T_1 contour.

Canny Edge Detection

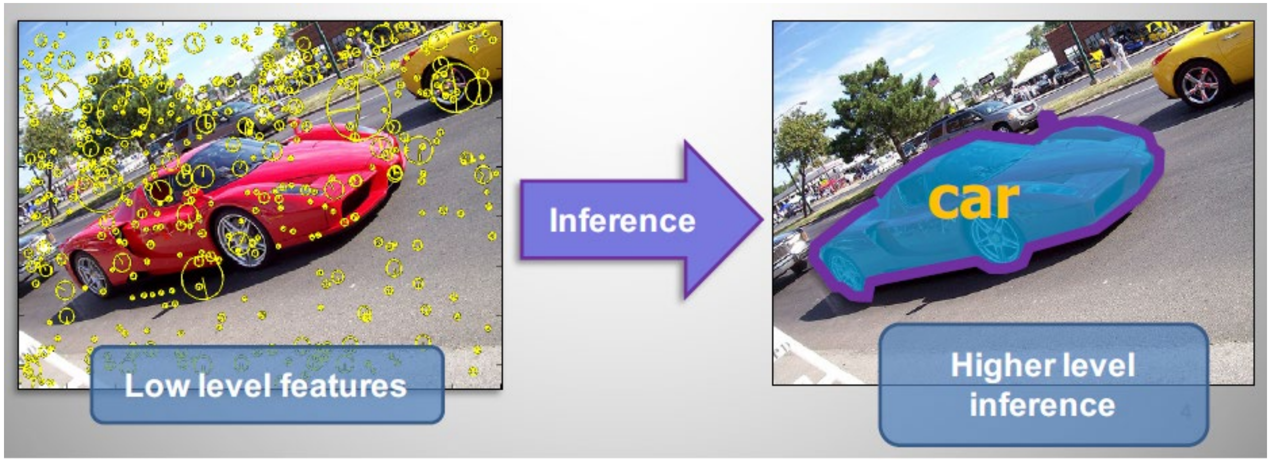
(a) Original image

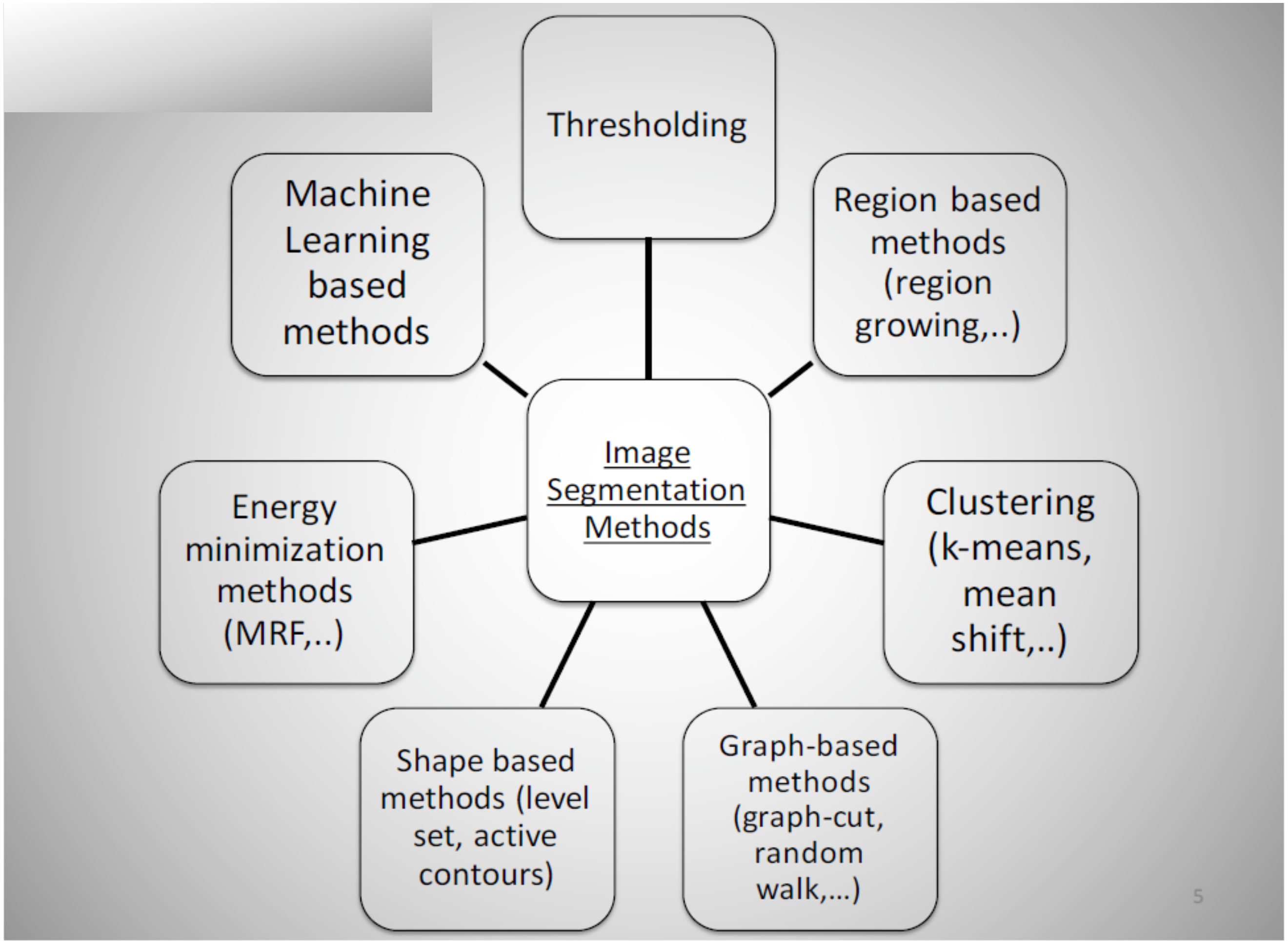
(b) Canny, σ=1.0, T_1=255, T_2=1

Edge Detection



- In computer vision, image segmentation is one of the oldest and most widely studied problems
- – *Early techniques -> region splitting or merging*
- – *More recent techniques -> Energy minimization*



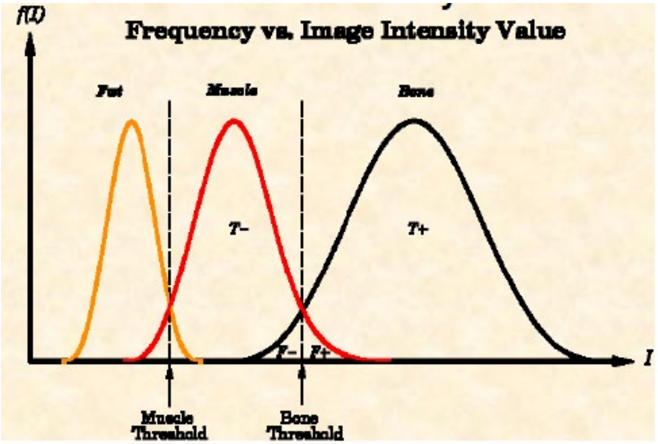

Thresholding
Machine Learning based methods
Region based methods (region growing,..)
Image Segmentation Methods
Energy minimization methods (MRF,..)
Clustering (k-means, mean shift,..)
Shape based methods (level set, active contours)
Graph-based methods (graph-cut, random walk,...)
5

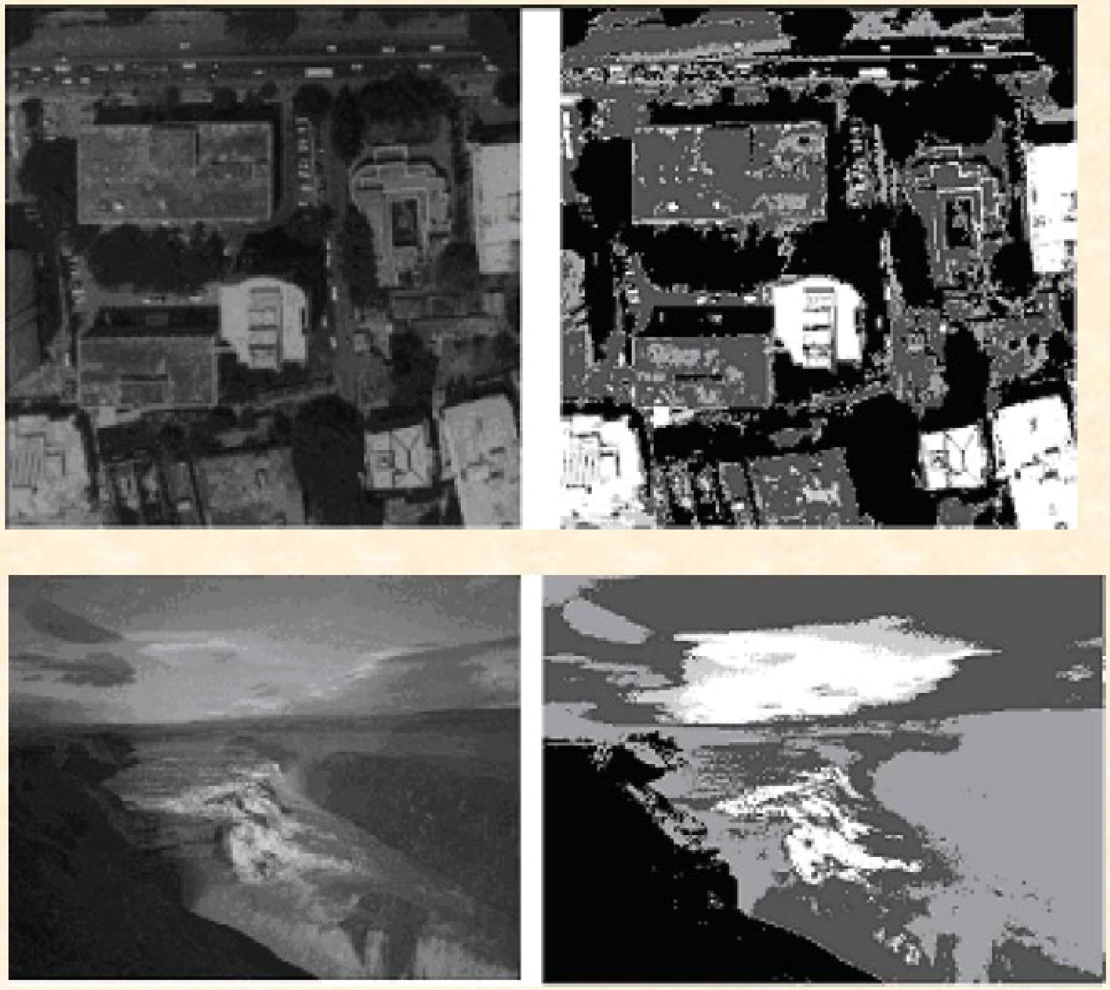
- Thresholding is the simplest way to perform segmentation, and it is used extensively in many image processing applications. Thresholding is based on the notion that regions corresponding to different regions can be classified by using a range function applied to the intensity values of image pixels. The assumption is that different regions in an image will have a distinct frequency distribution and can be discriminated on the basis of the mean and standard deviation of each distribution (see Figure).
- For example, given the histogram of a two-dimensional medical image I(x,y) , we can define a simple threshold rule to classify bony and fat tissues or a compound threshold rule to classify muscle tissue:

