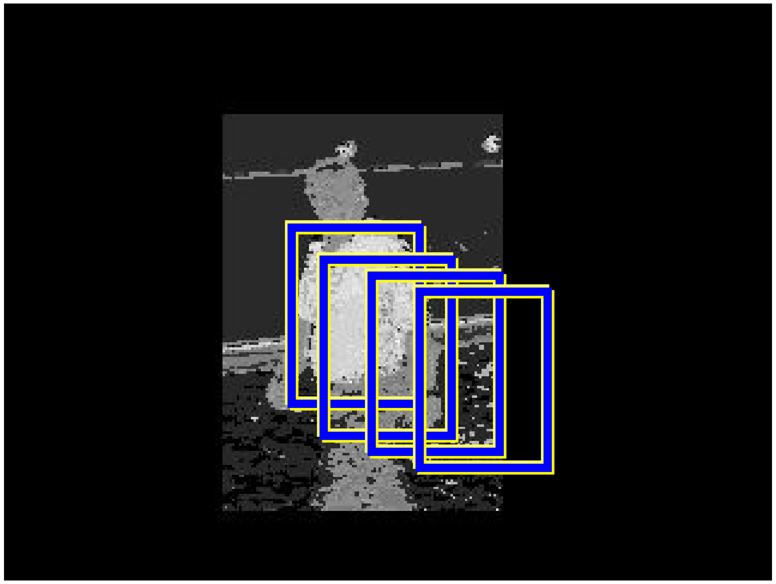
Robert Collins
CSE486, Penn State

Lecture 29:

Video Tracking: Mean-Shift



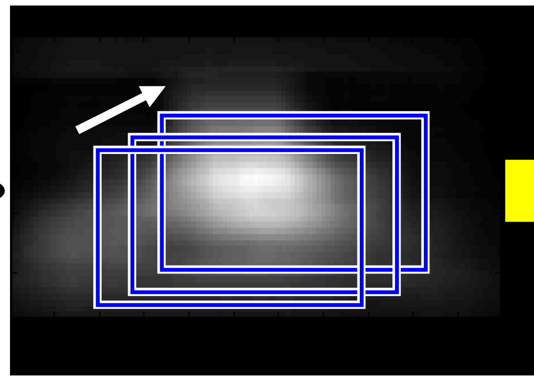
Appearance-Based Tracking

current frame +
previous location

appearance model
(e.g. image template, or

color; intensity; edge histograms)

likelihood over
object location



Mode-Seeking
(e.g. mean-shift; Lucas-Kanade;
particle filtering)

current location

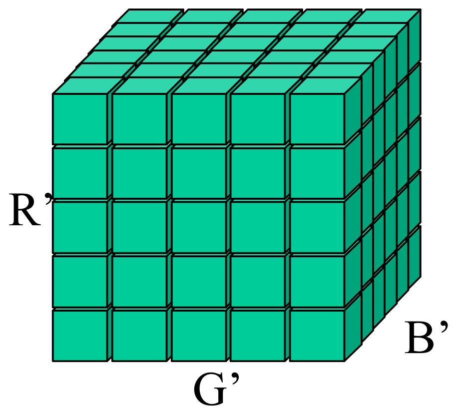
Histogram Appearance Models

- Motivation – to track non-rigid objects, (like a walking person), it is hard to specify an explicit 2D parametric motion model.

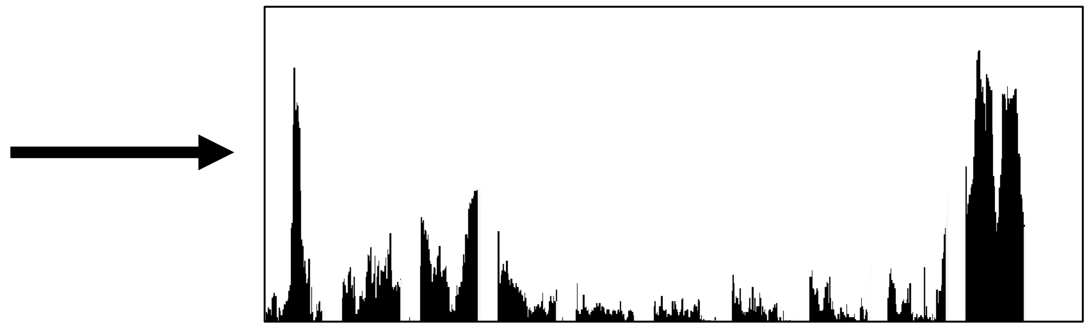
- Appearances of non-rigid objects can sometimes be modeled with color distributions

Robert Collins
CSE486, Penn State

Appearance via Color Histograms



discretize



Color distribution (1D histogram normalized to have unit weight)

R' = R << (8 - nbits)
G' = G << (8 - nbits)
B' = B << (8-nbits)

