[2016년 1학기 확률및통계]

Mean Shift
Theory and Applications

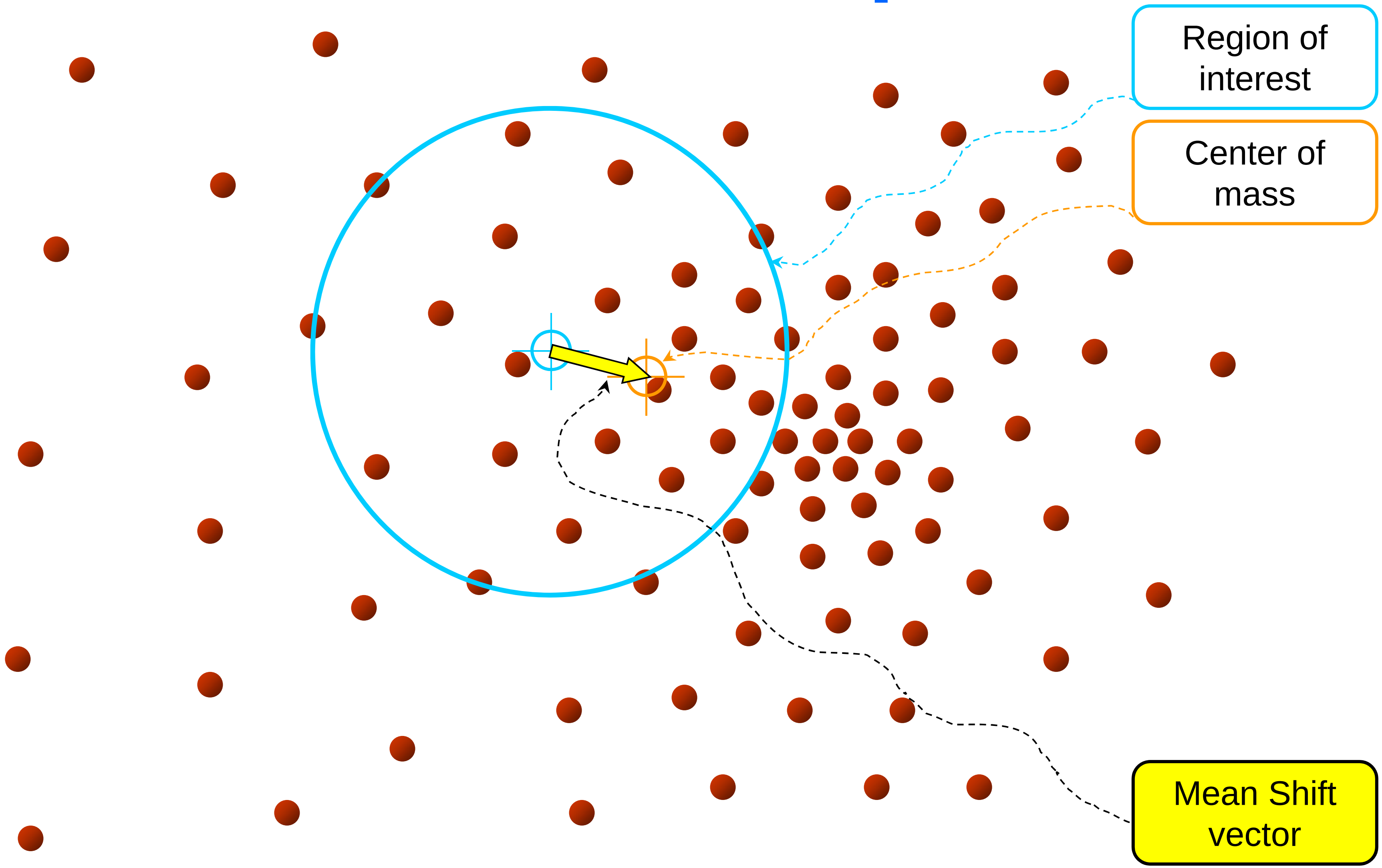
Reference

D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” IEEE T. PAMI, vol. 24, no. 5, pp. 603-619, May 2002.

이상화

2016년 4월 21일

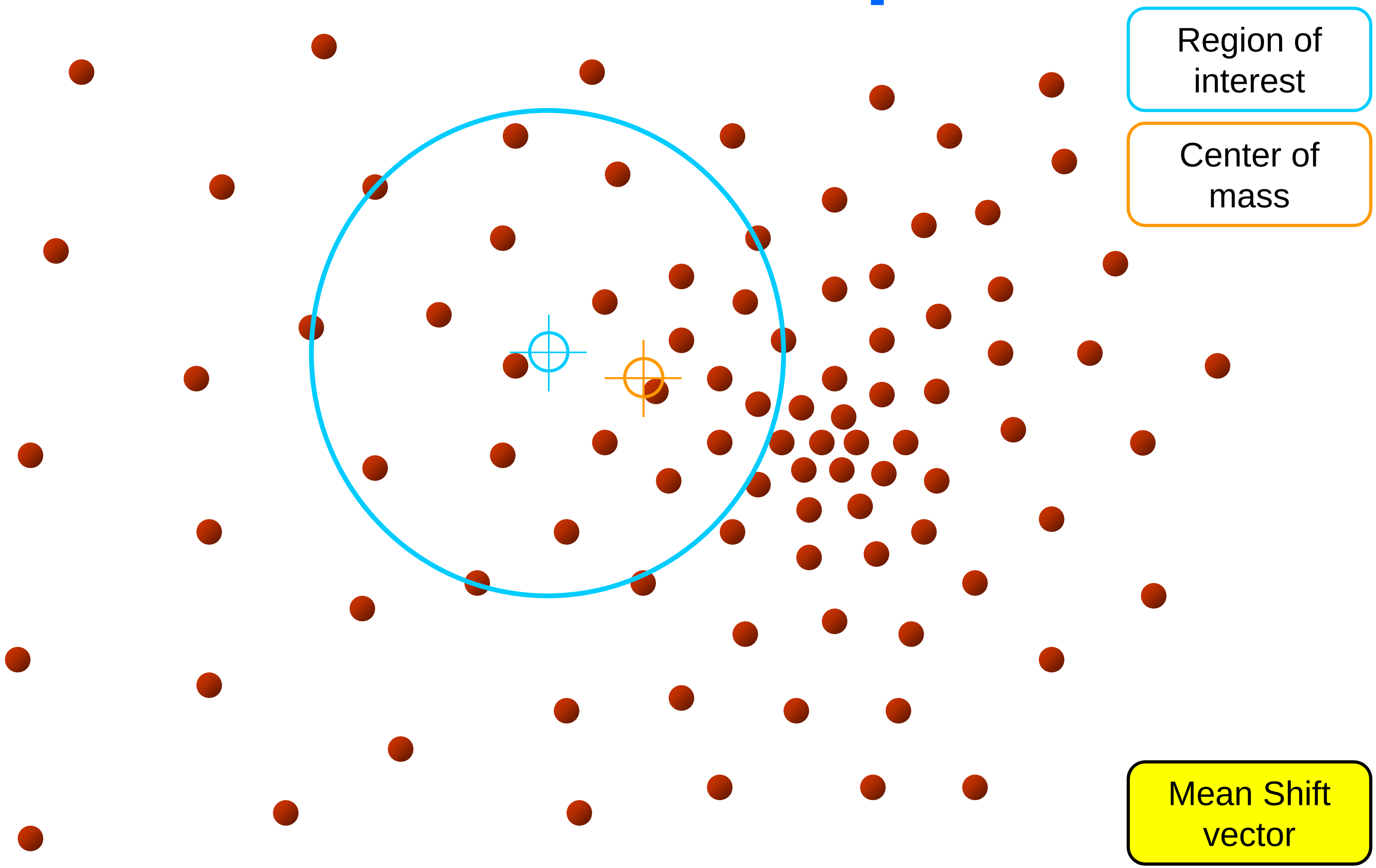
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