  If, $I(x,y) > T1$ => Bony

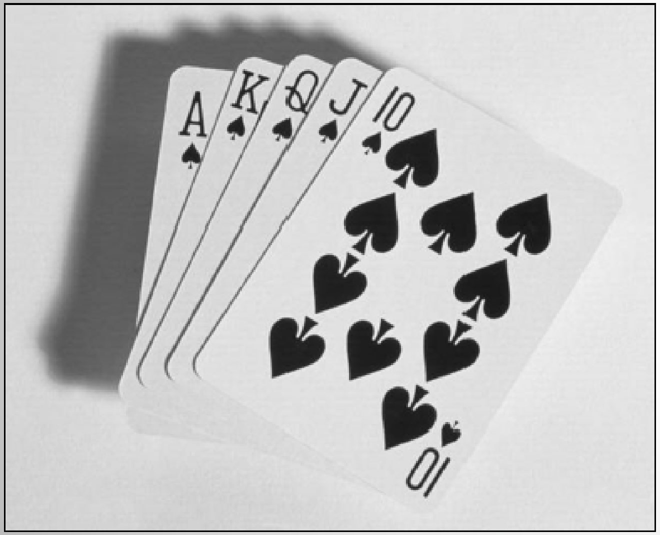
  If, $I(x,y) < T0$ => Fat

  If, $T0 < I(x,y) < T1$ => Muscle

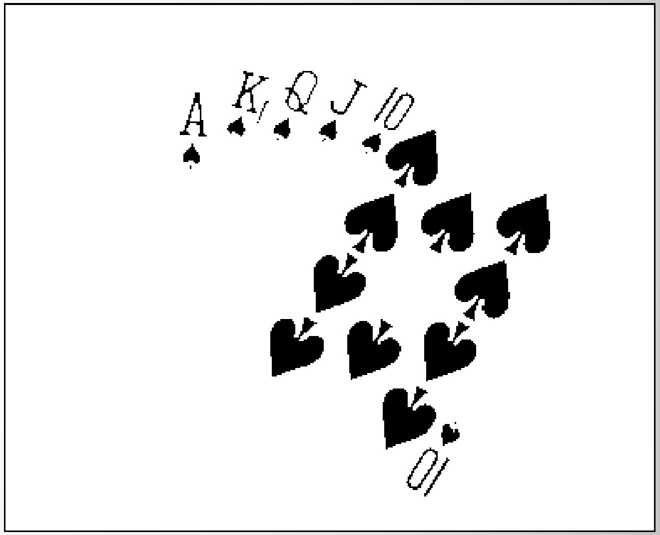




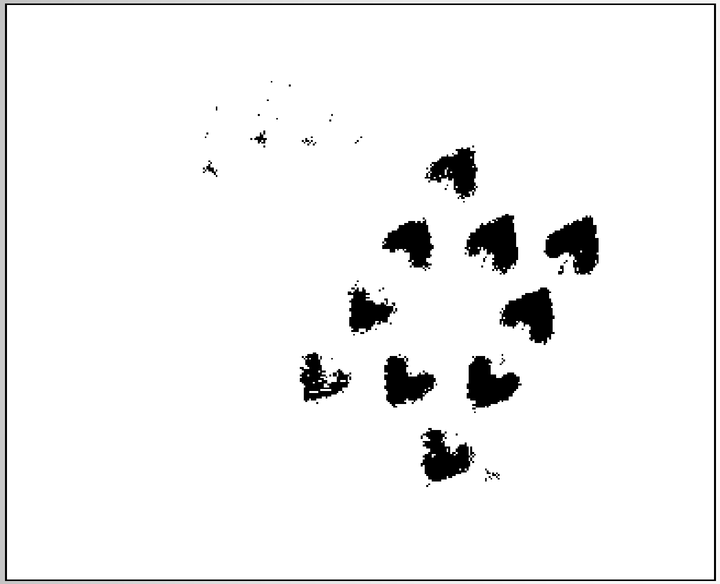
Two examples of gray level thresholding based segmentation



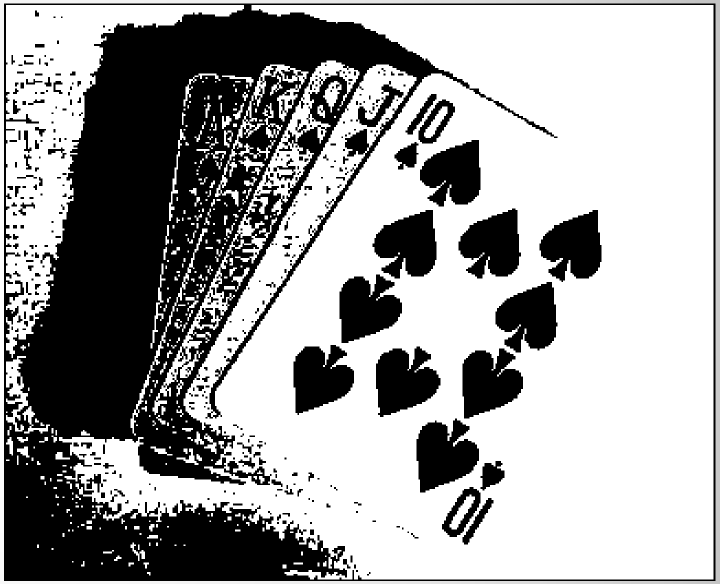
Original Image



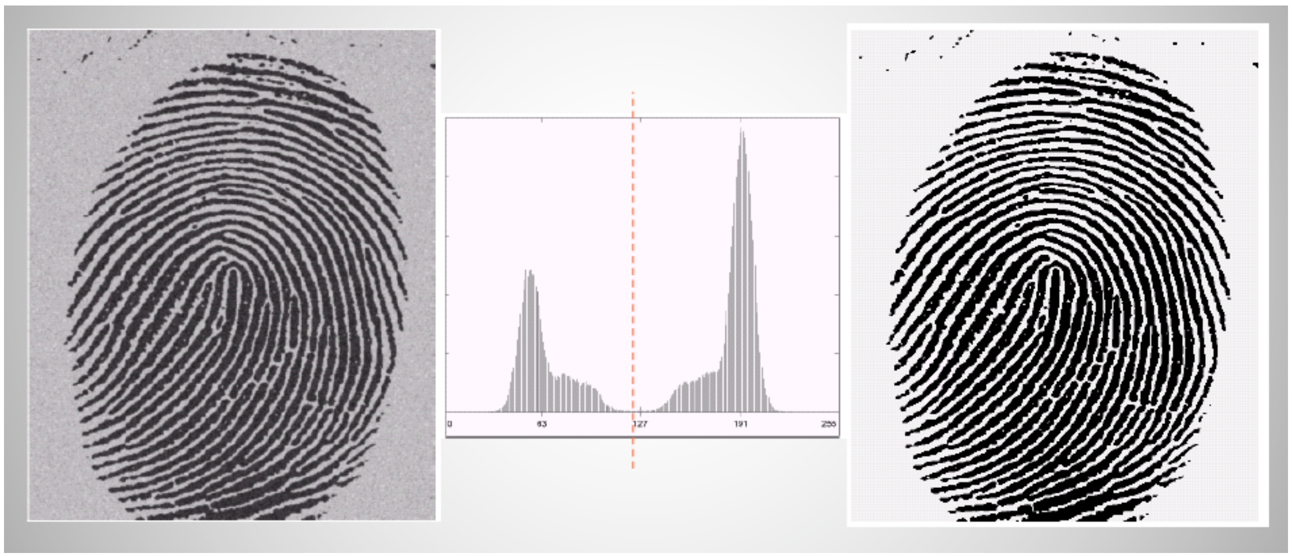
Thresholded Image



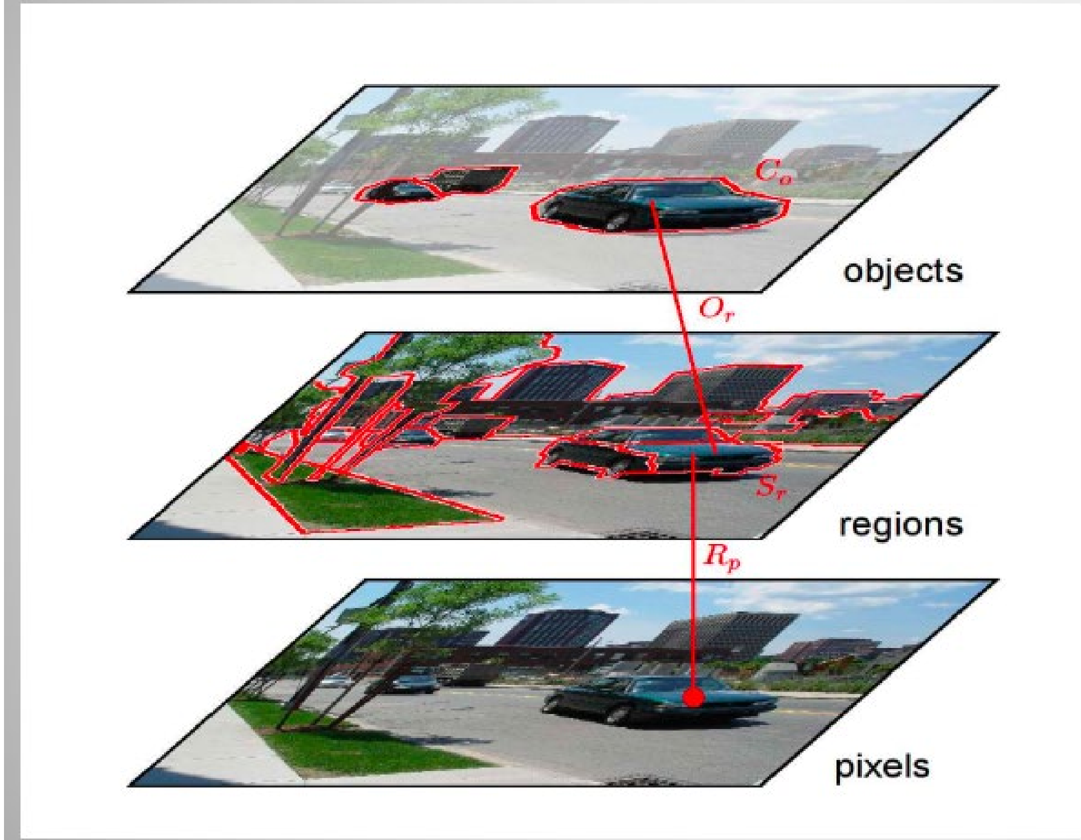
Threshold Too Low



Threshold Too High


0
63
127
191
255

Region Based Segmentation



Region:
A group of connected pixels with similar properties

Closed boundaries

Computation of regions is based on similarity

Regions may correspond to Objects in a scene or parts of objects

Spatial proximity + similarity

Region Growing

- For segment generation in grey-level or color images, we may start at one *seed pixel* (x,y,I(x,y)) and add recursively adjacent pixels that satisfy a "similarity criterion" with pixels contained in the so-far grown region around the *seed pixel.*

- Defining similarity criteria alone is not an effective basis for segmentation
- It is necessary to consider the adjacency spatial relationship between pixels

Algorithm

1. The absolute intensity difference between candidate pixel and the seed pixel must lie within a specified range
2. The absolute intensity difference between a candidate pixel and the running average intensity of the growing region must lie within a specified range;
3. The difference between the standard deviation in intensity over a specified local neighborhood of the candidate pixel and that over a local neighborhood of the candidate pixel must (or must not) exceed a certain threshold

25

Seeded Segmentation (Region Growing)