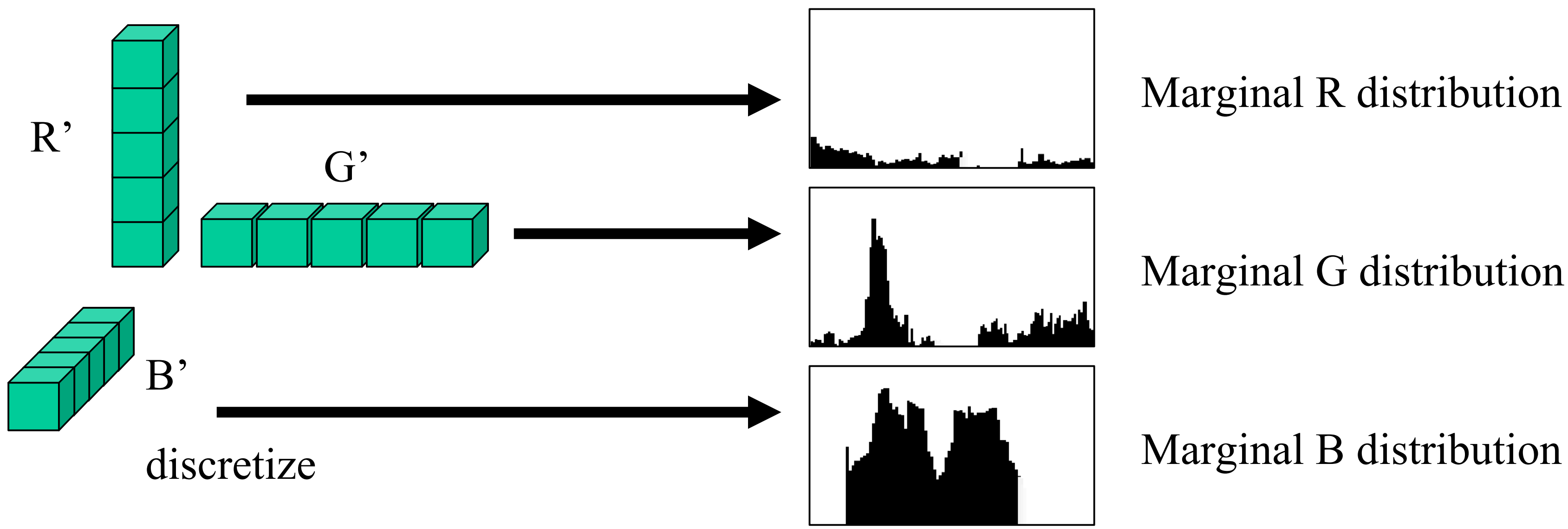
Total histogram size is (2^(8-nbits))^3

example, 4-bit encoding of R,G and B channels yields a histogram of size 16*16*16 = 4096.

Robert Collins
CSE486, Penn State

Smaller Color Histograms

Histogram information can be much much smaller if we are willing to accept a loss in color resolvability.