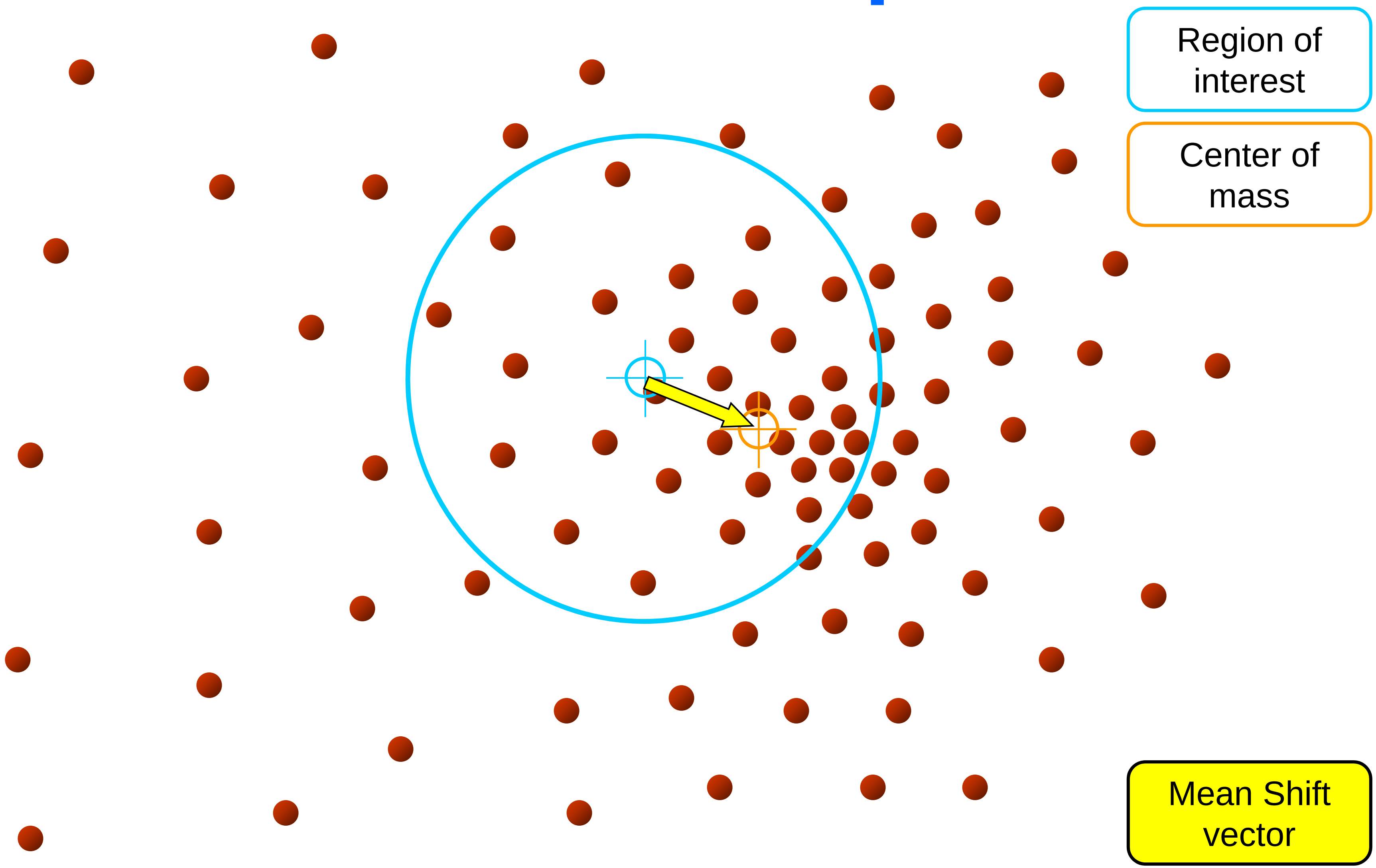
Intuitive Description



<u>Objective</u> : Find the densest region

Distribution of identical billiard balls

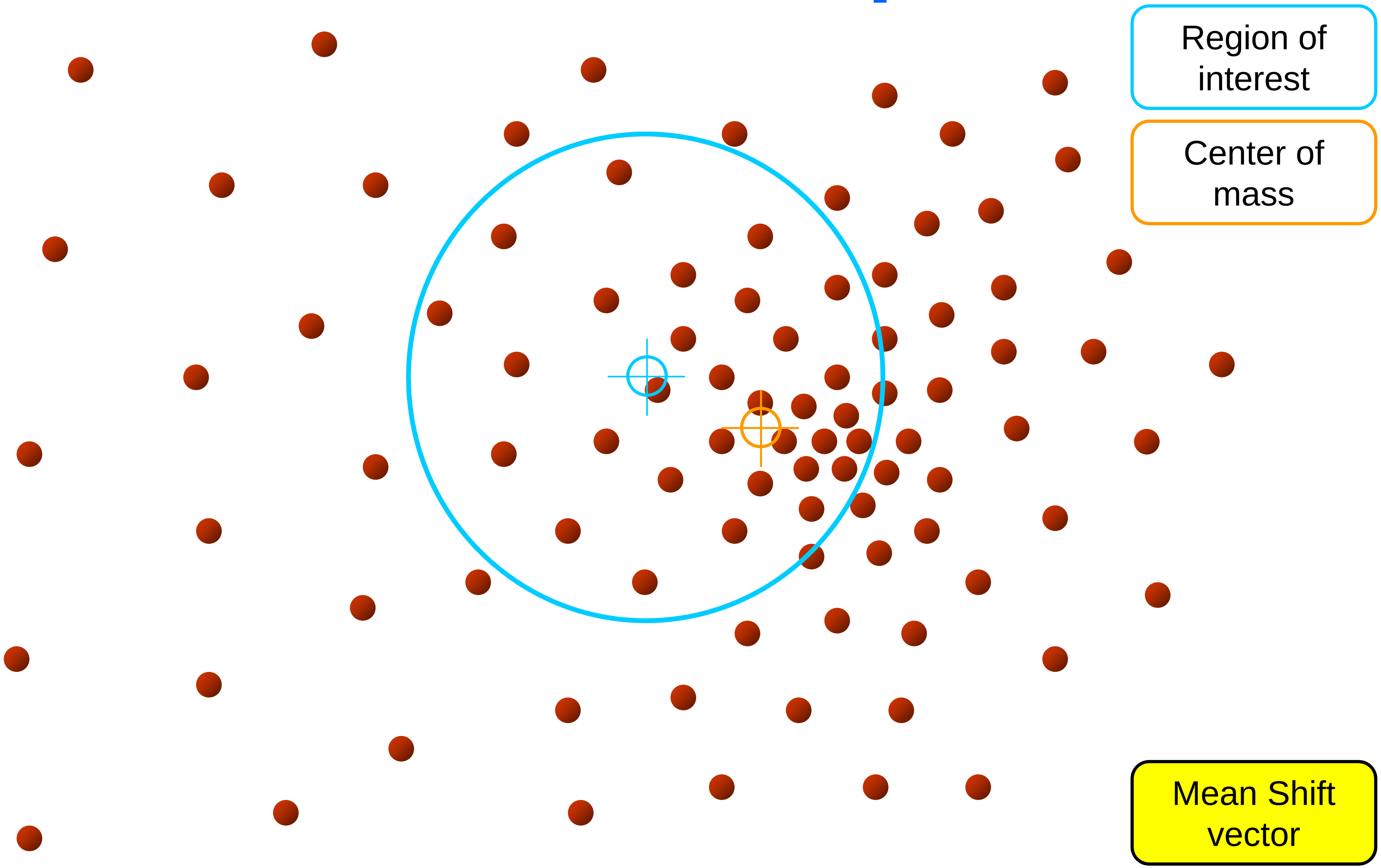
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

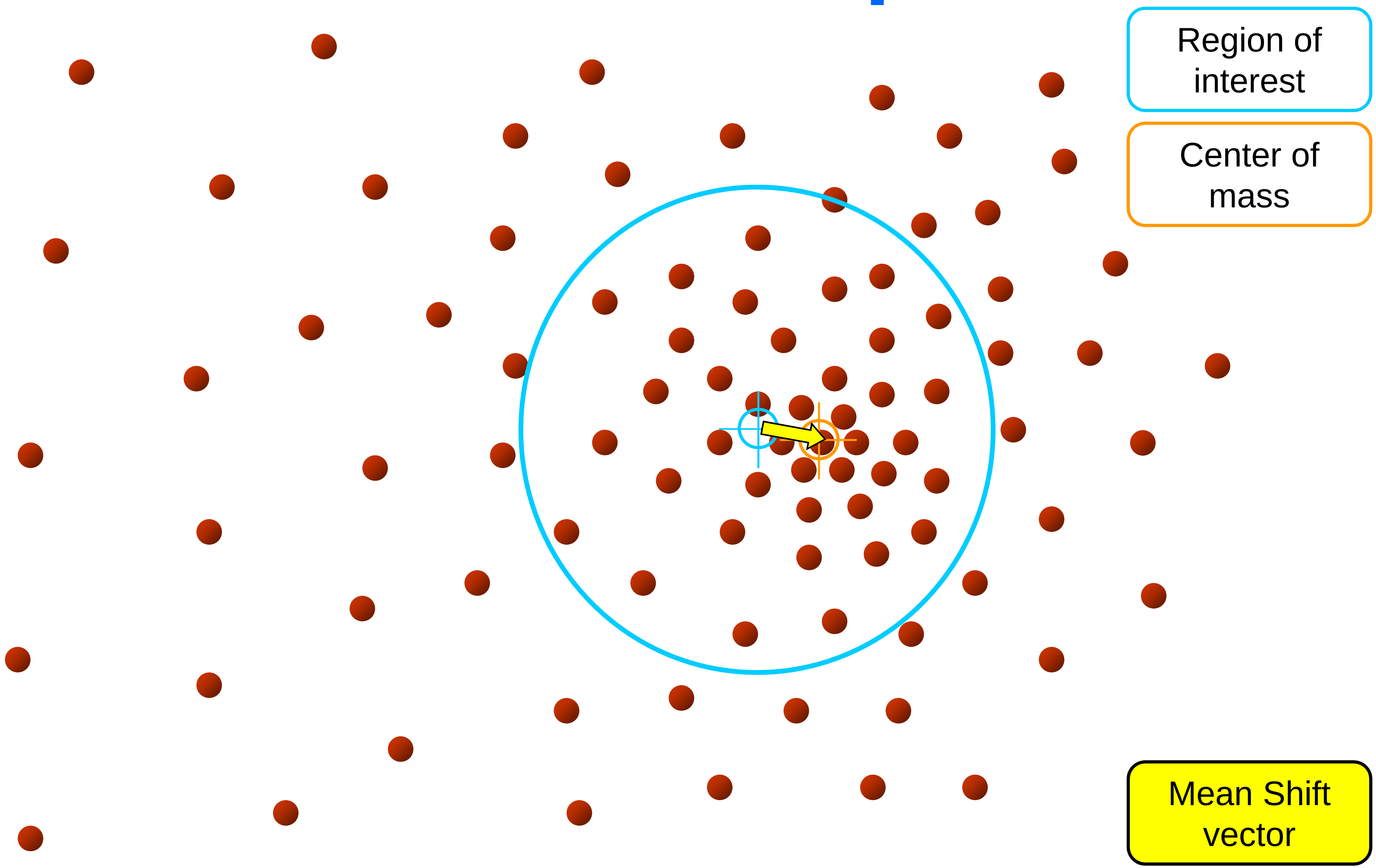
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

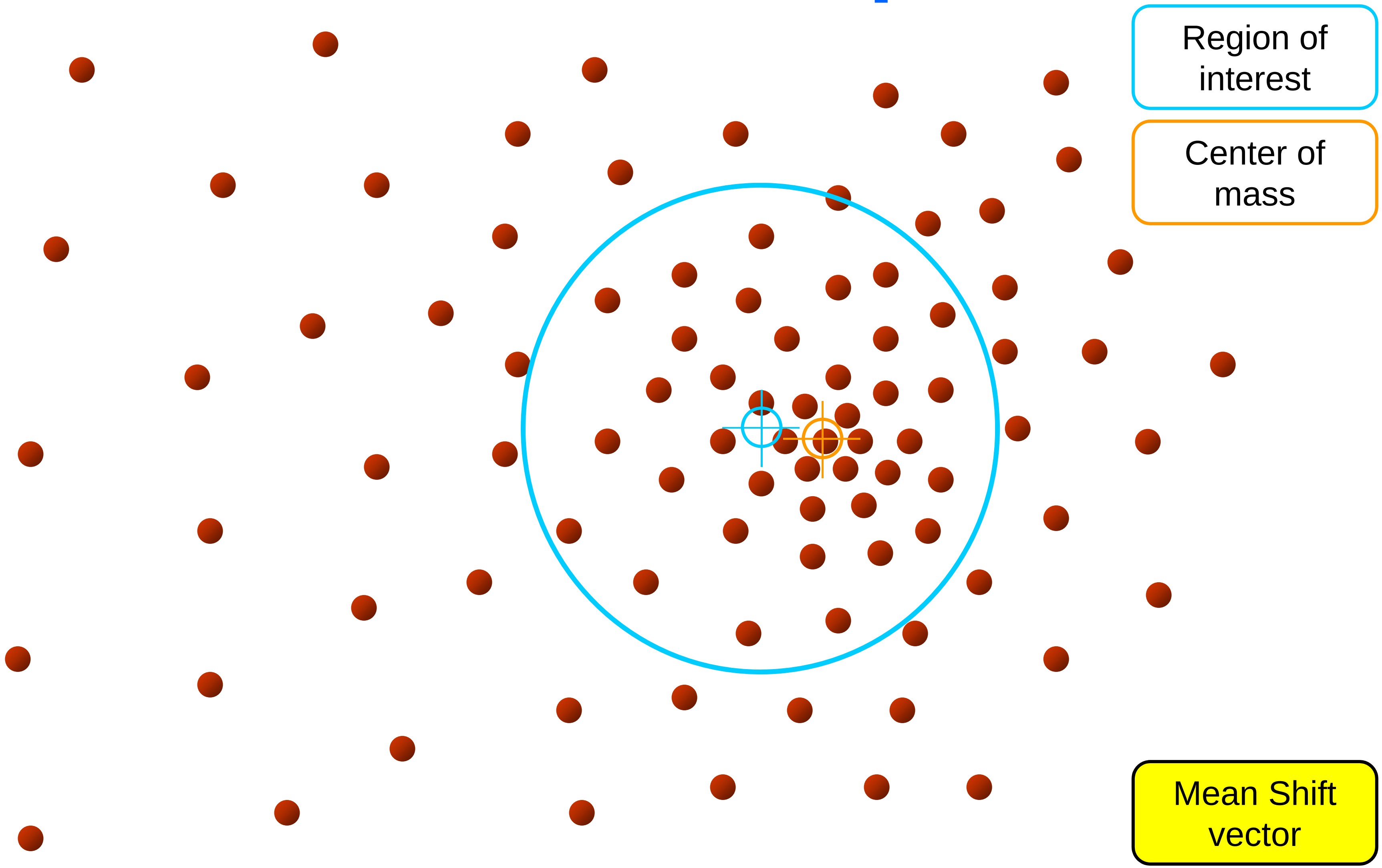
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

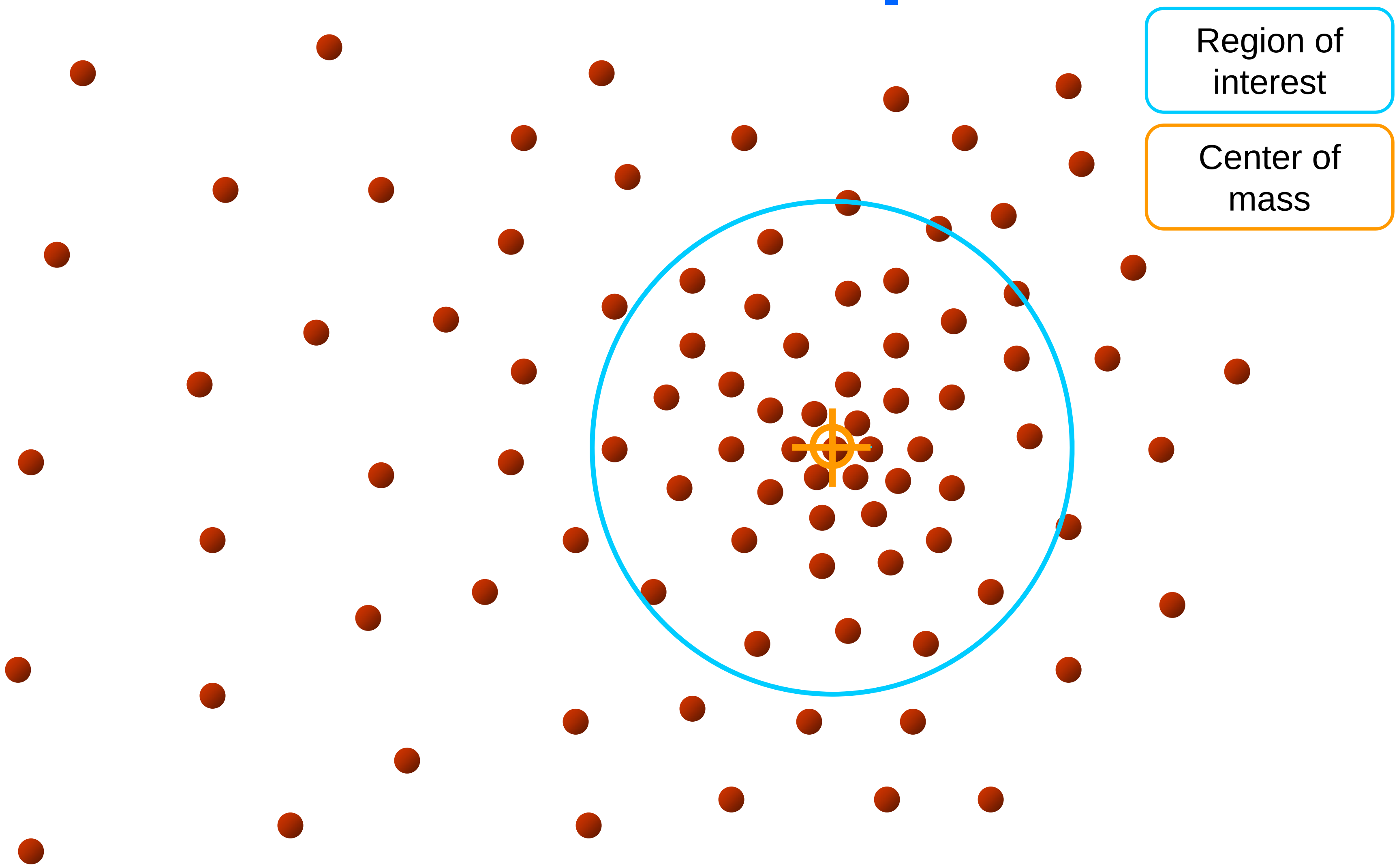
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

