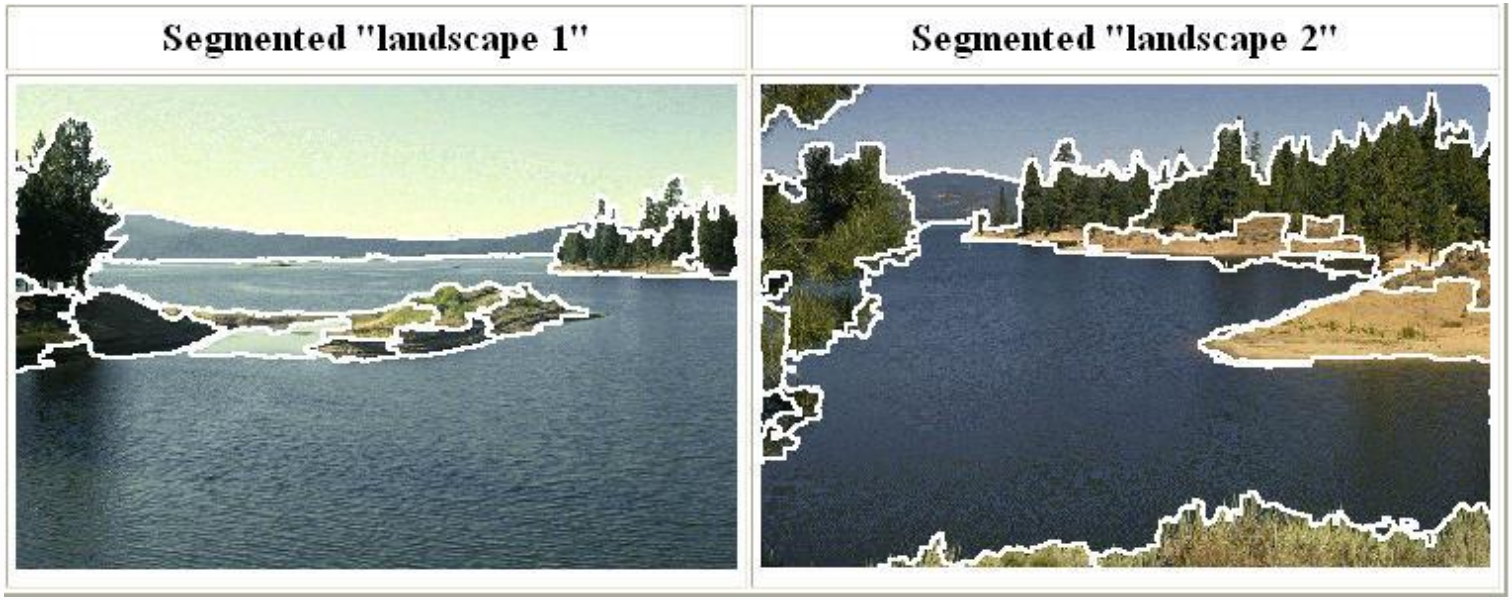
Mean shift segmentation

- Versatile technique for clustering-based segmentation



D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, May 2002

Mean shift clustering

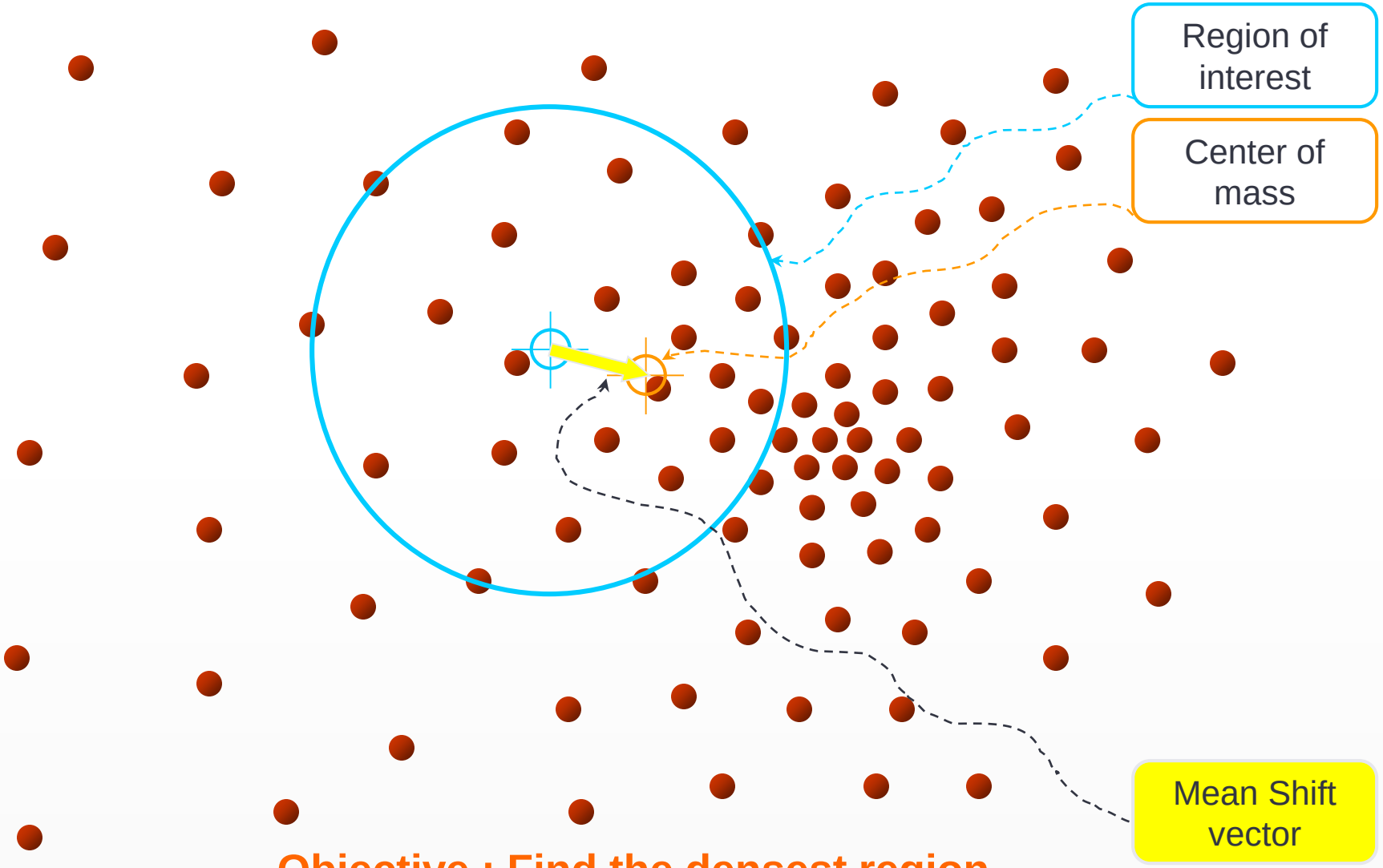
- Sources: Mean Shift Theory and Applications, presentation of Yaron Ukrainitz & Bernard Sarel
- Further credits:
  - Alper Yilmaz, Afshin Dehghan
  - Lecture of Mubarak Shah, UCF FL, USA
    - https://www.youtube.com/watch?v=M8B3RZVqgOo

Origin: Mean-Shift Clustering

- Non-parametric iterative clustering technique introduced in 1975 by Fukunaga and Hostetler.
- Do not need to know the number of clusters a priori.
- Does not constrain the shape of the cluster.
- Mean shift considers the points in the feature space as samples from an underlying probability density function.
- The objective of the algorithm is to find the modes of this PDF, and associate each point with the node it is „attracted to”.

Fukunaga and Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition", IEEE Transactions on Information Theory vol 21 , pp 32-40 ,1975

