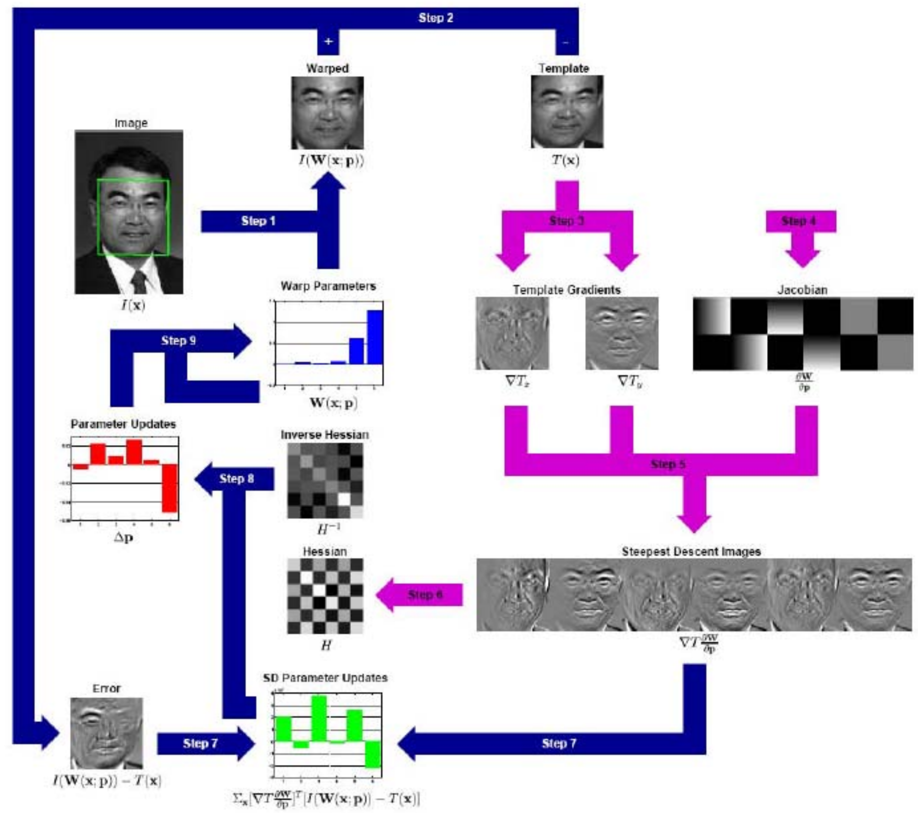
KLT Tracker

Tracker

1. Detect Harris corners in the first frame
2. For each Harris corner compute motion between consecutive frames (Alignment).
3. Link motion vectors in successive frames to get a track
4. Introduce new Harris points at every *m* frames
5. Track new and old Harris points using steps 2-4.

