KECE471 Computer Vision

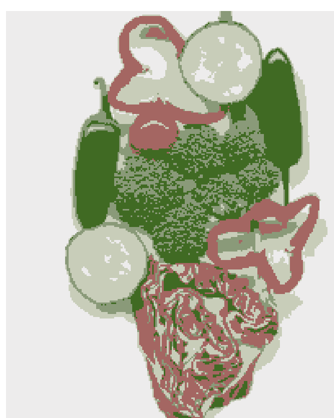
# Segmentation by Clustering

*Chang-Su Kim*

Chapter 14, Computer Vision by Forsyth and Ponce
Note: Dr. Forsyth's notes are partly used.
Jae-Kyun Ahn in Korea University made the first draft of these slides

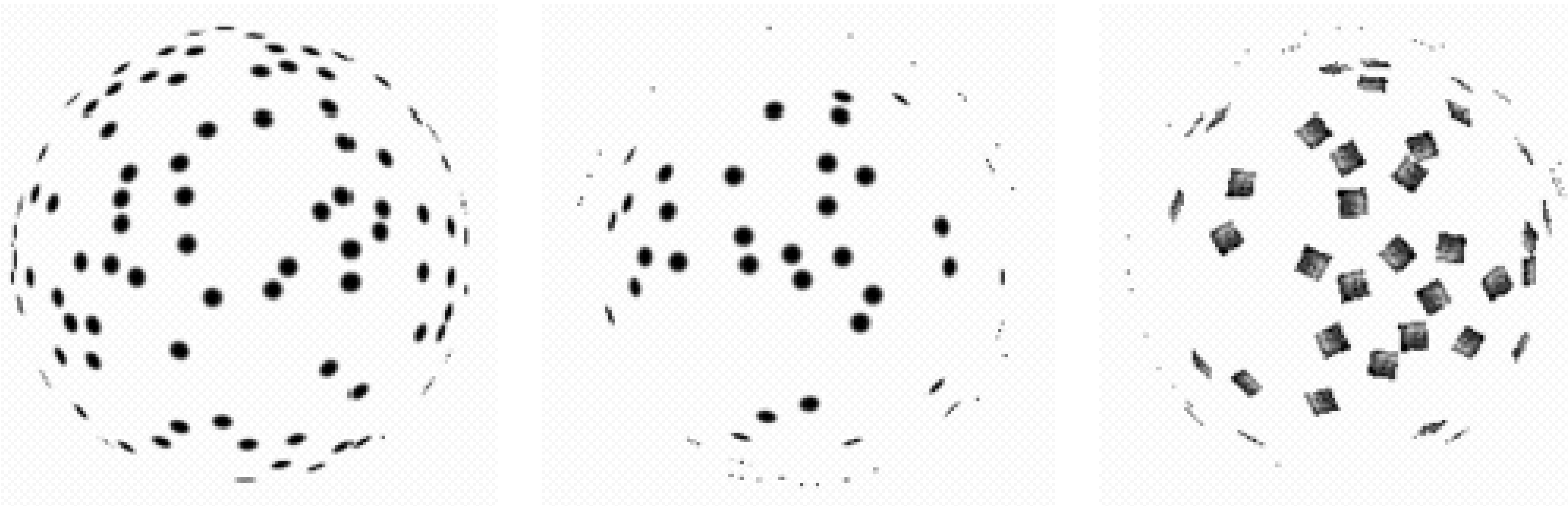
# Segmentation and Grouping

- Segmentation
  - Obtain **compact representation** from an image/motion sequence/set of tokens
  - Grouping (or clustering)
    - collect together tokens that belong together
  - Fitting
    - associate a model with tokens

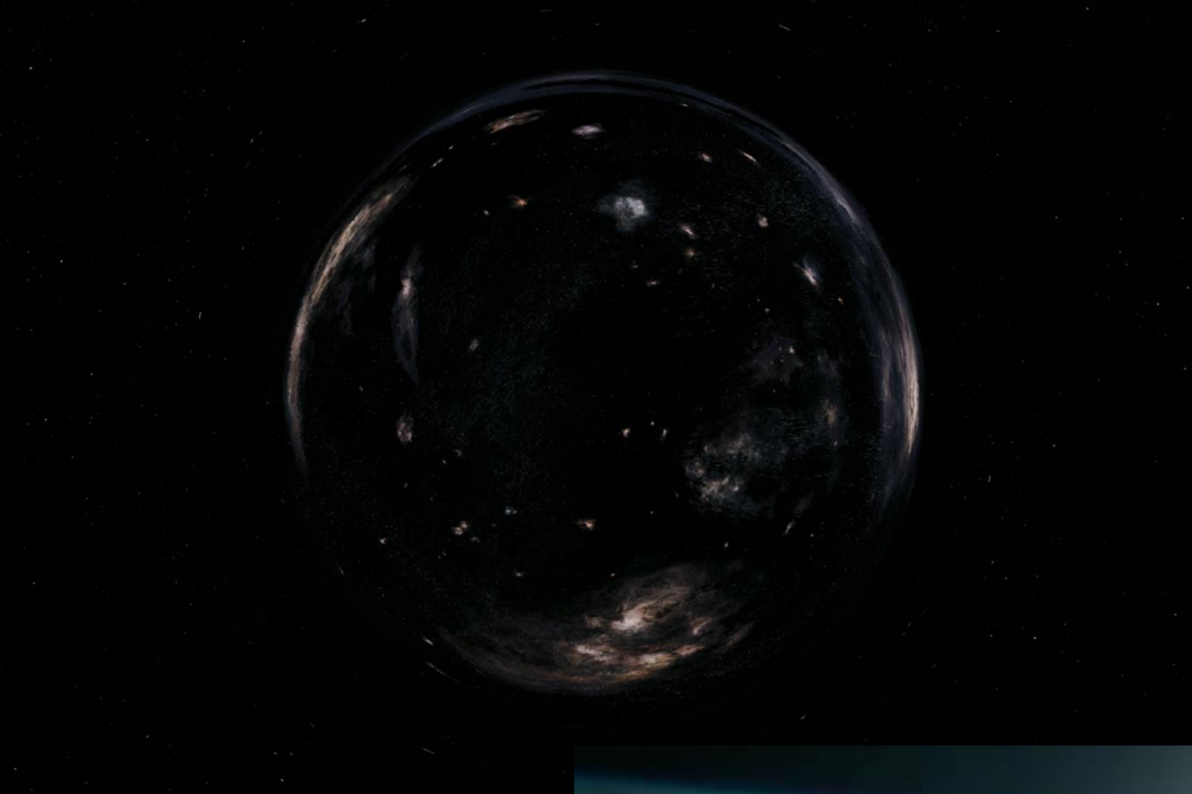



# General ideas

- Tokens
  - whatever we need to group (pixels, points, surface elements, etc.)
- Top down segmentation
  - tokens belong together because they lie on the same object
- Bottom up segmentation
  - tokens belong together because they are locally coherent
- These two are not mutually exclusive



Why do these tokens seem to belong together?