1. Choose the Seed pixel

2. Check the neighboring pixels and add them to the region if they are similar to the seed

3. Repeat step 2 for each of the newly added pixels; stop if no more pixels can be added

0	0	5	6	7
1	1	5	8	7
0	<u>1</u>	6	<u>7</u>	7
2	0	7	6	6
0	1	5	6	5

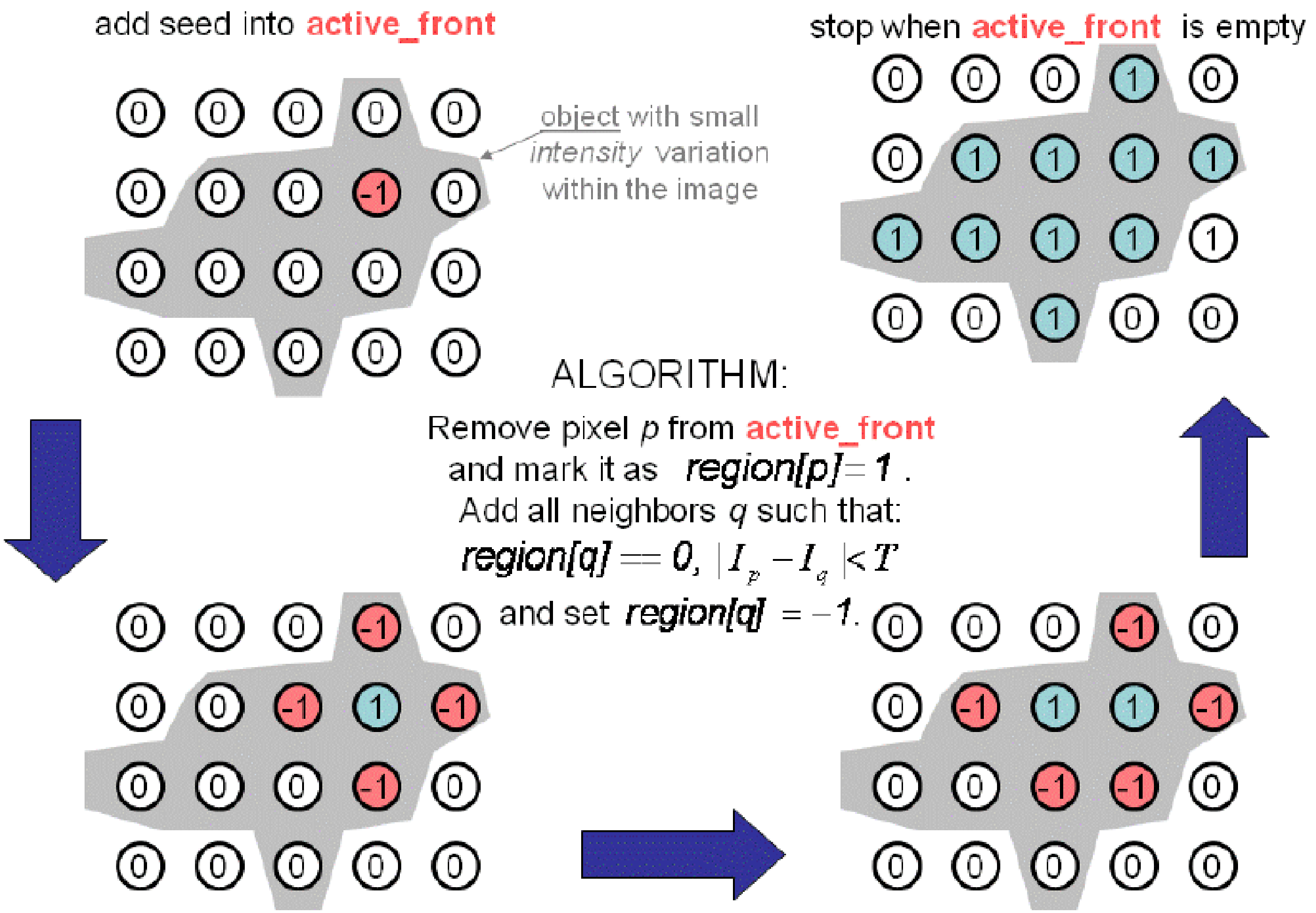
(a)

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

(b)

Region Growing Implementation

growRegion: red nodes are the "active_front" (queue or stack)



Limitations of Region Growing

Note that a complete segmentation of an image must satisfy a number of criteria:

1) All pixels must be assigned to regions

2) Each pixel must belong to a single region only

3) Each region must be a connected set of pixels

4) Each region must be uniform

5) Any merged pair of adjacent regions must be non-uniform

Comparison of Thresholding and Region Growing

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

threshold $T \geq 10$

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

threshold $T \geq 11$

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

threshold $T \geq 12$

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

region growing with variance of 2 in respect to value 11 with reference to threshold $T \geq 11$

Region Splitting and Merging Segmentation

Region splitting:

– Unlike region growing, which starts from a set of seed points, region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.

Region merging:

– Region merging is the opposite of splitting, and works as a way of avoiding over-segmentation

– Start with small regions (2x2 or 4x4 regions) and merge the regions that have similar characteristics (such as gray level, variance).

Region Splitting and Merging Segmentation

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

original image

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 1

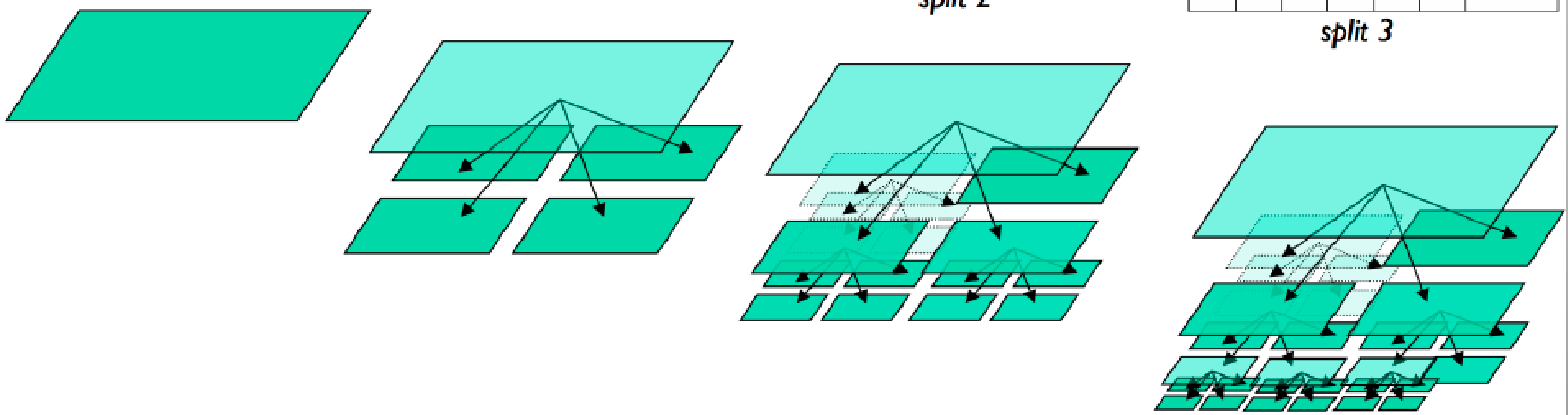
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 2

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 3

http:/



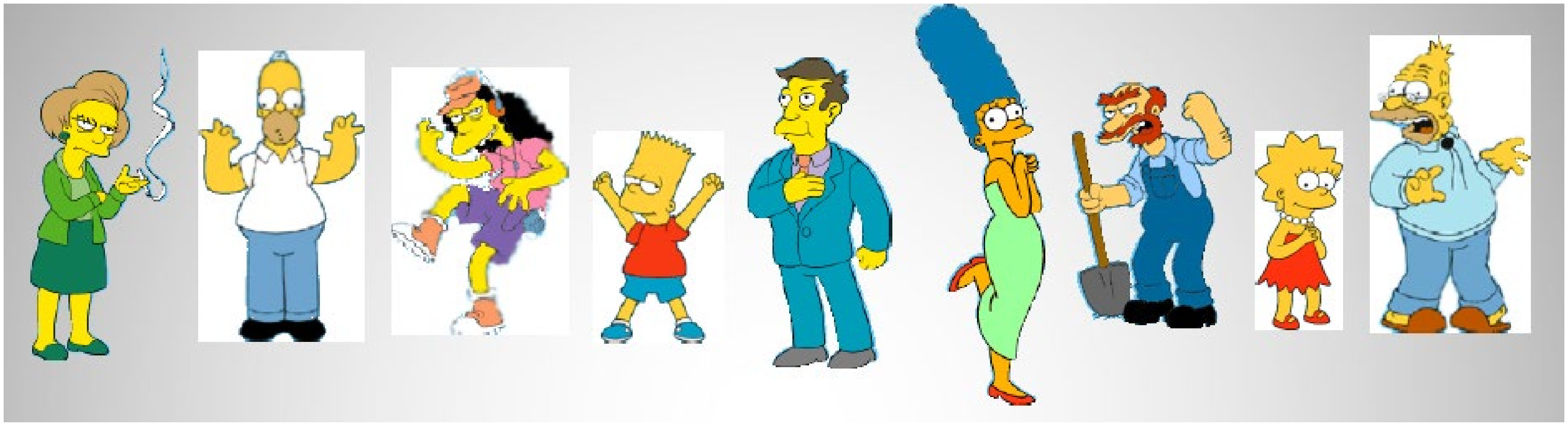
Clustering Based Segmentation

- What is Clustering?

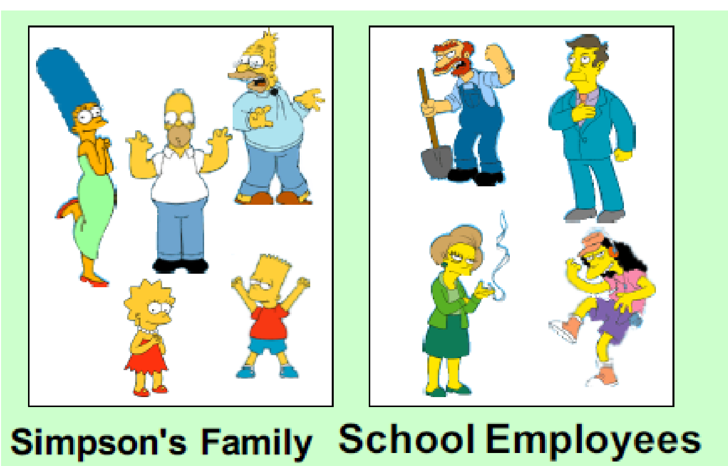
- Organizing data into classes such that:
  - High intra-class similarity
  - Low inter-class similarity

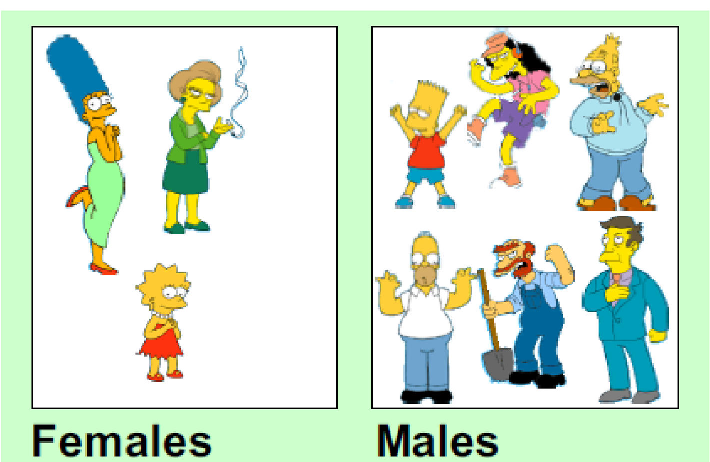
- Finding the class labels and the number of classes directly from the data (as opposed to *classification tasks*)