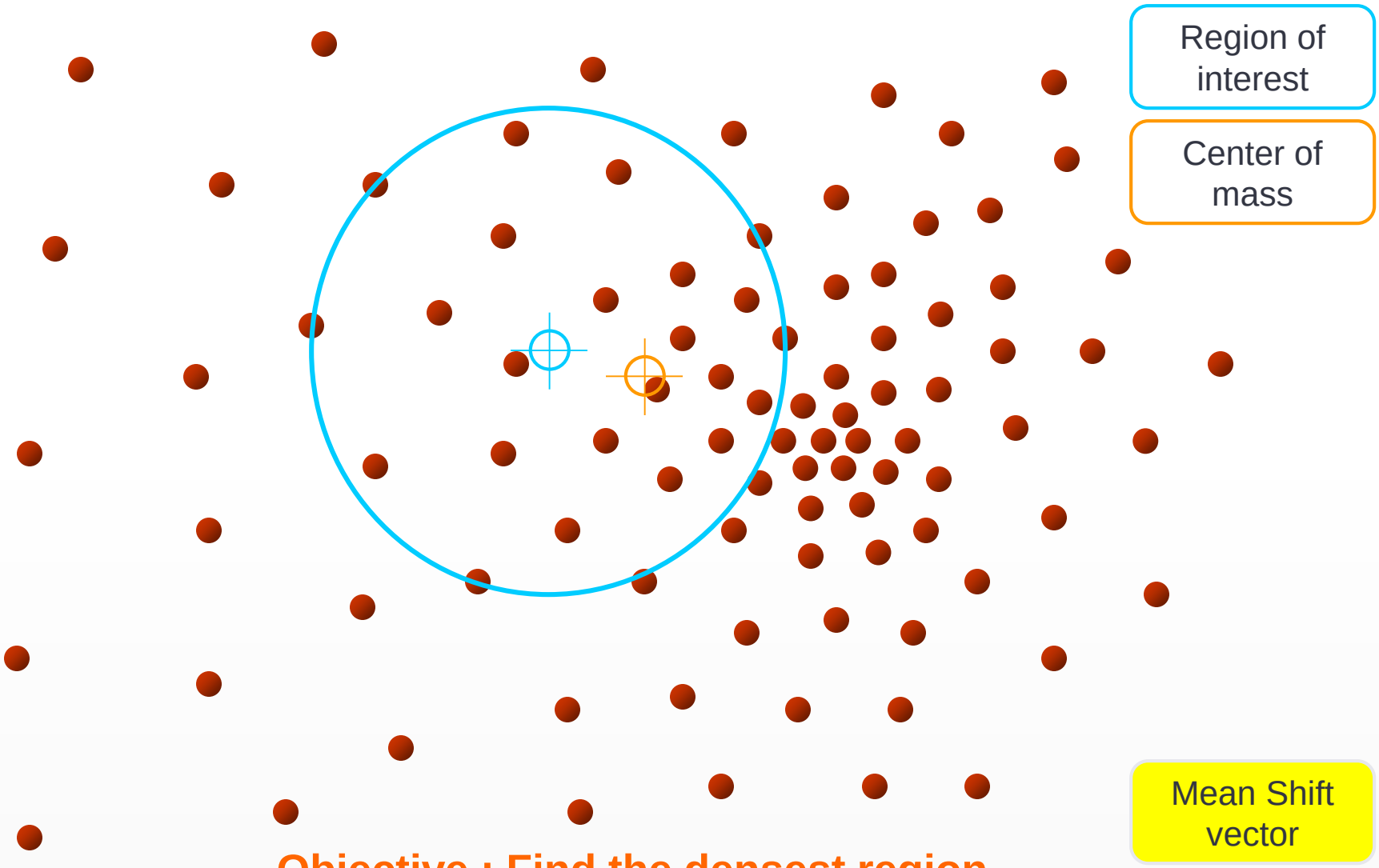
Mean shift – intuitive description



Objective : Find the densest region

Distribution of identical billiard balls

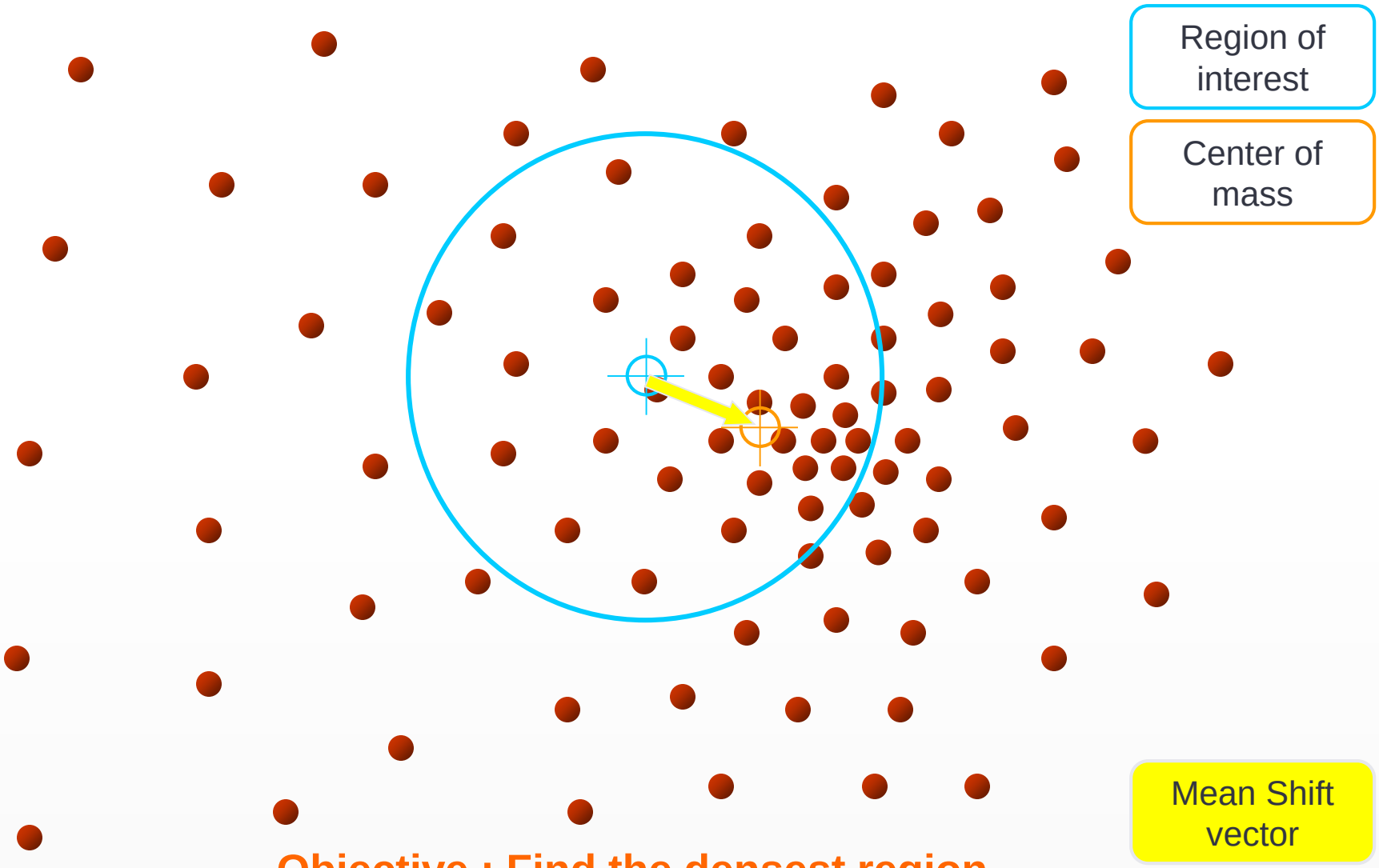
Mean shift – intuitive description



Objective : Find the densest region

Distribution of identical billiard balls

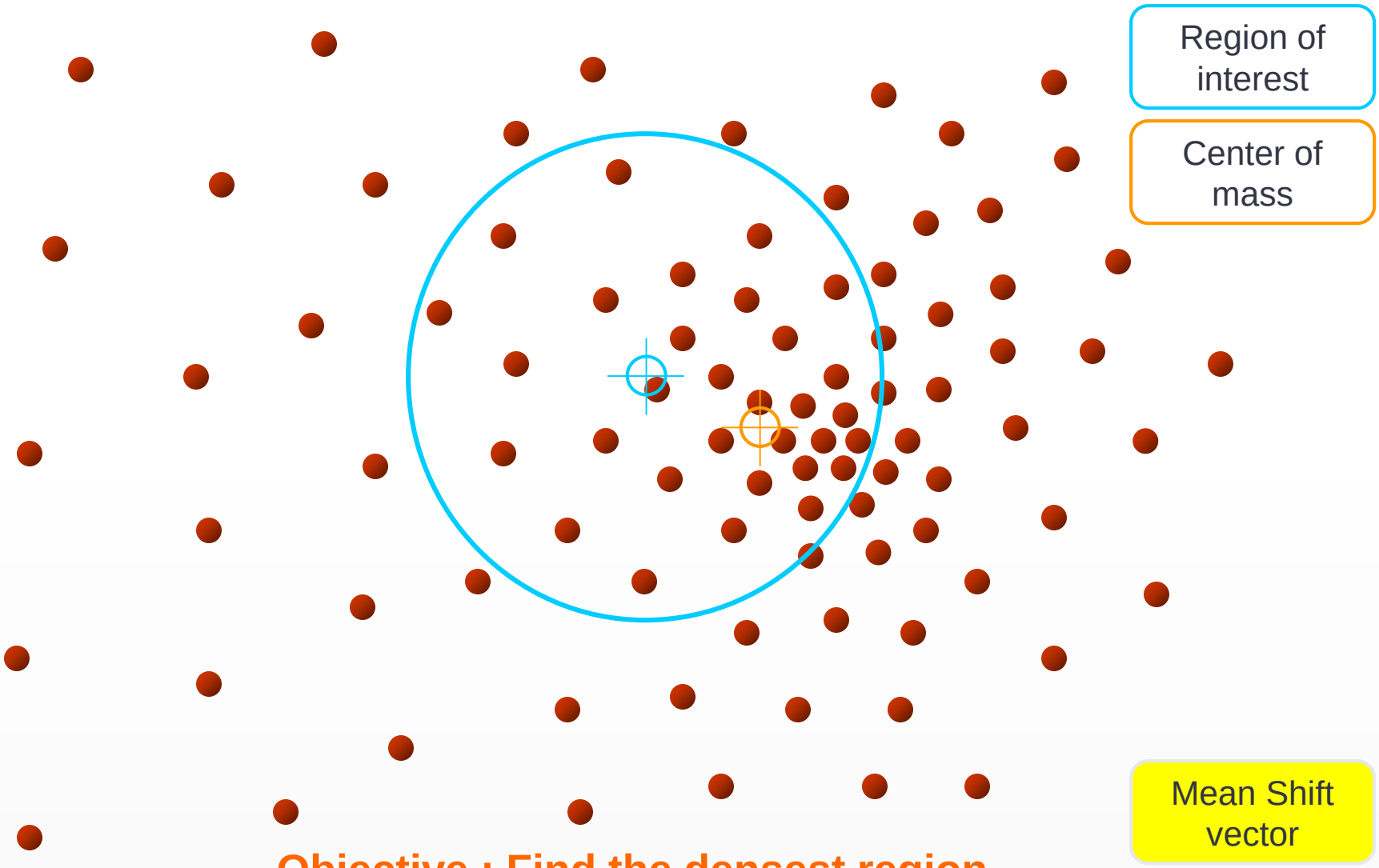
Mean shift – intuitive description



Objective : Find the densest region

Distribution of identical billiard balls

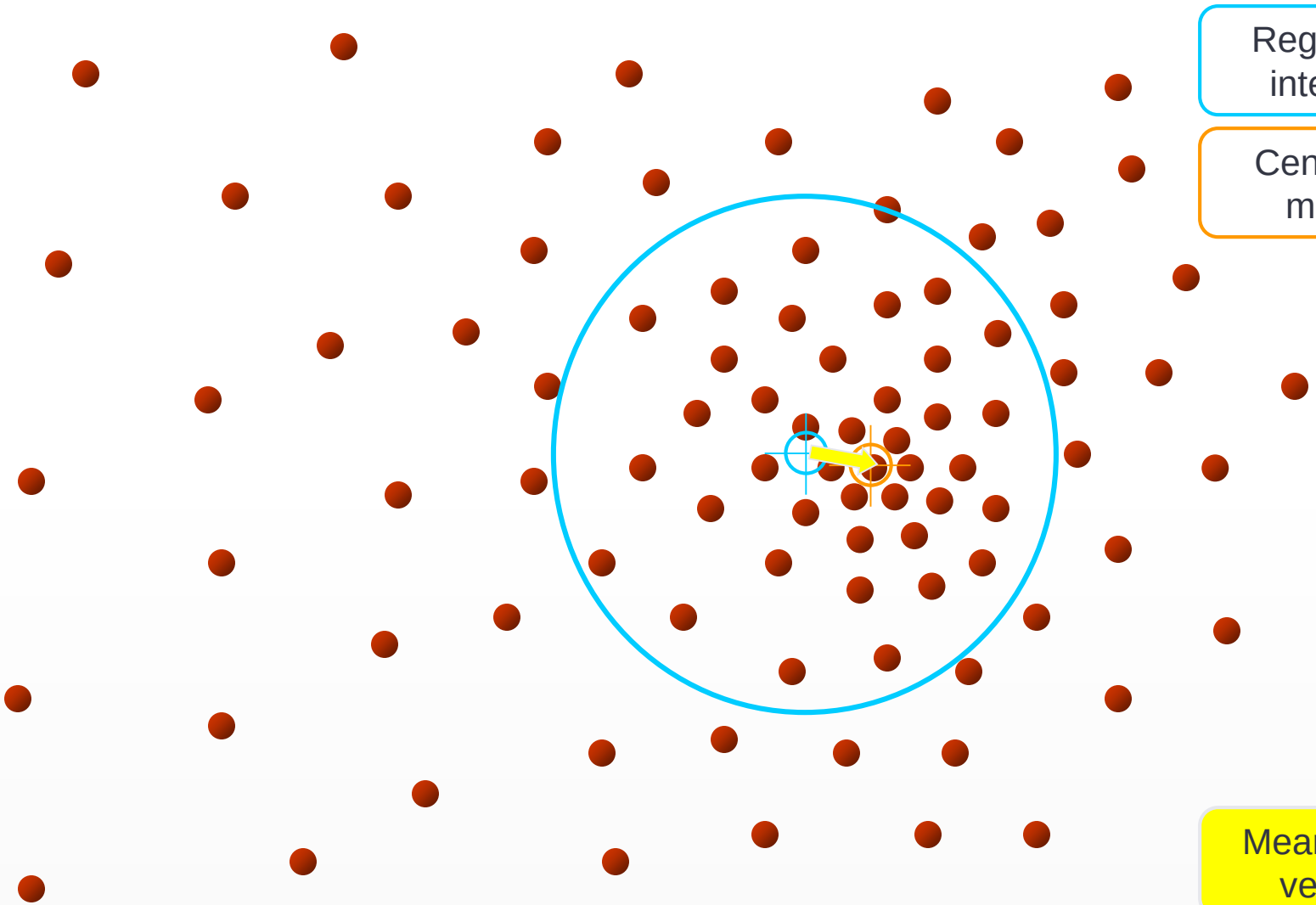
Mean shift – intuitive description



Objective : Find the densest region

Distribution of identical billiard balls

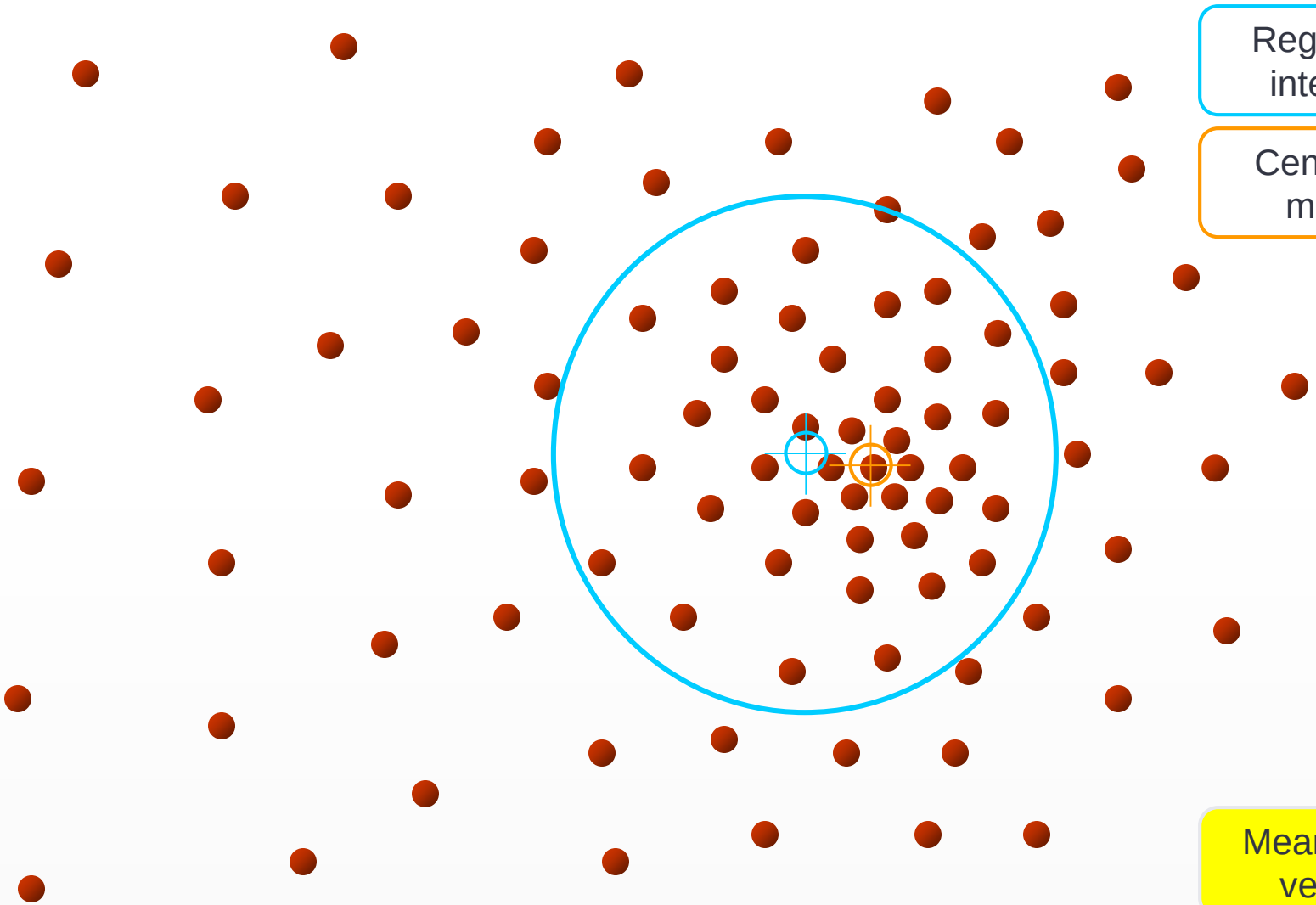
Mean shift – intuitive description



Objective : Find the densest region

Distribution of identical billiard balls

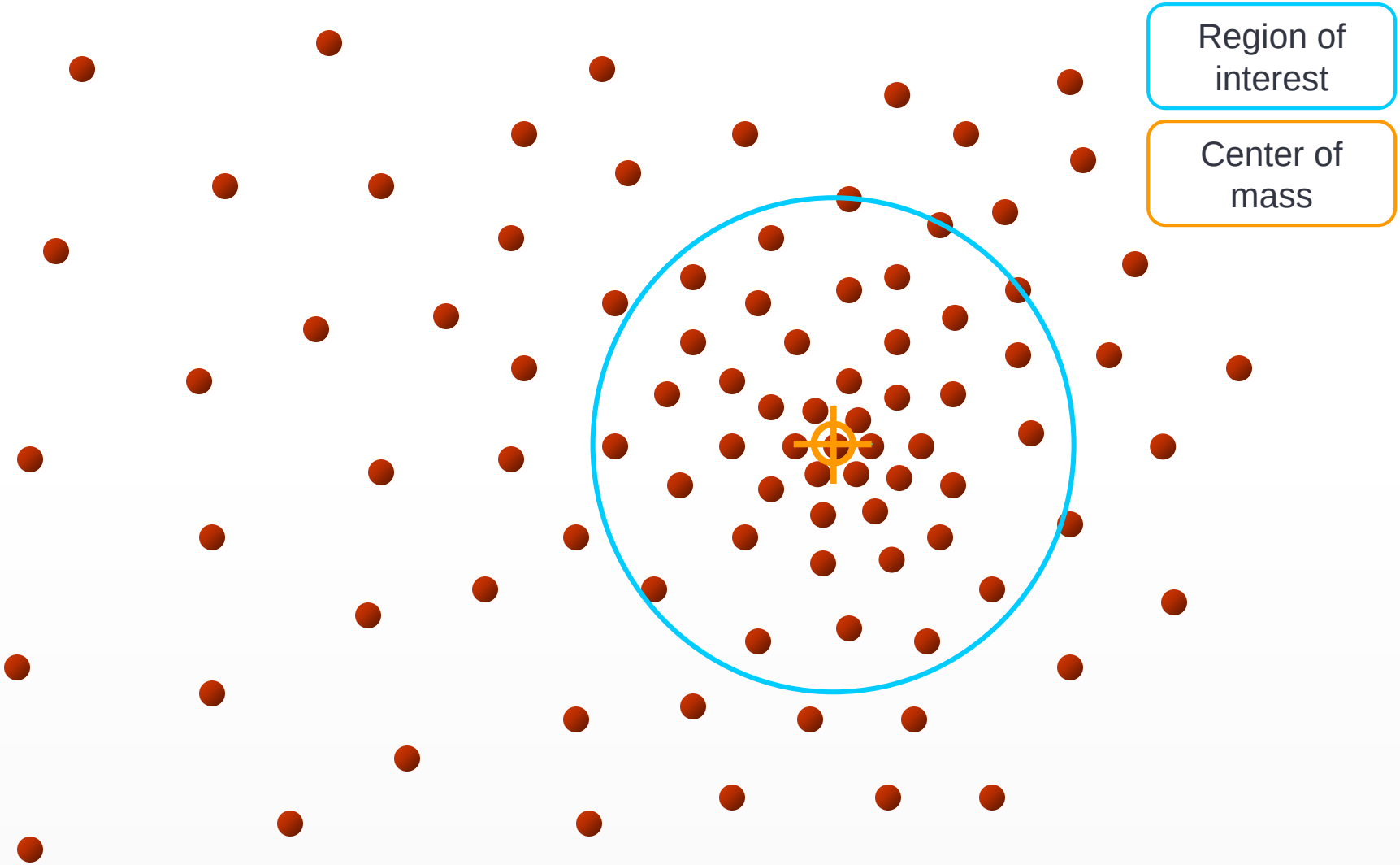
Mean shift – intuitive description



Objective : Find the densest region

Distribution of identical billiard balls

Mean shift – intuitive description



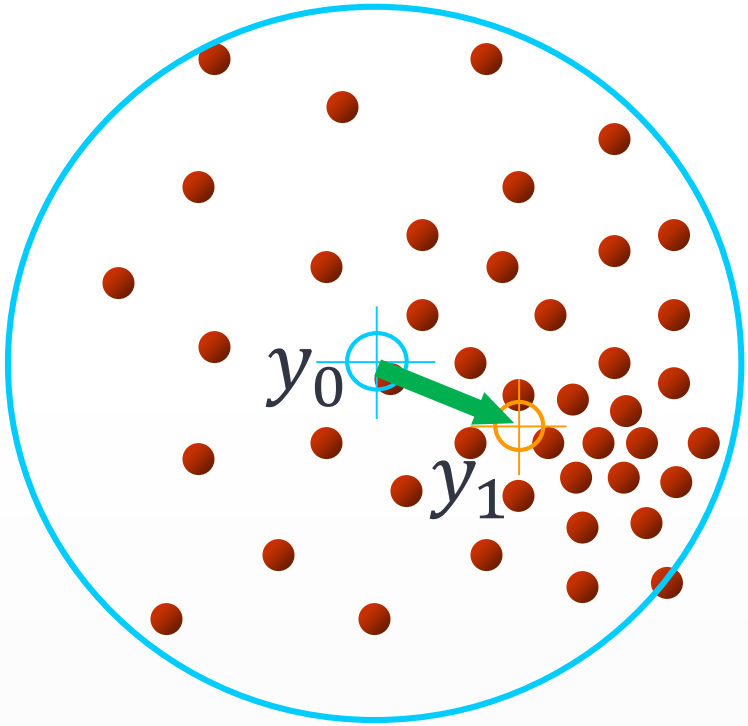
Objective : Find the densest region

Distribution of identical billiard balls

Mean shift vector

- Given:
  - Data points and approximate location of the mean of this data
- Task:
  - Estimate the exact location of the mean of the data by determining the shift vector from the initial mean
  - We do this iteratively, until we do not have to move (mean shift vector equals to zero)

Mean shift vector example