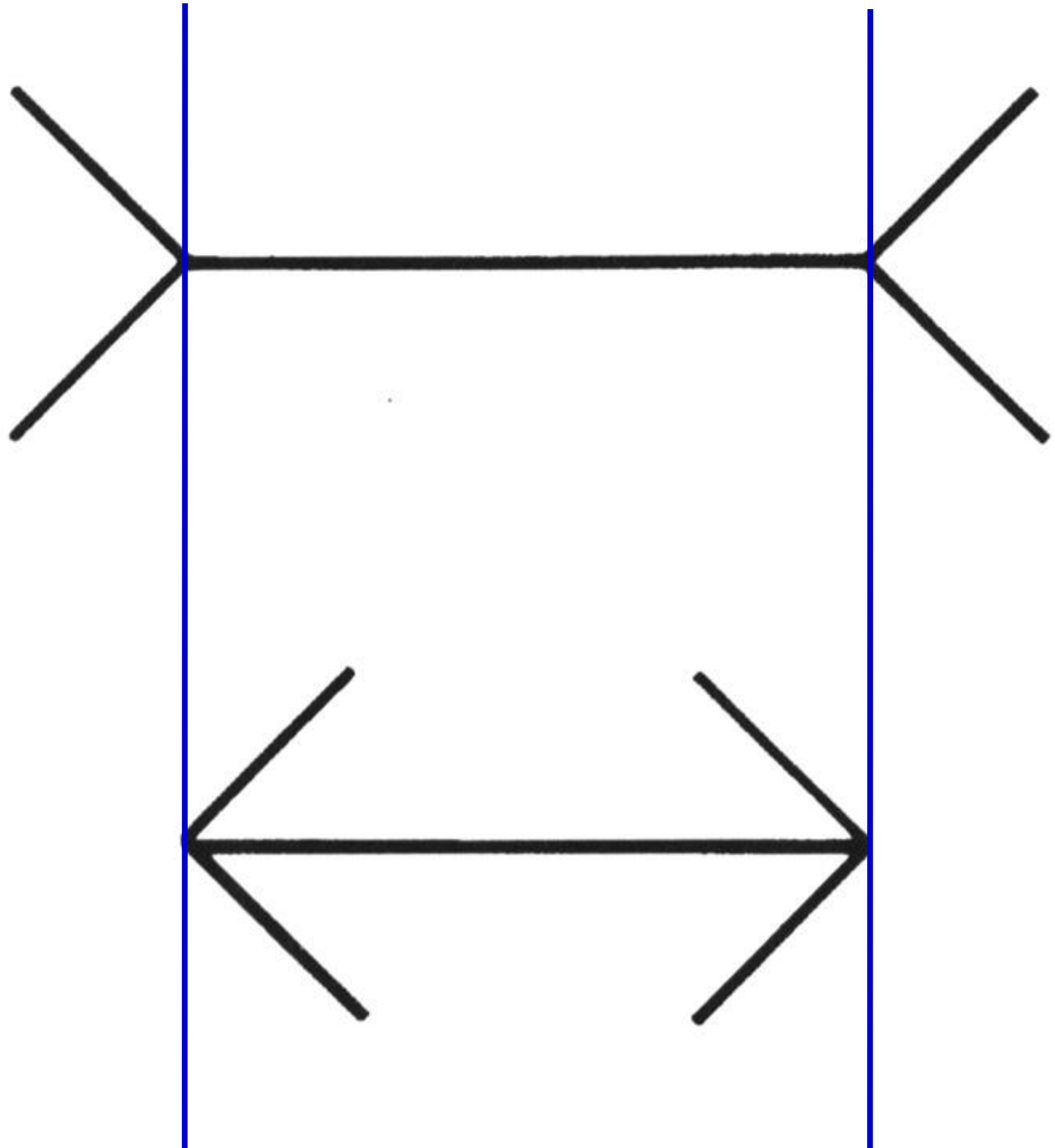

INTERSTELLAR

# Basic ideas of grouping by humans

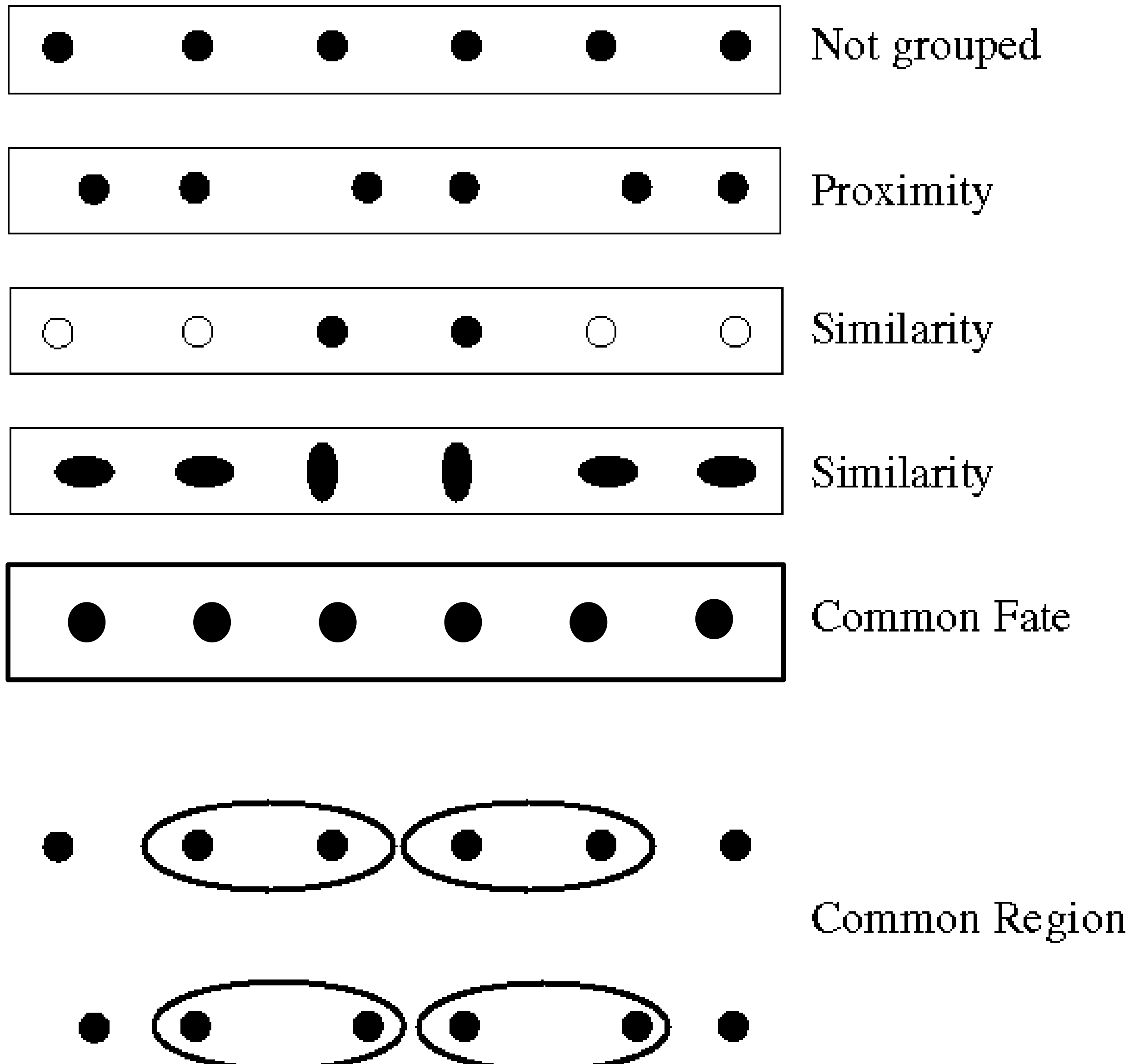
- Gestalt (group or whole)
  - Gestalt school (學派) of psychologist
  - The tendency of the visual system to assemble components of a picture together and perceive them together

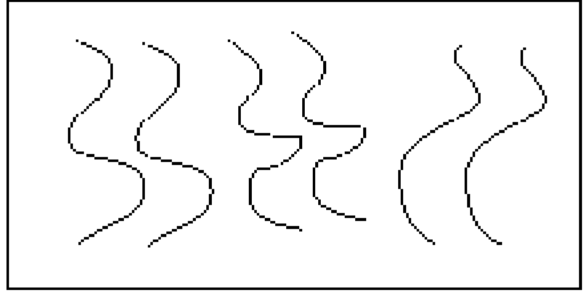
- Gestaltqualität (gestalt properties)
  - A set of factors that affect which elements should be grouped together

# Perceiving objects as groups

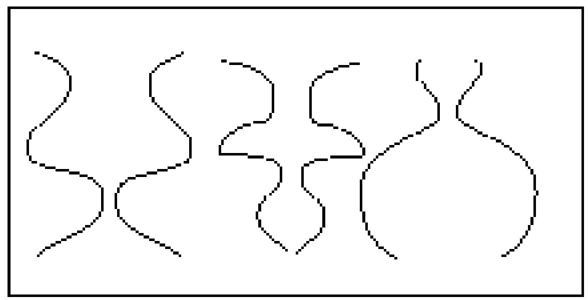


# Gestaltqualität

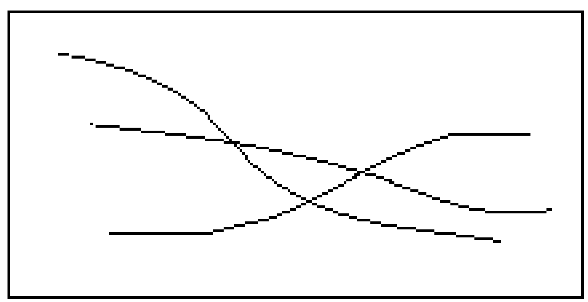




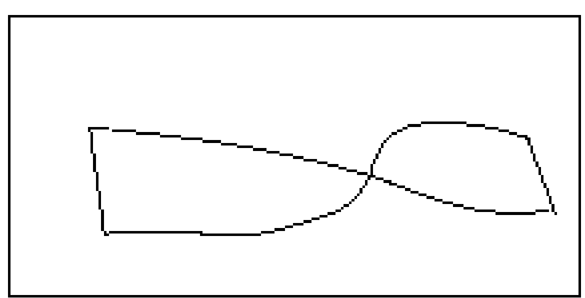
Parallelism



Symmetry



Continuity



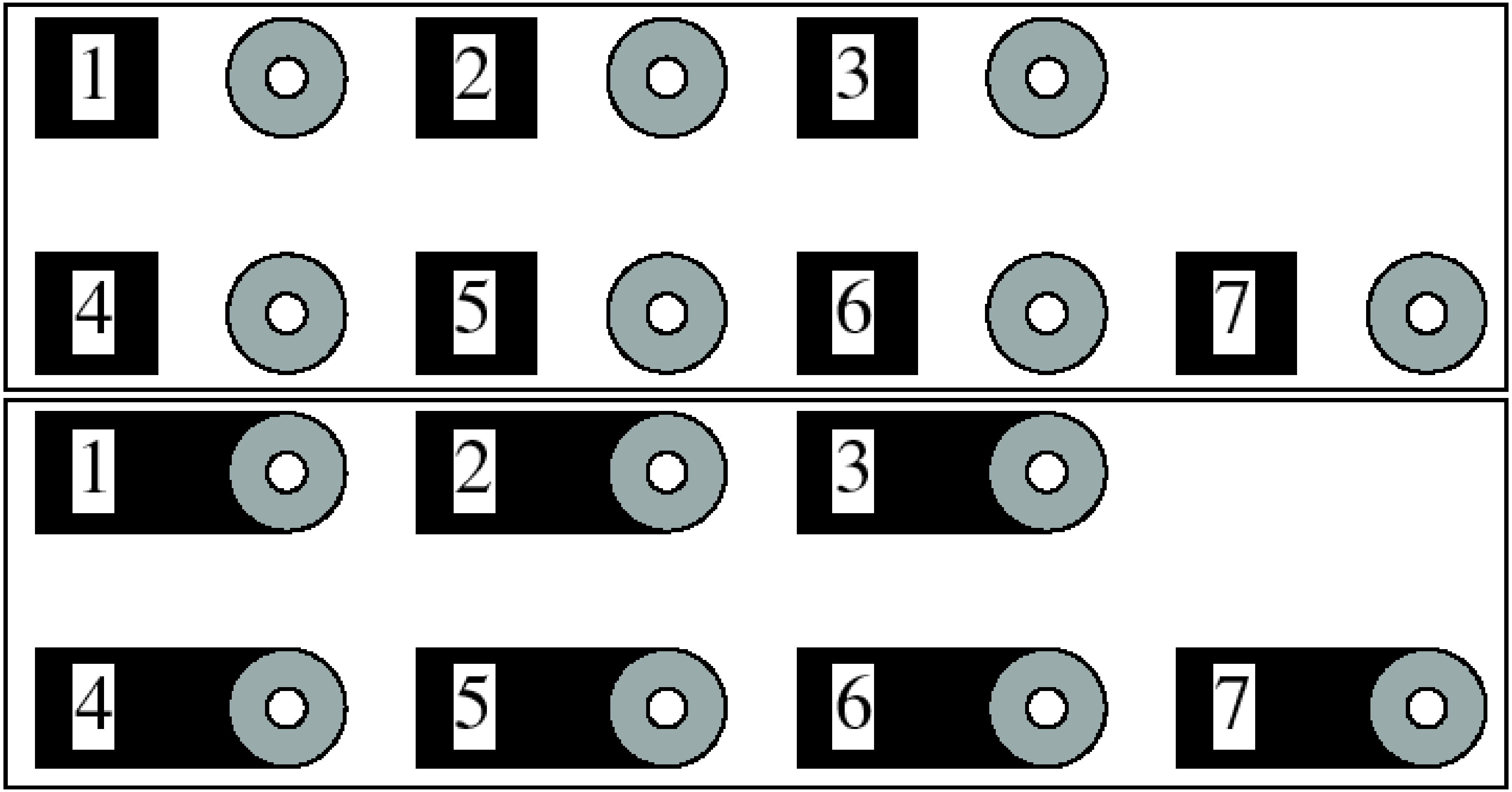
Closure

# Grouping in case of occlusion


© REISEGELEIDER.COM

# Illusory Contours

# Common Region Cues



# Example 1: Shot Boundary Detection

- Find the shots in a sequence of video
  - shot boundaries usually result in big differences between successive frames
- Strategy:
  - compute interframe distances
  - declare boundaries where these are big
- Possible distances
  - frame differences
  - histogram differences
  - edge differences
- Applications:
  - representation for movies, or video sequences
    - find shot boundaries
    - obtain "most representative" frame
  - supports search

# cf. Video Tapestry

## Video Tapestries with Continuous Temporal Zoom

Connelly Barnes[1] Dan B Goldman[2] Eli Shechtman[2,3] Adam Finkelstein[1]
[1]Princeton University [2]Adobe Systems [3]University of Washington