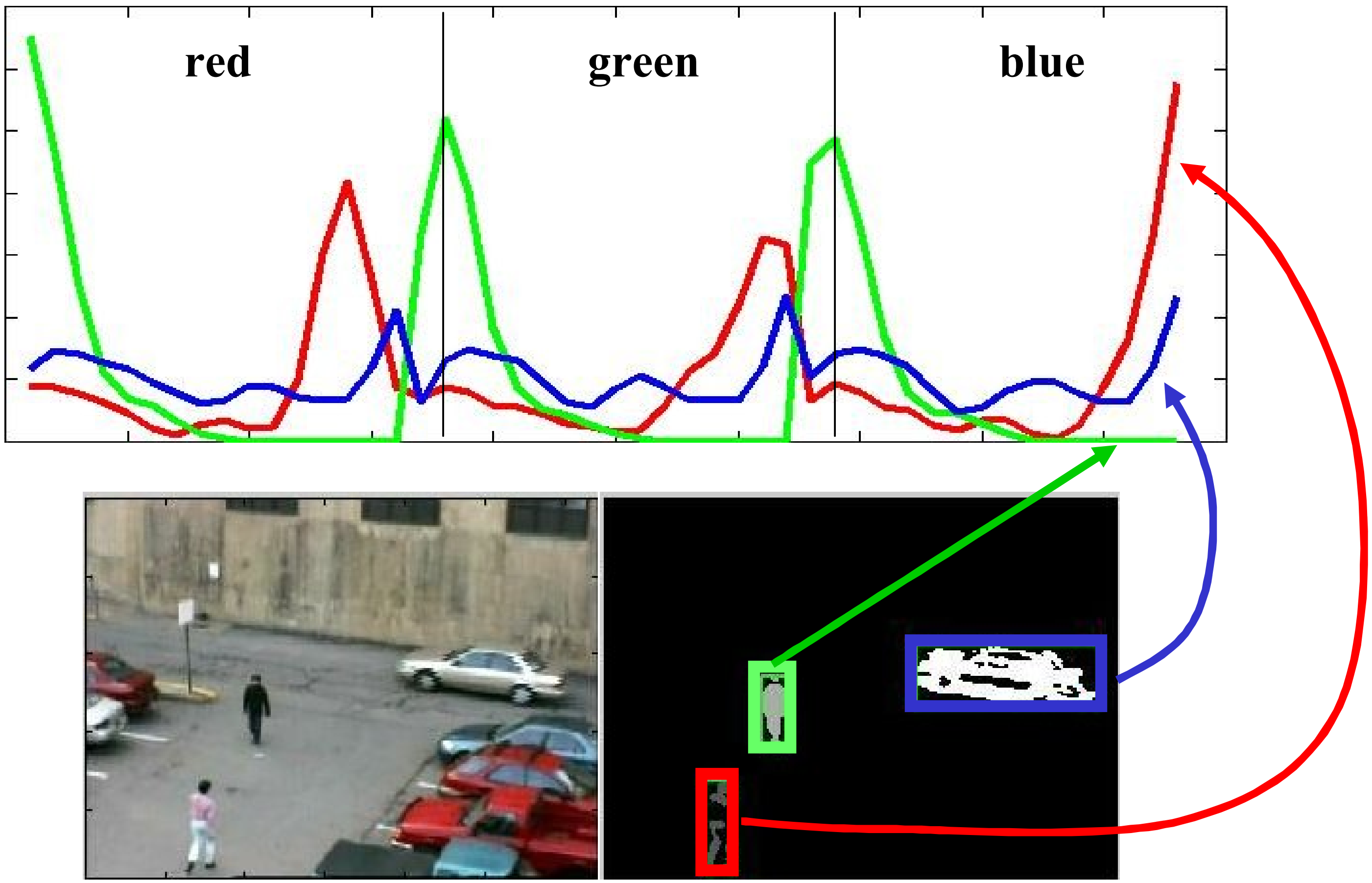
R’ = R << (8 - nbits)
G’ = G << (8 - nbits)
B’ = B << (8-nbits)

Total histogram size is 3*(2^(8-nbits))

example, 4-bit encoding of R,G and B channels yields a histogram of size 3*16 = 48.

Robert Collins
CSE486, Penn State

Color Histogram Example

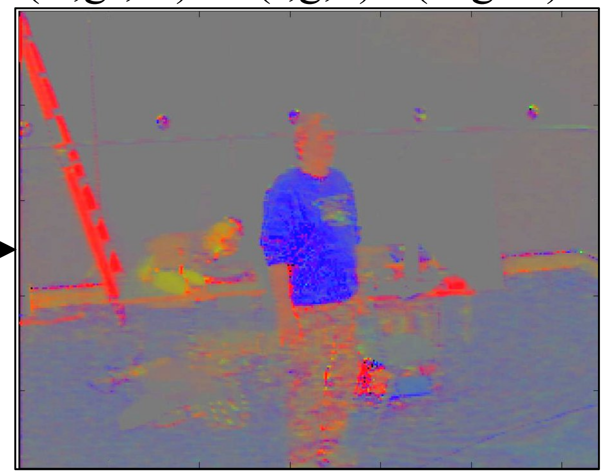


Robert Collins
CSE486, Penn State

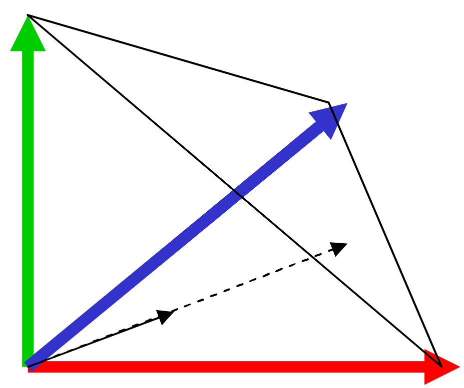
Normalized Color

(r,g,b)

(r',g',b') = (r,g,b) / (r+g+b)



Normalized color divides out pixel luminance (brightness), leaving behind only chromaticity (color) information. The result is less sensitive to variations due to illumination/shading.



Mean-Shift

The mean-shift algorithm is an efficient approach to tracking objects whose appearance is defined by color.

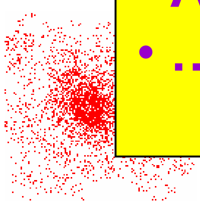
(not limited to only color, however. Could also use edge orientations, texture, motion)

What is Mean Shift ?