$$m_h(y) = \left[\frac{1}{n_x}\sum_{i=1}^{n_x} x_i\right] - y_0$$

- Mean shift vector always points towards the direction of the maximum increase in the density

Mean shift with point weights

$$m_h(y_0) = \left[\frac{\sum_{i=1}^{n_x} w_i(y_0) \cdot x_i}{\sum_{i=1}^{n_x} w_i(y_0)}\right] - y_0$$

- n_x: number of points in the kernel
- y_0: initial mean location
- x_i: data points
- h: kernel radius

- Weights are determined by different kernels:
  - Uniform, Gaussian, Epanechnikov

What is Mean Shift ?

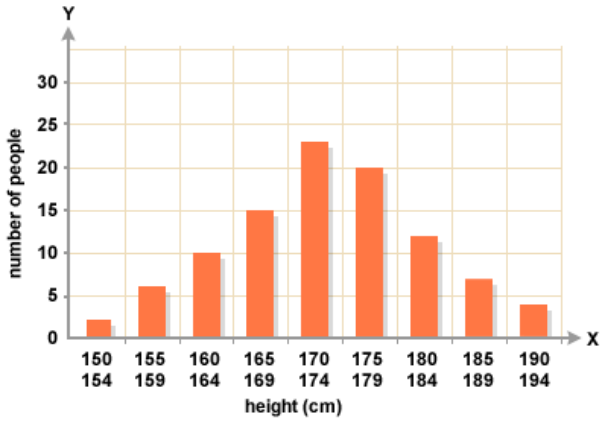
- A tool for:
  - Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in R^N
- PDF in feature space
  - Color space
  - Scale space
  - Actually any feature space you can conceive
  - …

Parametric vs. nonparametric distributions

- Problem: model the height distribution of people in the class
  - Approximate the histogram with a Gaussian density:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

    - μ and σ are empirical mean and stdev values calculated from the samples (i.e. people in the class)



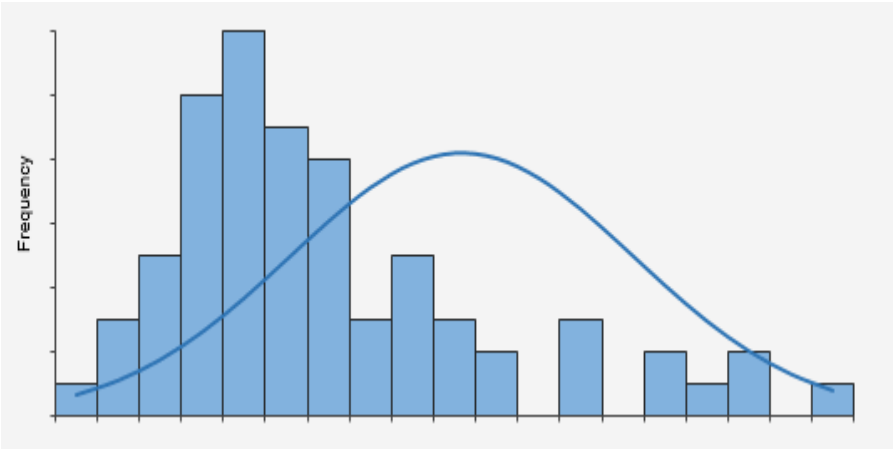
$$P(x_1 < x \leq x_2) = \int_{x_1}^{x_2} f(x)dx$$

- Parametric distributions:
  - We have a closed formula for the probability density function (PDF) with a few parameters
  - Estimate the PDF parameters from the samples, then forget the samples and use the pdf directly for probability calculation
  - Various distributions exist: Gaussian, Poisson, Gamma, Beta, etc…