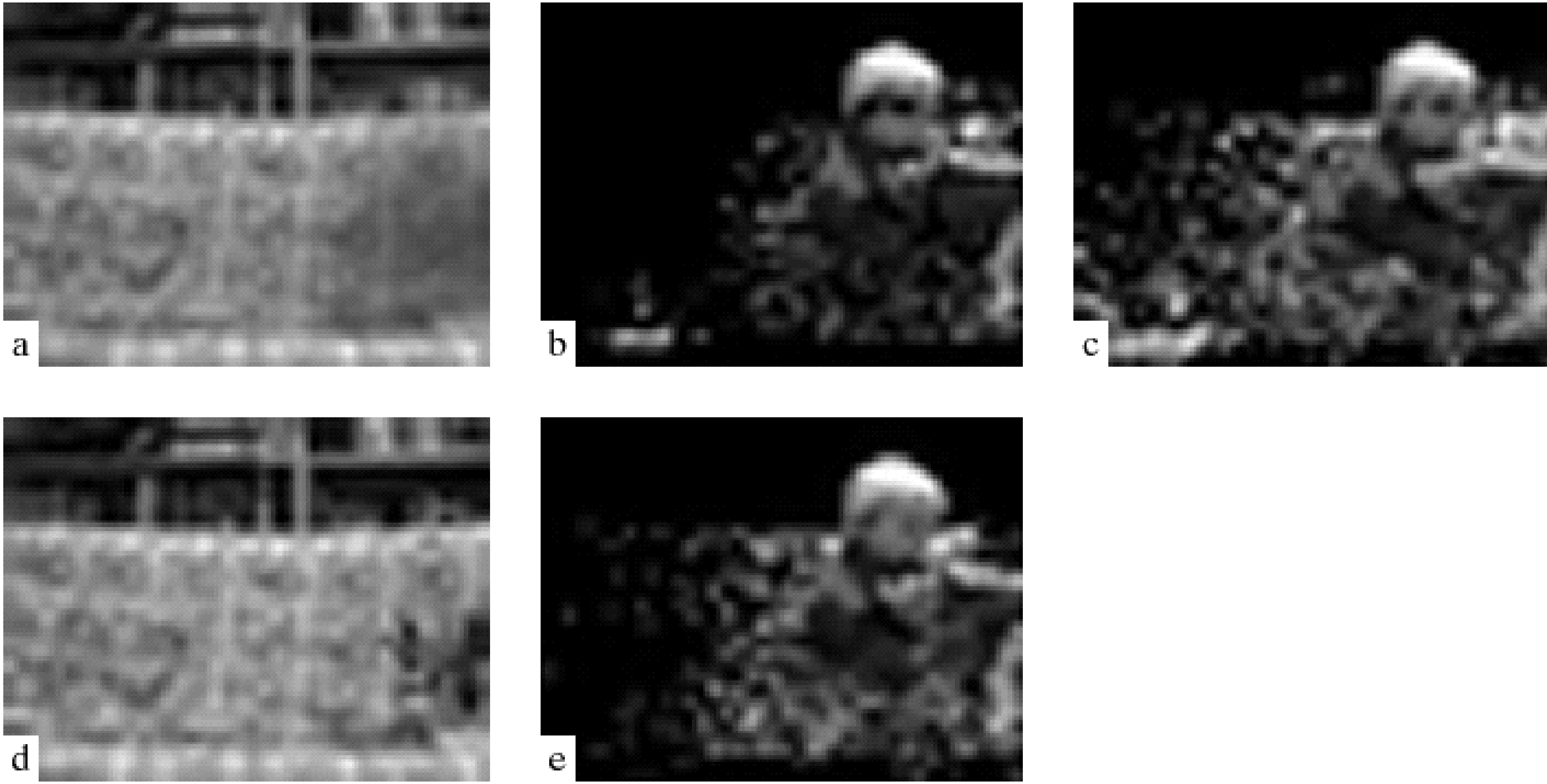
**Figure 1:** *A multiscale tapestry represents an input video as a seamless and zoomable summary image which can be used to navigate through the video. This visualization eliminates hard borders between frames, providing spatial continuity and also continuous zooms to finer temporal resolutions. This figure depicts three discrete scale levels for the film* Elephants Dream *(Courtesy of the Blender Foundation). The lines between each scale level indicate the corresponding domains between scales. See the video to view the continuous zoom animation between the scales. For Copyright reasons, the print and electronic versions of this paper contain different imagery in Figures 1, 4, 6, and 7.*

# Example 2: Background Subtraction

- If we know what the background looks like, it is easy to identify interesting pixels

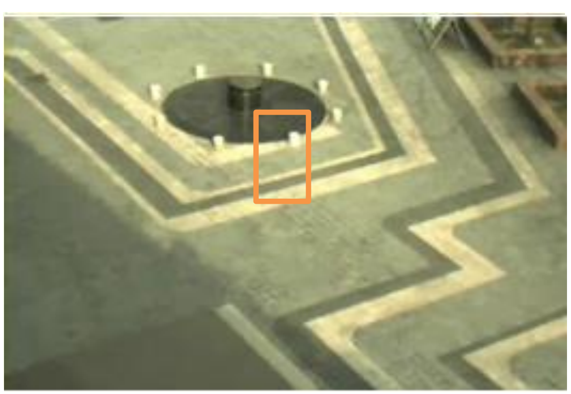
- Applications
  - Person in an office
  - Tracking cars on a road
  - surveillance

- Approach:
  - use a moving average to estimate background image
  - subtract from current frame
  - large absolute values are interesting pixels
    - trick: use morphological operations to clean up pixels

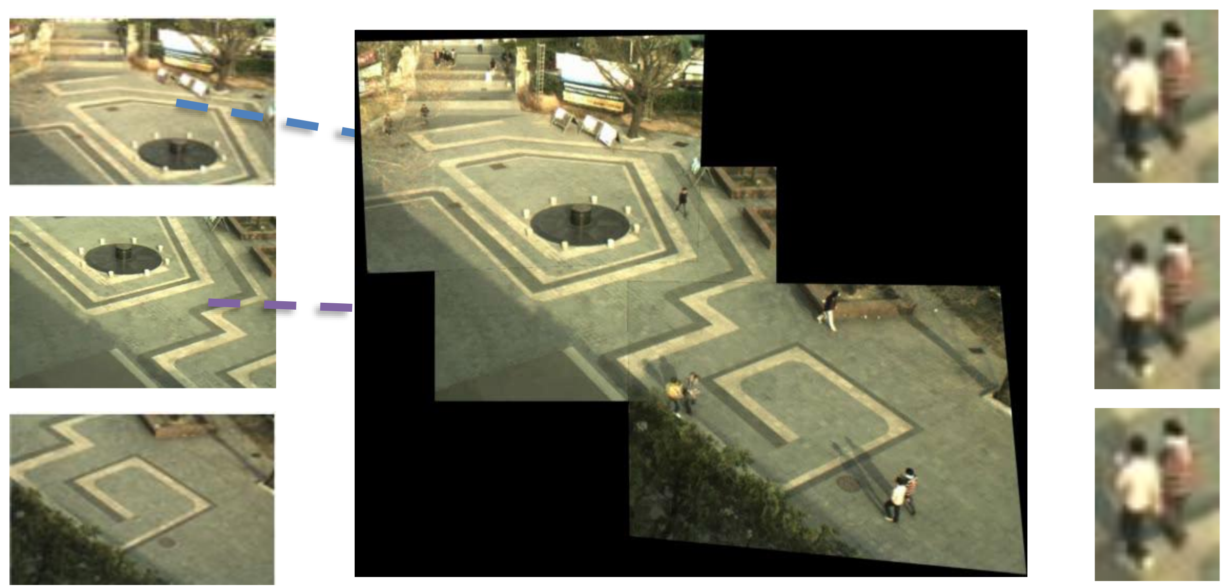

a
b
c
d
e

# 전역 시점 영상 합성

- 동영상에서의 전역 시점 영상 합성
  - 배경 영상 추정 기법
    - 확률 분포에 기반한 배경 영상 추정 및 획득
  - 전역 시점 영상 합성
    - 추정된 배경 영역들을 이용한 배경 합성 → 합성된 배경위에 객체를 합성



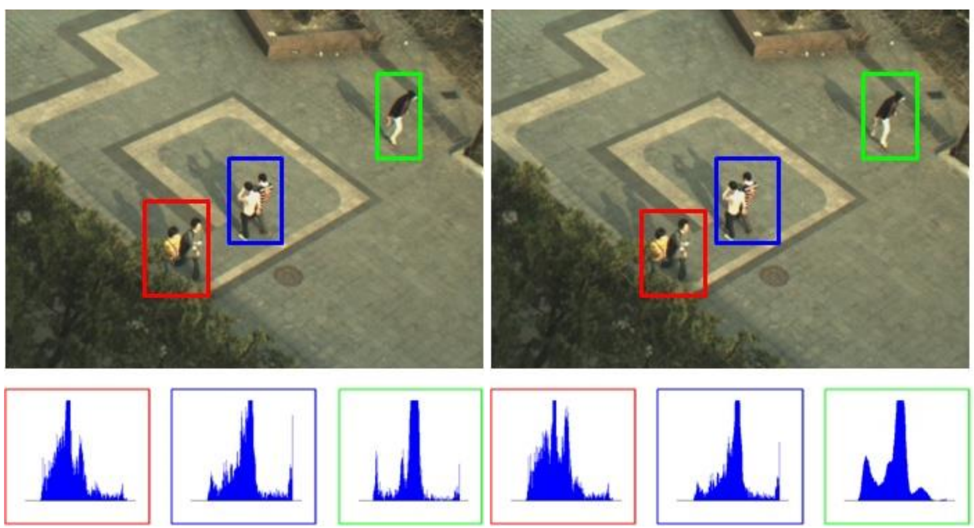
**배경 영상 추정**



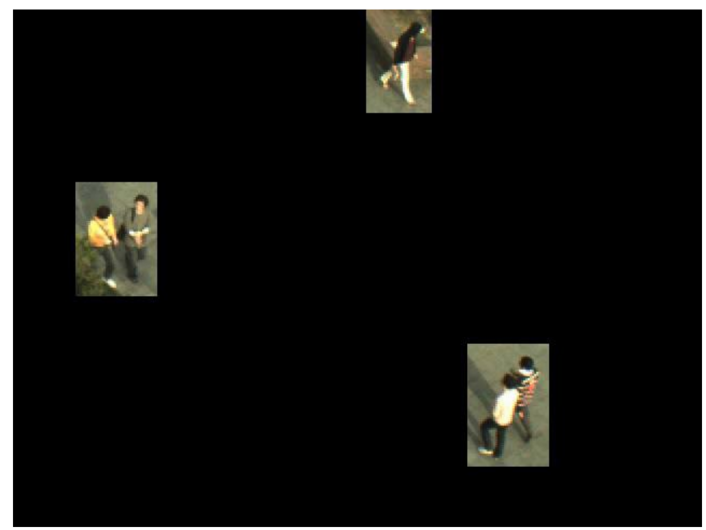
**전역 시점 영상 합성: 배경 + 객체**

# 개별 카메라의 이벤트 처리

- 객체 추적
  - 연속된 두 프레임에서 추출된 객체 추적 기법
    - 마스크 내의 히스토그램 유사성을 통한 객체 추적



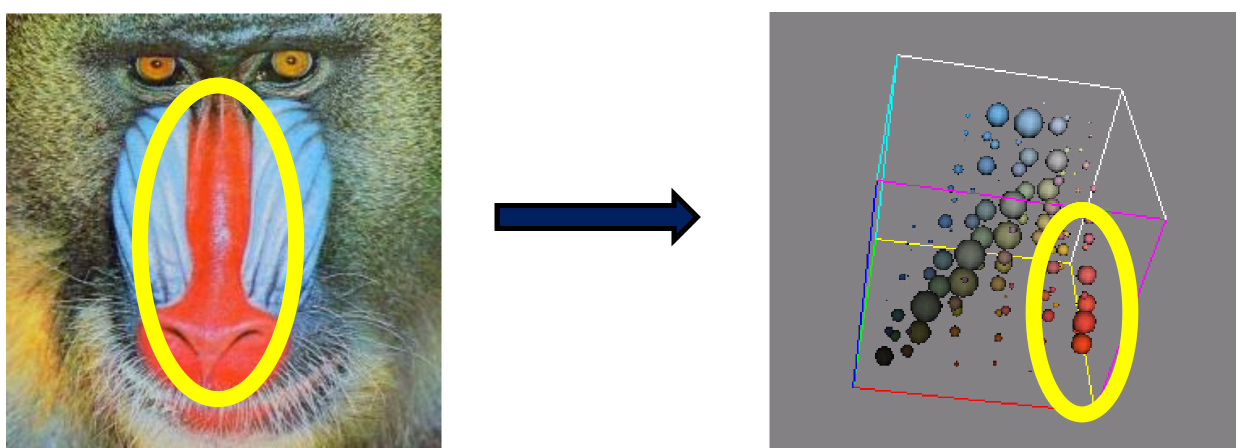
연속된 프레임에서 마스크내의 히스토그램 유사성



객체 추적 결과 영상

# Segmentation as Clustering

- Clustering
  - Cluster together pixels, tokens, and etc that belong together
  - Agglomerative clustering
    - combine two close clusters to make one
  - Divisive clustering
    - split a cluster along best boundary