Intuitive Description



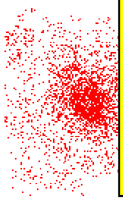
Objective : Find the densest region

Distribution of identical billiard balls

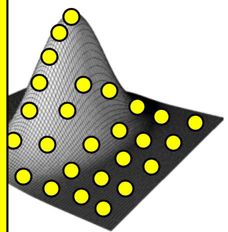
What is Mean Shift ?

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

<u>PDF in feature space</u>
- Color space
- Scale space
- Actually any feature space you can conceive
- …

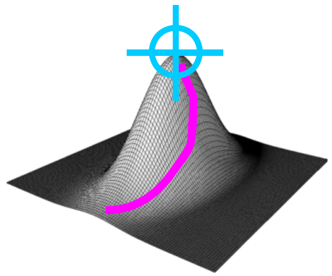


Data



DF Representation

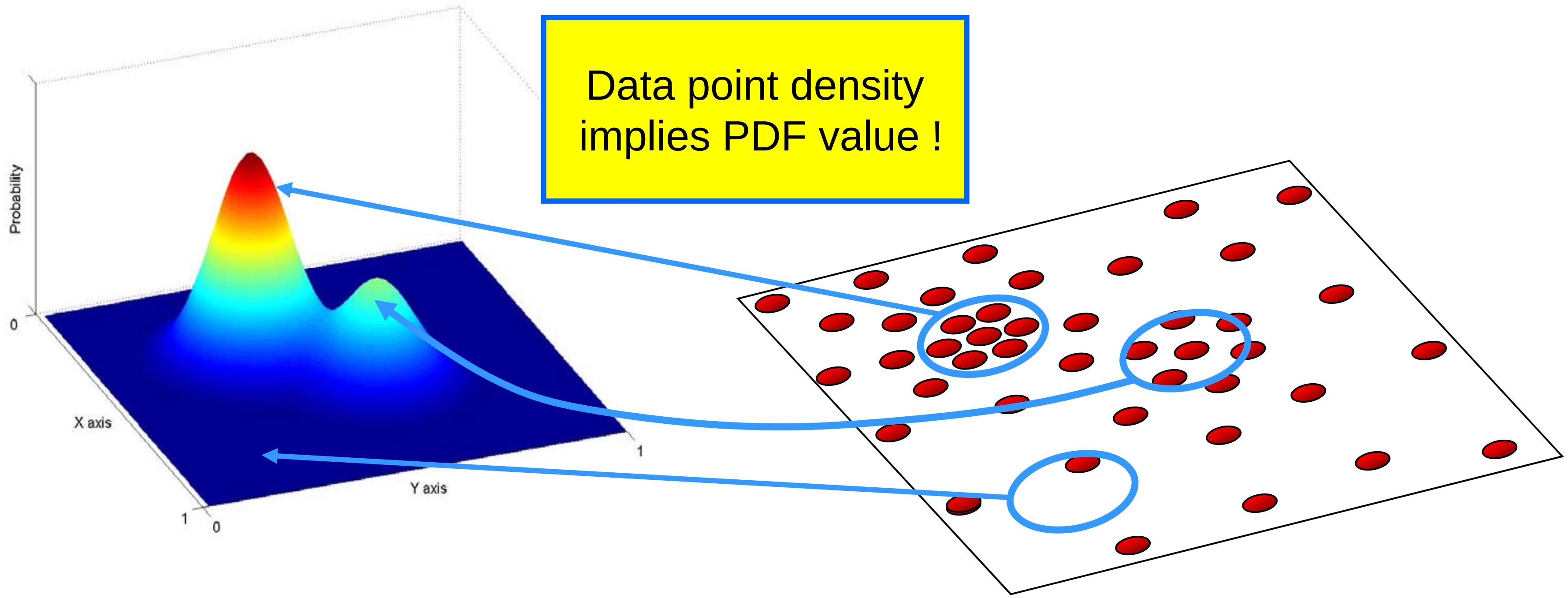
Non-parametric
Density **GRADIENT** Estimation
(Mean Shift)



PDF Analysis

Non-Parametric Density Estimation

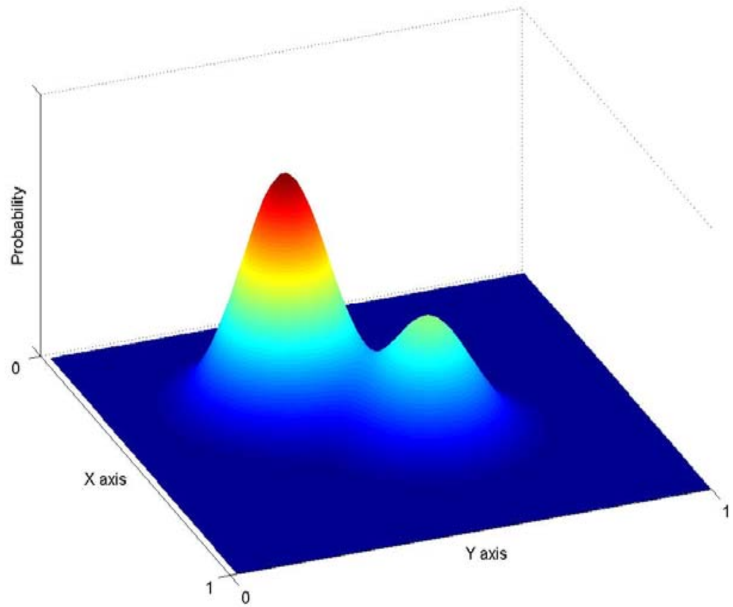
<u>Assumption</u> : The data points are sampled from an underlying PDF