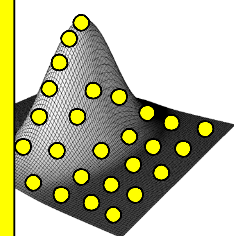
A tool for:
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in $\mathbf{R}^N$

PDF in feature space

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

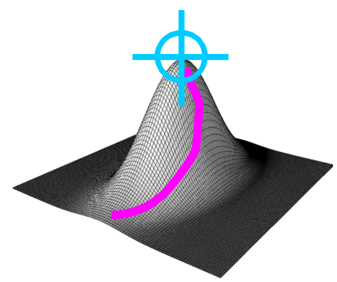


Data



DF Representation

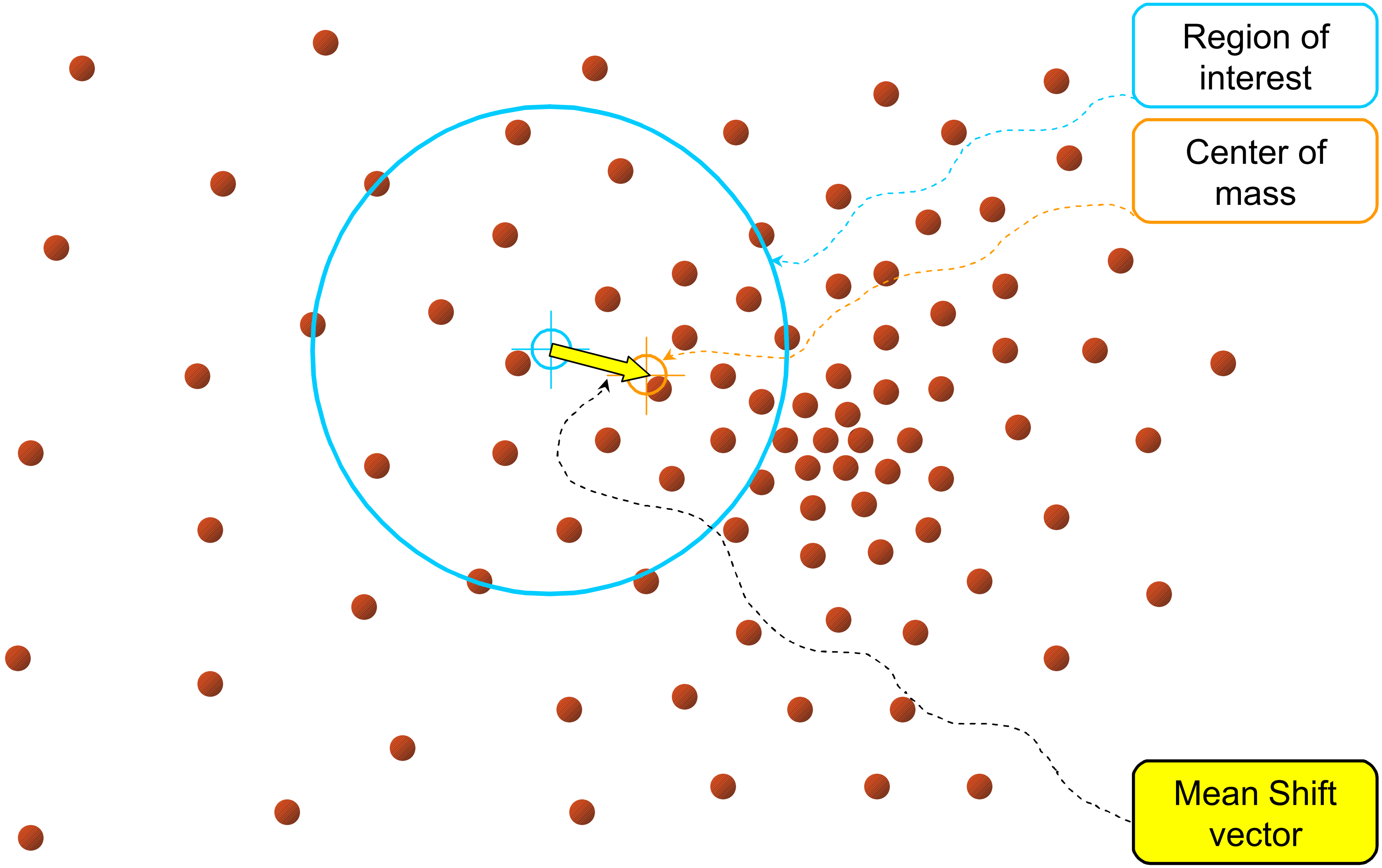
Non-parametric
Density **GRADIENT** Estimation
(Mean Shift)