# Agglomerative Clustering

- Each item is regarded as a cluster, and clusters are recursively merged to yield a good clustering
- Clustering by merging
- A bottom-up approach

```
Agglomerative Clustering


Make each point a separate cluster
Until the clustering is satisfactory
    Merge the two clusters with the smallest
    inter-cluster distance
End
```

# Divisive Clustering

- The entire set is regarded as a cluster, and then clusters are recursively split to yield a good clustering
- Clustering by splitting
- A top-down approach

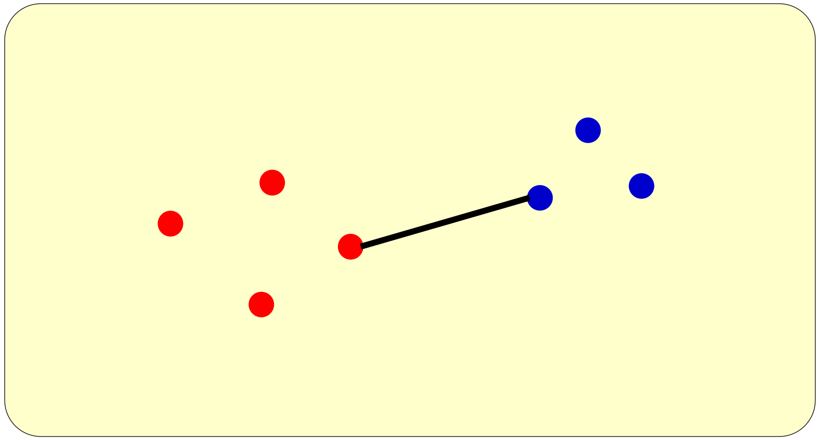
```
Divisive Clustering


Construct a single cluster containing all points
Until the clustering is satisfactory
     Split the cluster that yiels the two components
     with the largest inter-cluster distance
End
```

# Inter-Cluster Distance

- Single-link clustering

$$d(A,B) = \min_{a \in A, b \in B} d(a,b)$$

  - It may yield elongated clusters

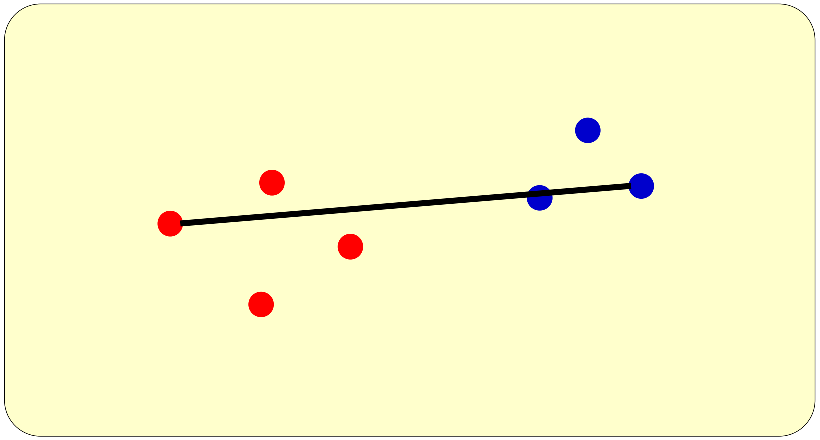


# Inter-Cluster Distance

- Complete-link clustering

$$d(A,B) = \max_{a \in A, b \in B} d(a,b)$$

  – It usually yields round clusters

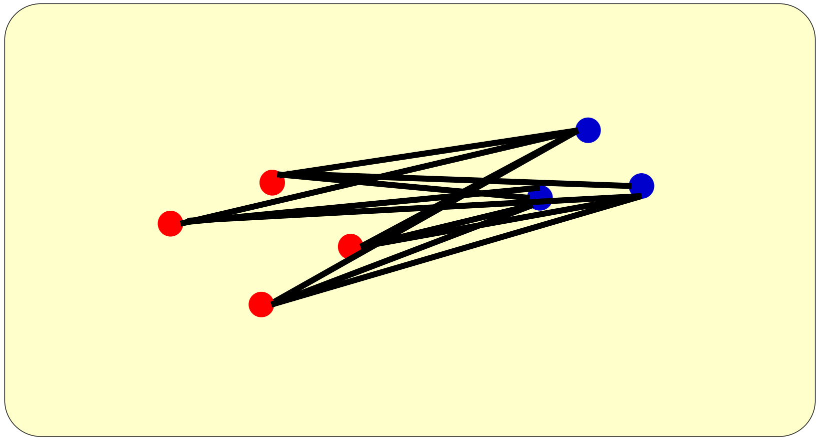


# Inter-Cluster Distance

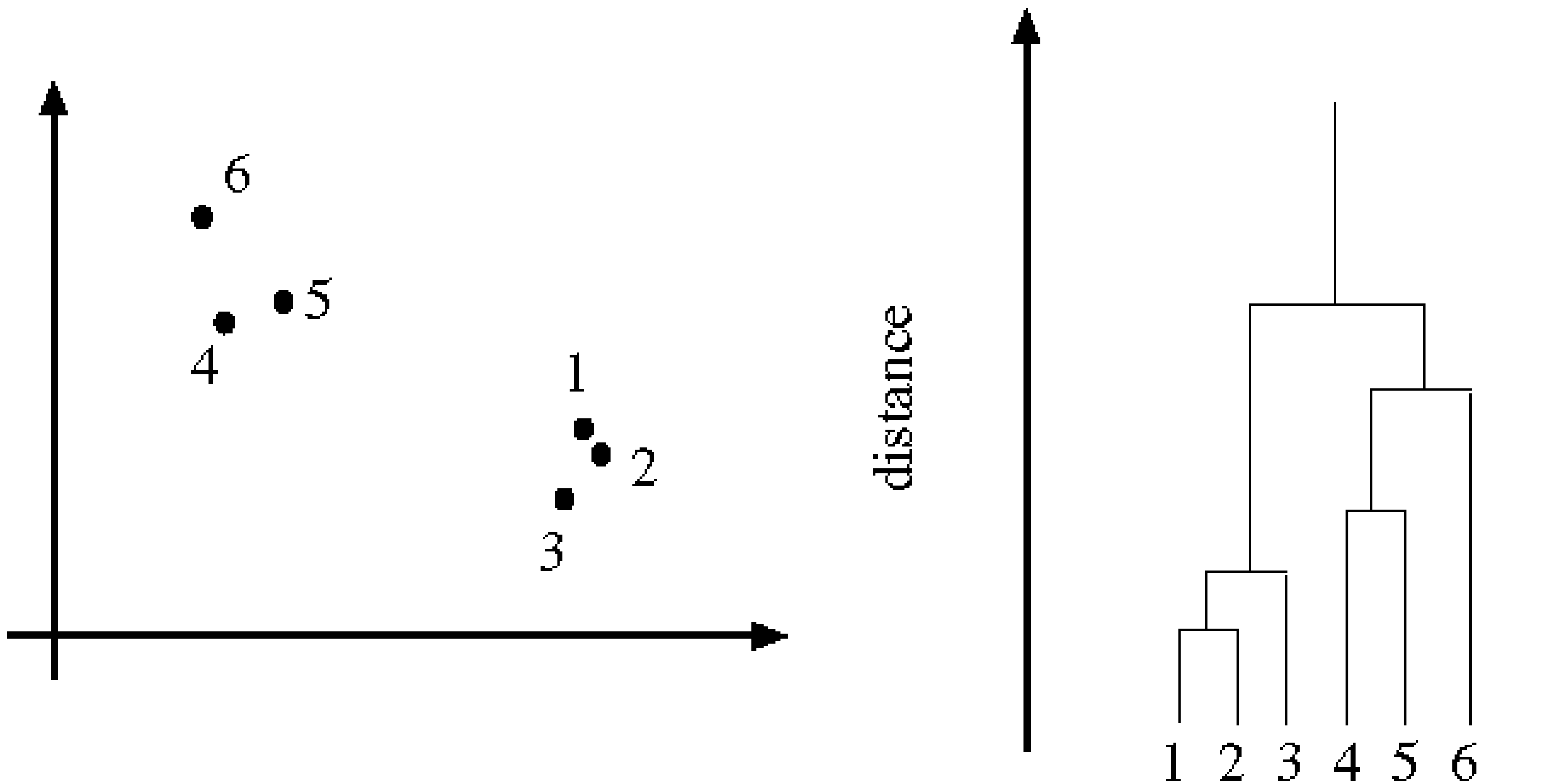
- Group-average clustering

$$d(A,B) = \frac{1}{|A| \times |B|} \sum_{a \in A, b \in B} d(a,b)$$

  - It usually yields round clusters

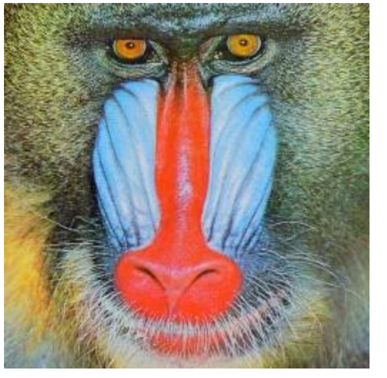


# Dendrogram for Agglomerative Clustering



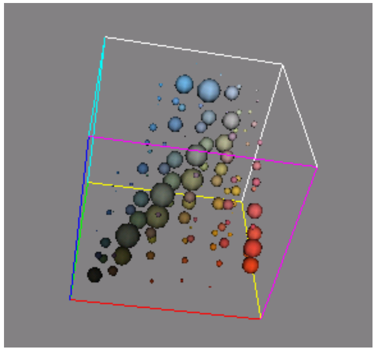
# K - Means

- Application of vector quantization
  - Choose a fixed number of clusters
  - Choose cluster centers and point-cluster allocations to minimize error
  - Repeat until centers converge