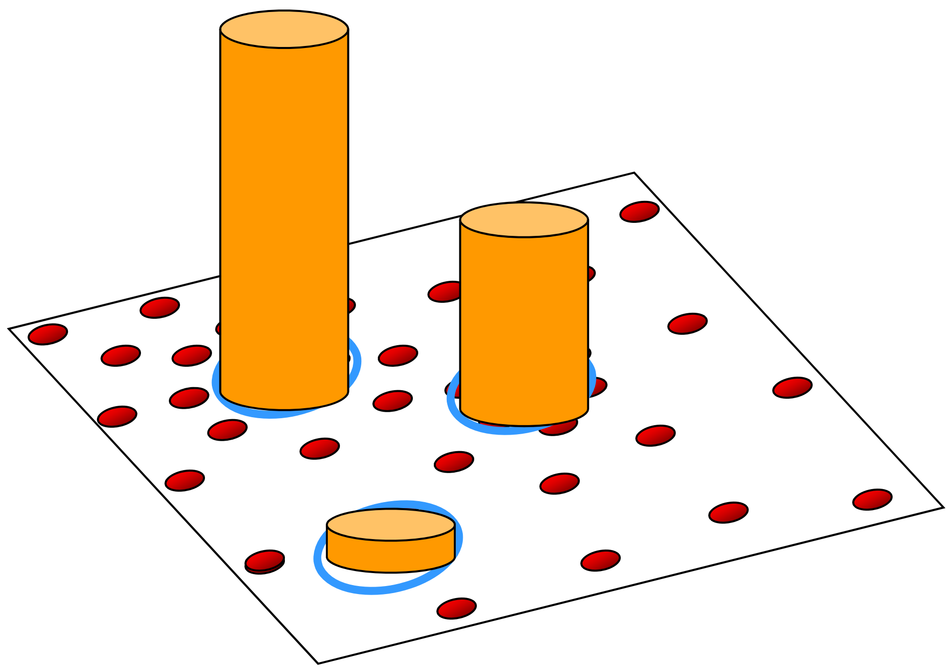
Assumed Underlying PDF

Real Data Samples

Non-Parametric Density Estimation

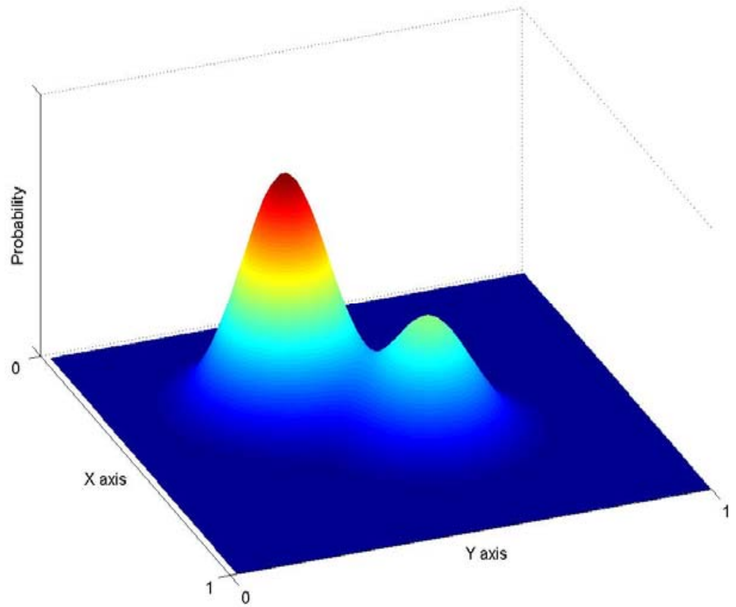


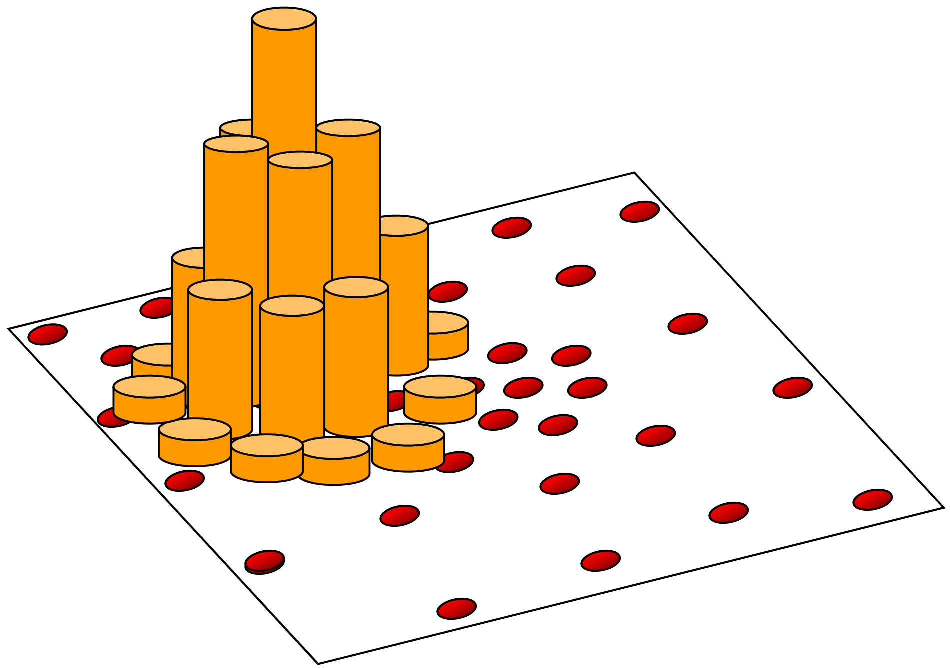
Assumed Underlying PDF



Real Data Samples

Non-Pa?rametric Density Estimation



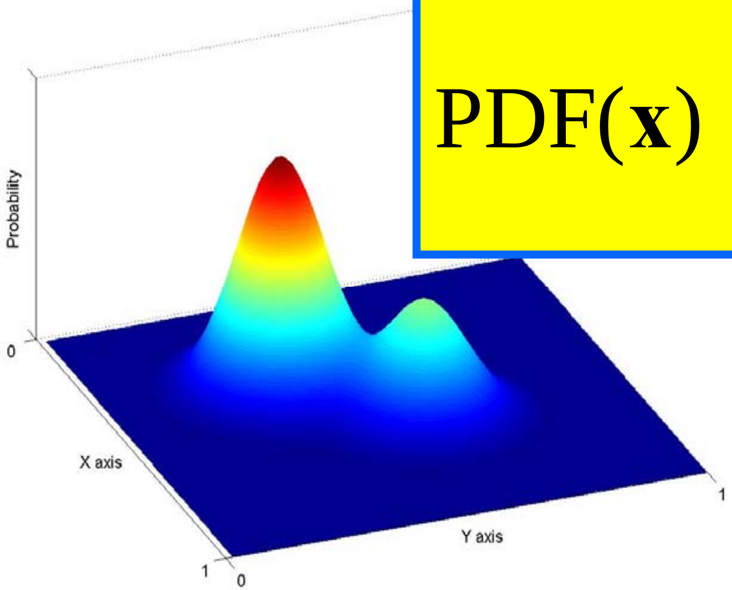


Assumed Underlying PDF **Real Data Samples**

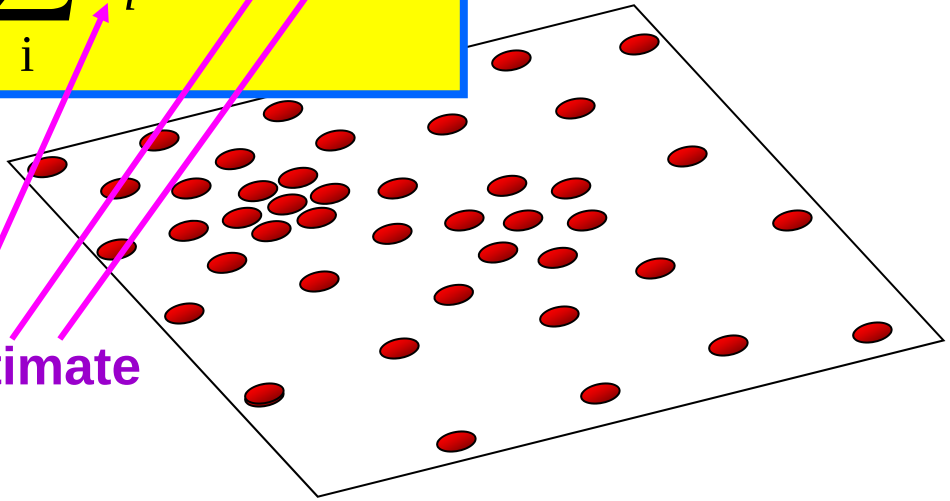
Parametric Density Estimation

<u>Assumption</u> : The data points are sampled from an underlying PDF

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



Estimate



Assumed Underlying PDF

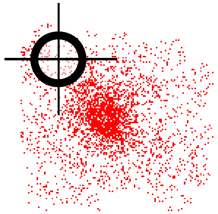
Real Data Samples

Kernel Density Estimation

Parzen Windows - General Framework

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

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



Data

Kernel Properties:

- Normalized

$$\int_{R^d} K(\mathbf{x})d\mathbf{x} = 1$$

- Symmetric

$$\int_{R^d} \mathbf{x}K(\mathbf{x})d\mathbf{x} = 0$$

- Exponential weight decay

$$\lim_{\|\mathbf{x}\|\to\infty} \|\mathbf{x}\|^d K(\mathbf{x}) = 0$$

- ???

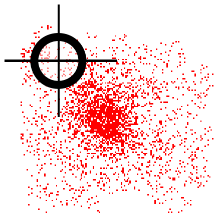
$$\int_{R^d} \mathbf{x}\mathbf{x}^T K(\mathbf{x})d\mathbf{x} = c\mathbf{I}$$

Kernel Density Estimation

Parzen Windows - Function Forms

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

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



Data

In practice one uses the forms:

$$K(\mathbf{x}) = c\prod_{i=1}^{d} k(x_i)$$ or $$K(\mathbf{x}) = ck\left(\|\mathbf{x}\|\right)$$

Same function on each dimension

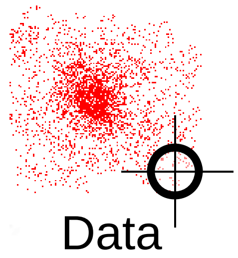
Function of vector length only

Kernel Density Estimation

Various Kernels

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

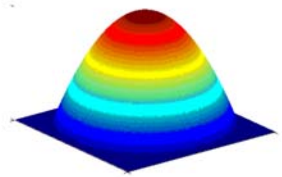
A function of some finite number of data points $x_1 \ldots x_n$



Examples:

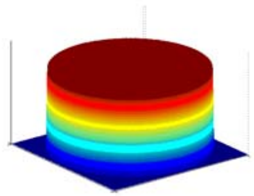
- Epanechnikov Kernel

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



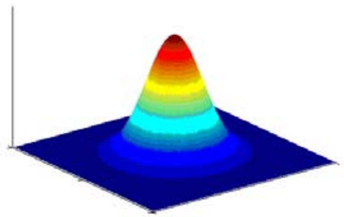
- Uniform Kernel

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



- Normal Kernel

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



Kernel Density $\underbrace{\text{Estimation}}$

Gradient

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

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 :

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

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

Kernel Density Estimation

Computing The Mean Shift

Gradient

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

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

Computing The Mean Shift

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

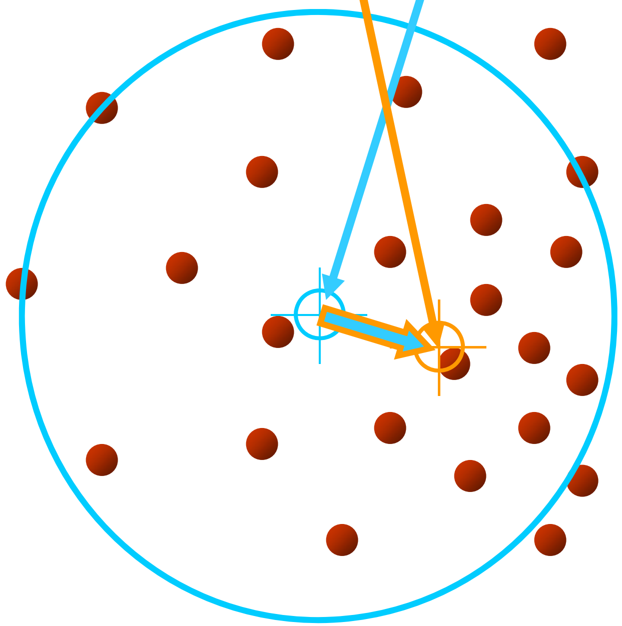
Yet another Kernel
density estimation !

<u>Simple Mean Shift procedure</u>:

- Compute mean shift vector

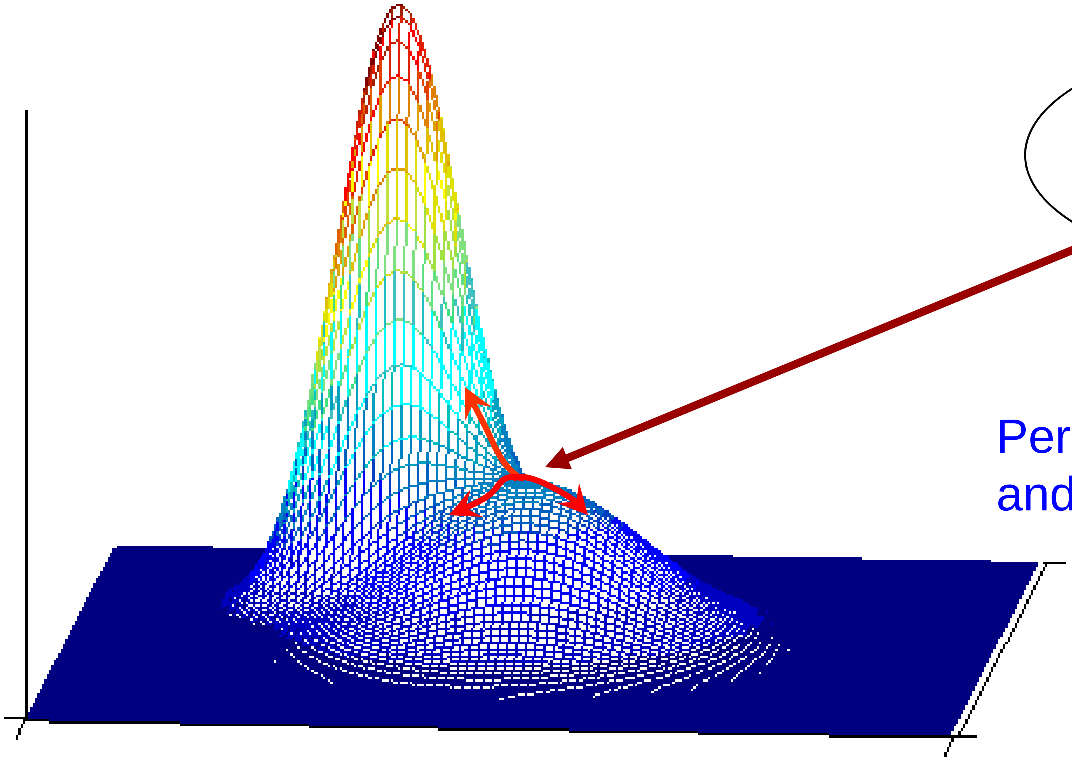
$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}-\mathbf{x}\right]$$

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



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

Mean Shift Mode Detection



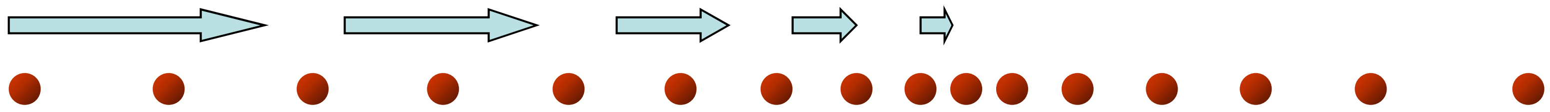
What happens if we reach a saddle point **?**

Perturb the mode position and check if we return back

<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

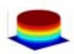


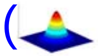
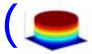
- Automatic convergence speed – the mean shift vector size depends on the gradient itself.