PDF Analysis

Robert Collins
CSE486, Penn State

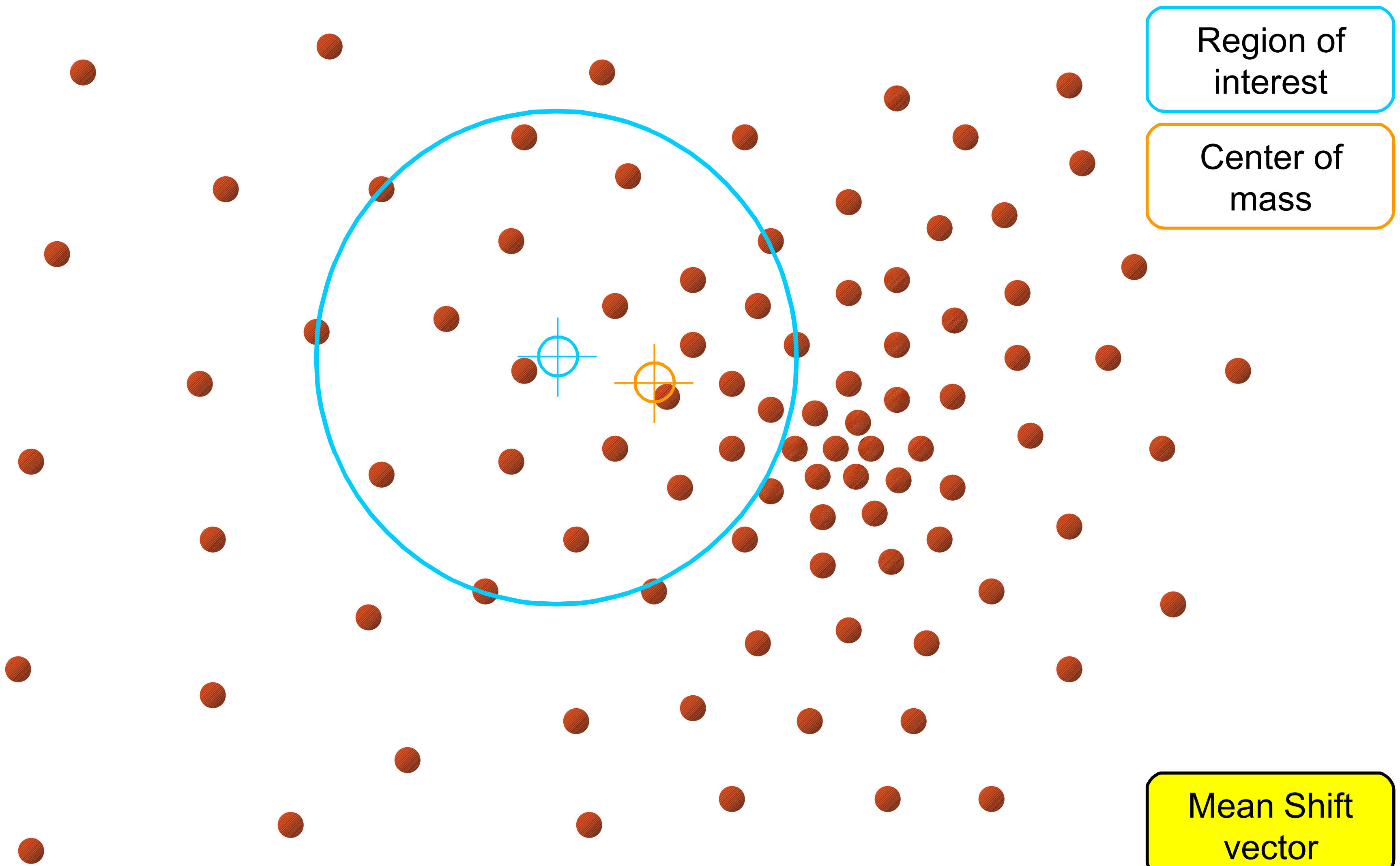
Intuitive Description



Objective : Find the densest region

Ukrainitz&Sarel, Weizmann

Intuitive Description

