Computer Vision

11. Clustering and Compact Data Representation

I-Chen Lin

College of Computer Science

National Yang Ming Chiao Tung University

Compacting Data

- More image samples → usually more fidelity.
  - How to keep more samples in the same devices (memory, disks …) ?

- Data compression in multimedia courses.
  - Lossless compression, e.g. Huffman coding
  - Lossy compression, e.g. vector quantization
    - JPEG
    - MPEG
    - MP3 (MPEG 1 Audio layer III)
    - ……

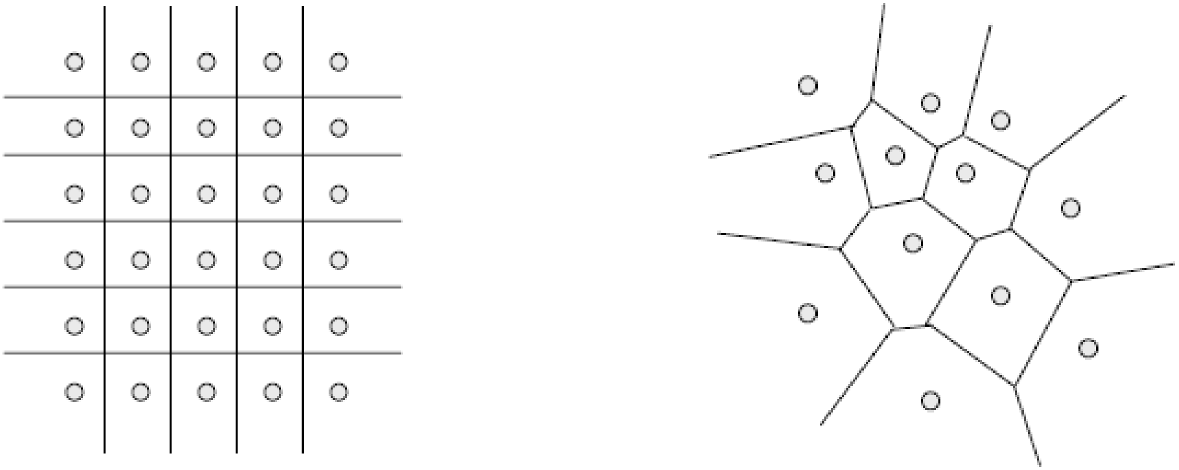
Outline

- Vector Quantization (VQ)
- Mean-shift Clustering
- Principal Component Analysis (PCA)

Vector Quantization (VQ)

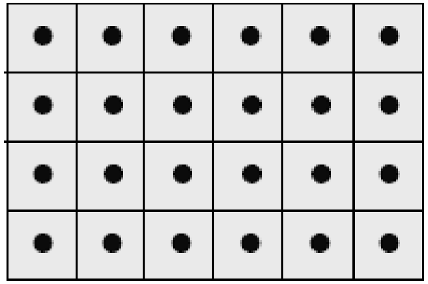
- To project a continuous input space on a discrete output space, while minimizing the loss of information.

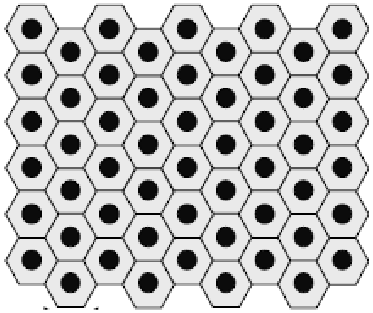
E.g. 2D space



Vector Quantization (cont.)

- VQ =
  - A codebook (set of centroid or codeword, etc.)
  - A quantization function
- E.g.





$E_{vq} = 0.962E_{sq}$

www.dice.ucl.ac.be/~verleyse/lectures/elec2870

Lloyd's algorithm

1. Choice of an initial codebook.

2. All points x_i are encoded; E_{VQ} is evaluated.

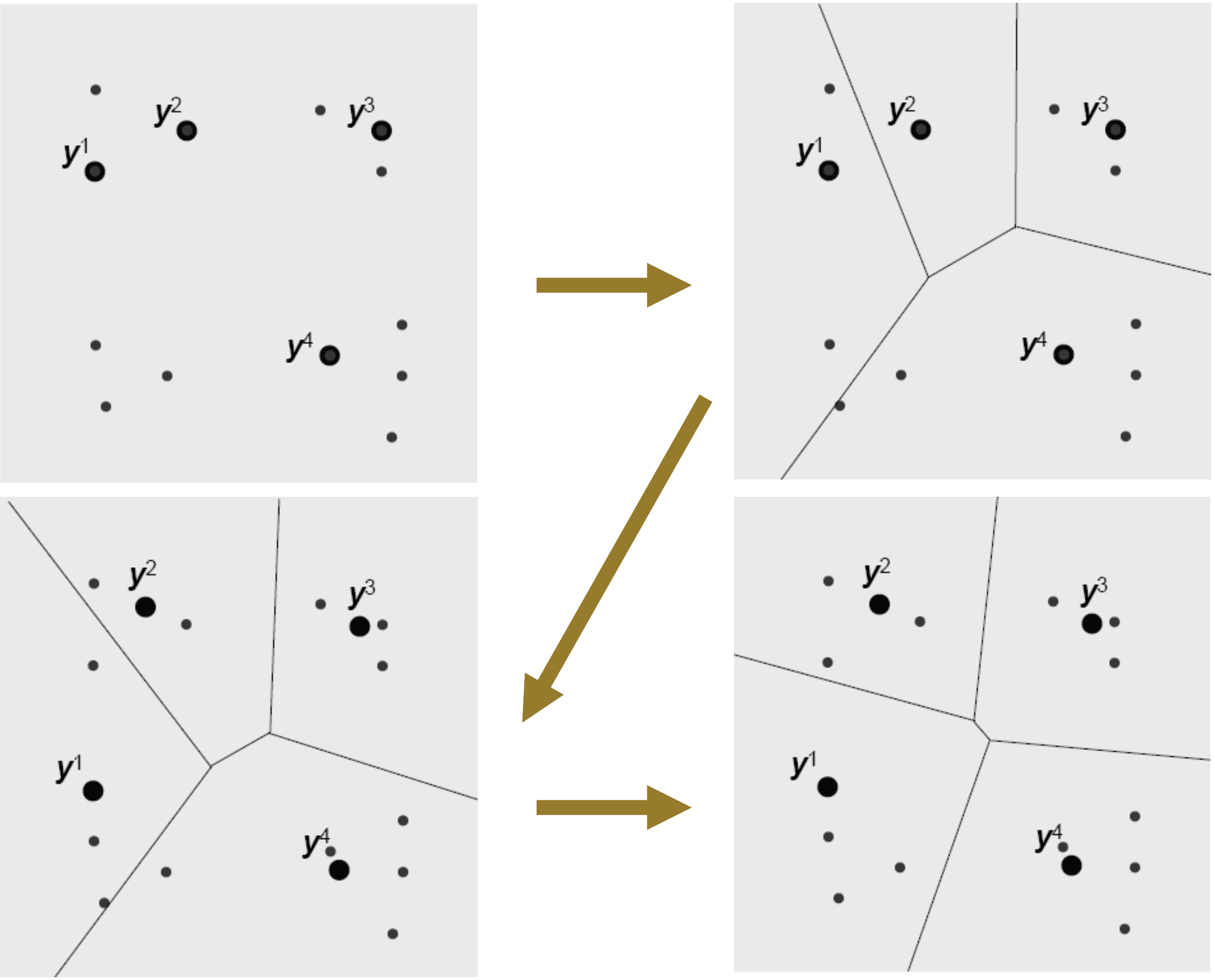
3. If E_{VQ} is small enough, then stop.

4. All centroids y_j are replaced by the center-of-gravity of the data x_i associated to y_j in step 2.

5. Back to step 2.

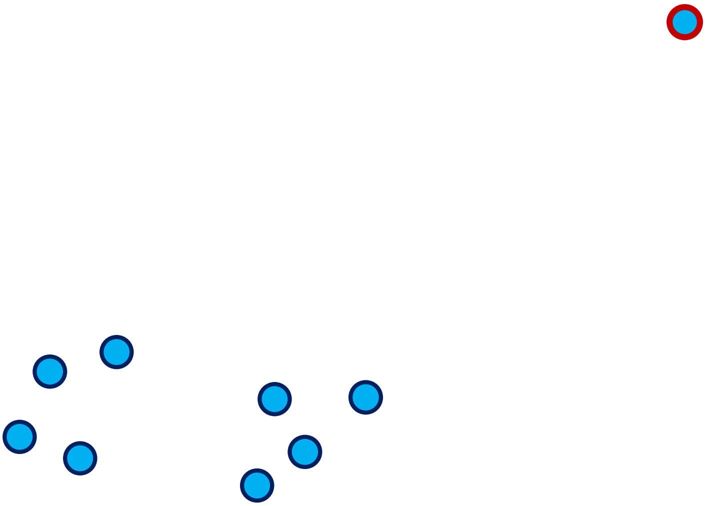
K-means clustering

Lloyd’s algorithm



K-means clustering

- Easy and intuitive to implementation.
- Computationally efficient (with reasonable stop criteria)
- How to select the “K” number?
- How about the effects of outliers?



Clustering for Image Segmentation

- Image segmentation: decompose an image into several meaningful or visually similar parts.

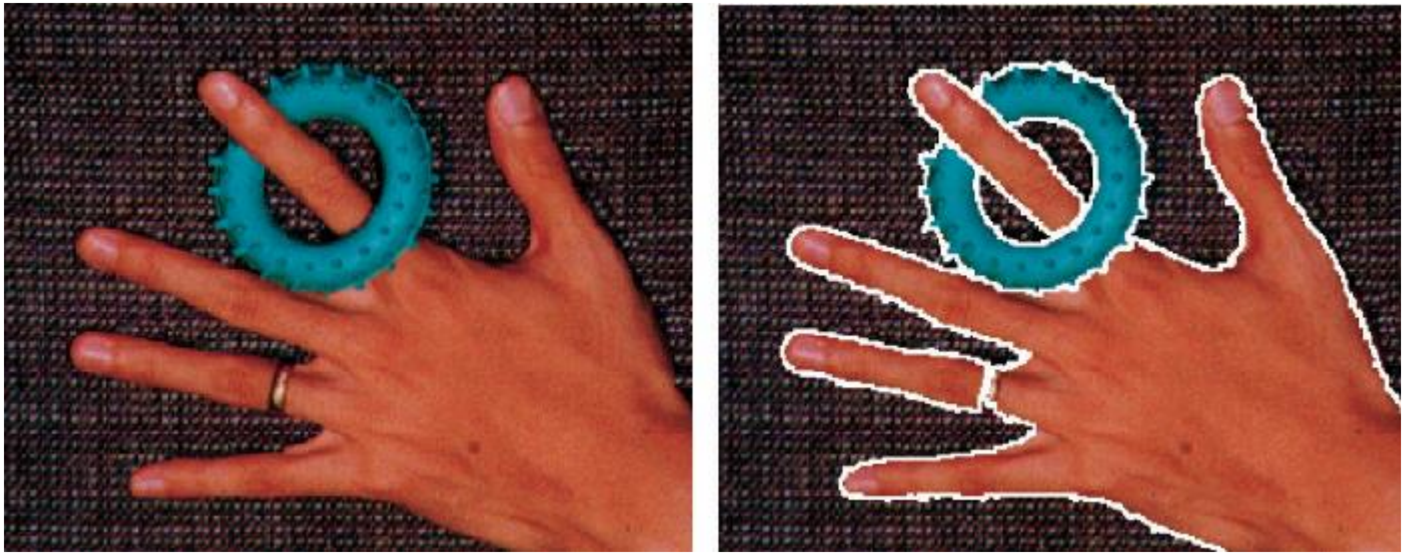
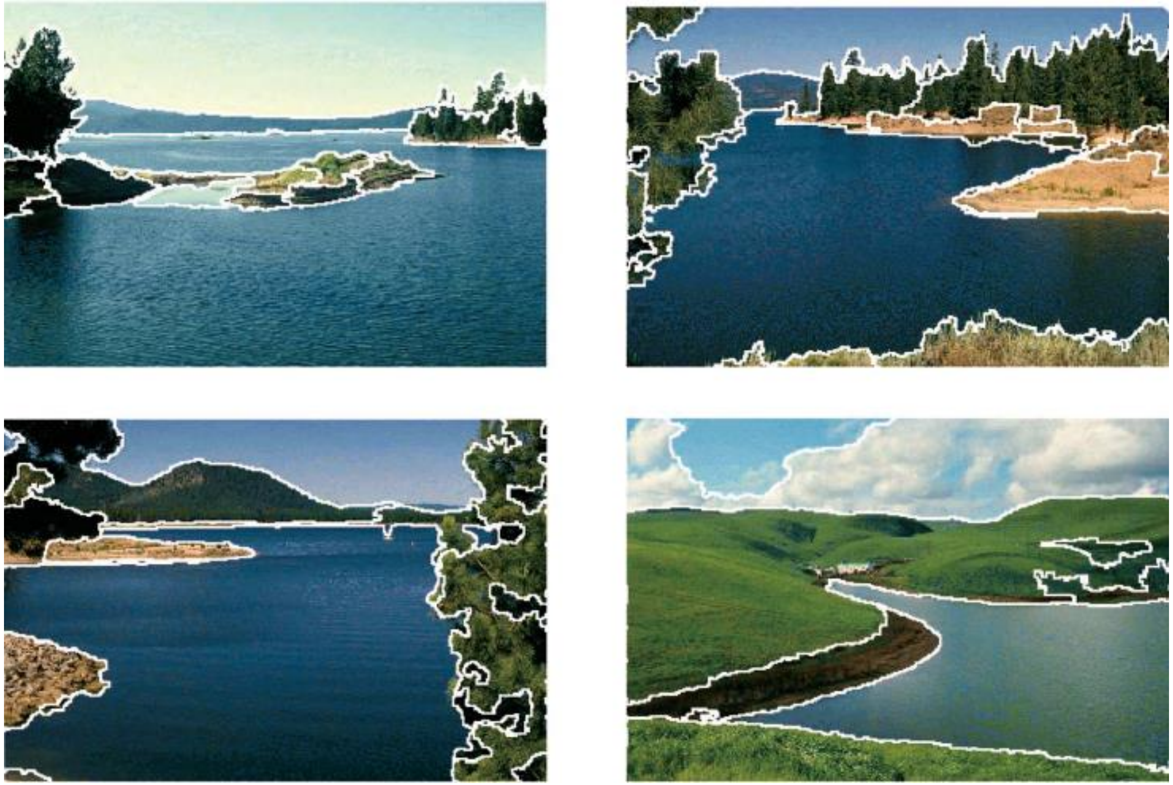