- Near maxima, the steps are small and refined

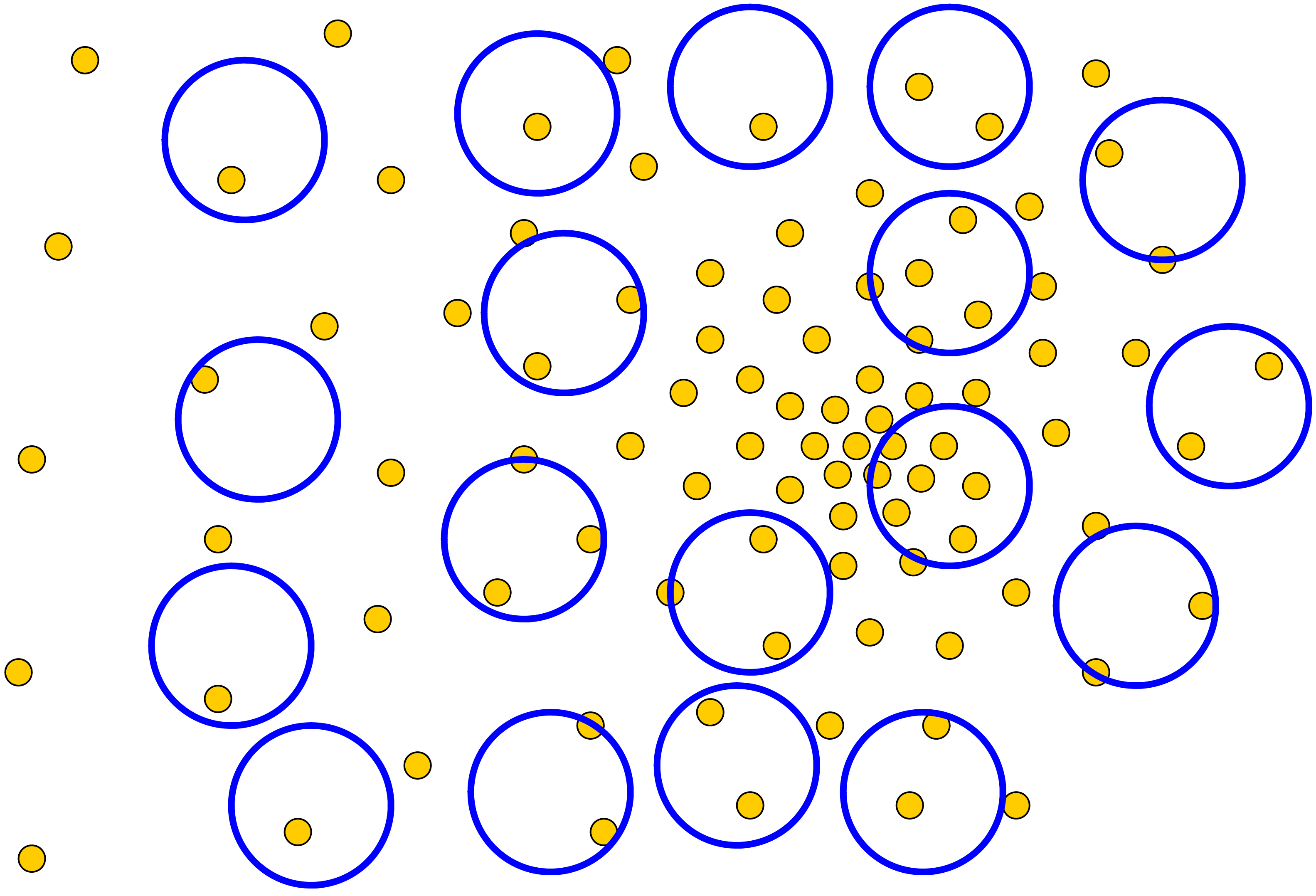
Adaptive Gradient Ascent

- 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 ().

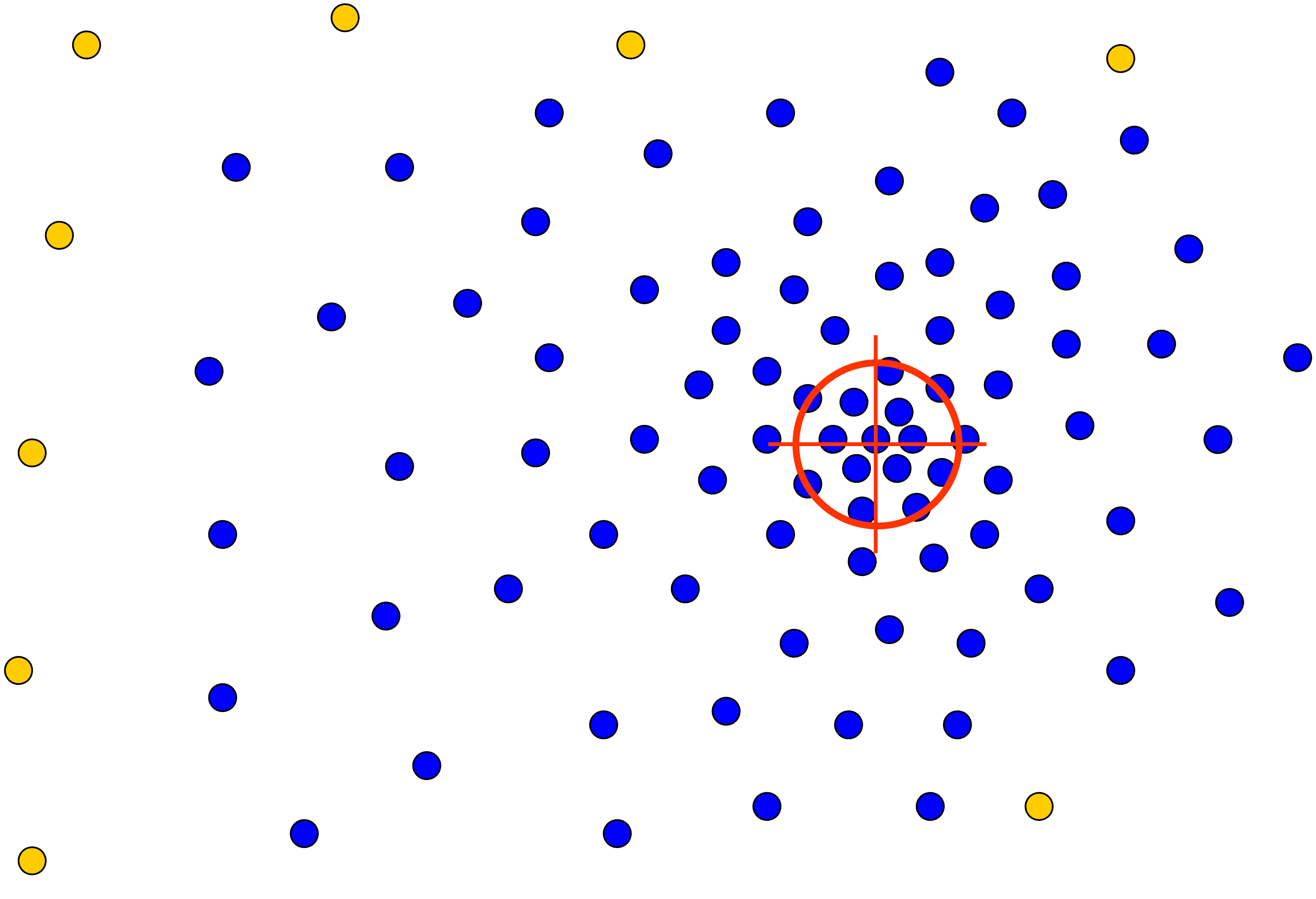
Real Modality Analysis



Tessellate the space
with windows

Run the procedure in parallel

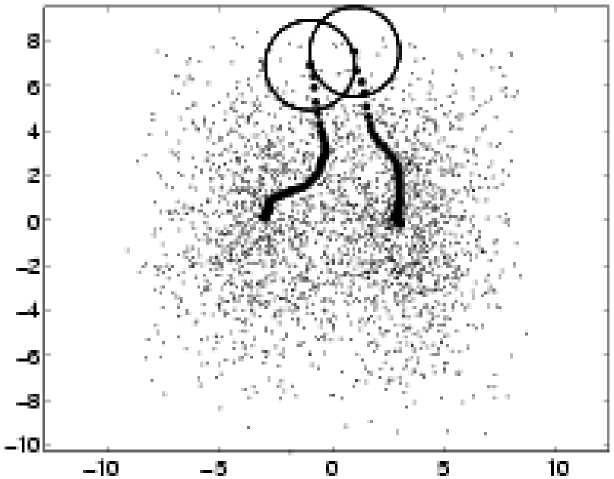
Real Modality Analysis



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

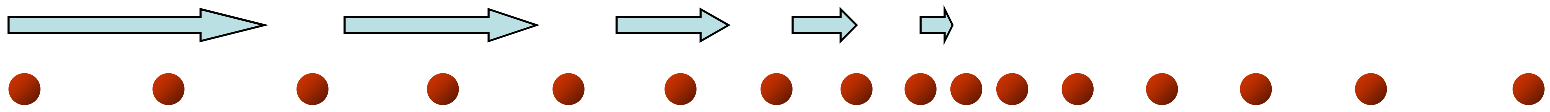
Real Modality Analysis

An example



Window tracks signify the steepest ascent directions

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

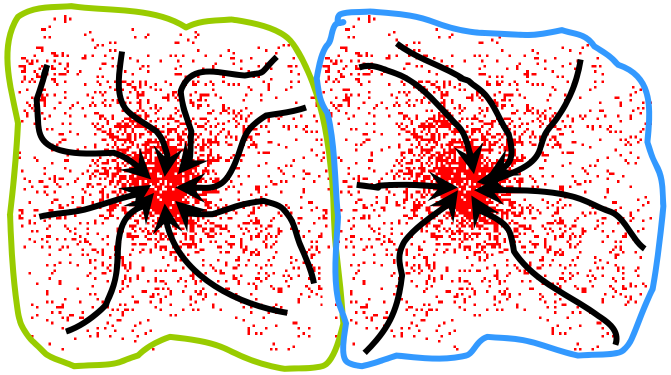
• Inappropriate window size can cause modes to be merged, or generate additional “shallow” modes → Use adaptive window size

Mean Shift Applications

Clustering

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

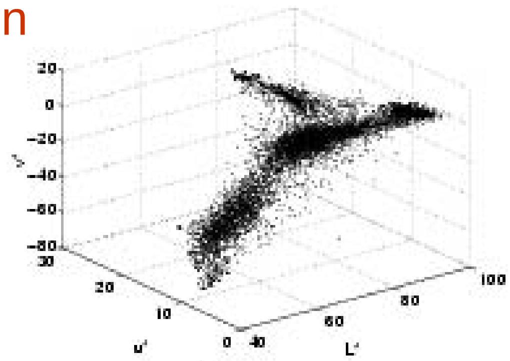
Attraction basin : the region for which all trajectories lead to the same mode



Clustering

Real Example

Feature space:
L*u*v representation



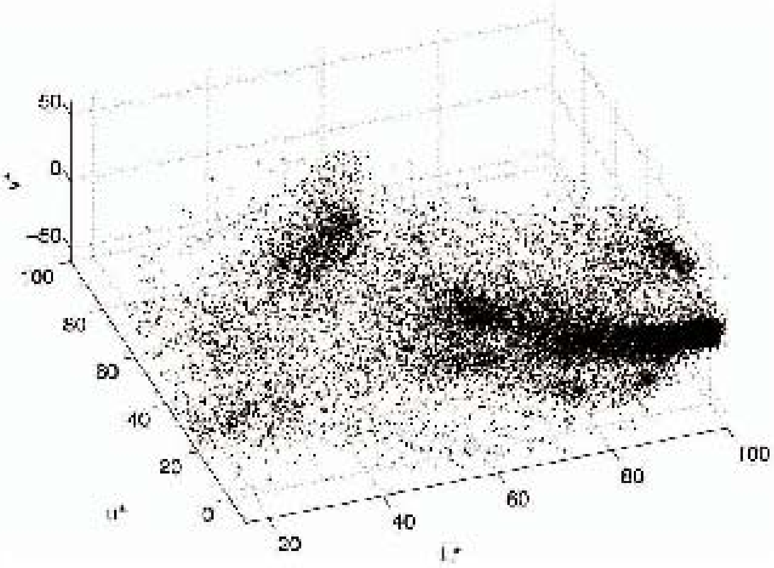
(a)