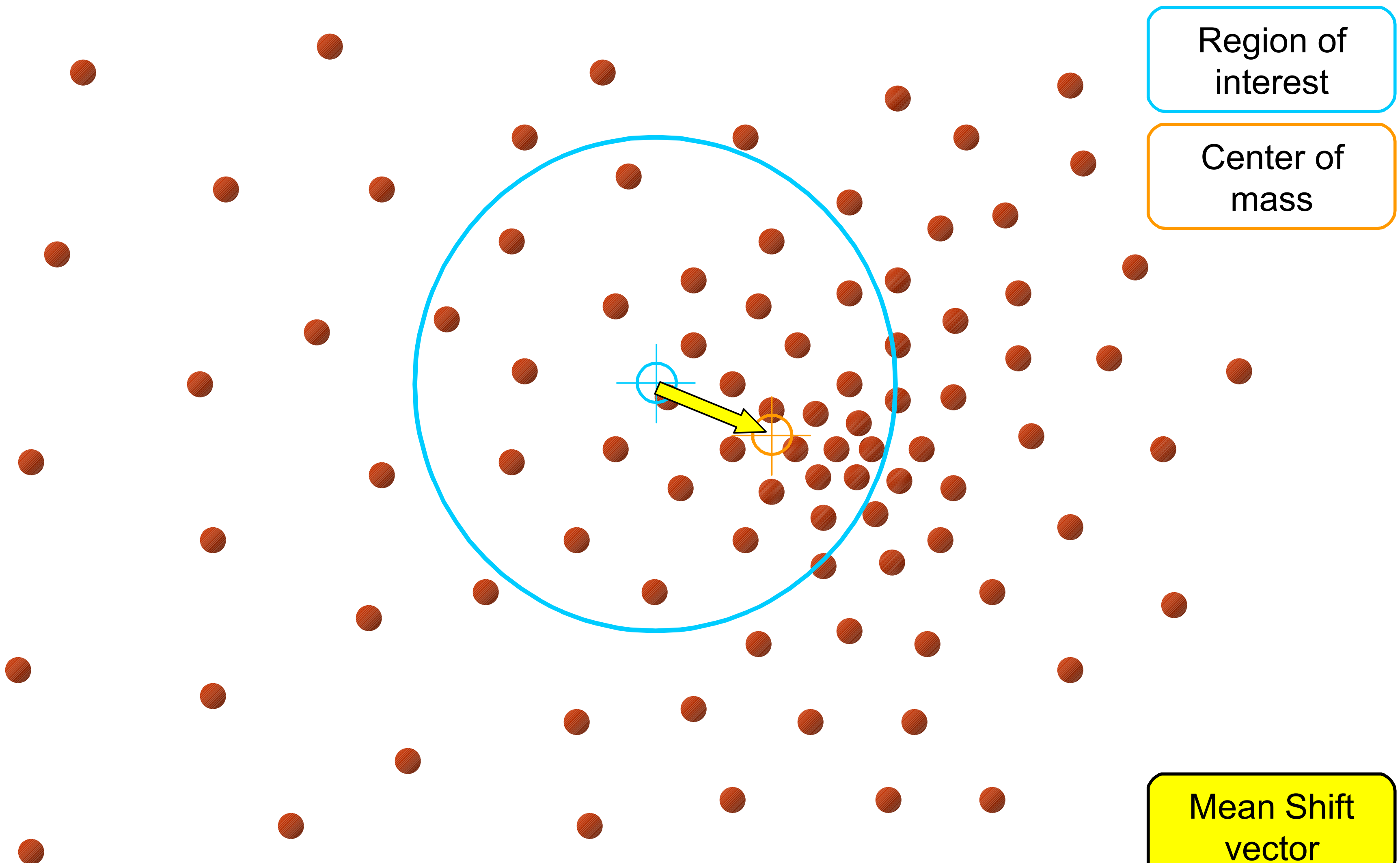


Objective : Find the densest region

Ukrainitz&Sarel, Weizmann

Robert Collins
CSE486, Penn State

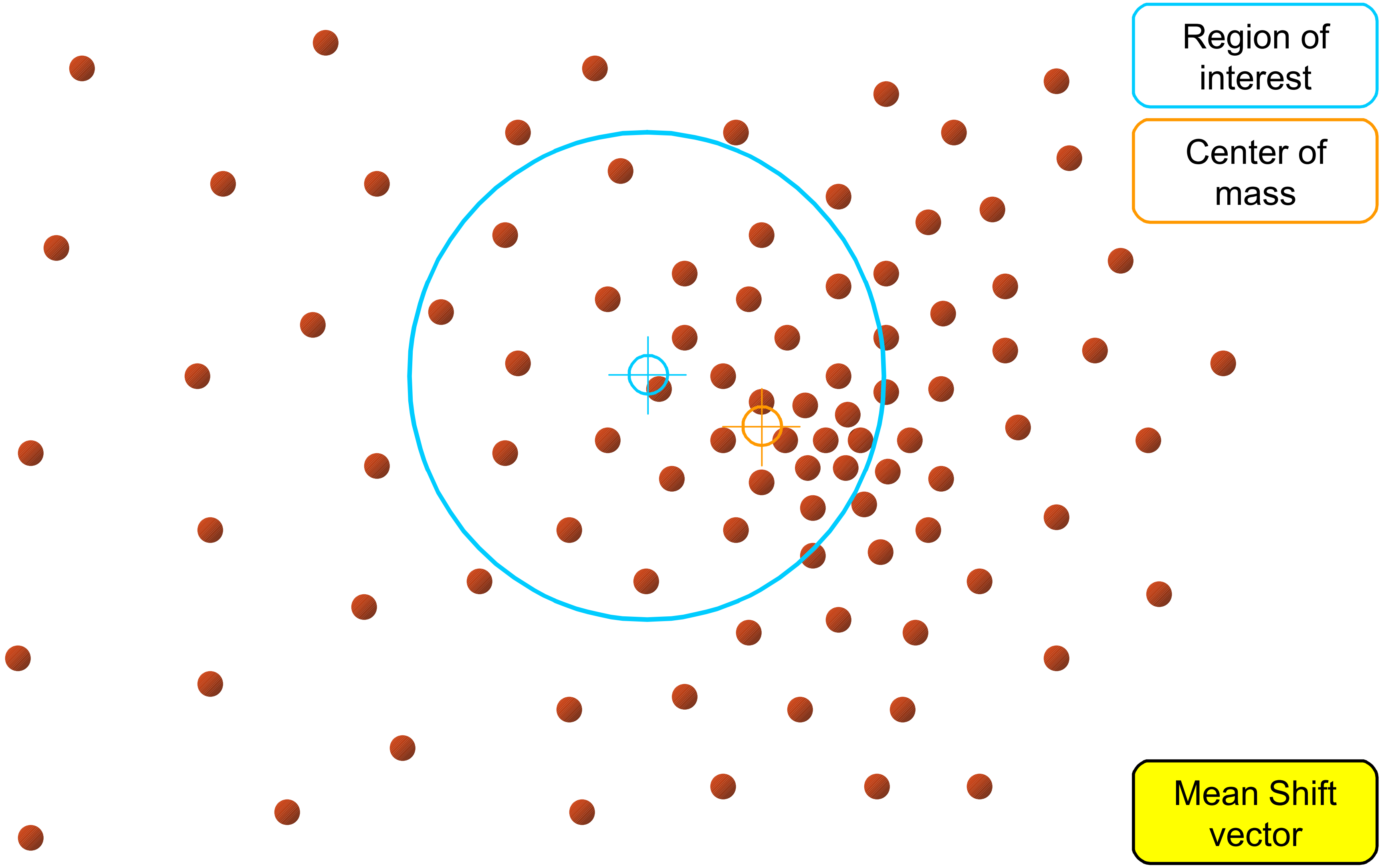
Intuitive Description