- Error or cost function

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



# K - Means

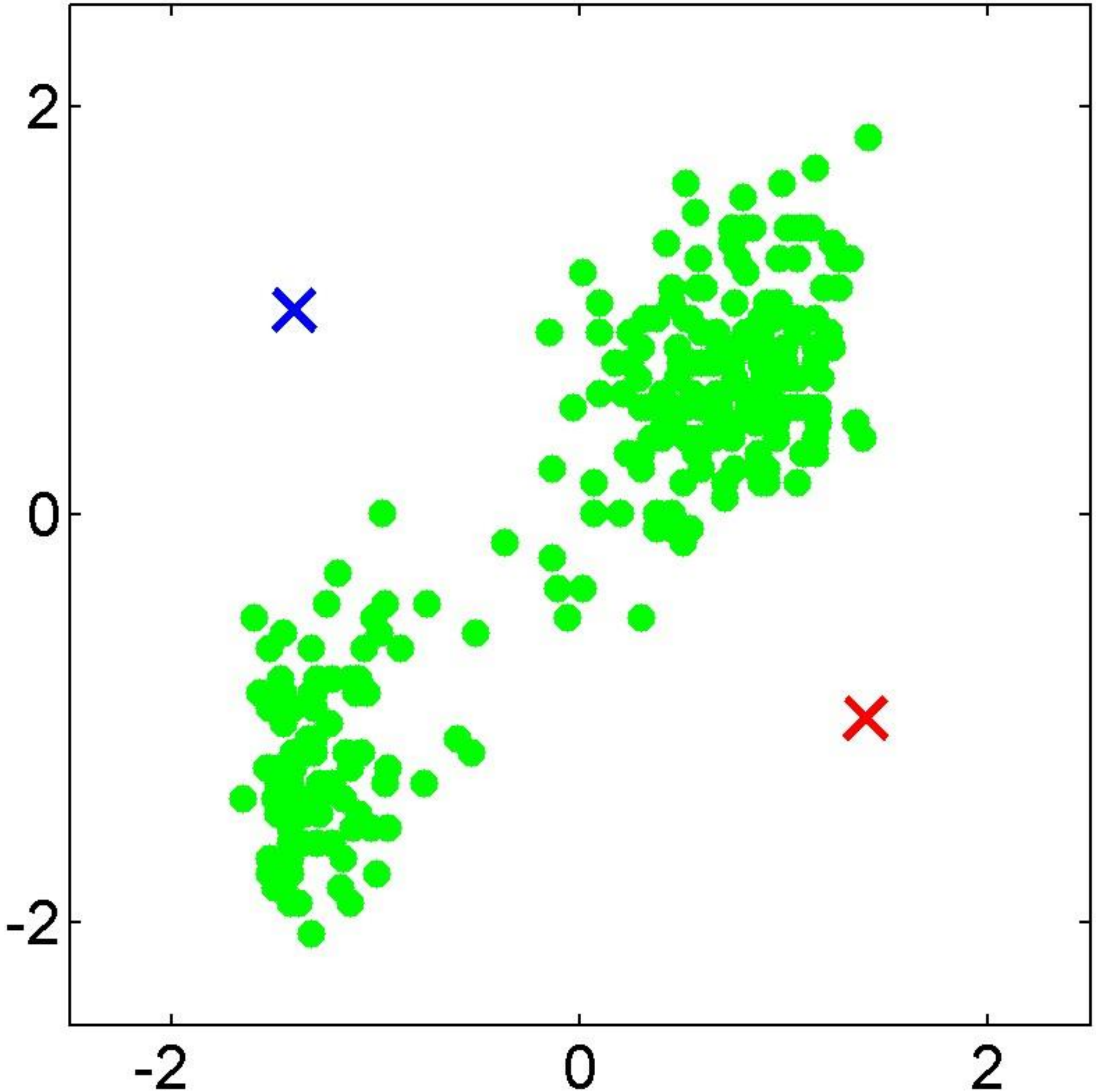
**K-Means Algorithm**

Choose k data points to act as cluster centers

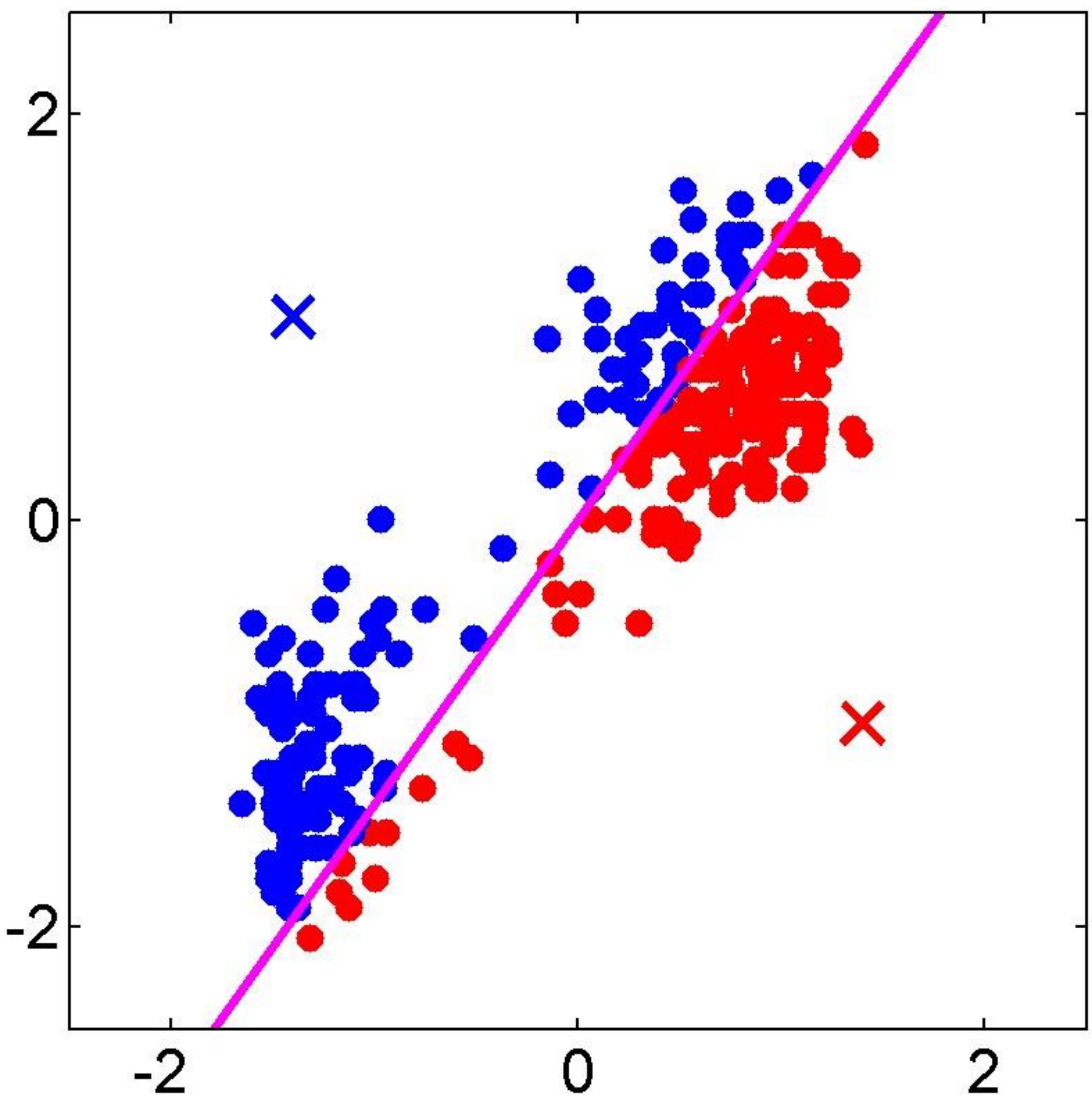
Until the cluster centers are unchanged

- Allocate each data point to cluster whose center is nearest (NN rule)
- Replace the cluster centers with the mean of the elements in their clusters (centroid rule)