What is Natural Grouping



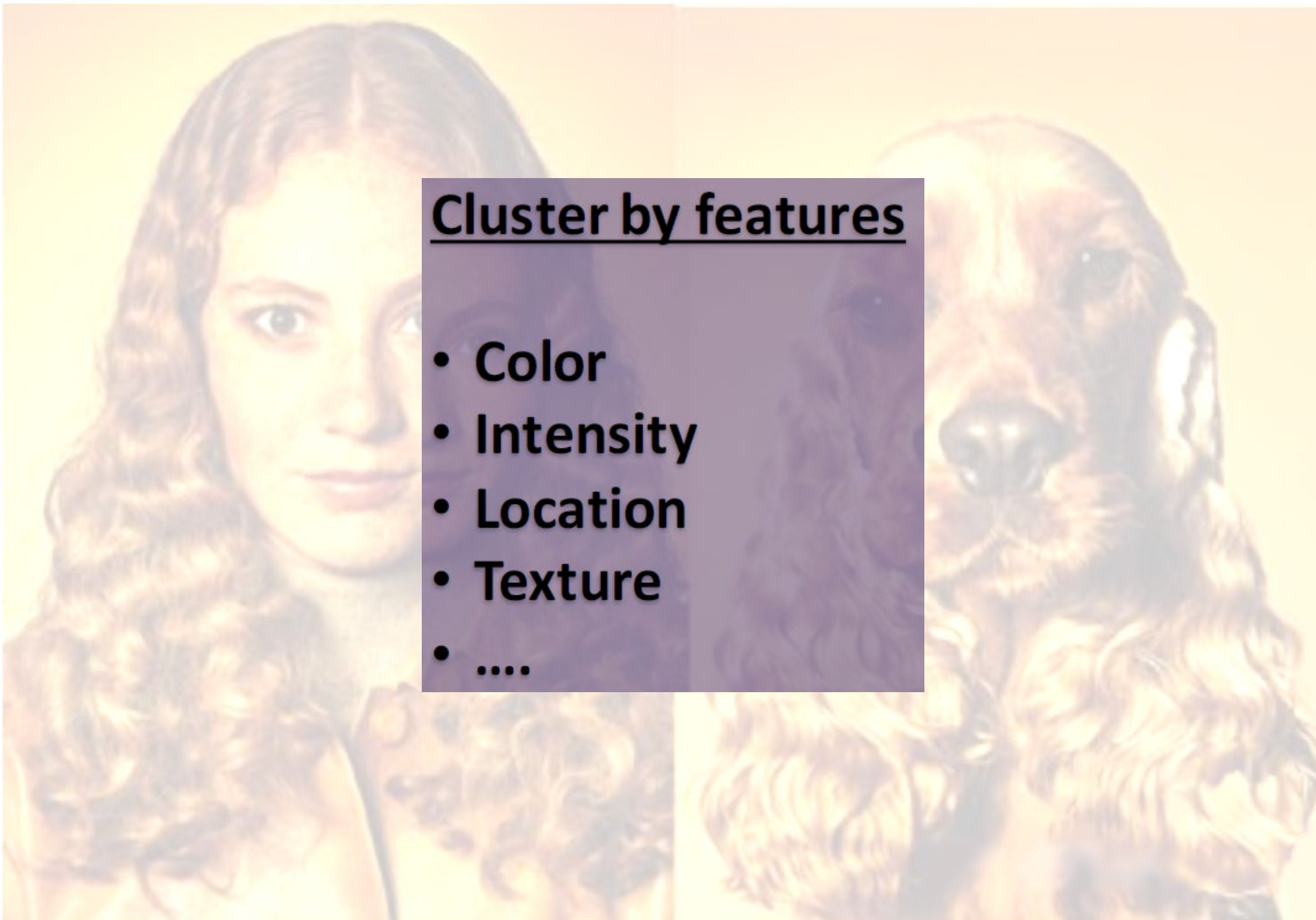
Clustering is Subjective



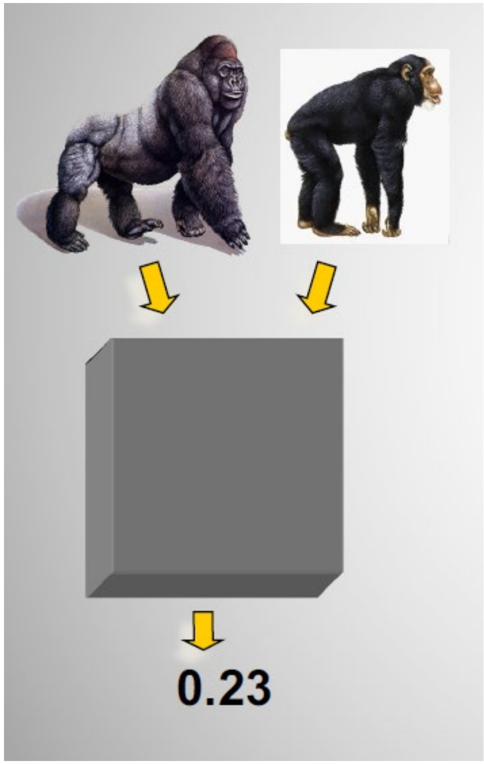


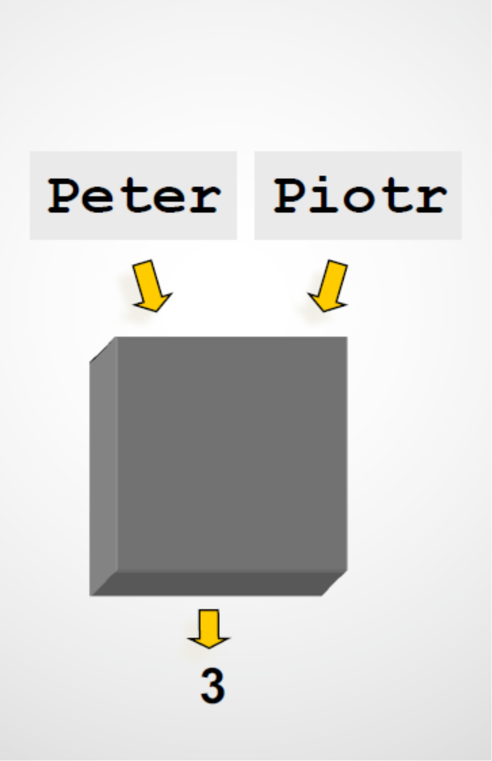
What is Similarity

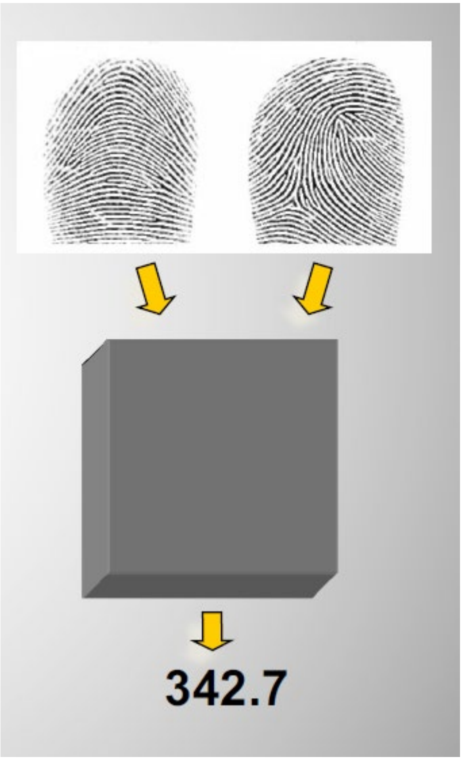
What is Similarity



Distance Metrics



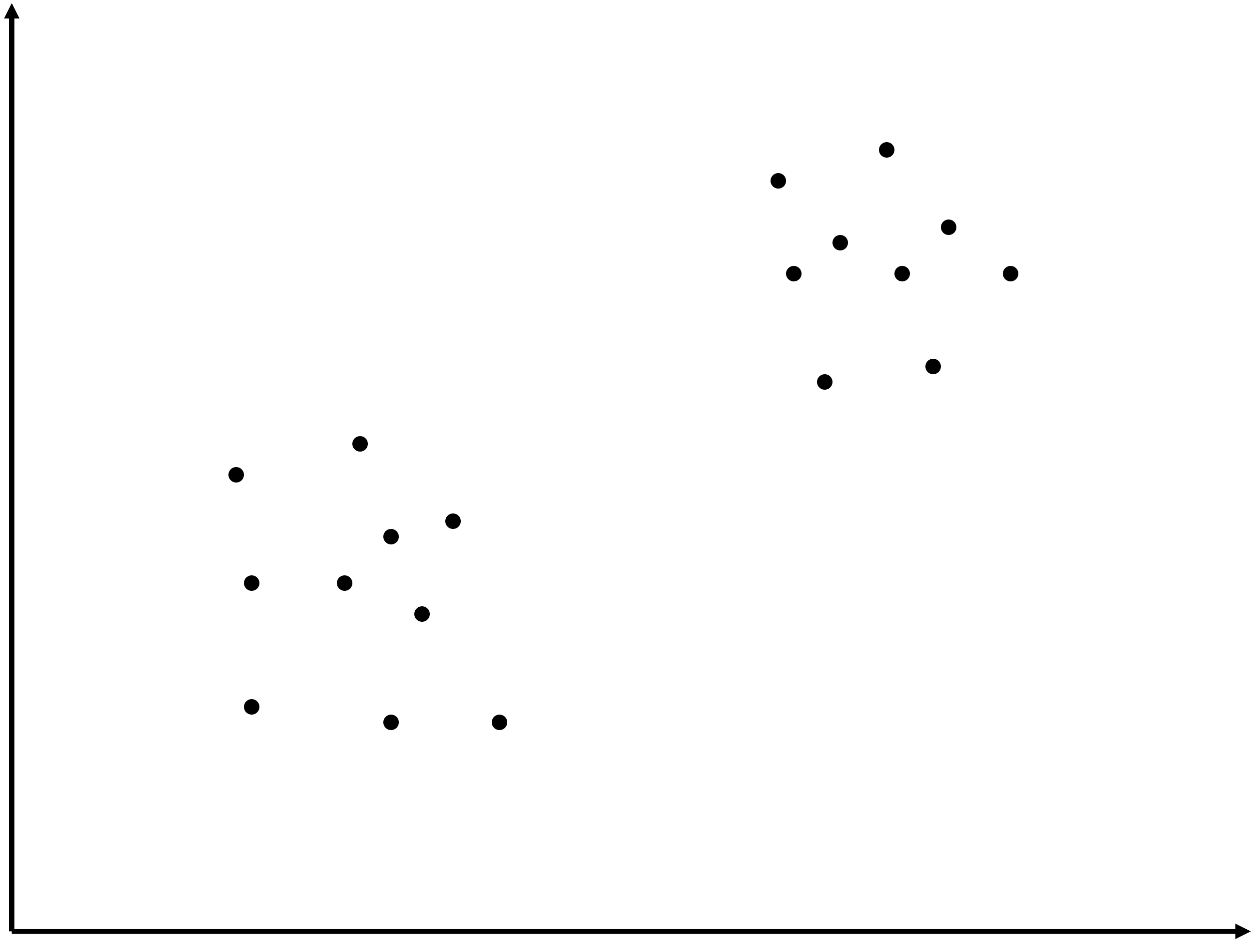




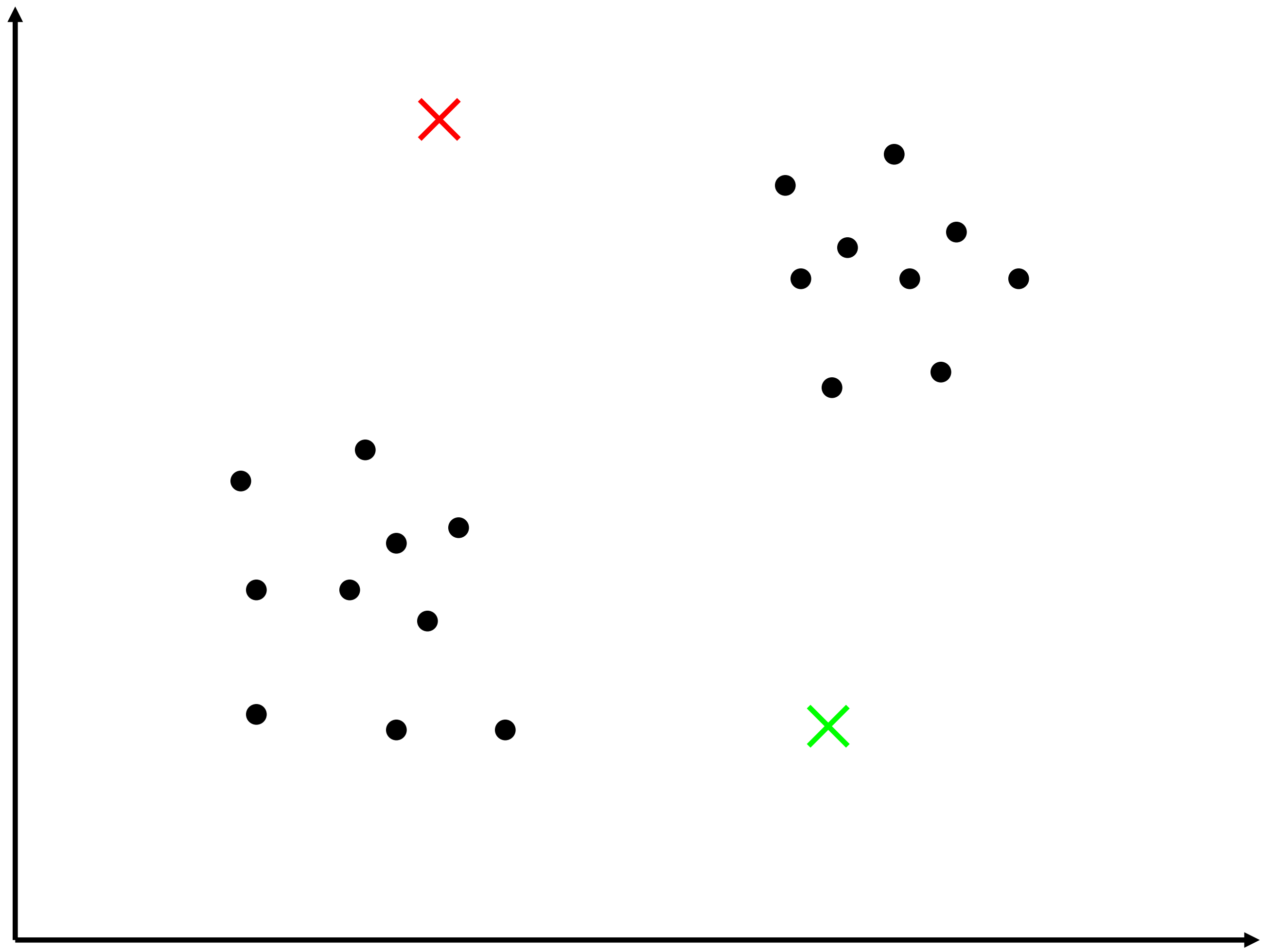
K-Means Clustering

1. Partition the data points into K clusters randomly. Find the centroids of each cluster.
2. For each data point:
   - Calculate the distance from the data point to each cluster.
   - Assign the data point to the closest cluster.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

