MAEG 5720:
Computer Vision in Practice

Lecture 6:

Segmentation

Dr. Terry Chang

2021-2022

Semester 1

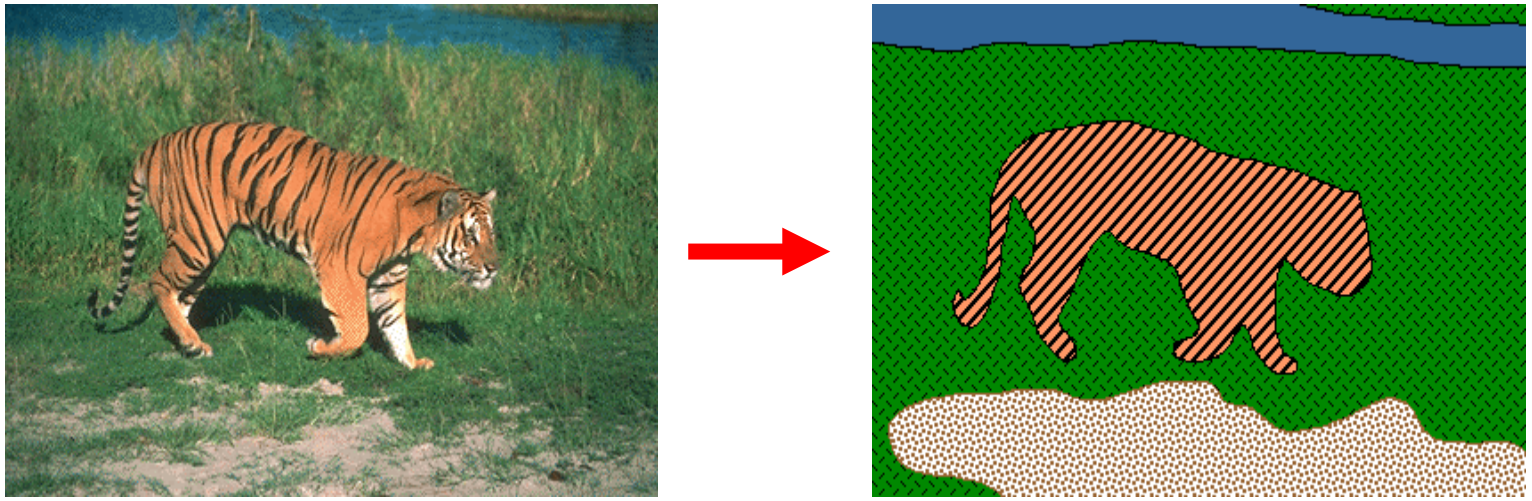
香港中文大學
The Chinese University of Hong Kong

Department of Mechanical and Automation Engineering
機械與自動化工程學系

Content

- Segmentation
  - What is segmentation
  - Human grouping
- Region-based Segmentation
  - Histogram based
- Edge-based Segmentation
- Clustering
  - K-means
  - Mean shift

Image Segmentation

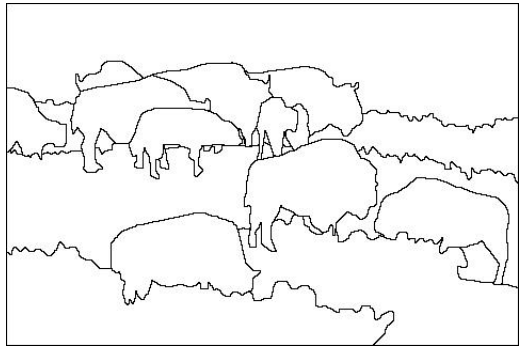


Slide credit: S. Seitz, K. Grauman

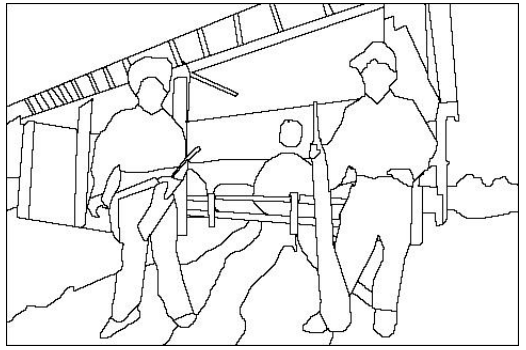
The goals of segmentation

- Separate image into coherent "objects"

image



human segmentation

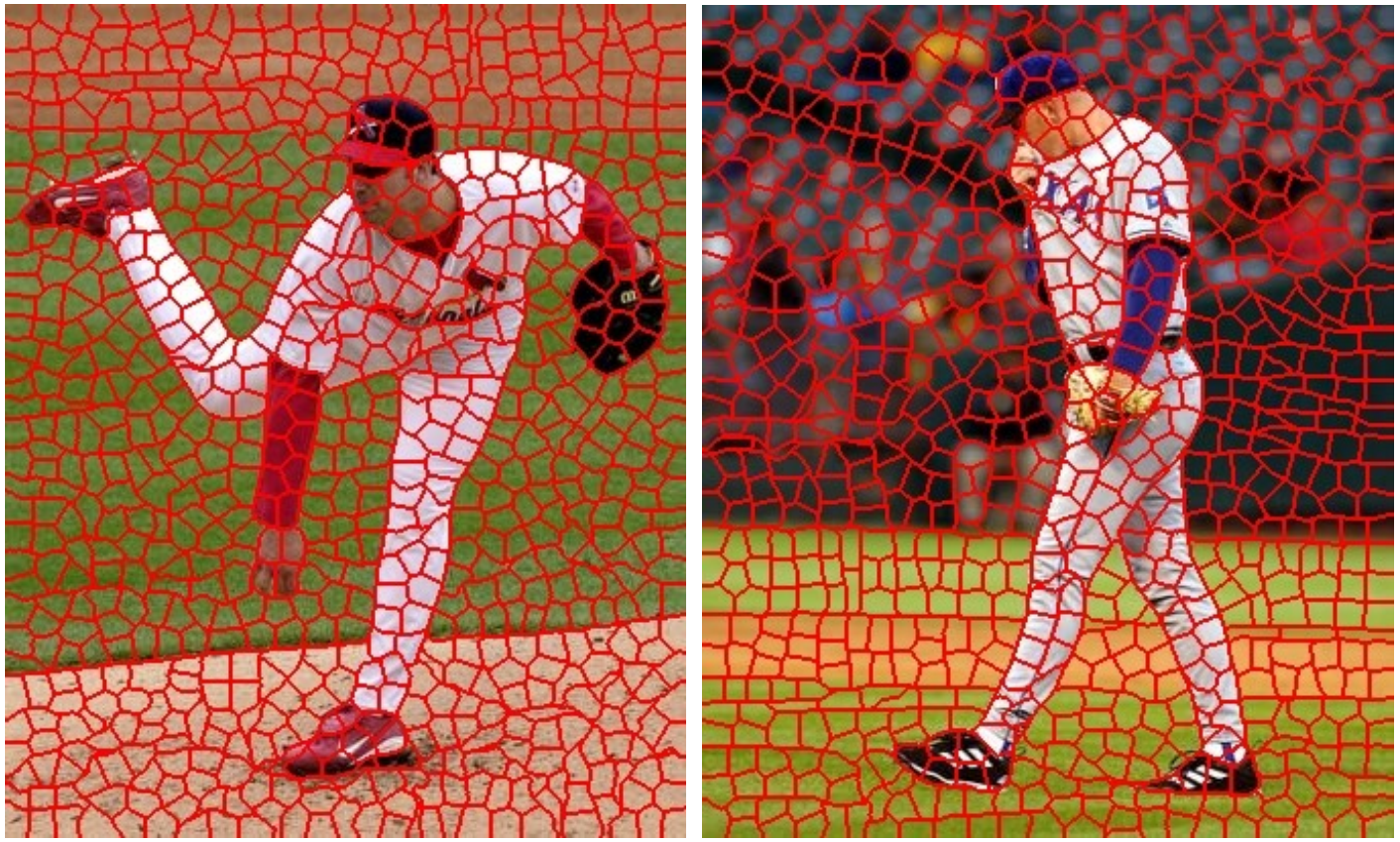


Slide credit: S. Lazebnik

The goals of Segmentation

- Separate image into coherent "objects"
- Group together similar-looking pixels for efficiency of further processing

"superpixels"

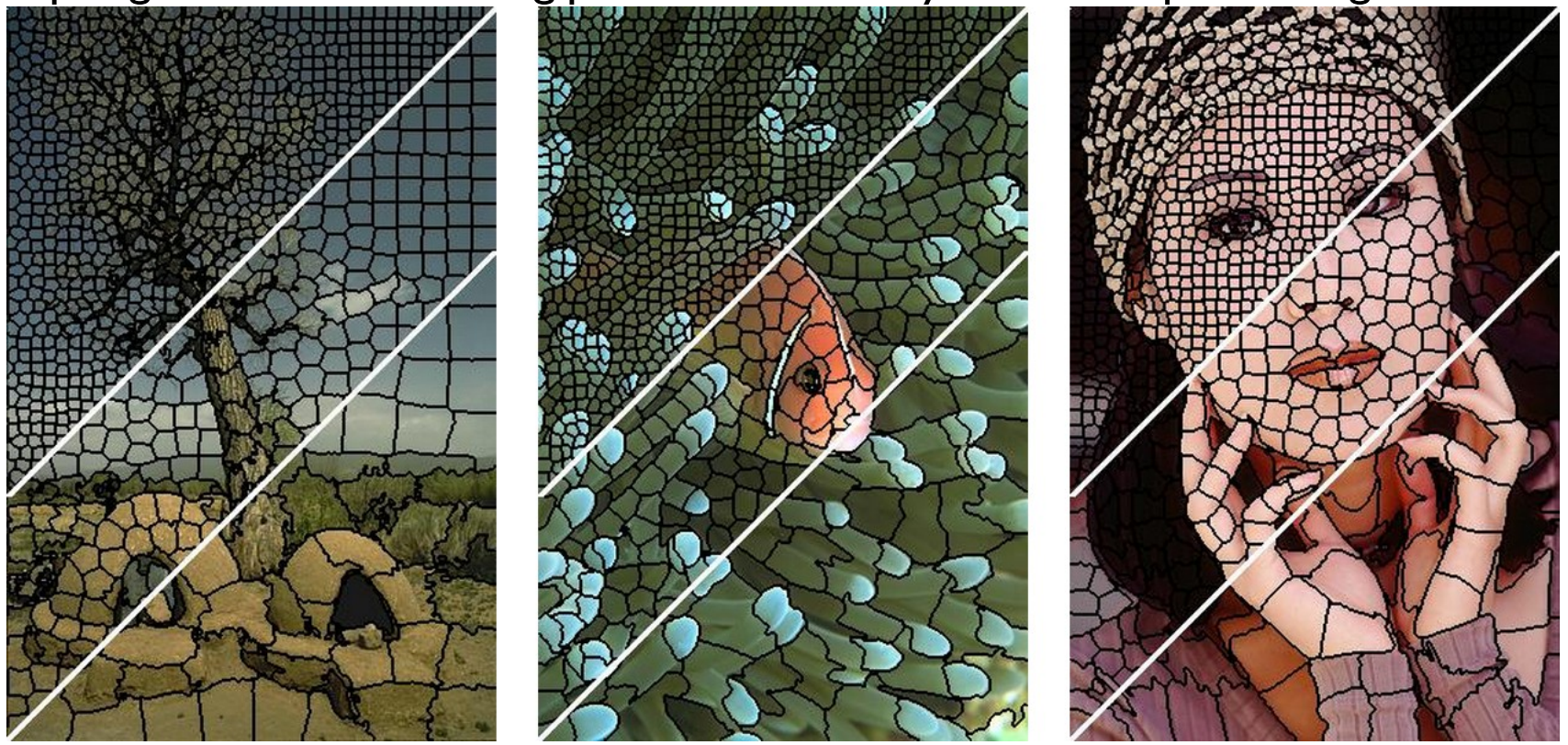


X. Ren and J. Malik. Learning a classification model for segmentation. ICCV 2003.

Slide credit: S. Lazebnik

The goals of segmentation

- Separate image into coherent "objects"
- Group together similar-looking pixels for efficiency of further processing "superpixels"



R. Achanta et al.. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. TPAMI 2012.

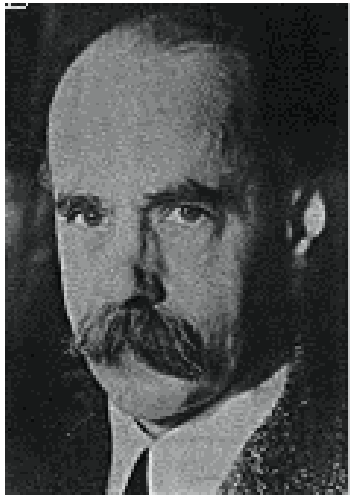
Human Segmentation

- Gestalt: whole or group
  - Whole is greater than sum of its parts
  - Relationships among parts can yield new properties/features

- Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

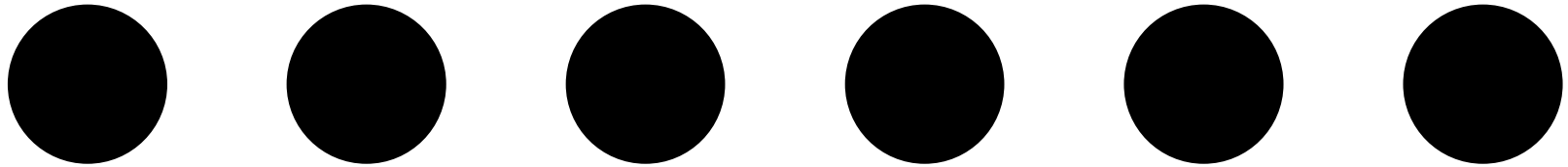
"I stand at the window and see a house, trees, sky.
Theoretically I might say there were 327 brightnesses
and nuances of colour. Do I have "327"? No. I have sky, house, and trees."

Max Wertheimer
(1880-1943)



Slide credit: J. Hays and Fei-Fei Li

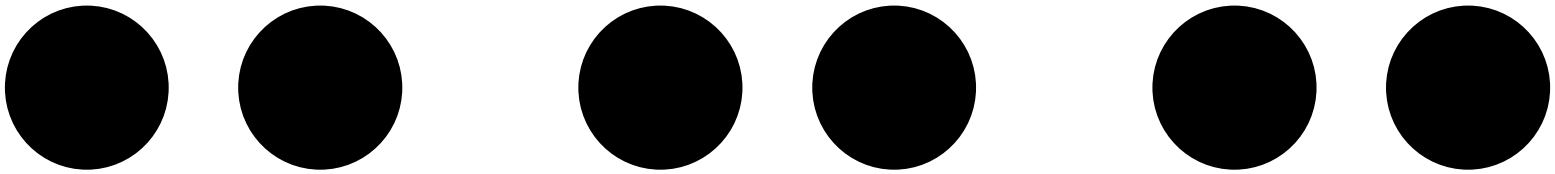
An experiment- What do you see?



Just six dots

Slide from: S. Birchfield, Clemson Univ

Now what do you see?



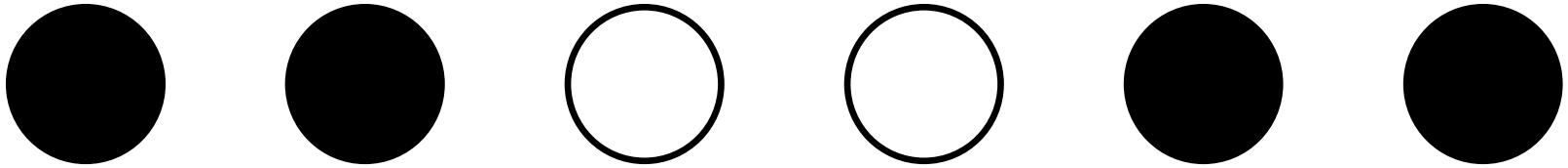
Three groups of dot pairs

Why?

Dots that are close together (“proximity”)
are grouped together by the human visual system

Slide from: S. Birchfield, Clemson Univ

And now?



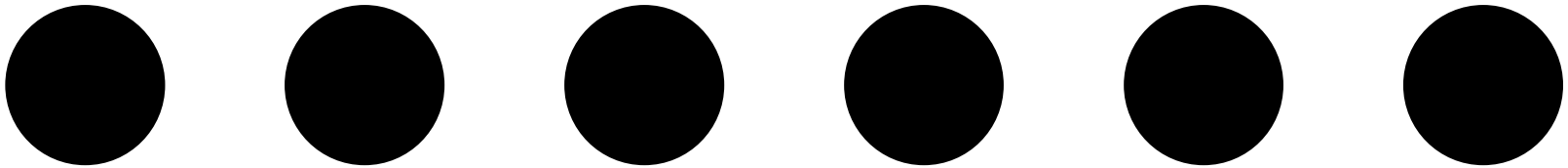
Again, three groups of dot pairs

Why?

Dots are similar in appearance (“similarity”)

Slide from: S. Birchfield, Clemson Univ

How about now?



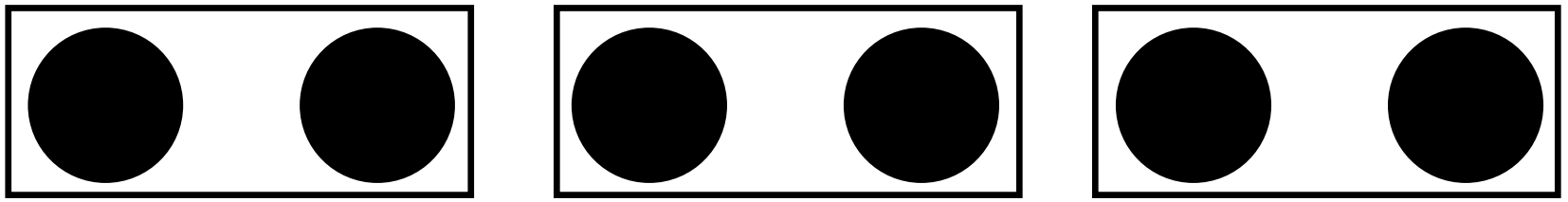
Again, three groups of dot pairs

Why?

Dots move similarly (“common fate”)

Slide from: S. Birchfield, Clemson Univ

Last one



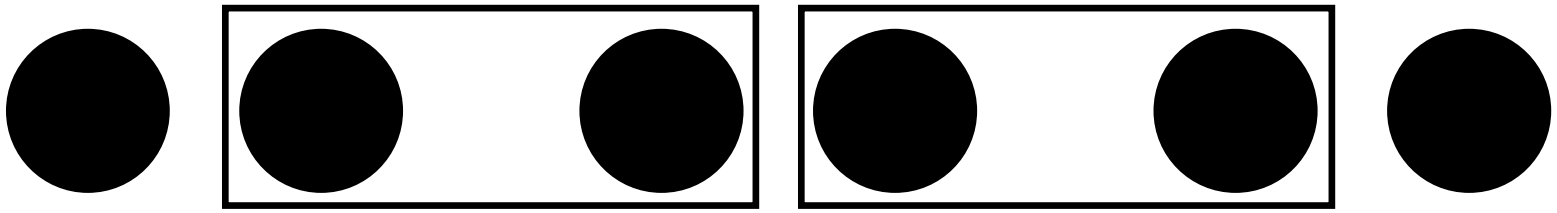
Again, three groups of dots

Why?

Dots are enclosed together (“common region”)

Slide from: S. Birchfield, Clemson Univ

But wait!



Note that the “common region” can overwhelm the “proximity” tendency

Slide from: S. Birchfield, Clemson Univ

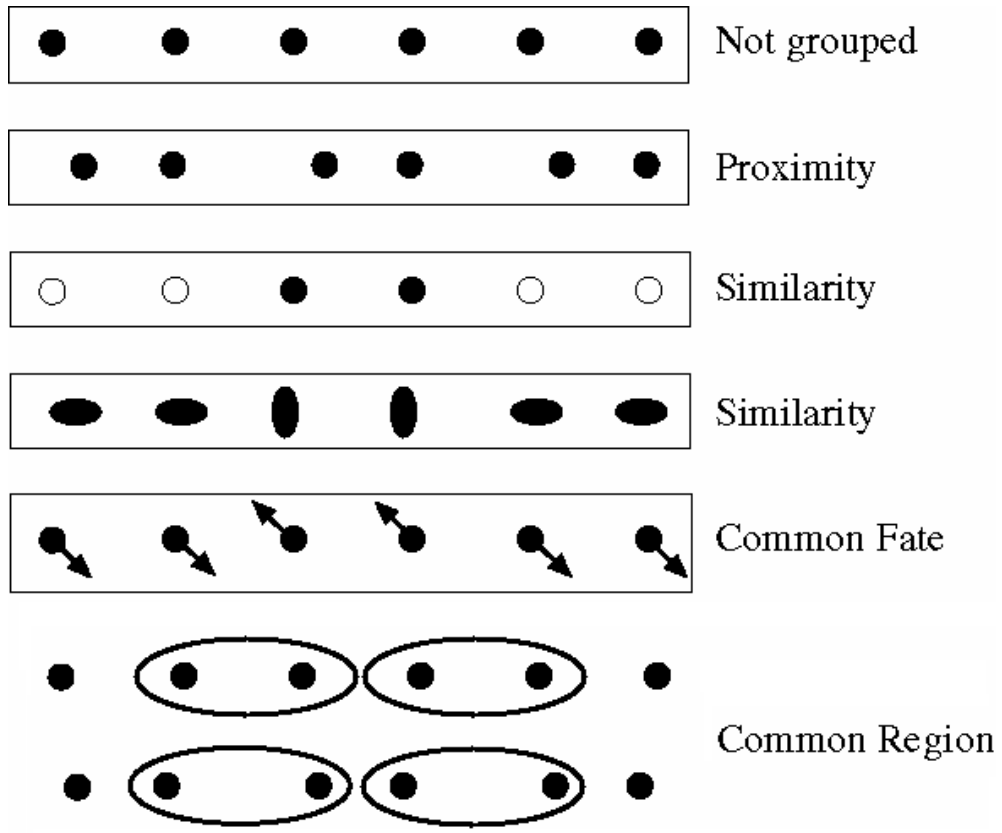
Gestalt psychology

- Gestalt school of psychologists emphasized grouping as the key to understanding visual perception.

Recall: Context affects how things are Perceived

gestalt – whole or group

gestalt qualitat – set of internal relationships that makes it a whole



Slide credit: B. Freeman and A. Torralba

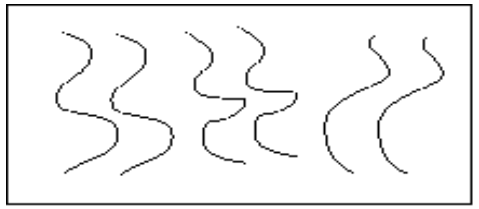
Gestalt psychology

- Gestalt school of psychologists emphasized grouping as the key to understanding visual perception.

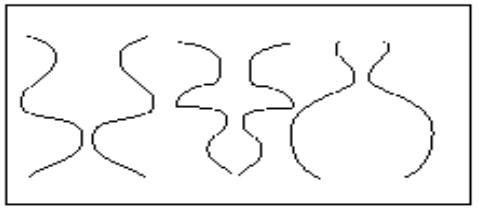
Recall: Context affects how things are Perceived

gestalt – whole or group

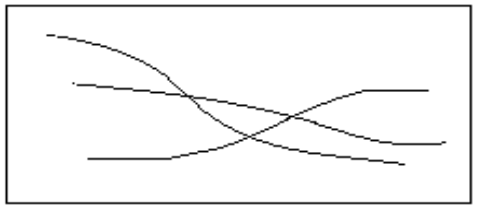
gestalt qualitat – set of internal relationships that makes it a whole



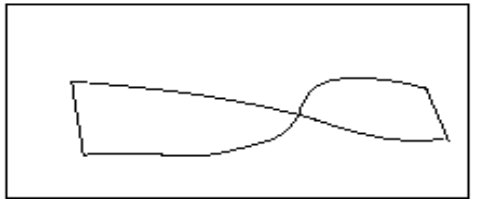
Parallelism



Symmetry



Continuity



Closure

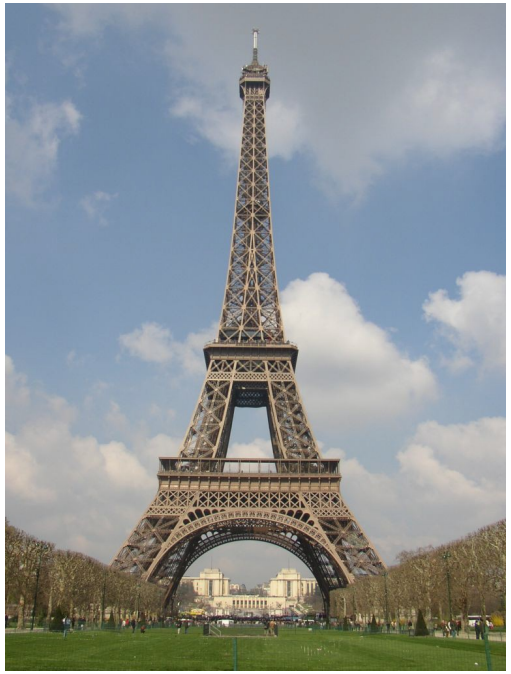
Familiarity

Slide credit: B. Freeman and A. Torralba

Similarity

Slide credit: K. Grauman

Symmetry



Slide credit: K. Grauman

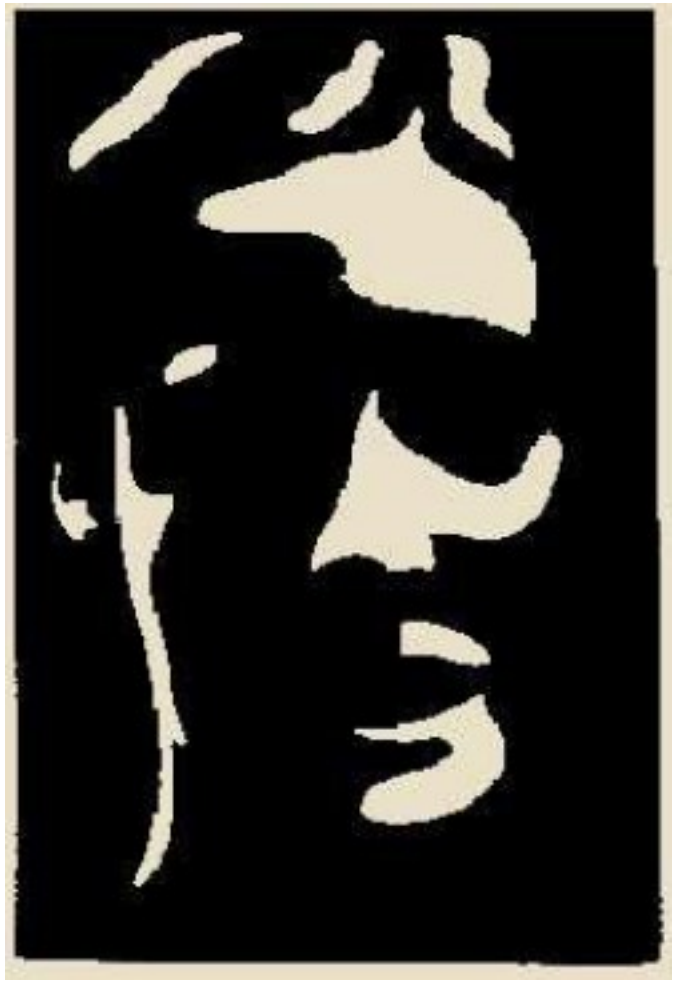
Common fate

Slide credit: K. Grauman

Proximity

Slide credit: K. Grauman

Familiarity



Slide credit: B. Freeman and A. Torralba

Gestalt cues

- Good intuition and basic principles for grouping
- Basis for many ideas in segmentation and occlusion reasoning
- Some (e.g., symmetry) are difficult to implement in practice

Suggest to watch https://www.youtube.com/watch?v=FryaH599ec0

Gestalt psychology - Wikipedia

Slide credit: J. Hays

Content

- Segmentation
  - What is segmentation
  - Human grouping
- Edge-based Segmentation

- Region-based Segmentation
  - Histogram based
- Clustering
  - Watershed
  - K-means
  - Mean shift

Snake: Active Contour

- Given: A rough contour around the desired object.
- Goal: Deform the contour to fit the object boundary

International Journal of Computer Vision, 321–331 (1988)

Snakes: Active Contour Models

MICHAEL KASS, ANDREW WITKIN, and DEMETRI TERZOPOULOS
Schlumberger Palo Alto Research, 3340 Hillview Ave., Palo Alto, CA 94304

Abstract

A snake is an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes are active contour models: they lock onto nearby edges, localizing them accurately. Scale-space continuation can be used to enlarge the capture region surrounding a feature. Snakes provide a unified account of a number of visual problems, including detection of edges, lines, and subjective contours; motion tracking; and stereo matching. We have used snakes successfully for interactive interpretation, in which user-imposed constraint forces guide the snake near features of interest.

1 Introduction

In recent computational vision research, low-level tasks such as edge or line detection, stereo matching, and motion tracking have been widely regarded as autonomous bottom-up processes. Marr and Nishihara [11], in a strong statement of this view, say that up to the 2.5D sketch, no "higher-level" information is yet brought to bear: the computations proceed by utilizing only what is available in the image itself. This rigidly sequential approach propagates mistakes made at a low level without opportunity for correction. It therefore imposes stringent demands on the reliability of low-level mechanisms. As a weaker but more attainable goal for low-level processing, we argue that it ought to provide sets of alternative organizations among which higher-level processes may choose, rather than shackling them prematurely with a unique answer.