Parametric vs. **nonparametric** distributions

- What happens if the distribution of samples…
  - … does not fits any well known parametric pdf formula, or…
  - … we cannot decide what sort of formula we need the use (too few samples)

$$f(x) = ???$$



Non-Gaussian distribution

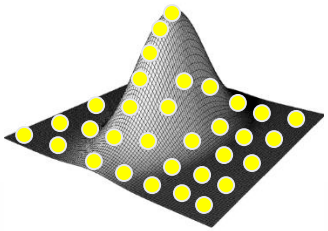
- Non-Parametric distributions:
  - We do not have a closed formula for the probability density function (PDF)
  - Instead, we need to store the samples, and use the samples directly to model the PDF
  - *Our desire:* the value of $f(x)$ should be „high”, if we find „a lot of samples” around x

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

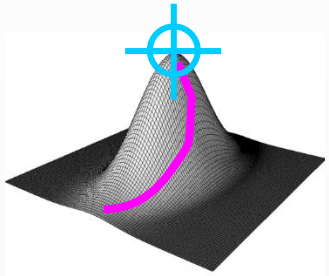
Data

Non-parametric Density Estimation



Discrete PDF Representation

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

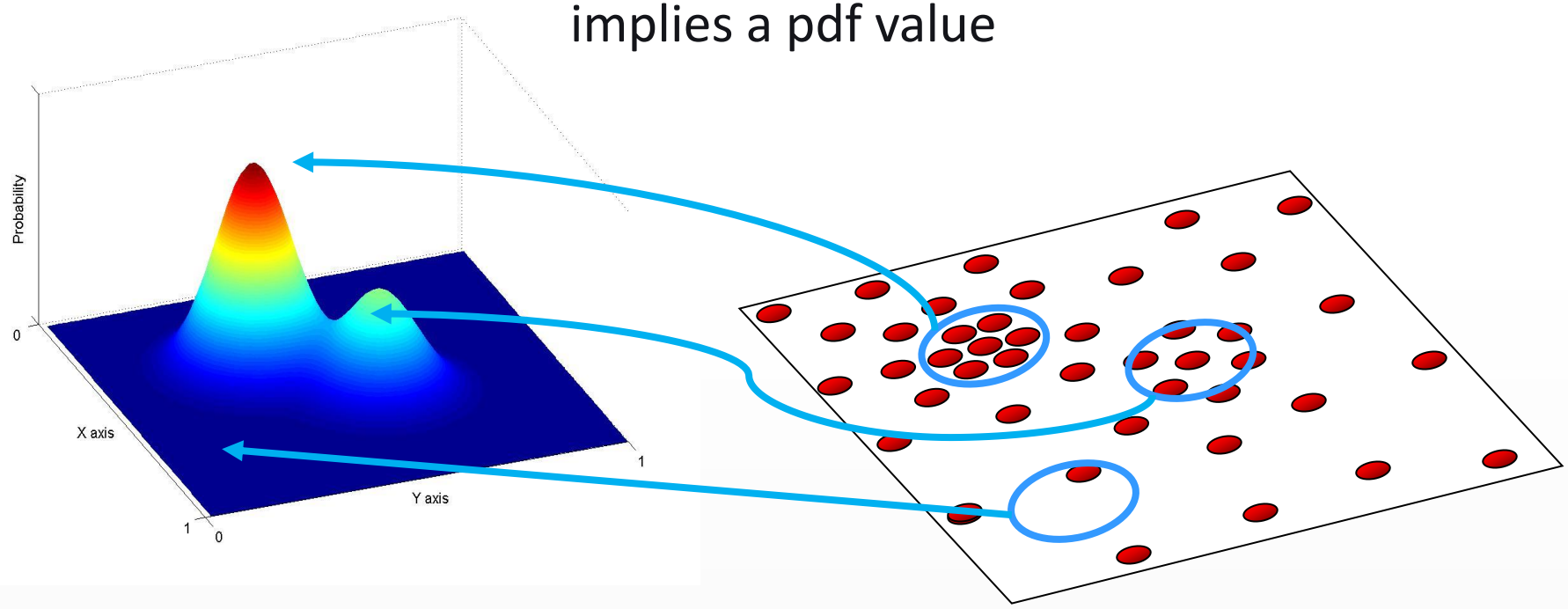


PDF Analysis

Non-Parametric Density Estimation

The data point density implies a pdf value

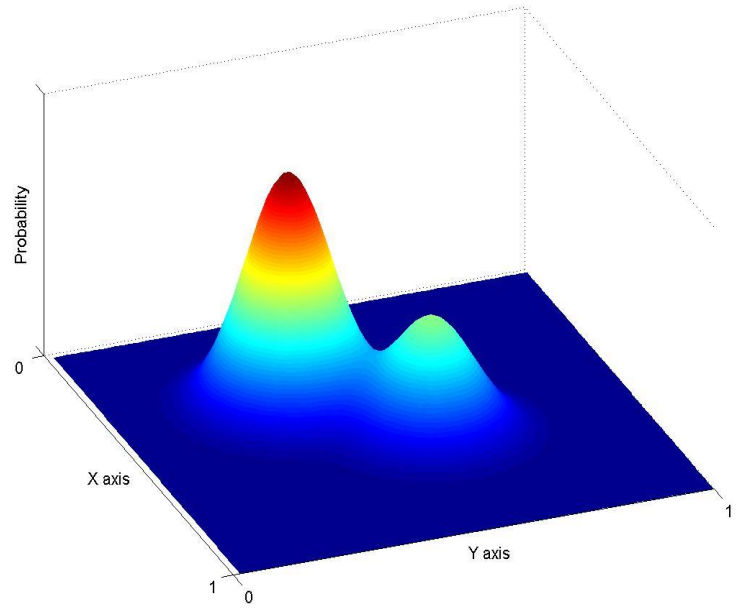
2D data points



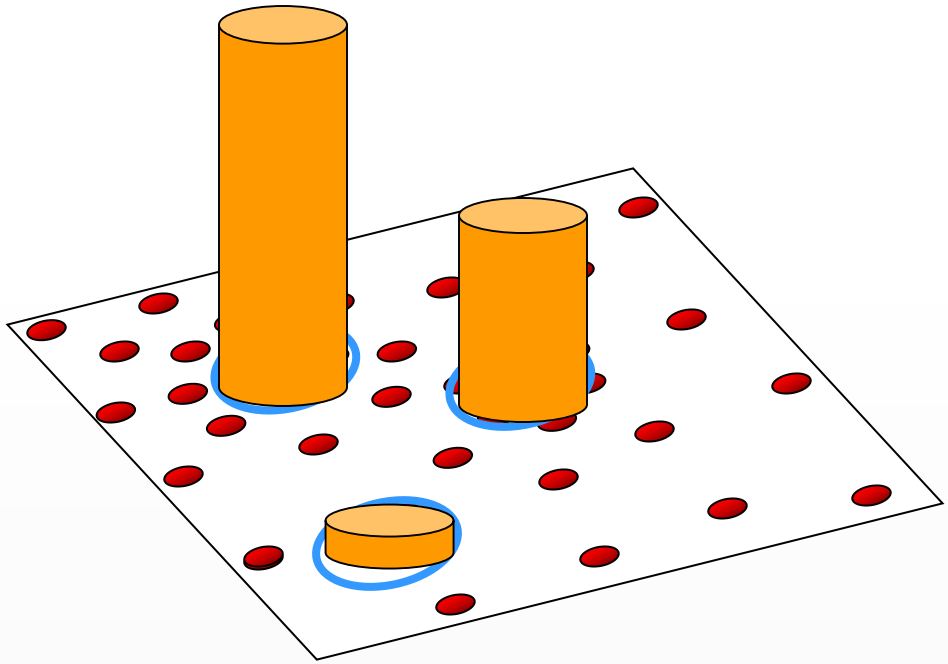
Assumed Underlying PDF

Real Data Samples

Non-Parametric Density Estimation

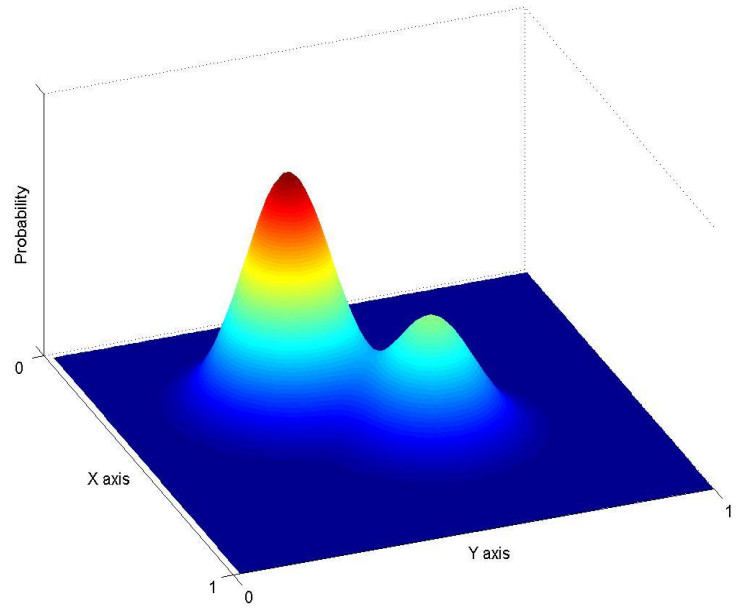


Assumed Underlying PDF

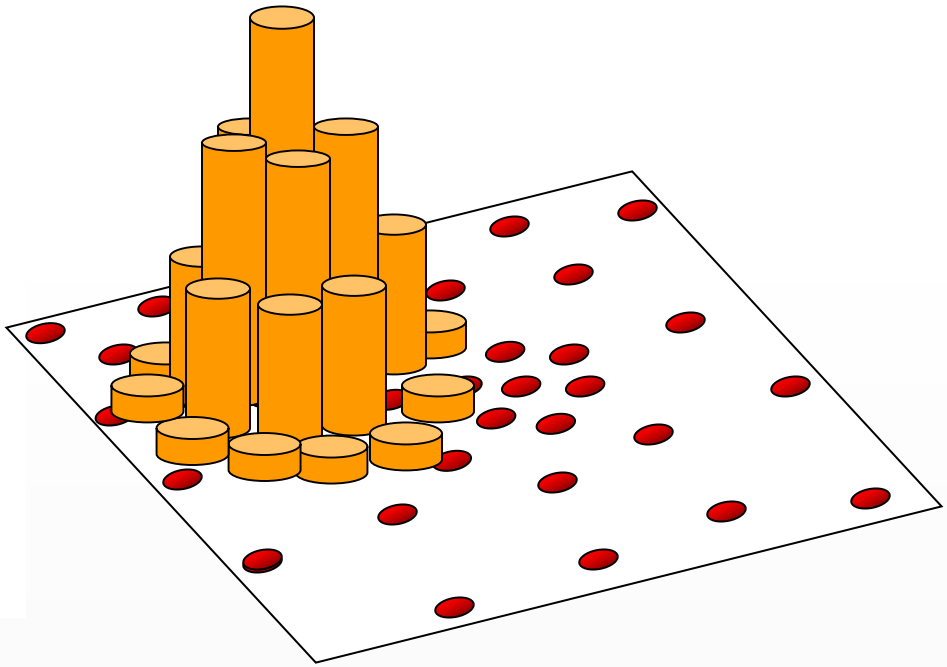


Real Data Samples

Non-Parametric Density Estimation



Assumed Underlying PDF

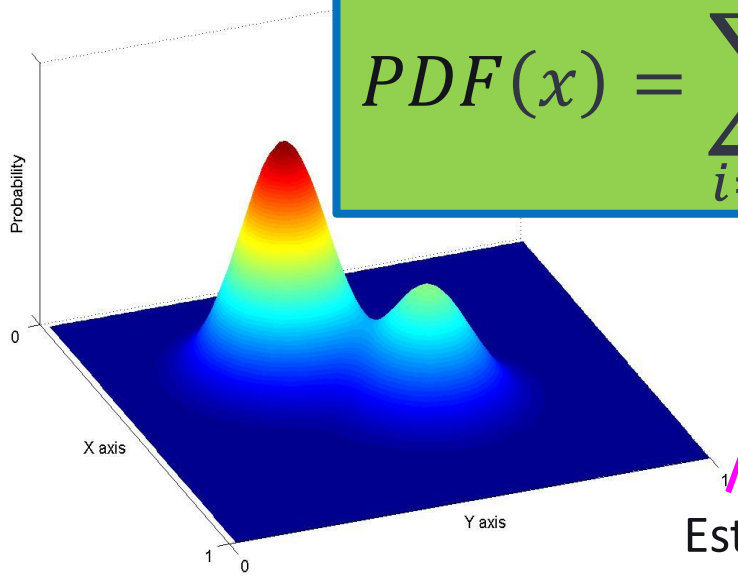


Real Data Samples

Kernel Density Estimation

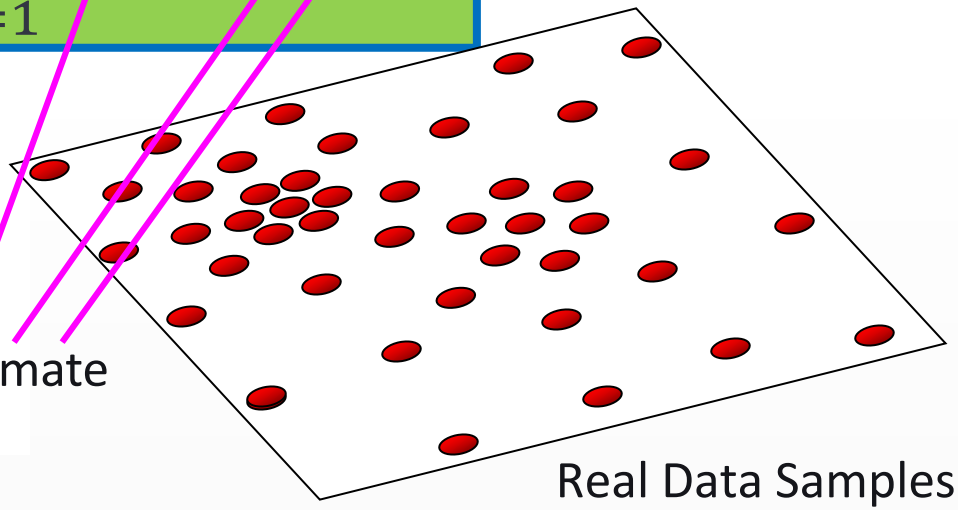
Assumption : The data points are sampled from an underlying PDF