Objective : Find the densest region

Ukrainitz&Sarel, Weizmann

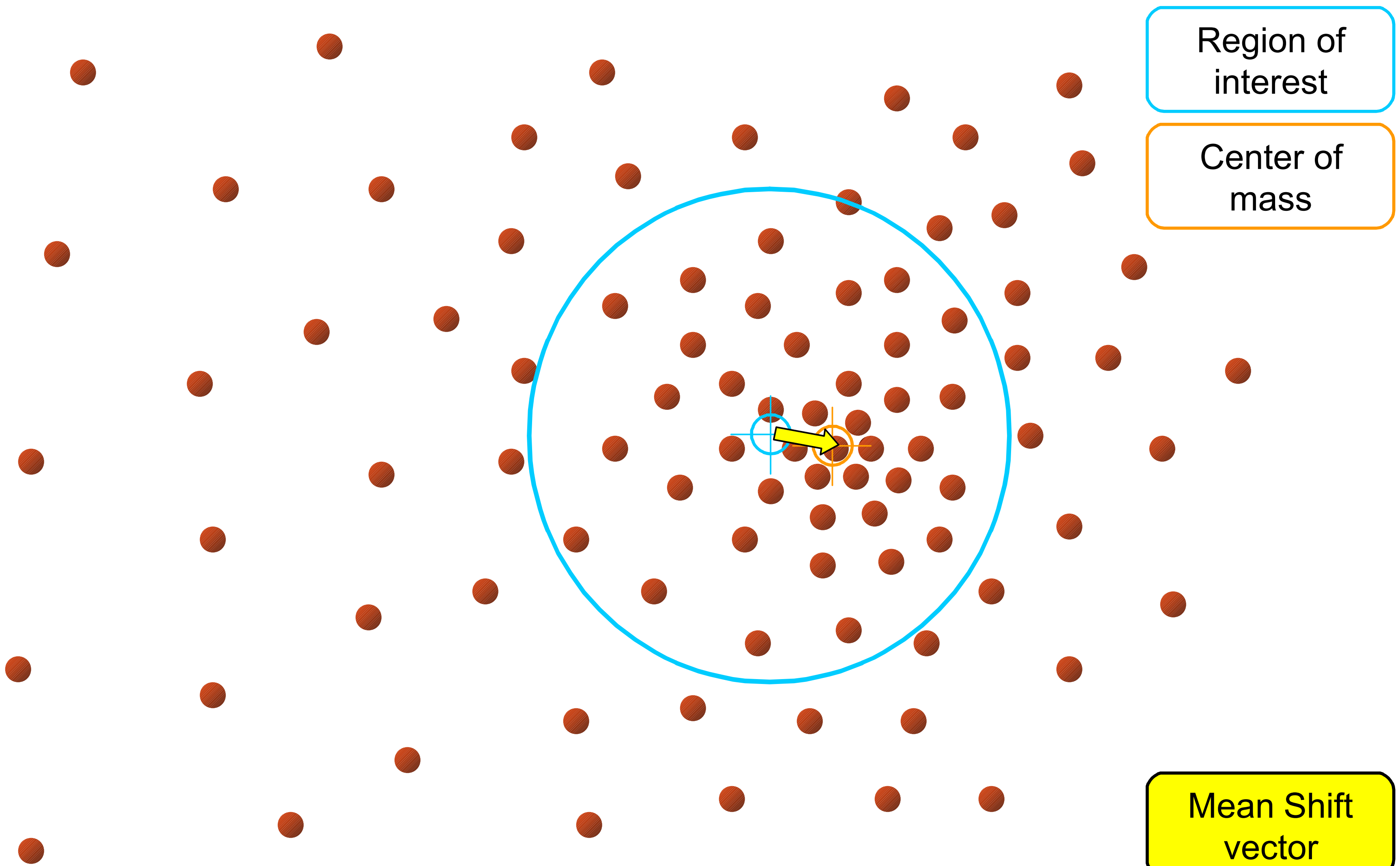
Intuitive Description



Objective : Find the densest region

Robert Collins