Alignment



Mean-Shift Tracking

Lecture-11

Mean-Shift Tracking

Mean-Shift Tracking

Mean-Shift Tracking

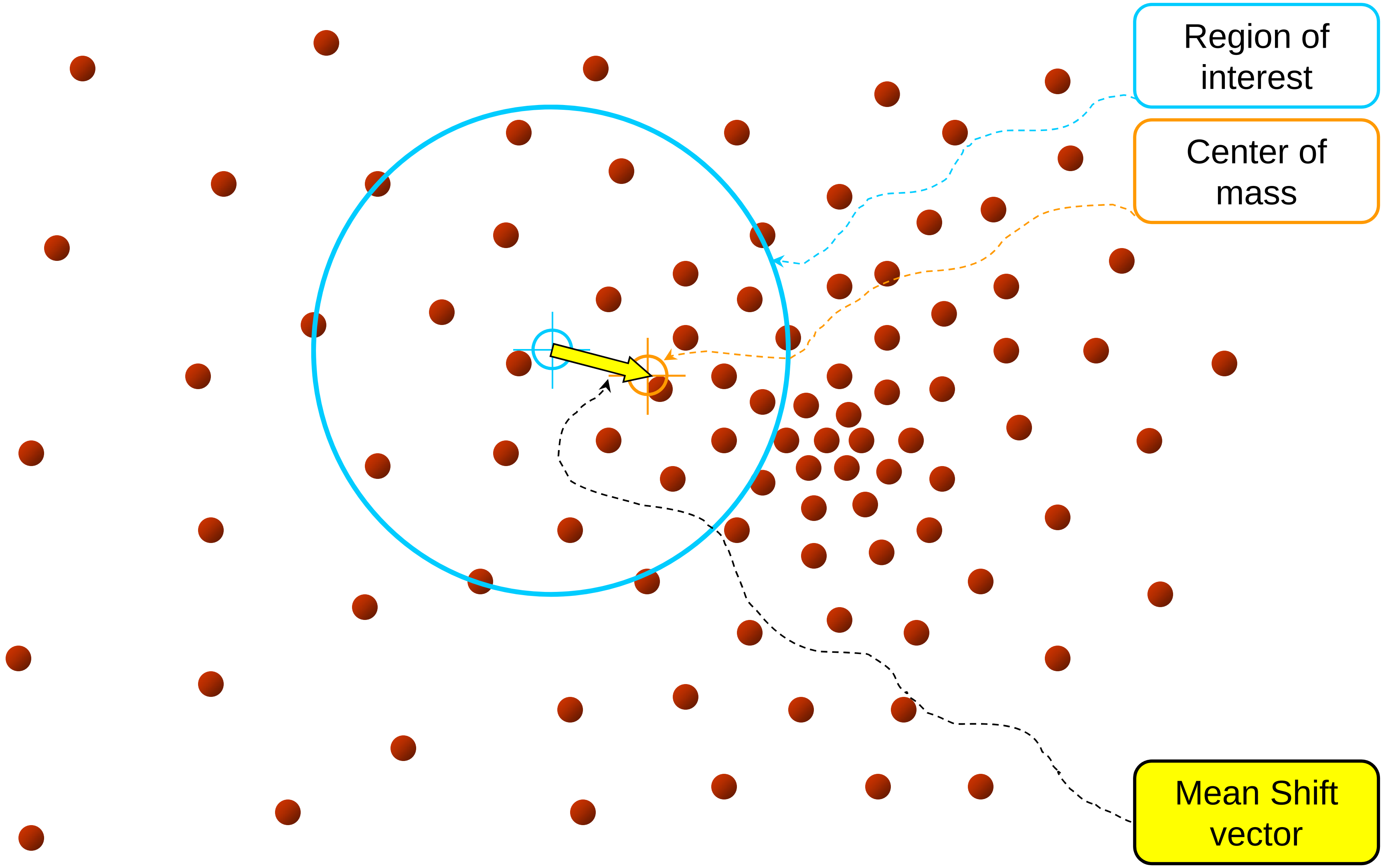
Presentations

- *Comaniciu* et *al*
- Alper Yilmaz
- Afshin Dehghan

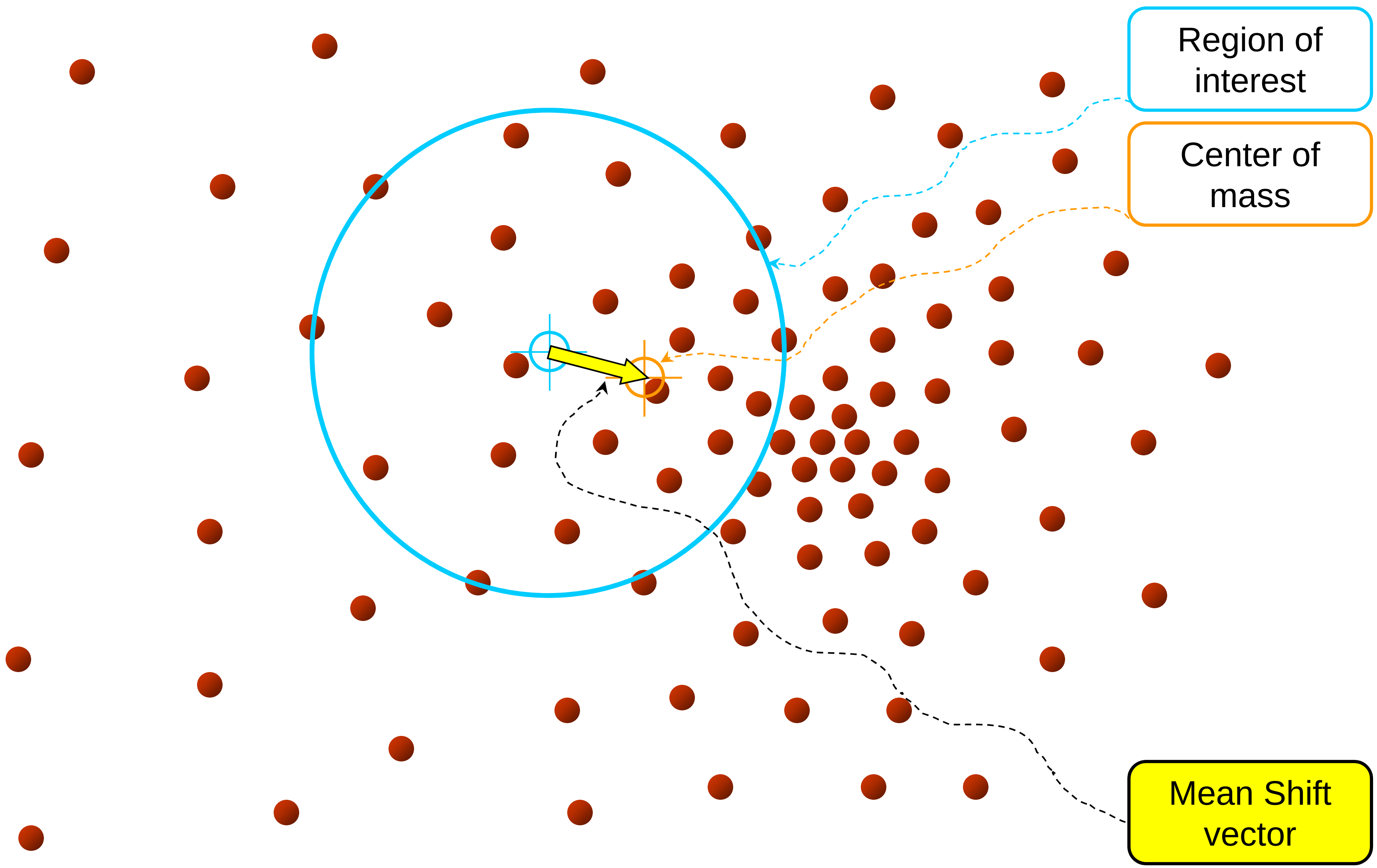
Mean-Shift Theory and Its Applications

Lecture-18

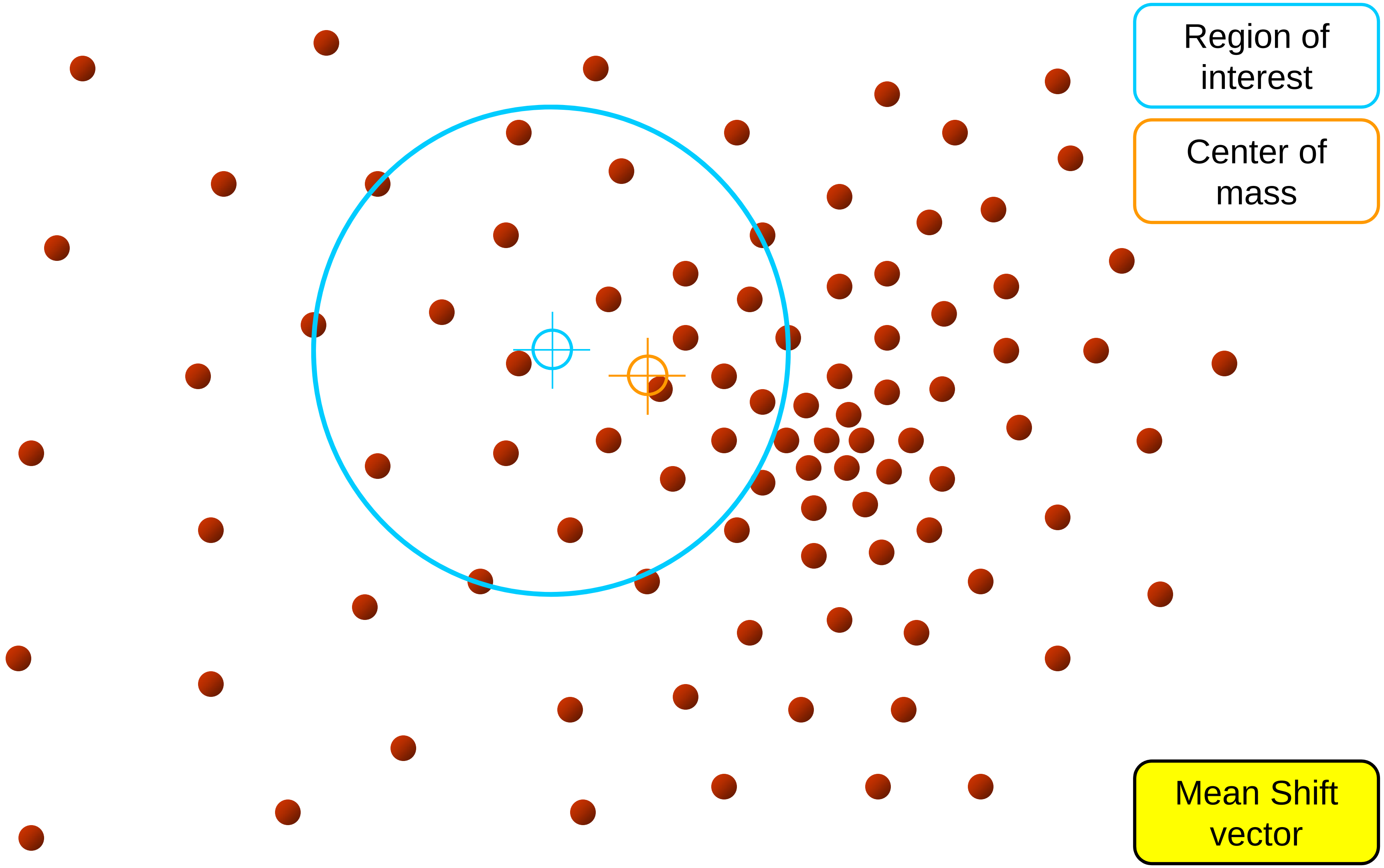
Mean Shift : A robust Approach Toward Feature Space Analysis, by Comaniciu, Meer, IEEE PAMI, Volume 24, No 5, May 2002, pages 603-619.