$$PDF(x) = \sum_{i=1}^{n} c_i \cdot e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$



Assumed Underlying PDF

Estimate



Real Data Samples

- Each sample point contributes to the *PDF* with an additive term (here: Gaussian) - μ_i : equal to the ith sample

Kernel Density Estimation

$$PDF(x) = \sum_{i=1}^{n} c_i \cdot e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$

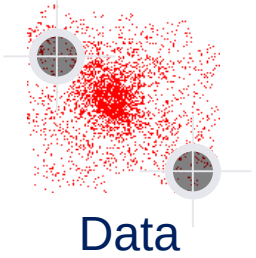
- Non-parametric *PDF* with Gaussian kernel:
  - Seems like a mixture of Gaussians, where the number of components is equal to the number of samples, and the mean values of the components are at the sample points $\mu_1, \mu_2, \ldots, \mu_n$
- Probability calculation for particular x value:
  - We calculate it as a weighted sum from the surrounding sample points - all the points contribute!
  - We look at the distance of x from each sample point
  - The *PDF* value is high for x which has a lot of samples around it

Kernel Density Estimation
Various Kernels

- ***Roles of kernels:*** they determine the weights of nearby points in the density calculation.

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

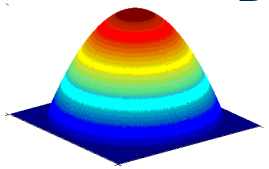
A function of some finite number of data points $x_1 \ldots x_n$



Examples:

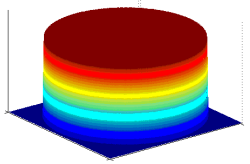
- Epanechnikov Kernel

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



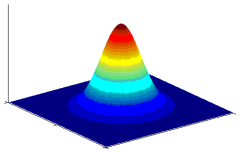
- Uniform Kernel

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



- Normal Kernel

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



Profile and kernel

- Radially symmetric kernel

$$K(x) = ck(\|x\|^2)$$

Profile

$$P(x) = \frac{1}{n}\sum_{i=1}^{n} K(x - x_i) = \frac{1}{n} c \sum_{i=1}^{n} k(\|x - x_i\|^2)$$

Kernel Density Estimation

- Non parametric probability function (pdf)
  - We do not have any assumptions about the closed form of the distribution (such as Gaussian or mixture of Gaussians)
  - We estimate the pdf directly from the sample points $x_1 \ldots x_n$

$$P(x) = \frac{1}{n} c \sum_{i=1}^{n} k(\|x - x_i\|^2)$$

Non parametric probability density calculation

$$P(x) = \frac{1}{n} c \sum_{i=1}^{n} k(\|x - x_i\|^2)$$

- Given feature vector x
  - e.g. 1D gray value, 3D color vector, 6D vector of color + texture components etc.
- Task: calculate the probability (density) value of x directly from the sample points $x_1 \dots x_n$
  - Calculate the Euclidean distance d_i of x from each x_i.
  - Use a kernel profile $k(.)$ which assigns a weight to x_i as a function of the calculated d_i distance (for lower distance higher weight, see different kernels)
  - Take the *pdf* value as a the normalized sum of the weights
  - High *pdf* values corresponds to x features which have several x_i-s „nearby”

Kernel Density Estimation

- Relations of nonparametric pdfs and means shift

$$P(x) = \frac{1}{n} c \sum_{i=1}^{n} k(\|x - x_i\|^2)$$

- Derivative of the pdf (gradient of the density):

$$\nabla P(x) = \frac{1}{n} c \sum_{i=1}^{n} \nabla k(\|x - x_i\|^2)$$

$$\nabla P(x) = \frac{1}{n} 2c \sum_{i=1}^{n} (x - x_i) k'(\|x - x_i\|^2)$$

Kernel Density Estimation

$$\nabla P(x) = \frac{1}{n} 2c \sum_{i=1}^{n} (x - x_i) k'(\|x - x_i\|^2)$$

$$g(x) := -k'(x)$$

$$\nabla P(x) = \frac{1}{n} 2c \sum_{i=1}^{n} (x_i - x) g(\|x - x_i\|^2)$$

$$\nabla P(x) = \frac{1}{n} 2c \sum_{i=1}^{n} x_i g(\|x - x_i\|^2) \; - \frac{1}{n} 2c \sum_{i=1}^{n} x g(\|x - x_i\|^2)$$

$$\nabla P(x) = \frac{1}{n} 2c \sum_{i=1}^{n} g(\|x - x_i\|^2) \left[\frac{\sum_{i=1}^{n} x_i g(\|x - x_i\|^2)}{\sum_{i=1}^{n} g(\|x - x_i\|^2)} - x \right]$$

Kernel Density Estimation

$$\nabla P(x) = \frac{1}{n} 2c \sum_{i=1}^{n} g(\|x - x_i\|^2) \left[\frac{\sum_{i=1}^{n} x_i g(\|x - x_i\|^2)}{\sum_{i=1}^{n} g(\|x - x_i\|^2)} - x \right]$$

$$\nabla P(x) = \frac{1}{n} 2c \sum_{i=1}^{n} g_i \left[\frac{\sum_{i=1}^{n} x_i g_i}{\sum_{i=1}^{n} g_i} - x \right]$$

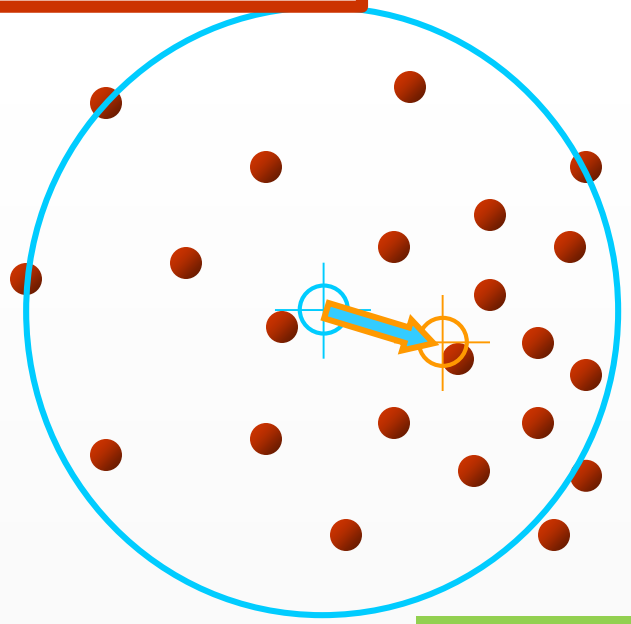
Mean shift & nonparametric density analysis

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

$m(x)$ mean shift vector

$$\nabla P(x) = \frac{c}{n}\sum_{i=1}^{n} g_i \times m(x)$$

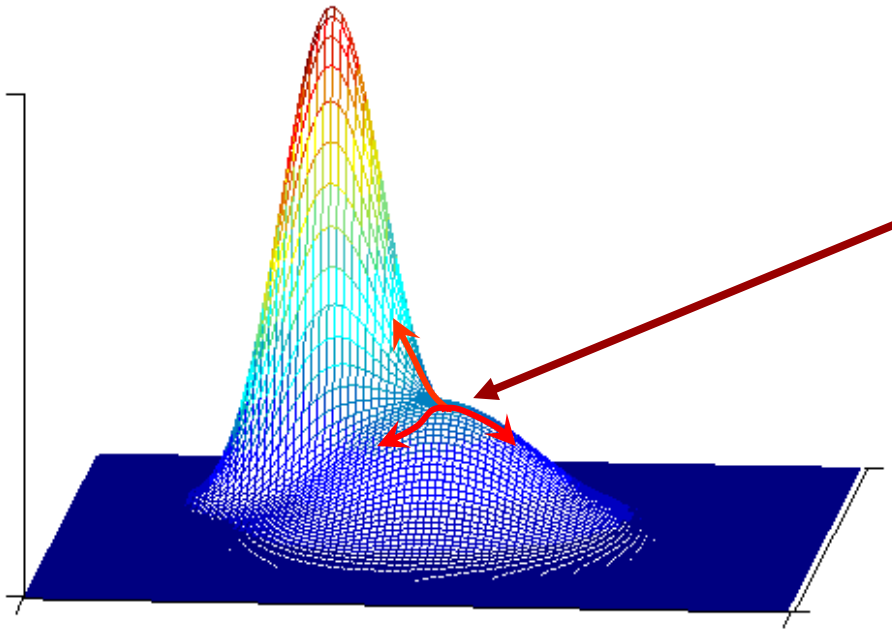
$$m(x) = \frac{\nabla P(x)}{\frac{c}{n}\sum_{i=1}^{n} g_i}$$



Main theoretic result: Mean shift vector is proportional to the gradient of the *nonparametric pdf*, therefore it is appropriate for mode seeking

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

Mean shift mode detection



What happens if we reach a saddle point **?**

Perturb the mode position and check if we return back

Updated Mean Shift Procedure:

- Find all modes using the Simple Mean Shift Procedure
- Prune modes by perturbing them (find saddle points and plateaus)
- Prune nearby – take highest mode in the window