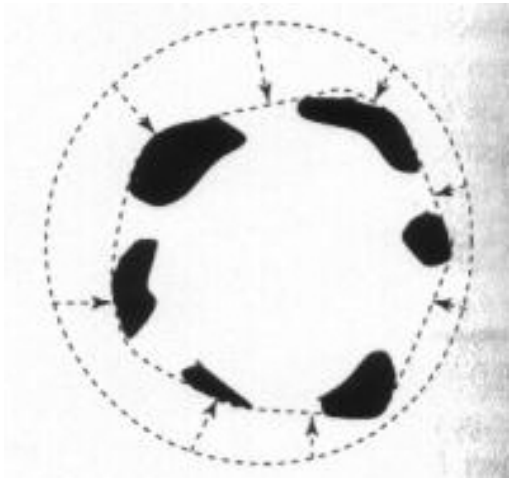
organizations. By adding suitable energy terms to the minimization, it is possible for a user to push the model out of a local minimum toward the desired solution. The result is an active model that falls into the desired solution when placed near it.

Energy minimizing models have a rich history in vision going back at least to Sperling's stereo model [16]. Such models have typically been regarded as autonomous, but we have developed interactive techniques for guiding them. Interacting with such models allows us to explore the energy landscape very easily and develop effective energy functions that have few local minima and little dependence on starting points. We hope thereby to make the job of high-level interpretation manageable yet not constrained unnecessarily by irreversible low-level decisions.

The problem domain we address is that of finding salient image contours—edges, lines, and

General Idea

- What is a snake?
- Snake is like a rubber band which is attracted by the image positions with high intensity gradient and also maintain certain *shape properties*.



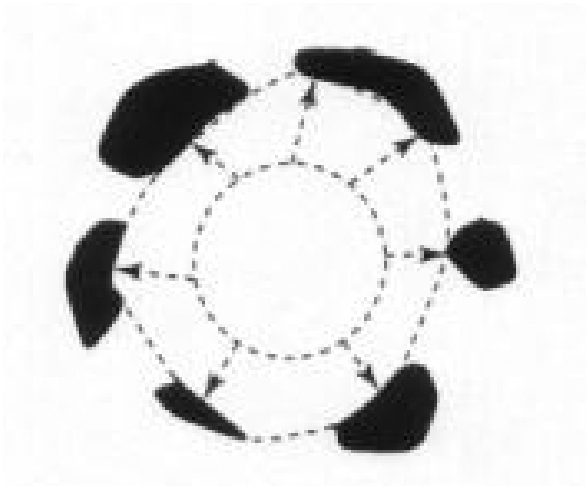
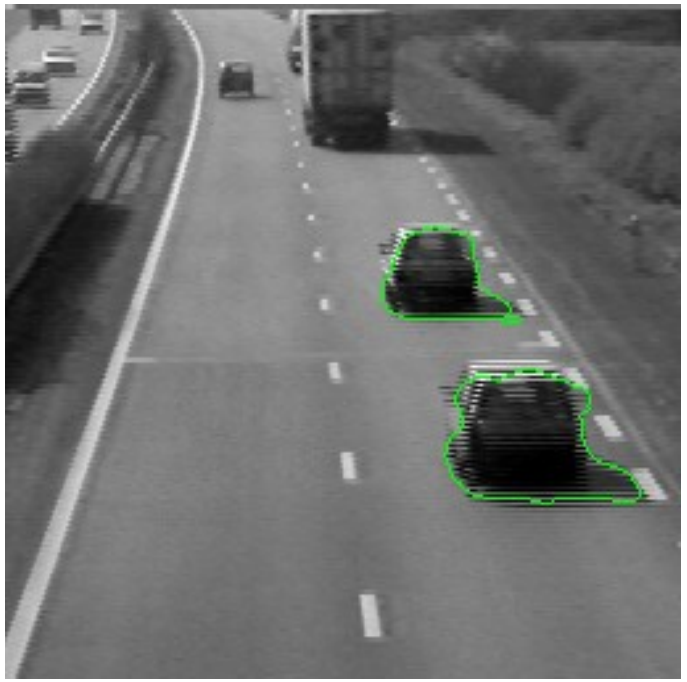
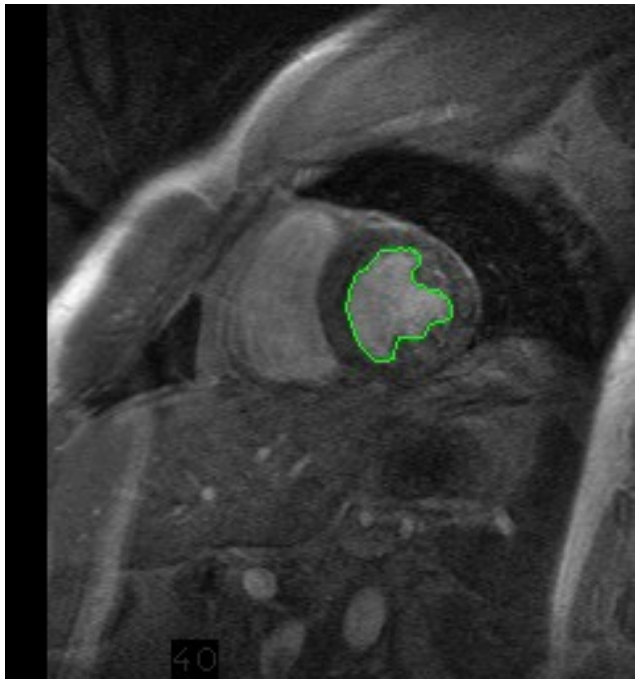


Image Credit : https://www.1-hp.org/blog/healthy-movement/fingerthumb-extension-with-rubber-band/

Why do we want to fit snakes?





- Some objects we need to track has deformable outline due to changing orientations or non-rigid bodies.

Snake Representation

The positions of the snake is represented parametrically

$$\boldsymbol{v}(s) = \big(x(s), y(s)\big)$$

In discrete form, it is represented by 2D points

$$\boldsymbol{v_i} = (x_i, y_i) \text{ for } i = 0,1, \dots, n$$

- At each step, $\boldsymbol{v_i}$ will be adjusted to it's neighbour locations to minimize the energy

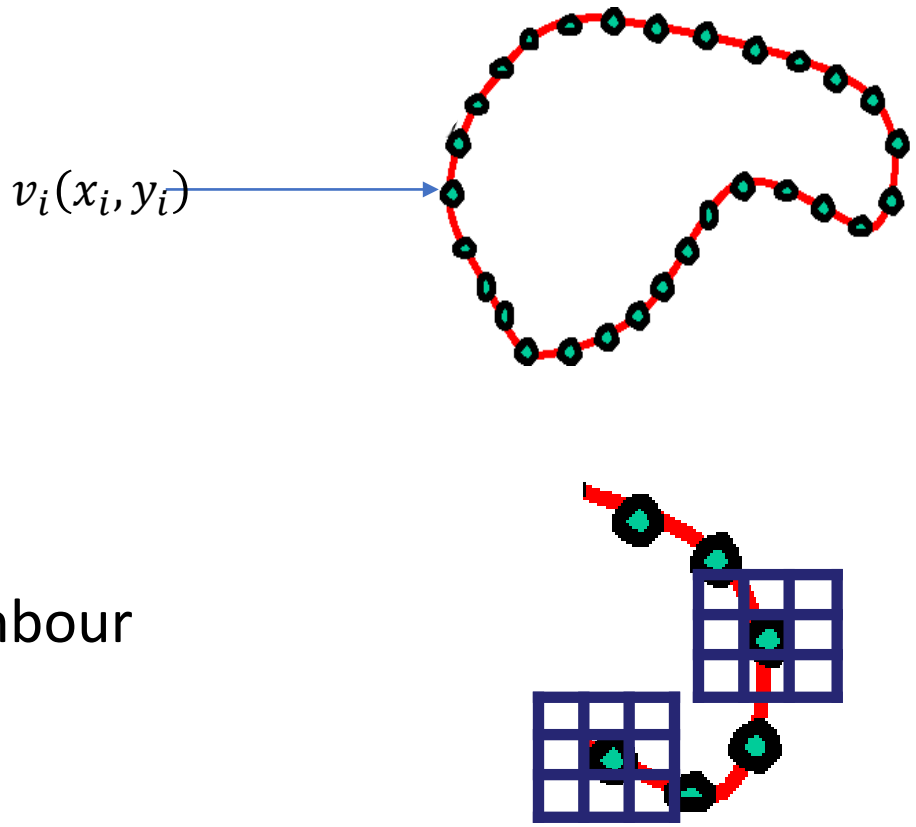


Image credit: E. Erdem (HUCVL)

Fitting Deformable Contour

How to adjust the current contour at each iteration?

- Define a cost function ("energy" function) that how good the configuration is.
- Minimizes that cost function by adjusting the configuration.

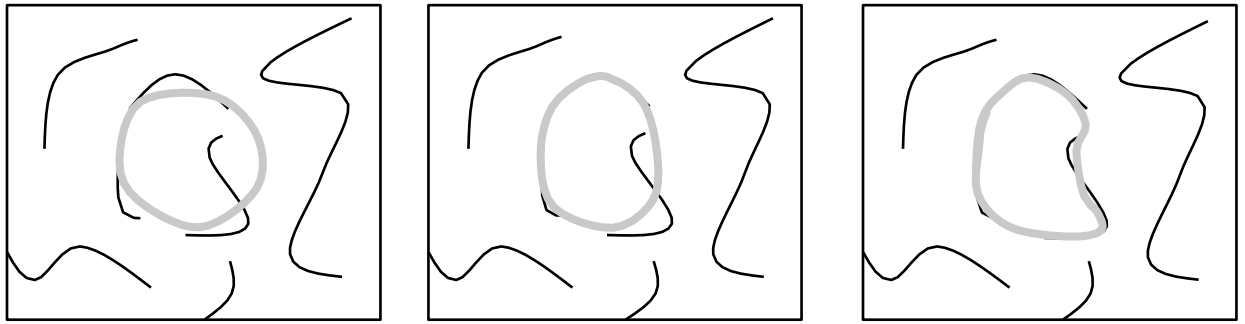


Image credit: E. Erdem (HUCVL)

The energy function of snake

- The ***total energy*** cost of a snake is defined as

$$E^*_{sanke} = \int_0^1 E_{snake}(v(s))ds$$
$$E^*_{sanke} = \int_0^1 E_{int}(v(s)) + E_{ext}\big(v(s)\big)ds$$

- ***Internal Energy***: Control the ***shape*** properties by ensuring ***Smoothness*** and ***Elasticity***
- ***External Energy:*** Also known as ***image energy*** by encouraging the snake fits to intensity gradient locations.
- **Goal:** minimize the energy cost function.

Internal Energy

- Control the ***shape*** properties by ensuring ***Smoothness*** and ***Elasticity***

- At position $v(s)$ the internal energy is defined as follow:

$$E_{internal} = \alpha \left|\frac{dv}{ds}\right|^2 + \beta \left|\frac{d^2v}{ds^2}\right|^2$$

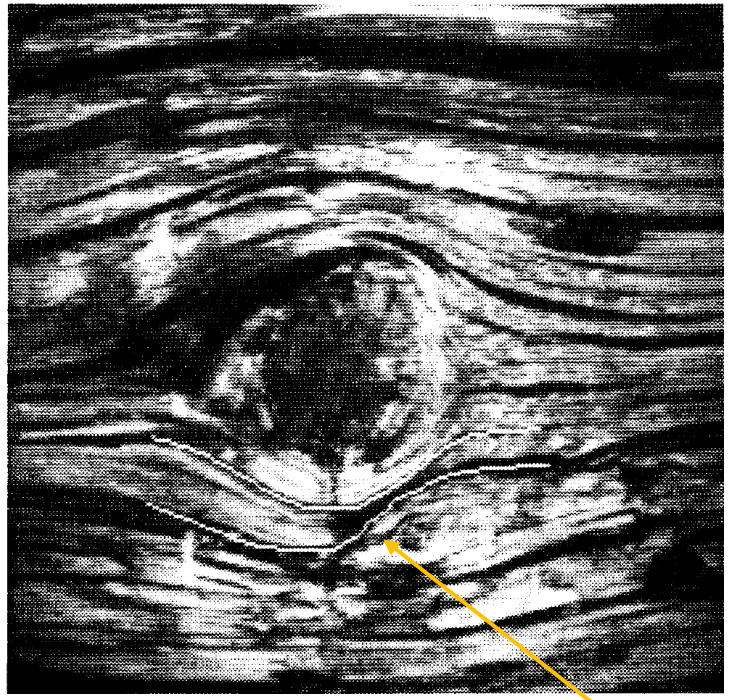
- **First-order term:** maintains the elasticity (tension) of the snake
- **Second-order term:** defines the smoothness (stiffness) of the snake

where α and β are the weight given to two terms.

External Image Energy

- Define how well ***snake*** matches the image features such as ***lines*** and ***edges***.
- ***Lines*** and ***edges*** are regions with ***high intensity gradient***
- Located at ***zero crossing*** of $G_\sigma * \nabla^2(I)$
- ***External Energy*** is defined as

$$\boldsymbol{E_{ext}(v) = -(G_x(v)^2 + G_y(v)^2)}$$



Total energy function

- $E_{total} = E_{internal} + \gamma E_{external}$
- $E_{external} = -\sum_{i=0}^{n-1} |G_x(x_i, y_i)|^2 + \left|G_y(x_i, y_i)\right|^2$
- $E_{internal} = \sum_{i=0}^{n-1} \alpha |(v_i - v_{i-1})|^2 + \beta |(v_{i-1} - 2v_i + v_{i+1})|^2$

The energy minimization

Methodology:

- For each point, *search window* around it and move to where energy function is minimal
  - Typical window size, e.g., 5 x 5 pixels
- Stop when predefined number of points have *not changed* in last iteration, or after max number of iterations
- Note:
  - Convergence not guaranteed
  - Need decent initialization

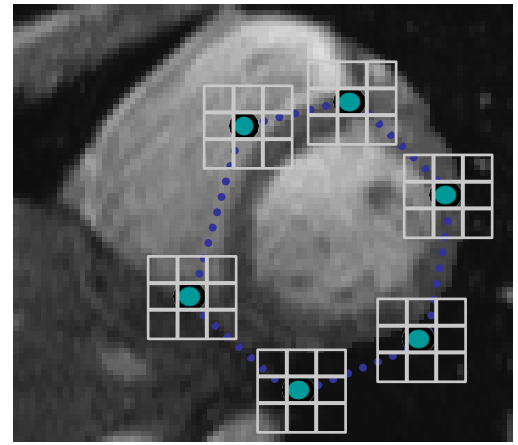
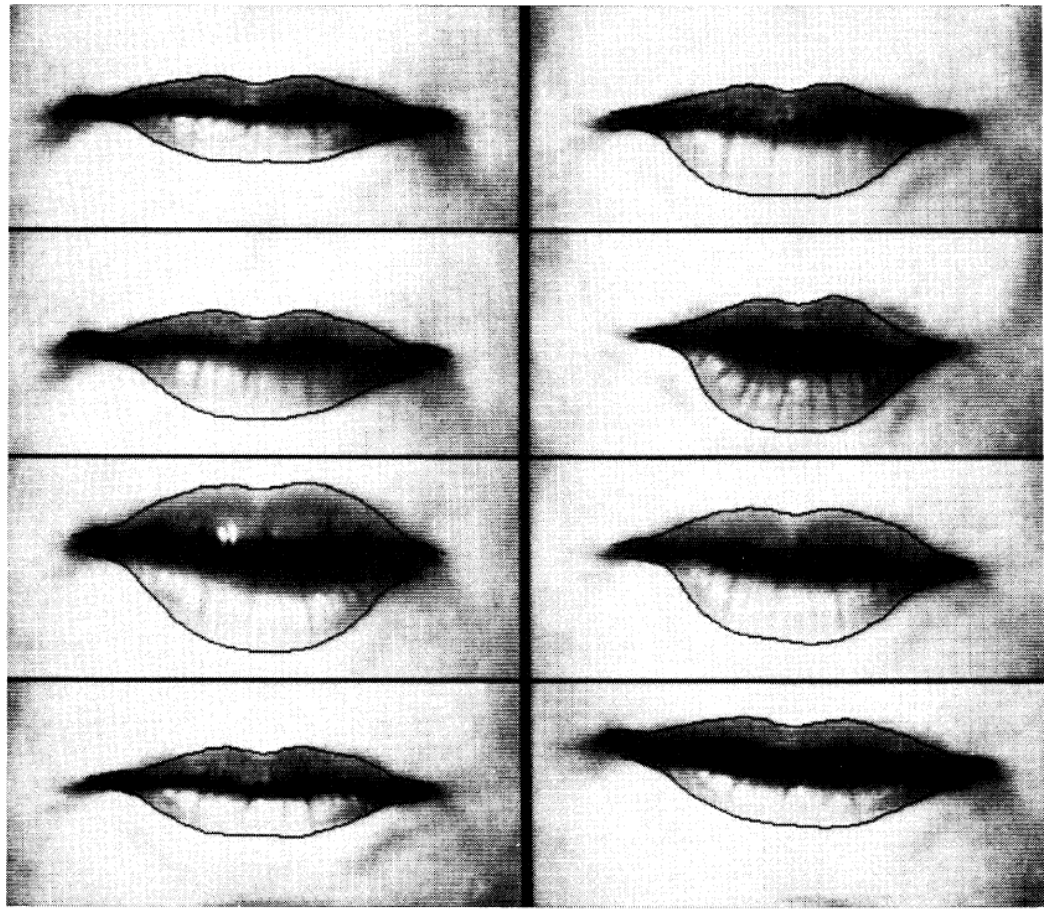


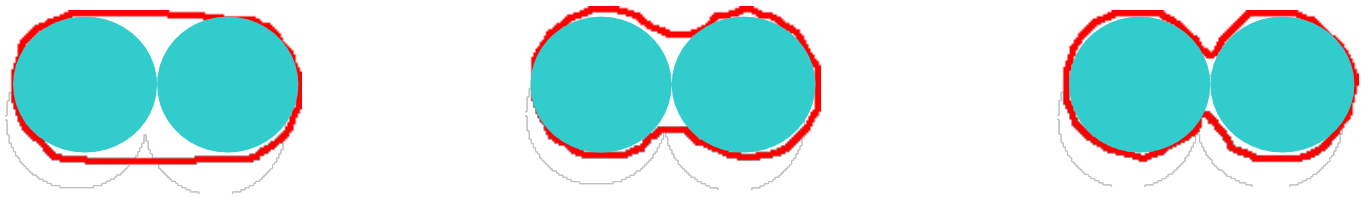
Image credit: E. Erdem (HUCVL)

Examples



Total Energy function weight

- e.g.,α weight controls the penalty for internal elasticity

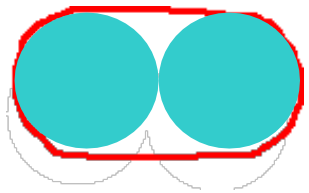


large α medium α small α

Fig from Y. Boykov

Limitations

- May over-smooth the boundary



- Cannot follow topological changes of objects

Image credit: E. Erdem (HUCVL)

Limitations

- The snake must be very close to the boundary at initial state

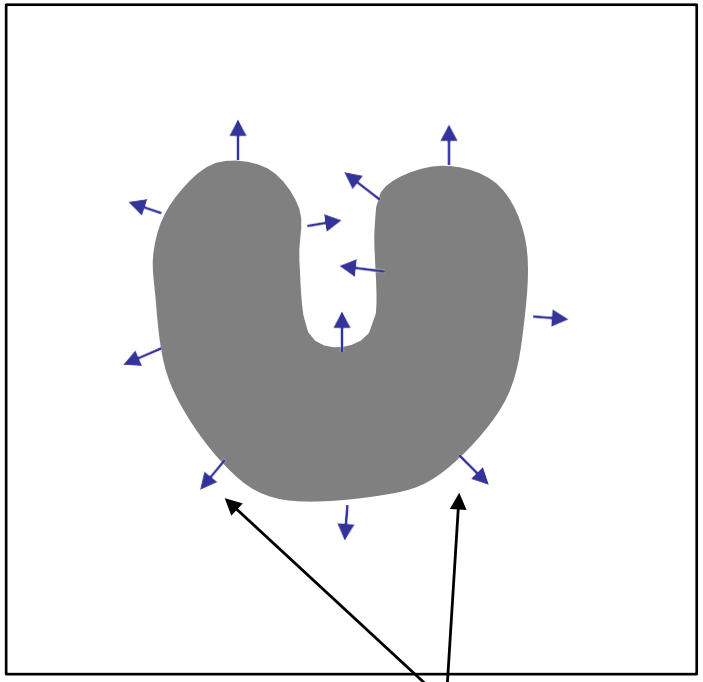


image gradients ∇I
are large only directly on the boundary

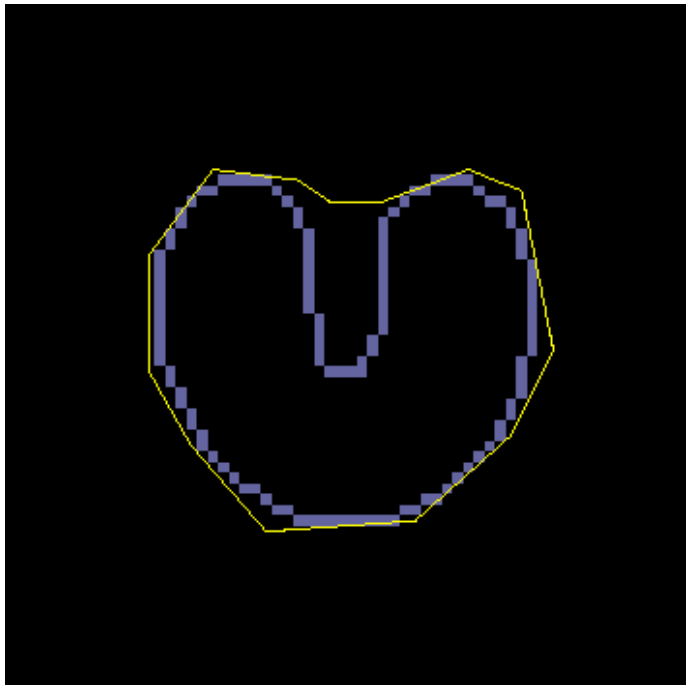


Image credit: E. Erdem (HUCVL)

Summary of Active Contour

- Advantages:
  - Useful to track non-rigid object
  - Close contour remains
  - Can adjust the weight of the energy function
- Disadvantages:
  - Need to interactively initialize near the boundary
  - Need to guess the parameters of the energy functions.

Content

- Segmentation
  - What is segmentation
  - Human grouping
- Edge-based Segmentation
  - Snake: Active Contour
- Region-based Segmentation
  - Histogram based
- Clustering
  - Watershed
  - K-means
  - Mean shift

Segmentation by Threshold

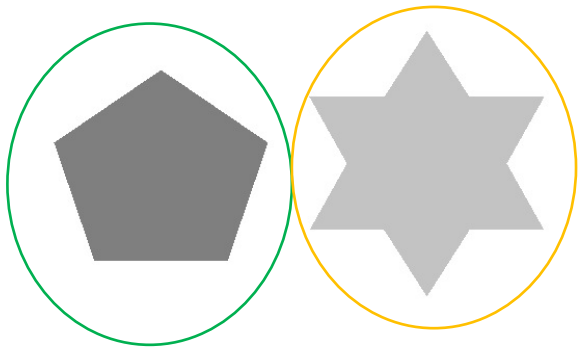
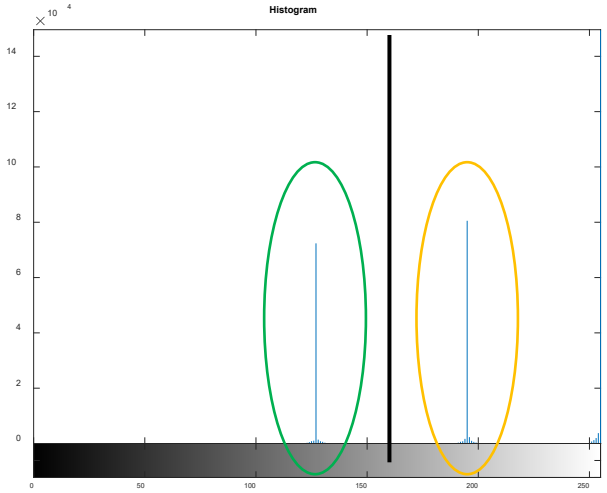


Image Input

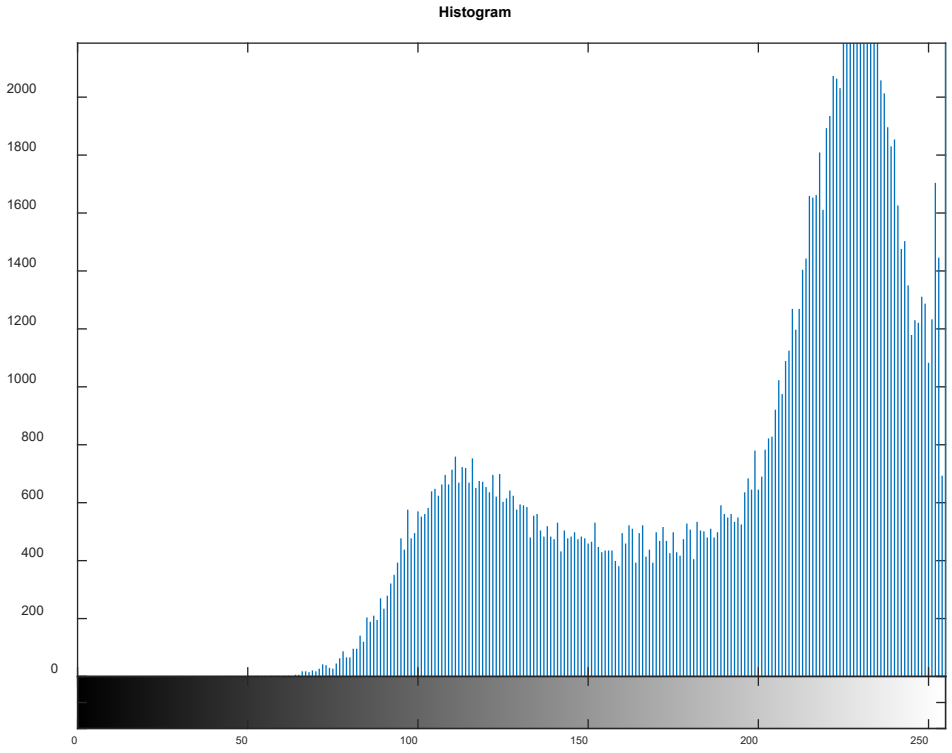


Histogram of Intensity

Two groups are identified

In real example (noisy image)

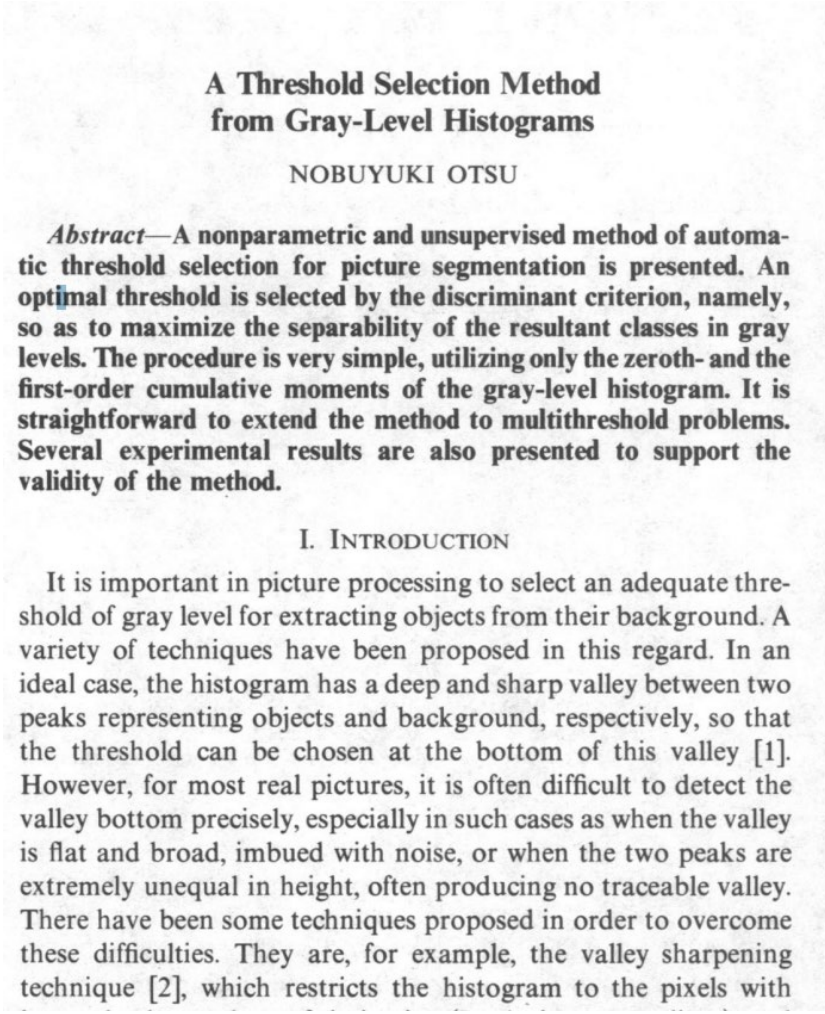
- Histogram based algorithm



Otsu's Method

Goal:

1. To perform *automatic image thresholding* and
2. Return a single intensity threshold that *separate* pixels into *2 classes*. (Foreground and Background)



A Threshold Selection Method
from Gray-Level Histograms

NOBUYUKI OTSU

Abstract—**A nonparametric and unsupervised method of automatic threshold selection for picture segmentation is presented. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. The procedure is very simple, utilizing only the zeroth- and the first-order cumulative moments of the gray-level histogram. It is straightforward to extend the method to multithreshold problems. Several experimental results are also presented to support the validity of the method.**

I. INTRODUCTION

It is important in picture processing to select an adequate threshold of gray level for extracting objects from their background. A variety of techniques have been proposed in this regard. In an ideal case, the histogram has a deep and sharp valley between two peaks representing objects and background, respectively, so that the threshold can be chosen at the bottom of this valley [1]. However, for most real pictures, it is often difficult to detect the valley bottom precisely, especially in such cases as when the valley is flat and broad, imbued with noise, or when the two peaks are extremely unequal in height, often producing no traceable valley. There have been some techniques proposed in order to overcome these difficulties. They are, for example, the valley sharpening technique [2], which restricts the histogram to the pixels with

Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". *IEEE Trans. Sys. Man. Cyber.* **9** (1): 62–66

Otsu's Method

- **General Idea:** Compute the within-class variance at each level of $k(1\sim256)$, minimize the value (MATLAB starts with 1)

- The ***Probability*** $P(k)$ of each bin is defined as

$$P(k) = n(k)/N \; k \in [1\sim256]$$

- The ***within-class variance*** σ_w^2 is defined as the weighted sum of variance of two class σ_0^2 and σ_1^2 at particular threshold value k

$$\sigma_w^2(k) = w_0(k)\sigma_0^2(k) + w_1(k)\sigma_1^2(k)$$

- weight $w_0(k)$ and $w_1(k)$ are the probabilities of two classes separated by threshold k

- w_0, w_1 are computed from the L-bins histogram

$$w_0(k) = \sum_{i=0}^{k-1} p(i)$$
$$w_1(k) = \sum_{i=k}^{L} p(i) = 1 - w_0(k)$$

Otsu's Method

- The ***within-class variance*** σ_w^2:

$$\sigma_w^2(k) = w_0(k)\sigma_0^2(k) + w_1(k)\sigma_1^2(k)$$

- The ***individual class variance*** is defined as

$$\sigma_0^2 = \sum_{i=1}^{k} (i - \mu_0)^2 p_i / w_0$$

$$\sigma_1^2 = \sum_{i=k+1} (i - \mu_0)^2 p_i / w_1$$

where μ_0 and μ_1 are the mean of two groups.

$$\mu_0 = \sum_{i=1}^{k} i p_i / w_0$$

$$\mu_1 = \sum_{i=k+1}^{L} i p_i / w_1$$

$$\mu_T = \sum_{i=1}^{L} i p_i / w_T = w_0\mu_0 + w_1\mu_1$$

Otsu's Method

We define

$$\sigma_W^2 + \sigma_B^2 = \sigma_T^2$$

- σ_B^2 is the between classes variance
- σ_W^2 is the within class variance
- σ_T^2 is the total variance

- We already have within-class variance:

$$\sigma_w^2 = w_0\sigma_0^2 + w_1\sigma_1^2$$

The between class variance is defined as

$$\sigma_B^2 = w_0 w_1 (\mu_1 - \mu_0)^2$$

Goal: find the optimal k^* where

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

Pseudo Code for Otsus' Method

- 10 Read the image (Gray)
- 20 Generate the histogram of intensity [1, 256]
- 30 Calculate the probability $P(k)$, of each intensity level
- 40 for k = 2 to 255

  calculate weight $w_0(k)$, $w_1(k)$

  calculate mean $\mu_0(k)$, $\mu_1(k)$

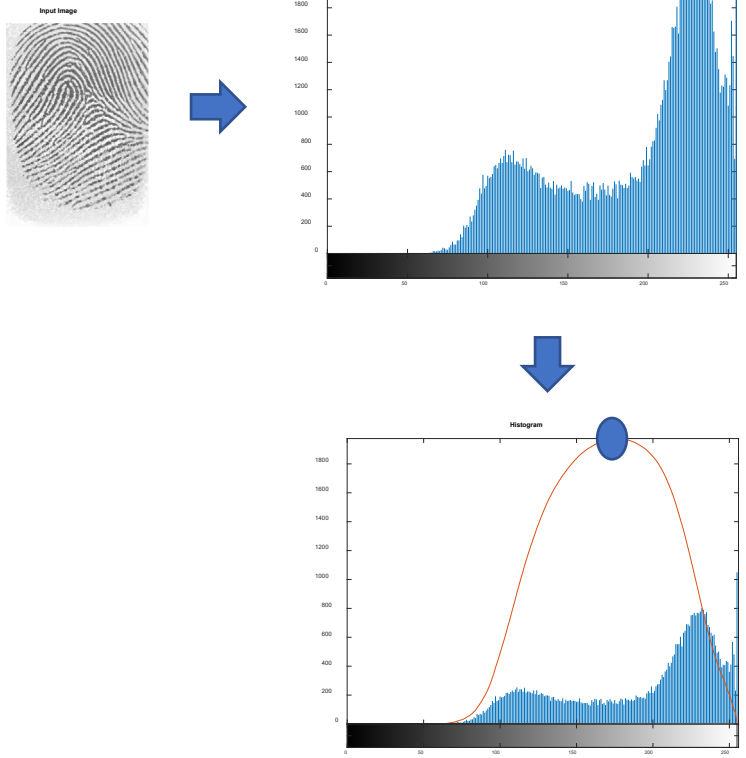
  calculate $\sigma_B^2\ (k)$

  end

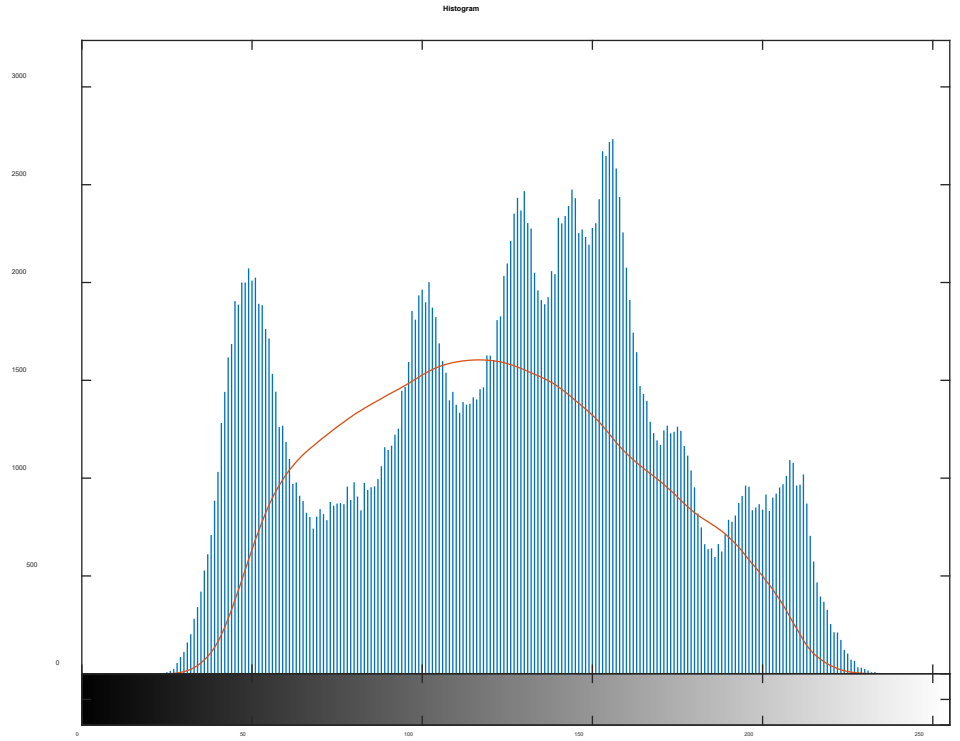
50 Find the value of k where $\sigma_B^2(k)$ is maximum.

60 Threshold = k

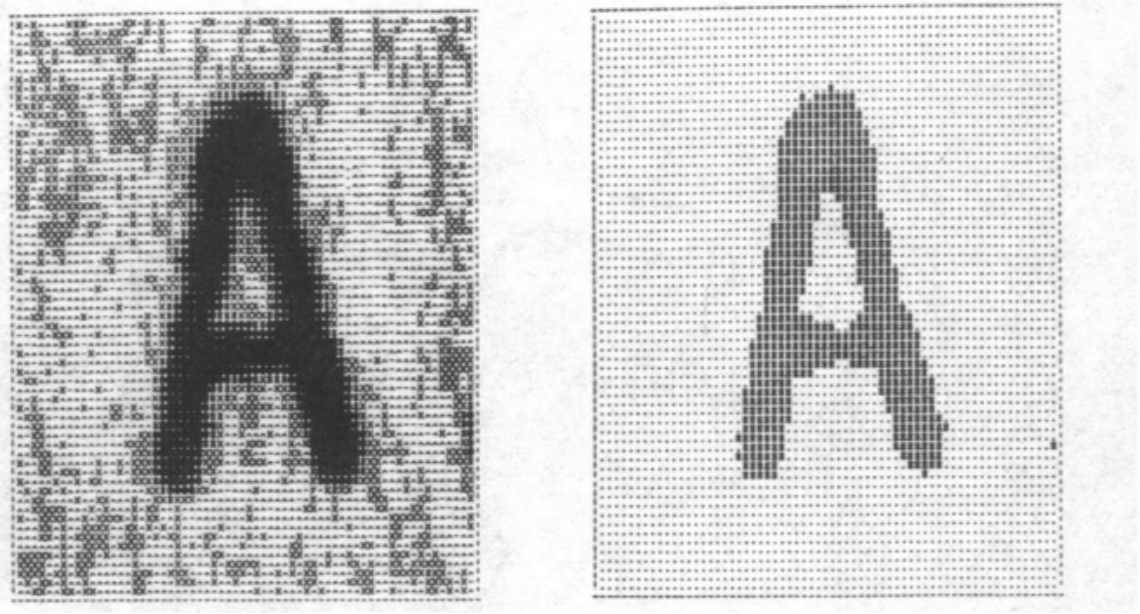
70 Convert the image to Black and White using k



Example



Other Examples from the paper



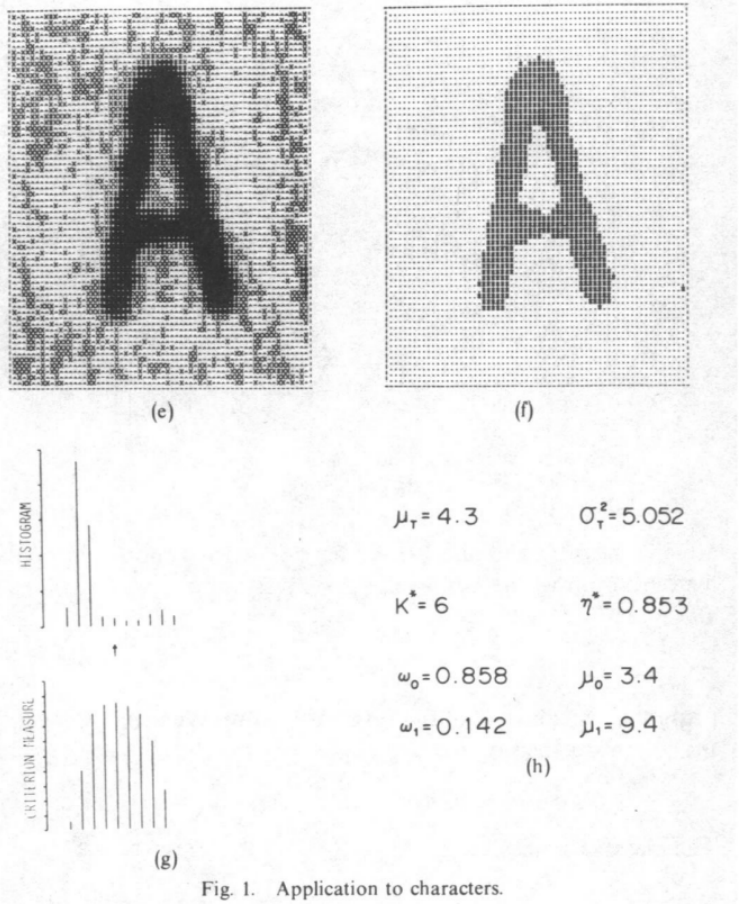
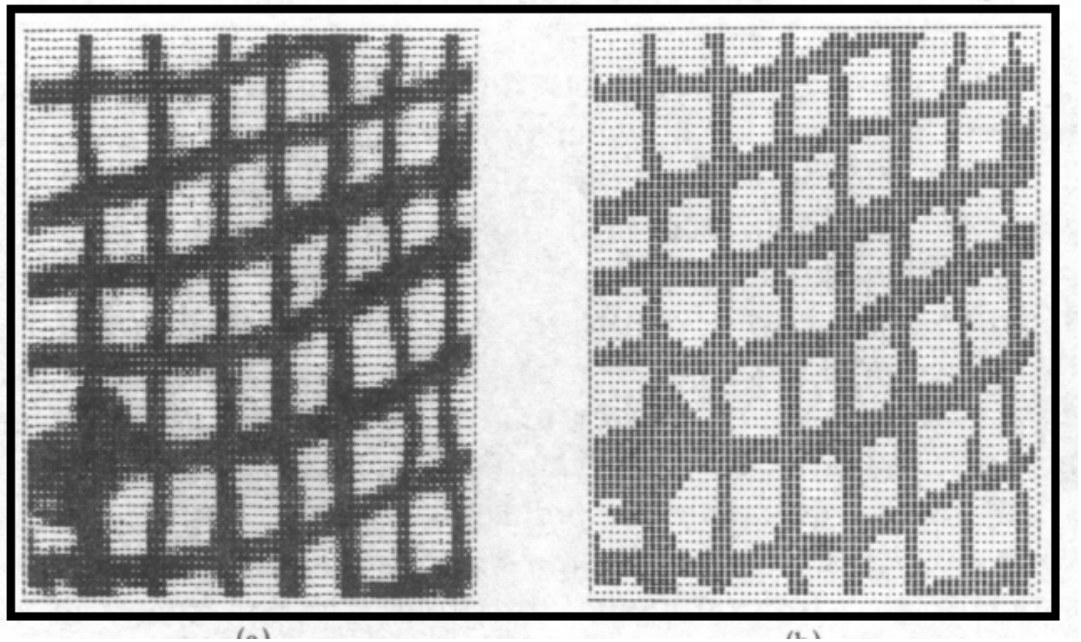
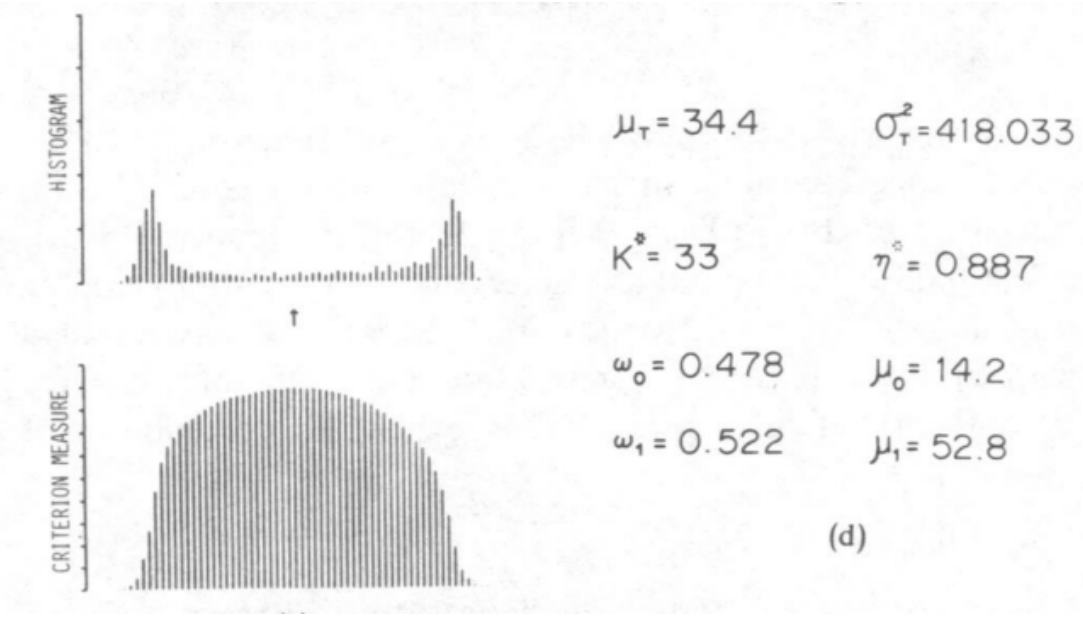


Fig. 1. Application to characters.

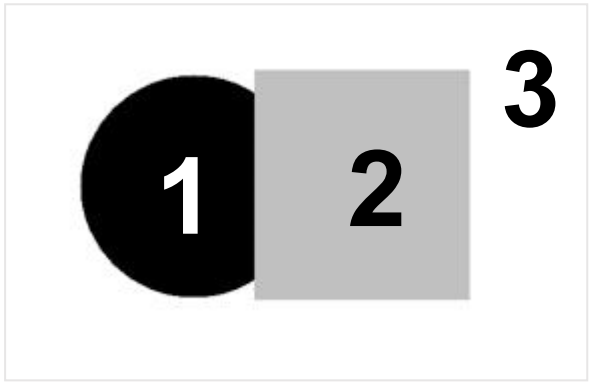
Other Examples from the paper



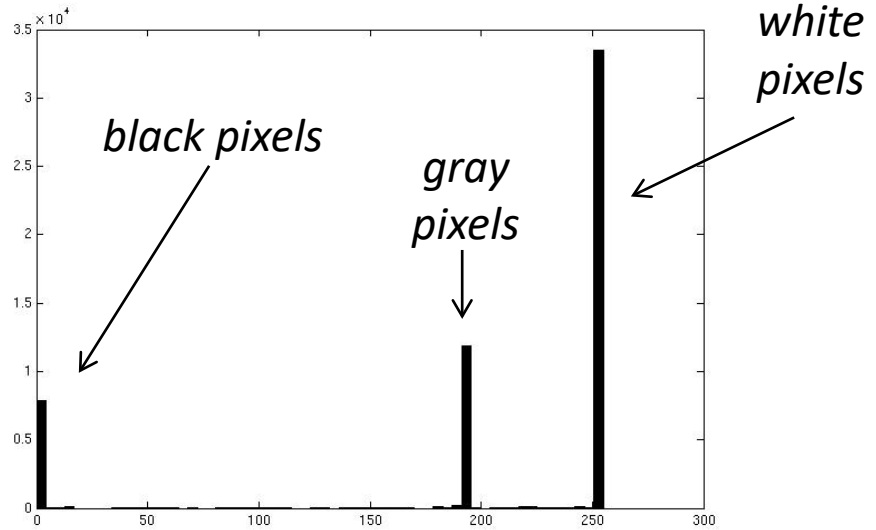


What if we have more group?

Segmentation Toy Examples



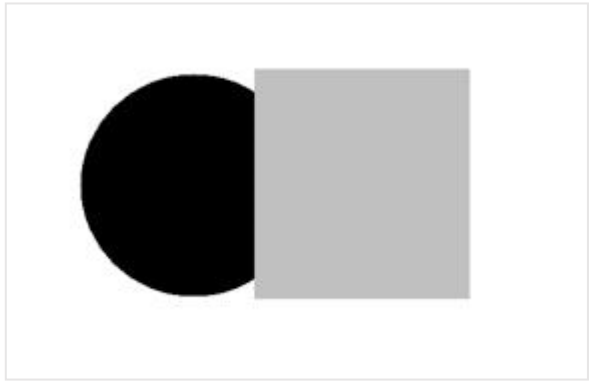
input image



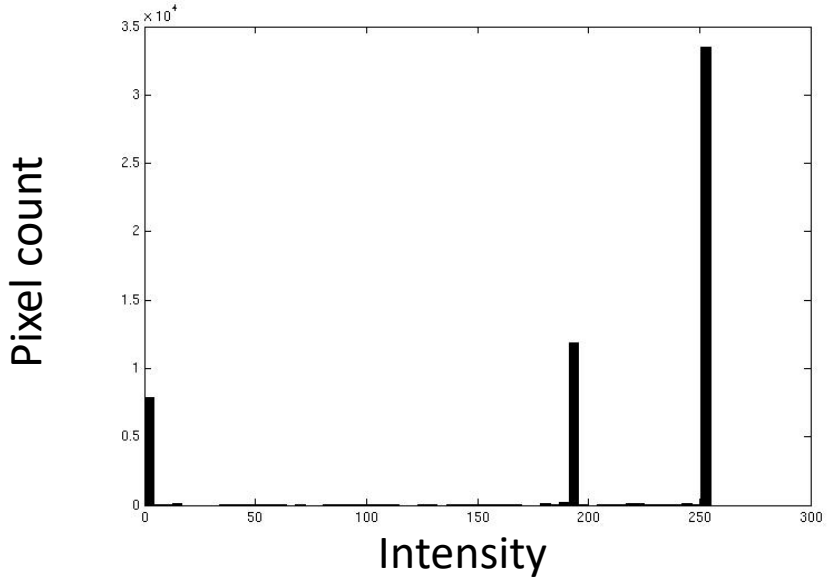
intensity

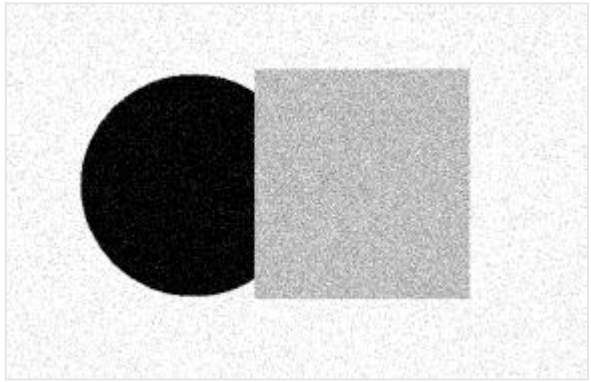
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.
- What if the image isn't quite so simple?

Slide credit: Kristen Grauman

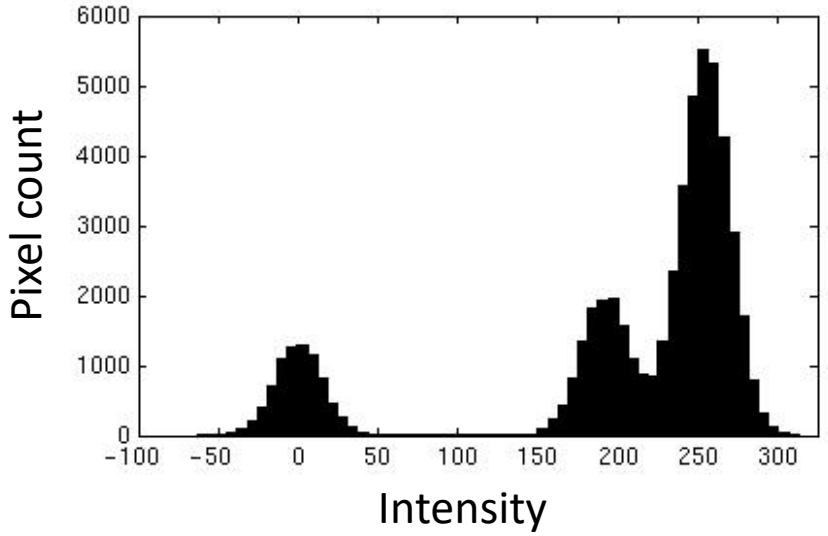


Input image



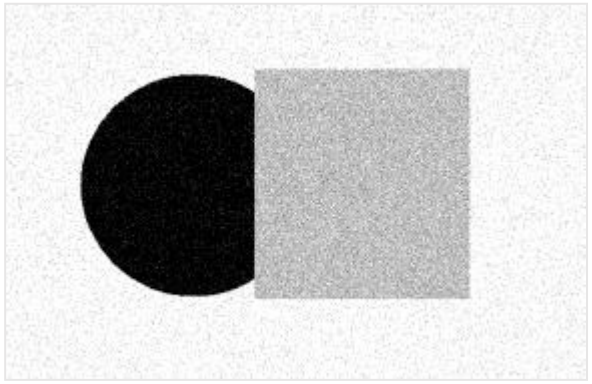


Input image

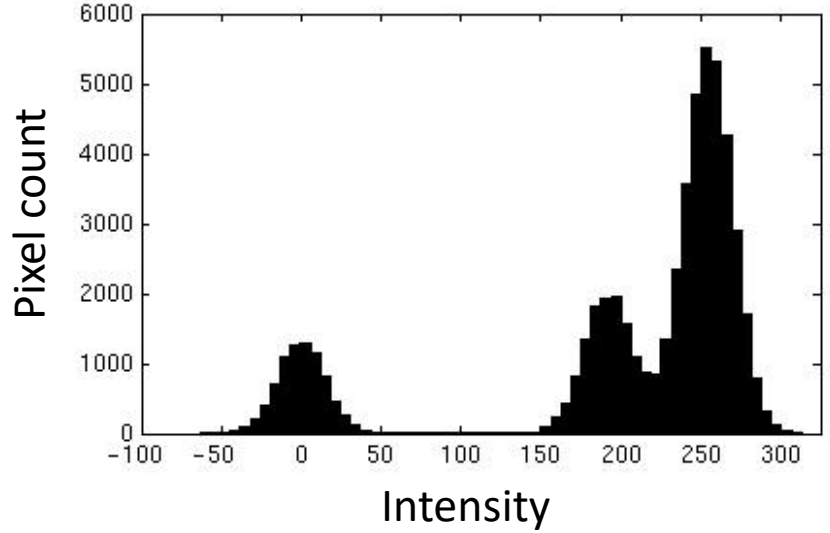


Slide credit: Kristen Grauman

Segmentation: Toy Example



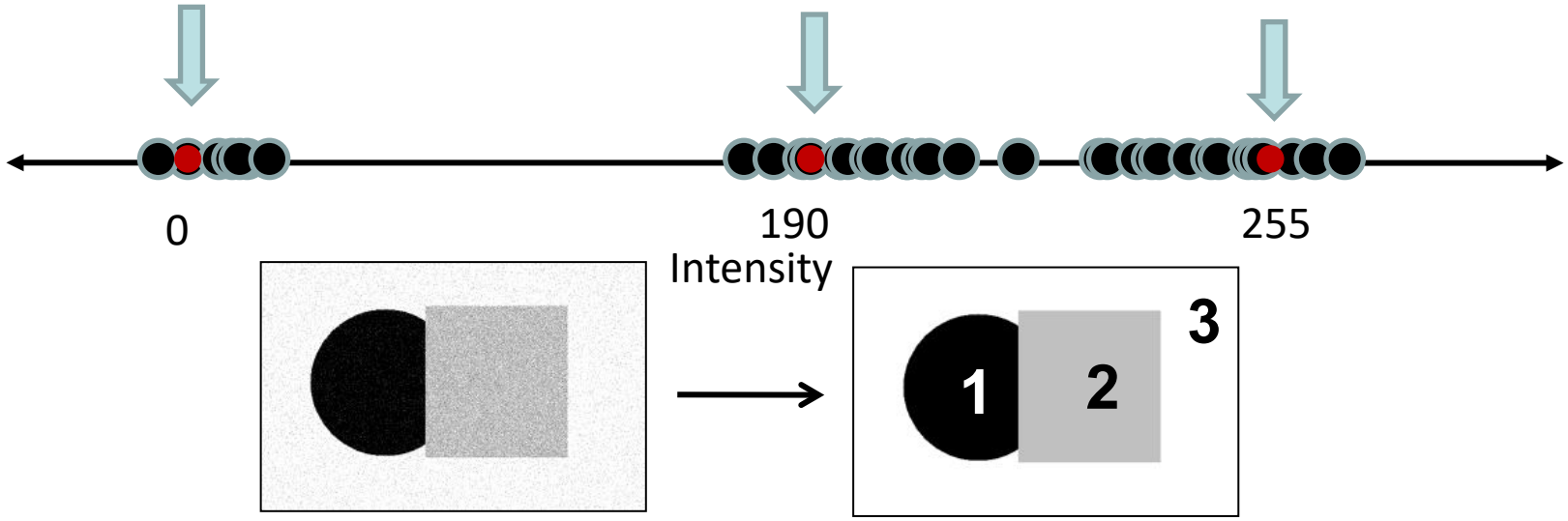
Input image



- Now how to determine the three main intensities that define our groups?
- We need to ***cluster***.