Objective : Find the densest region

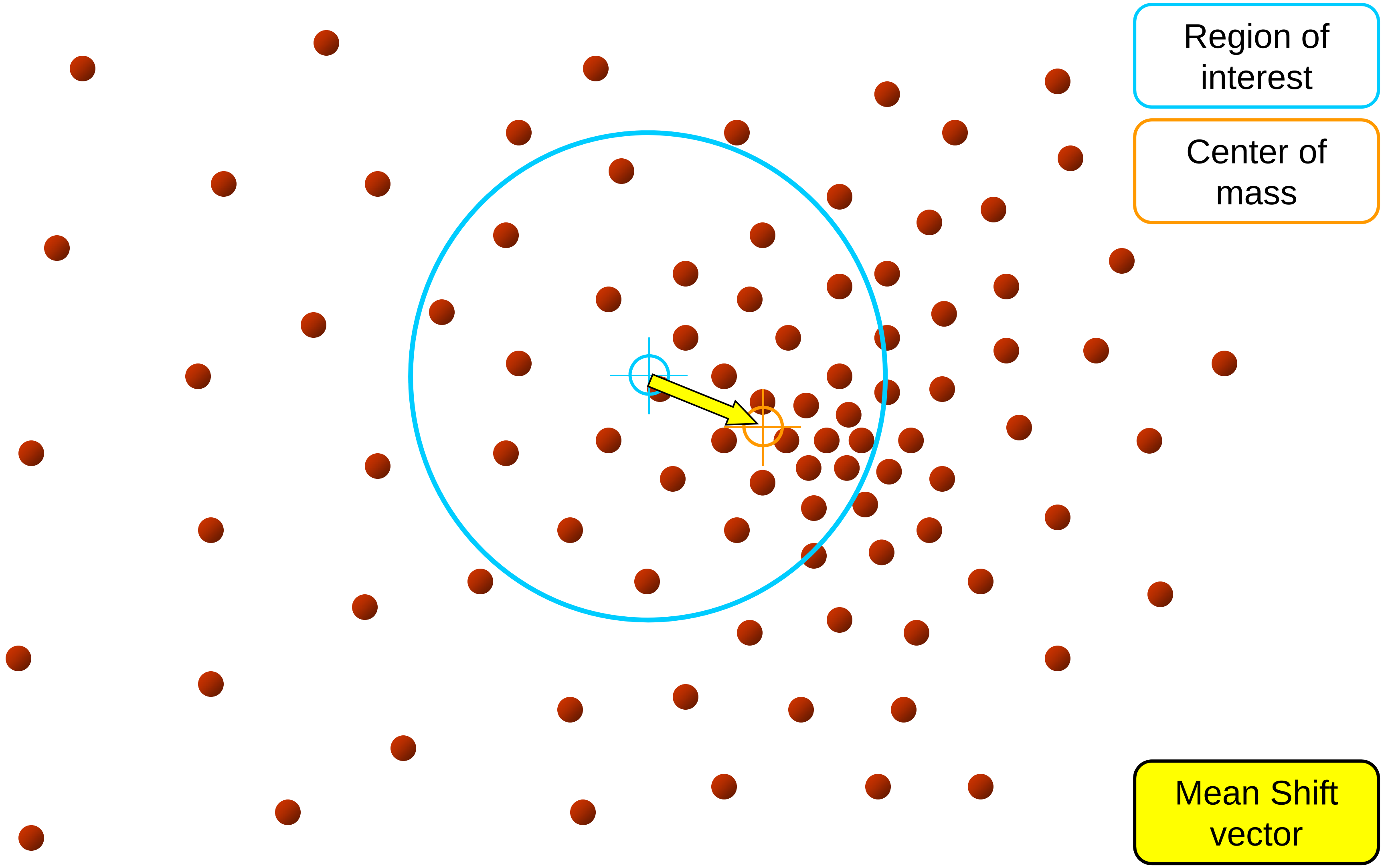


Objective : Find the densest region
Distribution of identical billiard balls



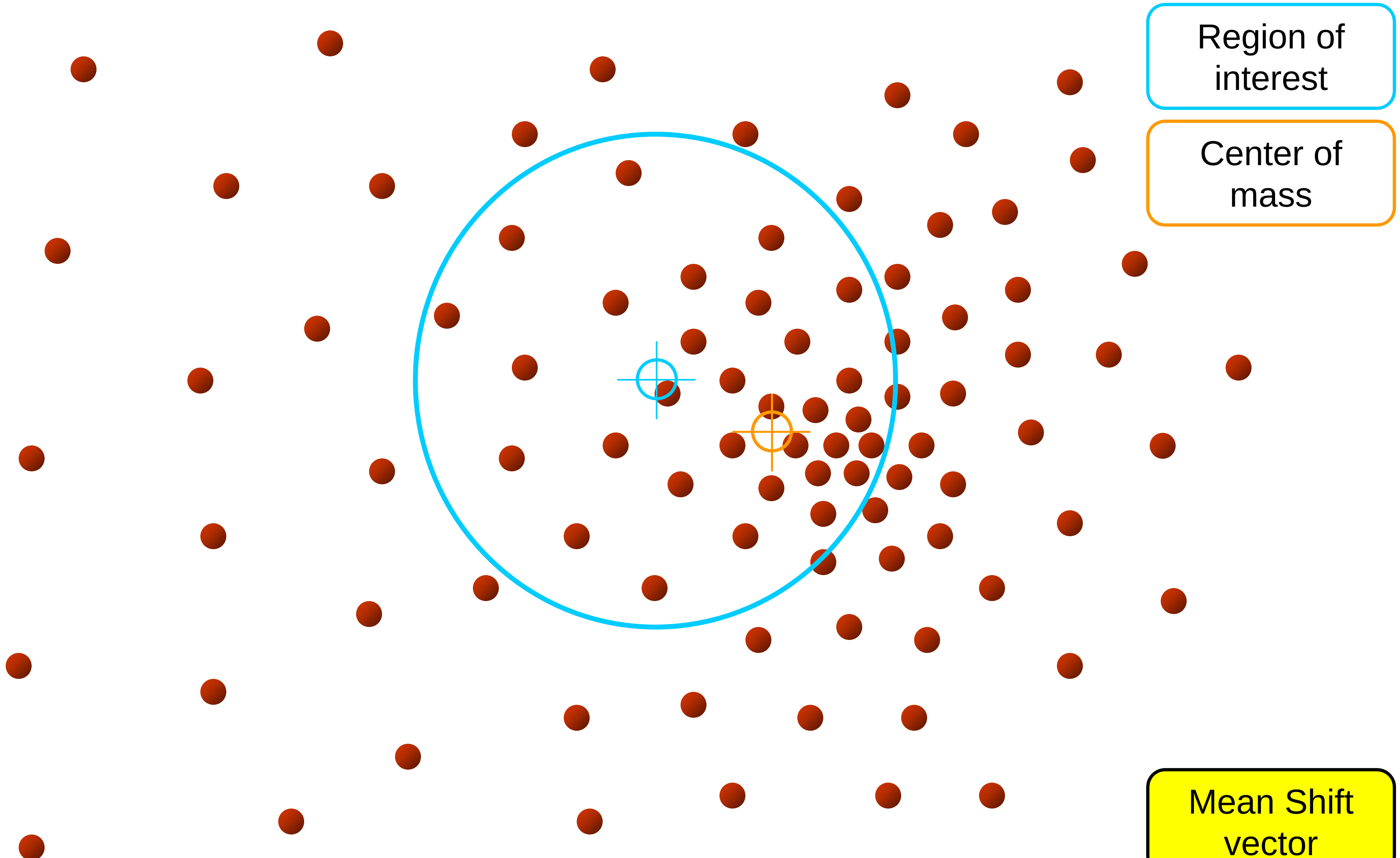
Objective : Find the densest region

Distribution of identical billiard balls

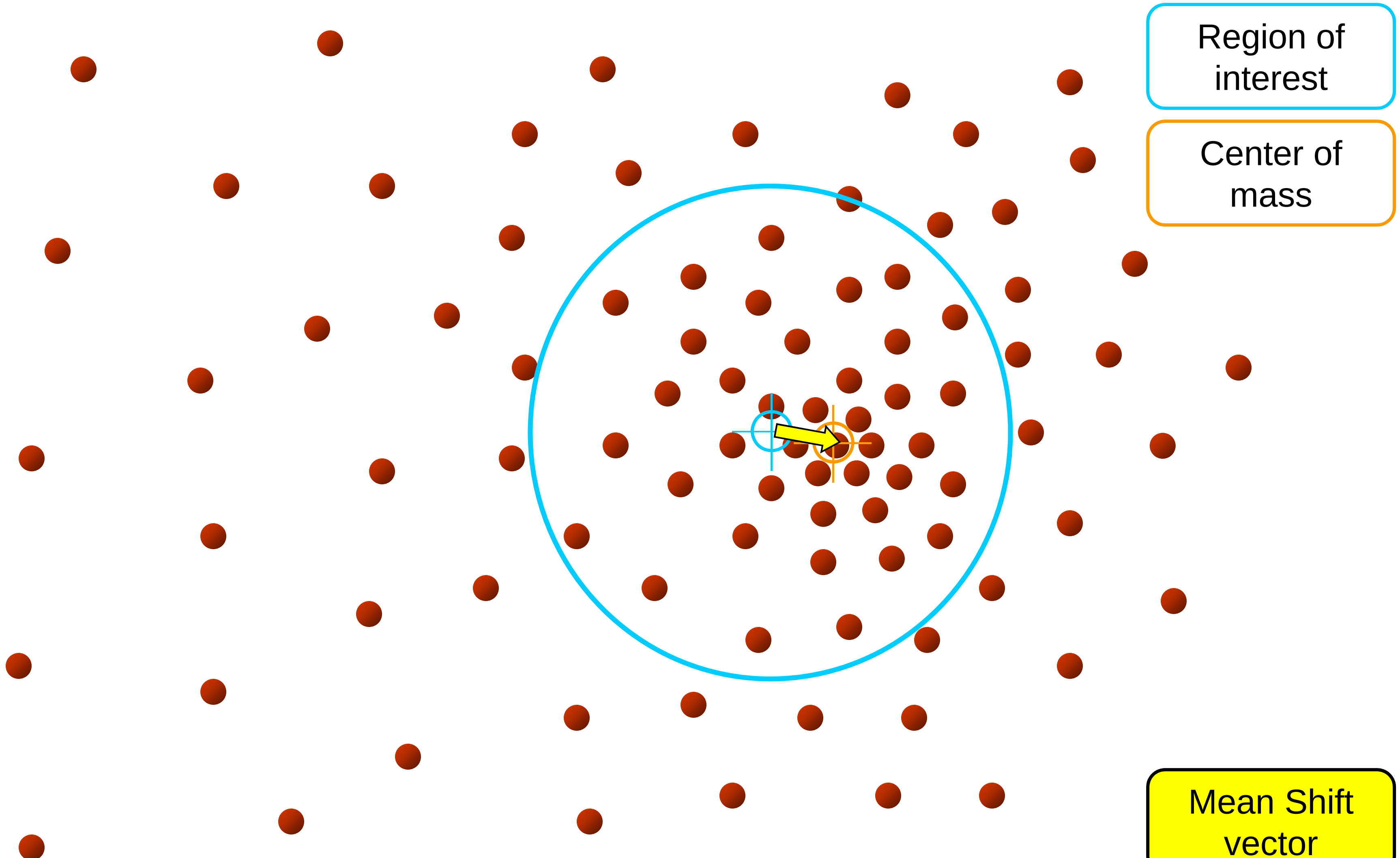


Objective : Find the densest region

Distribution of identical billiard balls

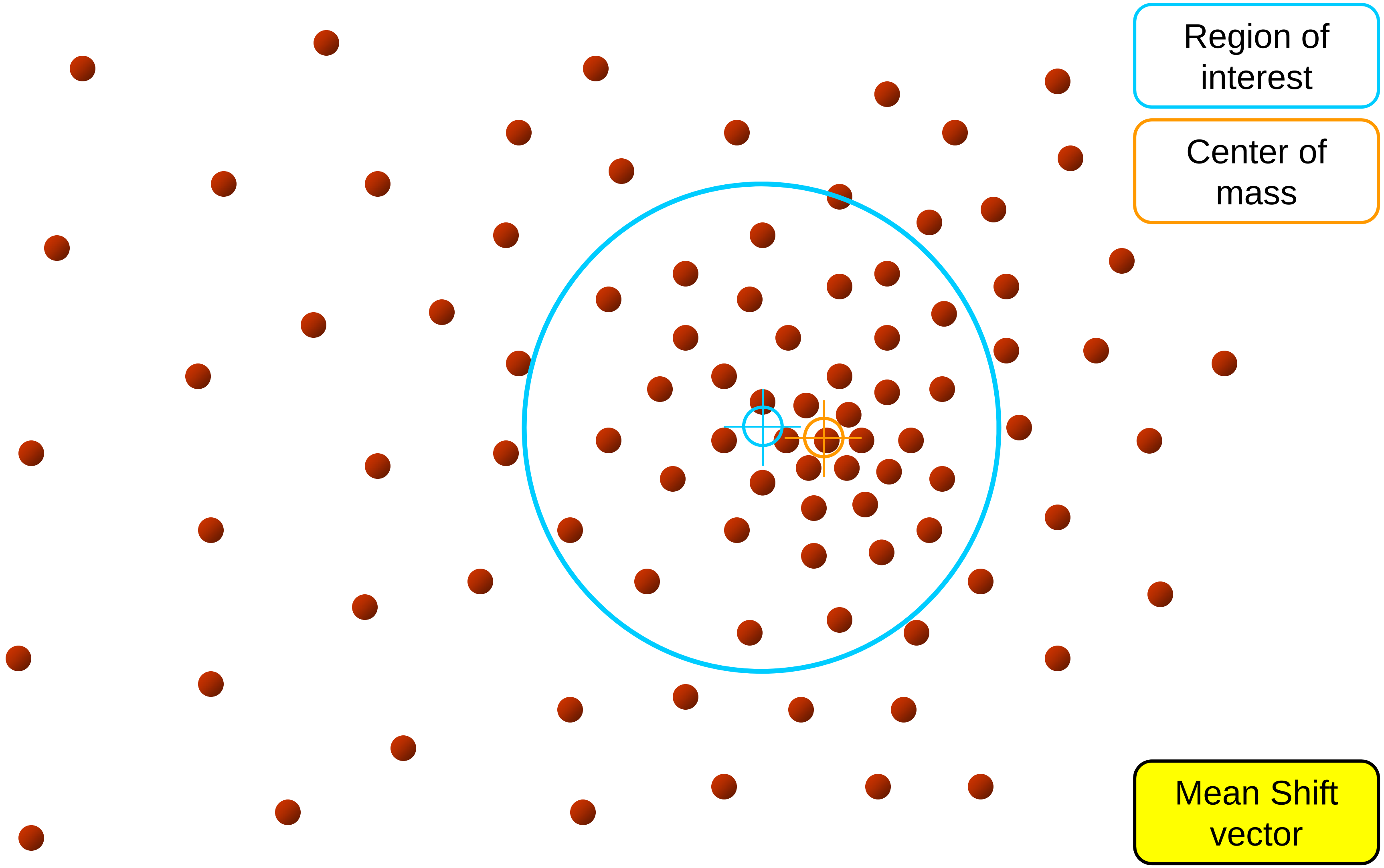


Objective : Find the densest region
Distribution of identical billiard balls



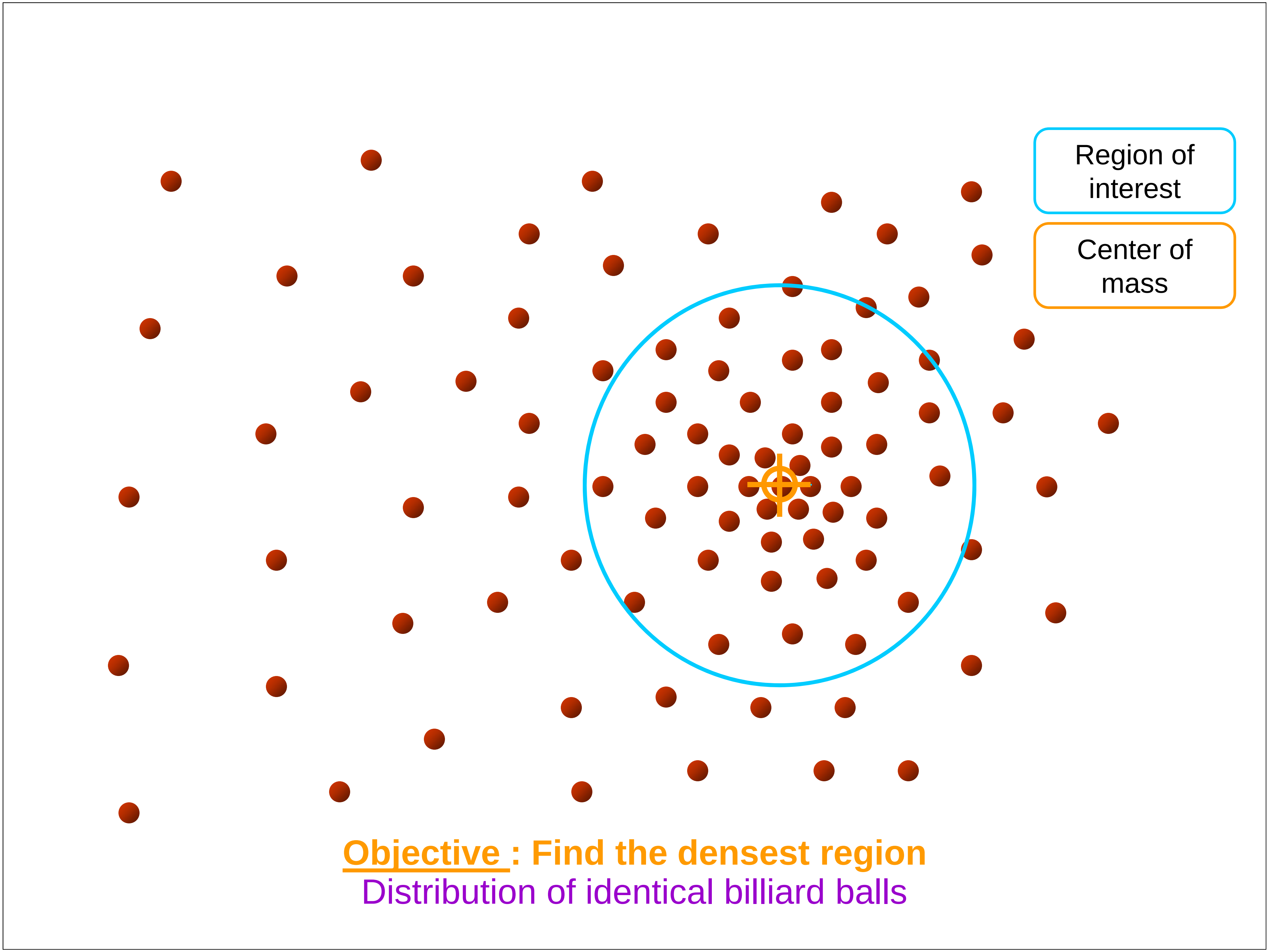
Objective : Find the densest region

Distribution of identical billiard balls



Objective : Find the densest region

Distribution of identical billiard balls


Region of interest
Center of mass
Objective : Find the densest region
Distribution of identical billiard balls

Mean Shift Vector

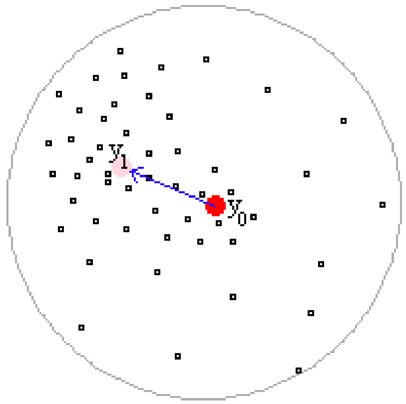
Given:

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.

Mean Shift Vector Example



$$M_h(\mathbf{y}) = \left[\frac{1}{n_x} \sum_{i=1}^{n_x} \mathbf{x}_i \right] - \mathbf{y}_0$$

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

Mean Shift (Weighted)

$$M_h(\mathbf{y}_0) = \left[\frac{\sum_{i=1}^{n_x} w_i(\mathbf{y}_0)\mathbf{x}_i}{\sum_{i=1}^{n_x} w_i(\mathbf{y}_0)} \right] - \mathbf{y}_0$$

n_x : number of points in the kernel
$\mathbf{y}_0$: initial mean location
$\mathbf{x}_i$: data points
h : kernel radius

Weights are determined using kernels (masks):
Uniform, Gaussian or Epanechnikov

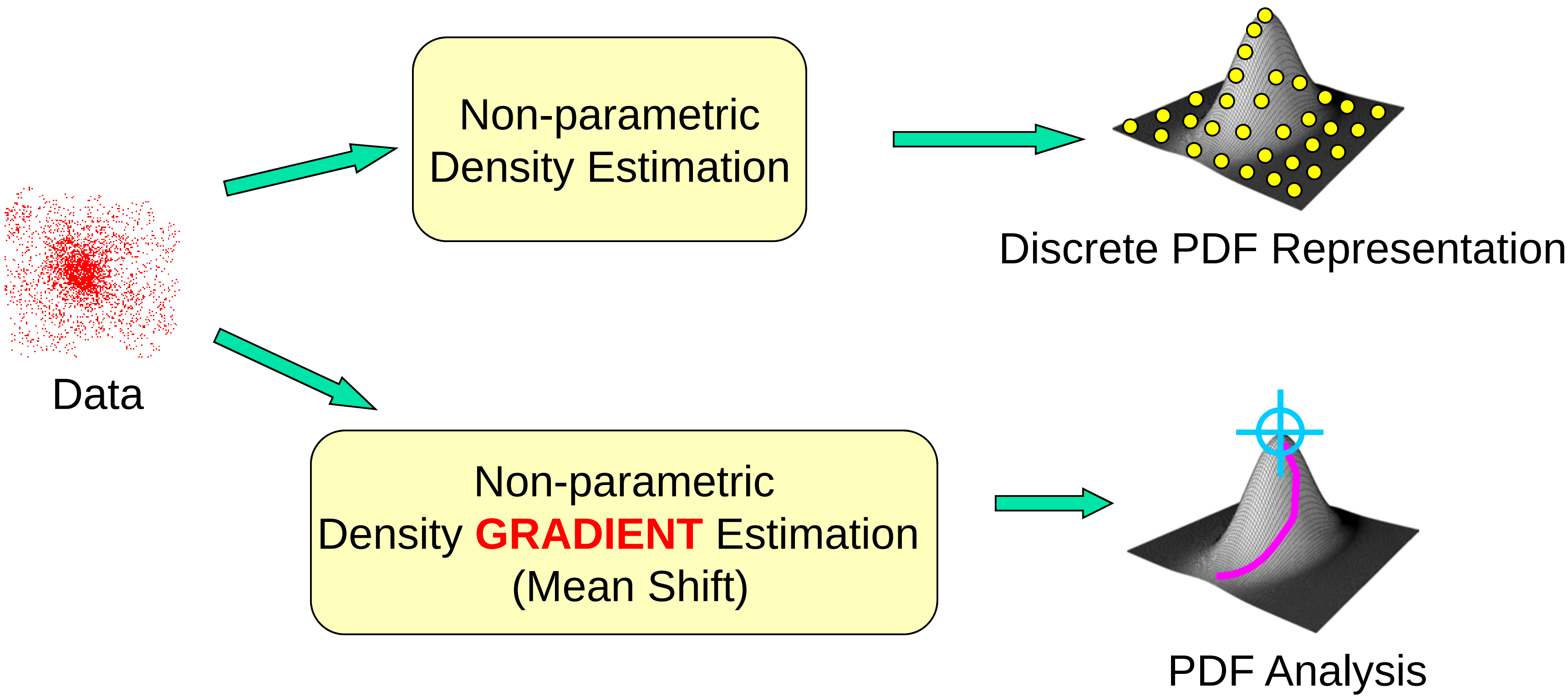
Properties of Mean Shift

- Mean shift vector has the direction of the gradient of the density estimate.

- It is computed iteratively for obtaining the maximum density in the local neighborhood.

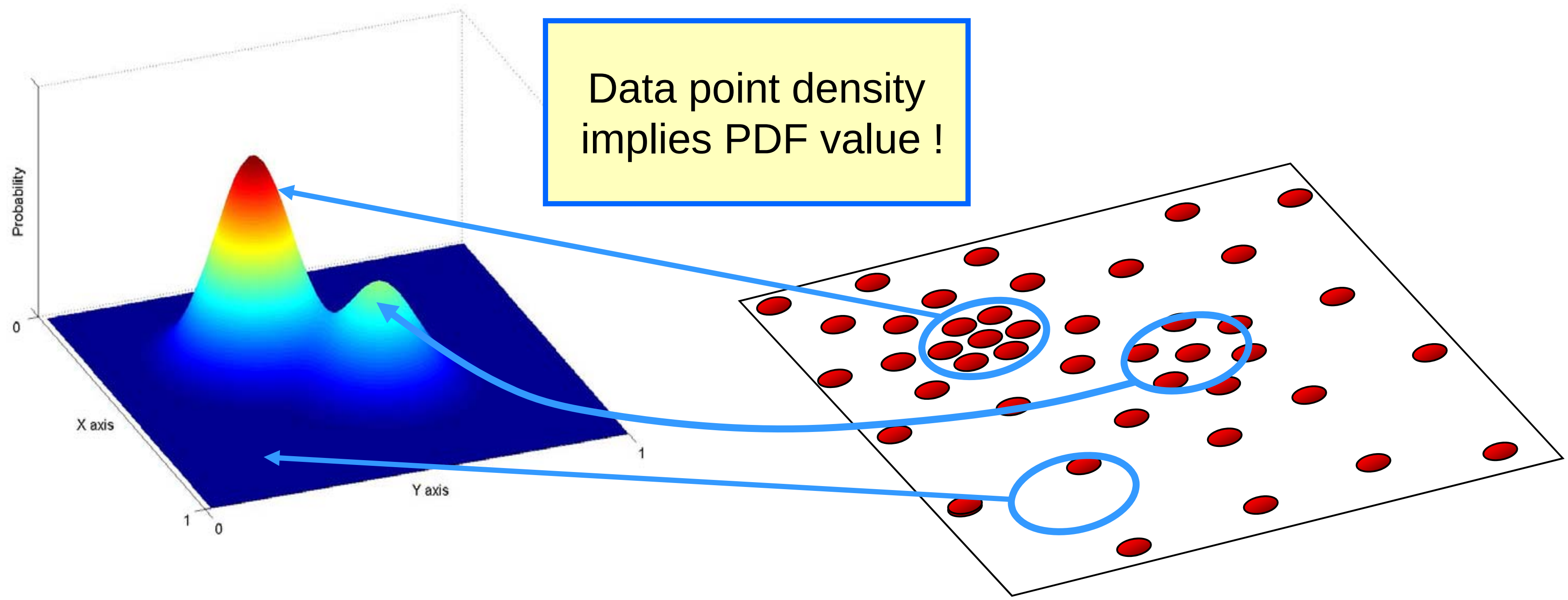
What is Mean-Shift?