Mean-Shift Clustering

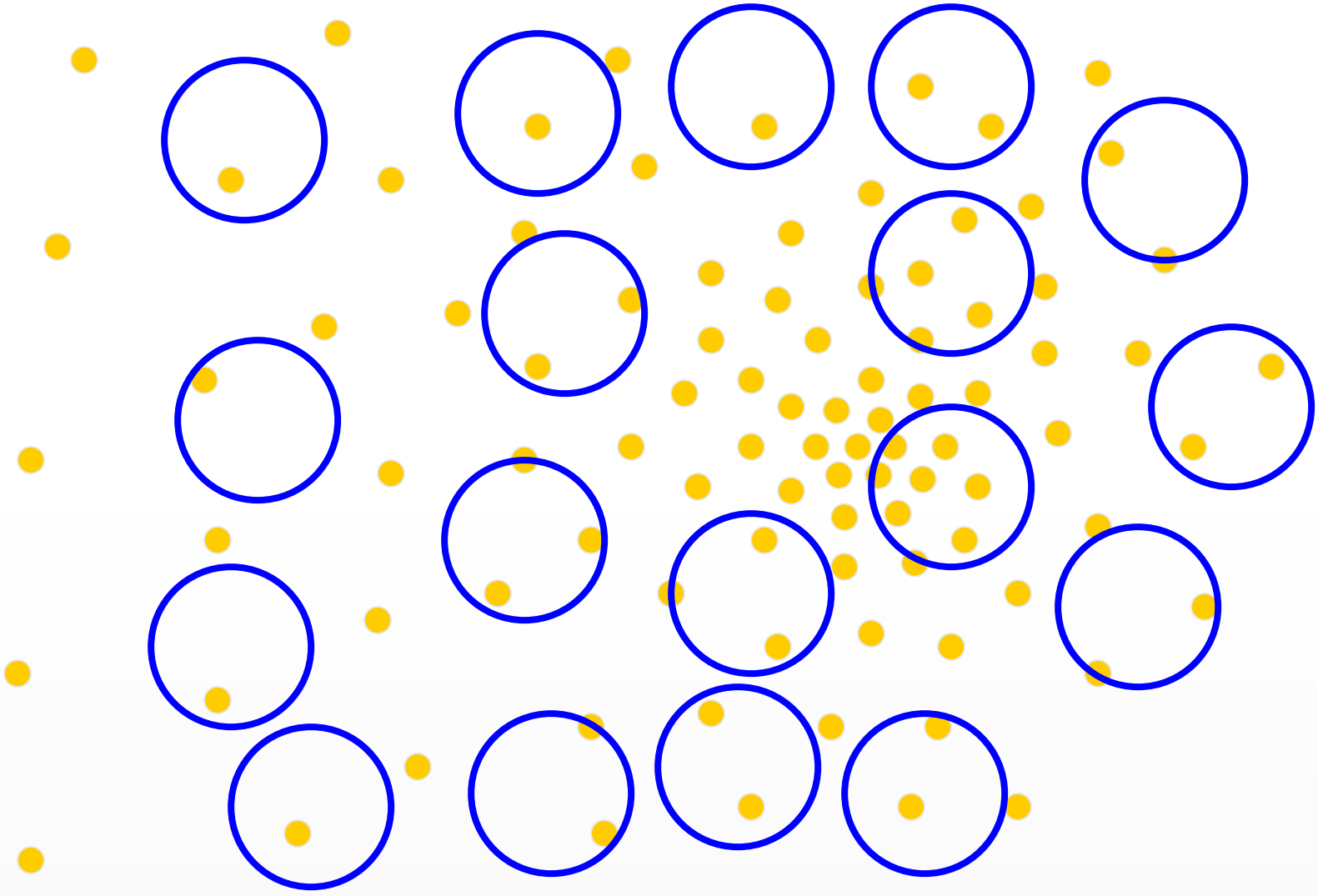
- Main steps:
  1. A density estimation window (e.g. a Gaussian window) is placed on each sample point.
  2. Within each window the mean shift vector is calculated, which points toward the maximum density:

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i \, g\left(\left\| \frac{x - x_i}{h} \right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2\right)} - x$$

where …
x is a *d* dimensional feature point,
g(x)=-K'(x), where ***K*** is a kernel function (e.g. Gaussian kernel)
h is the bandwidth parameter of the kernel

  3. The window is shifted with the mean shift vector.
  4. Step 2 and 3 are repeated until convergence to a local density maximum.
  5. The sample points that converged to the same local maximum will belong to the same cluster.

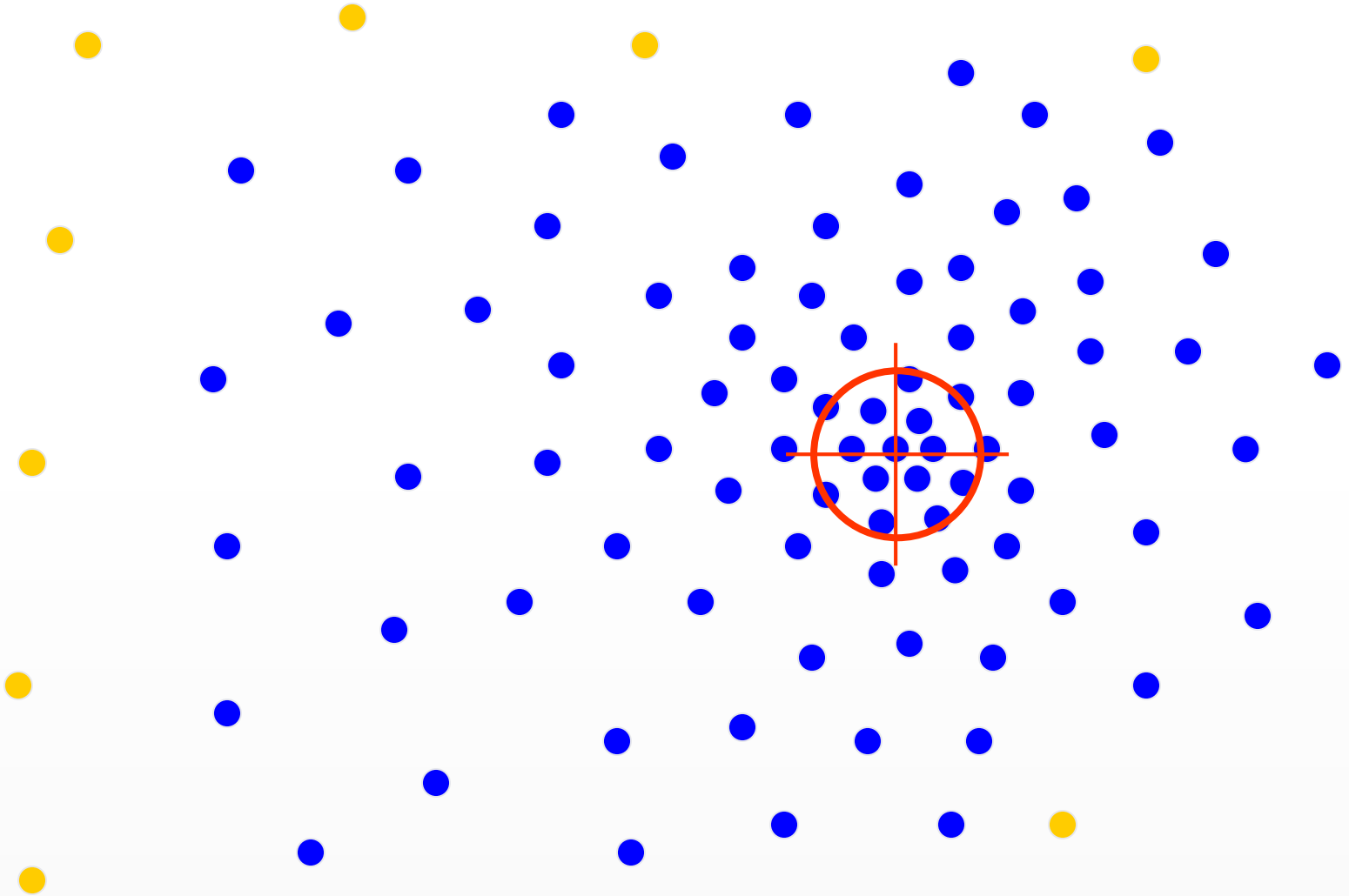
Real Modality Analysis



Tessellate the space with windows

Run the procedure in parallel

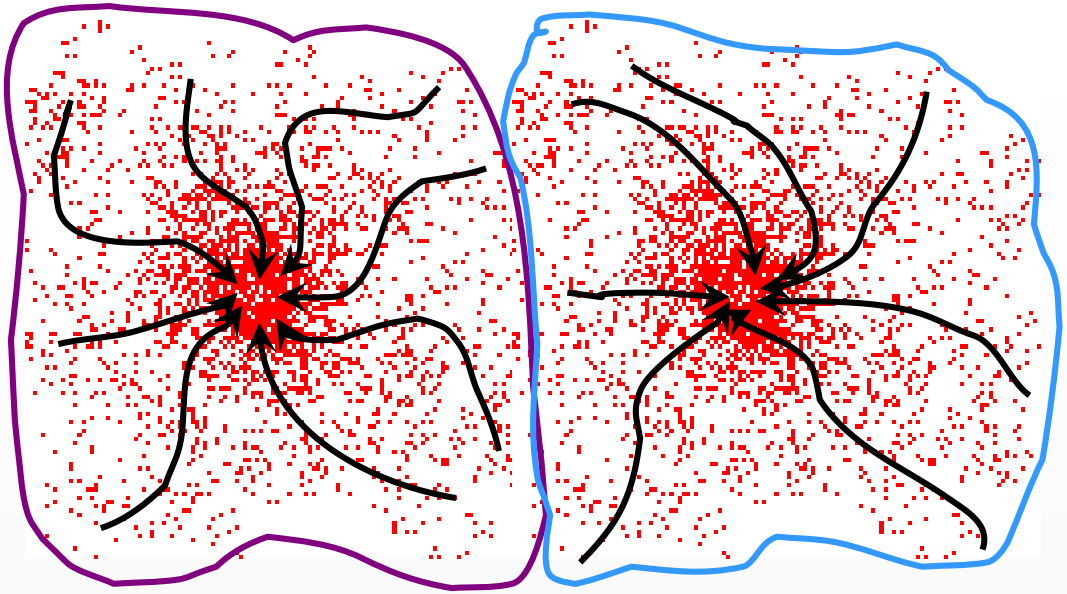
Real Modality Analysis



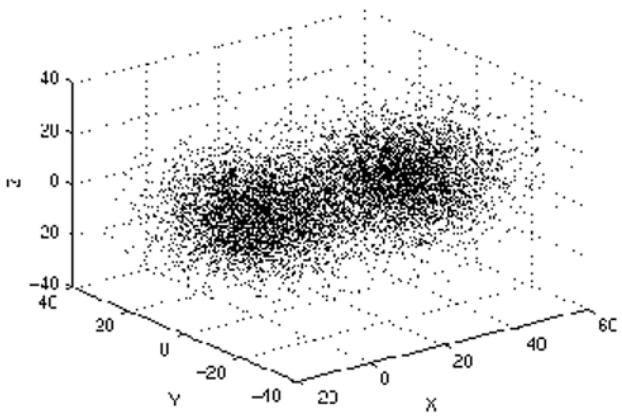
The blue data points were traversed by the windows towards the mode

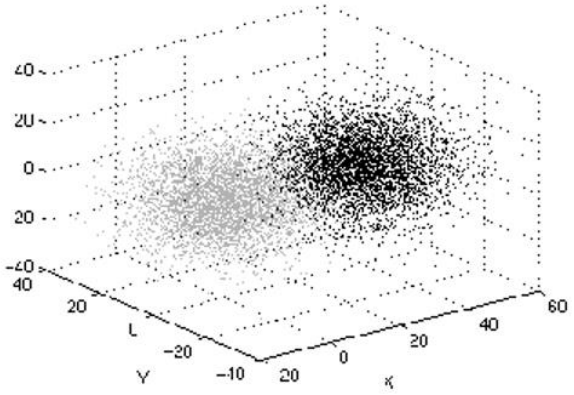
Attraction basin

- Attraction basin: the region for which all trajectories lead to the same mode
- Cluster: all data points in the attraction basin of a mode