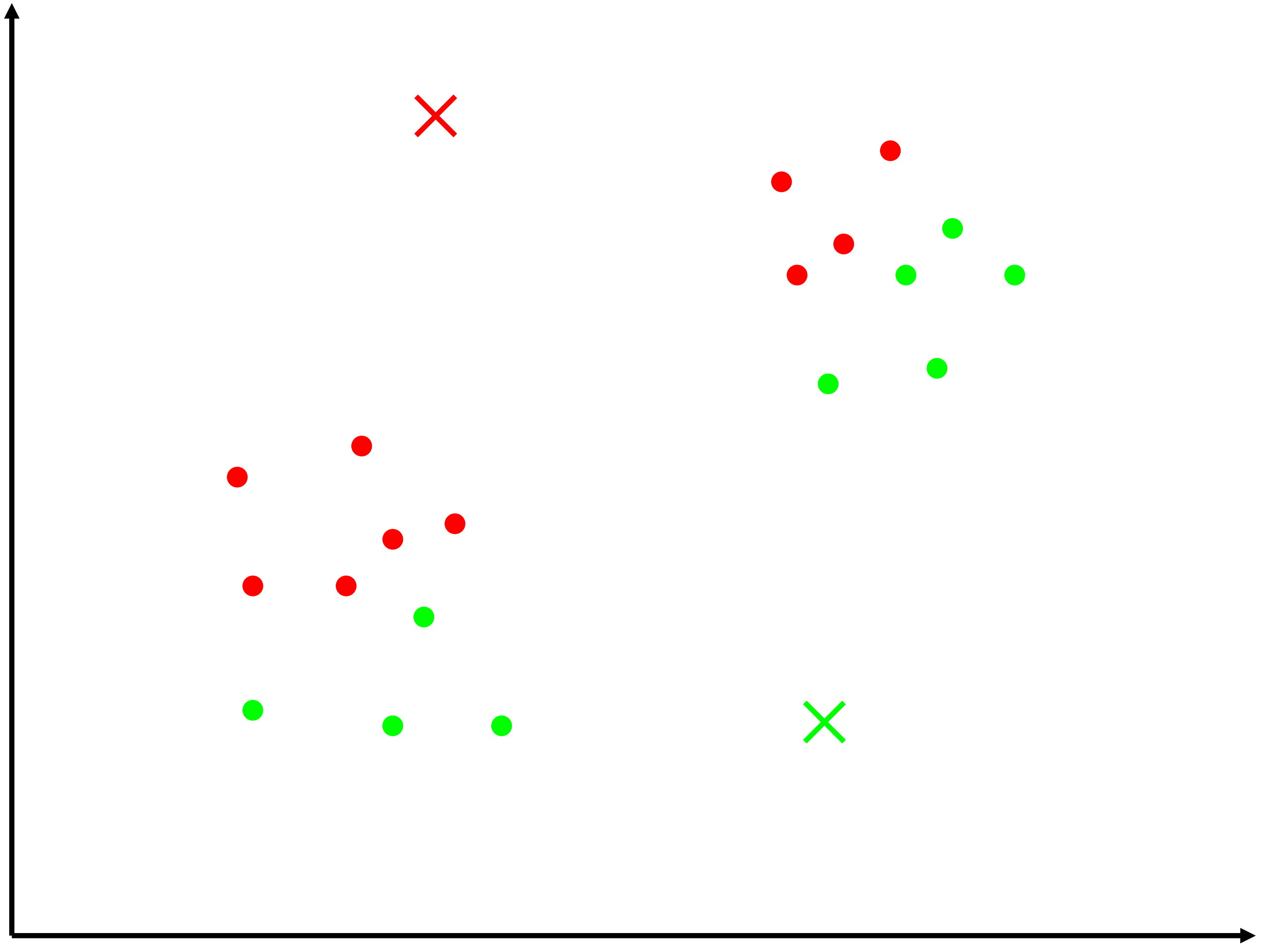
K-Means Clustering



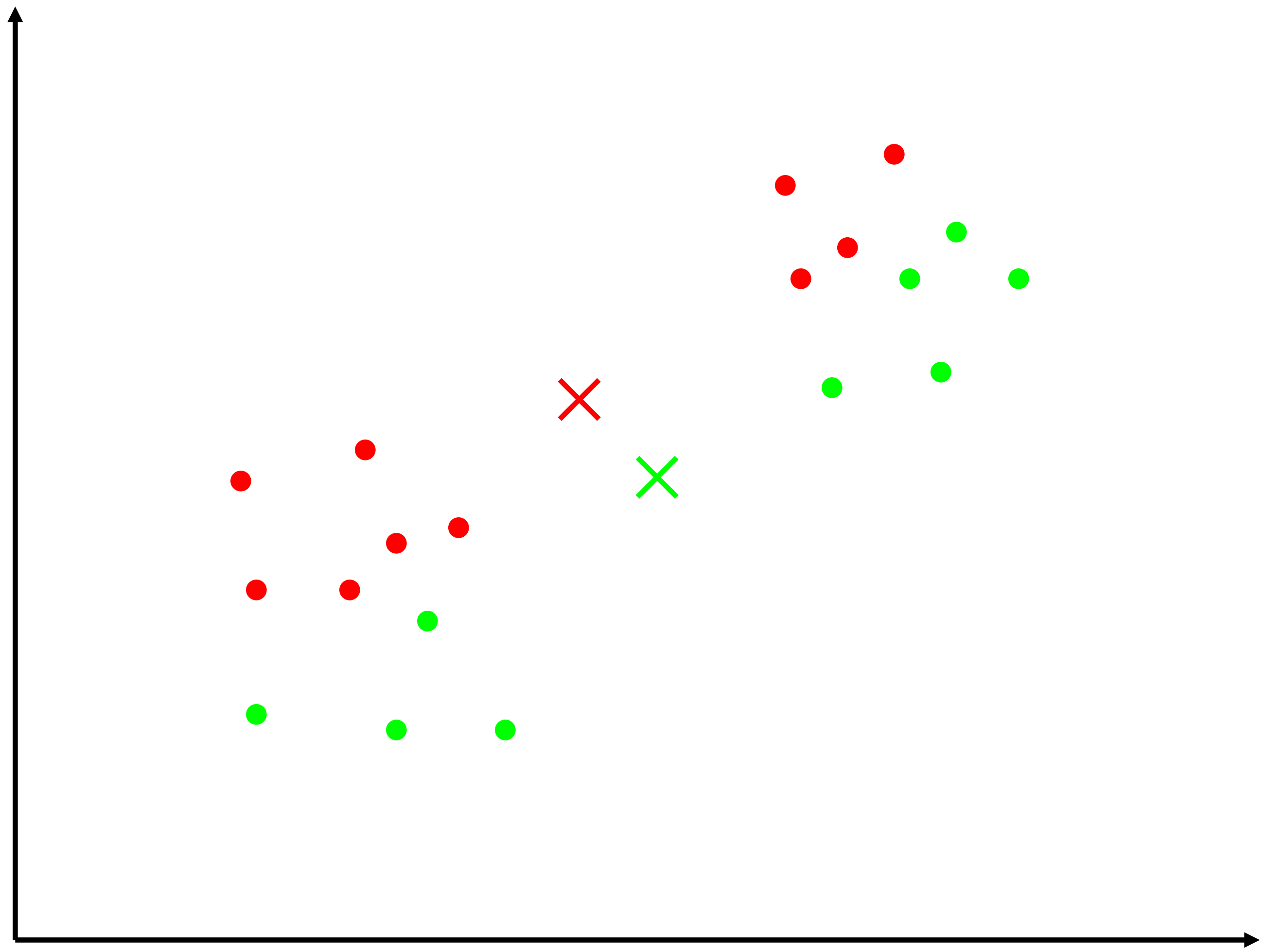
K-Means Clustering



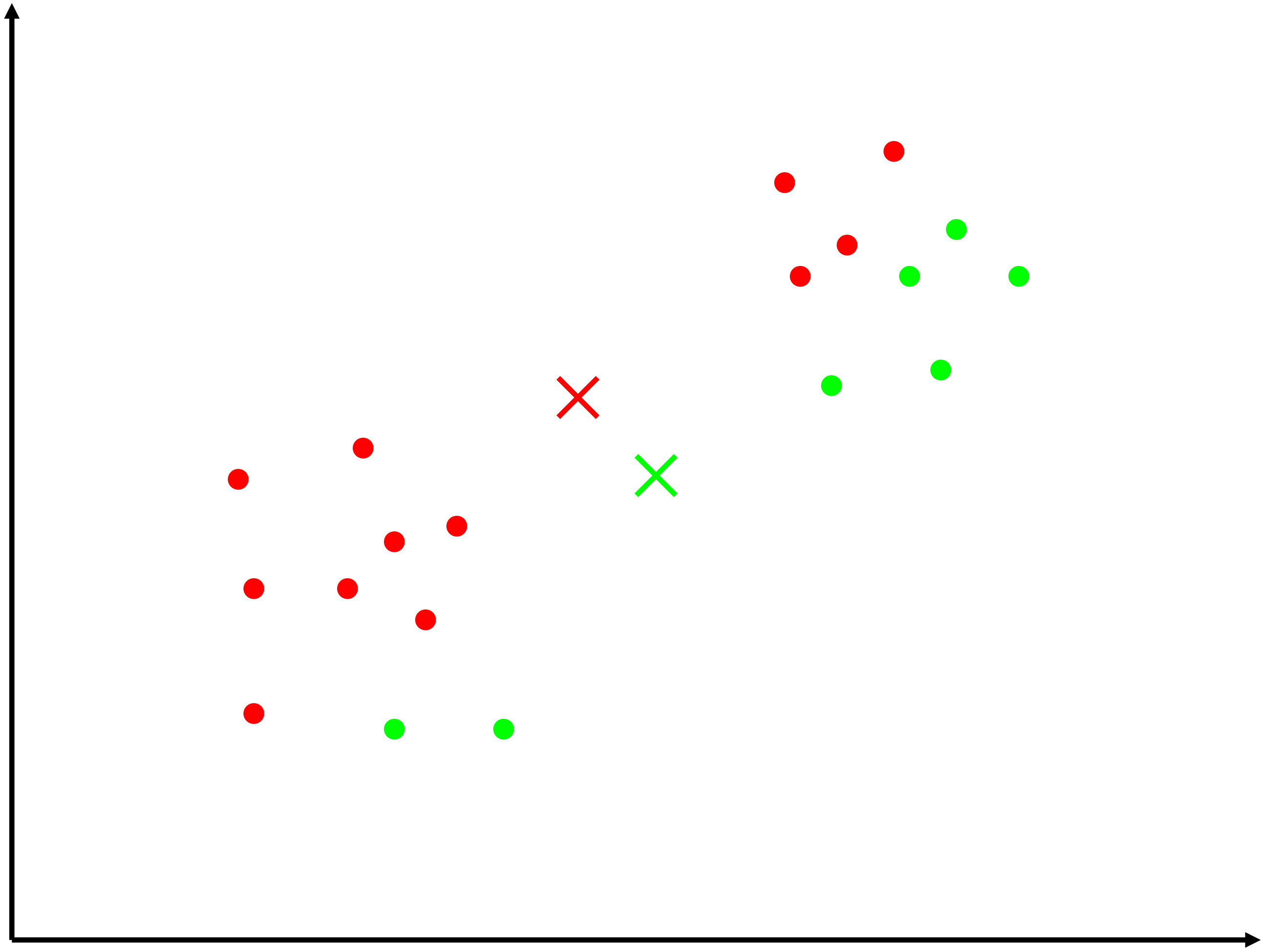
K-Means Clustering



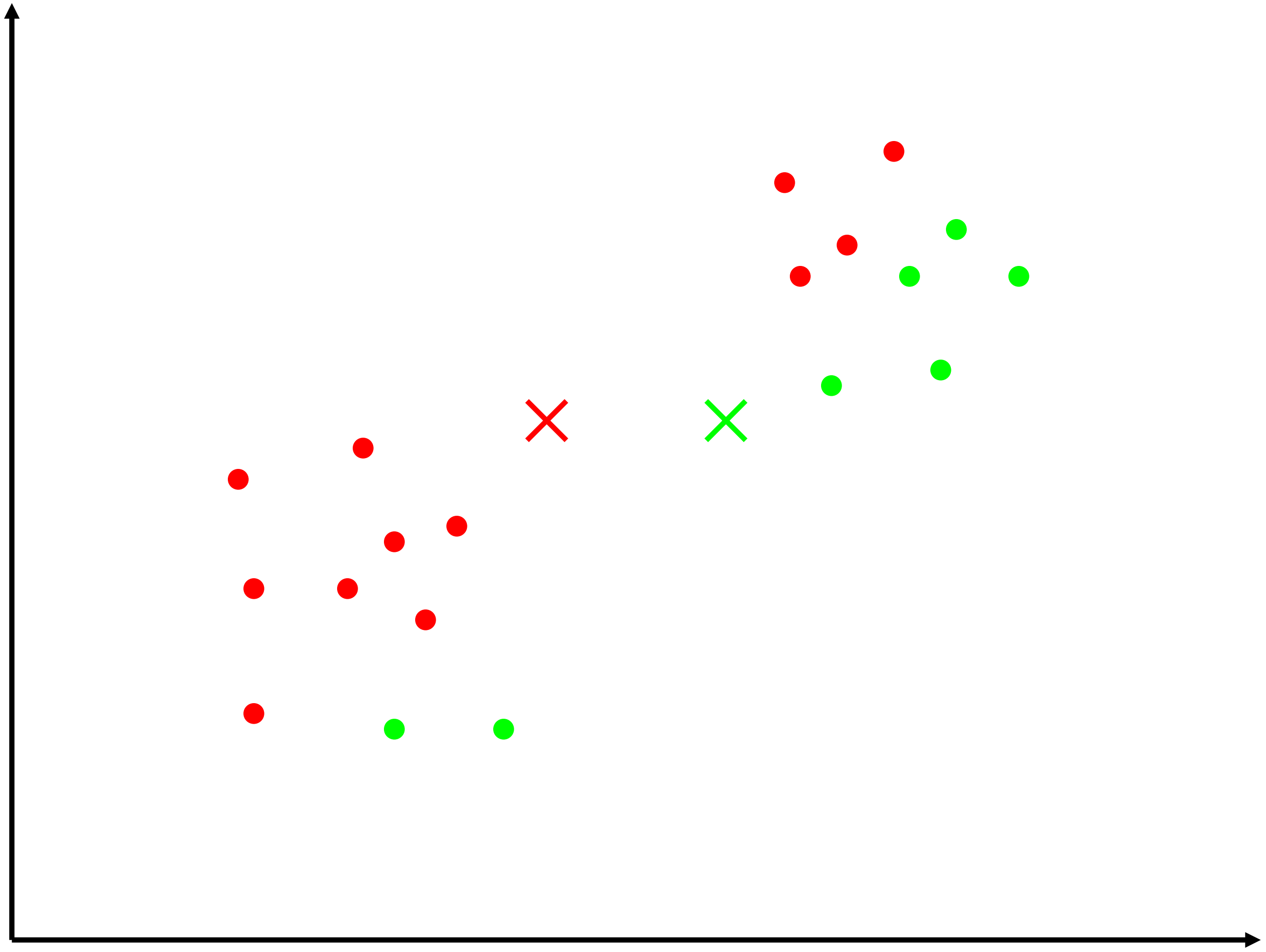