'nitial window
enters

M

pruning

Clustering
Real Example



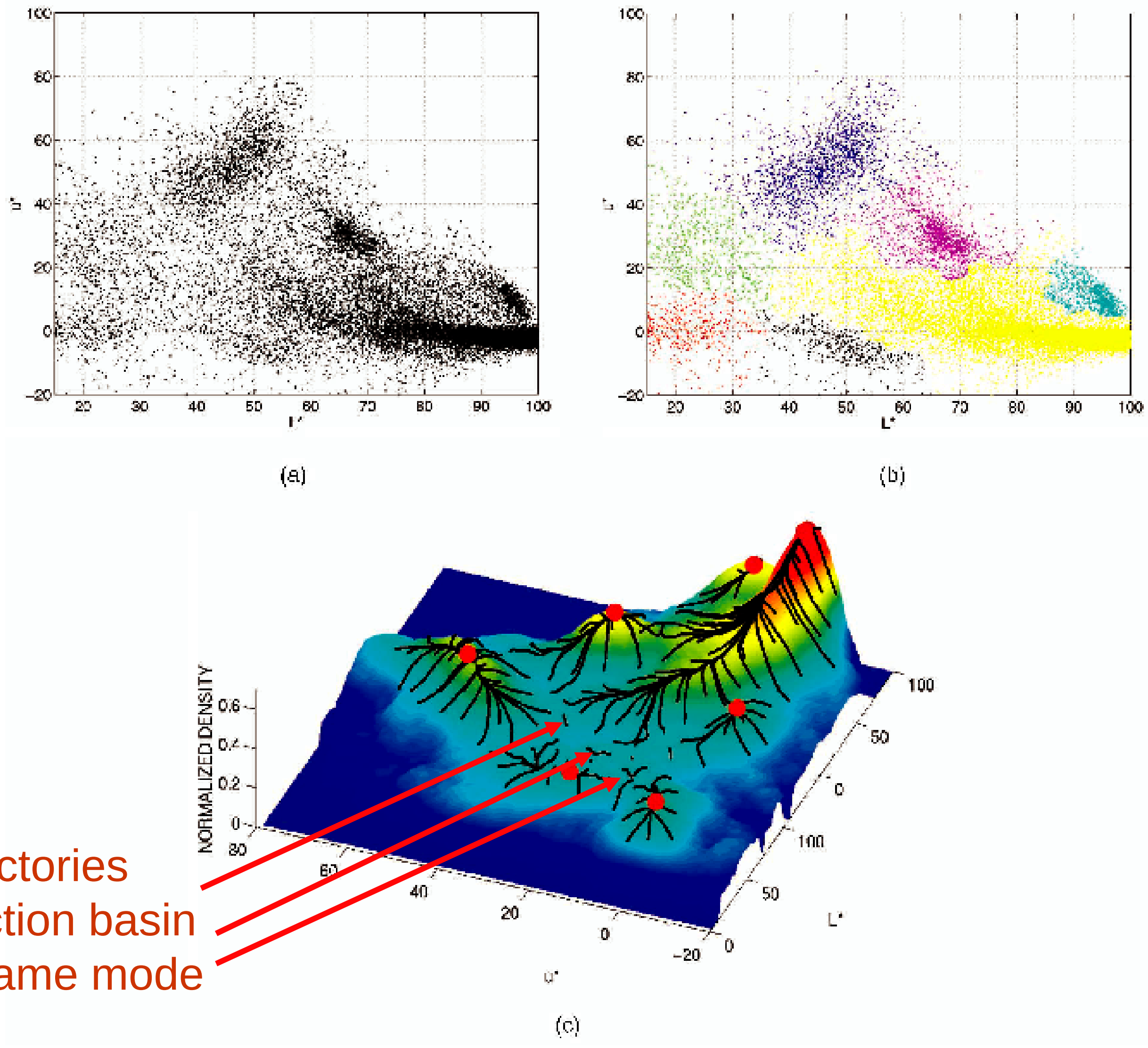
L*u*v space representation

Clustering

Real Example

2D (L*u) space representation

Final clusters



Not all trajectories in the attraction basin reach the same mode

Discontinuity Preserving Smoothing

Feature space : Joint domain = spatial coordinates + color space

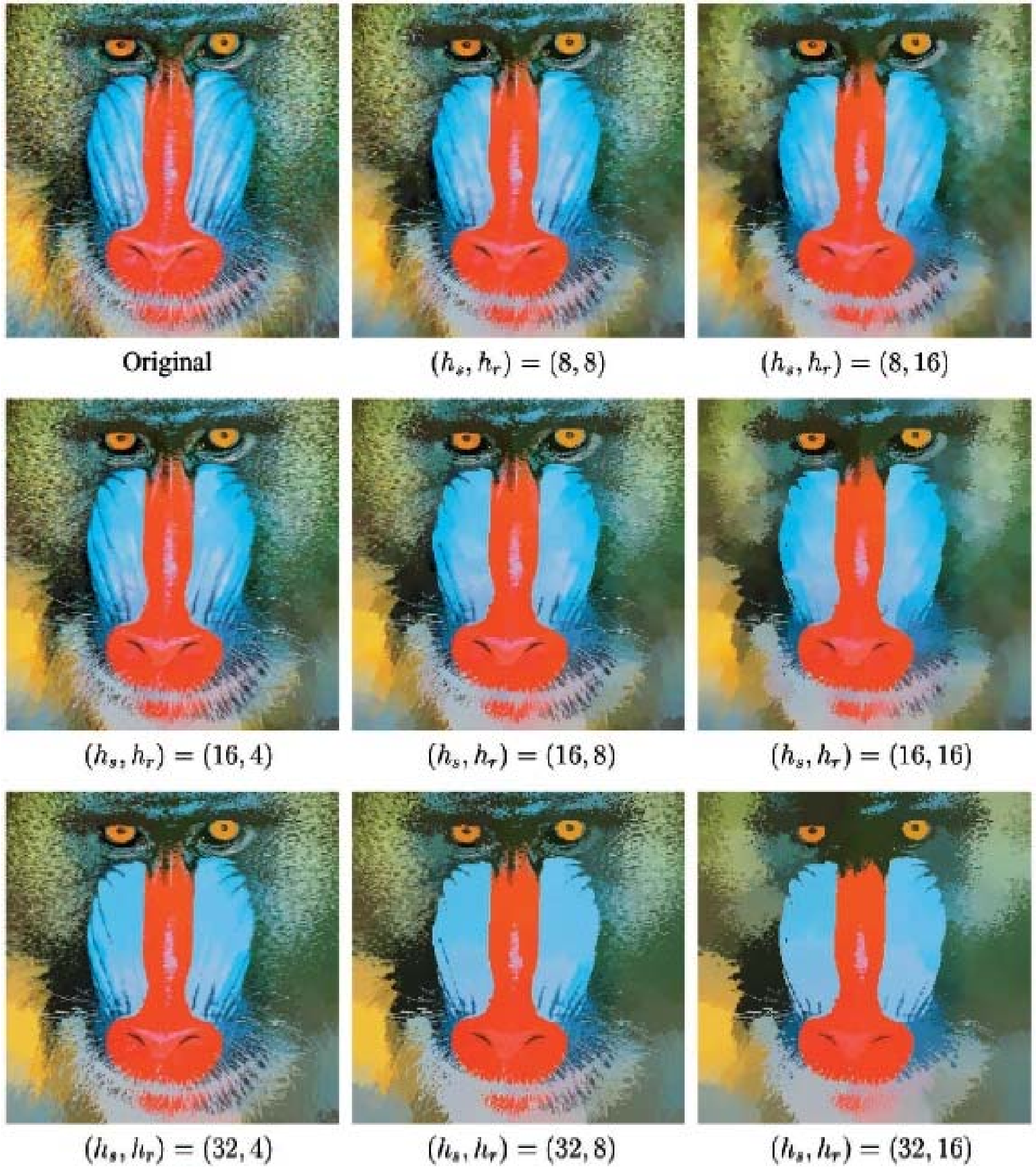
$$K(\mathbf{x}) = C \cdot k_s\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|\right) \cdot k_r\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|\right)$$