- A tool for finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in R_N



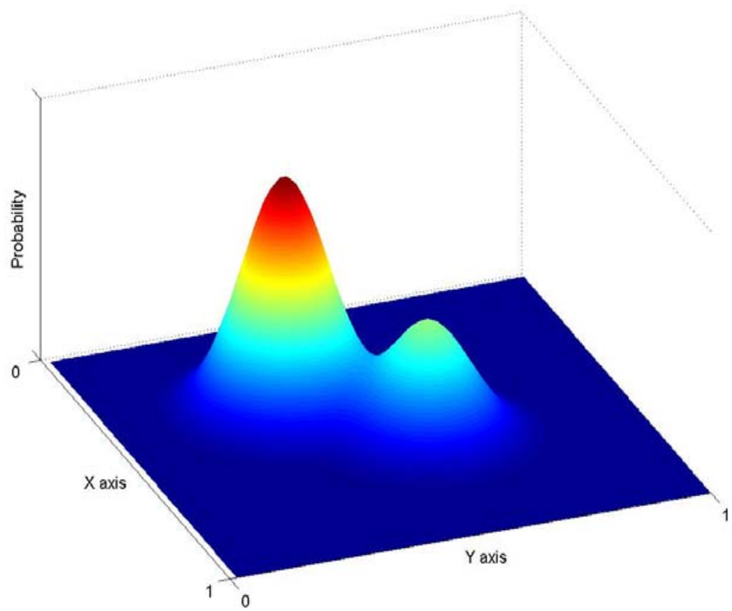
Non-Parametric Density Estimation

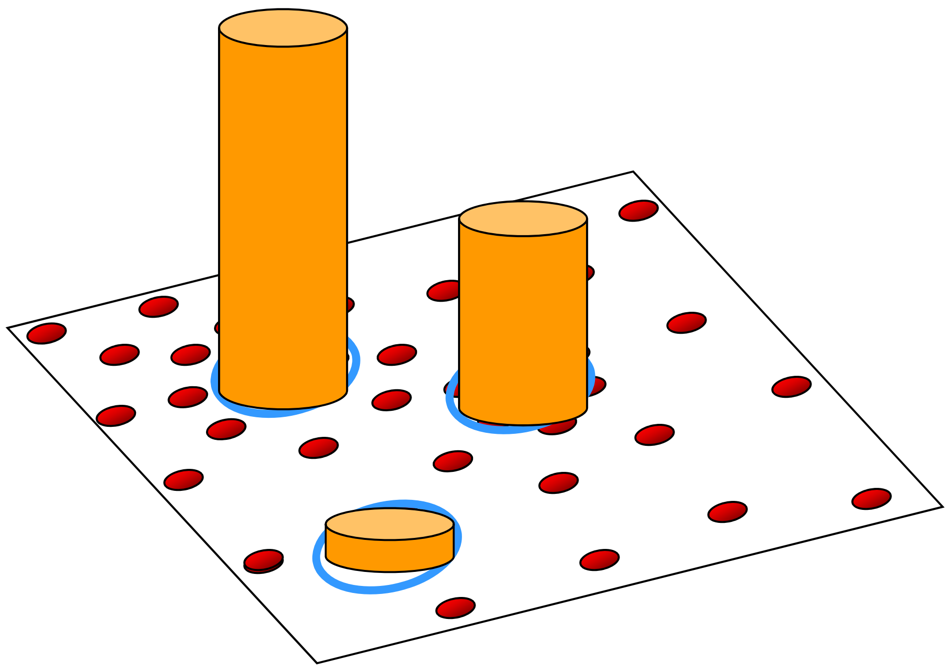
- Assumption : The data points are sampled from an underlying PDF