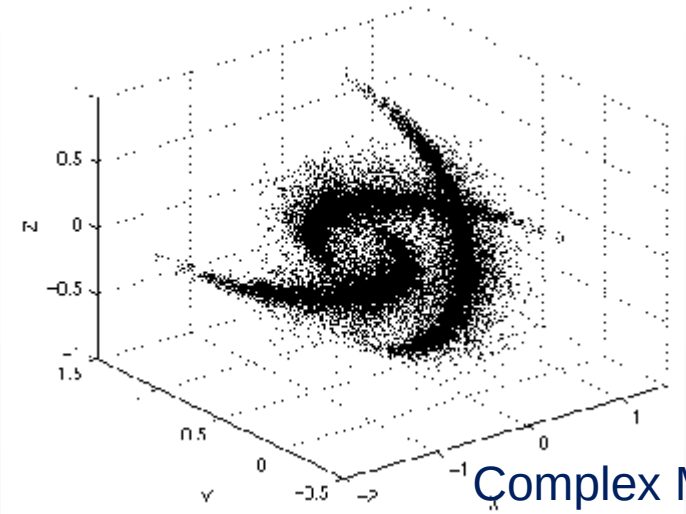


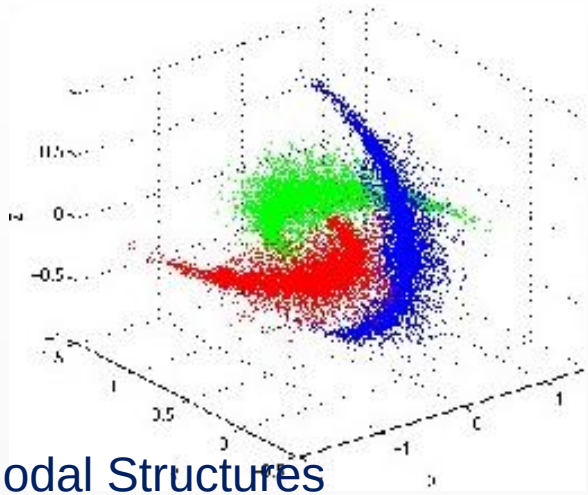
Clustering
Synthetic Examples





Simple Modal Structures

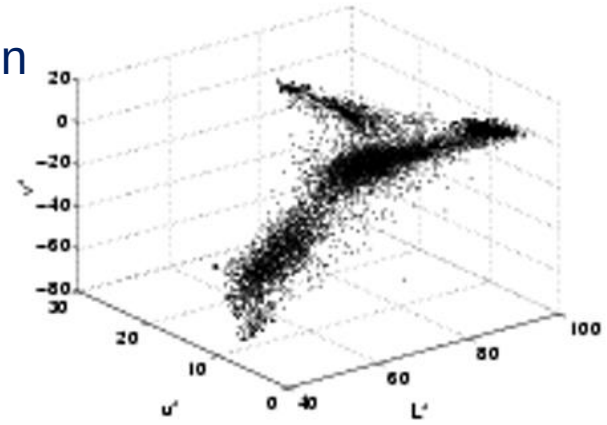




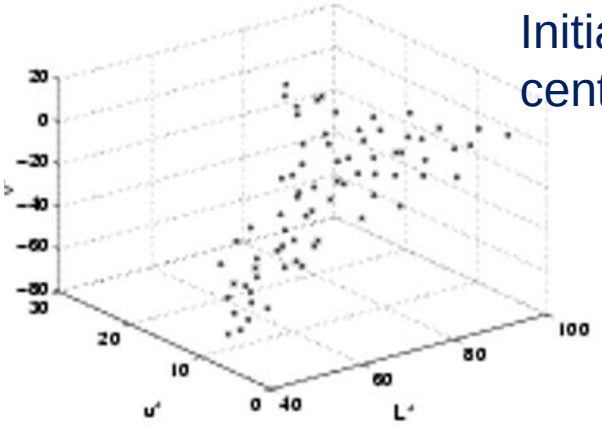
Complex Modal Structures

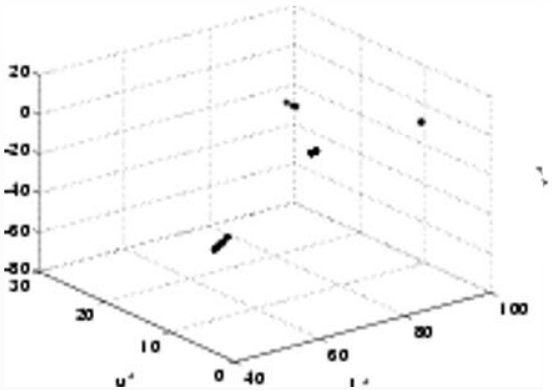
Clustering
Real example

Feature space:
L*u*v representation

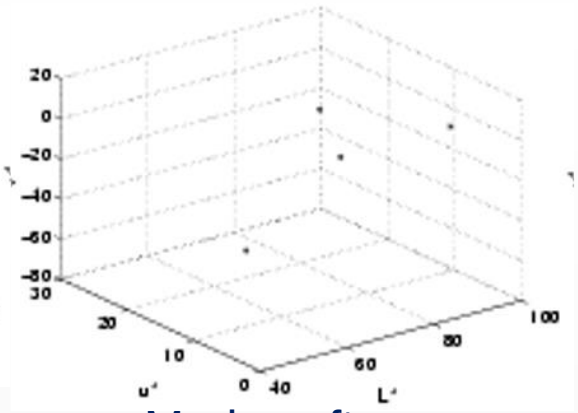


Initial window centers

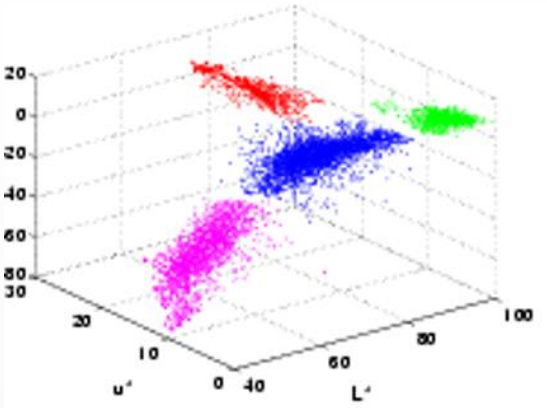




Modes found

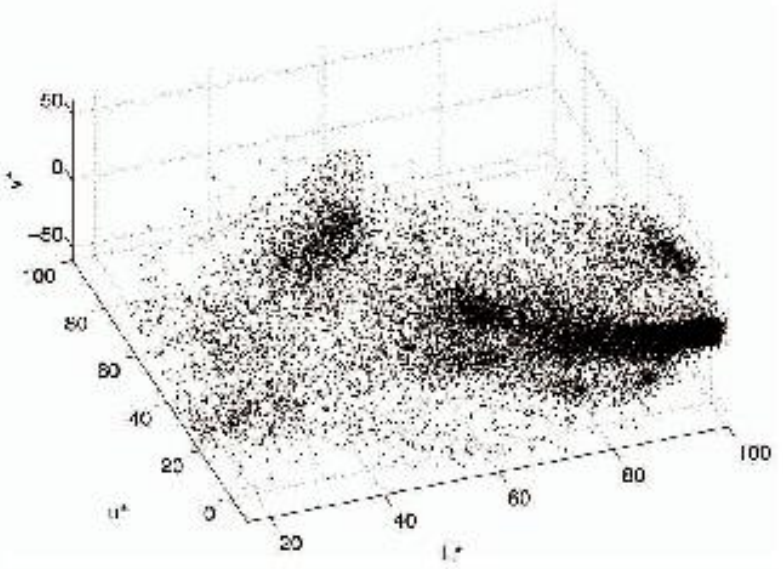


Modes after pruning



Final clusters

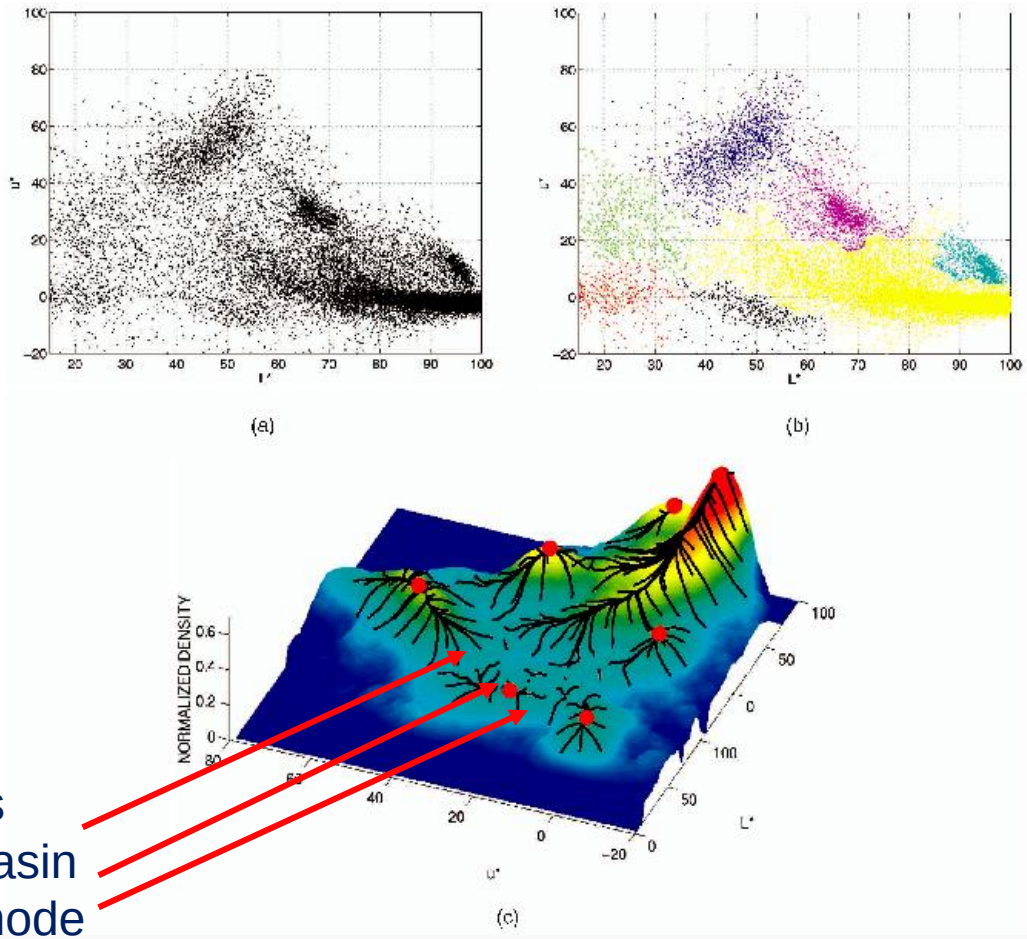
Clustering
Real example



L*u*v space representation

Clustering
Real example

2D (L*u) space representation



Final clusters

Not all trajectories in the attraction basin reach the same mode

Mean shift for image segmentation

- Segmented regions
  - Similar color/texture values
  - Spatially connected pixels
- Grayscale image segmentation model
  - Each pixel = a „billiard ball” $\boldsymbol{x}$ in the 3D joint spatial-intensity space:
    $\boldsymbol{x} = [x, y, z(x, y)\,] \in \mathbb{R}^3$
    where $z(x,y)$ is the gray level of pixel (x,y)
  - *Segmentation:* find the modes of this 3D distribution – i.e. dense regions with their attraction basins

Discontinuity Preserving Smoothing

Feature space : Joint domain = spatial coordinates + color space

$$K(\mathbf{x}) = C \cdot k_s\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|\right) \cdot k_r\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|\right)$$

Meaning : treat the image as data points in the spatial and gray level domain:

$\boldsymbol{x} = [x^s, x^r] = [x, y, z(x,y)\,] \in \mathbb{R}^3$ where $z(x,y)$ is the gray level of pixel (x,y)

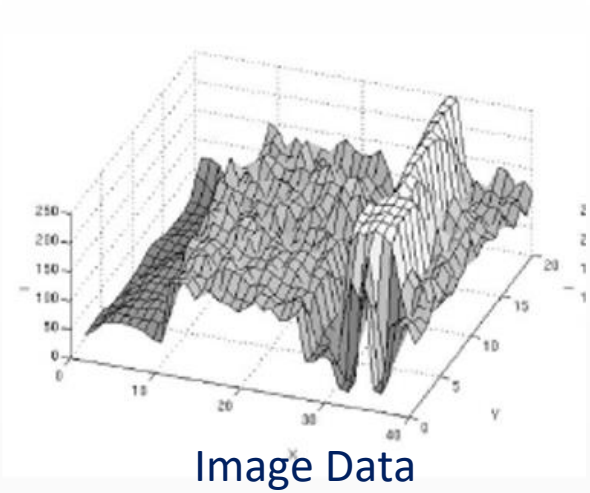
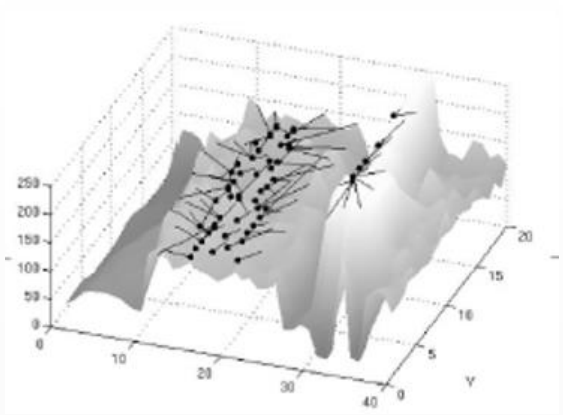
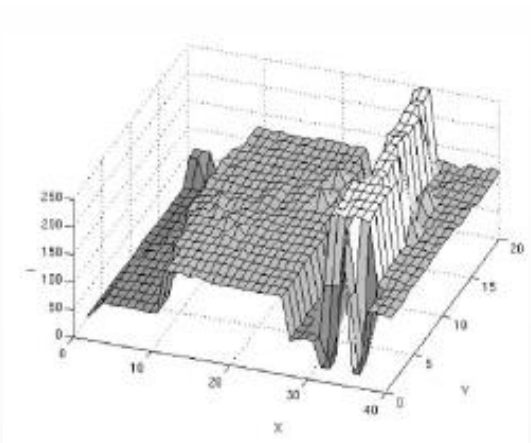