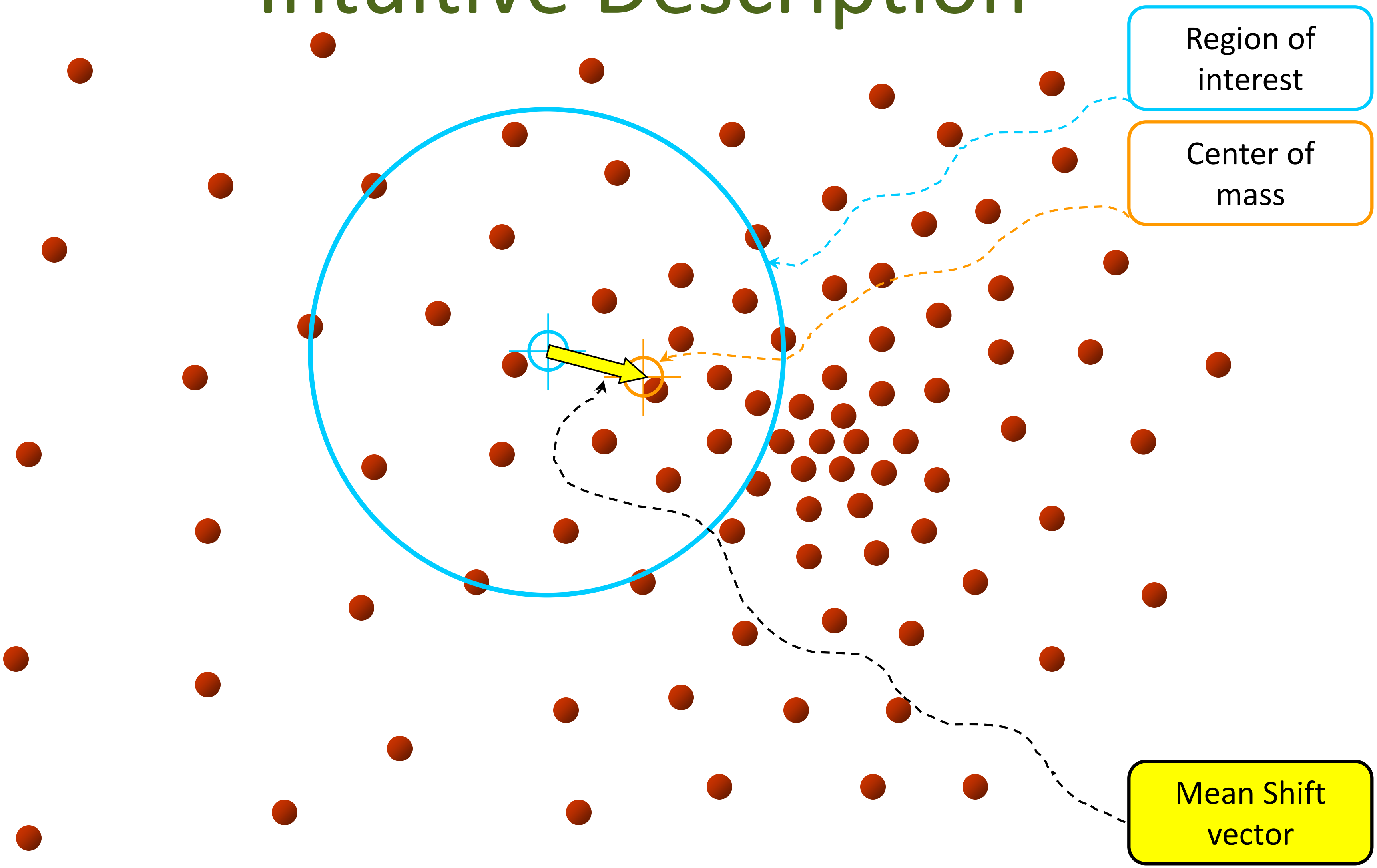
Figure from D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach toward Feature Space Analysis”, IEEE T. PAMI, 2002.

Mean Shift for Clustering and Segmentation

- D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach toward Feature Space Analysis”, IEEE T. PAMI, 2002.