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

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

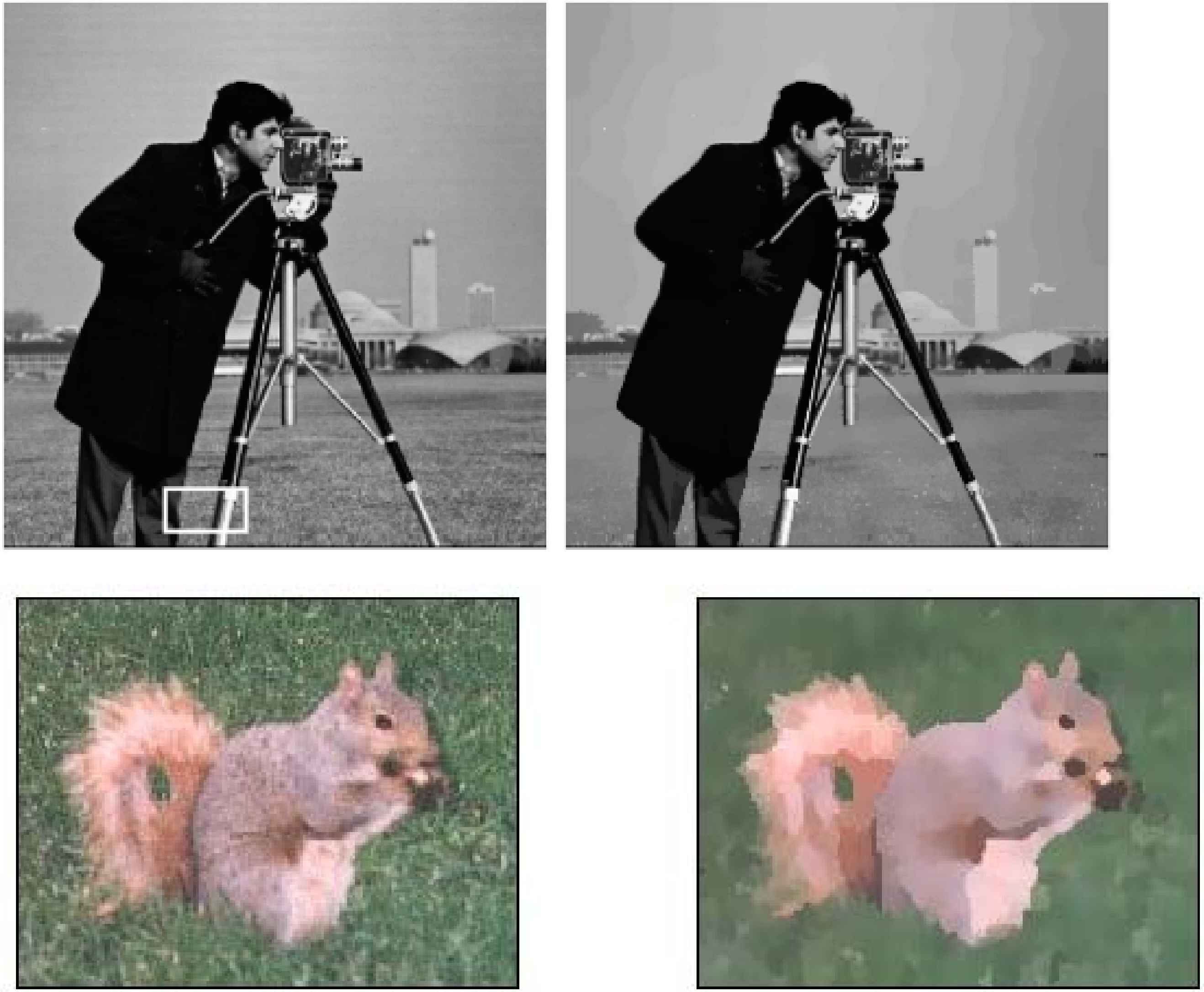
Discontinuity Preserving Smoothing

The effect of window size in spatial and range spaces



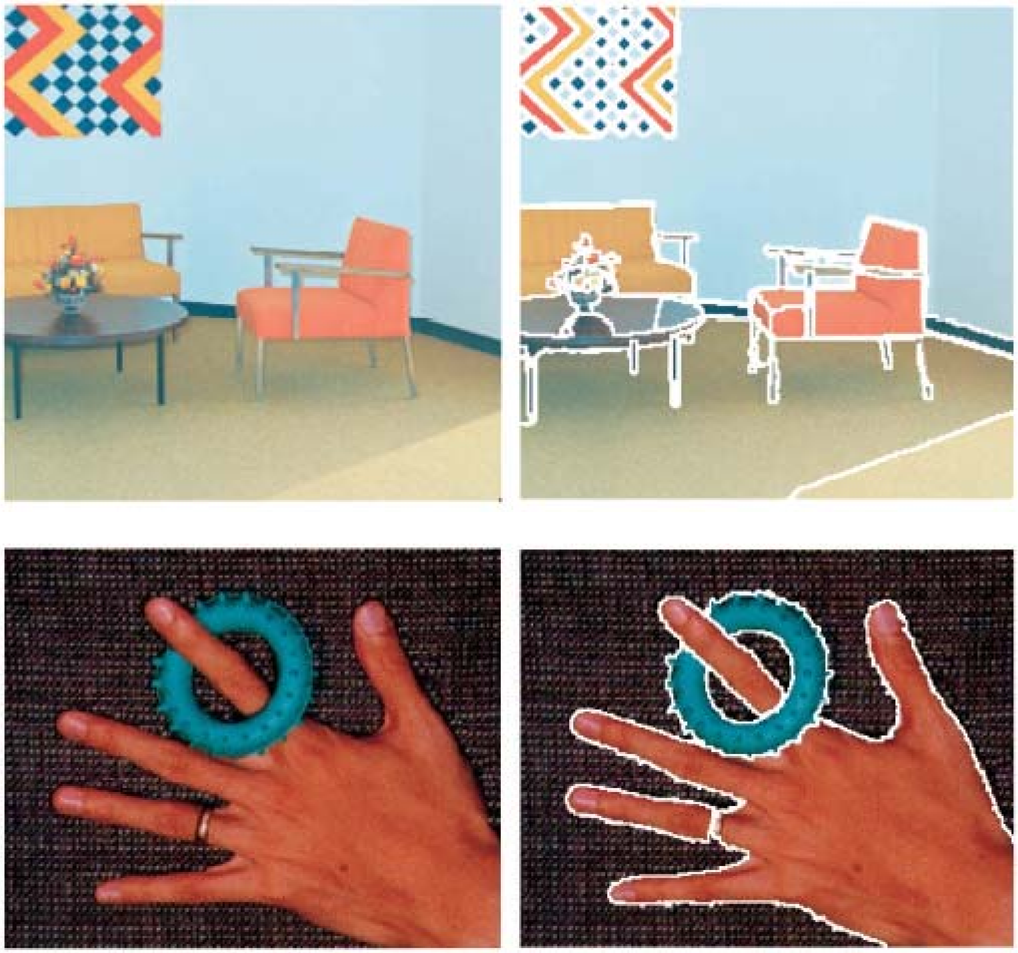
Discontinuity Preserving Smoothing

Example



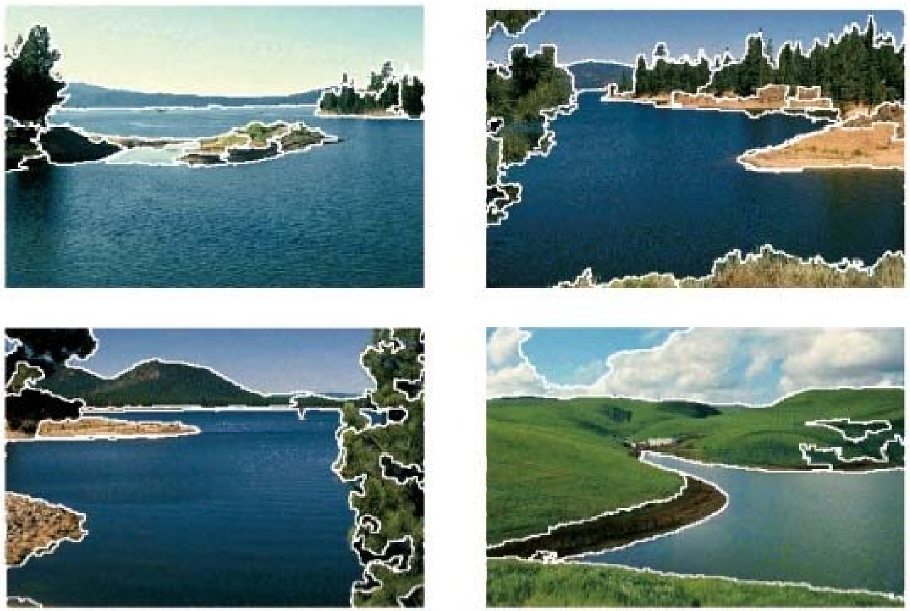
Color Segmentation (1)

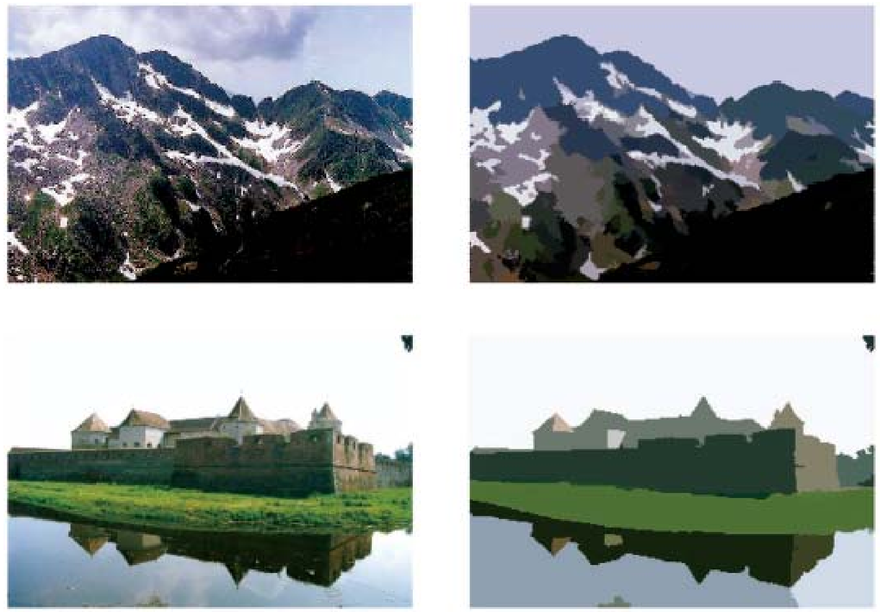
Example



Color Segmentation (2)

Example



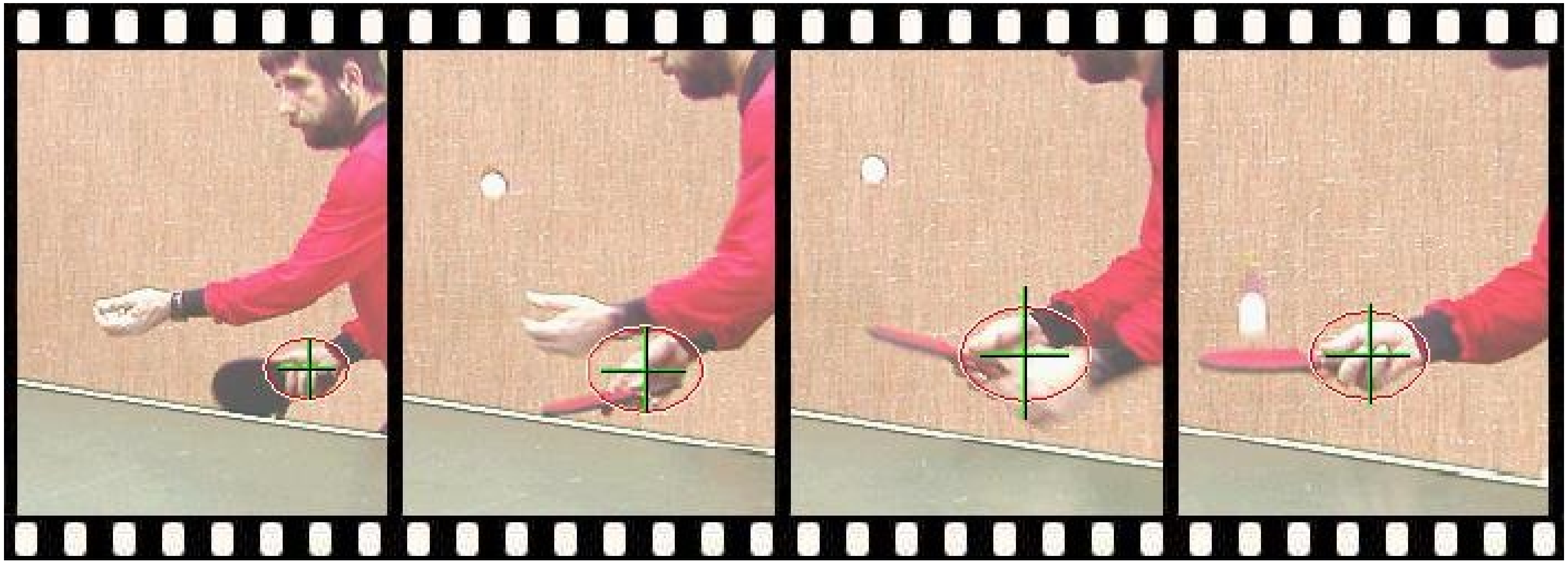


Non-Rigid Object Tracking

Block Matching is not proper to compare the similarity of non-rigid (deformable) objects.

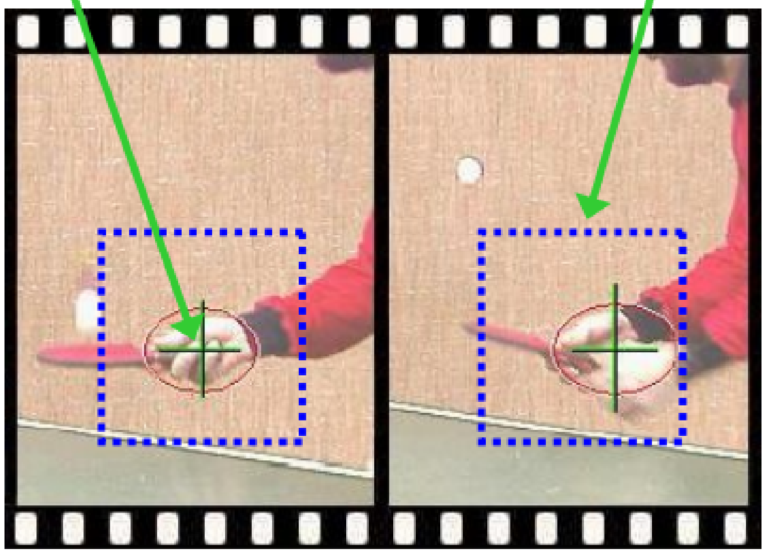
⇒ Histogram matching

⇒ Mean shift processing

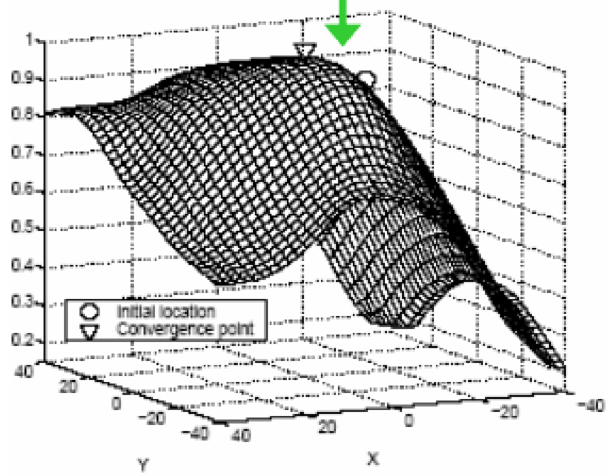


Mean Shift Tracking

Start from the position of the model in the current frame → Search in the model's neighborhood in next frame → Find best candidate by maximizing a similarity func.



$\vec{q}$ $\quad$ $\vec{p}(y)$

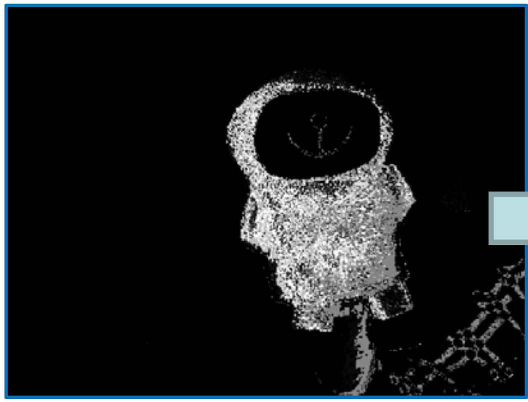


$f\left[\vec{p}(y), \vec{q}\right]$

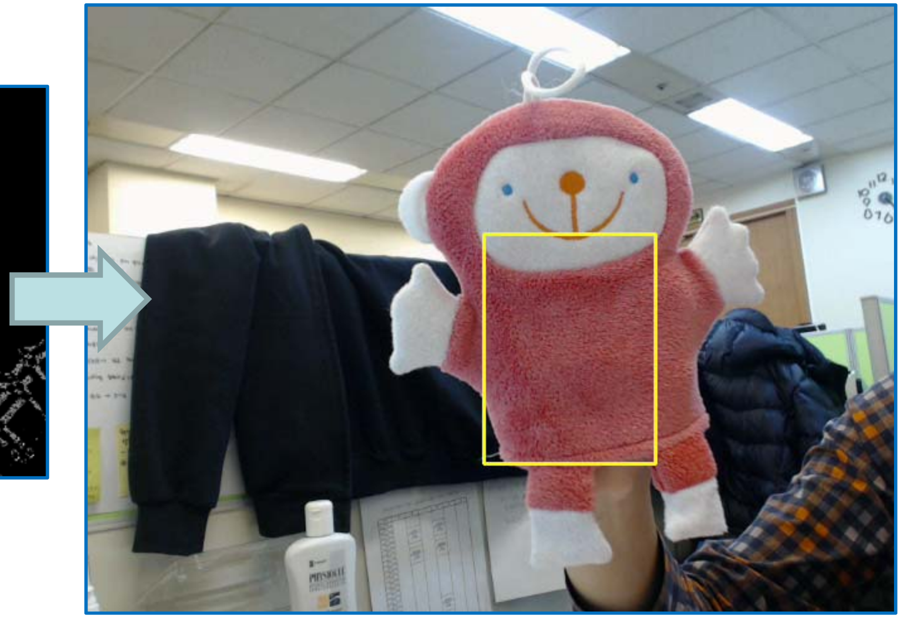
Back Projection

- Assign probability to each pixel using the object histogram
- Mean shift process for the pixels coordinates with the probability weights.