K-Means Clustering



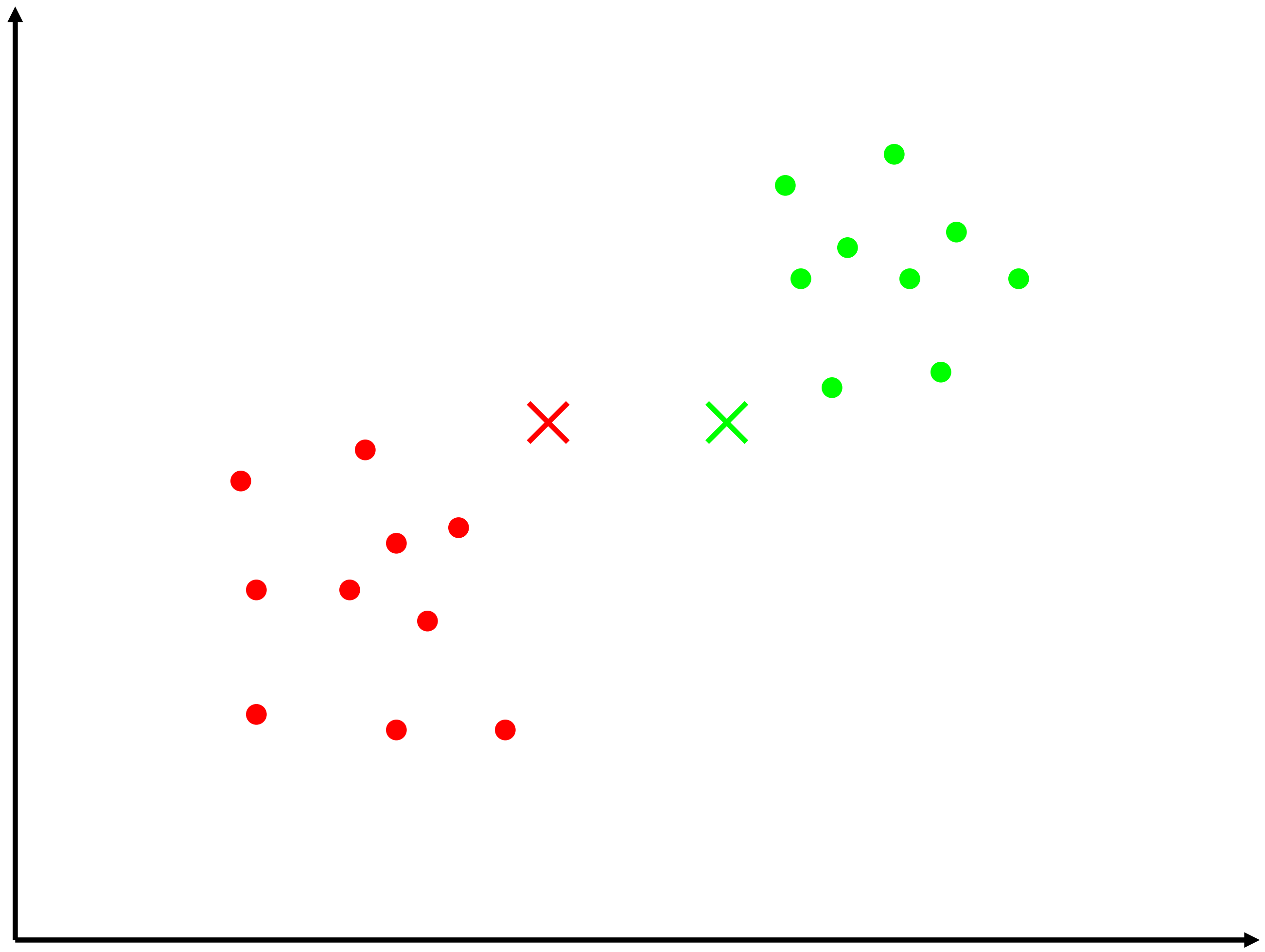
K-Means Clustering



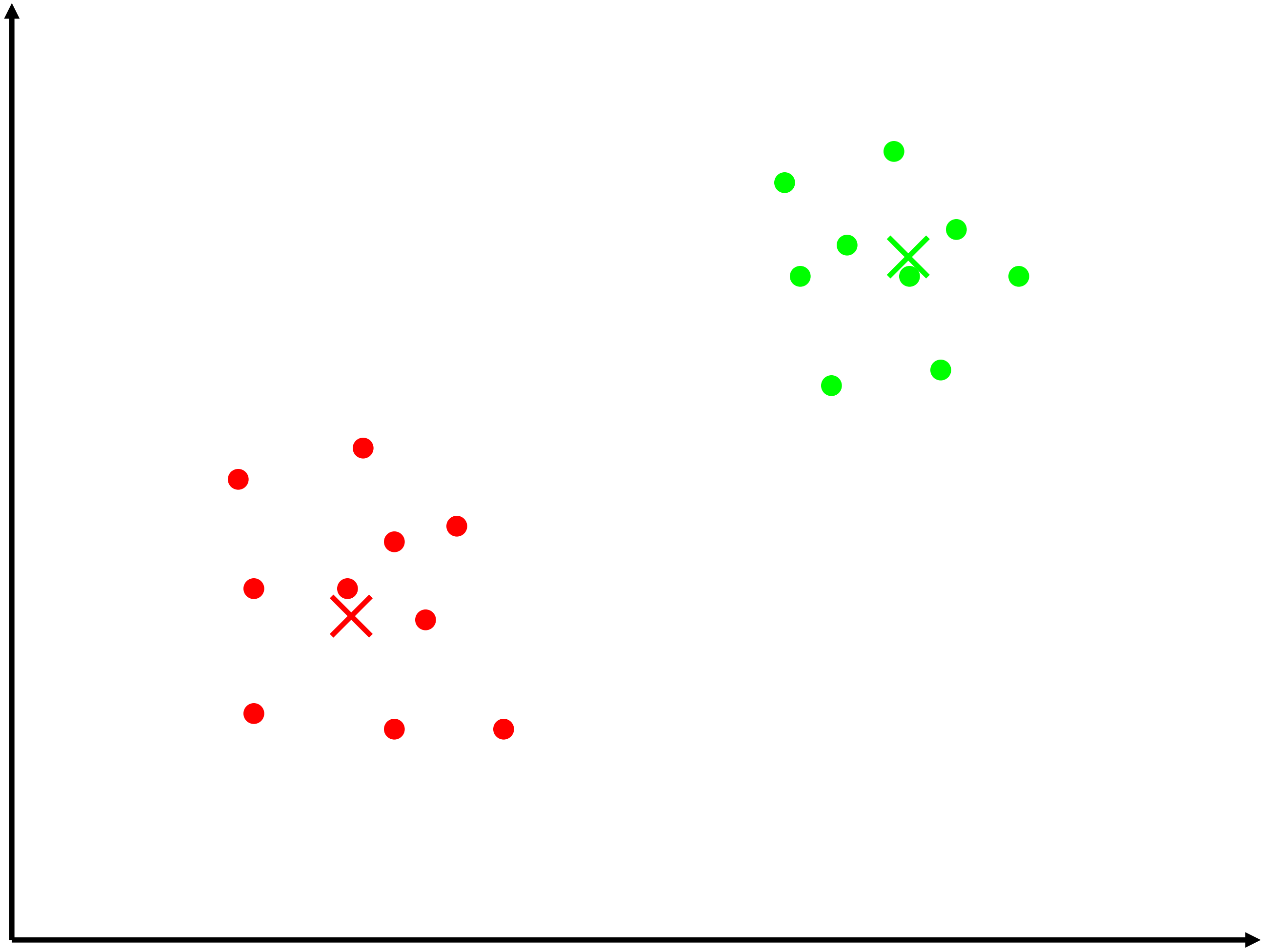
K-Means Clustering



K-Means Clustering

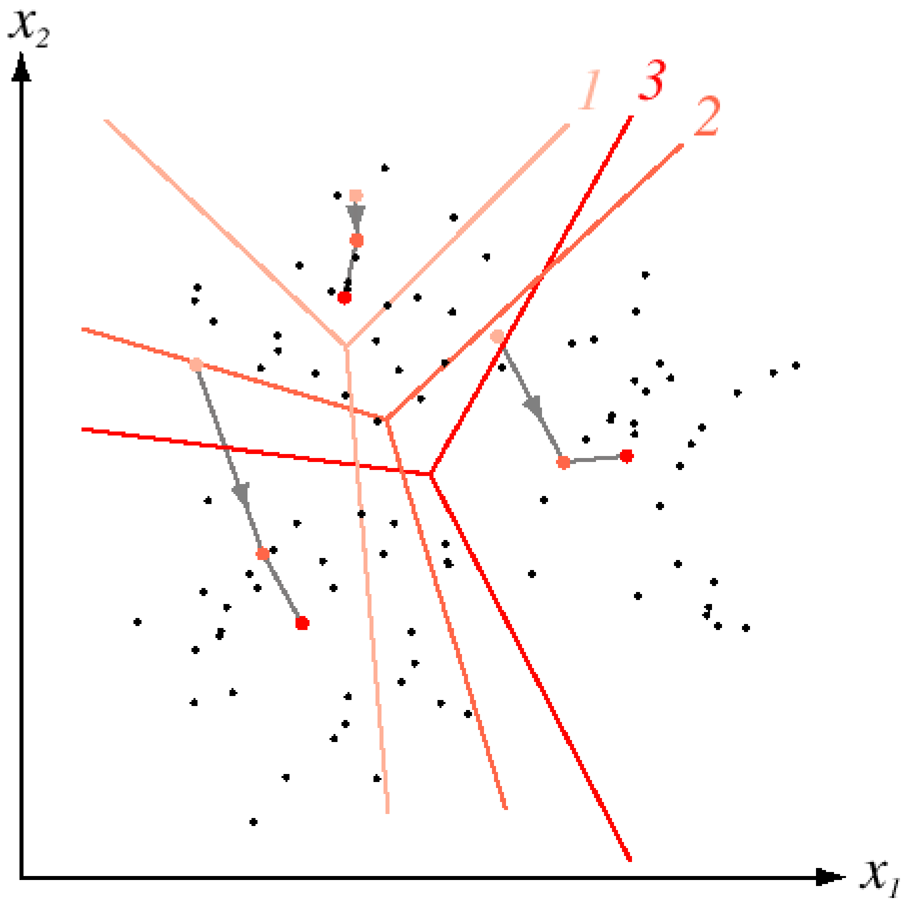


K-Means Clustering



K-Means Clustering