Intuitive Description

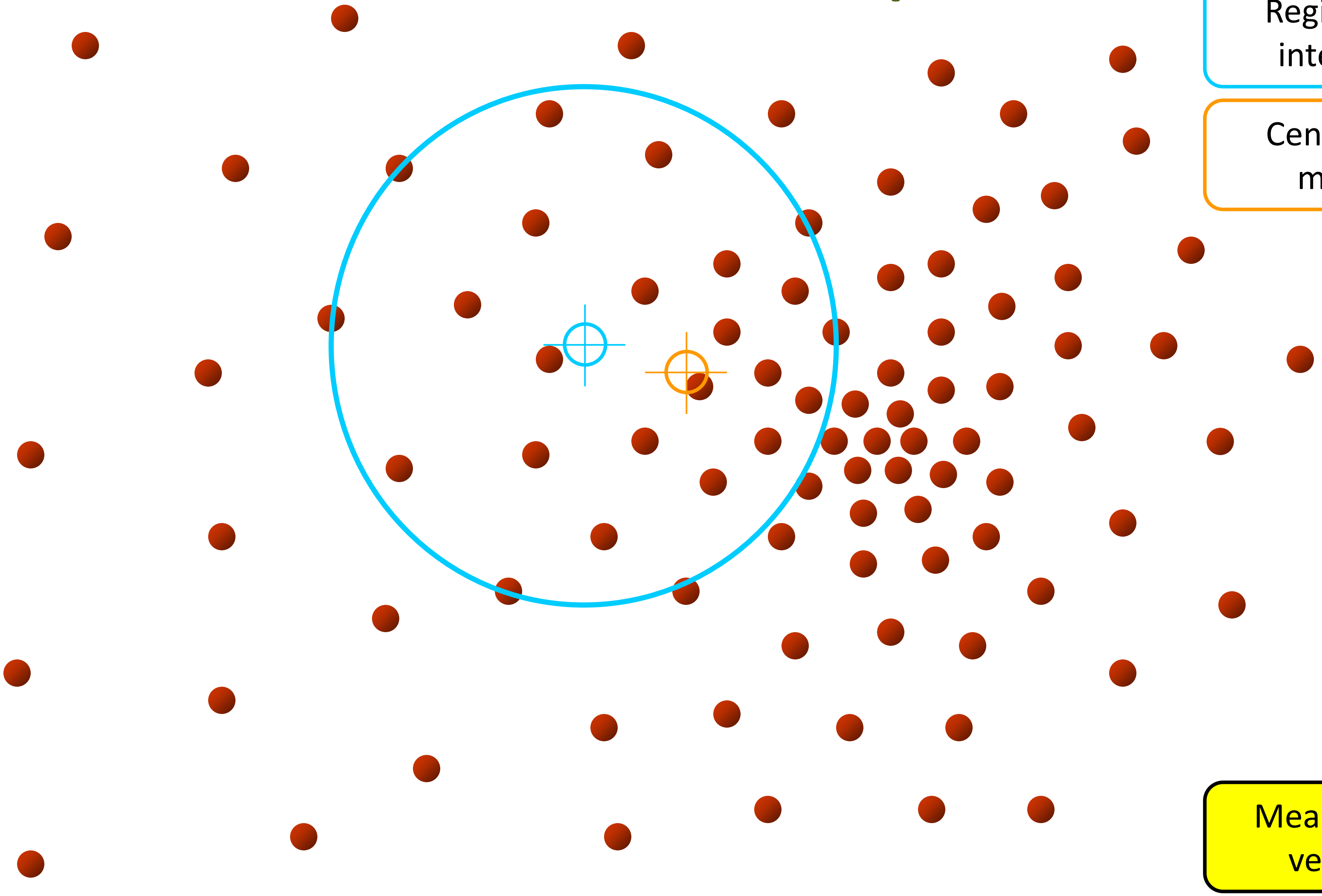


Objective : Find the densest region

Distribution of identical billiard balls

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Intuitive Description



Region of interest

Center of mass

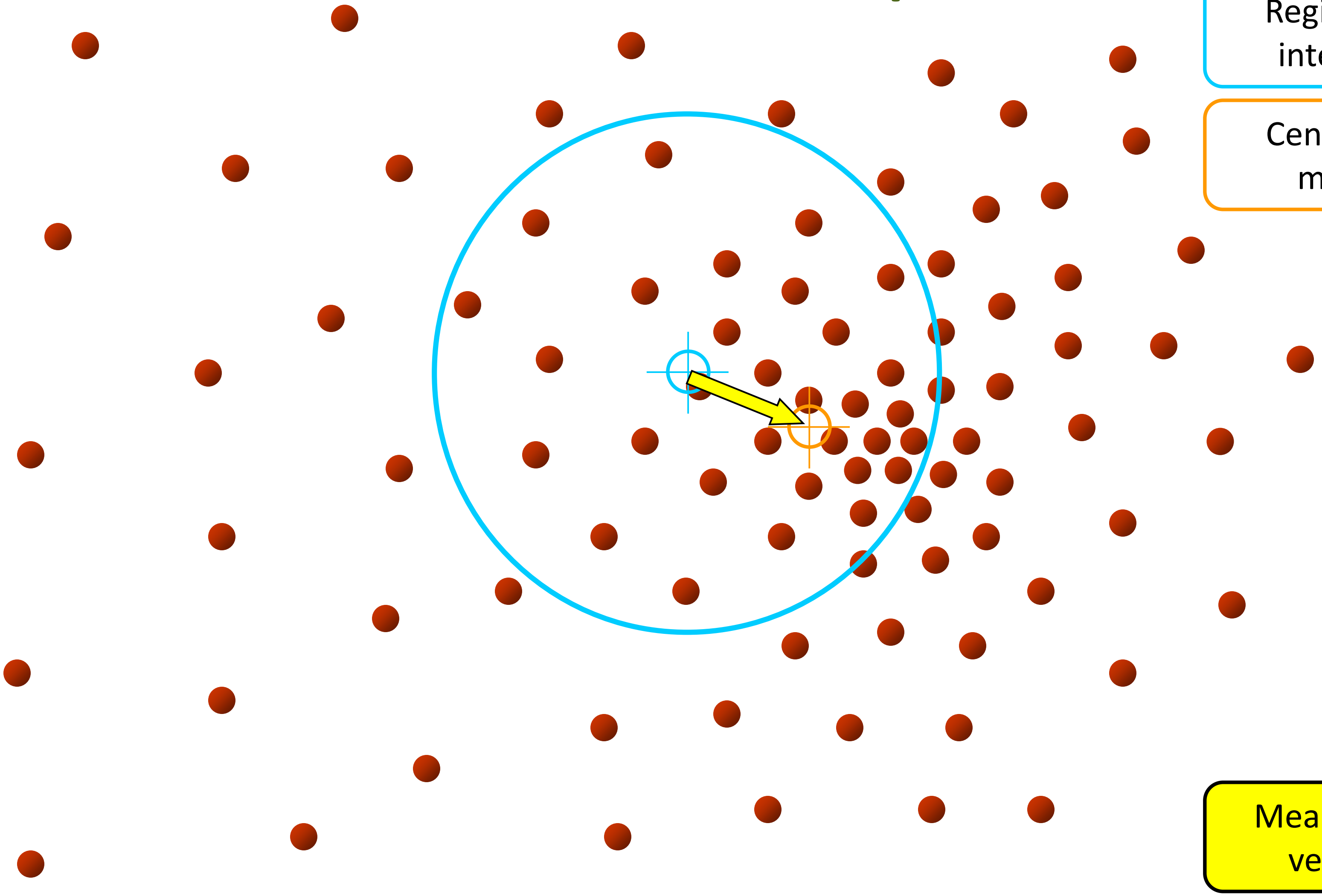
Mean Shift vector

Objective : Find the densest region

Distribution of identical billiard balls

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Intuitive Description



Region of interest

Center of mass

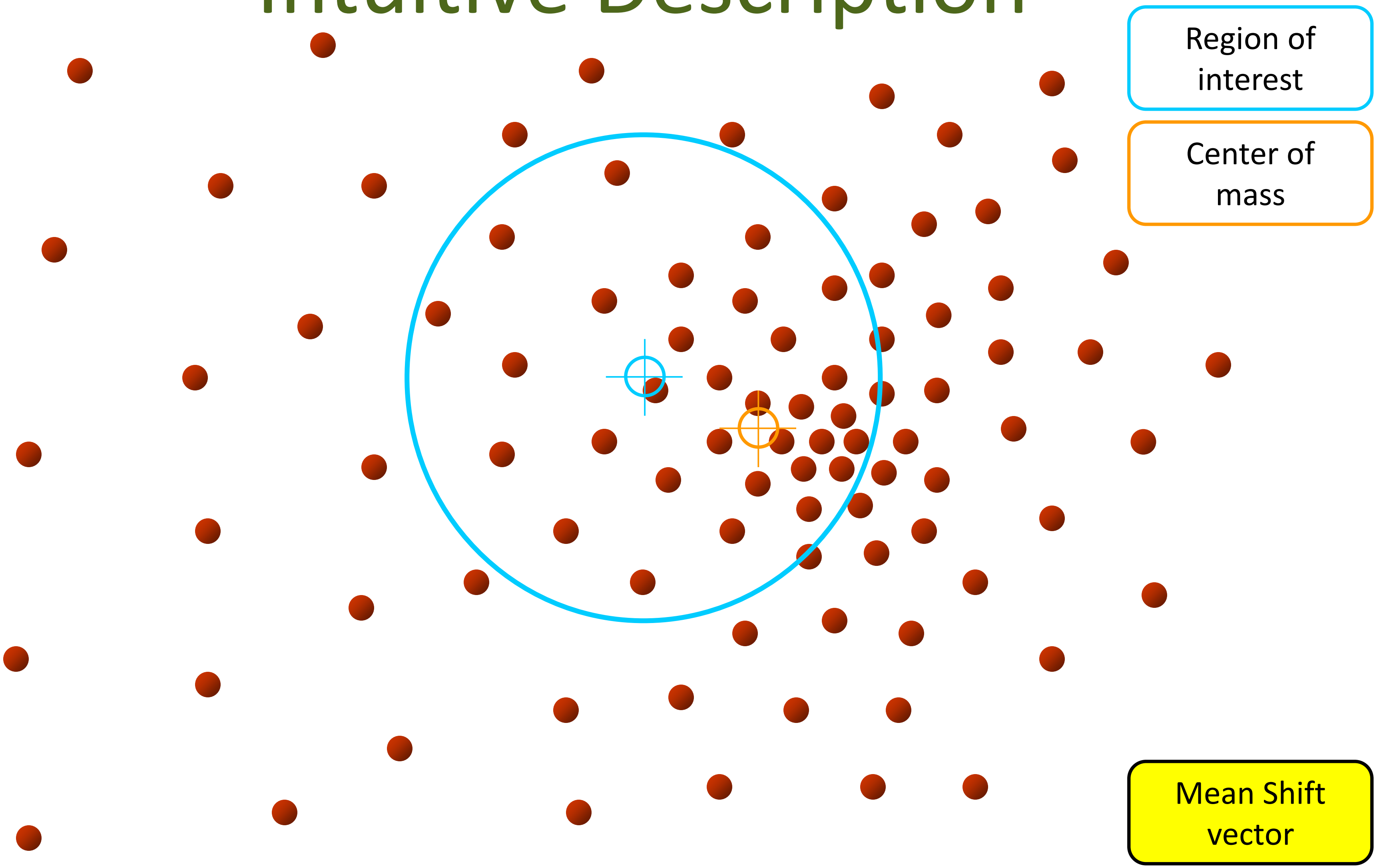
Mean Shift vector

Objective : Find the densest region

Distribution of identical billiard balls

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on "Mean Shift Theory and Applications"