Slide credit: Kristen Grauman

Clustering



- **Goal:** choose three "***centers***" as the representative intensities, and label every pixel according to which of these centers it is nearest to.
- **Best cluster centers** are those that ***minimize*** Sum of Square Distance (SSD) between all points and their nearest cluster center c_i:

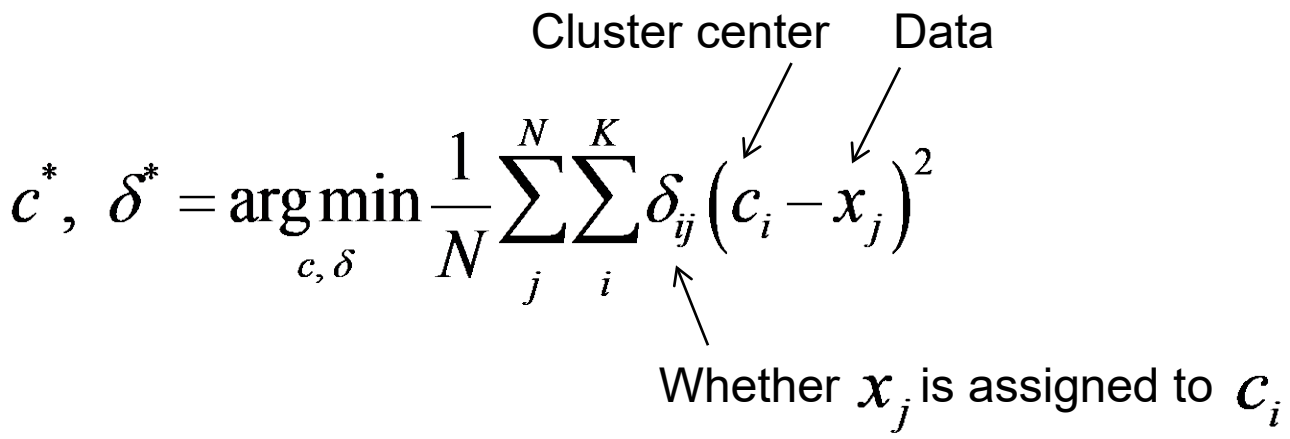
$$SSD = \sum_{cluster\, i} \sum_{x \in cluster\, i} (x - c_i)^2$$

Slide credit: Kristen Grauman

Goal for Clustering

Goal: cluster to minimize variance in data given clusters

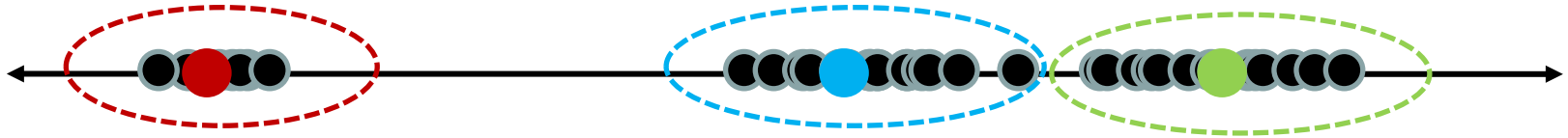
- Preserve information



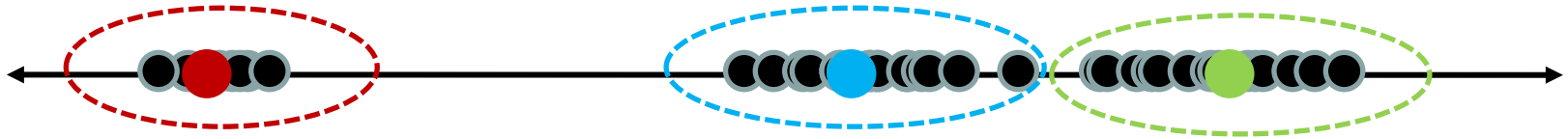
Slide: Derek Hoiem

Clustering

- With this objective, it is a “chicken and egg” problem:
  - If we knew the *cluster centers*, we could allocate points to groups by assigning each to its closest center.



  - If we knew the *group memberships*, we could get the centers by computing the mean per group.



Slide credit: Kristen Grauman

Content

- Segmentation
  - What is segmentation
  - Human grouping
- Region-based Segmentation
  - Histogram based
- Clustering
  - Watershed
  - K-means
  - Mean shift

K-means clustering

1. Initialize ($t = 0$): cluster centers $c_1, \ldots, c_K$

2. Compute δ^t : assign each point to the closest center
   - δ^t denotes the set of assignment for each x_j to cluster c_i at iteration *t*

$$\delta^t = \underset{\delta}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^{t-1} \left(c_i^{t-1} - x_j \right)^2$$

3. Computer c^t: update cluster centers as the mean of the points

$$c^t = \underset{c}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^t \left(c_i^{t-1} - x_j \right)^2$$

4. Update $t = t + 1$, Repeat Step 2-3 till stopped

Slide: Derek Hoiem

K-means clustering

1. Initialize ($t = 0$): cluster centers $c_1, \ldots, c_K$
   - Commonly used: random initialization
2. Compute δ^t : assign each point to the closest center
   - Typical distance measure:
     - Euclidean
     - Cosine
     - Others
3. Computer c^t: update cluster centers as the mean of the points

$$c^t = \underset{c}{\operatorname{argmin}} \frac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \delta_{ij}^t \left(c_i^{t-1} - x_j \right)^2$$

4. Update $t = t + 1$, Repeat Step 2-3 till stopped
   - c^t doesn't change anymore.

Slide: Derek Hoiem

Common Similarity/distance Measures

- P-norms
  - City Block (L1)
  - Euclidean (L2)
  - L-infinity

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$$

$$\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}$$

$$\|\mathbf{x}\|_\infty := \max\left(|x_1|, \ldots, |x_n|\right)$$

Here x_i is the distance btw. two points

- Mahalanobis
  - Scaled Euclidean

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}},$$

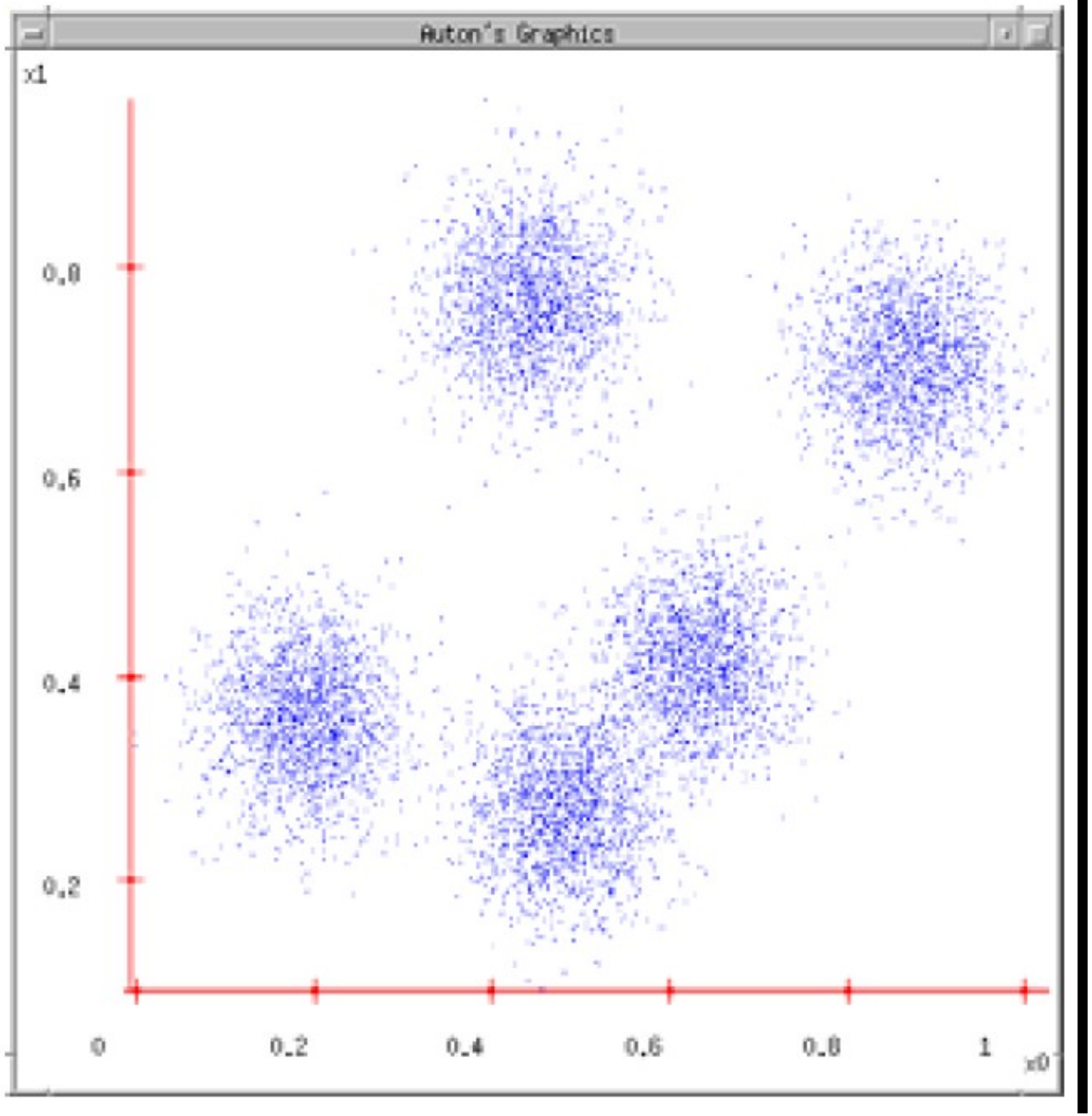
- Cosine distance

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Slide credit: D. Hoiem

K-means

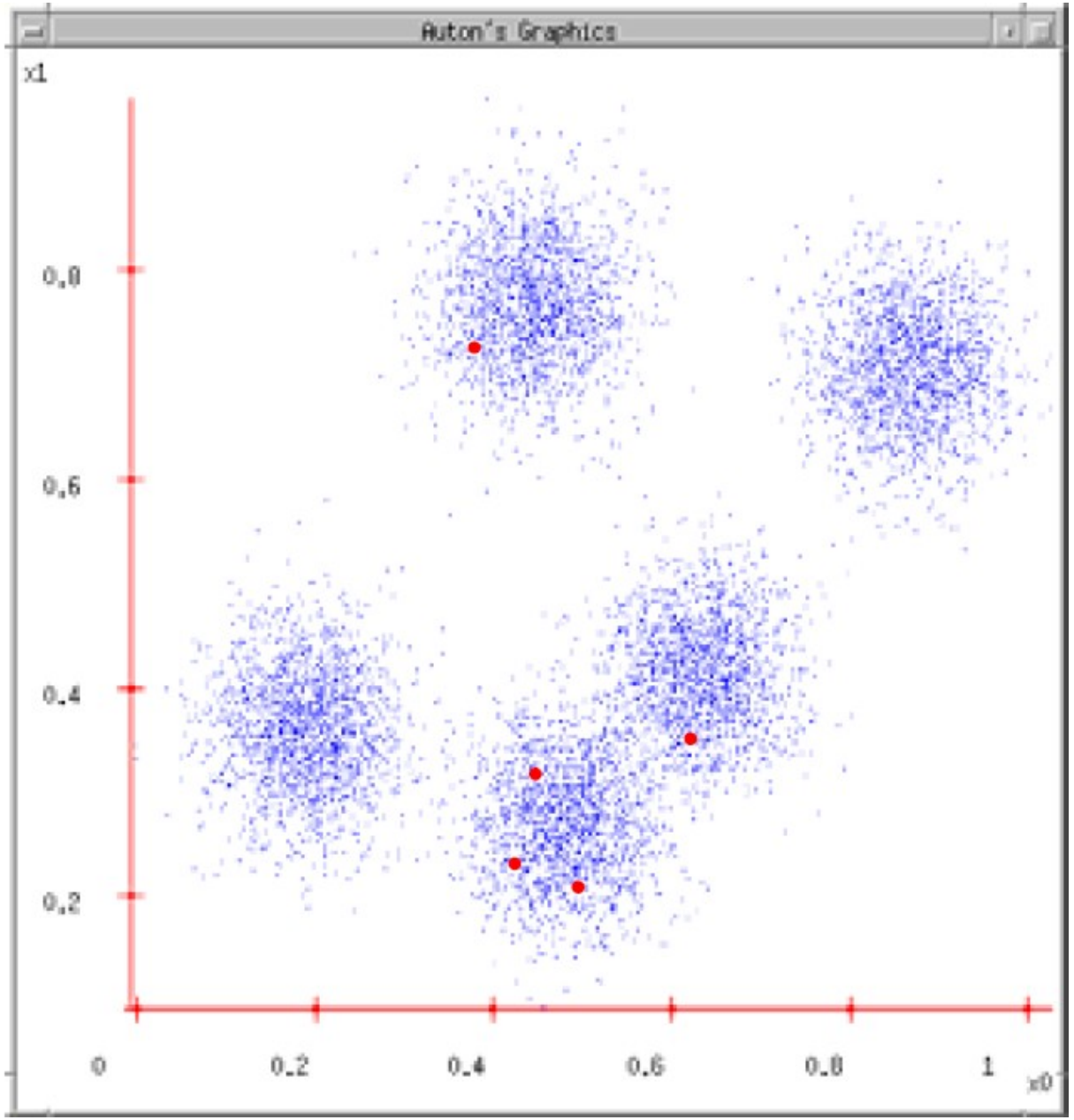
1. Ask user how many clusters they'd like. *(e.g. k=5)*



Slide credit: K Grauman, A. Moore

K-means

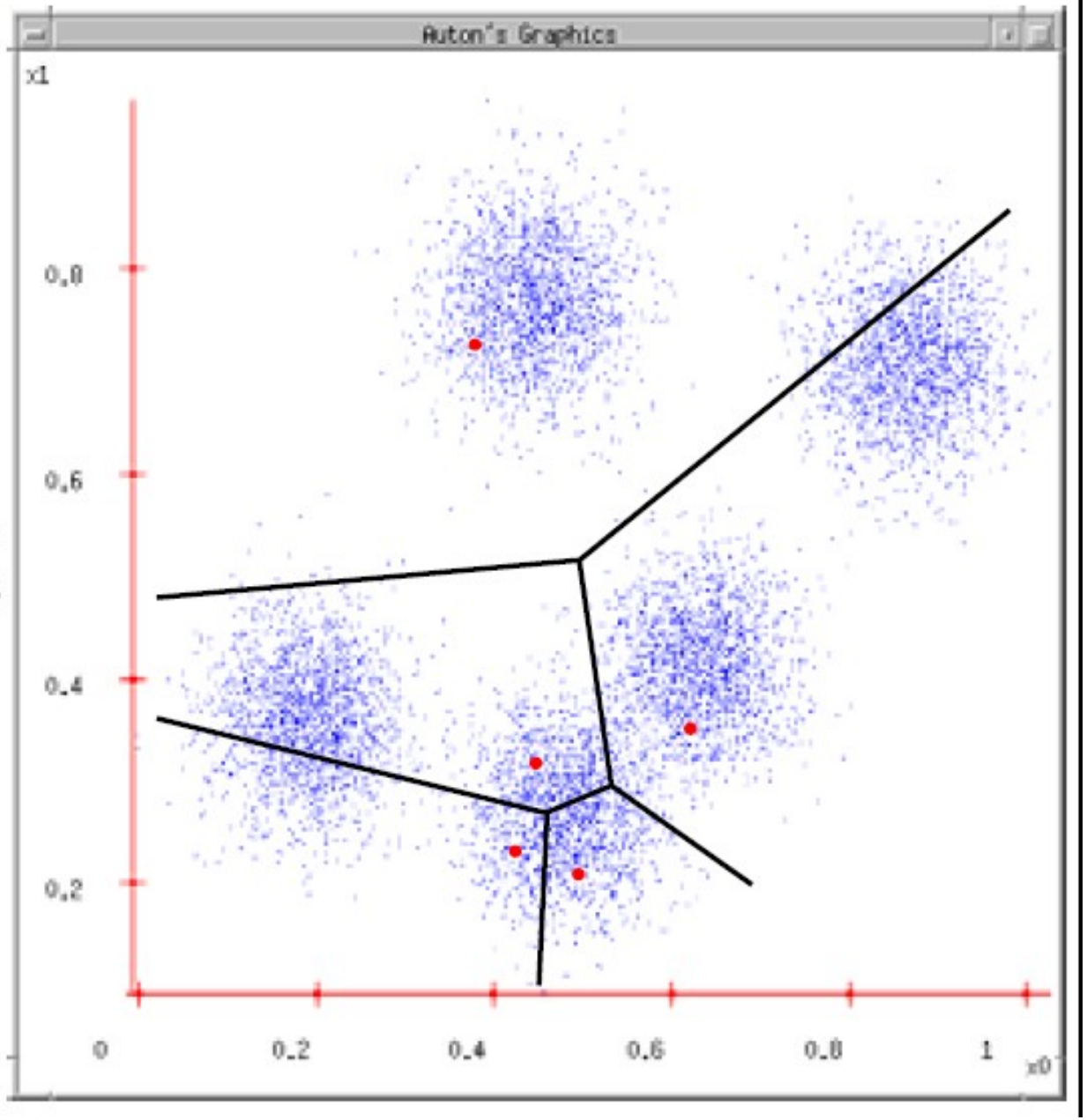
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



Slide credit: K Grauman, A. Moore

K-means

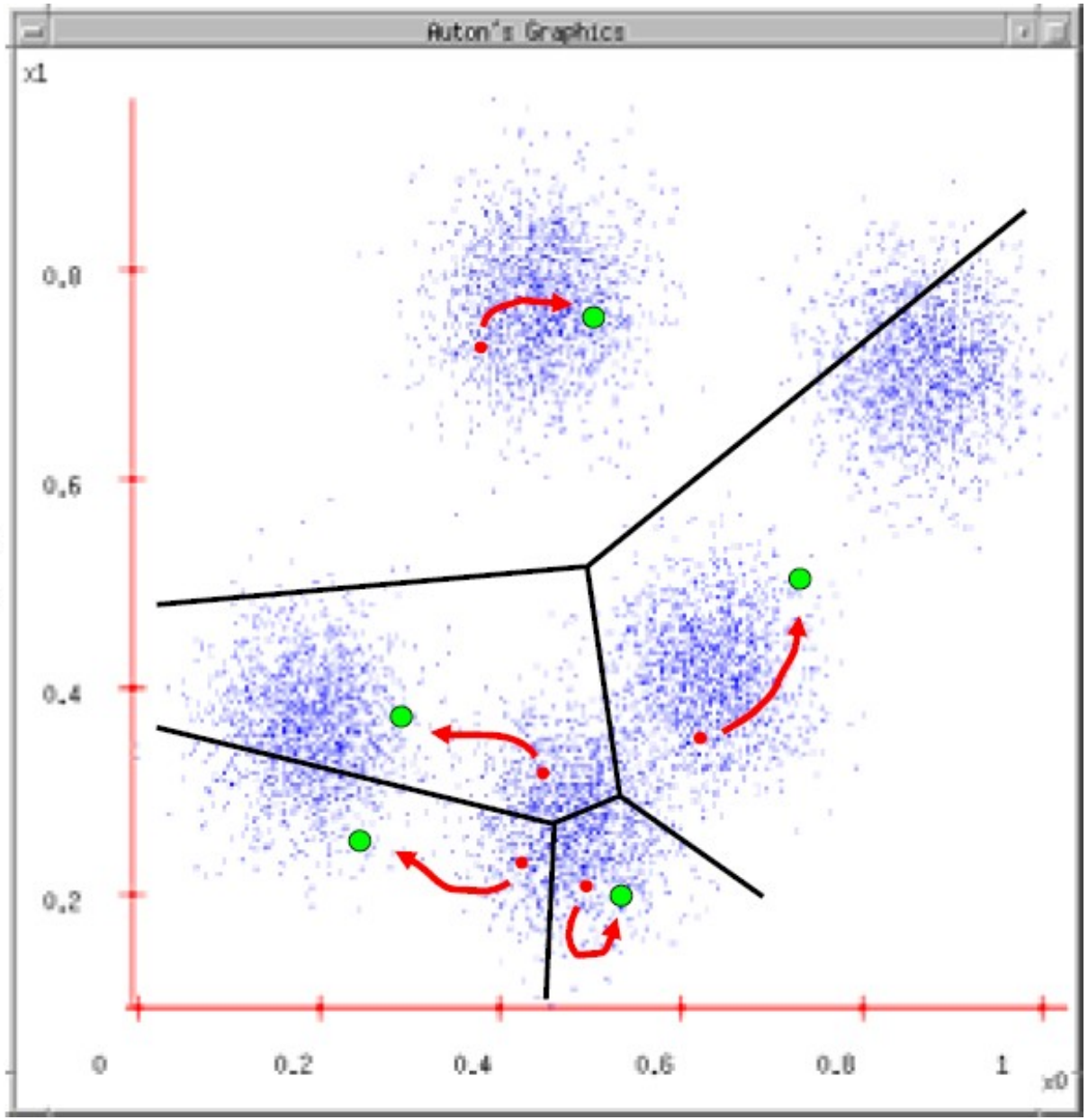
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



Slide credit: K Grauman, A. Moore

K-means

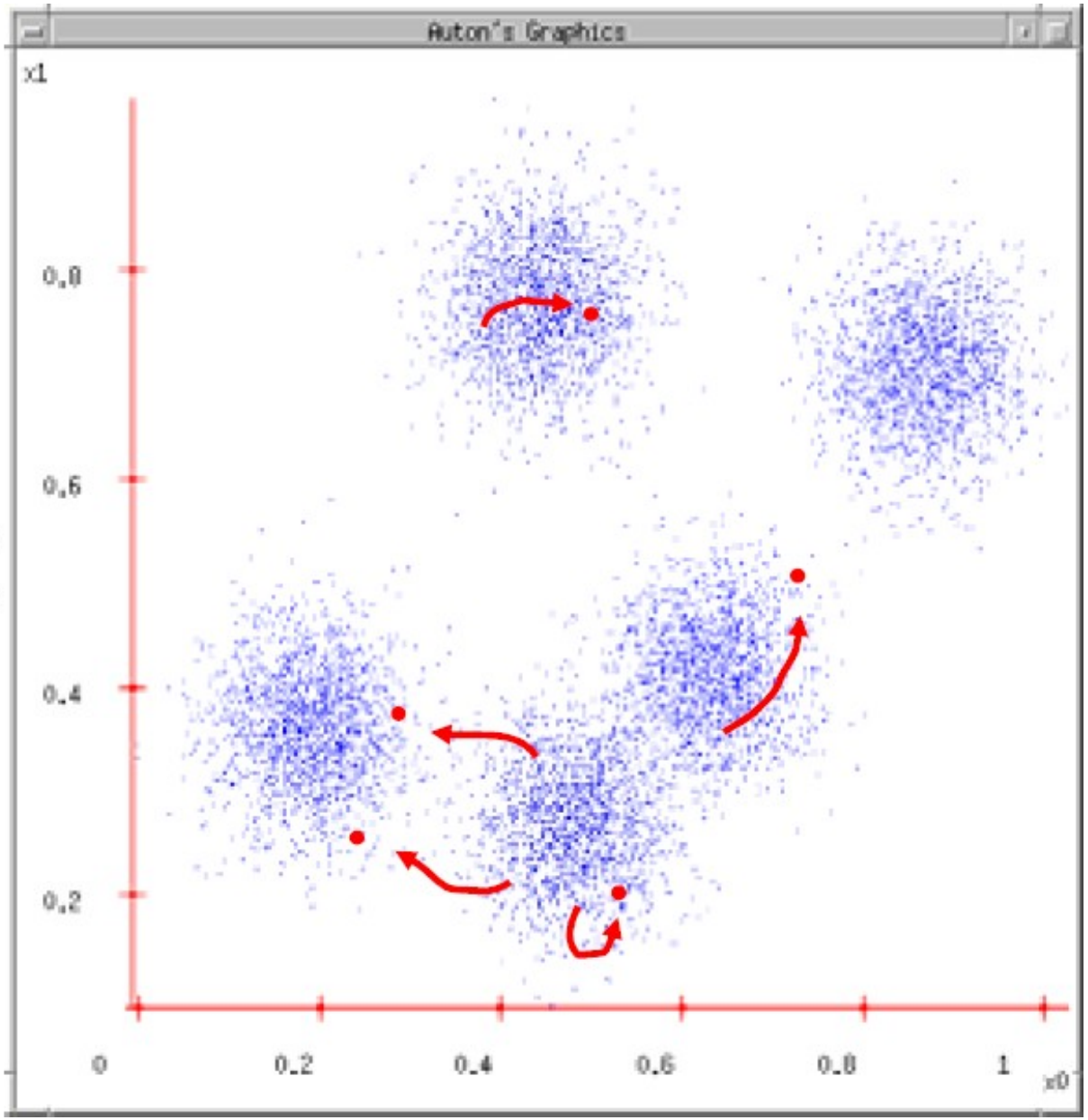
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



Slide credit: K Grauman, A. Moore

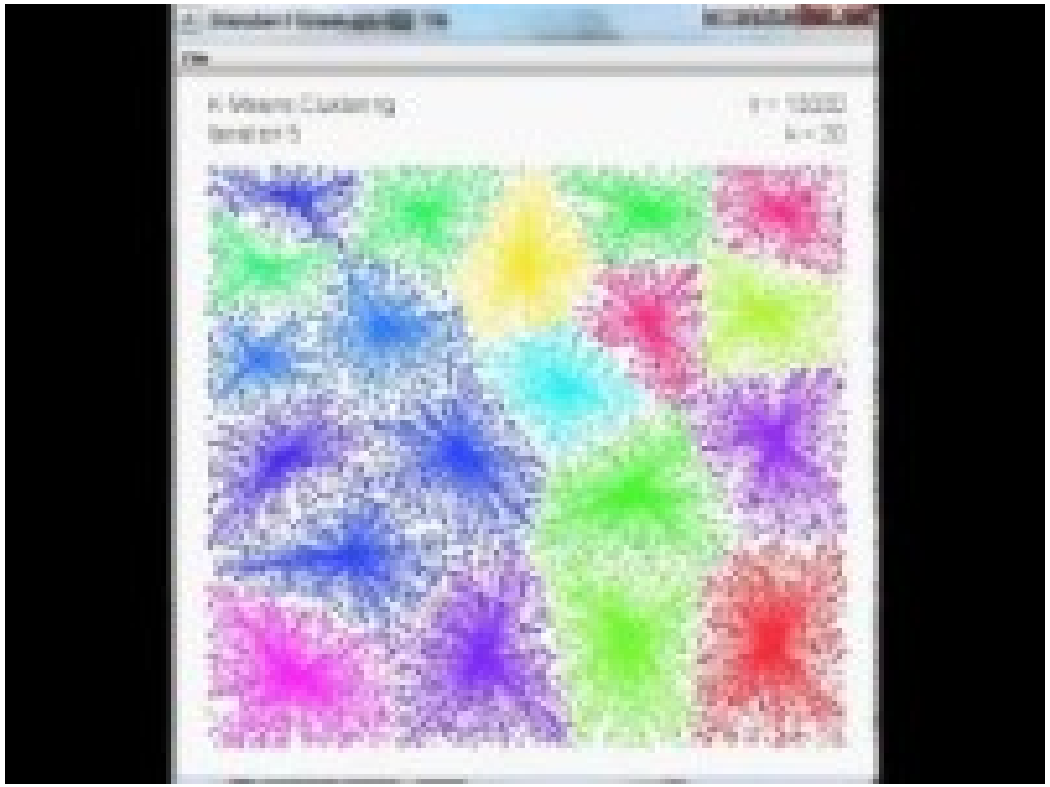
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



Slide credit: K Grauman, A. Moore

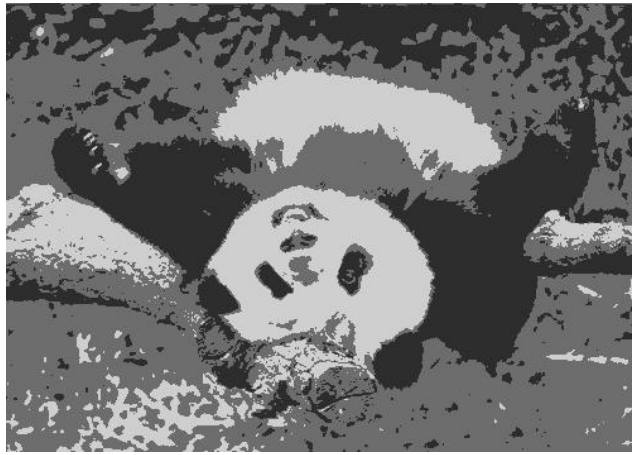
Online Example



Segmentation as Clustering

Original image

2 clusters



3 clusters

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different

- Grouping pixels based on intensity similarity

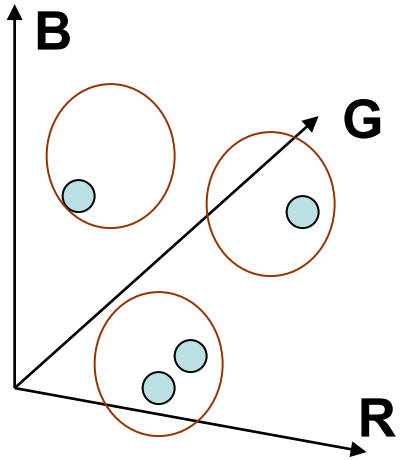
- Feature space: intensity value (1D)