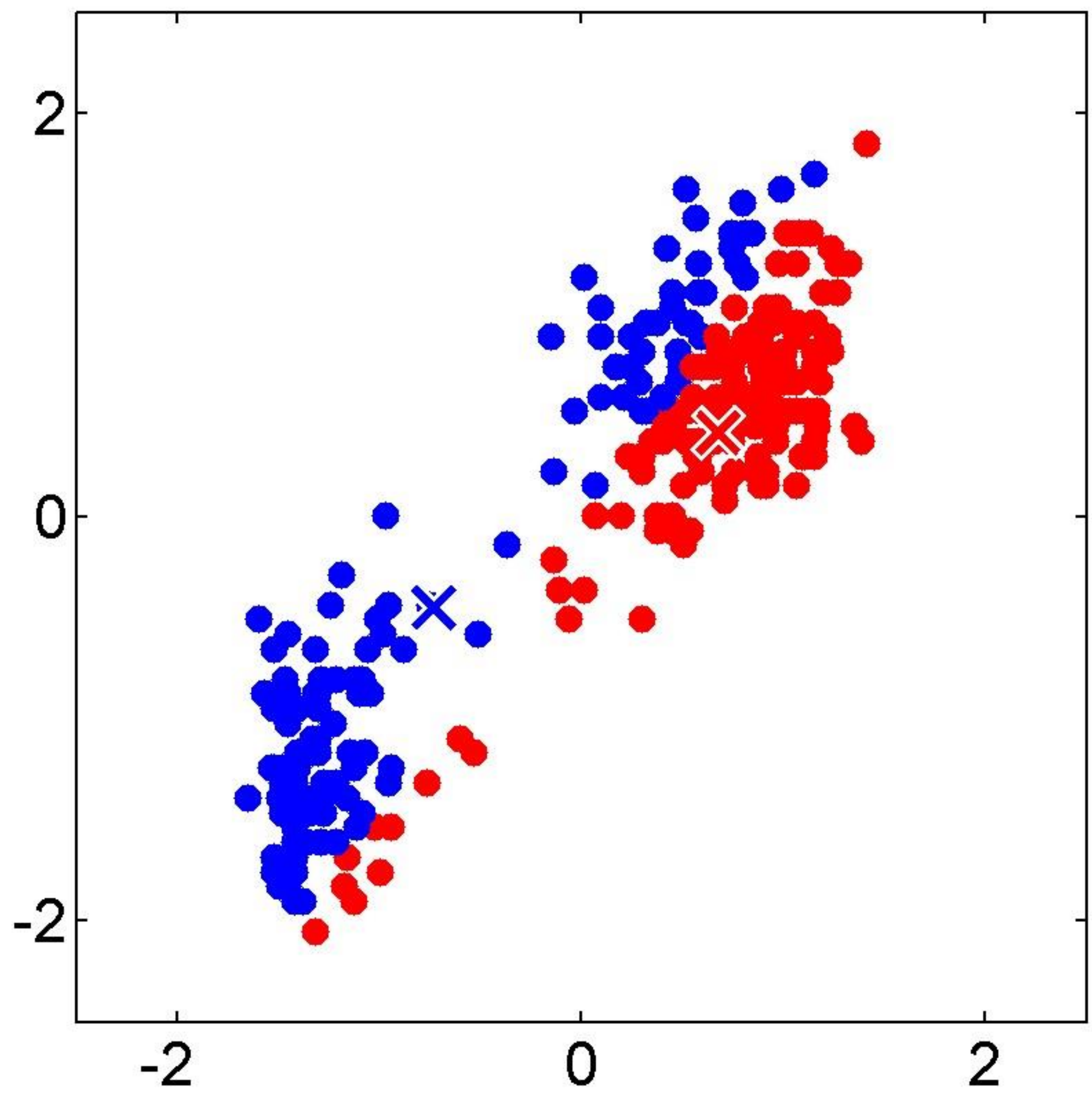
End

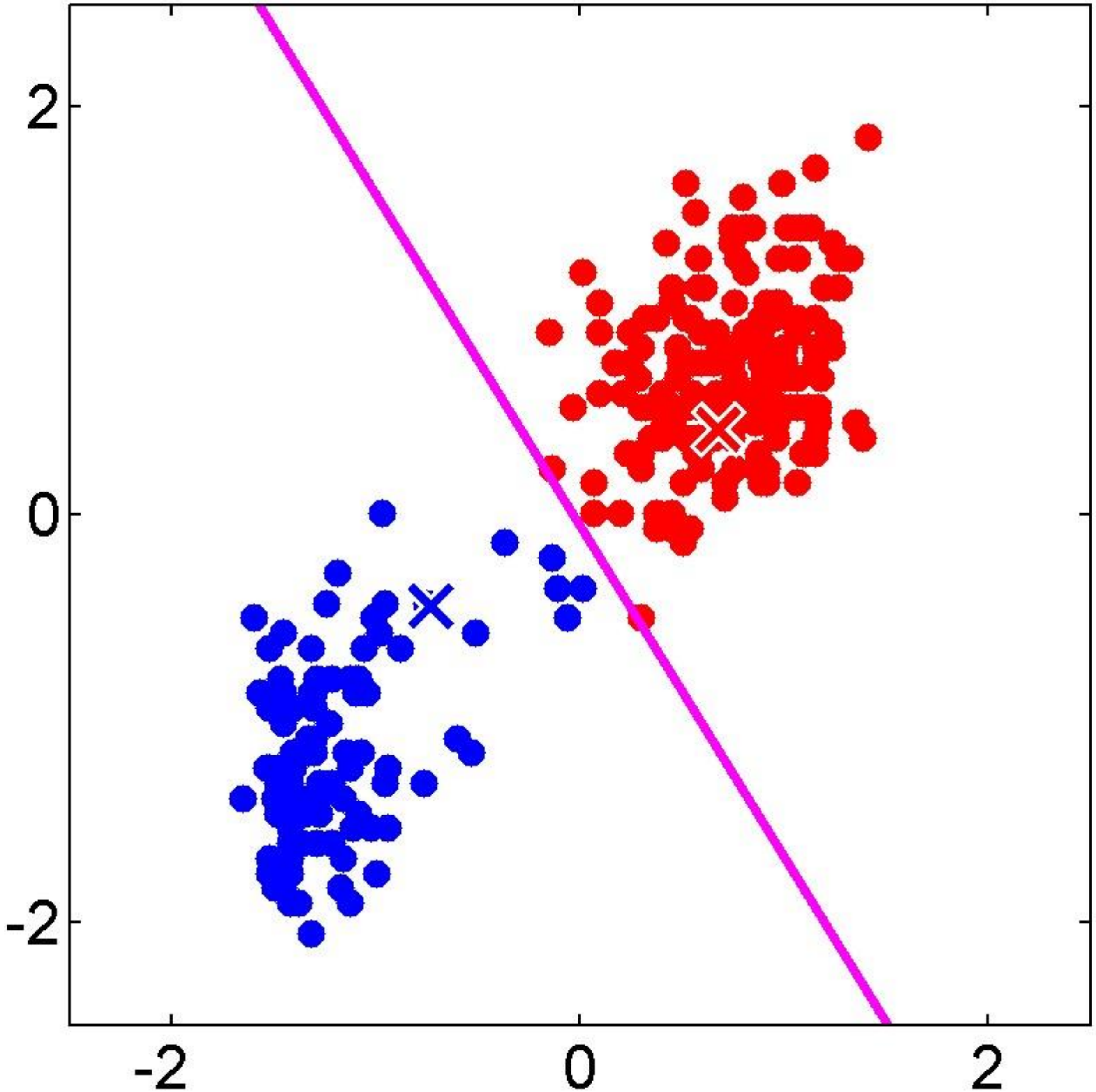


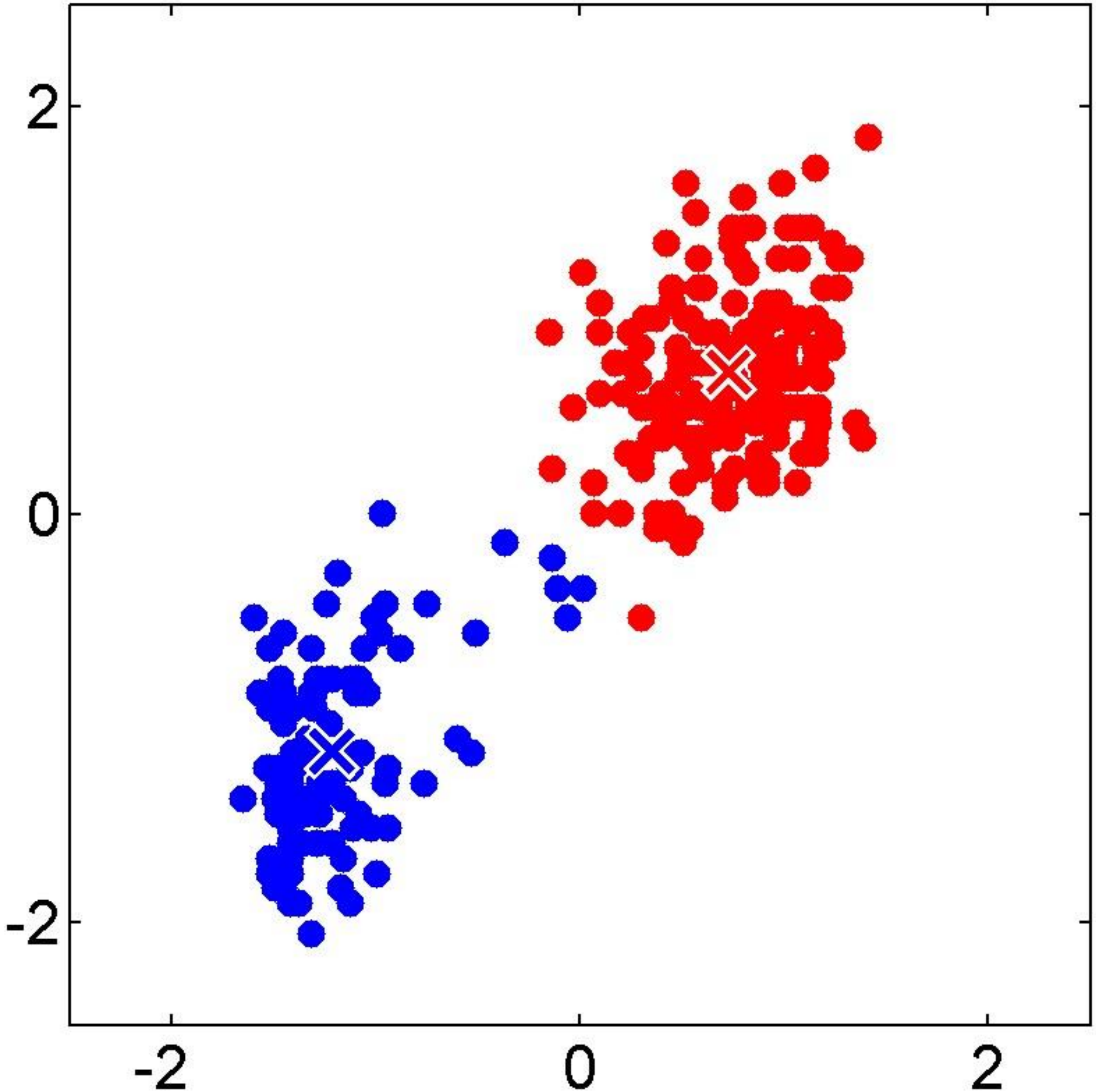
2
0
-2
-2
0
2

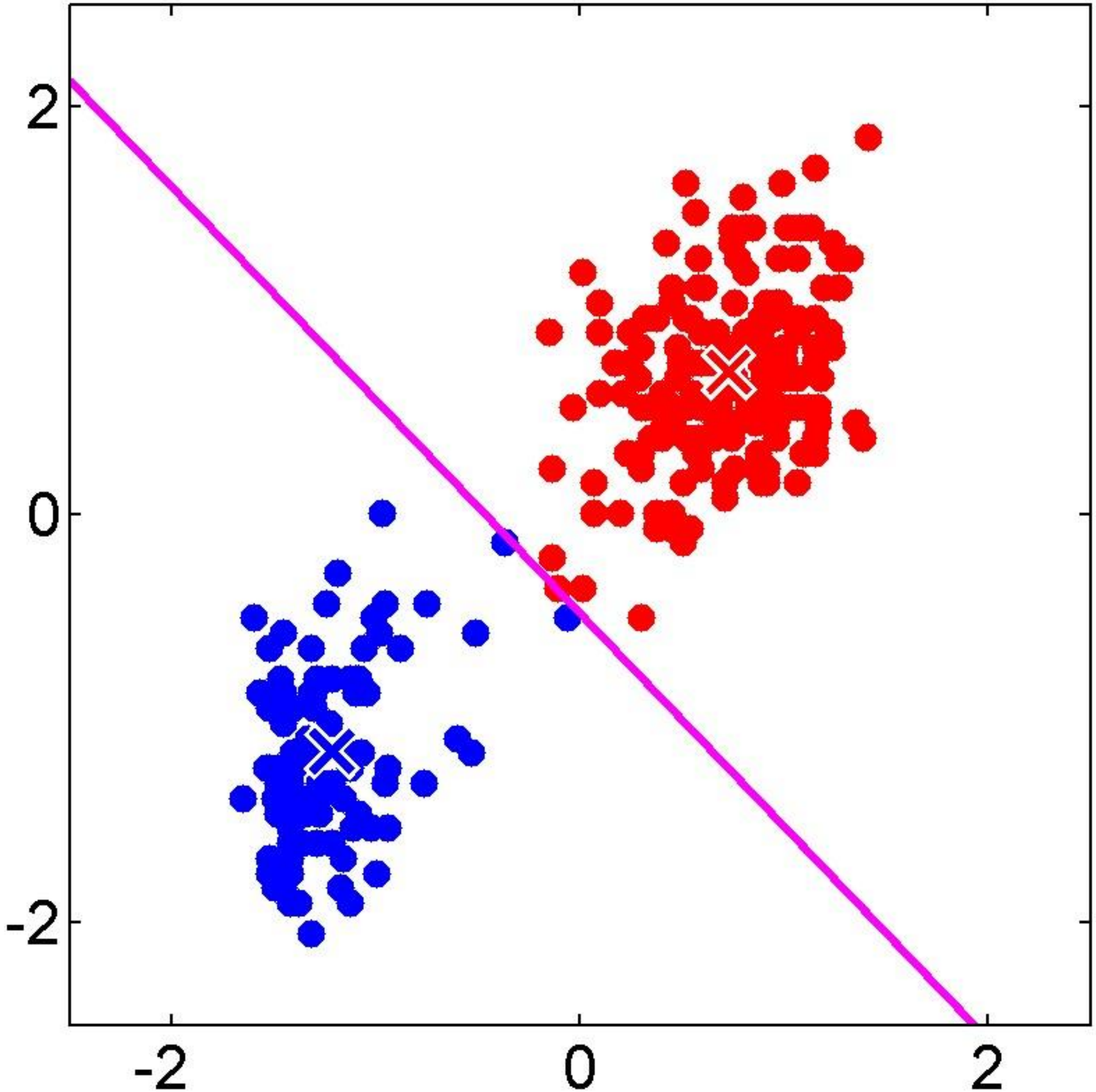


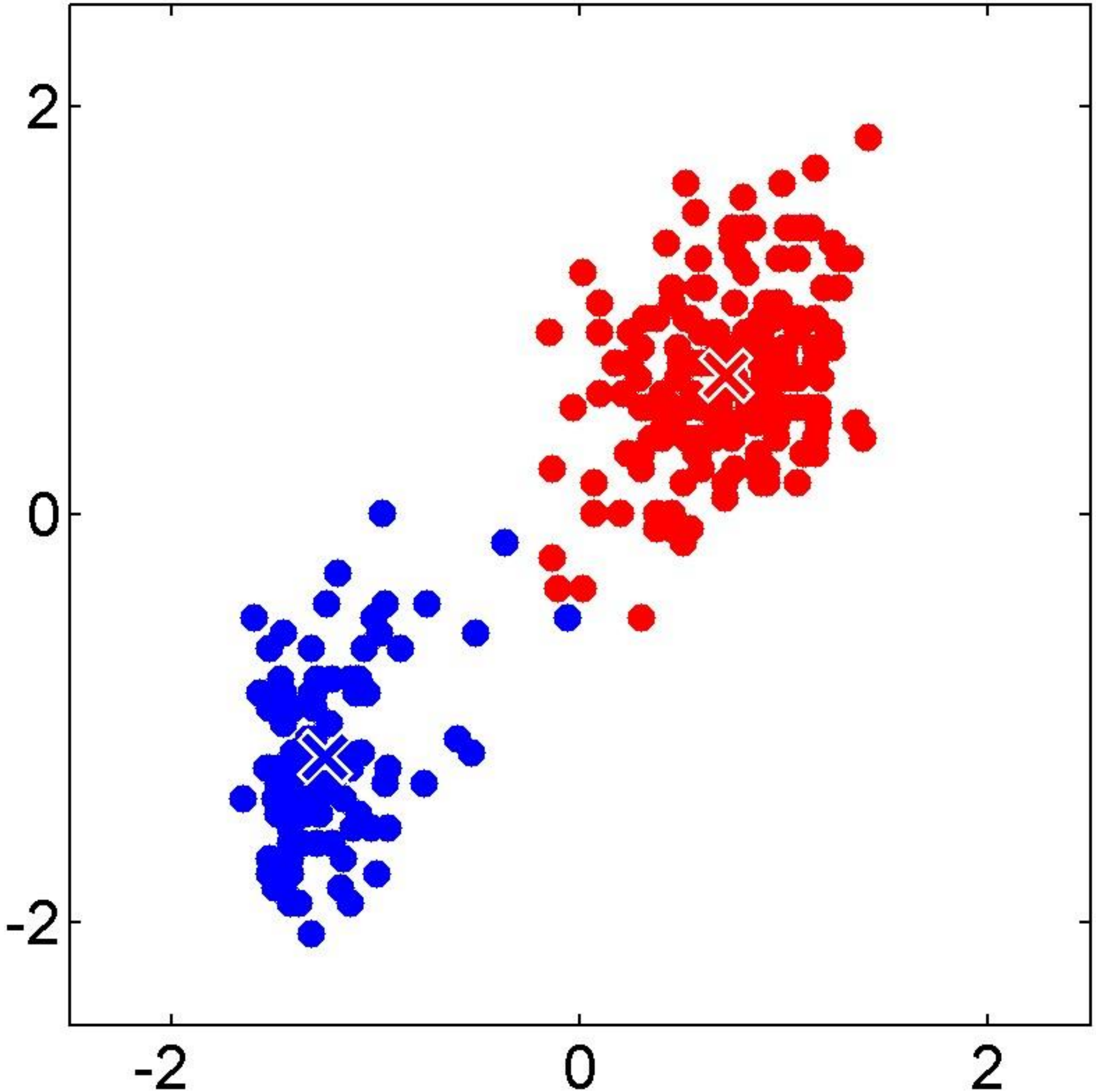
2
0
-2
-2
0
2

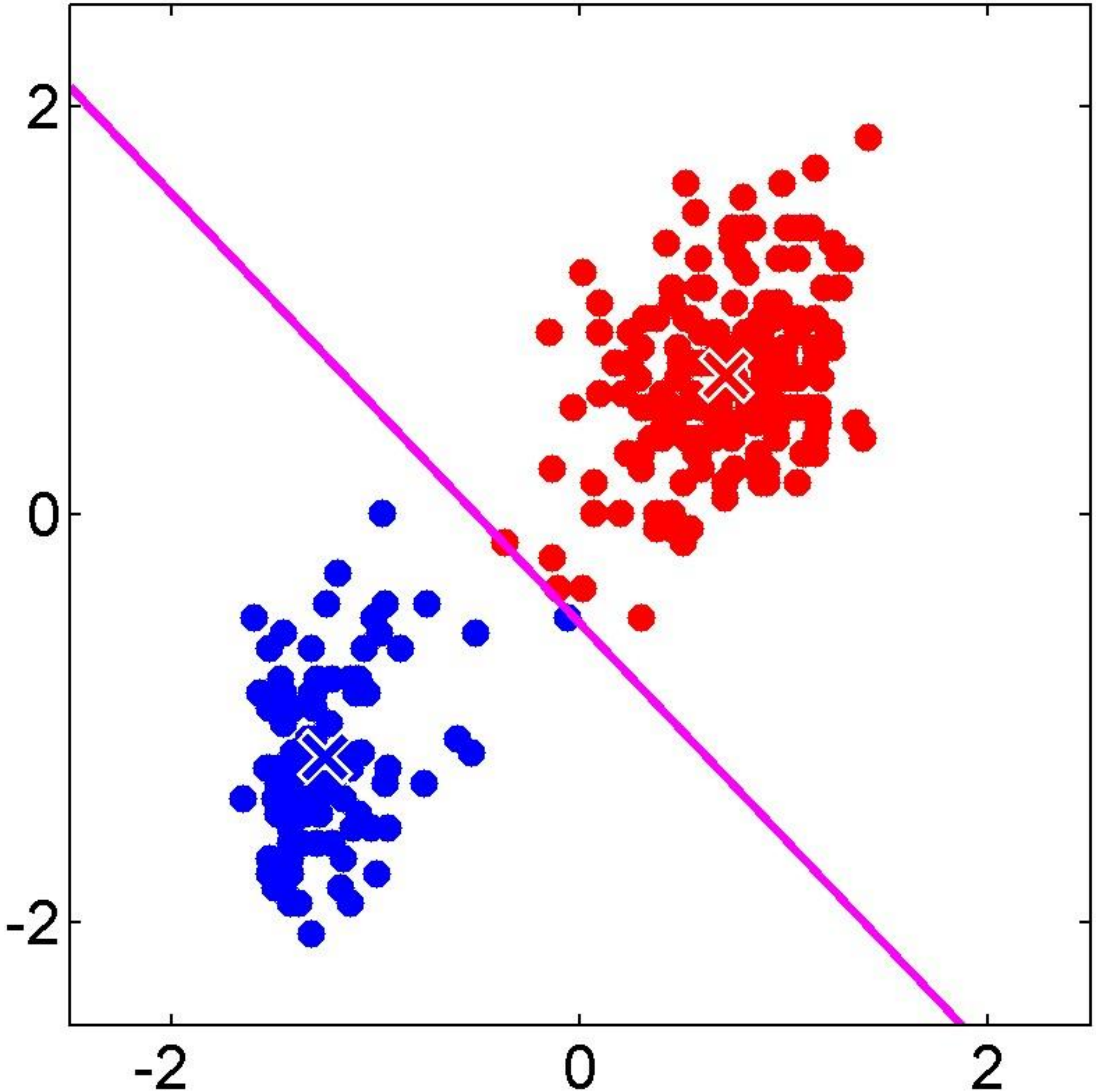