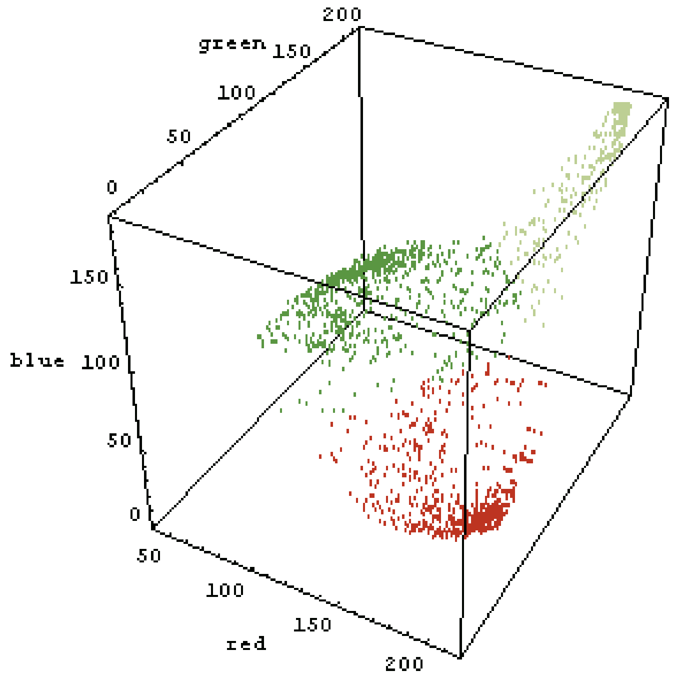
- Example



Duda et al.

K-Means Clustering

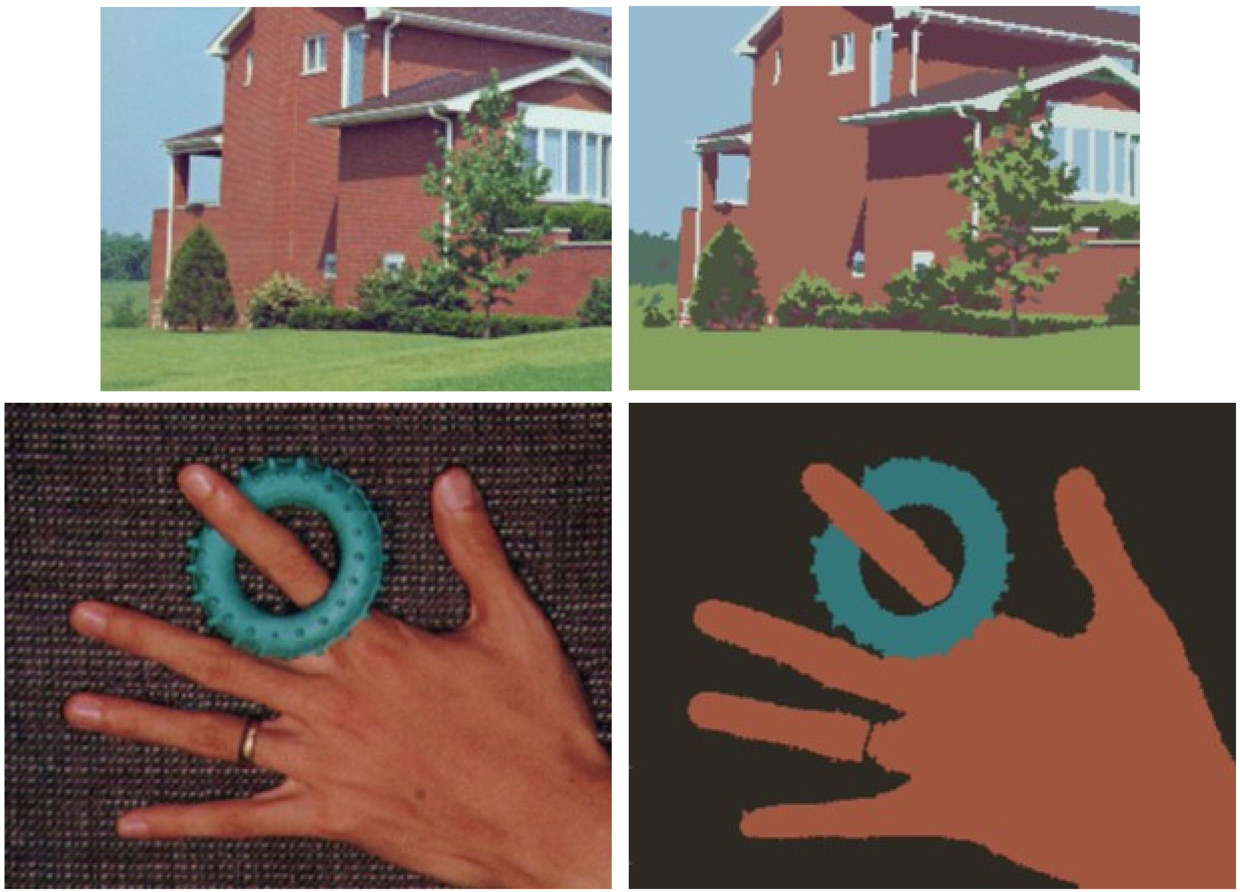
- RGB vector



K-means clustering minimizes $$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