Assumed Underlying PDF **Real Data Samples**

Non-Parametric Density Estimation

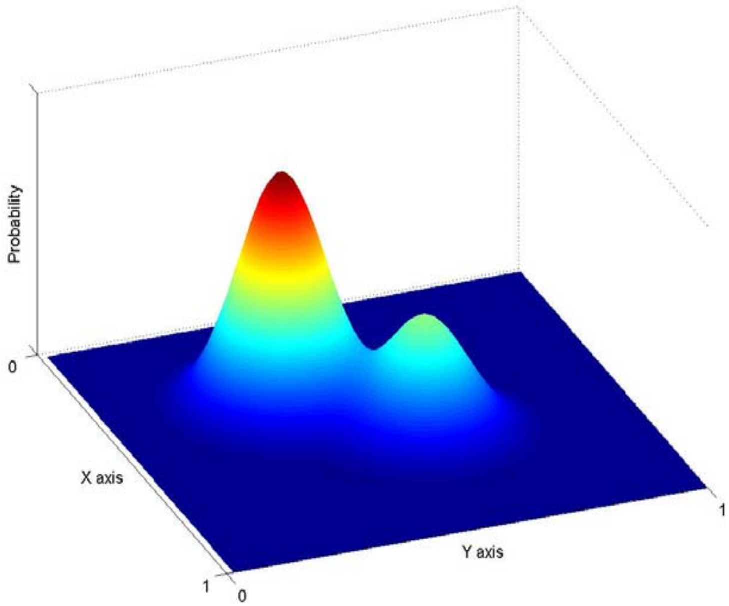




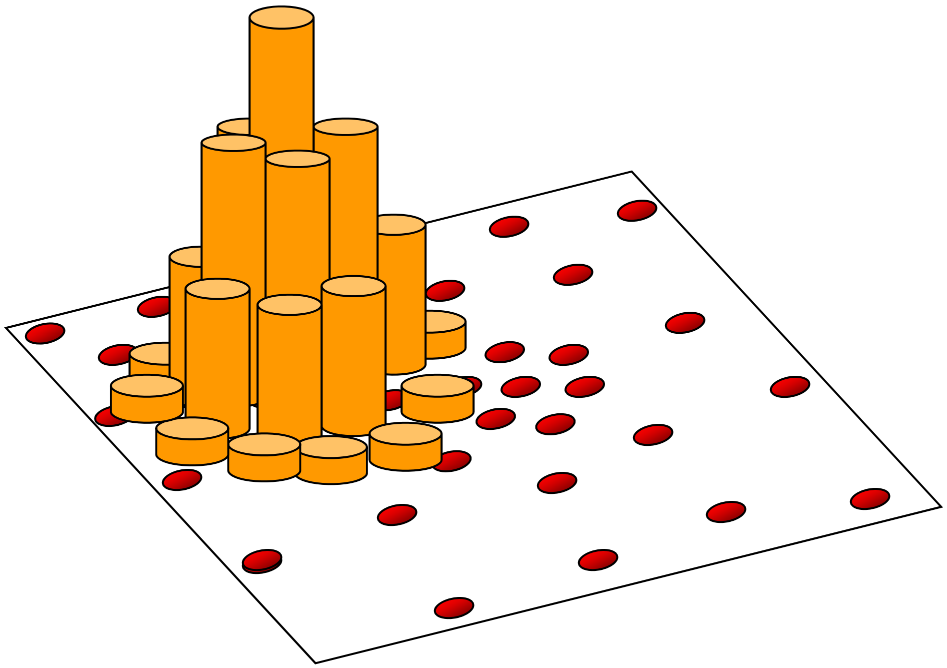
Assumed Underlying PDF

Real Data Samples

Non-Parametric Density Estimation



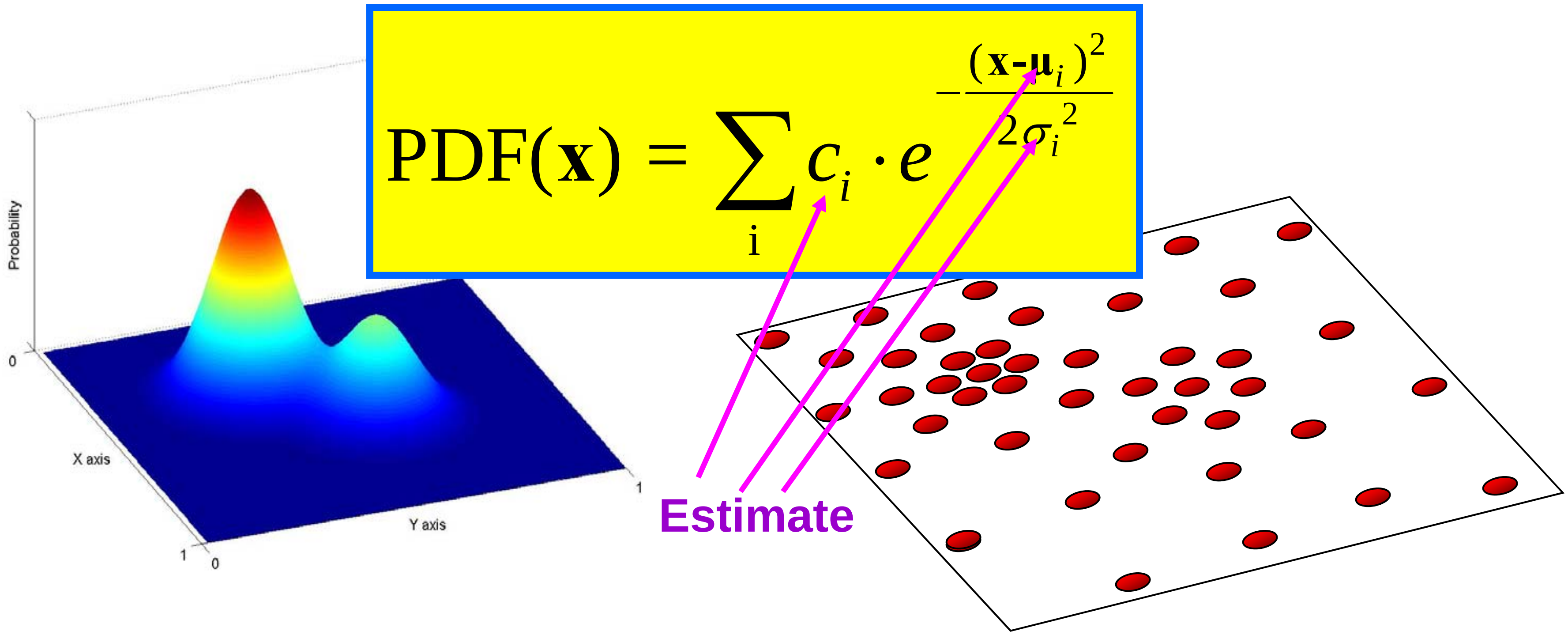
Assumed Underlying PDF



Real Data Samples

Parametric Density Estimation

- Assumption : The data points are sampled from an underlying PDF



Assumed Underlying PDF

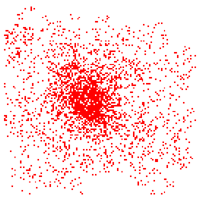
Real Data Samples

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 $x_1 \ldots x_n$

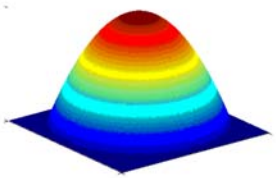


Data

Examples:

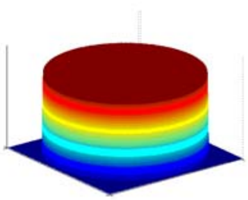
- Epanechnikov Kernel

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



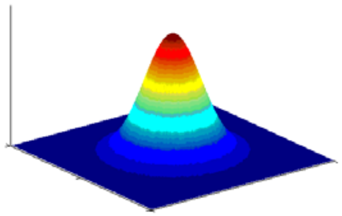
- Uniform Kernel

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



- Normal Kernel

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



Profile and Kernel

Radially symmetric Kernel

$$K(\mathrm{x}) = \mathrm{c}k(\|\mathrm{x}\|^2)$$

Profile

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

Kernel Density Estimation

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

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

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

Kernel Density Estimation

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

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

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

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

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

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

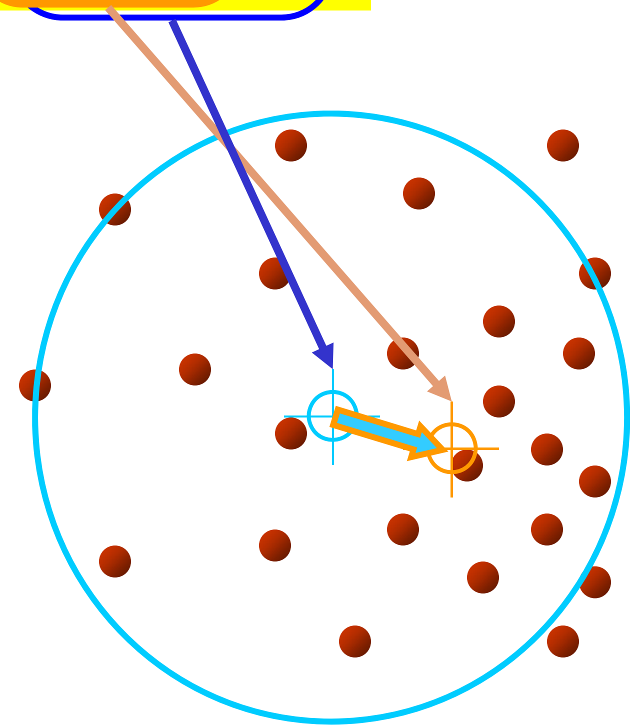
$$\nabla P(\mathrm{x}) = \frac{1}{n} 2c \sum_{i=1}^{n} g_i \left[\frac{\sum_{i=1}^{n} \mathrm{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathrm{x} \right]$$

Computing The Mean Shift

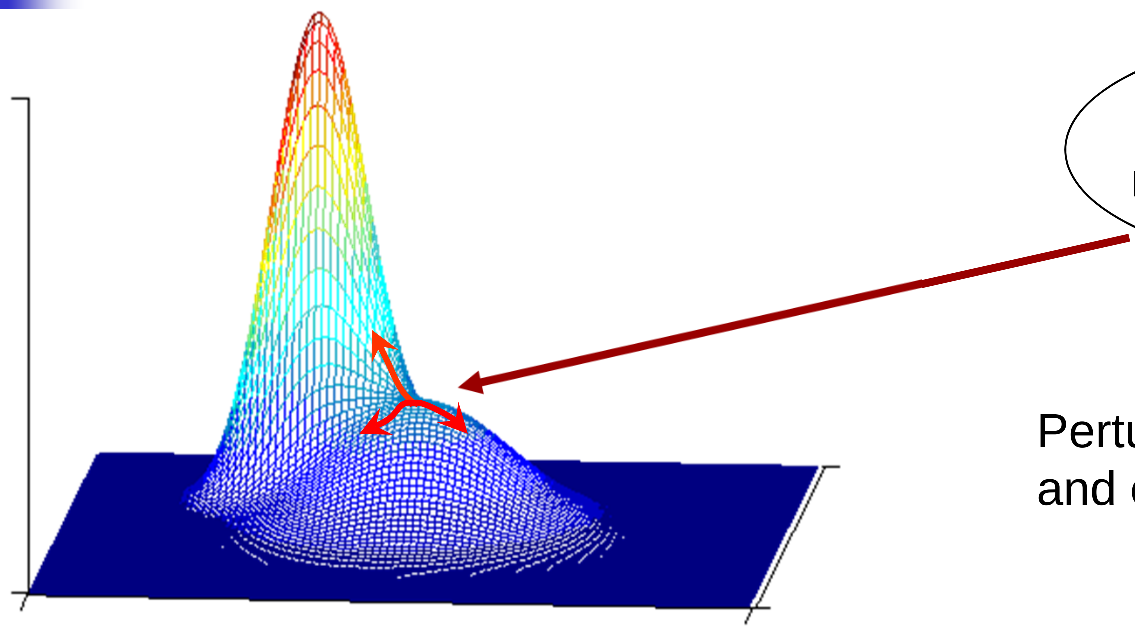
$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right]\left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

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

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



Mean Shift Mode Detection



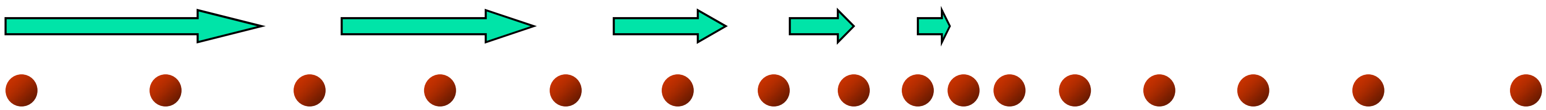
What happens if we reach a saddle point**?**

Perturb the mode position and check if we return back

<u>Updated Mean Shift Procedure:</u>

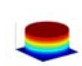
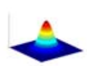
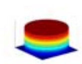
- Find all modes using the Simple Mean Shift Procedure
- Prune modes by perturbing them (find saddle points and plateaus)
- Prune nearby – take highest mode in the window

Mean Shift Properties

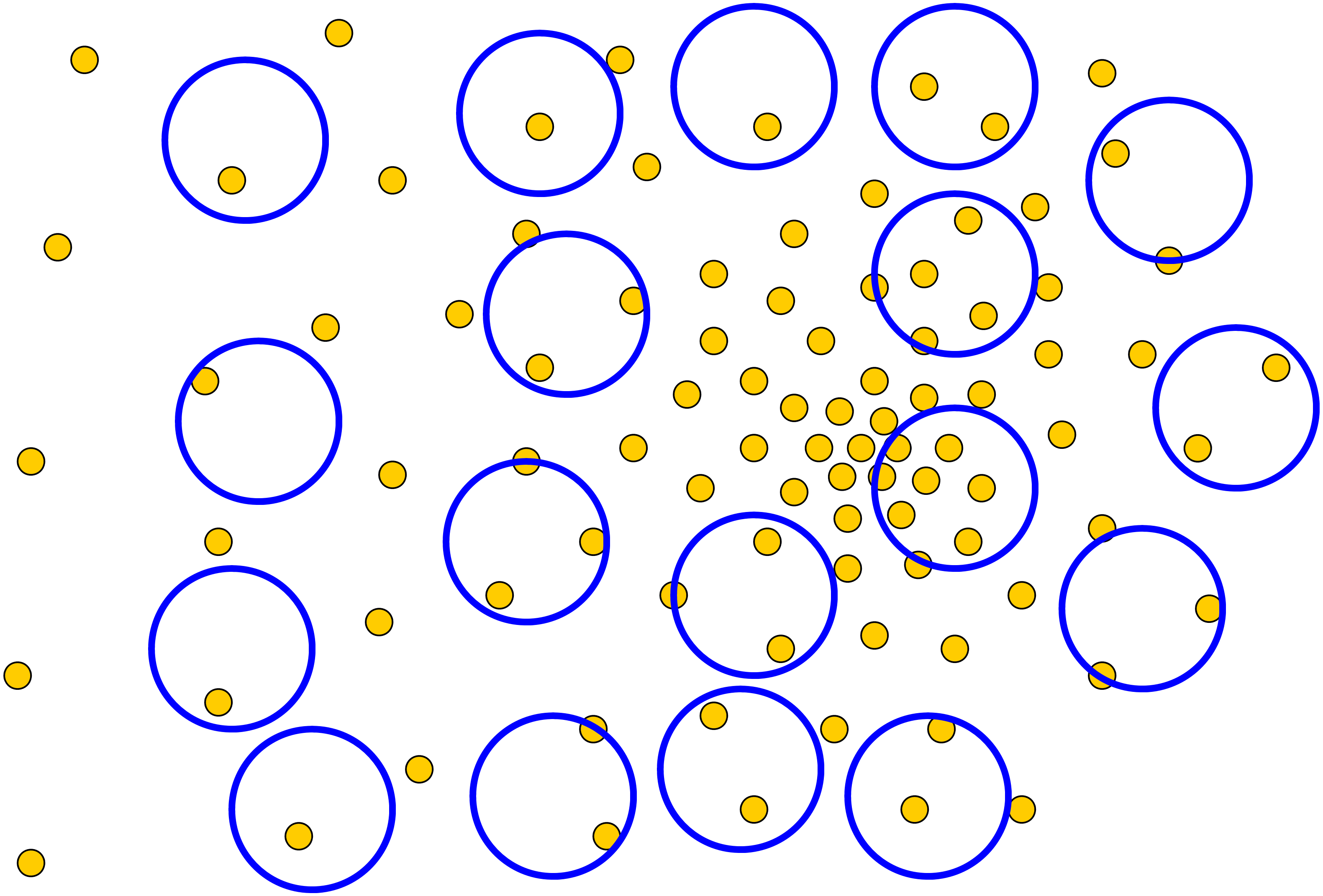


- Automatic convergence speed
  - mean shift vector size depends on gradient.
- Near maxima, the steps are small and refined

} **Adaptive** Gradient Ascent

- Convergence is guaranteed for infinitesimal steps only, (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 ().