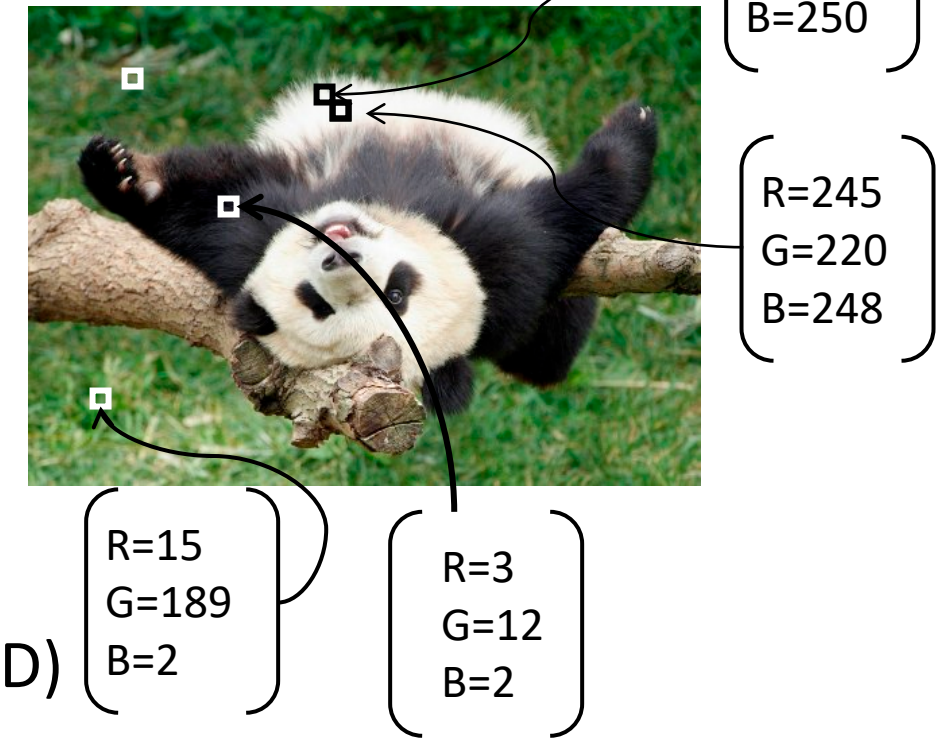
Slide credit: Kristen Grauman

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on color similarity





- Feature space: color value (3D)

$$dist(p_i,\ c_j) = \sqrt{\left(R_{p_i} - R_{c_j}\right)^2 + \left(G_{p_i} - G_{c_j}\right)^2 + \left(B_{p_i} - B_{c_j}\right)^2}$$

Slide credit: Kristen Grauman

K-Means Clustering Results

- K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  - Clusters don't have to be spatially coherent

Image

Intensity-based clusters

Color-based clusters

Image source: Forsyth & Ponce

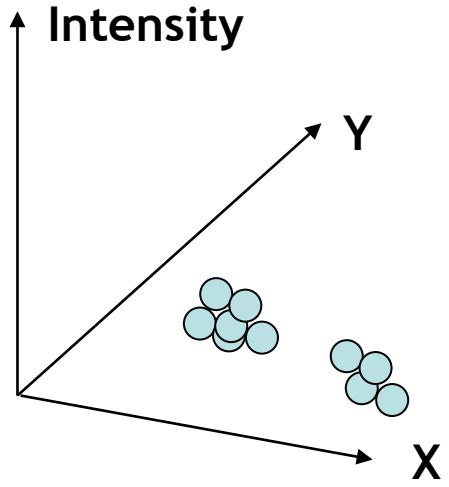
Color Segmentation

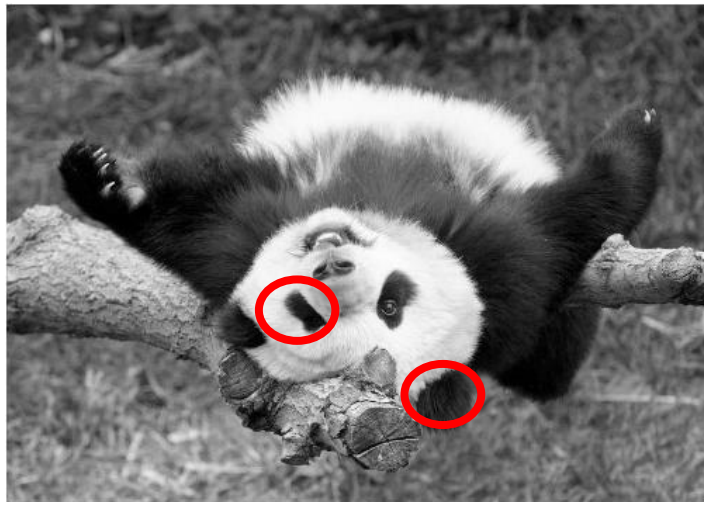
- ***Color alone often will not yield salient segments!***

Segmentation as Clustering

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on *intensity+position* similarity





⇒ Way to encode both *similarity* and *proximity.*

$$dist(p_i,\ c_j) = \sqrt{\left(i_{p_i} - i_{c_j}\right)^2 + \left(x_{p_i} - x_{c_j}\right)^2 + \left(y_{p_i} - y_{c_j}\right)^2}$$

Remember to normalize your features

Slide credit: Kristen Grauman

K-Means Clustering Results

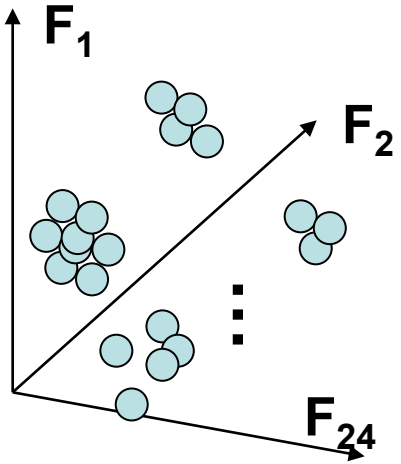
- K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  - Clusters don’t have to be spatially coherent
- Clustering based on (r,g,b,x,y) values enforces more spatial coherence

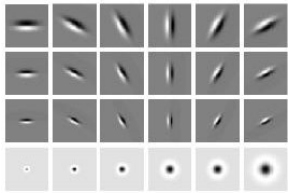
Image source: Forsyth & Ponce

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on texture similarity



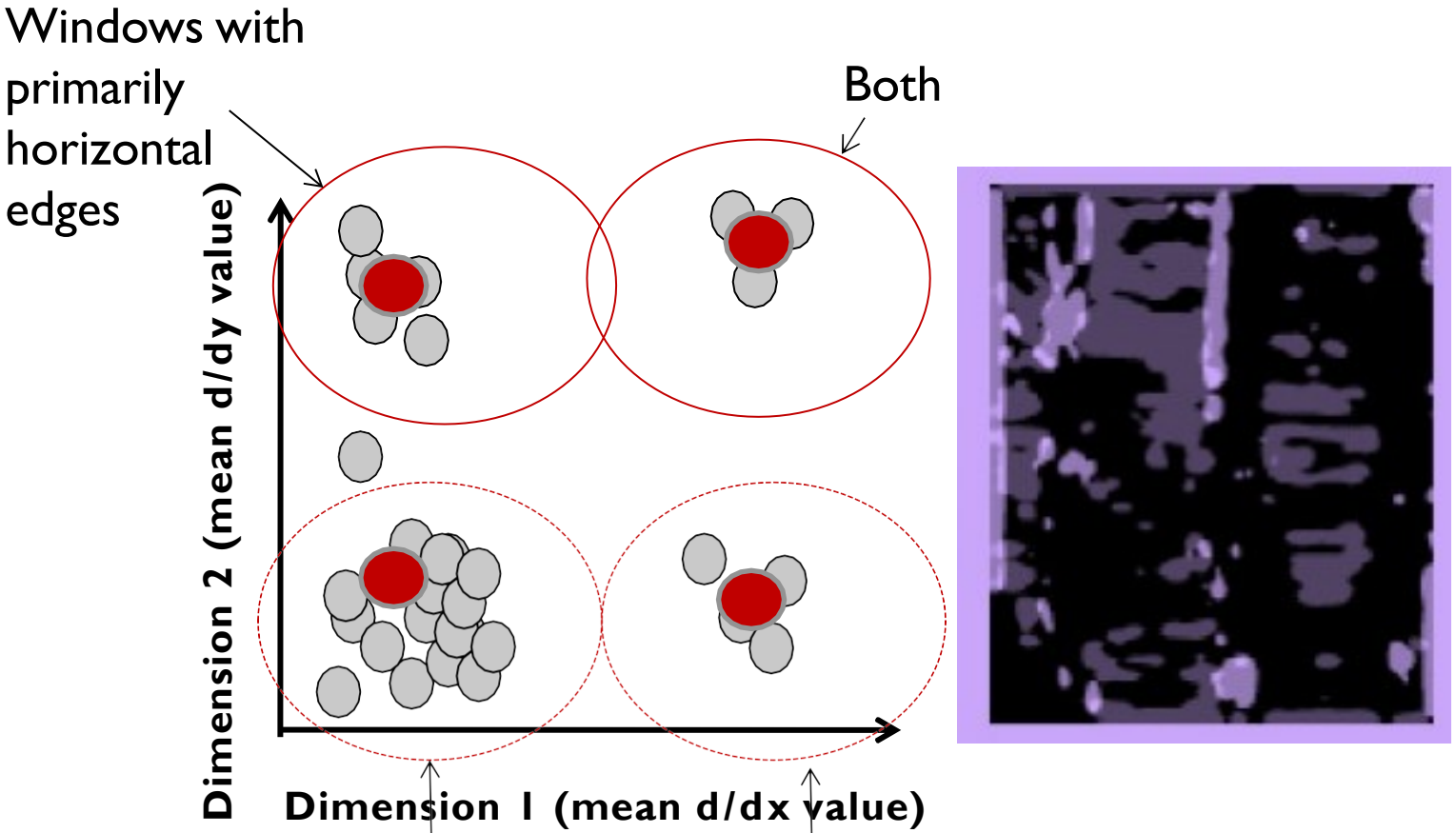


Filter bank of 24 filters

- Feature space: filter bank responses (e.g., 24D)

Slide credit: Kristen Grauman

Texture Representation Example



	mean d/dx value	mean d/dy value
Win.#1	4	10
Win.#2	18	7
...		
Win.#9	20	20

statistics to summarize patterns in small windows

Slide credit: K Grauman

How to evaluate clusters?

- Generative
  - How well are points reconstructed from the clusters?

- Discriminative
  - How well do the clusters correspond to labels?
    - Can we correctly classify which pixels belong to the panda?

  - Note: unsupervised clustering does not aim to be discriminative as we don't have the labels.

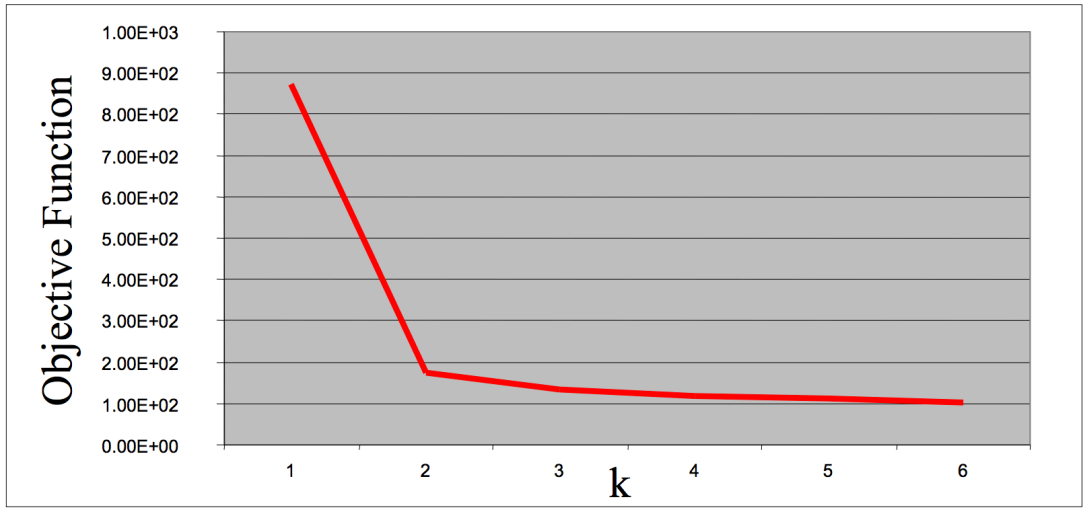
Slide: Derek Hoiem

How to choose the number of clusters?

Try different numbers of clusters in a validation set and look at performance.

We can plot the objective function values for k equals 1 to 6…

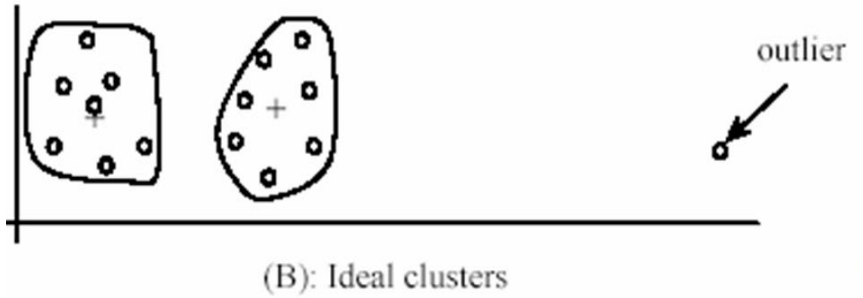
The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.



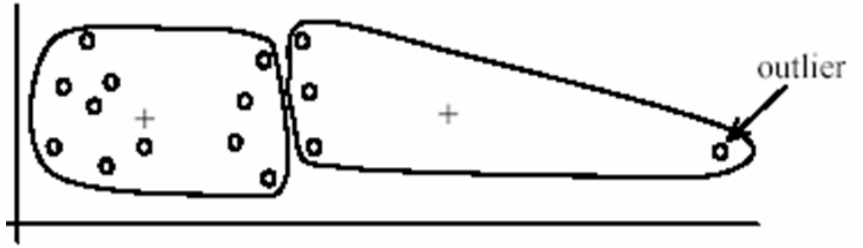
Slide: Derek Hoiem

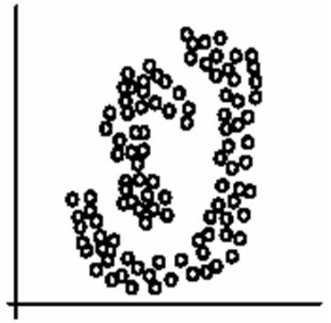
K-Means pros and cons

- **Pros**
  - Finds cluster centers that minimize conditional variance (good representation of data)
  - Simple and fast, Easy to implement
- **Cons**
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima
  - All clusters have the same parameters (e.g., distance measure is non-adaptive)
  - *Can be slow: each iteration is O(KNd) for N d-dimensional points
- **Usage**
  - Unsupervised clustering
  - Rarely used for pixel segmentation

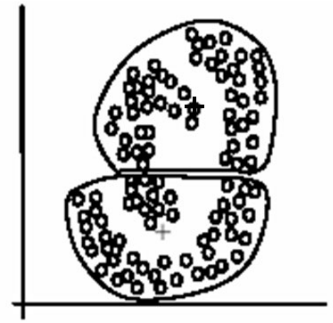


(B): Ideal clusters





(A): Two natural clusters



(B): *k*-means clusters

Pseudo Code for k-means

Algorithm ***k*-means**

Input: RGB Images with $m \times n$ pixels. The user defined number of clusters, k

Max iteration number it_{max}

Output: A set of k labels, $\iota_1, \iota_2 \ldots \iota_k$

1. Initialise k cluster centre $c_1, c_2 \ldots c_k$
2. **do**
3. for each pixel p_i, **do**
4. $\iota_i = 0$
5. $dist_{min} = \infty$
6. for each cluster centre, c_j **do**
7. $dist_j$= distance of p_i to c_j
8. **if** $dist_j < dist_{min}$ **then**
9. $\iota_i = j$
10. $dist_{min} = dist_j$
11. **end if**
12. **end for**
13. **end for**
14. **for** each cluster centre, c_k **do**
15. $c_k = mean(p_i$ which label $\iota_i = k)$
16. **end for**
17. **end do** while $n < it_{max}$
18. change each pixel p_i to it's label c_k where $k = \iota_i$ for display

Content

- Segmentation
  - What is segmentation
  - Human grouping
- Region-based Segmentation
  - Histogram based
- Edge-based Segmentation
- Clustering
  - K-means
  - Mean shift

Mean Shift Clustering

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002 603

Mean Shift: A Robust Approach Toward Feature Space Analysis

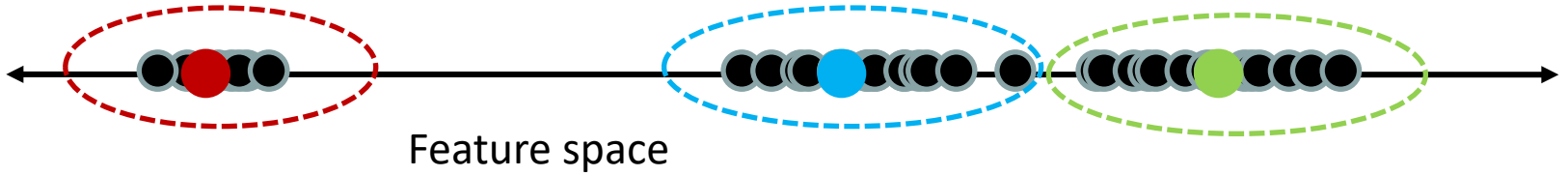
Dorin Comaniciu, *Member*, *IEEE*, and Peter Meer, *Senior Member*, *IEEE*

Abstract—A general nonparametric technique is proposed for the analysis of a complex multimodal feature space and to delineate arbitrarily shaped clusters in it. The basic computational module of the technique is an old pattern recognition procedure, the mean shift. We prove for discrete data the convergence of a recursive mean shift procedure to the nearest stationary point of the underlying density function and, thus, its utility in detecting the modes of the density. The relation of the mean shift procedure to the Nadaraya-Watson estimator from kernel regression and the robust M-estimators of location is also established. Algorithms for two low-level vision tasks, discontinuity preserving smoothing and image segmentation, are described as applications. In these algorithms, the only user set parameter is the resolution of the analysis and either gray level or color images are accepted as input. Extensive experimental results illustrate their excellent performance.

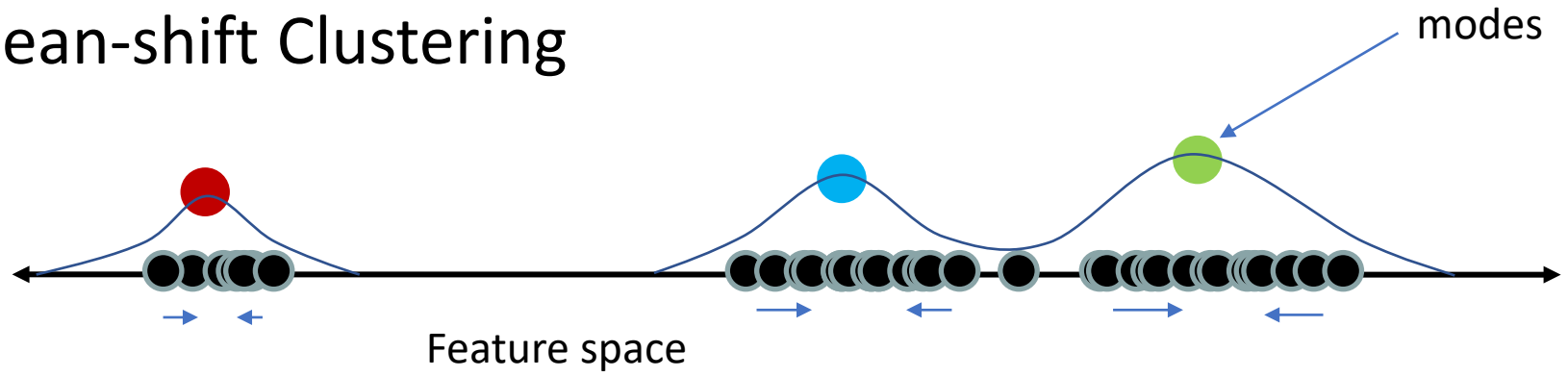
Index Terms—Mean shift, clustering, image segmentation, image smoothing, feature space, low-level vision.

Intuitive Idea

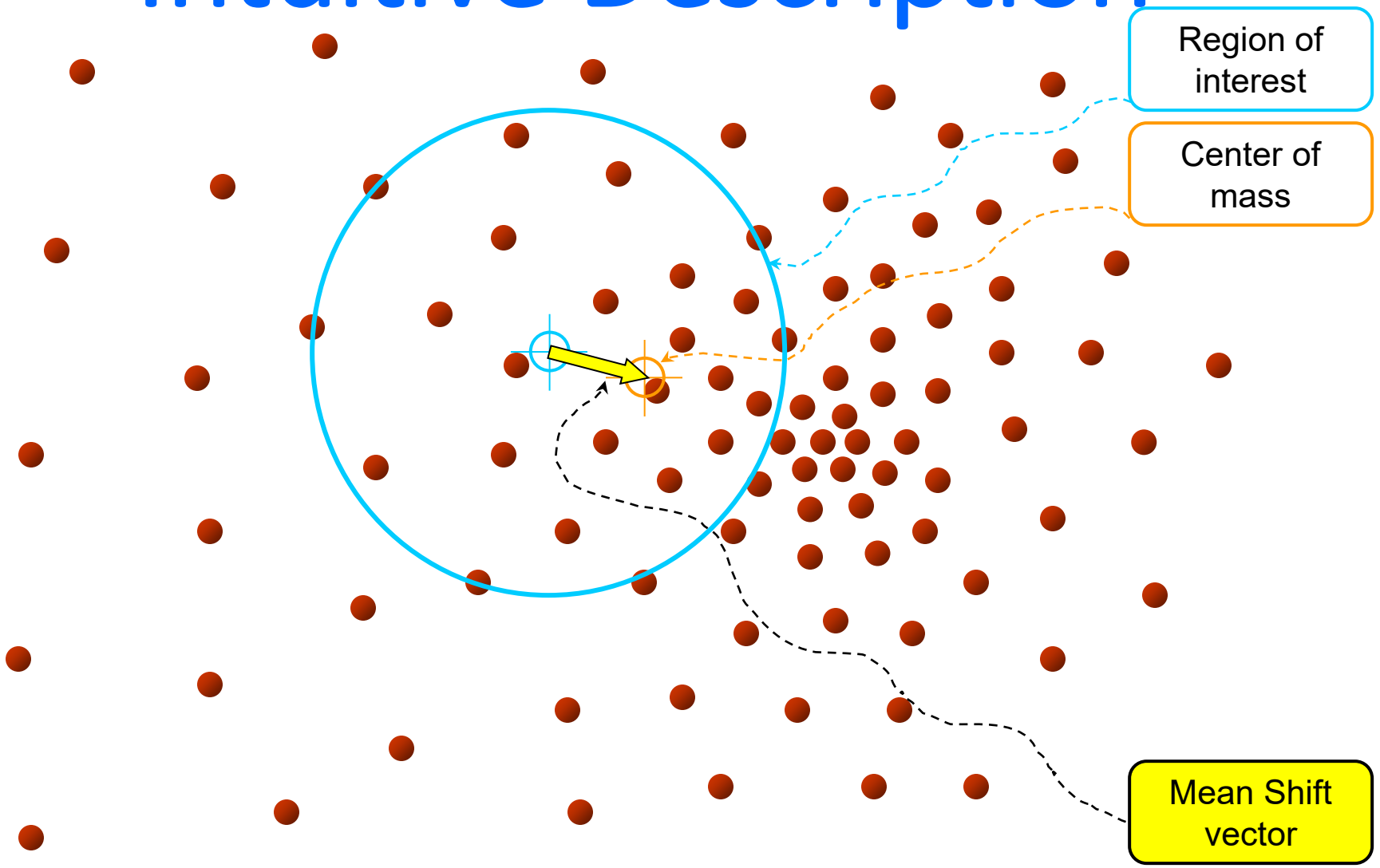
- K-means Clustering randomly select the cluster centres and calculate the ‘distance’ of each point in the feature space. Shift the cluster centres.



- Mean-shift Clustering



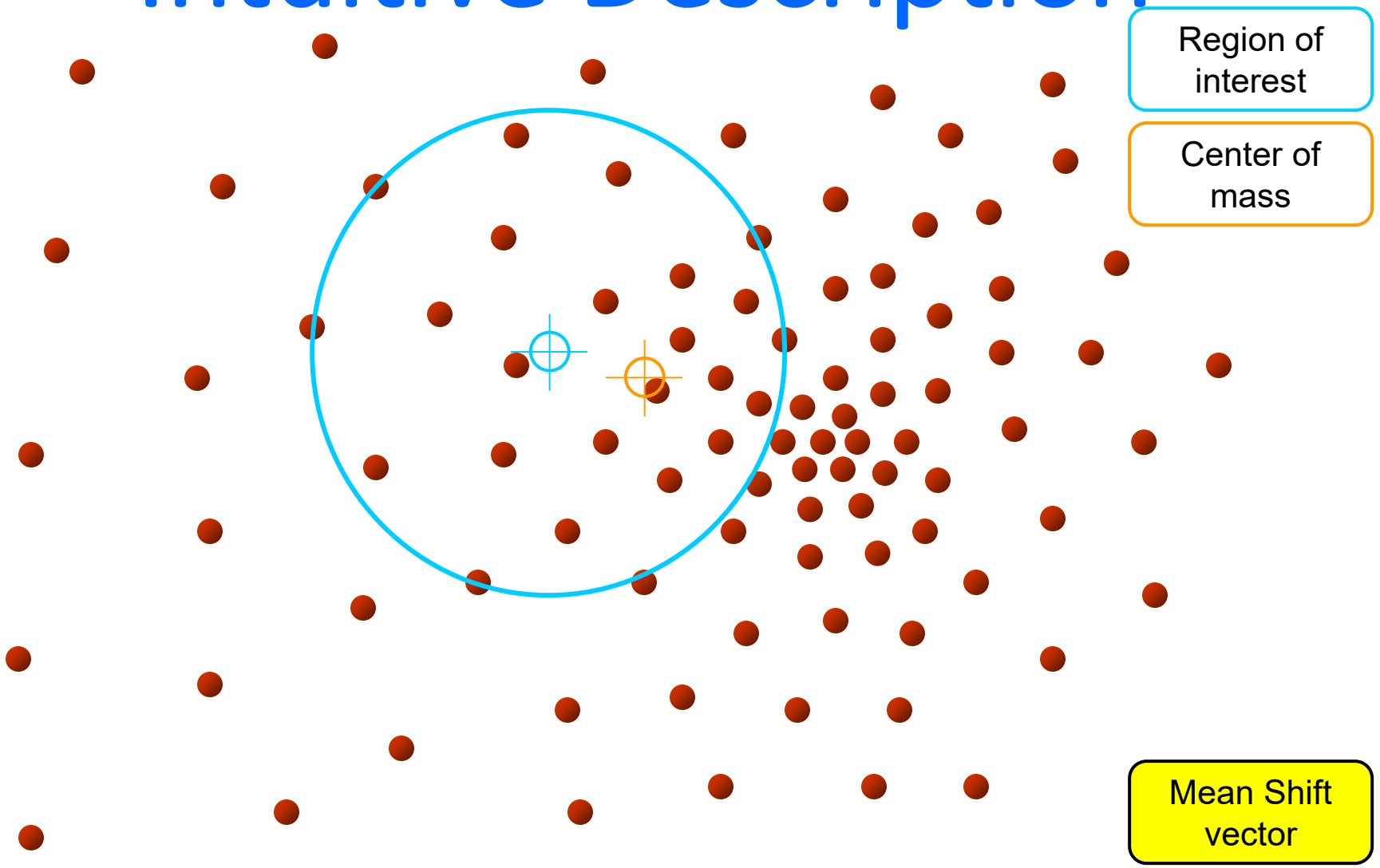
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