Intuitive Description

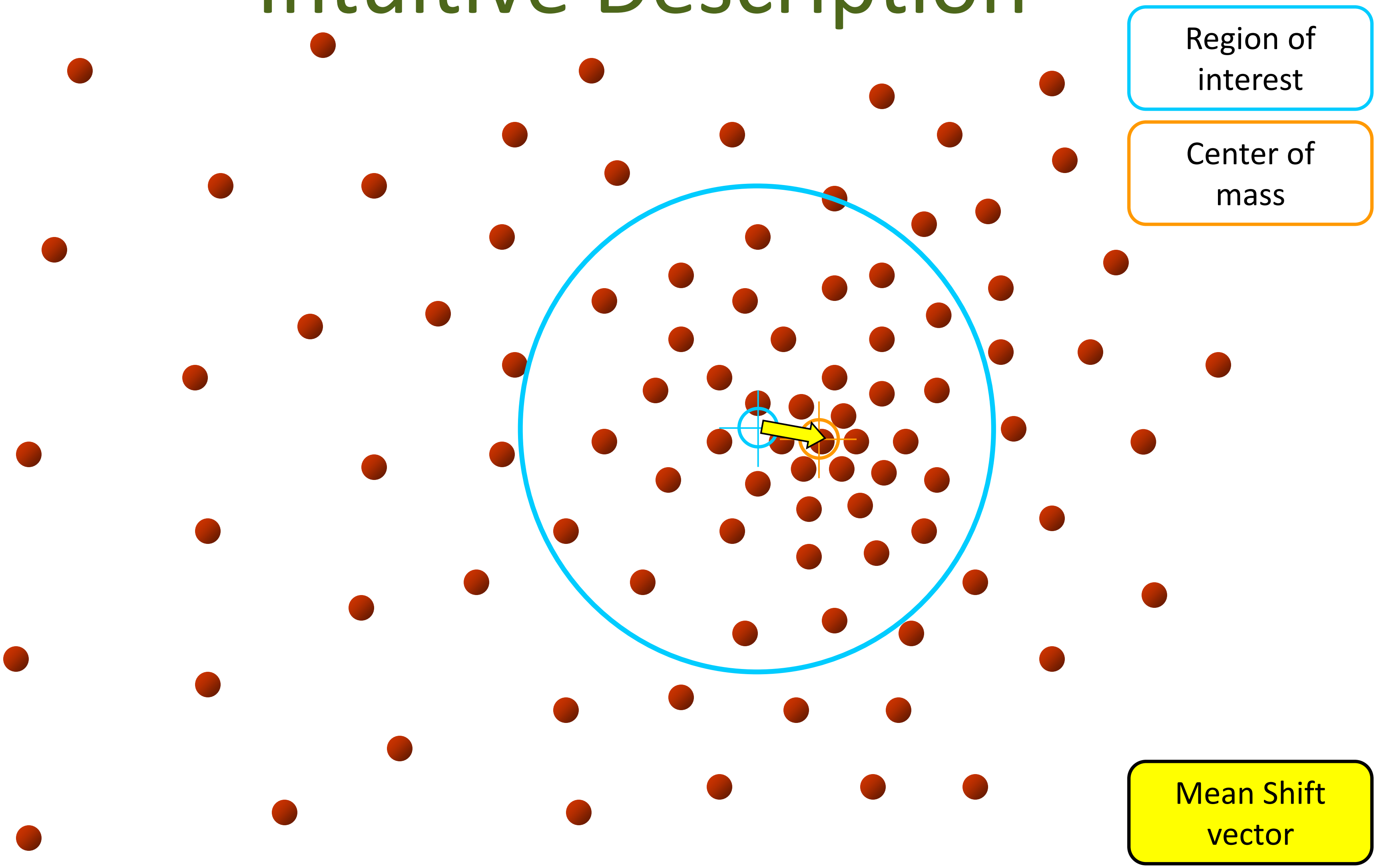


Objective : Find the densest region

Distribution of identical billiard balls

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Intuitive Description



Region of interest

Center of mass

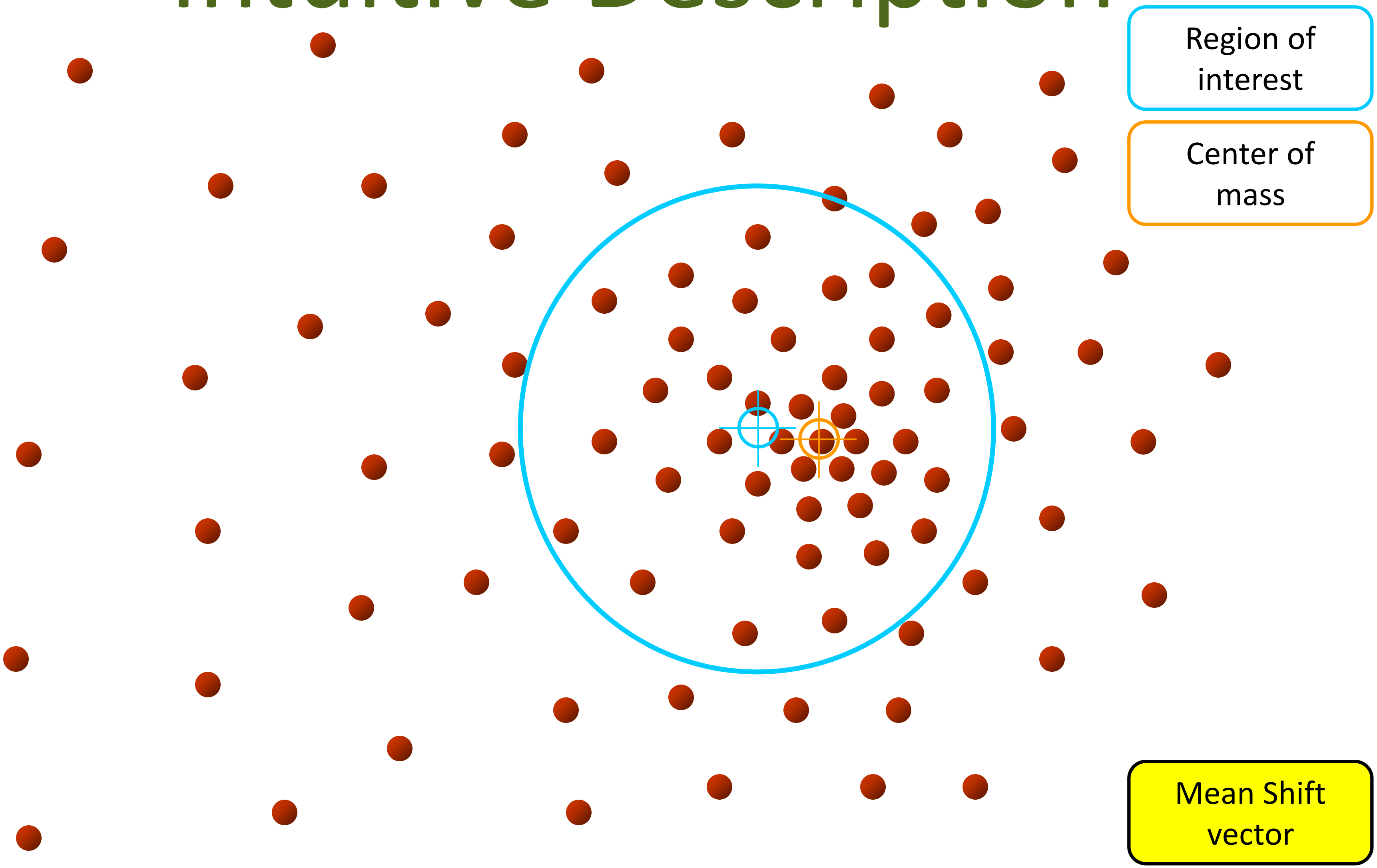
Mean Shift vector

Objective : Find the densest region

Distribution of identical billiard balls

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Intuitive Description

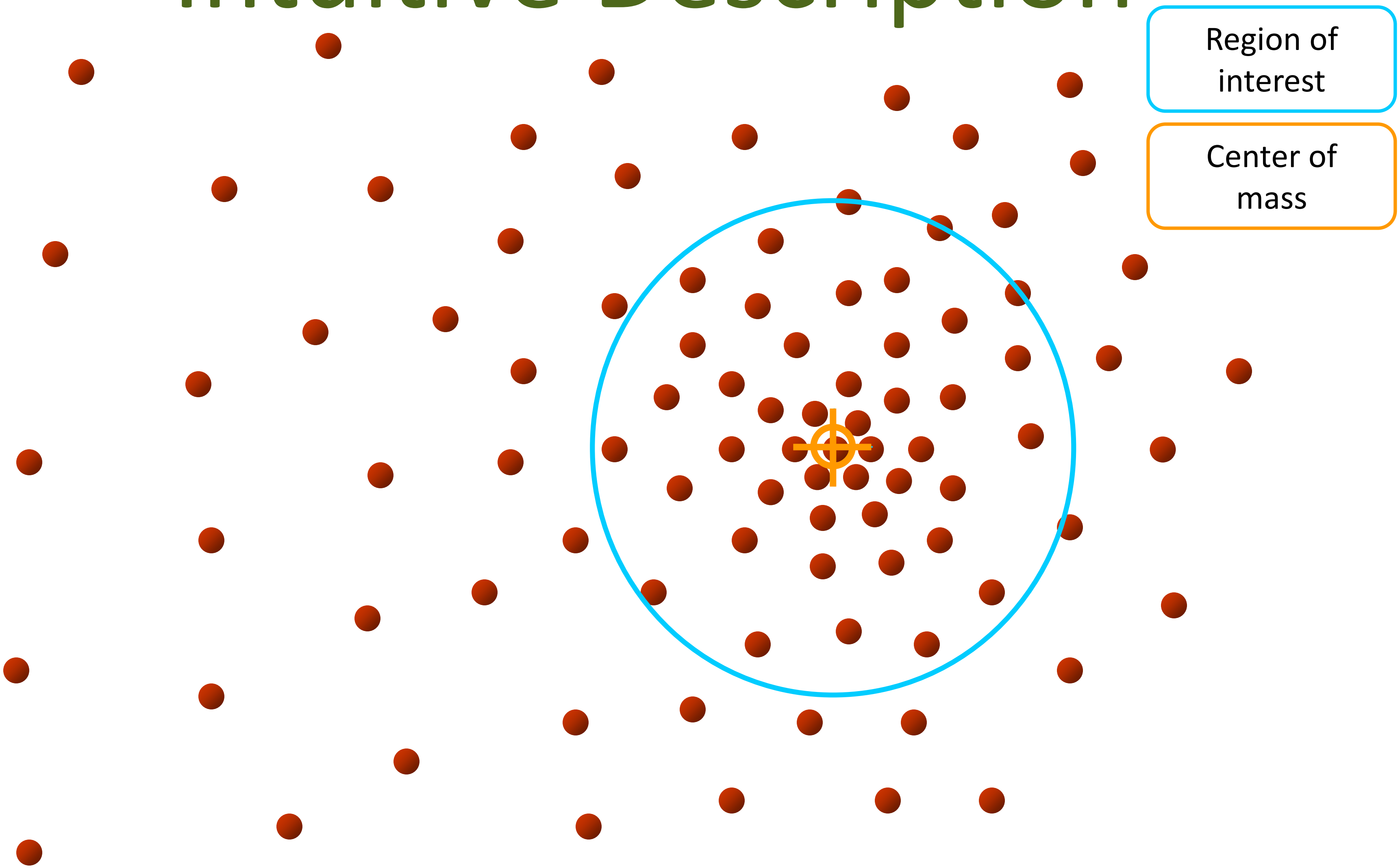


Objective : Find the densest region

Distribution of identical billiard balls

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

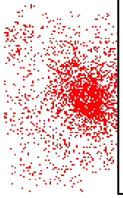
What is Mean Shift ?

A tool for:

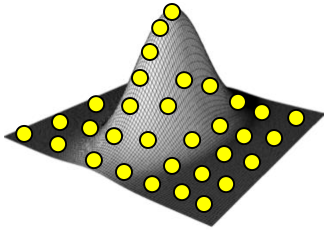
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in R^N

PDF in feature space

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

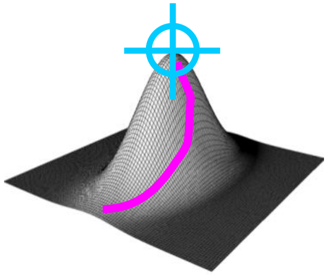


Data



rete PDF Representation

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



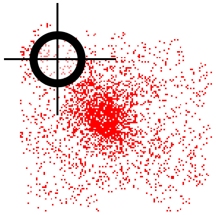
PDF Analysis

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Kernel Density Estimation (Various Kernels)

$$P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x} - \mathbf{x}_i)$$

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

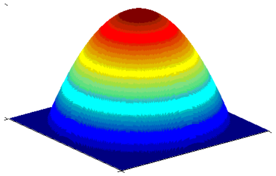


Data

Examples:

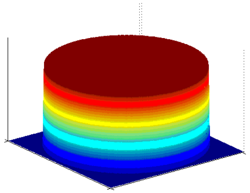
- Epanechnikov Kernel

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



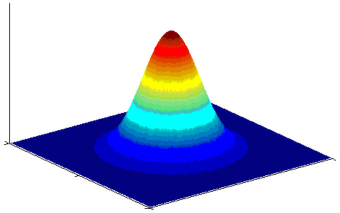
- Uniform Kernel

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



- Normal Kernel

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



Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Kernel Density *Gradient* Estimation

$$\nabla P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla K(\mathbf{x}-\mathbf{x}_i)$$

Give up estimating the PDF !
Estimate **ONLY** the gradient

Using the Kernel form:

$$K(\mathbf{x}-\mathbf{x}_i) = ck\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

Size of window

We get :

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \Box \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

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

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Computing The Mean Shift

Kernel Density *Gradient* Estimation

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n}\nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \Box \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

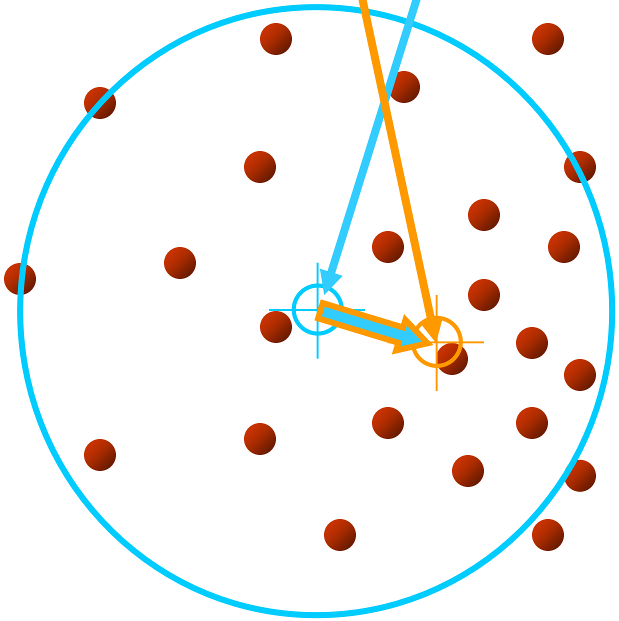
$$\mathrm{g}(\mathbf{x}) = -k'(\mathbf{x})$$

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Computing The Mean Shift

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n}\nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \square \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

Yet another Kernel density estimation !



Simple Mean Shift procedure:

- Compute mean shift vector

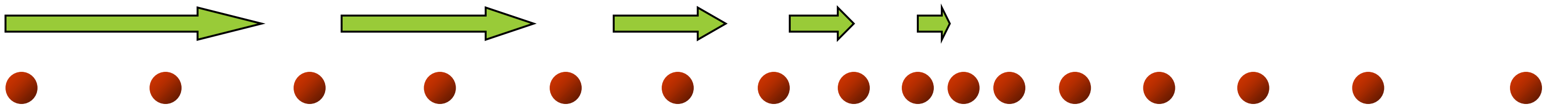
$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\frac{\left\|\mathbf{x}-\mathbf{x}_i\right\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\left\|\mathbf{x}-\mathbf{x}_i\right\|^2}{h}\right)} - \mathbf{x}\right]$$

- Translate the Kernel window by **m(x)**

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

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

Mean Shift Properties

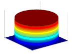


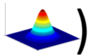
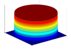
- Automatic convergence speed – the mean shift vector size depends on the gradient itself.

- Near maxima, the steps are small and refined

Adaptive
Gradient
Ascent

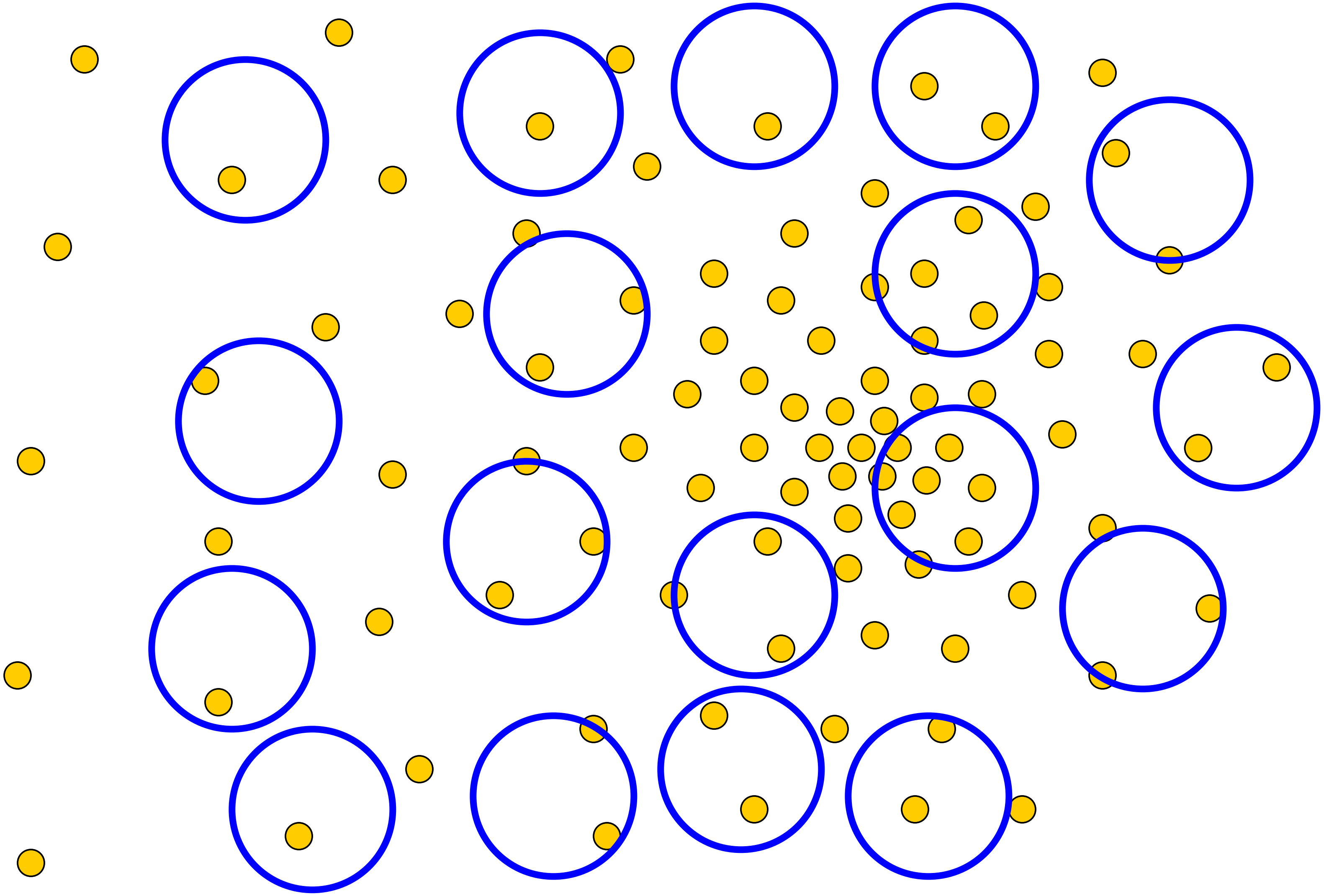
- Convergence is guaranteed for infinitesimal steps only ➔ infinitely convergent, (therefore set a lower bound)

- For Uniform Kernel (), convergence is achieved in a finite number of steps

- Normal Kernel () exhibits a smooth trajectory, but is slower than Uniform Kernel ().