color histogram



back projection



Mean shift process

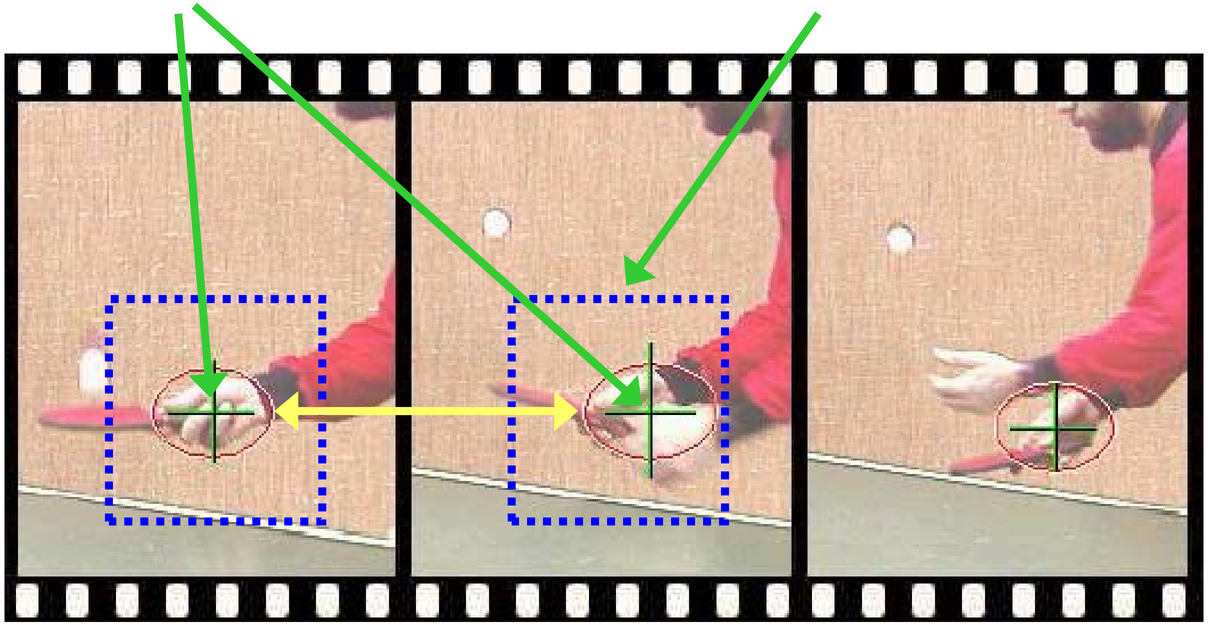
Mean-Shift Object Tracking

General Framework: Target Localization

Start from the position of the model in the current frame

Search in the model's neighborhood in next frame

Find best candidate by maximizing a similarity func.



Model

Candidate

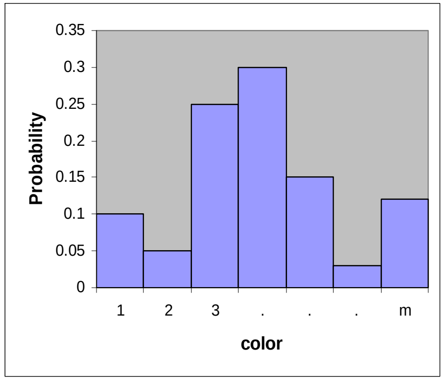
… Current frame …

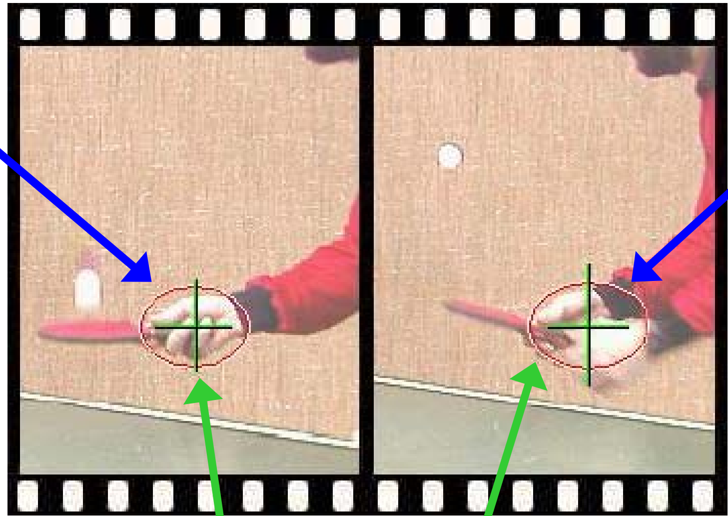
Repeat the same process in the next pair of frames

Mean-Shift Object Tracking

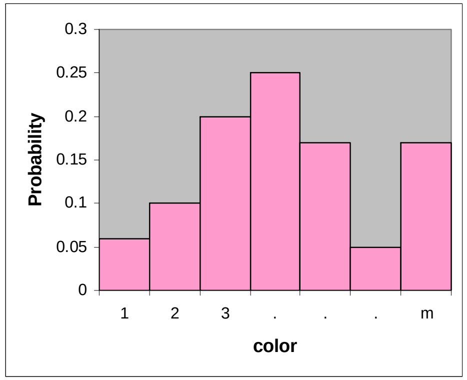
PDF Representation

Target Model
(centered at 0)





Target Candidate
(centered at y)



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

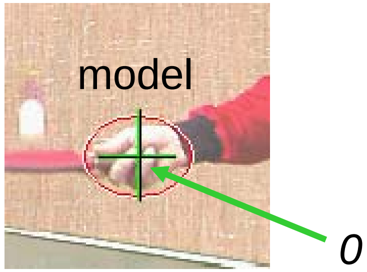
$$\vec{p}(y) = \{p_u(y)\}_{u=1..m} \qquad \sum_{u=1}^{m} p_u = 1$$

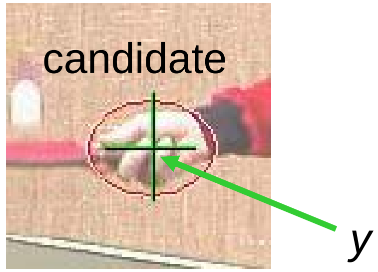
Similarity Function: $f(y) = f\left[\vec{q}, \vec{p}(y)\right]$

Mean-Shift Object Tracking

Finding the PDF of the target model

$\{x_i\}_{i=1..n}$ Target pixel locations





$k(x)$ A differentiable, isotropic, convex, monotonically decreasing kernel

- Peripheral pixels are affected by occlusion and background interference

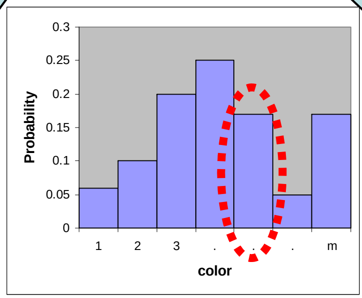
$b(x)$ The color bin index (1..*m*) of pixel *x*

Probability of feature u in model

$$q_u = C \sum_{b(x_i)=u} \underbrace{k\left(\left\|x_i\right\|^2\right)}$$

Normalization factor

Pixel weight

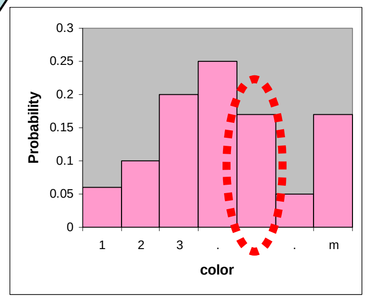


Probability of feature u in candidate

$$p_u(y) = C_h \sum_{b(x_i)=u} \underbrace{k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}$$

Normalization factor

Pixel weight



Mean-Shift Object Tracking

Similarity Function

Target model: $\vec{q} = (q_1, \ldots, q_m)$

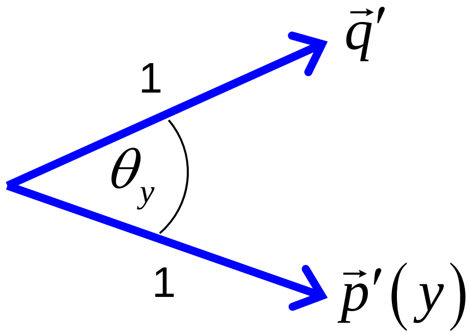
Target candidate: $\vec{p}(y) = (p_1(y), \ldots, p_m(y))$

Similarity function: $f(y) = f\left[\vec{p}(y), \vec{q}\right] = ?$

The Bhattacharyya Coefficient

$$\vec{q}' = \left(\sqrt{q_1}, \ldots, \sqrt{q_m}\right)$$

$$\vec{p}'(y) = \left(\sqrt{p_1(y)}, \ldots, \sqrt{p_m(y)}\right)$$



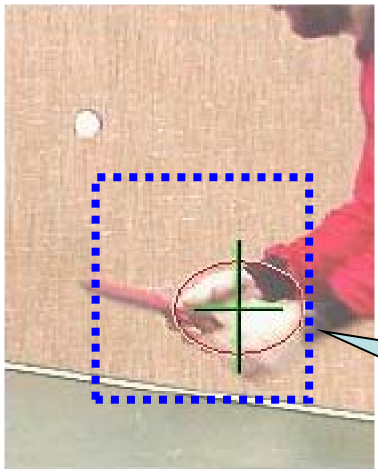
$$f(y) = \cos\theta_y = \frac{p'(y)^T q'}{\|p'(y)\| \cdot \|q'\|} = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}$$

Mean-Shift Object Tracking

Maximizing the Similarity Function

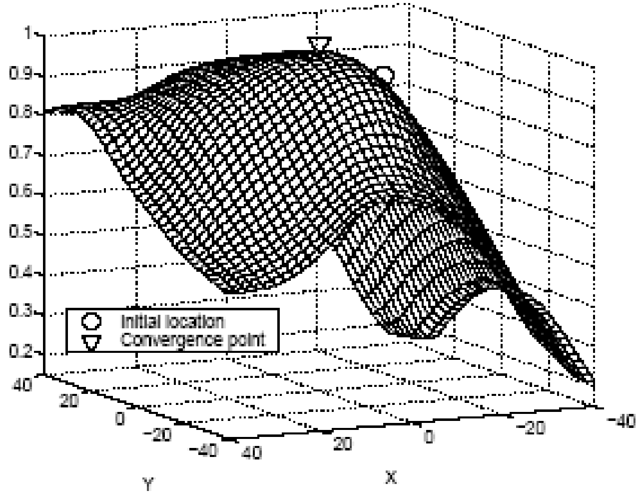
The mode of $\frac{C_h}{2}\sum_{i=1}^{n} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)$ = sought maximum

Important Assumption:



The target representation provides sufficient discrimination

One mode in the searched neighborhood



$f\left[\vec{p}(y),\vec{q}\right]$

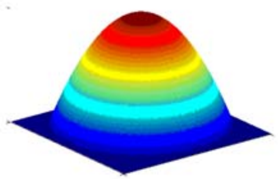
Mean-Shift Object Tracking

Choosing the Kernel

A special class of radially symmetric kernels:

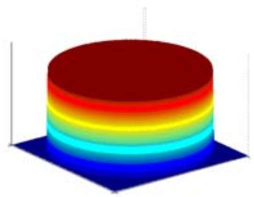
$$K(x) = ck\left(\|x\|^2\right)$$

Epanechnikov kernel:



$$k(x) = \begin{cases} 1 - x & \text{if } \|x\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Uniform kernel:



$$g(x) = -k(x) = \begin{cases} 1 & \text{if } \|x\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Extended Mean-Shift:

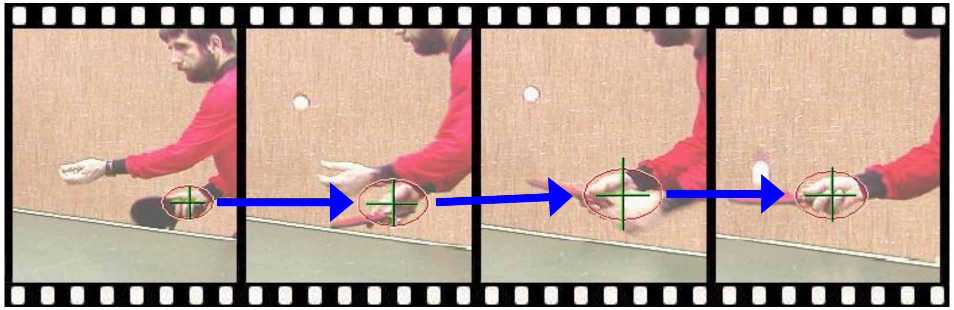
$$y_1 = \frac{\sum_{i=1}^{n} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} \longrightarrow y_1 = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}$$

Mean-Shift Object Tracking

Adaptive Scale

Problem:

The scale of the target changes in time → The scale (h) of the kernel must be adapted



Solution:

Run localization 3 times with different h → Choose h that achieves maximum similarity

Tracking Result

- PTZ control