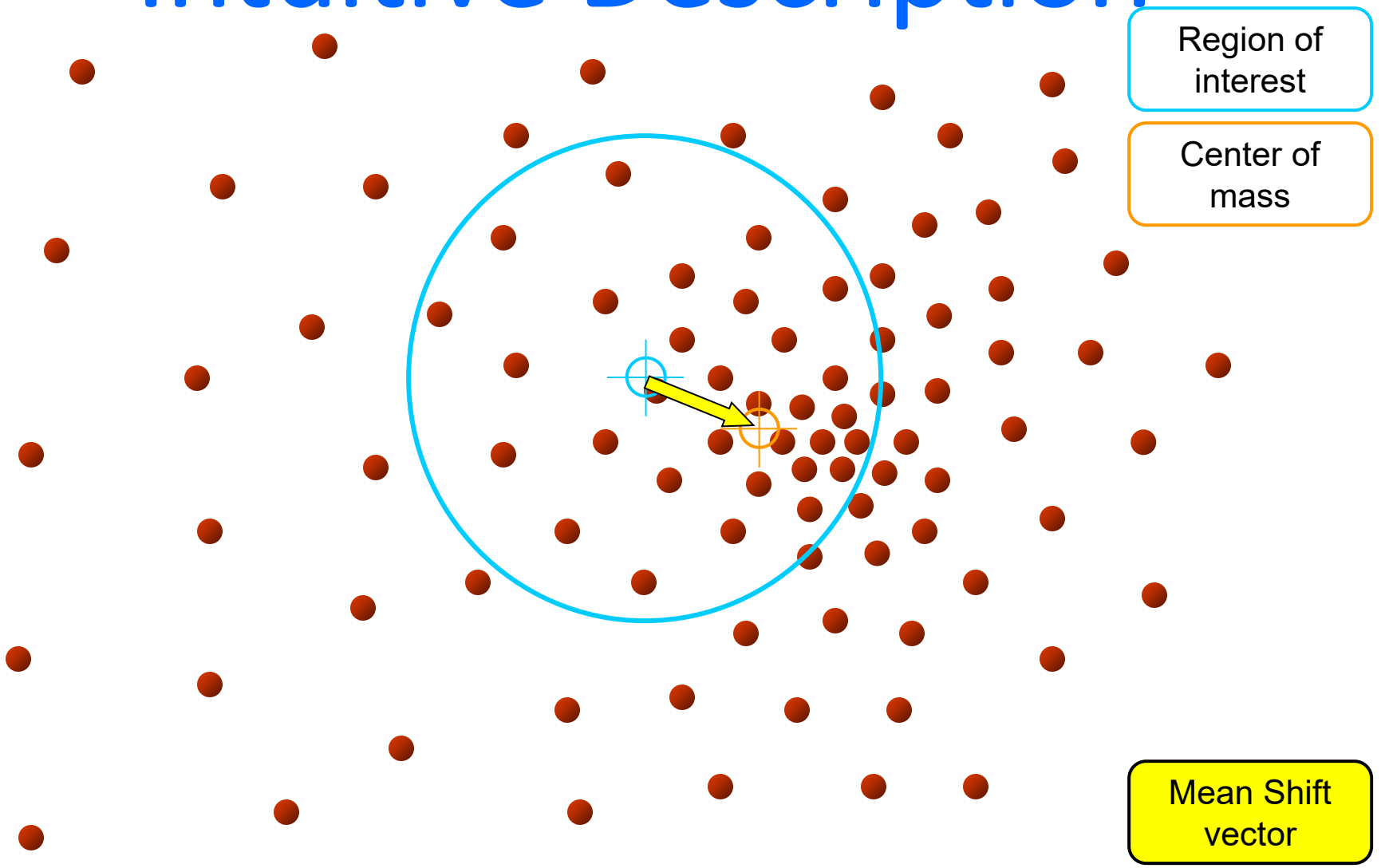
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

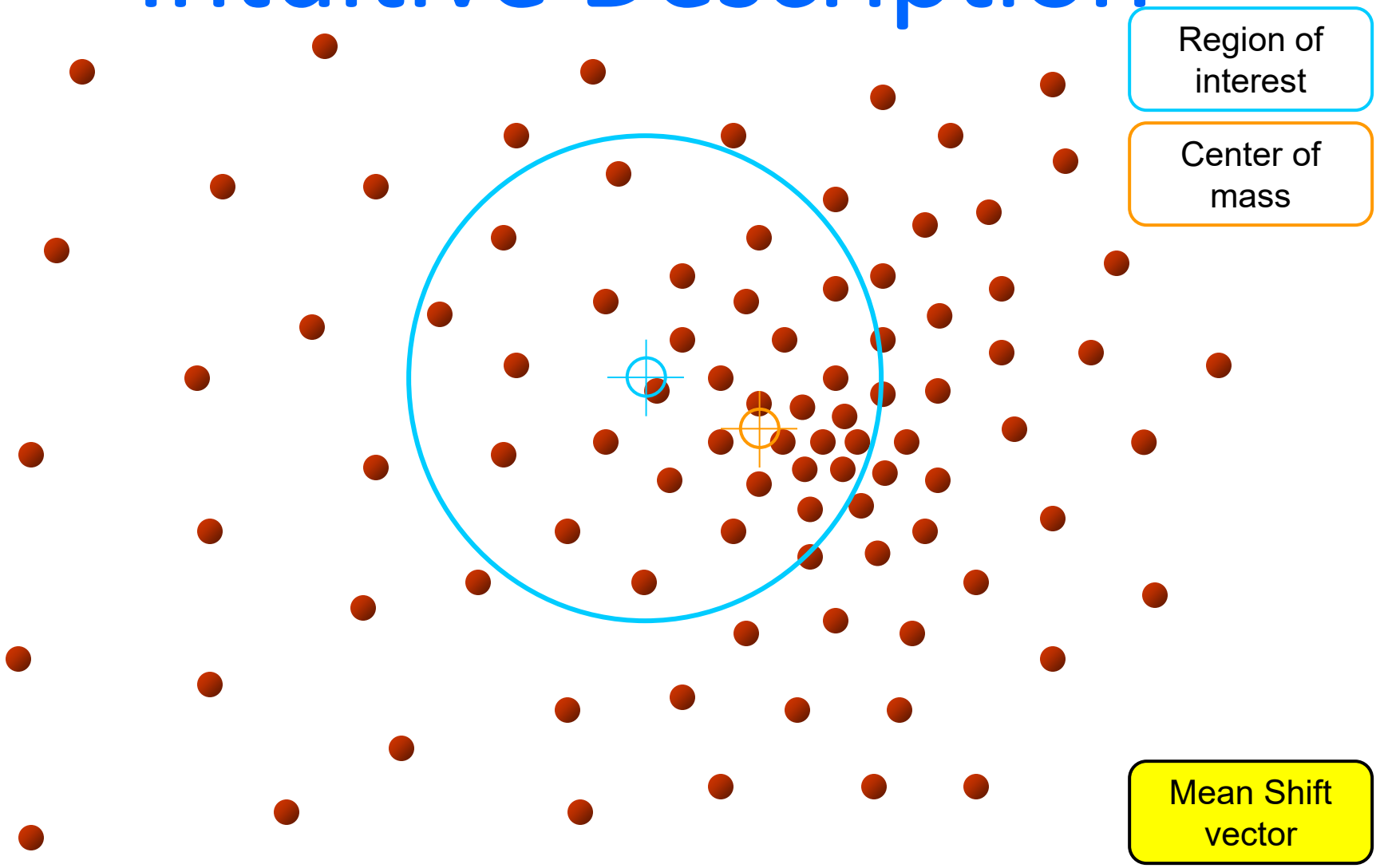
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

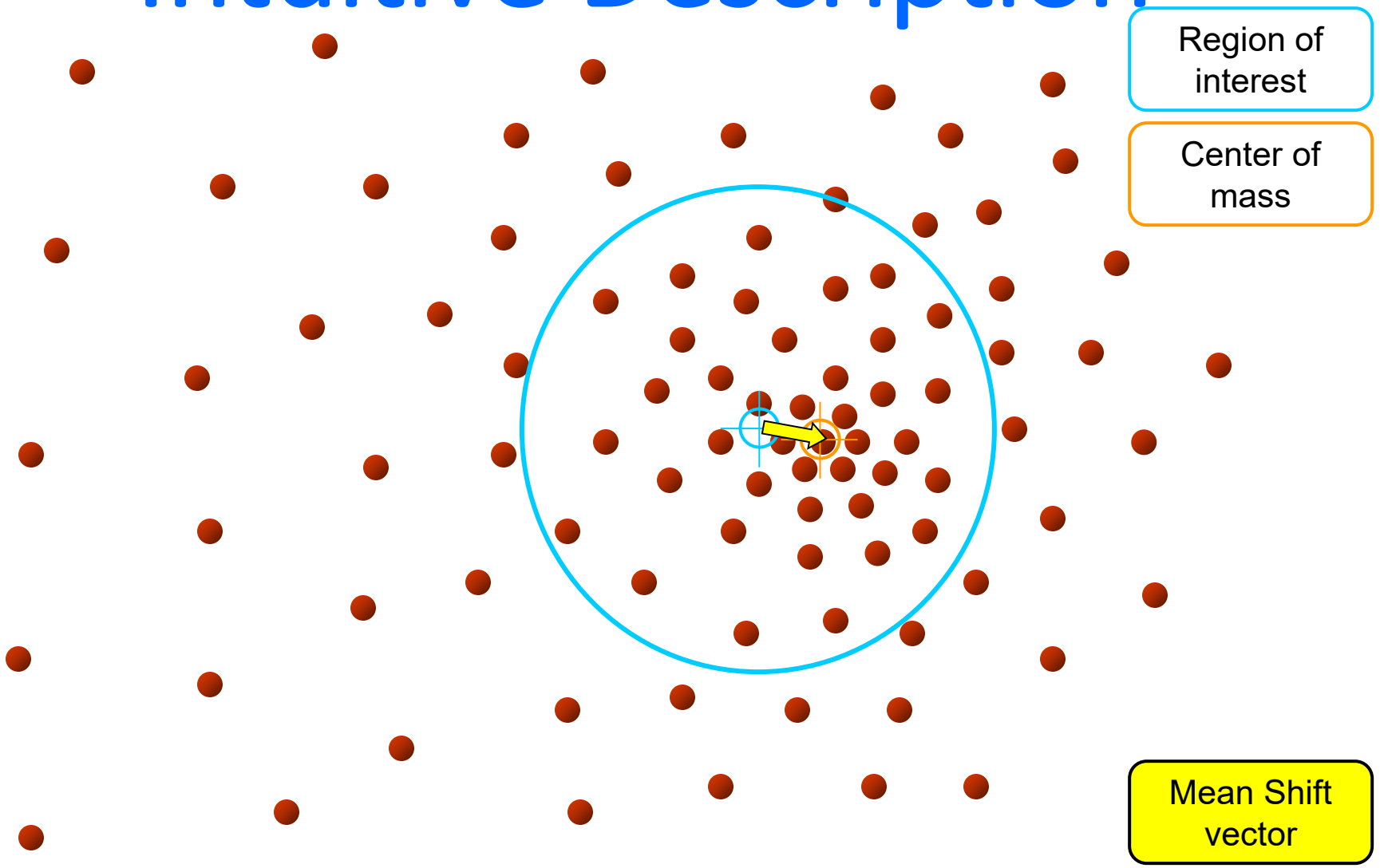
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

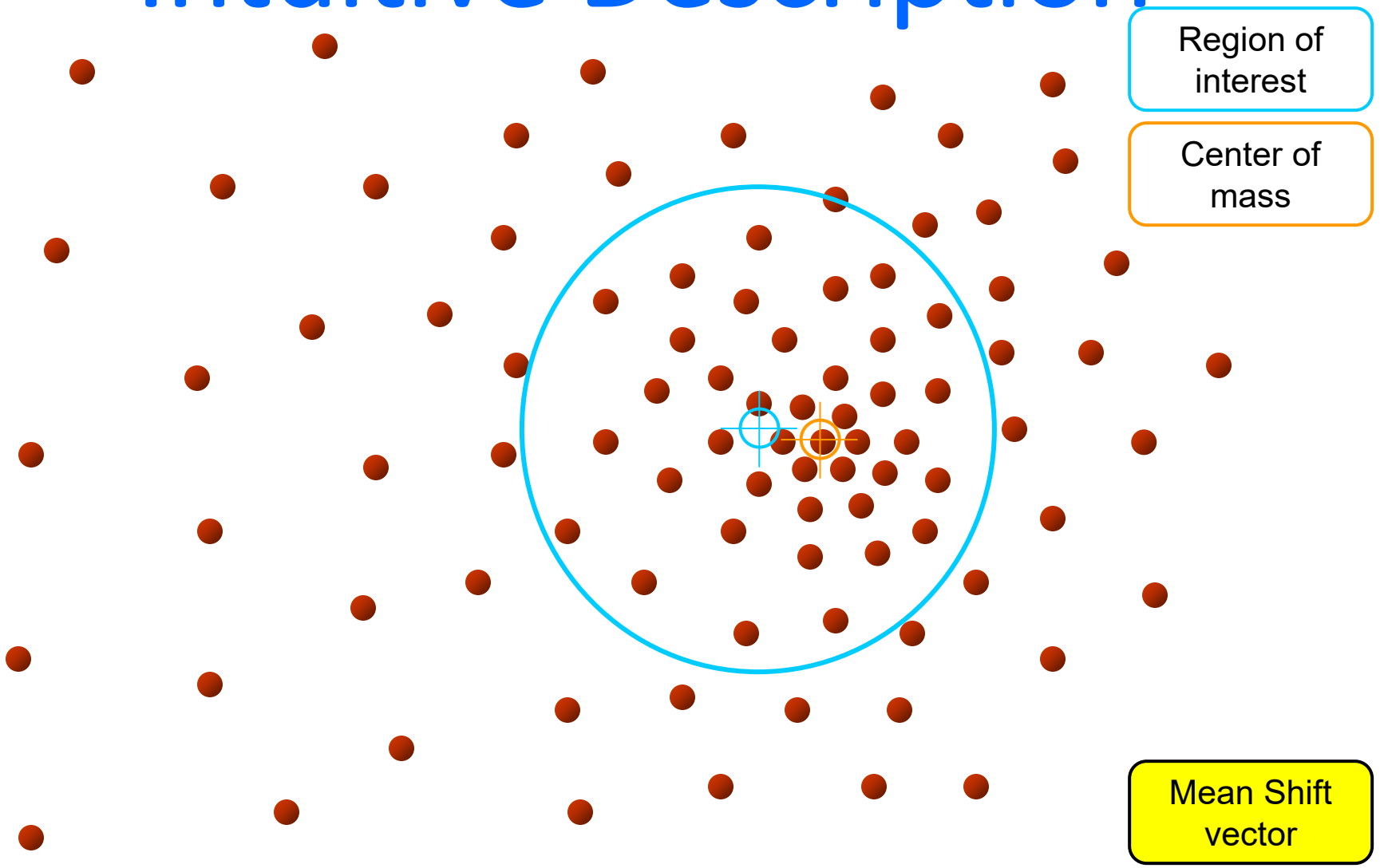
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

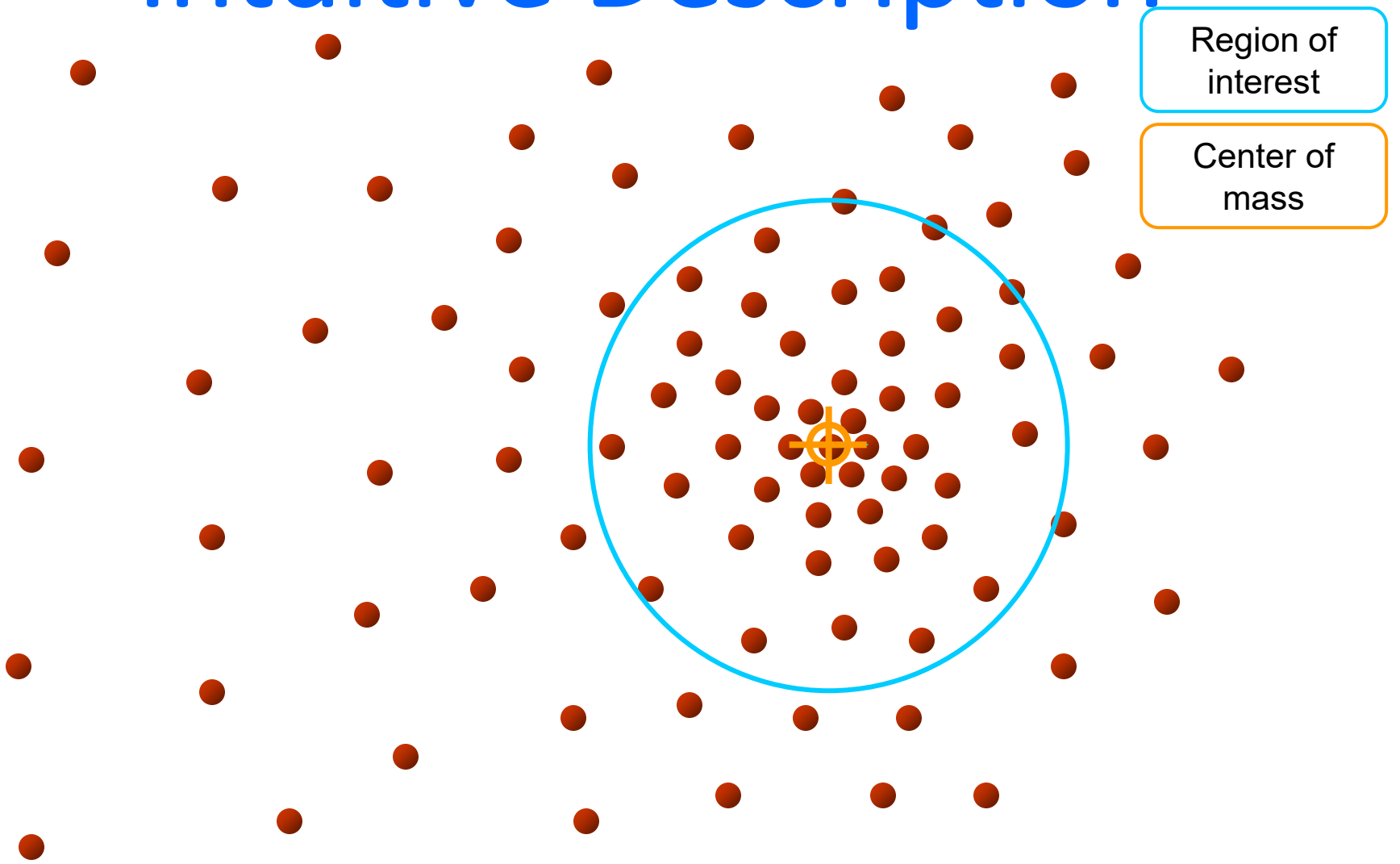
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

Intuitive Description



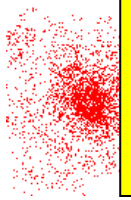
Objective : Find the densest region

Distribution of identical billiard balls

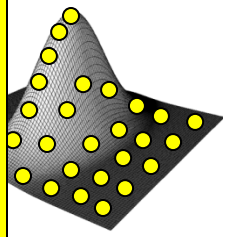
What is Mean Shift ?

A tool for:
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in R^N

PDF in feature space
- **Color space**
- **Scale space**
- **Actually any feature space you can conceive**
- **…**

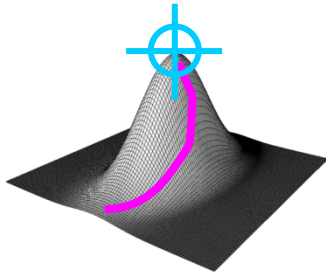


Data



DF Representation

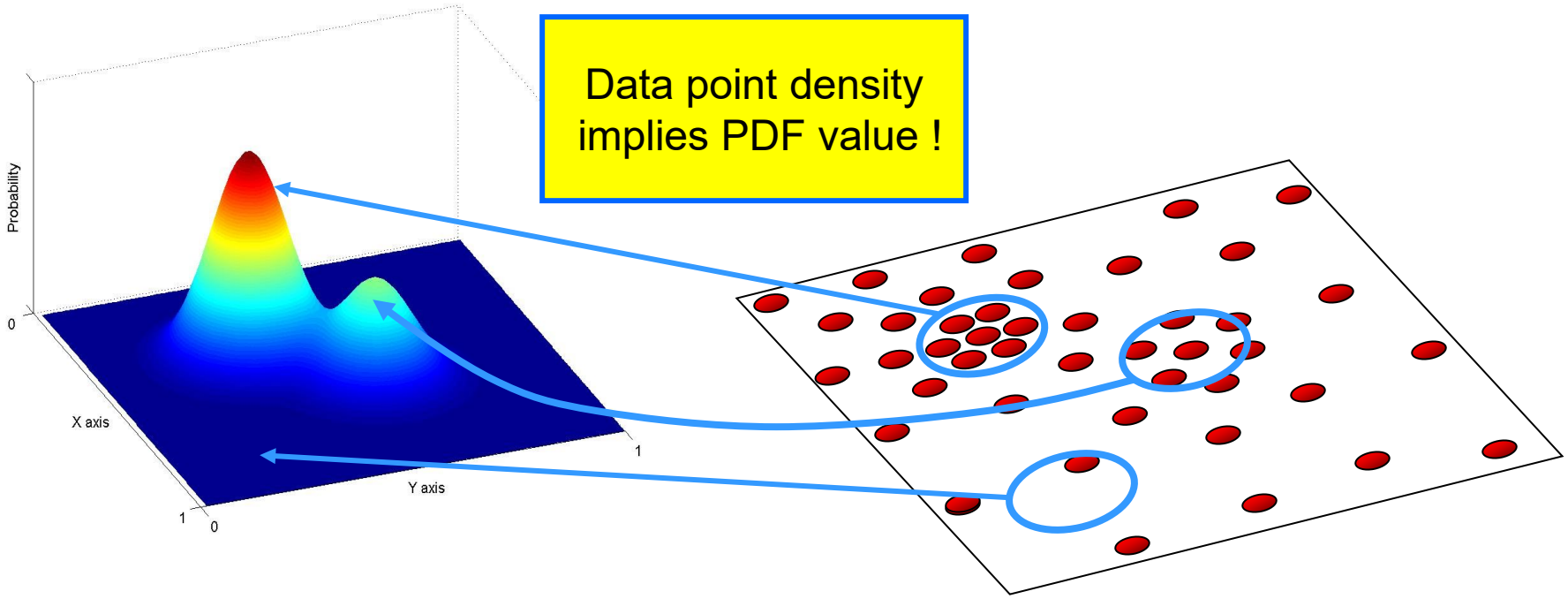
Non-parametric
Density **GRADIENT** Estimation
(Mean Shift)



PDF Analysis

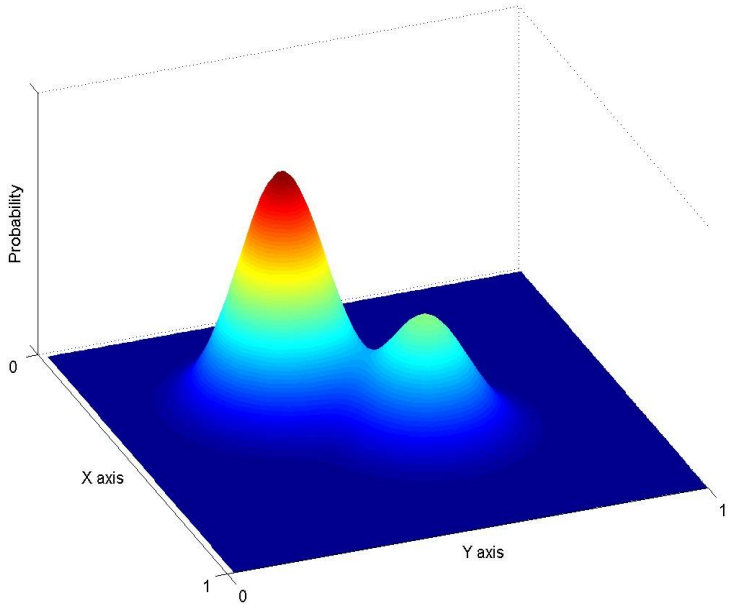
Non-Parametric Density Estimation

Assumption : The data points are sampled from an underlying PDF

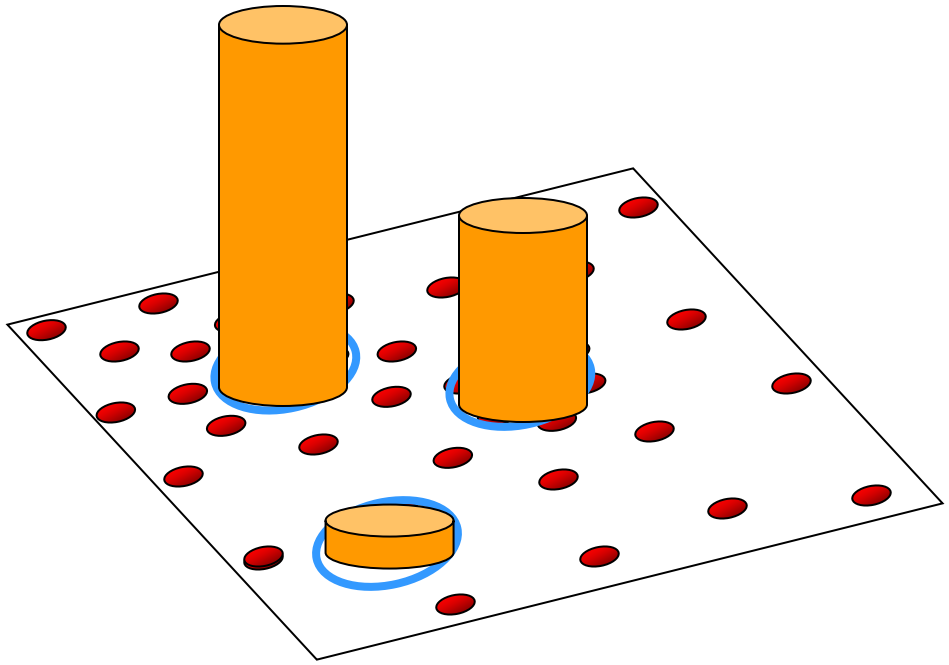


Assumed Underlying PDF **Real Data Samples**

Non-Parametric Density Estimation

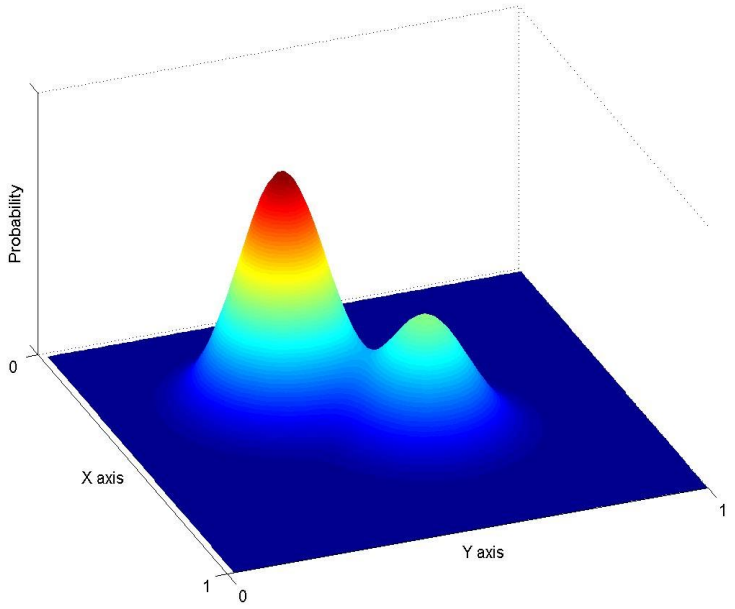


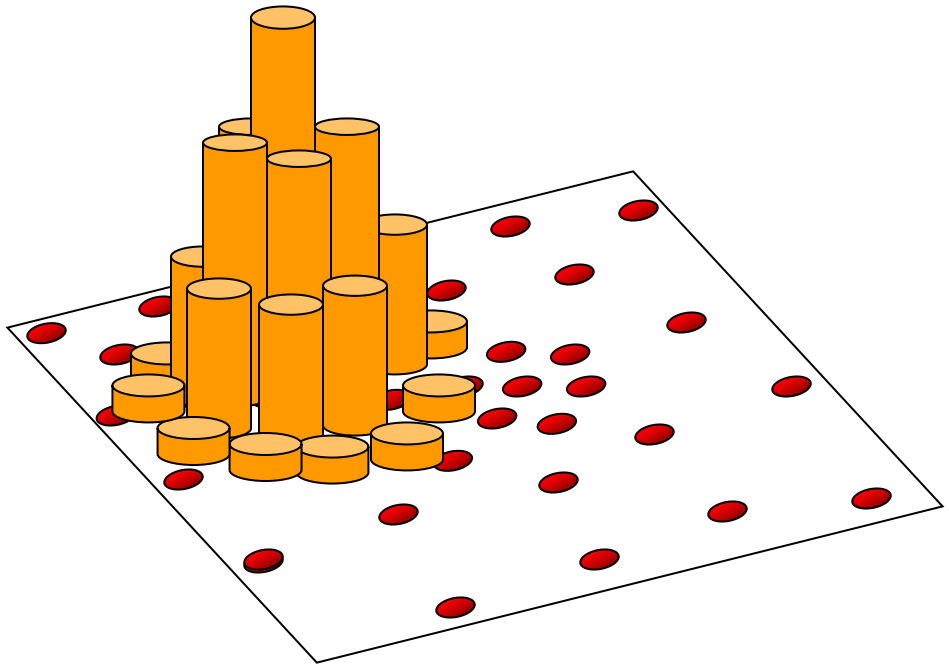
Assumed Underlying PDF



Real Data Samples

Non-Parametric Density Estimation





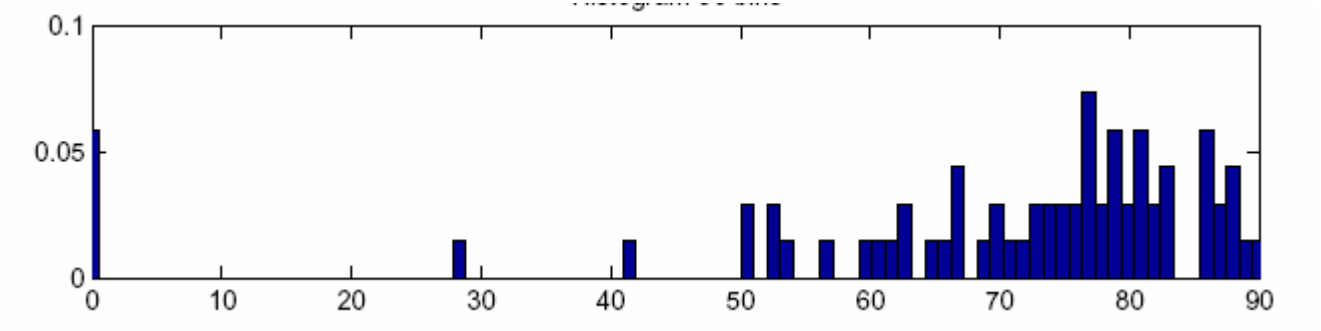
Assumed Underlying PDF

Real Data Samples

[Mean-Shift Clustering Visualization - YouTube]