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Real Modality Analysis

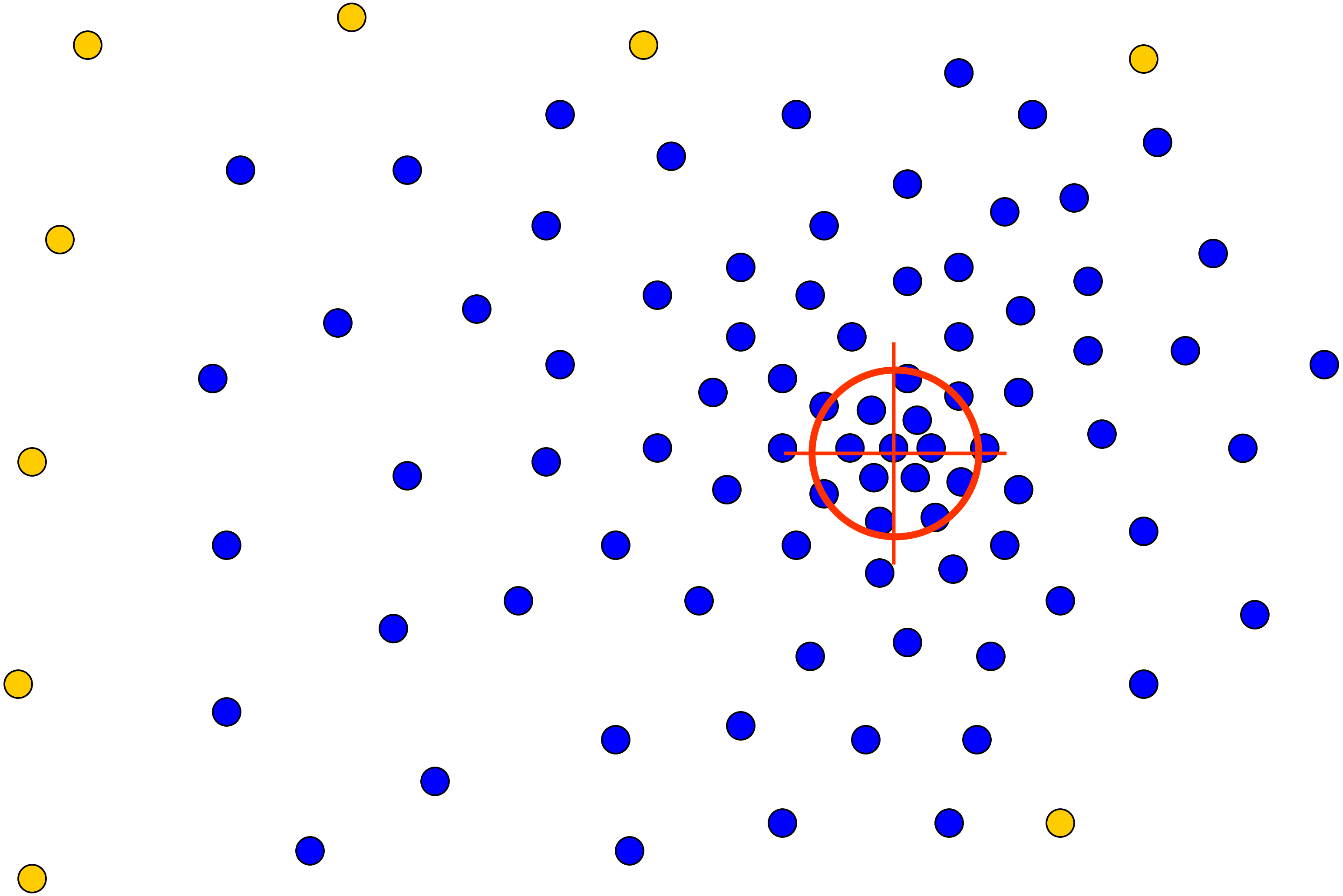


Tessellate the space with windows

Run the procedure in parallel

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Real Modality Analysis



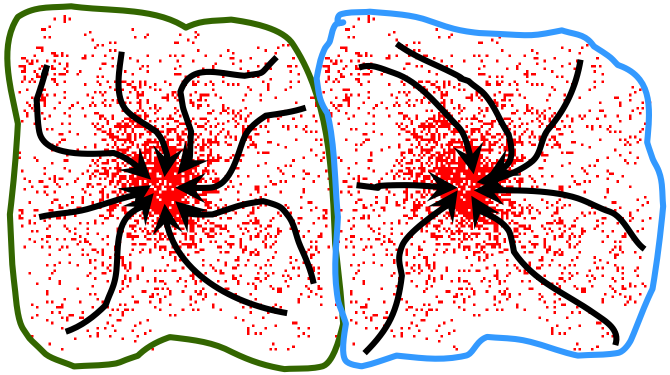
The blue data points were traversed by the windows towards the mode

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

Clustering

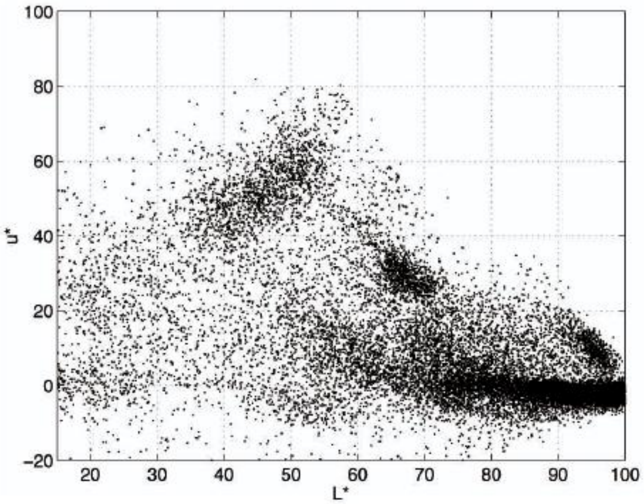
Cluster : All data points in the ***attraction basin*** of a mode

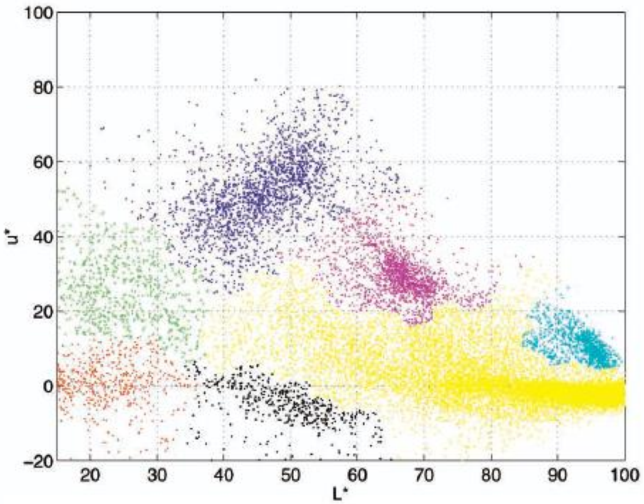
Attraction basin : the region for which all trajectories lead to the same mode