Clustering

- Example



D. Comaniciu and P. Meer, *Robust Analysis of Feature Spaces: Color Image Segmentation*, 1997.

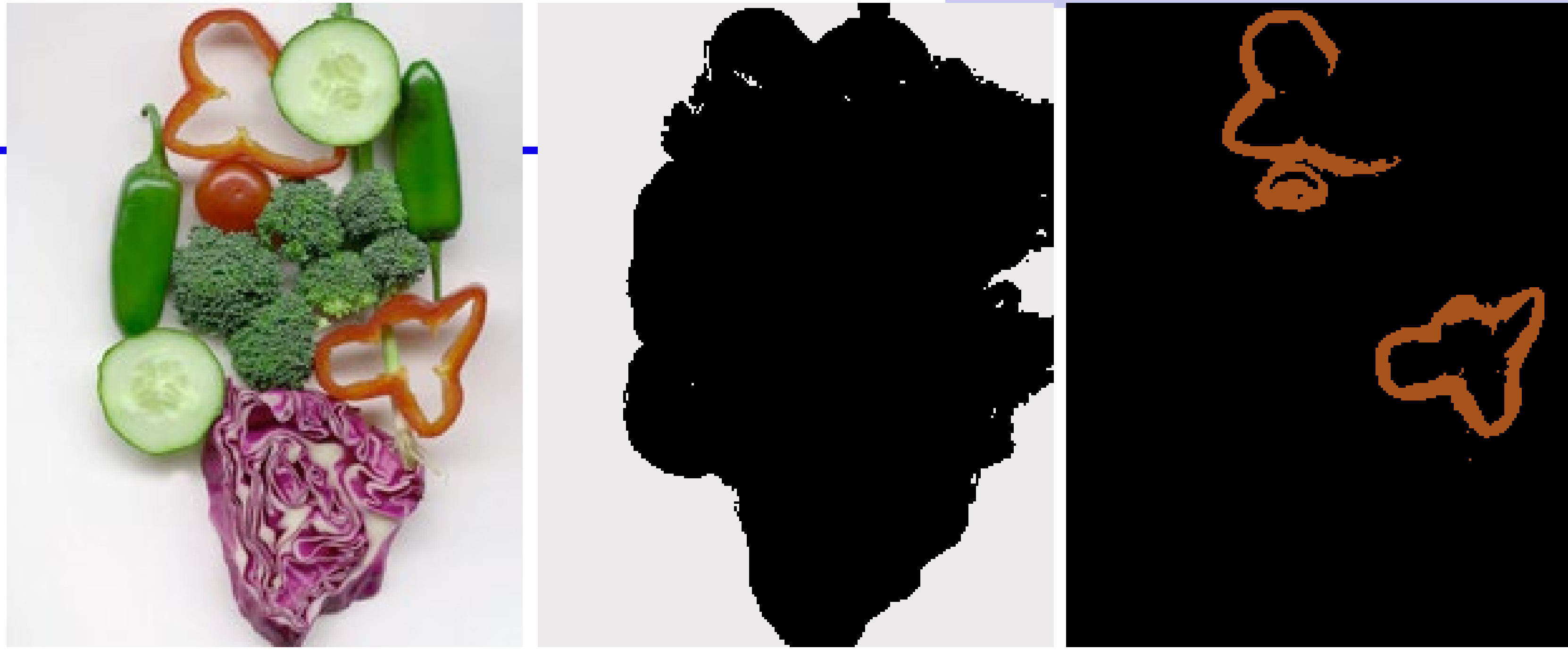
K-Means Clustering

- Example

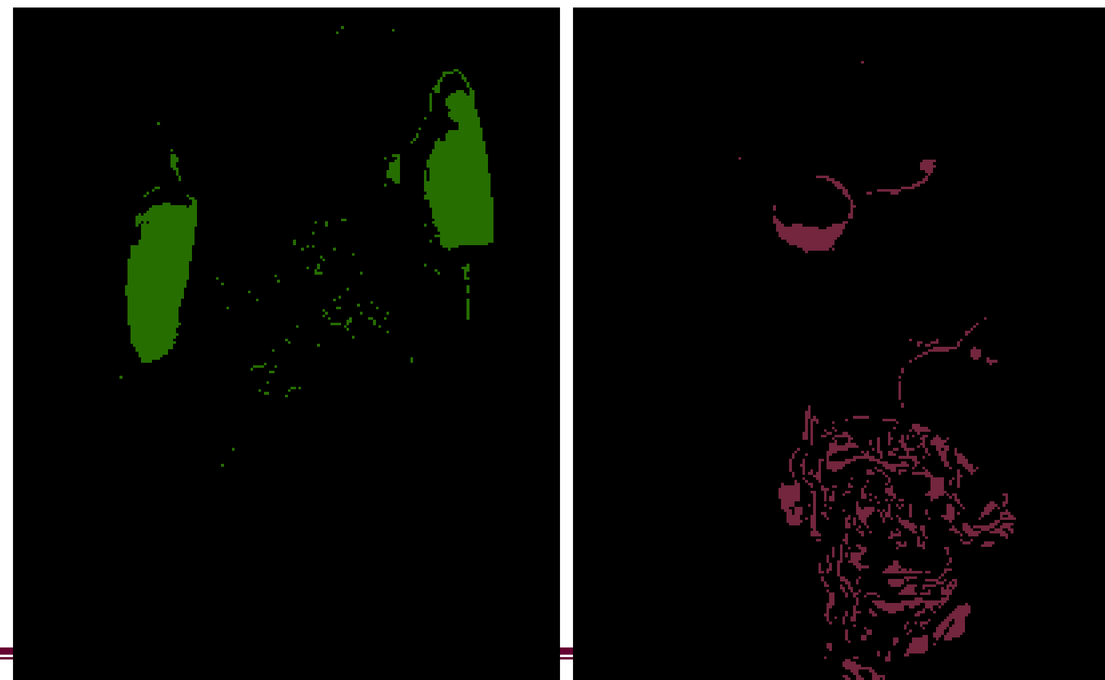
Original

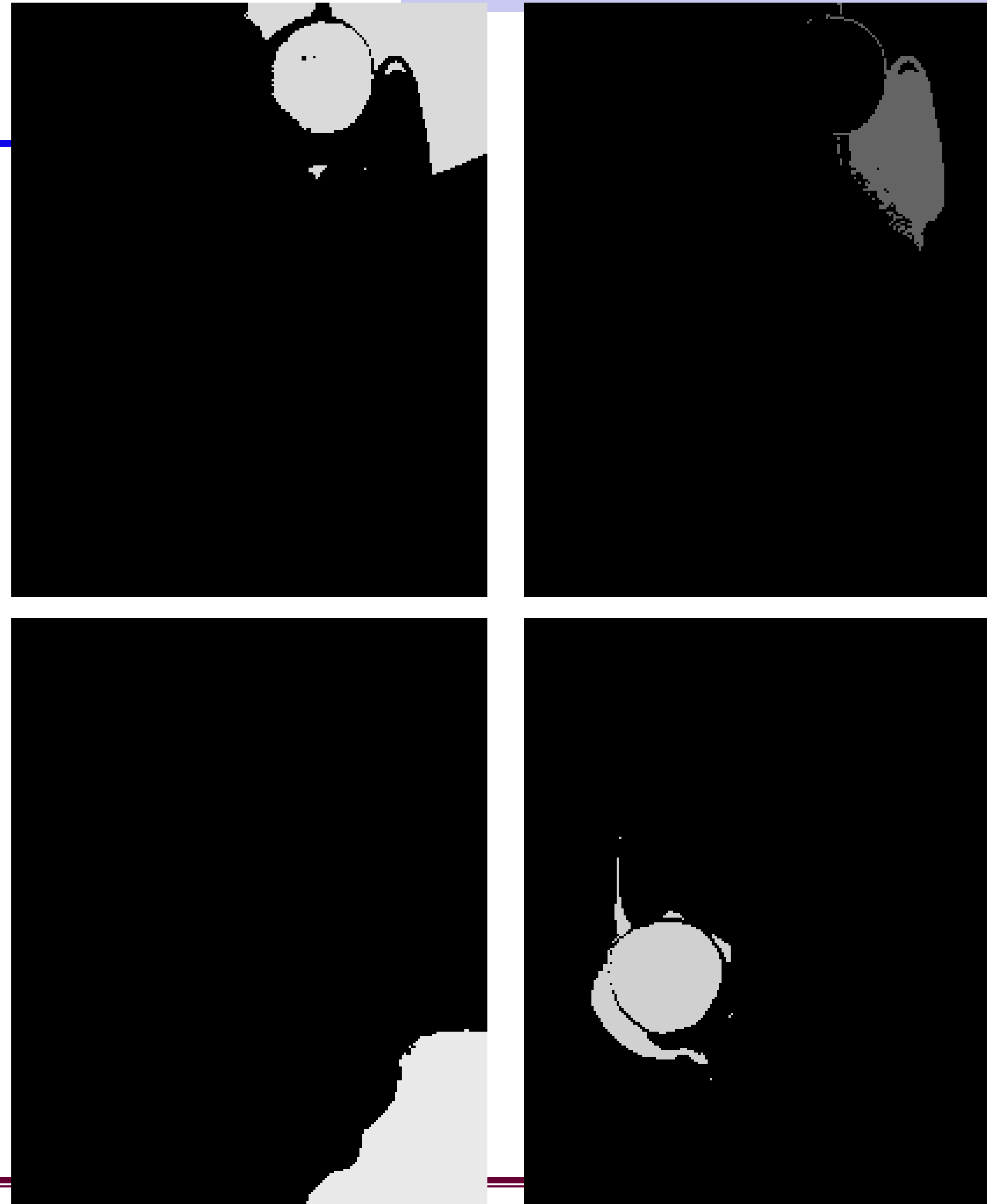
K=5

K=11



K-means, only color is used in segmentation, four clusters (out of 20) are shown here.





K-means, color and position is used in segmentation, four clusters (out of 20) are shown here.

Each vector is (R,G,B,x,y).

K-Means Clustering: Axis Scaling

- Features of different types may have different scales.
  - For example, pixel coordinates on a 100x100 image vs. RGB color values in the range [0,1].
- Problem: Features with larger scales dominate clustering.
- Solution: Scale the features.