Parzen Estimation Aka Kernel Density Estimation

- Kernel Density Estimation are closely related to histogram
- Mathematical model of how histogram are formed
- Assume continuous data points

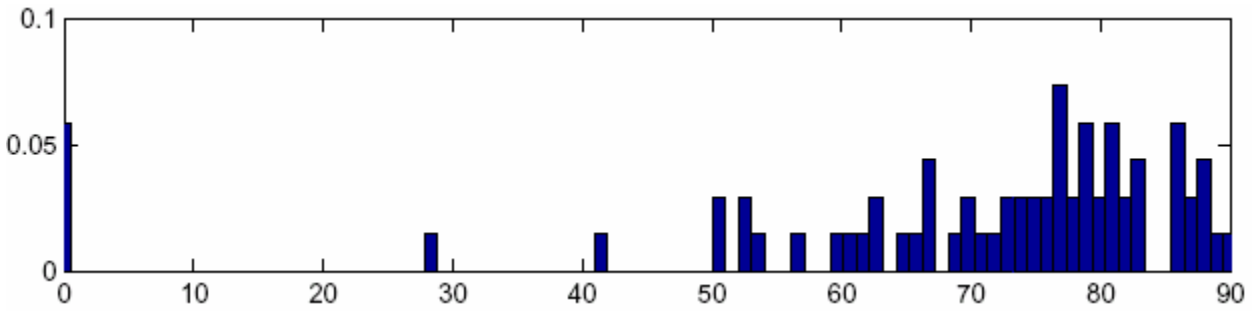


Convolve with box filter of width w (e.g. [1 1 1]) Take samples of result, with spacing w Resulting value at point u represents count of data points falling in range u-w/2 to u+w/2

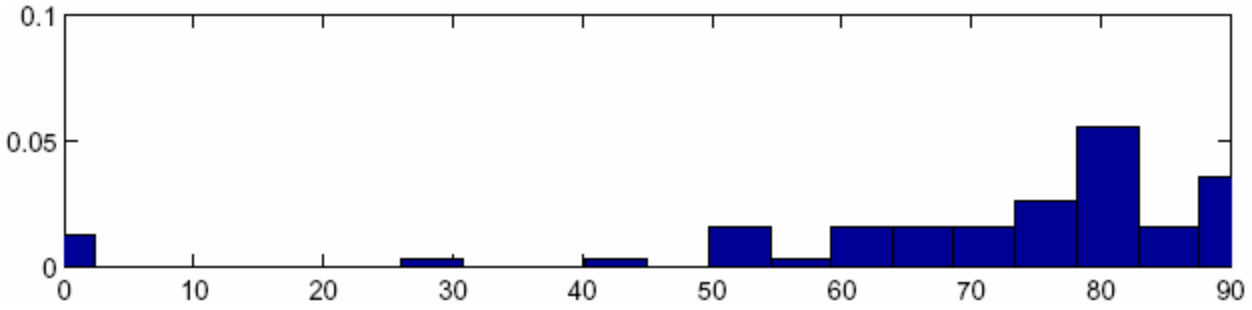
Example Histograms

Box filter

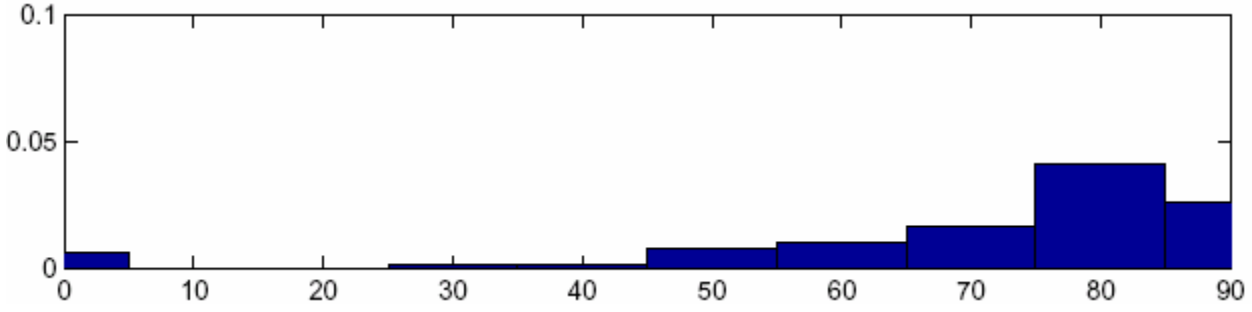
[1 1 1]



[1 1 1 1 1 1 1]



[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]



Increased smoothing

Why Formulate it this way

- Generalize from box filter to other filters (for example Gaussian)
- Gaussian acts as a smoothing filter.

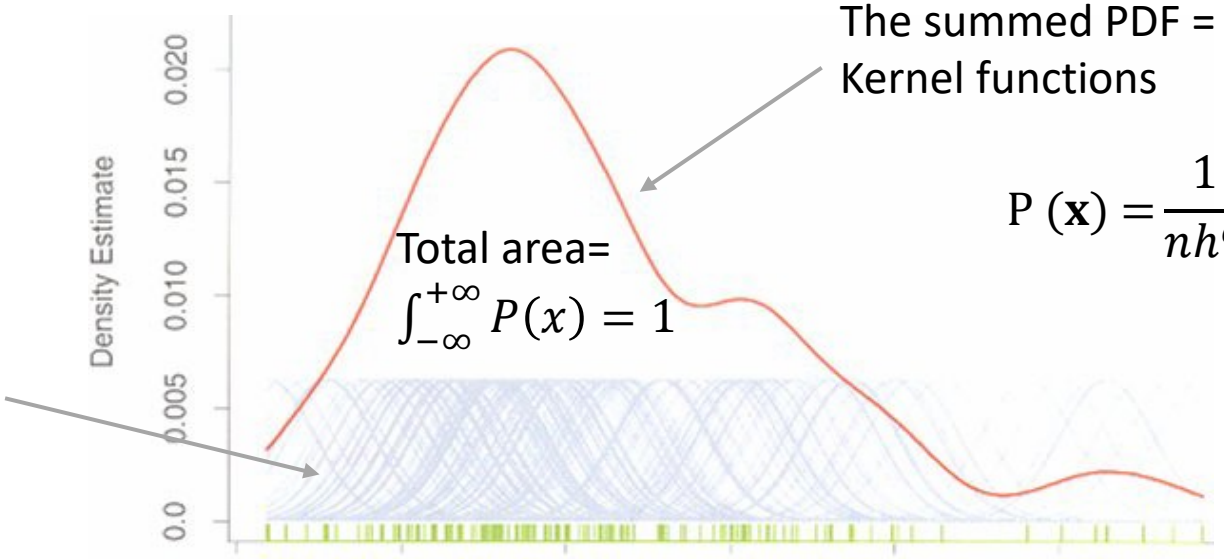
- How to generate the ***Non-Parametric Density Estimation*** (aka ***Kernel Density Estimation***)?

Parzen Estimation Aka Kernel Density Estimation

- **Parzen windows**: Approximate probability density by estimating ***local density of points*** (same idea as a histogram)
  - Convolve points with window/kernel function (e.g., Gaussian) using scale parameter (e.g., sigma)

The summed PDF = Sum of many small Gaussian Kernel functions

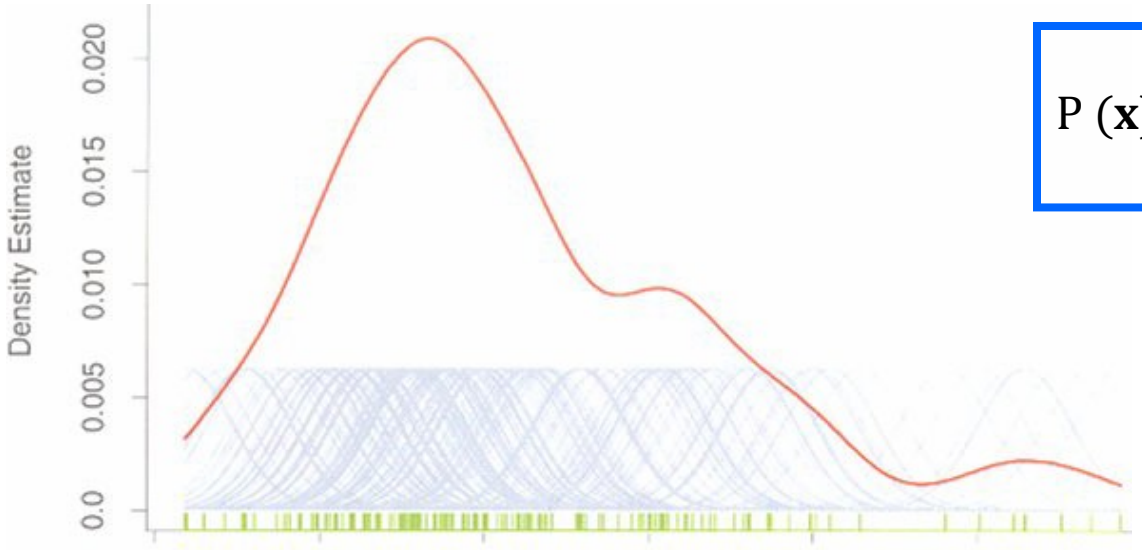
$$\mathrm{P}\,(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$



Kernel function at each sample points

Parzen Estimation Aka Kernel Density Estimation

- **Parzen windows**: Approximate probability density by estimating local density of points (same idea as a histogram)
  - Convolve points with window/kernel function (e.g., Gaussian) using scale parameter (e.g., sigma)



$$\mathrm{P}(\mathbf{x}) = \sum_{\mathrm{i}} c_i \cdot e^{-\frac{(\mathbf{x}-\boldsymbol{\mu}_i)^2}{2{\sigma_i}^2}}$$

Sum of Gaussian contribution aka 'Kernel'
Different types of Kernels !

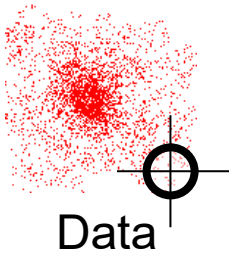
Kernel Density Estimation

Various Kernels

$$\mathrm{P}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

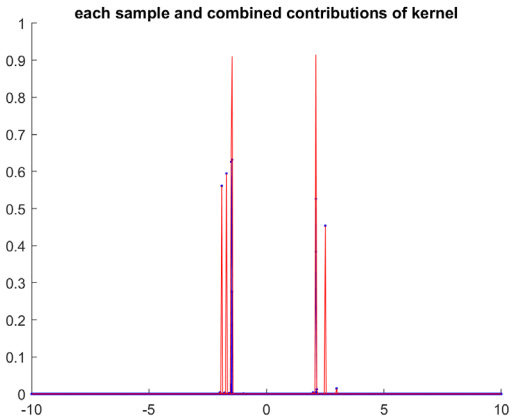
n = *number of samples*
h = *bandwidth (window radius)*
d = *dimension*
x = *target position*
x_i = *samples*

A function of some finite number of data points
$x_1 \ldots x_n$

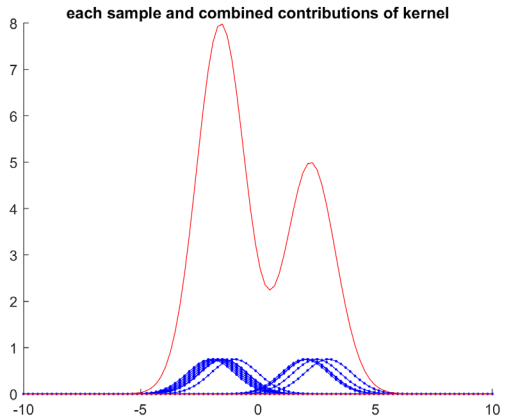


Kernel Density Estimation

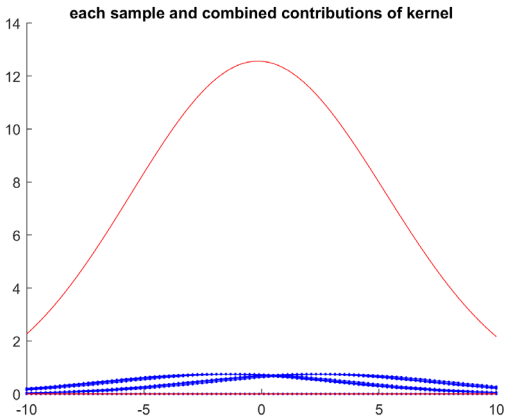
- **Effect of different bandwidth**



Small bandwidth
Give many peaks



Medium bandwidth
Peaks generated



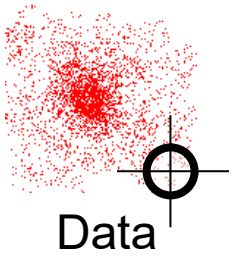
Large bandwidth
Peaks cannot be found

Kernel Density Estimation

Various Kernels

$$\mathrm{P}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

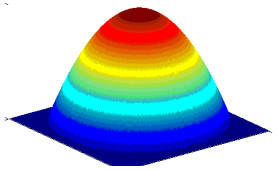
A function of some finite number of data points $x_1 \ldots x_n$



Examples:

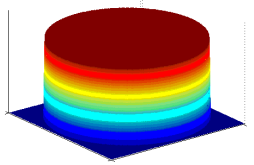
- Epanechnikov Kernel

$$K_E(\mathbf{x}) = \begin{cases} c\left(1 - \|\mathbf{x}\|^2\right) & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



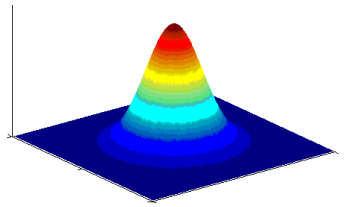
- Uniform Kernel

$$K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



- Normal Kernel (Gaussian)

$$K_N(\mathbf{x}) = c \cdot \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$

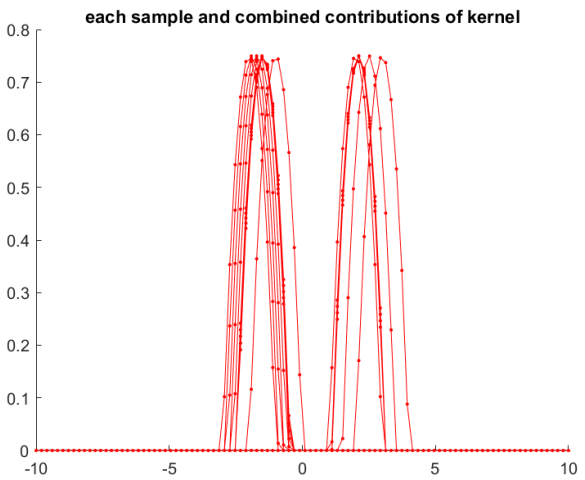


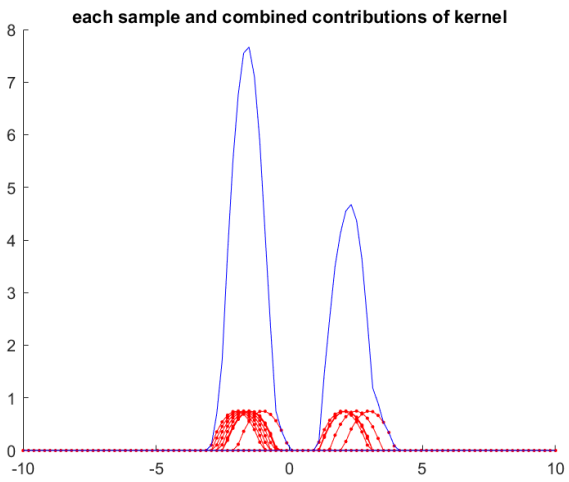
MATLAB Implementation of Parzen Windows

- Epanechnikov Kernel

$$K_E(\mathbf{x}) = \begin{cases} c\left(1 - \|\mathbf{x}\|^2\right) & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{P}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$



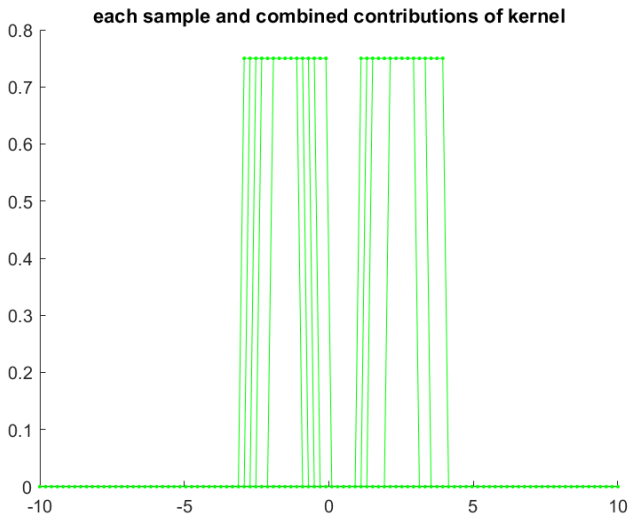


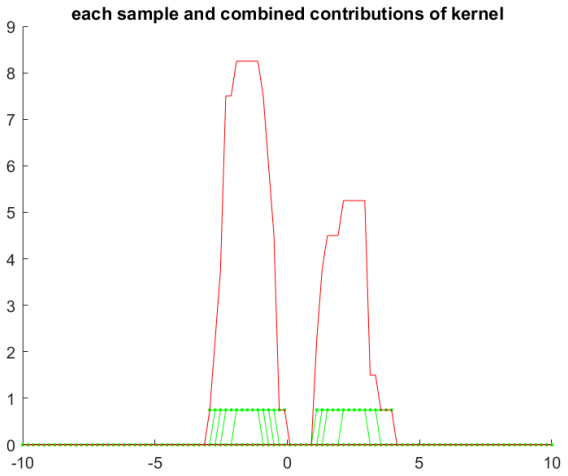
MATLAB Implementation of Parzen Windows

- Uniform Kernel

$$K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{P}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$



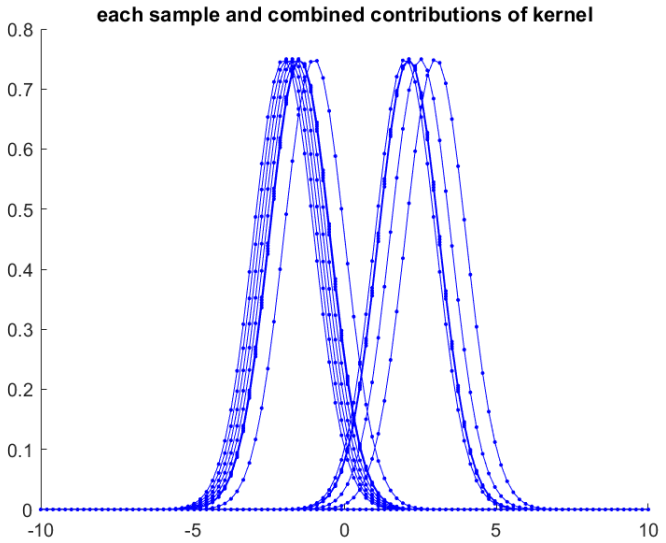


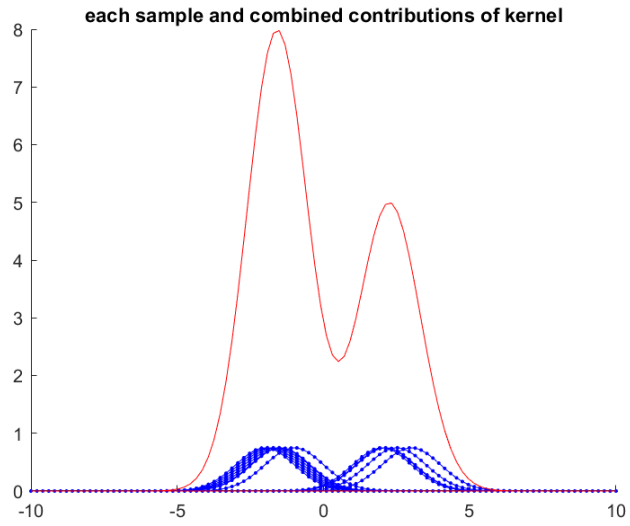
MATLAB Implementation of Parzen Windows

- Normal Kernel

$$K_N(\mathbf{x}) = c \cdot \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$

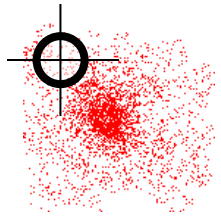
$$\mathrm{P}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$





Relating KDE to Mean-Shift

- Given:
  - Finite number of data points $x_1 \ldots x_n$
  - *A Kernel Function* $K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$



Data

- Goal:
  - *Find the mean sample point* $\mathbf{x}$
  - *Find the peak of the Probability Density Estimation*

Mean-Shift Algorithm

- Steps for mean-shift

- 1. Initialize $\mathbf{x}$
- 2. Compute mean $\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i K(\frac{\mathbf{x}-\mathbf{x}_i}{h})}{\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x}_i}{h})}\right]$
- 3. Compute the shift $\mathbf{v}(\mathbf{x}) = \mathrm{m}(\mathbf{x}) - \mathbf{x}$
- 4. update $\mathbf{x} = \mathbf{x} + \mathbf{v}(\mathbf{x})$
- 5. if $\mathbf{v}(\mathbf{x}) > \varepsilon$ repeat 2~4

How is KDE related to mean shift?

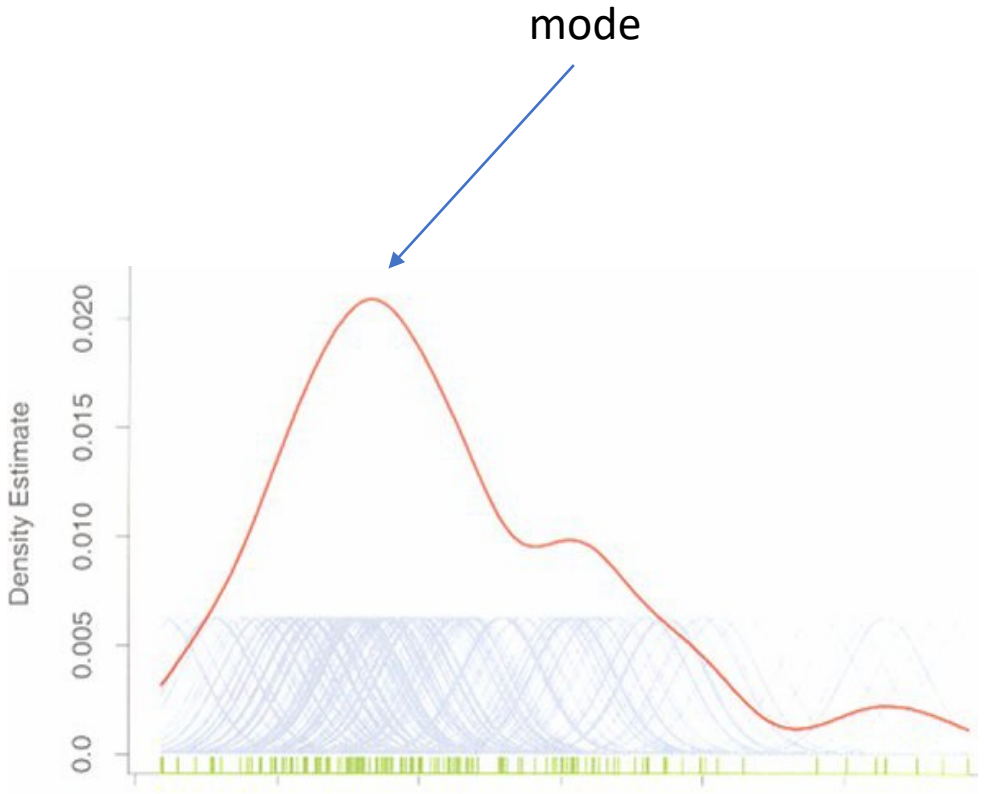
- Recall KDE can be expressed as:

$$P(\mathbf{x}) = \frac{1}{nh^d} c \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

- Gradient of PDE is related to the mean shift vector

$$\nabla P(\mathbf{x}) \propto \mathbf{v}(\mathbf{x})$$

- The ***mean shift vector*** in the direction of the gradient of KDE
- Mean-shift algorithm is ***maximizing*** the objective function



How is KDE related to mean shift?

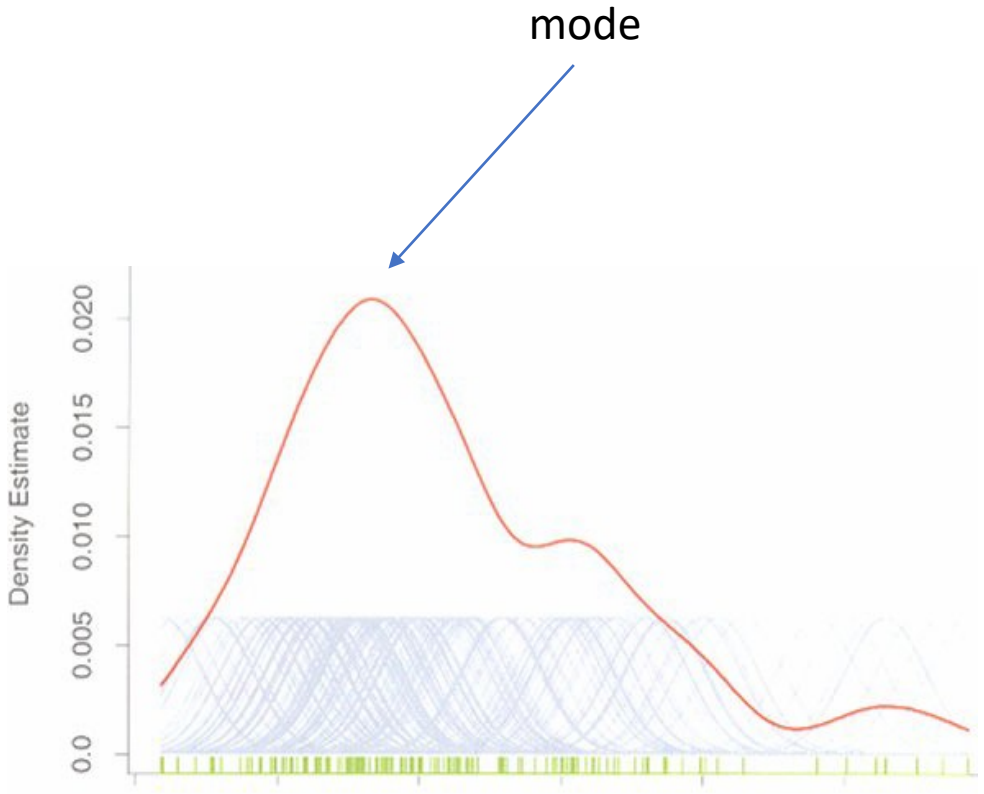
- In mean-shift we are trying to optimize

$$\arg\max_{\mathbf{x}} P(\mathbf{x})$$

$$\arg\max_{\mathbf{x}} = \frac{1}{nh^d} c \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

- How to do this?

Using gradient ascent!