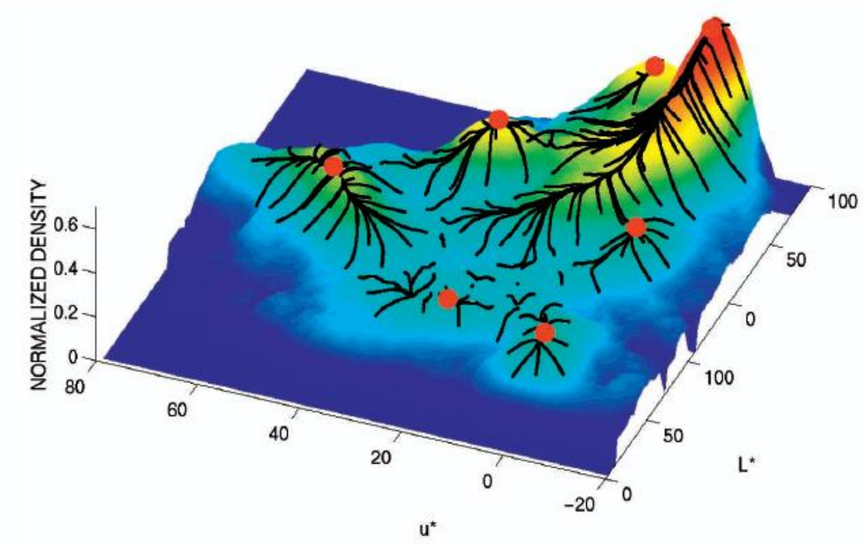


Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

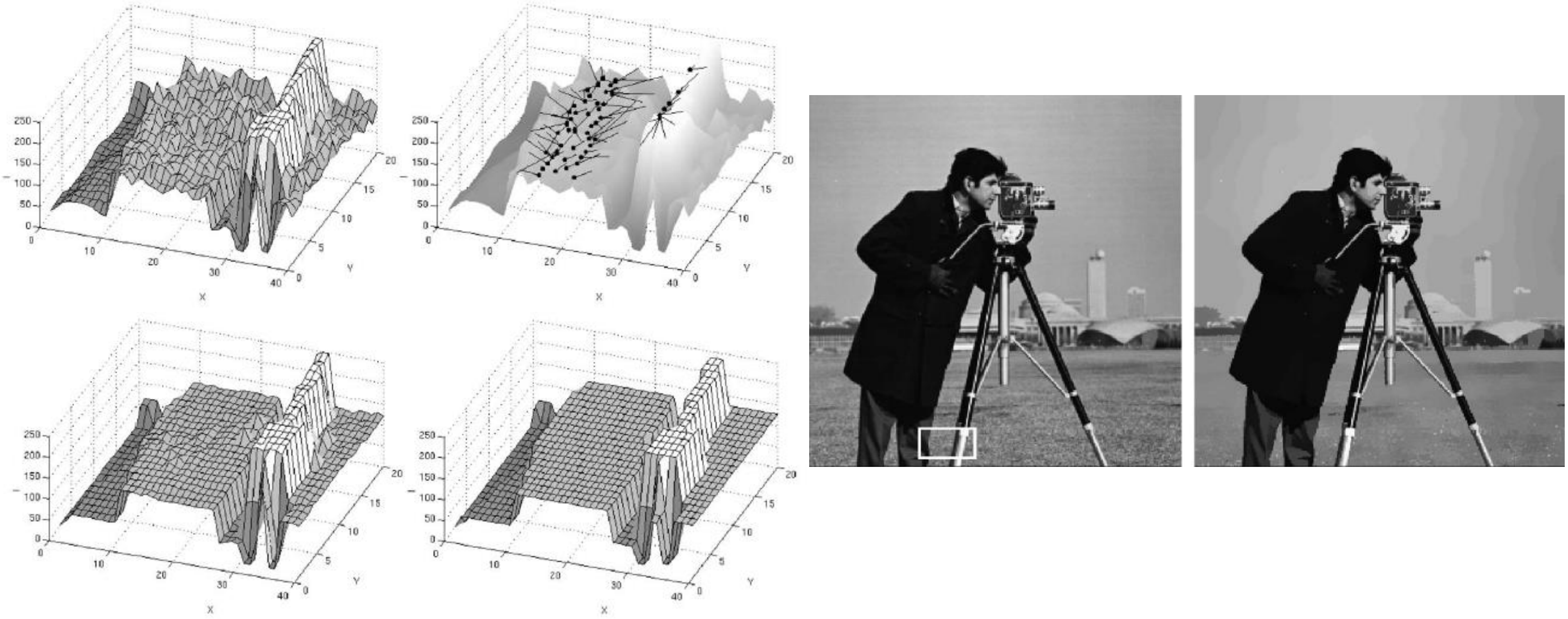
Example of 2D Feature analysis





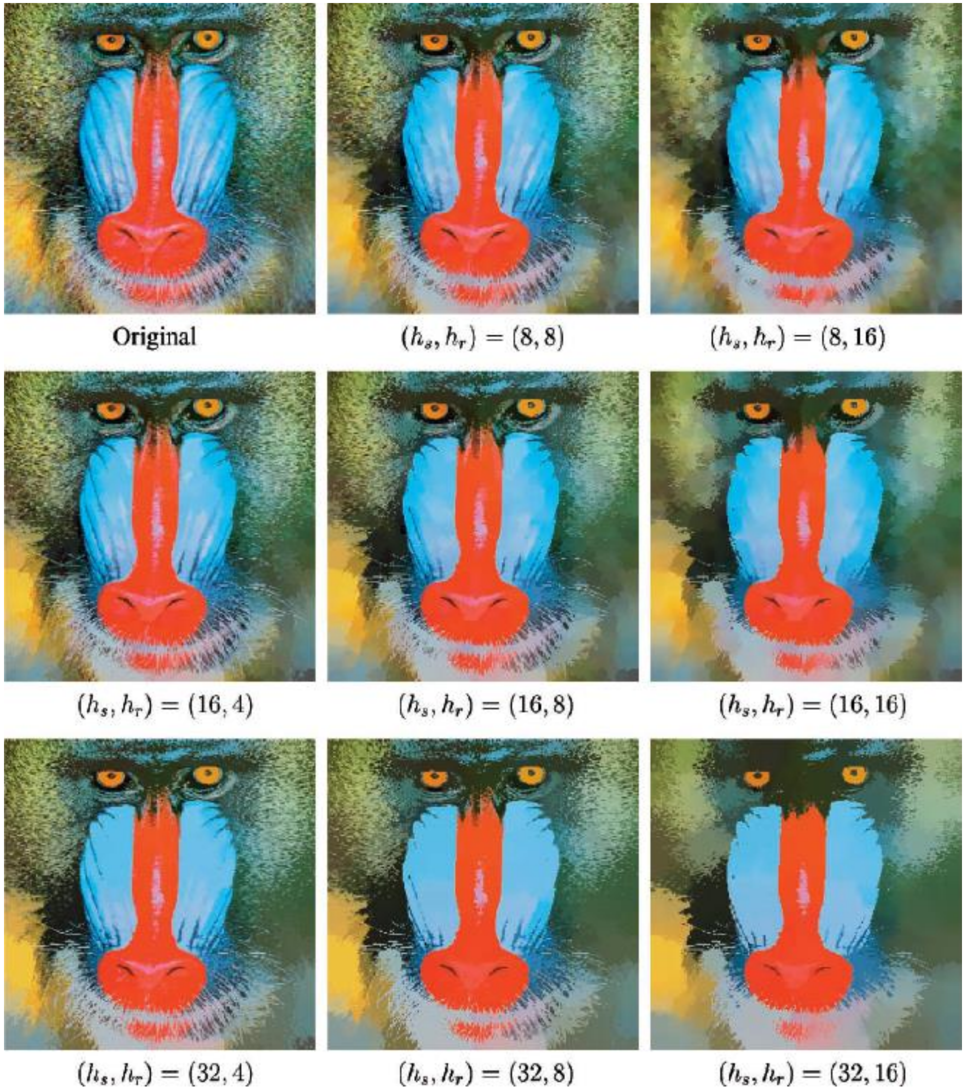


Mean Shift Filtering



Visualizing the mean-shift path, filtering result (h_s, h_r) = (8,4) and the segmentation results

Mean Shift Filtering



Original

$(h_s, h_r) = (8, 8)$

$(h_s, h_r) = (8, 16)$

$(h_s, h_r) = (16, 4)$

$(h_s, h_r) = (16, 8)$

$(h_s, h_r) = (16, 16)$

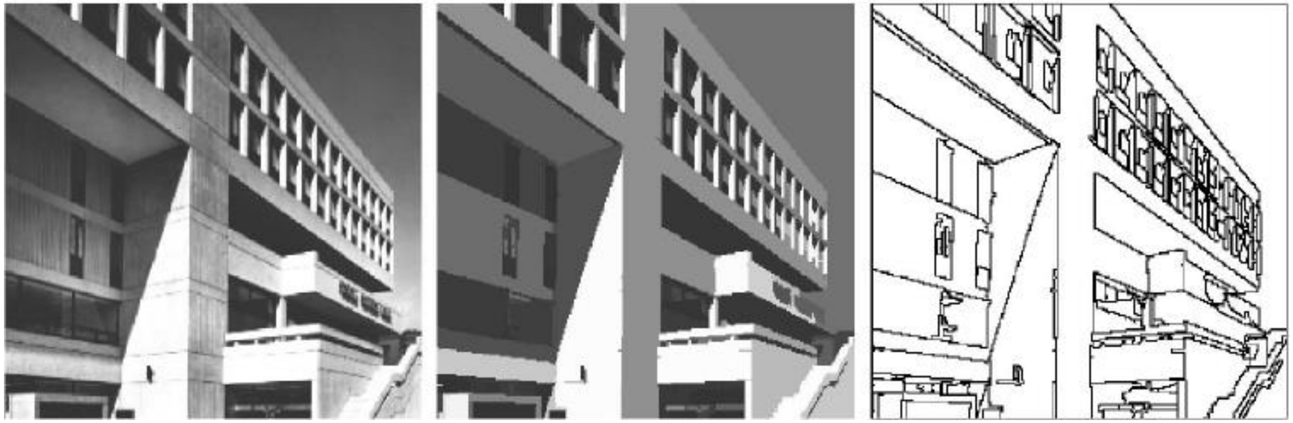
$(h_s, h_r) = (32, 4)$

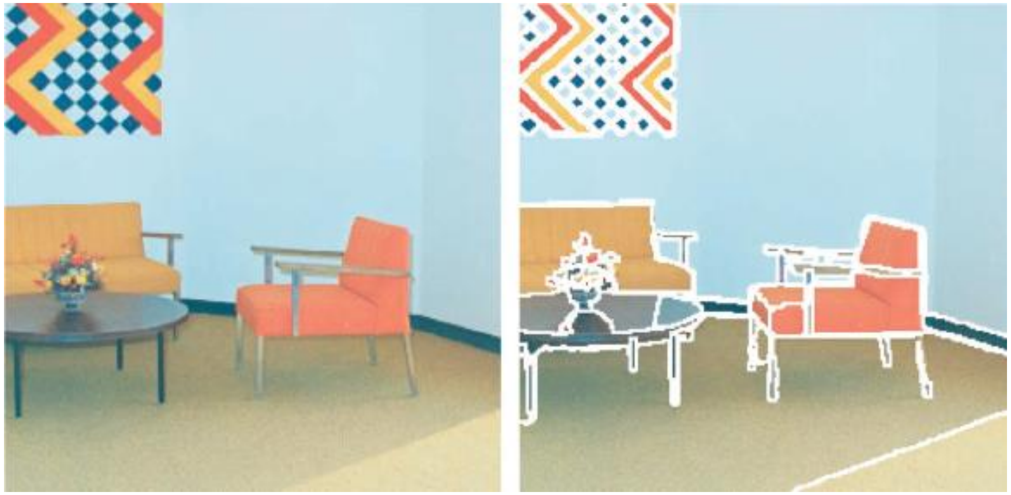
$(h_s, h_r) = (32, 8)$

$(h_s, h_r) = (32, 16)$

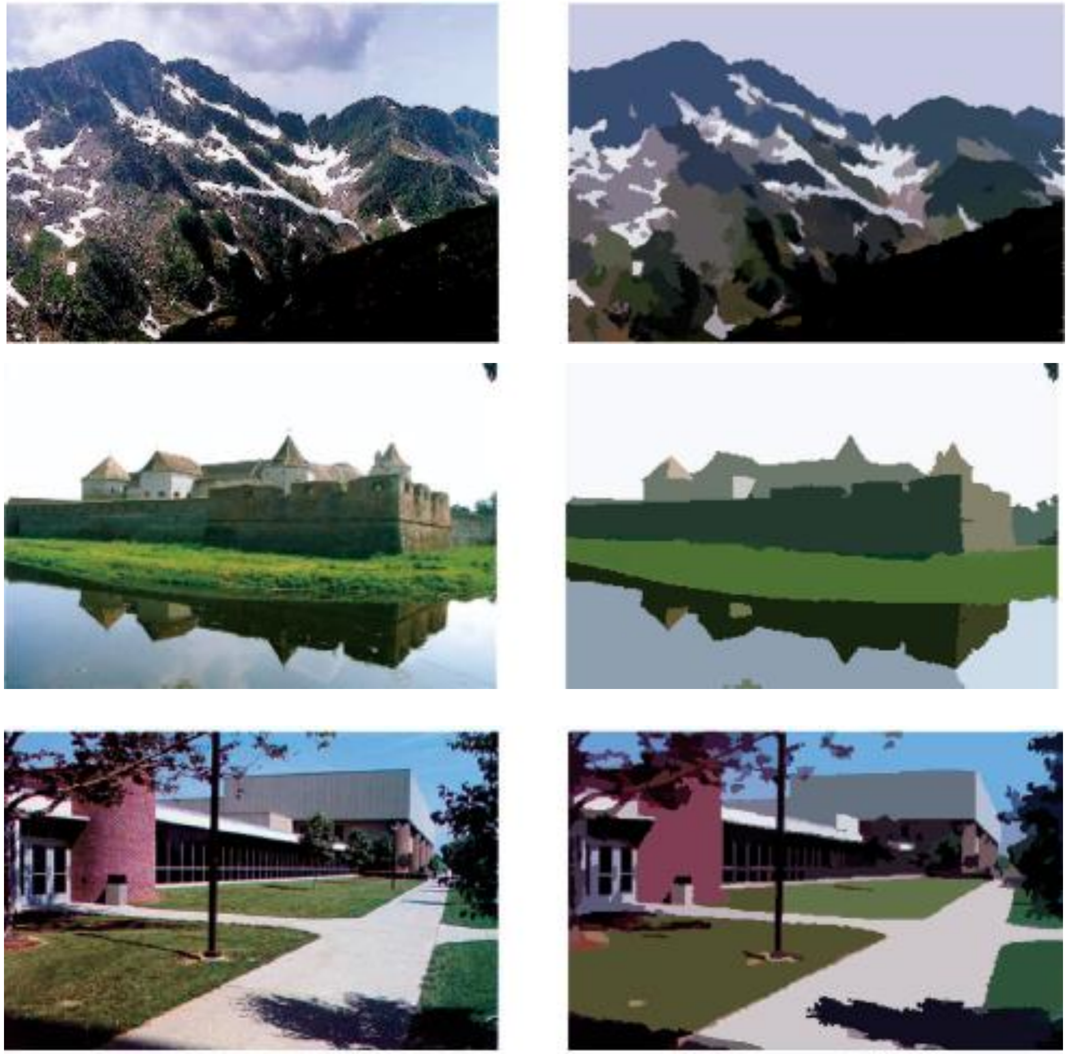
(h_s, h_r) are control the bandwidth of kernel in spatial and range (color).

Mean shift segmentation with boundary

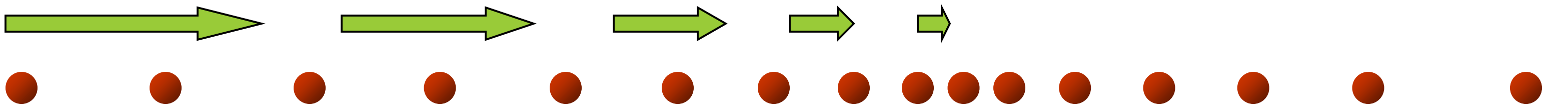




Mean shift segmentation



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

Slides from Y. Ukrainitz & B. Sarel, Lecture notes on “Mean Shift Theory and Applications”

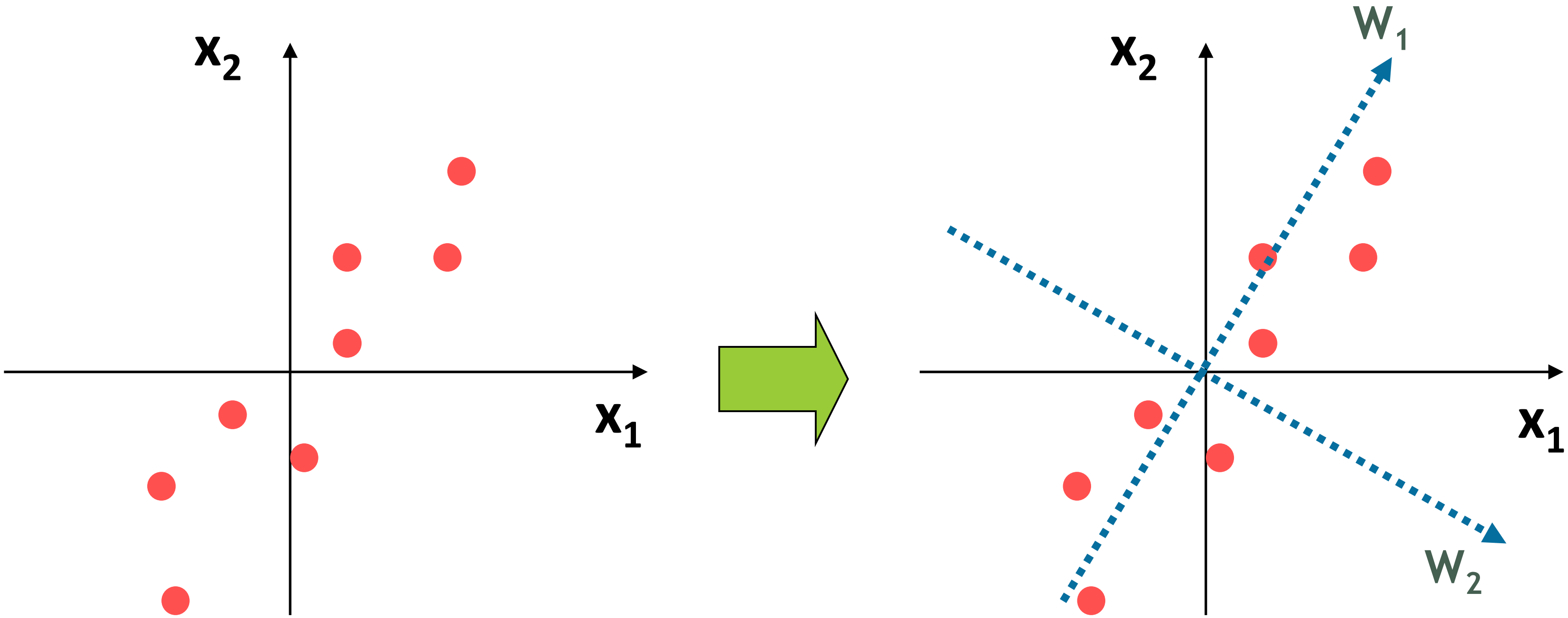
Principal Component Analysis

- Principal component analysis (PCA) is a technique for compression and classification of data.