2
0
-2
-2
0
2

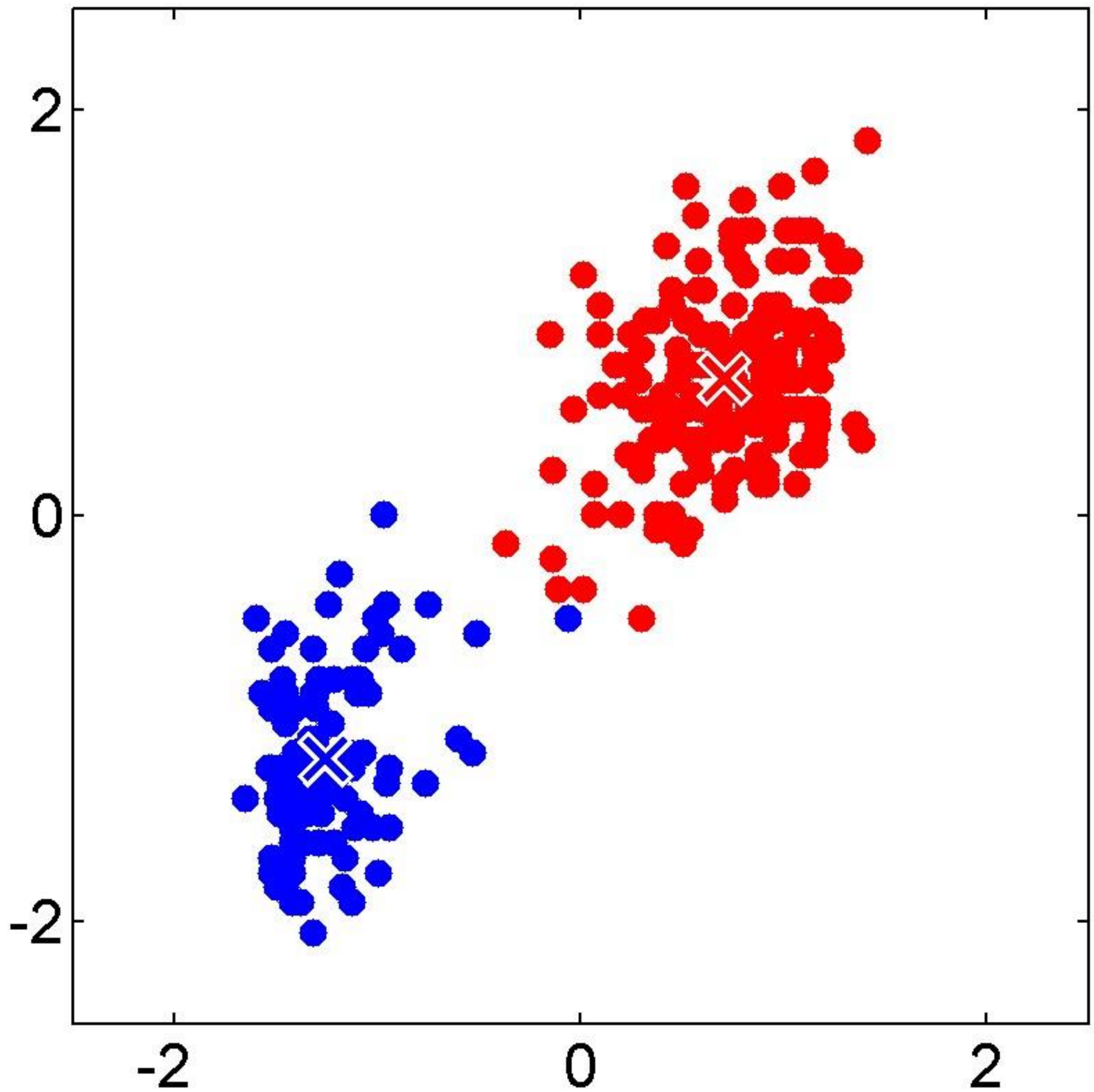

2
0
-2
-2
0
2


2
0
-2
-2
0
2


2
0
-2
-2
0
2


2
0
-2
-2
0
2

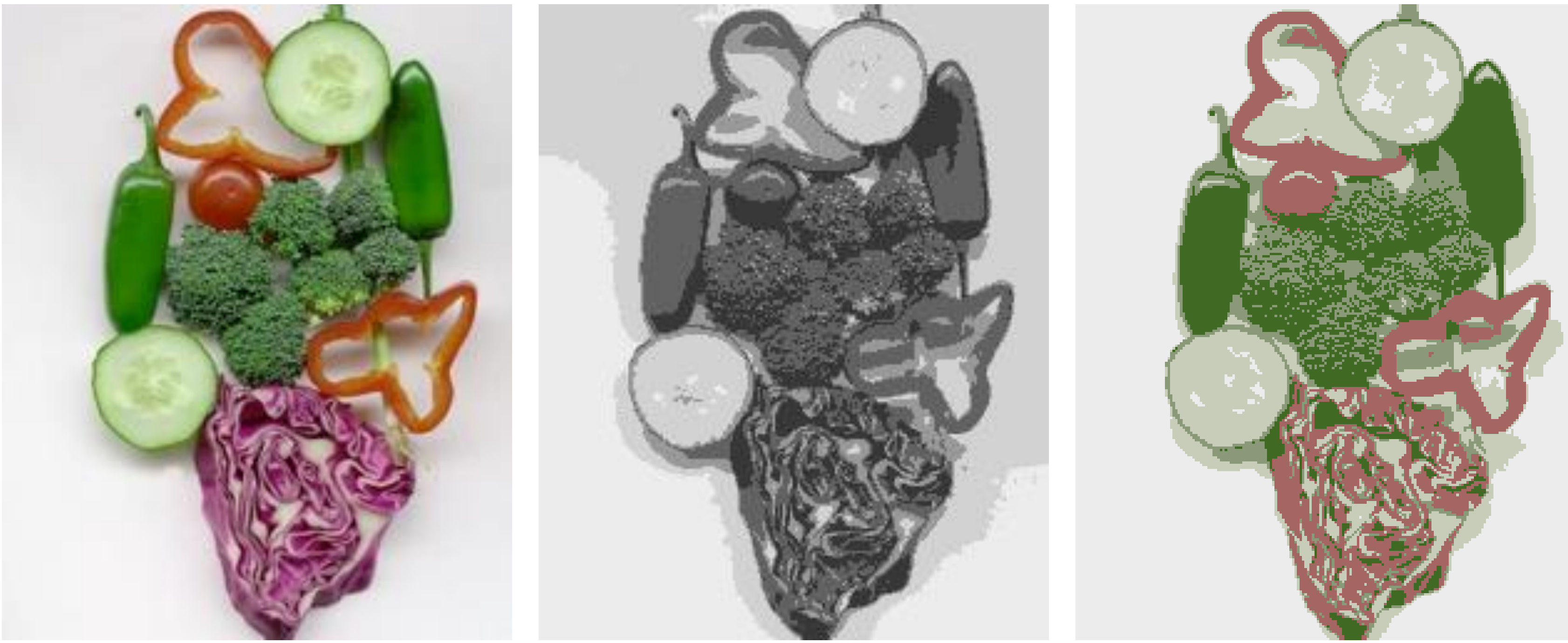

2
0
-2
-2
0
2



2
0
-2
-2
0
2

# K - Means

Image | Clusters on intensity | Clusters on color

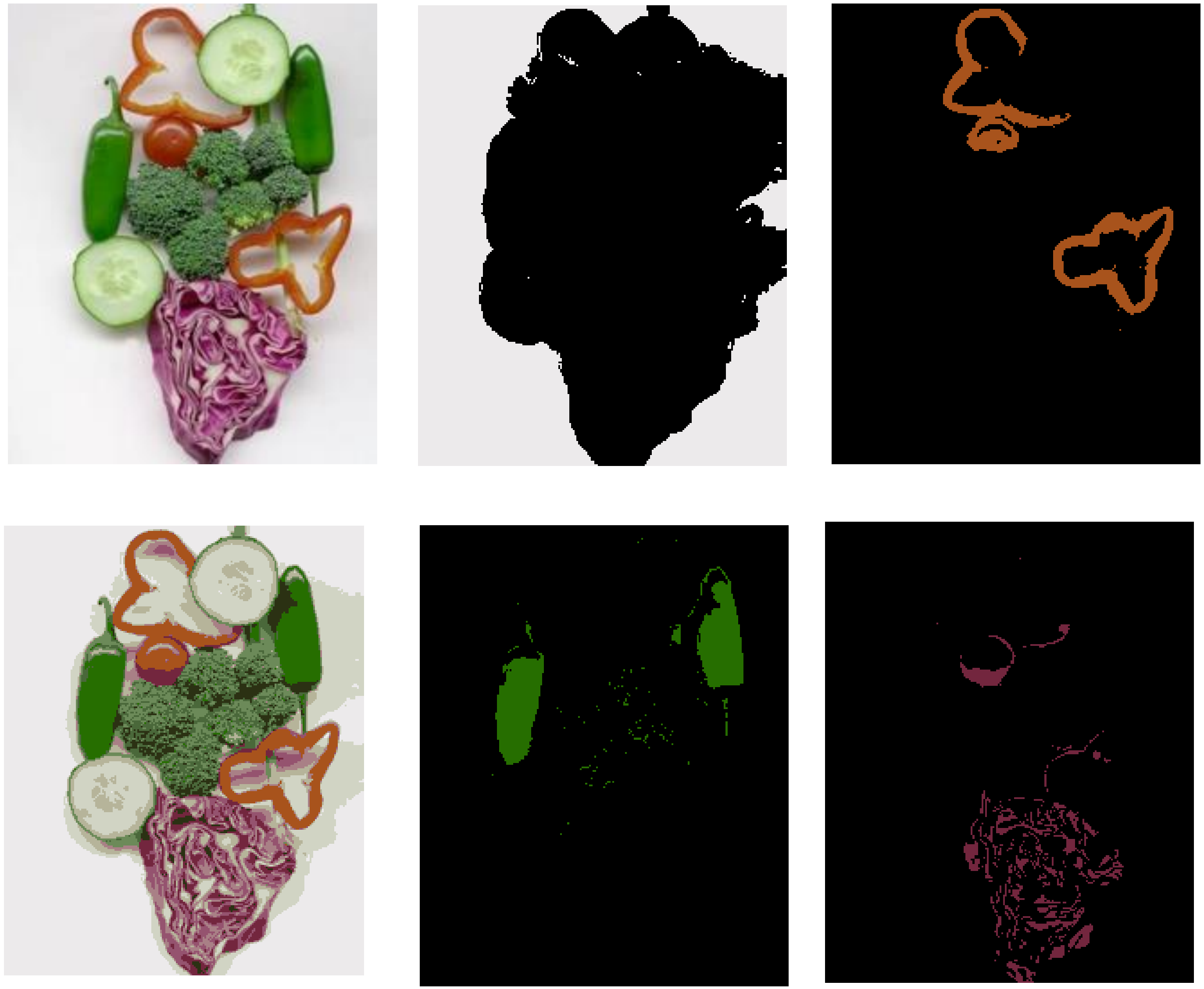