Nonparametric Kernel Density Estimation

- The KDE

$$P(\mathbf{x}) = \frac{1}{nh^d} c \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)$$

- The Gradient

$$\nabla P(\mathbf{x}) = \frac{1}{nh^d} c \sum_{i=1}^{n} \nabla k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)$$

- Taking Derivative

$$\nabla P(\mathbf{x}) = \frac{1}{nh^{d+2}} 2c \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)$$

Define another function

$$k'(\mathbf{x}) = -g(\mathbf{x})$$

Proof:

- **Proof:** $\nabla P(\mathbf{x}) = \frac{1}{nh^{d+2}} 2c \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$

- Since $P(\mathbf{x}) = \frac{1}{nh^d} c \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$

$$\nabla P(\mathbf{x}) = \frac{\partial}{\partial \mathrm{x}} \left(\frac{1}{nh^d} c \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \right)$$

$$\nabla P(\mathbf{x}) = \frac{1}{nh^d} c \sum_{i=1}^{n} \frac{\partial k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\partial \mathrm{x}})$$

By Chain Rule

$$\frac{\partial k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\partial \mathrm{x}} = \frac{\partial k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\partial\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} \cdot \frac{\partial\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\partial\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)} \cdot \frac{\partial\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\partial \mathbf{x}} = \frac{\partial k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\partial k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} 2\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\left(\frac{1}{h}\right)$$

Proof: Cont'd

By Chain Rule

$$\frac{\partial k(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2}{\partial \mathrm{x}} = \frac{\partial k(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2)}{\partial(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|)^2} \cdot \frac{\partial k(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2)}{\partial(\frac{\mathbf{x}-\mathbf{x}_i}{h})} \cdot \frac{\partial\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\partial \mathbf{x}} = \frac{\partial k(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2)}{\partial(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|)^2} 2\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\left(\frac{1}{h}\right)$$

$$\frac{\partial k(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2}{\partial \mathrm{x}} = k'(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2)(\,2\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\left(\frac{1}{h}\right)$$

Hence:

$$\nabla P(\mathbf{x}) = \frac{1}{nh^d} c \sum_{i=1}^{n} (\frac{\partial k(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2}{\partial \mathrm{x}})$$

$$\nabla P(\mathbf{x}) = \frac{2}{nh^{d+2}} c \sum_{i=1}^{n} (\mathbf{x}-\mathbf{x}_i)\ \ k'(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2)$$

Proof: Cont'd

- We have

$$\nabla P(\mathbf{x}) = \frac{1}{nh^{d+2}} 2c \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}) g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \text{ where } k'(\mathbf{x}) = -g(\mathbf{x})$$

- Expanding

$$\nabla P(\mathbf{x}) = \frac{1}{nh^{d+2}} 2c \sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) - \frac{1}{n} 2c \sum_{i=1}^{n} \mathbf{x} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

Multiply by one

- Grouping like term

$$\nabla P(\mathbf{x}) = \frac{1}{nh^{d+2}} 2c \sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \left(\frac{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}\right) - \frac{1}{n} 2c \sum_{i=1}^{n} \mathbf{x} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

- Grouping like terms.

$$\nabla P(\mathbf{x}) = \frac{1}{nh^{d+2}} 2c \sum_{i=1}^{n} g(\|\mathbf{x}-\mathbf{x}_i\|^2) \left(\frac{\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right)$$

$\nabla P(\mathbf{x}) \propto mean\ shift\ vector$

- The equation becomes:

$$\nabla P(\mathbf{x}) = \frac{1}{n} 2c \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \left(\frac{\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right)$$

constant | mean | Shift

- Mean-shift Vector = $\mathrm{v}(\mathbf{x}) = \left(\frac{\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right)$

$$\nabla P(\mathbf{x}) \propto \left(\frac{\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right) \text{ if } g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \text{ is positive}$$

$\nabla P(\mathbf{x}) \propto mean\ shift\ vector$

- The equation becomes:

$$\nabla P(\mathbf{x}) = \underbrace{\frac{1}{n} 2c \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}_{\text{constant}} \underbrace{\left(\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}\right)}_{\text{mean}} \underbrace{- \mathbf{x})}_{\text{Shift}}$$

- Remember in mean shift, we have

$$\mathbf{v}(\mathbf{x}) = \mathbf{m}(\mathbf{x}) - \mathbf{x}$$

$$\mathbf{v}(\mathbf{x}) = \frac{\nabla P(\mathbf{x})}{\frac{1}{n} 2c \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}$$

Proof: g is positive constant

- Given $g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) = -\frac{\partial k(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2)}{\partial(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|)^2}$

For Epanechnikov Kernel

$$K_E(\tfrac{\mathbf{x}-\mathbf{x}_i}{h}) = \begin{cases} c\left(1-\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) & \left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) = -1 * \begin{cases} \frac{\partial c\left(1-\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\partial(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2)} & \left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) = -1 * \begin{cases} -c & \left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases} = \begin{cases} c & \left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Therefore

$g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$ is positive constant!

We are interested only in a special class of radially symmetric kernels satisfying

$$K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2), \tag{5}$$

in which case it suffices to define the function $k(x)$ called the *profile* of the kernel, only for $x \geq 0$. The normalization constant $c_{k,d}$, which makes $K(\mathbf{x})$ integrate to one, is assumed strictly positive.

$$\hat{f}_{h,K}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \tag{11}$$

The first step in the analysis of a feature space with the underlying density $f(\mathbf{x})$ is to find the modes of this density. The modes are located among the zeros of the gradient $\nabla f(\mathbf{x}) = \mathbf{0}$ and the mean shift procedure is an elegant way to locate these zeros *without* estimating the density.

$$\hat{\nabla} f_{h,K}(\mathbf{x}) \equiv \nabla \hat{f}_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k'\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right).$$

We define the function

$$g(x) = -k'(x), \tag{13}$$

assuming that the derivative of the kernel profile k exists for all $x \in [0, \infty)$, except for a finite set of points. Now, using $g(x)$ for profile, the kernel $G(\mathbf{x})$ is defined as

$$G(\mathbf{x}) = c_{g,d} g\left(\|\mathbf{x}\|^2 \right), \tag{14}$$

where $c_{g,d}$ is the corresponding normalization constant. The

$$\begin{aligned}
&\hat{\nabla} f_{h,K}(\mathbf{x}) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}) g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right],
\end{aligned}$$

$$\hat{f}_{h,G}(\mathbf{x}) = \frac{c_{g,d}}{nh^d} \sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \tag{16}$$

The second term is the *mean shift*

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x}, \tag{17}$$

i.e., the difference between the weighted mean, using the kernel G for weights, and $\mathbf{x}$, the center of the kernel (window). From (16) and (17), (15) becomes

$$\hat{\nabla} f_{h,K}(\mathbf{x}) = \hat{f}_{h,G}(\mathbf{x}) \frac{2c_{k,d}}{h^2 c_{g,d}} \mathbf{m}_{h,G}(\mathbf{x}), \tag{18}$$

points reside. Since the mean shift vector is aligned with the local gradient estimate, it can define a path leading to a stationary point of the *estimated* density. The modes of the density are such stationary points. The *mean shift procedure*, obtained by successive

- computation of the mean shift vector $\mathbf{m}_{h,G}(\mathbf{x})$,
- translation of the kernel (window) $G(\mathbf{x})$ by $\mathbf{m}_{h,G}(\mathbf{x})$,

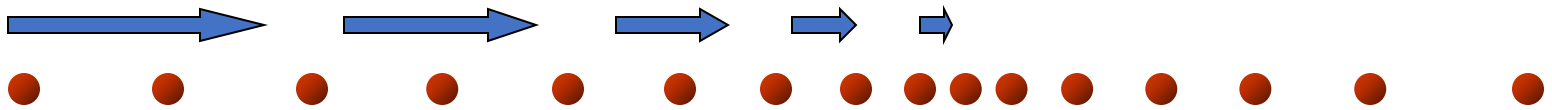
is guaranteed to converge at a nearby point where the estimate (11) has zero gradient, as will be shown in the next section. The

Summary - Mean-Shift Algorithm

- Steps for mean-shift

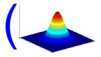
- 1. Initialize $\mathbf{x}$
- 2. Compute mean $\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i K(\frac{\mathbf{x}-\mathbf{x}_i}{h})}{\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x}_i}{h})}\right]$
- 3. Compute the shift $\mathbf{v}(\mathbf{x}) = \mathrm{m}(\mathbf{x}) - \mathbf{x}$
- 4. update $\mathbf{x} = \mathbf{x} + \mathbf{v}(\mathbf{x})$
- 5. if $\mathbf{v}(\mathbf{x}) > \varepsilon$ repeat 2~4

Mean Shift Properties

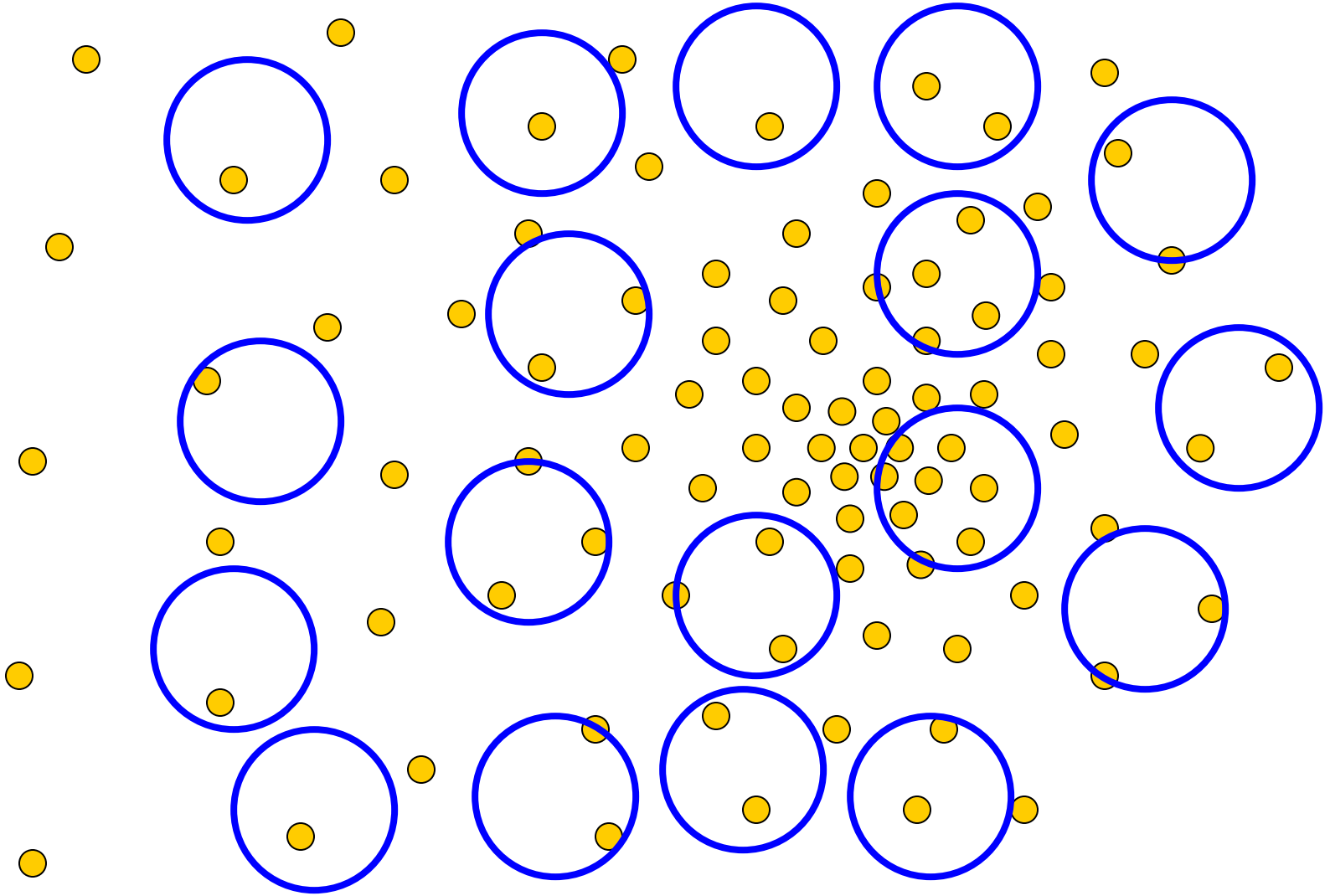


- Automatic convergence speed – the mean shift vector size depends on the gradient itself.
- Near maxima, the steps are small and refined

Adaptive Gradient Ascent

- Convergence is guaranteed for infinitesimal steps only ➔ infinitely convergent, (therefore set a lower bound)
- For Uniform Kernel (), convergence is achieved in a finite number of steps
- Normal Kernel () exhibits a smooth trajectory, but is slower than Uniform Kernel ().

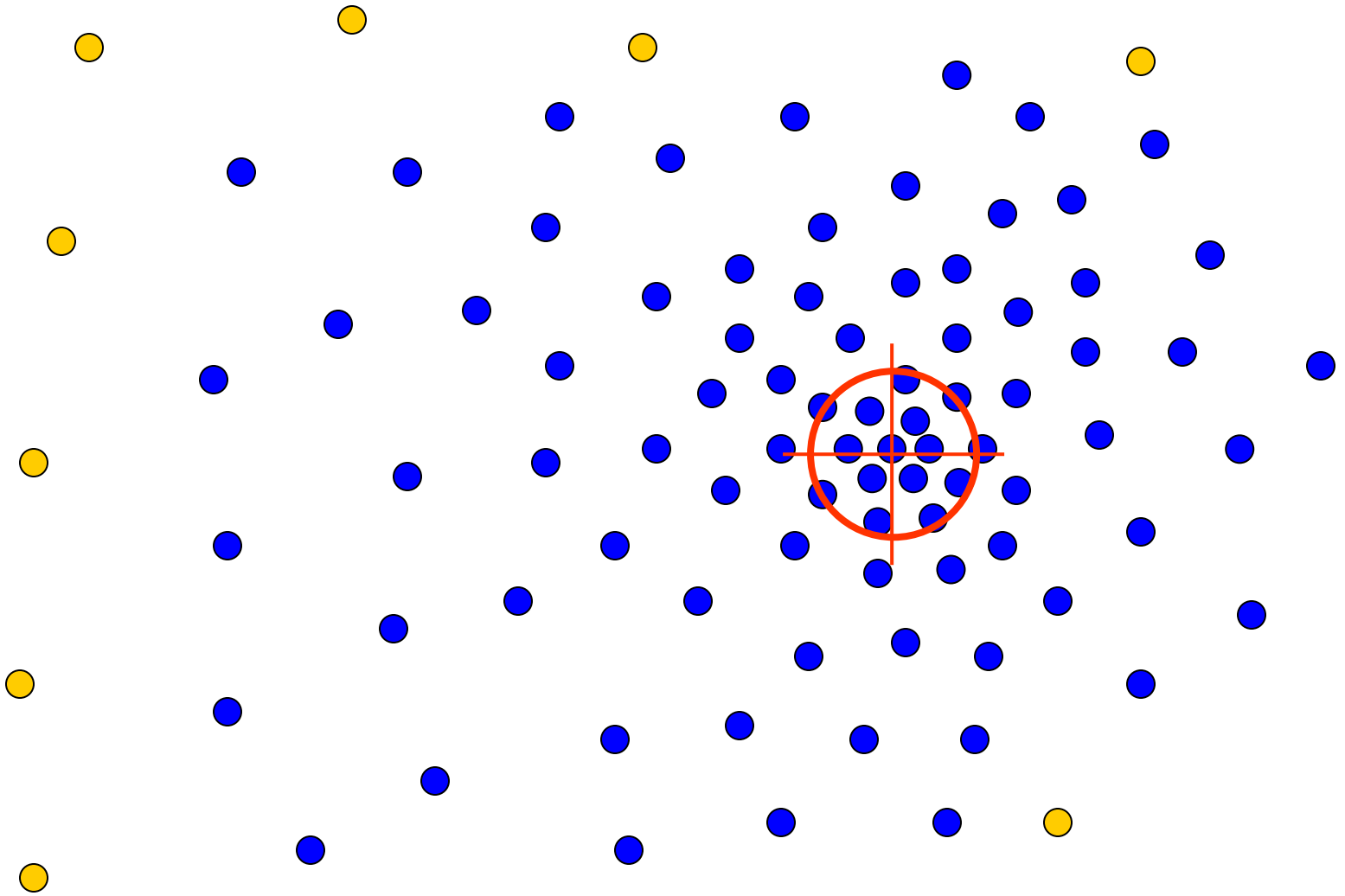
Real Modality Analysis



Tessellate the space with windows

Run the procedure in parallel

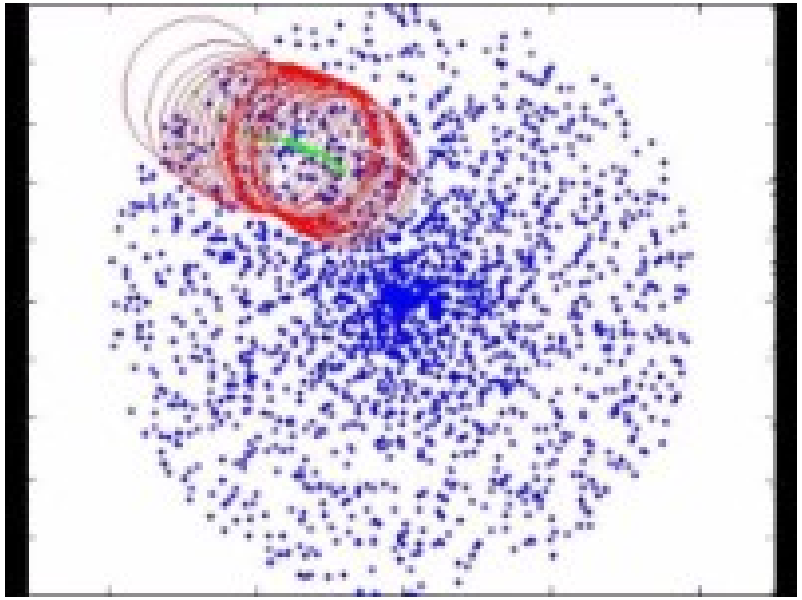
Real Modality Analysis



The blue data points were traversed by the windows towards the mode

Real Modality Analysis

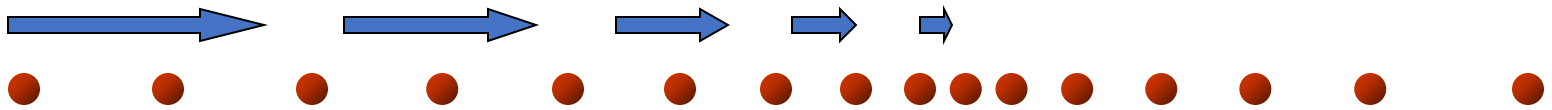
An example



Window tracks signify the steepest ascent directions

https://www.youtube.com/watch?v=hJg7ik4x95U

Mean Shift Strengths & Weaknesses



Strengths :

- Application independent tool
- Suitable for real data analysis
- Does not assume any prior shape (e.g. elliptical) on data clusters
- Can handle arbitrary feature spaces
- Only ONE parameter to choose
- *h* (window size) has a physical meaning, unlike K-Means

Weaknesses :

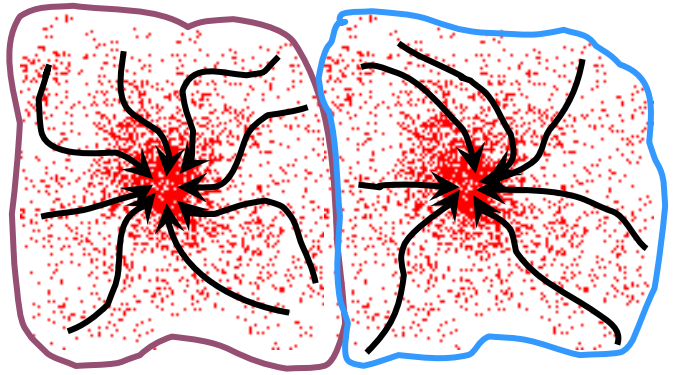
- The window size (bandwidth selection) is not trivial
- Inappropriate window size can cause modes to be merged, or generate additional "shallow" modes ➔ Use adaptive window size

Mean Shift Applications

Clustering

Cluster : All data points in the ***attraction basin*** of a mode

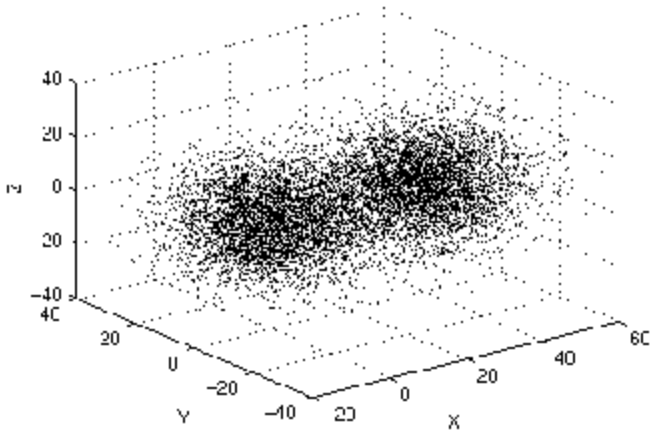
Attraction basin : the region for which all trajectories lead to the same mode



Mean Shift : A robust Approach Toward Feature Space Analysis, by Comaniciu, Meer

Clustering

Synthetic Examples



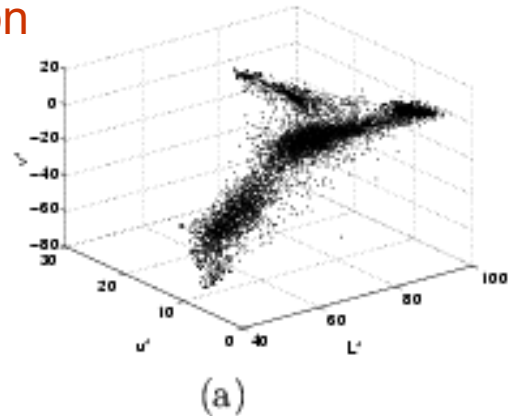
Simple Modal Structures

Complex Modal Structures

Clustering

Real Example

Feature space:
L*u*v representation



(a)

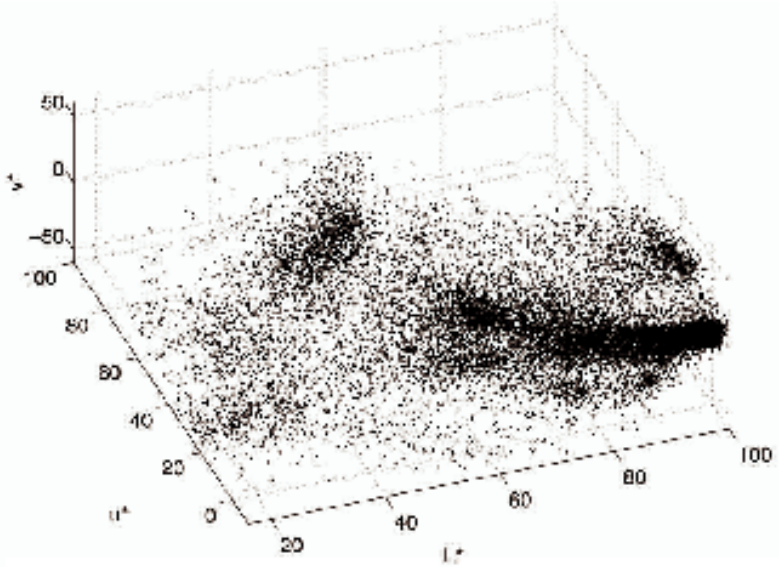
Initial window centers

M

pruning

Clustering

Real Example

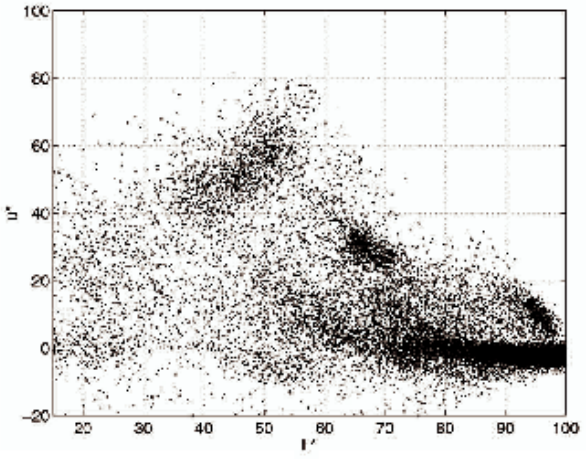


L*u*v space representation

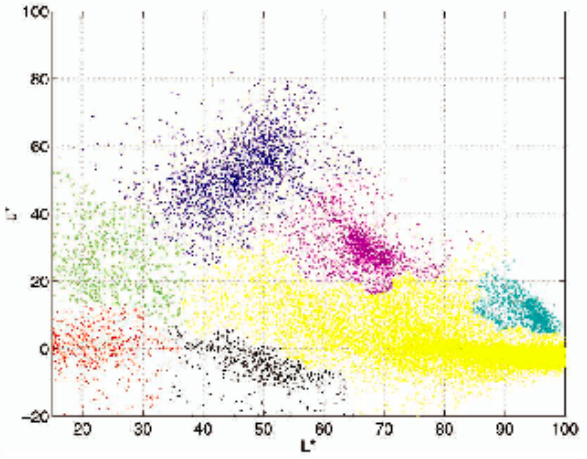
Clustering

Real Example

2D (L*u) space representation

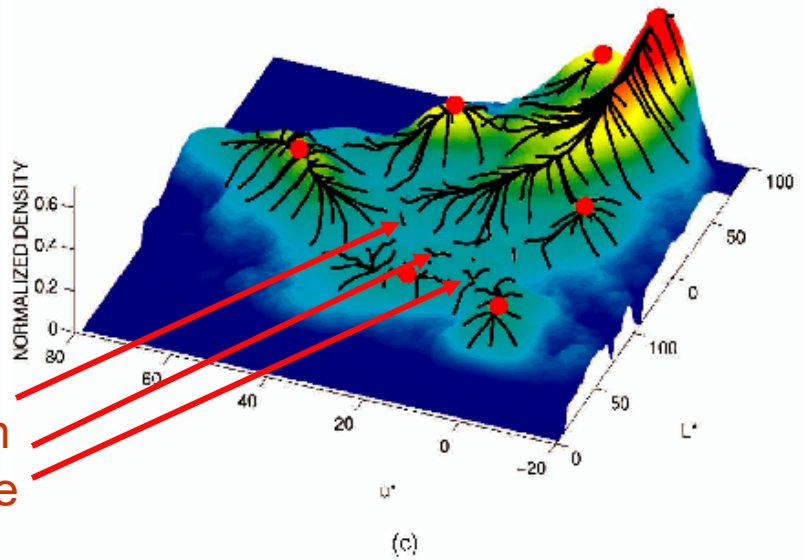


(a)



(b)

Final clusters

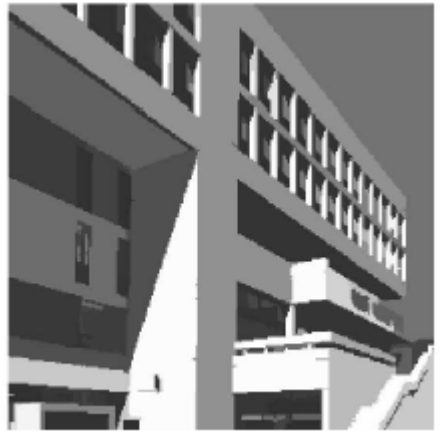


(c)

Not all trajectories in the attraction basin reach the same mode

Segmentation

Example



…when feature space is only gray levels…

Segmentation

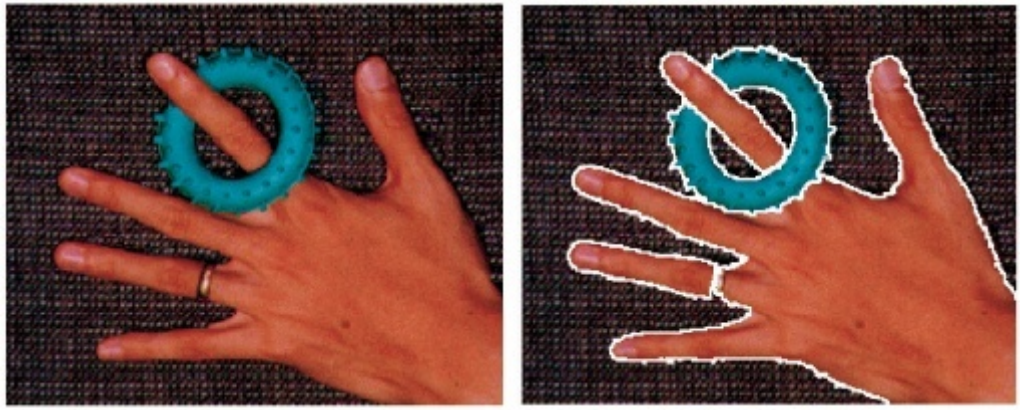
Example

Segmentation

Example

Segmentation

Example

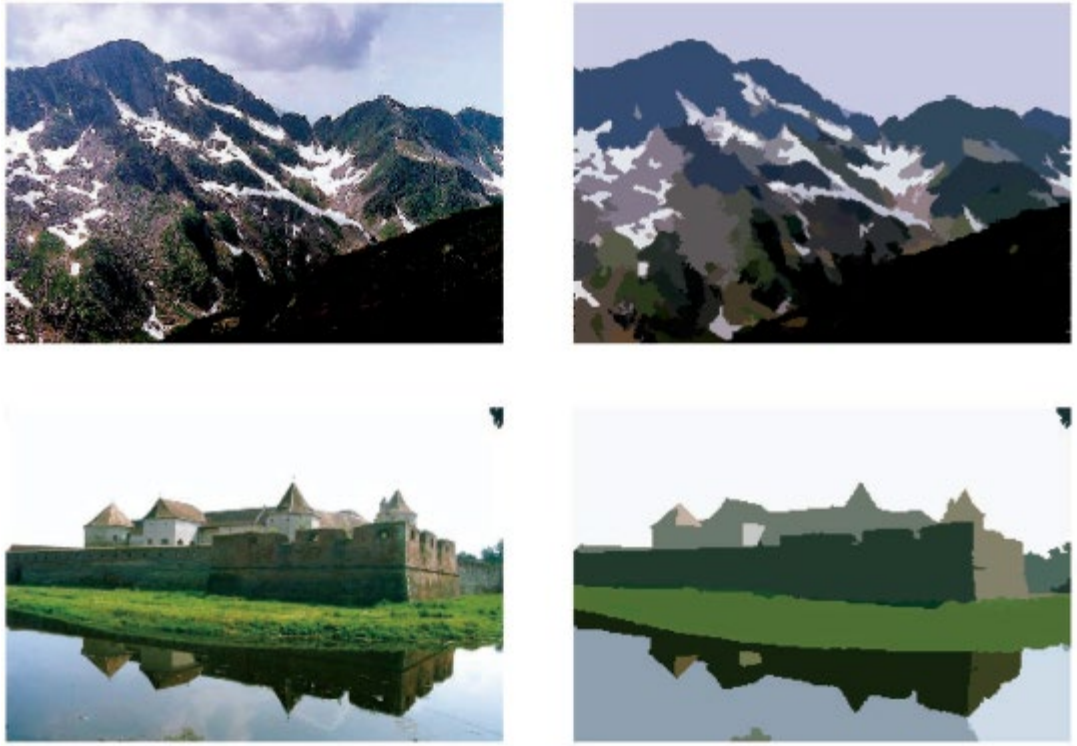


Segmentation

Example

Segmentation

Example



Segmentation

Example