- Reducing the dimensionality of a data set (sample).
  - by finding a new set of variables, smaller than the original set of variables.
  - Retaining most of a sample's information.

- The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains

PCA



X_1 axis : $(1, 0)^t$

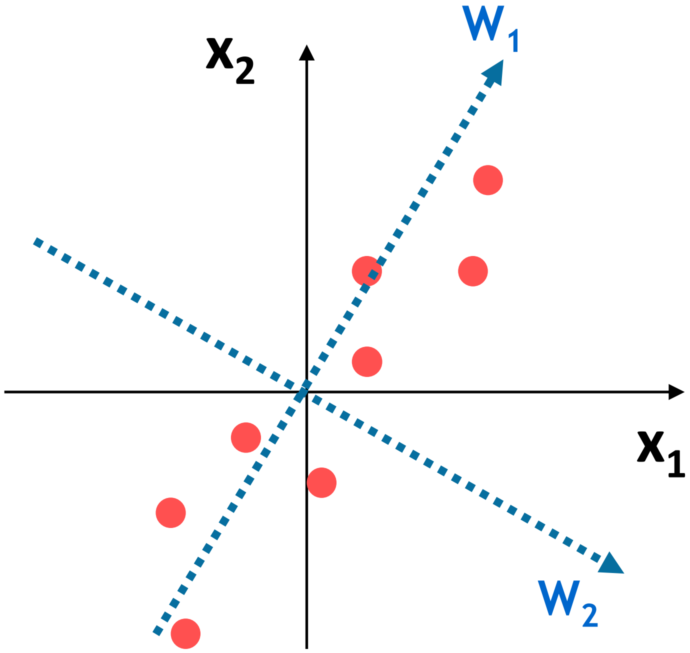
X_2 axis : $(0, 1)^t$

W_1 axis : $(0.5, 0.866)^t$

W_2 axis : $(0,866, -0.5)^t$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} = 1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 2 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 2.2321 \cdot \begin{bmatrix} 1/2 \\ \sqrt{3}/2 \end{bmatrix} + (-0.134) \cdot \begin{bmatrix} \sqrt{3}/2 \\ -1/2 \end{bmatrix}$$

Principal Components



W_1 axis : $(0.5, 0.866)^t$

W_2 axis : $(0,866, -0.5)^t$

$$2.2321 \cdot \begin{bmatrix} 1/2 \\ \sqrt{3}/2 \end{bmatrix} + (-0.134) \cdot \begin{bmatrix} \sqrt{3}/2 \\ -1/2 \end{bmatrix}$$

$$= \begin{bmatrix} 1.116 \\ 1.933 \end{bmatrix} + \begin{bmatrix} -0.116 \\ 0.067 \end{bmatrix}$$

Projection on W_1

Projection on W_2

- The 1st PC W_1 is a minimum ? in X space
- The 2nd PC W_2 is a minimum ? in the plane perpendicular to the 1st PC

Principal Components (cont.)

- PCs are a series of linear least squares fits to samples
  - each PC is orthogonal to all the previous.

- Given a data set of zero mean:

$$w_1 = \arg\max_{\mathrm{w}} \ \mathrm{E}\left((w^T x)^2\right), \text{ where } |w| = 1$$

Max var(z_1)

$$\hat{x}_{k-1} = x - \sum_{i=1}^{k-1} (w_i^T x) w_i$$

Residual of the first k-1 components

$$w_k = \arg\max_{\mathrm{w}} \ \mathrm{E}\left((w^T \hat{x}_{k-1})^2\right), \text{ where } |w| = 1$$

PCA (cont.)

Given a sample of *n* observations on a vector of *p* variables

$$X = (x_1,\ x_2,\ \ldots\ldots x_p)$$

The 1st principal component

$$z_1 = w_1^T X = \sum_{i=1}^{p} w_{1i} x_i$$

w_1 is chosen such that var[z_1] is maximum (for all samples)

$$w_1^T w_1 = 1$$

PCA (cont.)

Likewise, define the k^{th} PC of the sample by

$$z_k = w_k^T X = \sum_{i=1}^{p} w_{ki} x_i$$

w_k is chosen such that var[z_k] is maximum

subject to $\operatorname{cov}[z_k, z_l] = 0, \text{ for } k > l \geq 1$

and to $w_k^T w_k = 1$

PCA (cont.)

$$\mathrm{var}[z_1] = E\left(z_1^2\right) - \left(E(z_1)\right)^2$$

$$= \frac{1}{n}\sum_{k=1}^{n}\left(\sum_{j=1}^{p} w_{1j}x_j^{(k)}\right)\left(\sum_{i=1}^{p} w_{1i}x_i^{(k)}\right) - \left(\frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{p} w_{1j}x_j^{(k)}\right)^2$$

$$= \sum_{j=1}^{p}\sum_{i=1}^{p} w_{1j}w_{1i}E\left(x_i x_j\right) - \sum_{j=1}^{p}\sum_{i=1}^{p} w_{1j}w_{1i}E(x_i)E\left(x_j\right)$$

$$= \sum_{j=1}^{p}\sum_{i=1}^{p} w_{1j}w_{1i}S_{ij}$$

$$= w_1^T S w_1$$

S is the covariance matrix for the samples

$$S_{ij} = \mathrm{cov}(x_i, x_j) = E\left((x_i - \mu_i)(x_j - \mu_j)\right)$$

PCA (cont.)

To find w_1 that maximize var[z_1] , subject to |w_1|=1

Let λ be a Lagrange multiplier

then maximize $$w_1^T S w_1 - \lambda\left(w^T w - 1\right)$$

the differentiation should be 0 $$S w_1 - \lambda w_1 = 0$$

Therefore, w_1 is an eigenvector of S

PCA (cont.)

Since we want to maximize

$$\operatorname{var}\left[z_1\right] = w_1^T S w_1$$
$$= w_1^T \lambda_1 w_1 = \lambda_1 w_1^T w_1 = \lambda_1$$

So λ_1 is the largest eigenvalue of S

w_1 is the correspondent eigenvector

PCA (cont.)

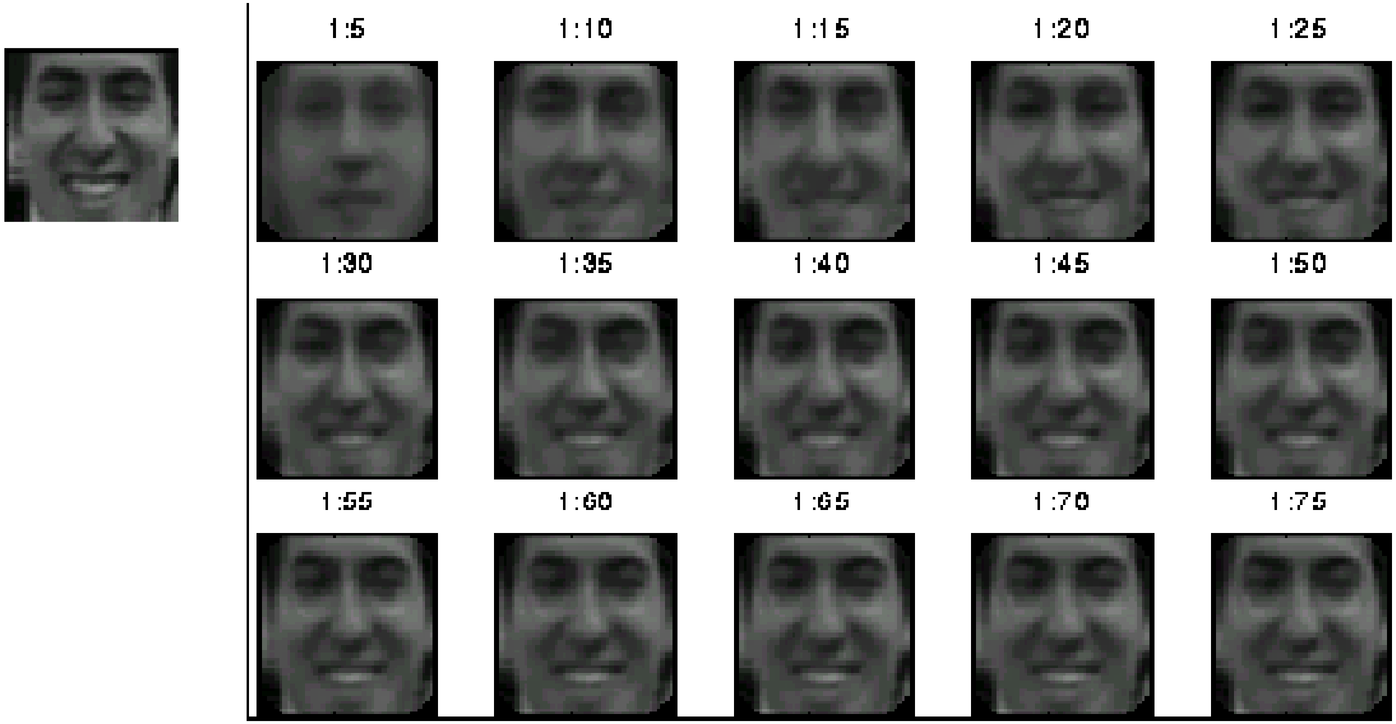
- From similar deduction, Kth PC is the eigenvector corresponding to the Kth largest eigenvalue.
- The kth largest eigenvalue of S is the variance of the kth PC.
- The kth PC retains the kth greatest fraction of the variation in the sample.

Calculating PCA

1. Calculate the mean.
2. Subtract the mean.
3. Calculate the covariance matrix S.
4. Calculate the eigenvalues and eigenvectors S.
5. Choose the components.
6. Derive the new data set.