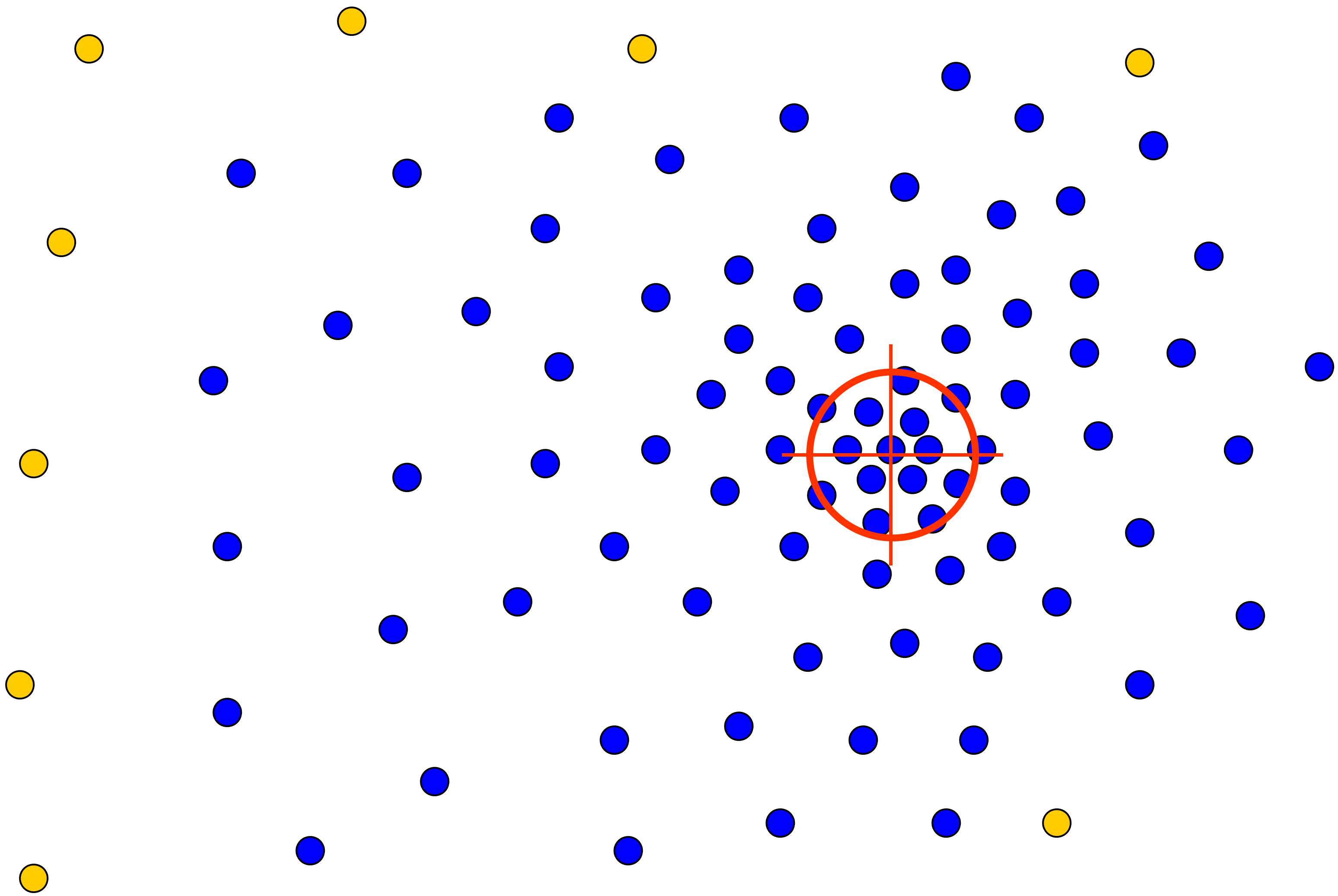
Real Modality Analysis



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

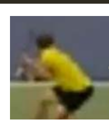
Mean Shift Applications

Mean-Shift Object Tracking

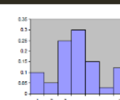
- General Framework

Target Representation

Choose the model in the initial frame → Choose a feature space → Represent the model in the selected feature space



Quantized Color Space → PDF

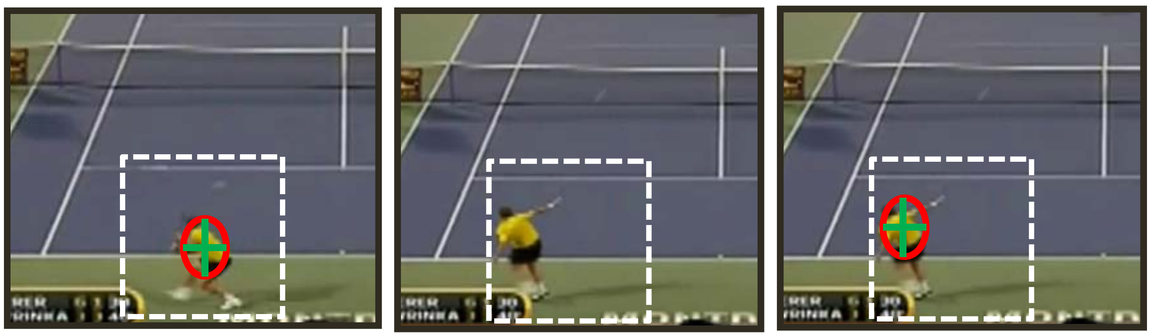


- The object is being modeled using color probability density

Mean-Shift Object Tracking

- General framework

Target Localization-Tracking



Select a ROI around the target location in current frame

Find the most similar candidate based on the similarity func

Mean-Shift Object Tracking

- PDF Representation

Target Model : color distribution by discrete m-bin color histogram

Current Frame

Next Frame

Candidate Model : color distribution by discrete m-bin color histogram

$$\vec{q} = \{q_u\}_{u=1,\ldots,m} \qquad \sum_{u=1}^{m} q_u = 1$$

$$f(y) = f[\vec{q}, \overrightarrow{p(y)}]$$

Similarity Funcion → ?

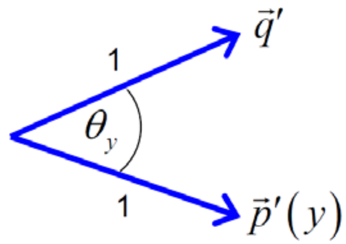
$$\overrightarrow{p(y)} = \{p_u(y)\}_{u=1,\ldots,m} \qquad \sum_{u=1}^{m} p_u = 1$$

The Bhattacharyya Coefficient

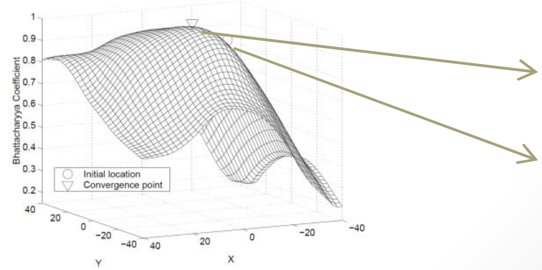
Mean-Shift Object Tracking

- The Bhattacharyya Coefficient
  - Measures similarity between object model *q* and color *p* of target at location *y*

$$\rho(p(y), q) = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}$$



  - ρ is the cosine of vectors $(\sqrt{p_1}, \dots, \sqrt{p_m})^T$ and $(\sqrt{q_1}, \dots, \sqrt{q_m})^T$.
  - Large ρ means good match between candidate and target model
  - In order to find the new target location we try to maximize the Bhattacharyya coefficient



Convergence Point

Initial Location

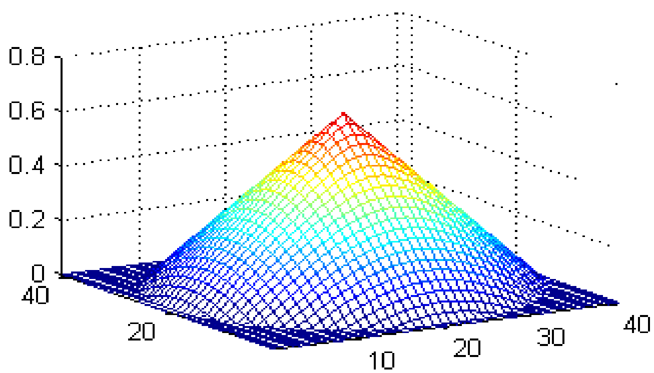
Target Model for Tracking

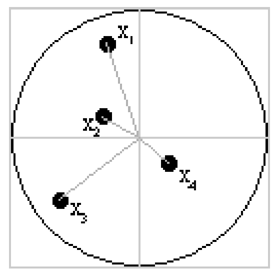
- Features used for tracking include:
  - Gray level
  - Color
  - Gradient

- Feature probability distribution are calculated by using weighted histograms.

- The weights are derived from Epanechnikov profile.