K-means clustering using intensity alone and color alone
(5 clusters in each case)

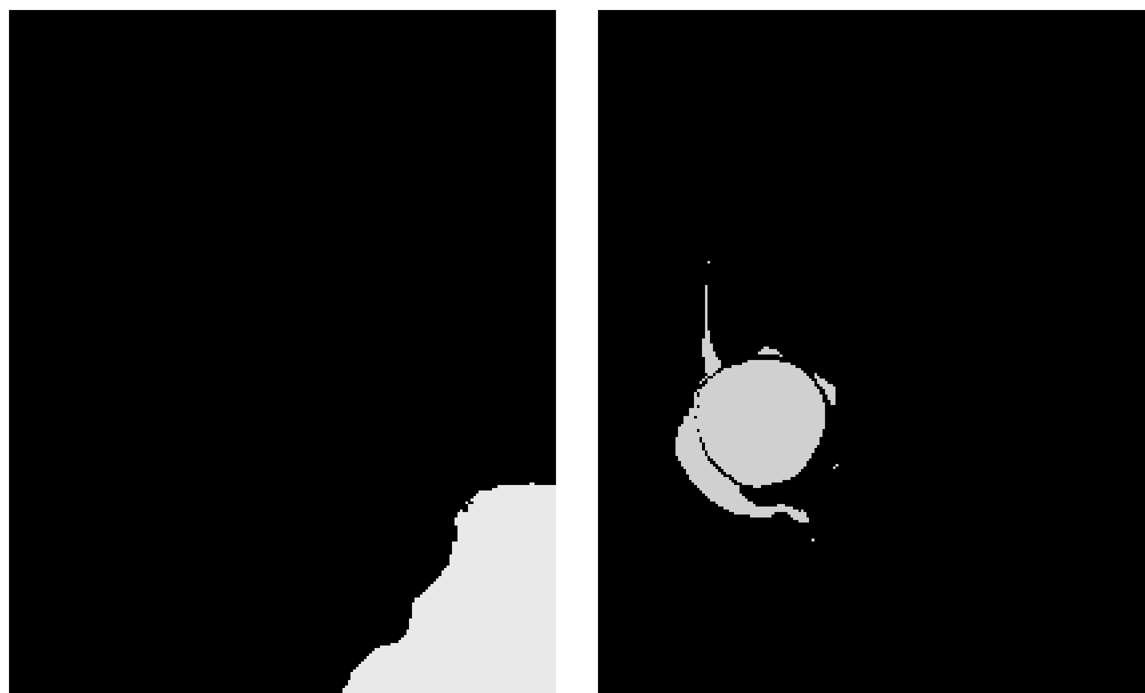
# K - Means

Image

Clusters using color alone (11 clusters)



K-means using colour and position, 20 segments



# Multiple Random Walkers