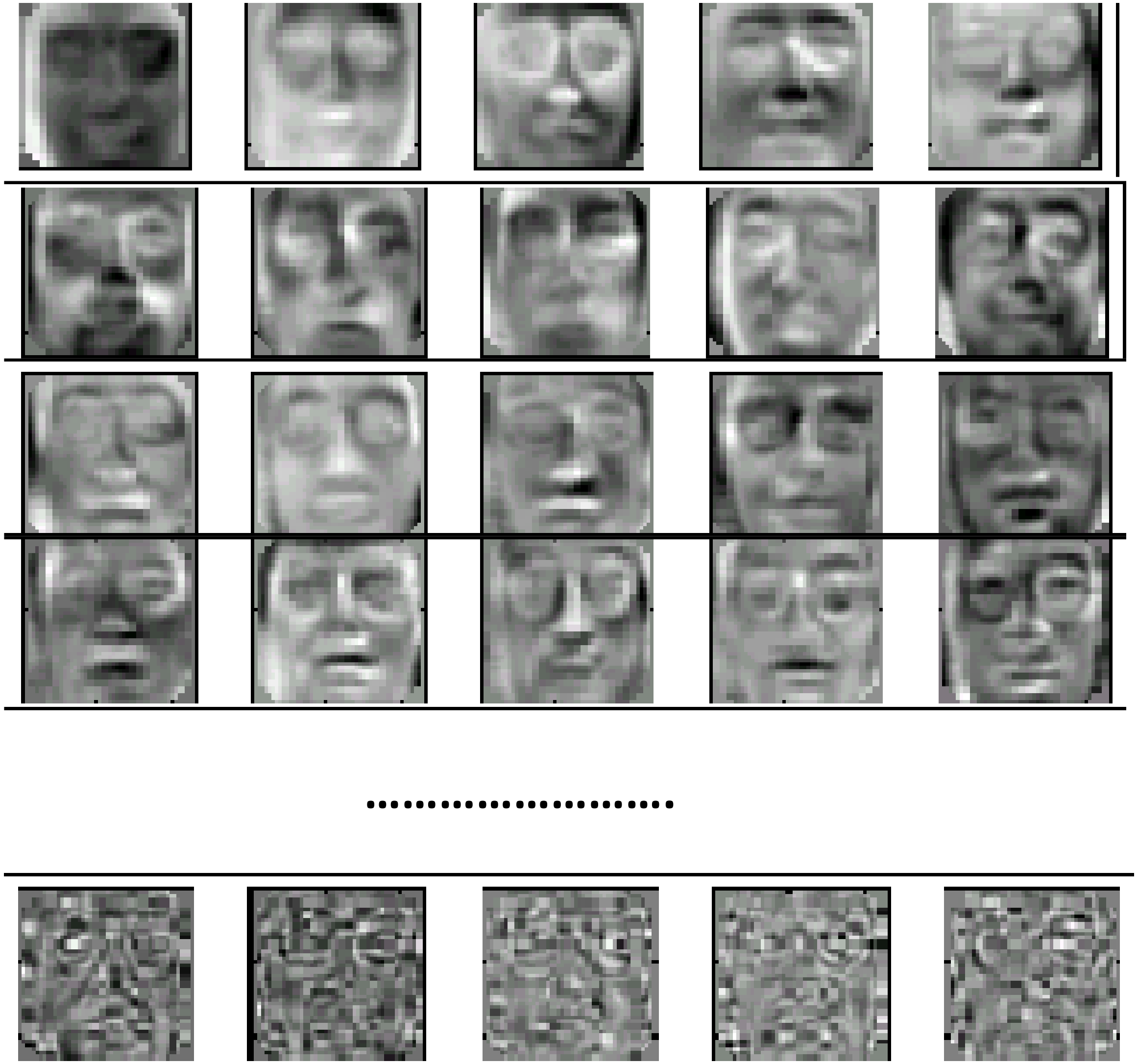
Applications of PCA

- Eigenfaces



http://www.stat.ucla.edu/~dinov/

Applications of PCA



Principal Components

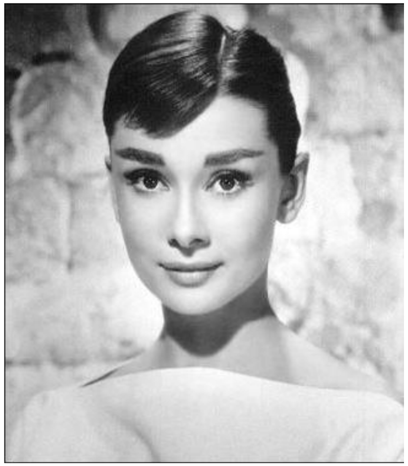
Applications of PCA

- Data compression
  - Keep “important” information

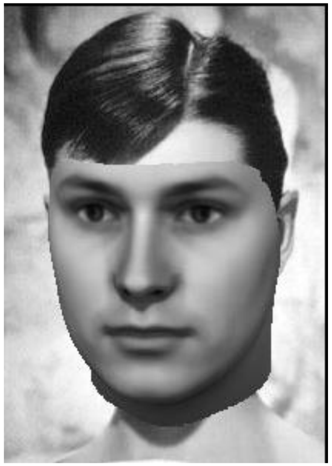
- Data analysis
  - Reduce dimensions for classification or recognition.

- There’re efficient algorithms of PCA (in memory and computation)

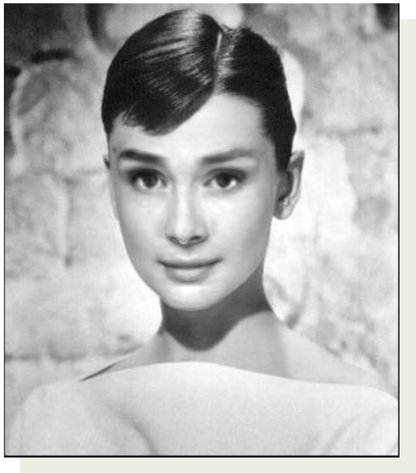
Applications of PCA



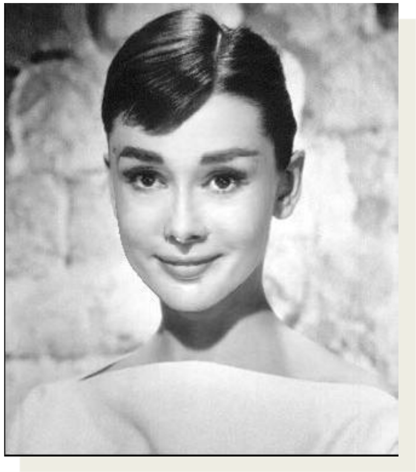
Input images

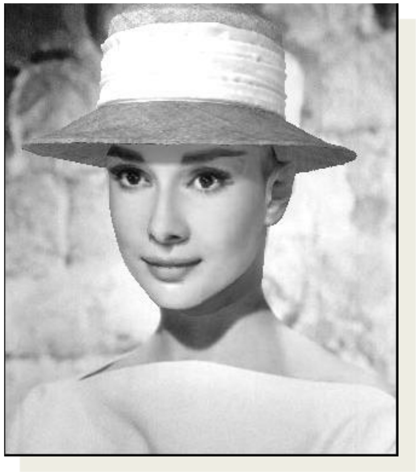


Initialization



3D reconstruction with texture



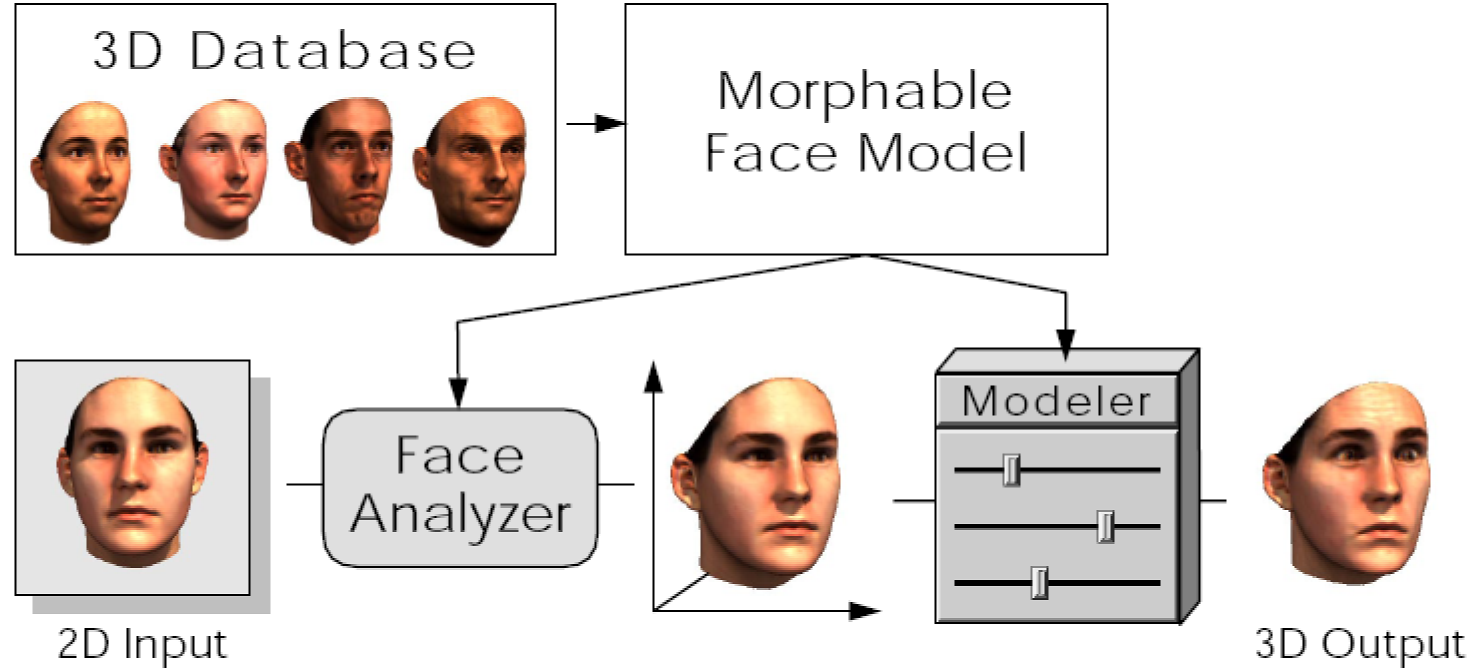


V. Blanz, T. Vetter, “A Morphable Model For The Synthesis Of 3D Faces”, Proc. SIGGRAPH’99, pp. 187-194.

Appendix: Morphable Model

- A basis-based data analysis and interpolation.

$$S_{model} = \overline{S} + \sum_{i=1}^{m-1} \alpha_i s_i \ , \quad T_{model} = \overline{T} + \sum_{i=1}^{m-1} \beta_i t_i$$



Appendix: Fitting the Model to an Image

- Coefficients of the 3D model $(a_1, a_2, \ldots, a_m)^T$ and $(b_1, b_2, \ldots, b_m)^T$ are optimized together with the rendering parameters $\boldsymbol{\rho}$.

- Min

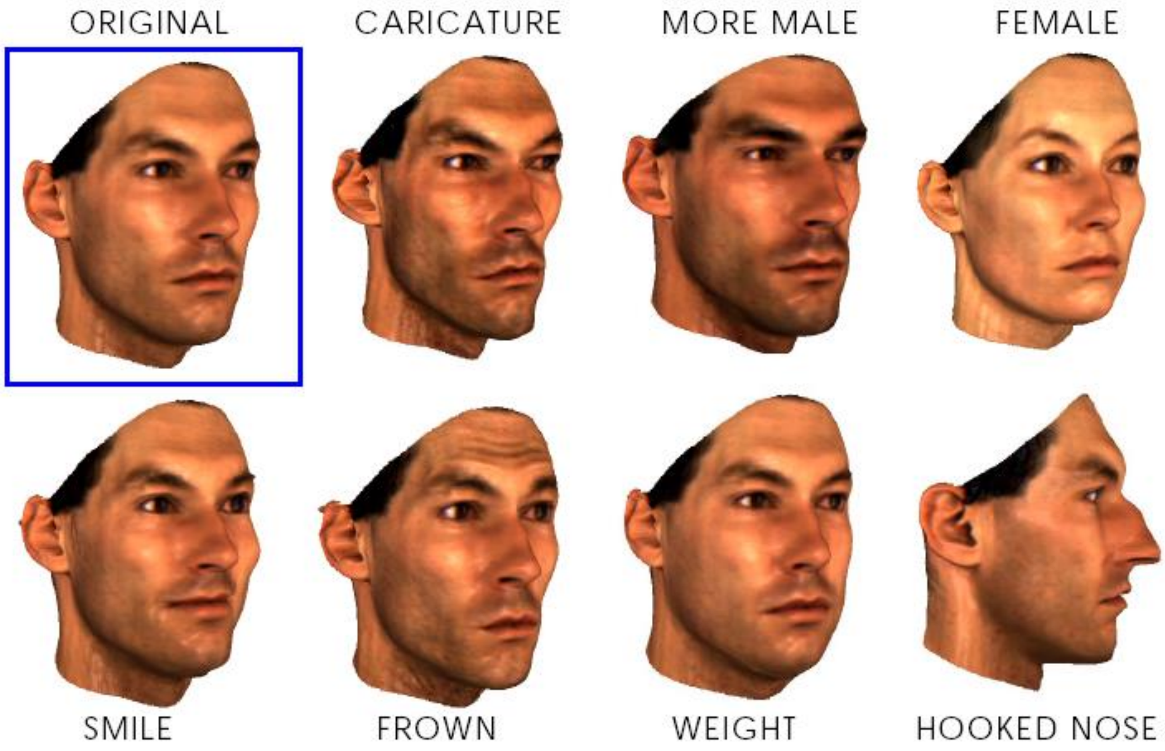
$$E = \frac{E_I}{2\sigma_N^2} + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\rho_i - \overline{\rho_i})_i^2}{\sigma_{\rho,i}^2}$$

$$E_I = \sum_{x,y} \| \mathbf{I}_{input}(x, y) - \mathbf{I}_{model}(x, y) \|^2$$

- A coarse-to-fine strategy is employed.
  - The first iterations are performed on a sub-sampled input and a low resolution model.
  - The highest principal components are used at first, and more are added later on.

Appendix: Morphable Model

- How to map the semantic attributes to basis / components?


ORIGINAL
CARICATURE
MORE MALE
FEMALE
SMILE
FROWN
WEIGHT
HOOKED NOSE