Distribution



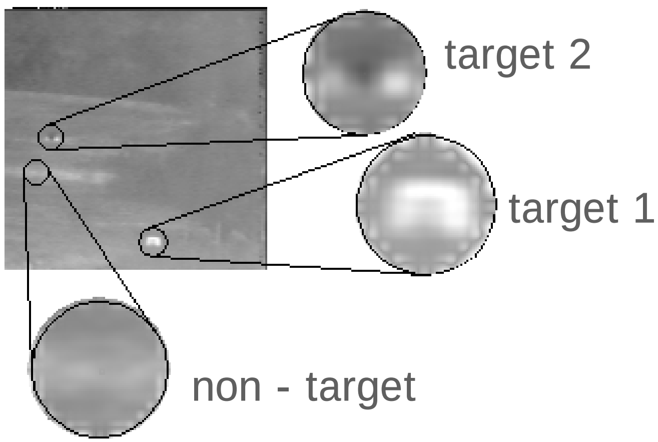


x_1, x_2, x_3, x_4 have the same feature, such as gray level.

$$p(u) = C \sum_{\mathbf{x}_i \in S} k\left(\|\mathbf{x}_i\|^2\right) \delta[\ S(\mathbf{x}_i) - u]$$

$S(\mathbf{x}_i)$ is the color at *x_i*

Target Gray Level Feature



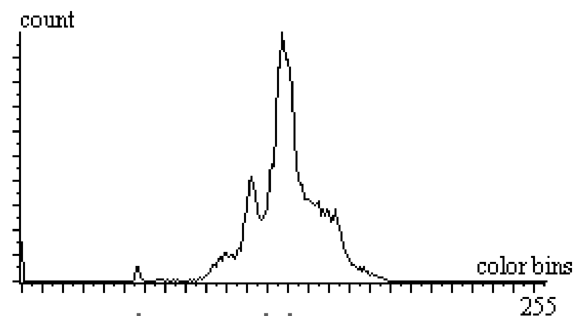
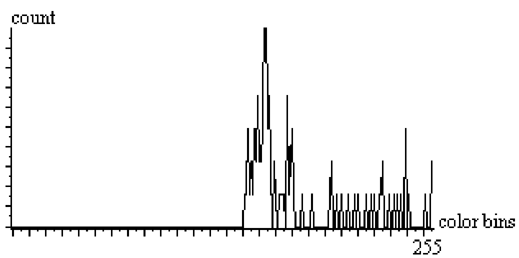
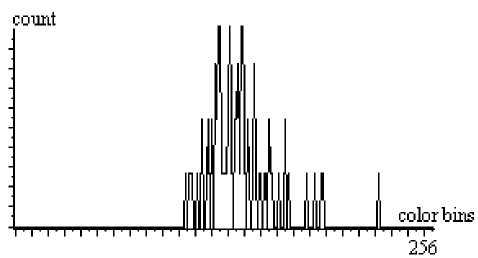


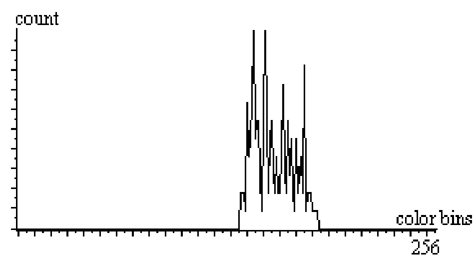
image histogram



target 1 distribution



target 2 distribution



non target distribution

Similarity of Target and Candidate Distributions

Target : q_u.
Candidate : $\hat{p}_u$.

$$d(\mathbf{y}) = \sqrt{1 - \rho(\mathbf{y})}$$

$$\rho(\mathbf{y}) = \rho[\hat{p}(\mathbf{y}), q] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y}) q_u}$$

$\rho(\mathbf{y})$:Bhattacharya coefficient.

Distance Minimization

Minimizing the distance corresponds to *maximizing* Bhattacharya coefficient.

$$\rho[\hat{p}(\mathbf{y}), q] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y}) q_u}$$

Taylor expansion around $\hat{p}(\mathbf{y}_0)$

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \mathbf{q}] \cong \rho[\hat{\mathbf{p}}(\mathbf{y}_0), \mathbf{q}] + \frac{1}{2} \sum_{i=1}^{m} \hat{p}_u(\mathbf{y}) \sqrt{\frac{q_u}{\hat{p}_u(\mathbf{y}_0)}}$$

Maximizing Bhattacharya coefficient can be obtained by ***maximizing the blue term***.

Likelihood Maximization

$$\rho[\hat{\mathbf{p}}(\mathbf{y}),\mathbf{q}] \cong \rho[\hat{\mathbf{p}}(\mathbf{y}_0),\mathbf{q}] + \frac{1}{2}\sum_{i=1}^{m}\hat{p}_u(\mathbf{y})\sqrt{\frac{q_u}{\hat{p}_u(\mathbf{y}_0)}}$$

↓

$$\frac{C_h}{2}\sum_{i=1}^{n_x}\left[\sum_{u=1}^{m}\delta[S(\mathbf{x}_i)-u]\sqrt{\frac{q_u}{\hat{p}_u(\mathbf{y}_0)}}\right]k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|\right)$$

h : radius of sphere
C_h : normalization constant
$S(x_i)$: gray level at x
y : kernel center
m : number of bins

likelihood maximization depends on maximizing w_i.

Likelihood Maximization Using Mean Shift Vector

Maximization of the likelihood of target and candidate depends on the weights:

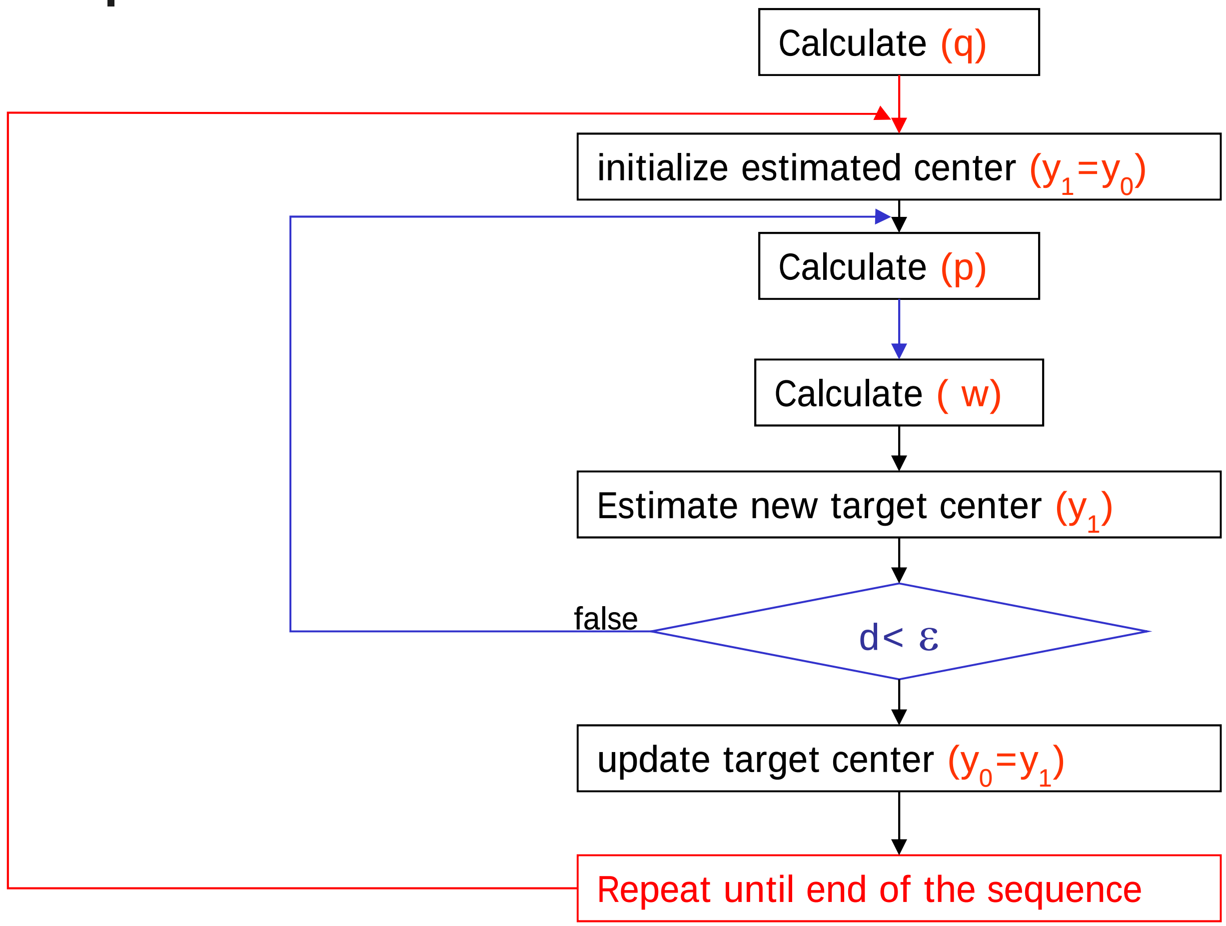
$$w_i(\mathbf{y}_o) = \sum_{u=1}^{m} \delta[S(\mathbf{x}_i) - u] \sqrt{\frac{q_u}{\hat{p}_u(\mathbf{y}_o)}} \quad where \quad 0 \le w_i \le 1$$

Since $\sum_{i=1}^{n_x} w_i(\mathbf{y}_0)$ is strictly positive, mean shift vector can be written as

$$M_h(\mathbf{y}_0) = \frac{\sum_{i=1}^{n_x} w_i(\mathbf{y}_0)\mathbf{x}_i}{\sum_{i=1}^{n_x} w_i(\mathbf{y}_0)} - \mathbf{y}_0$$

Thus, new target center is $\hat{\mathbf{y}} = \mathbf{y}_0 + M_h(\mathbf{y}_0)$

Algorithm


Calculate (q)
initialize estimated center $(y_1=y_0)$
Calculate (p)
Calculate (w)
Estimate new target center (y_1)
false
$d< \varepsilon$
update target center $(y_0=y_1)$
Repeat until end of the sequence

Tracking A Single Point

References

- D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In IEEE Proc. on Computer Vision and Pattern Recognition on, pages673–678, 2000.

- D. Comaniciu, V. Ramesh, and P. Meer. Mean shift: A robust approach towards feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(5):603–619, 2002.