Image Data (slice)

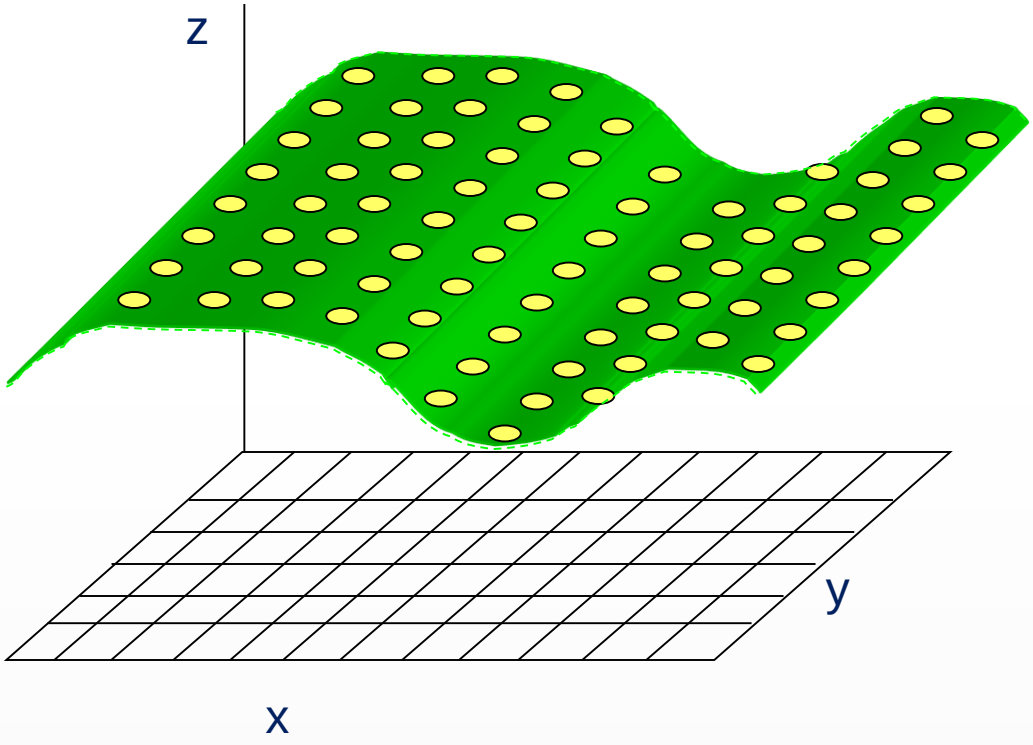


Mean Shift vectors



Smoothing result

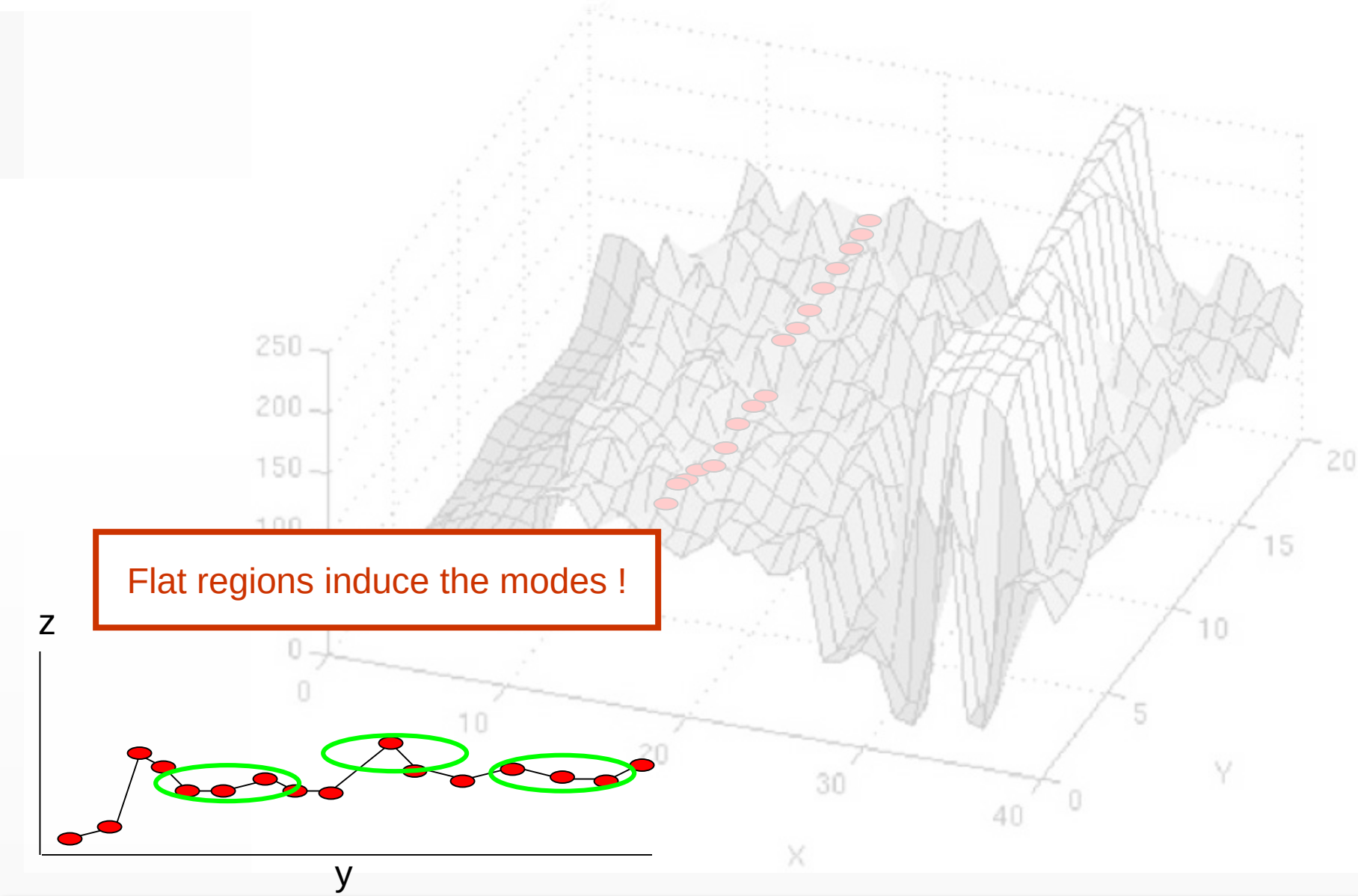
Discontinuity Preserving Smoothing



The image gray levels…

… can be viewed as data points in the *x, y, z* space (joined spatial and color space)

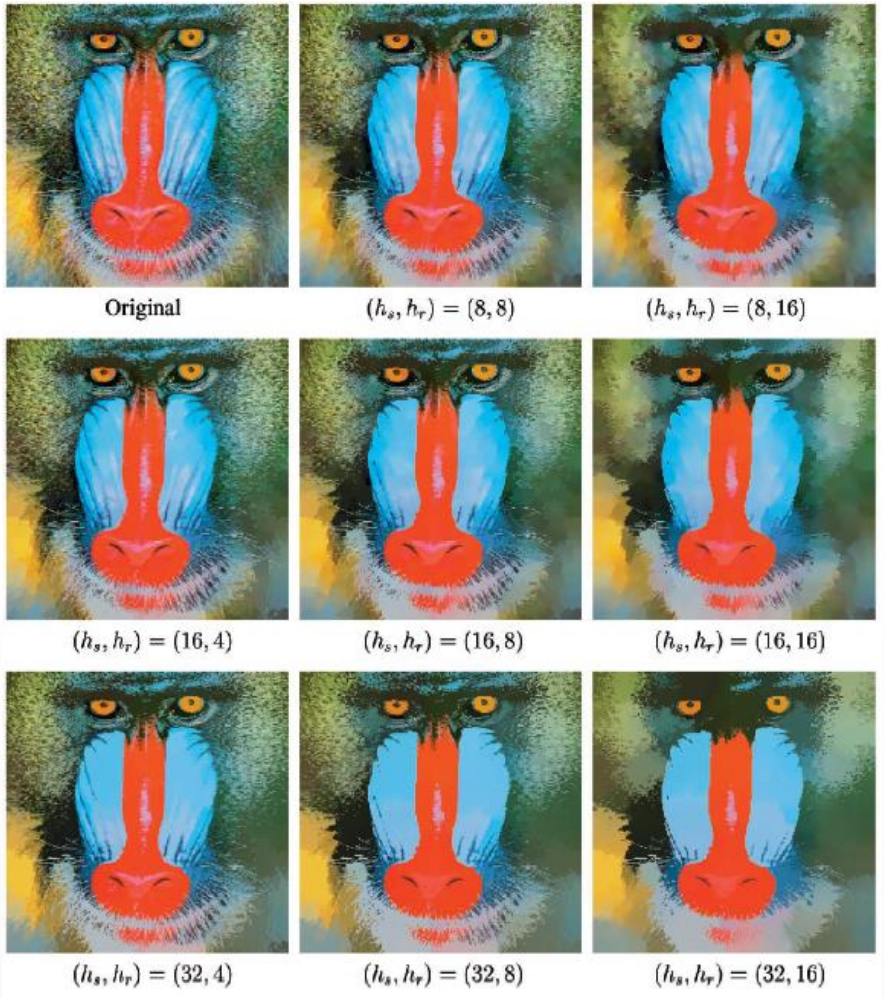
Discontinuity Preserving Smoothing



Flat regions induce the modes !

Discontinuity Preserving Smoothing

- The effect of window size in spatial and range spaces



Discontinuity Preserving Smoothing

Discontinuity Preserving Smoothing

Segmentation

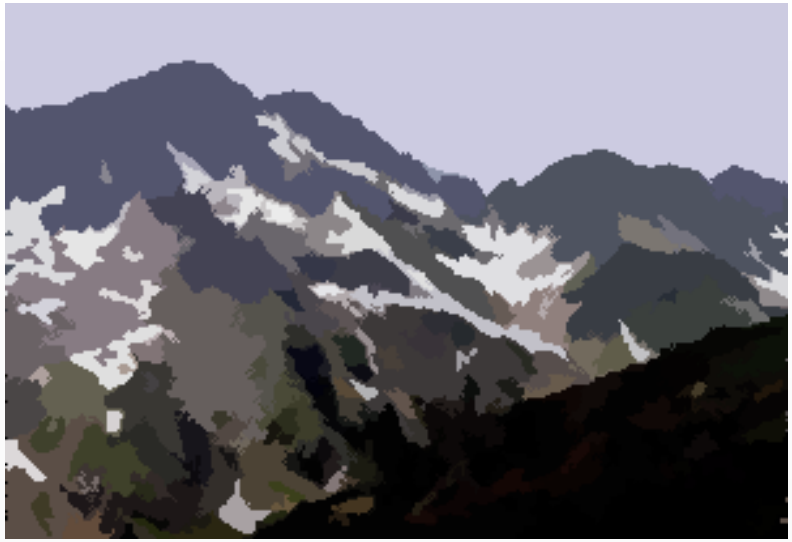
- Segment = Cluster, or Cluster of Clusters
- Algorithm:
  - Run Filtering (discontinuity preserving smoothing)
  - Cluster the clusters which are closer than window size

Segmentation

…when feature space is only gray levels…

Segmentation

Mean shift segmentation results



http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html




http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Mean-shift: other issues

- Speedups
  - Uniform kernel (much faster but not as good)
  - Binning or hierarchical methods
  - Approximate nearest neighbor search
- Methods to adapt kernel size depending on data density
- Lots of theoretical support

  D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Mean shift pros and cons

- Pros
  - Good general-practice segmentation
  - Finds variable number of regions
  - Robust to outliers
- Cons
  - Have to choose kernel size in advance
  - Original algorithm doesn’t deal well with high dimensions
- When to use it
  - Oversegmentatoin
  - Multiple segmentations
  - Other tracking and clustering applications