CSE486, Penn State

Intuitive Description

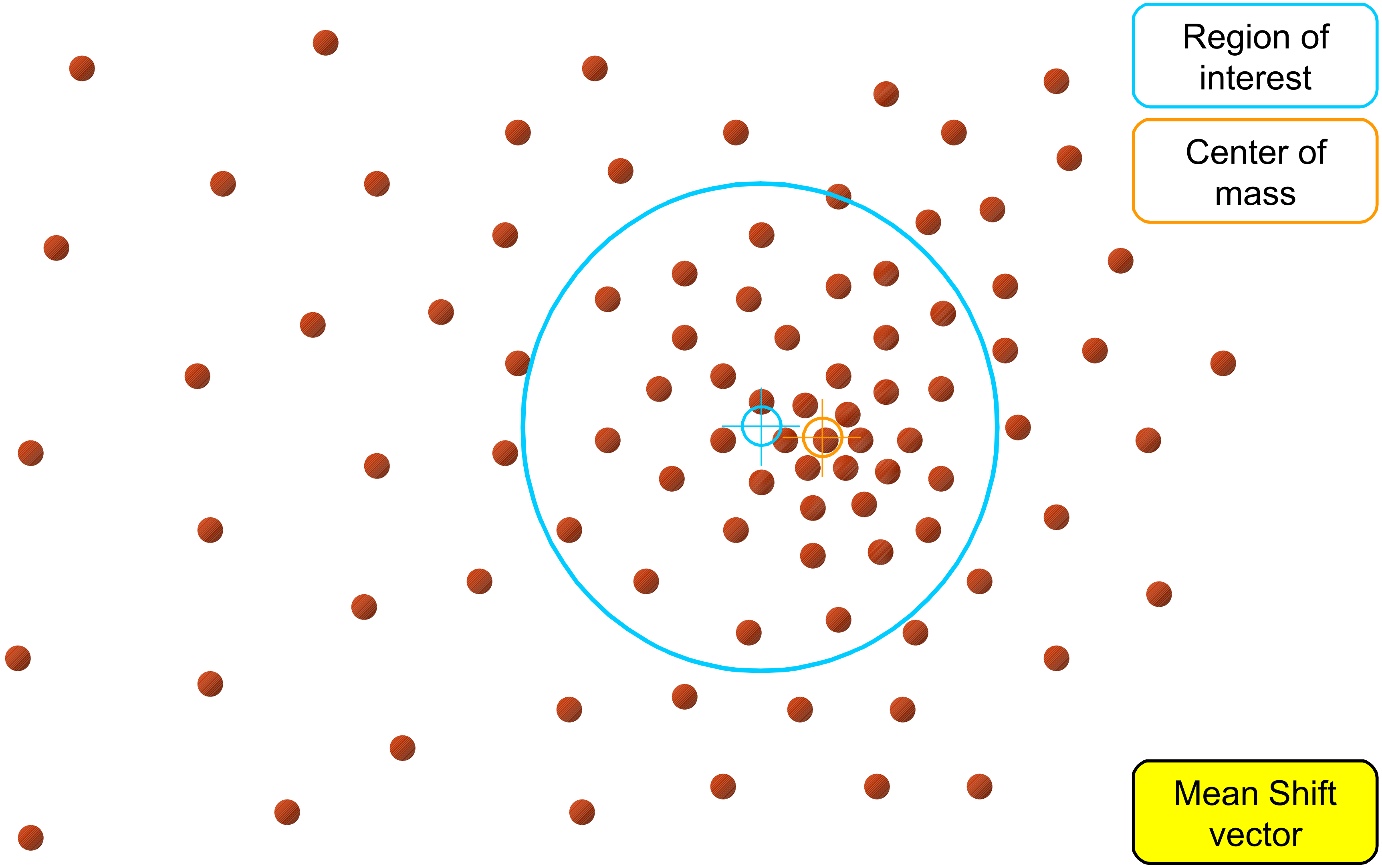


Objective : Find the densest region

Ukrainitz&Sarel, Weizmann

Robert Collins
CSE486, Penn State

Intuitive Description

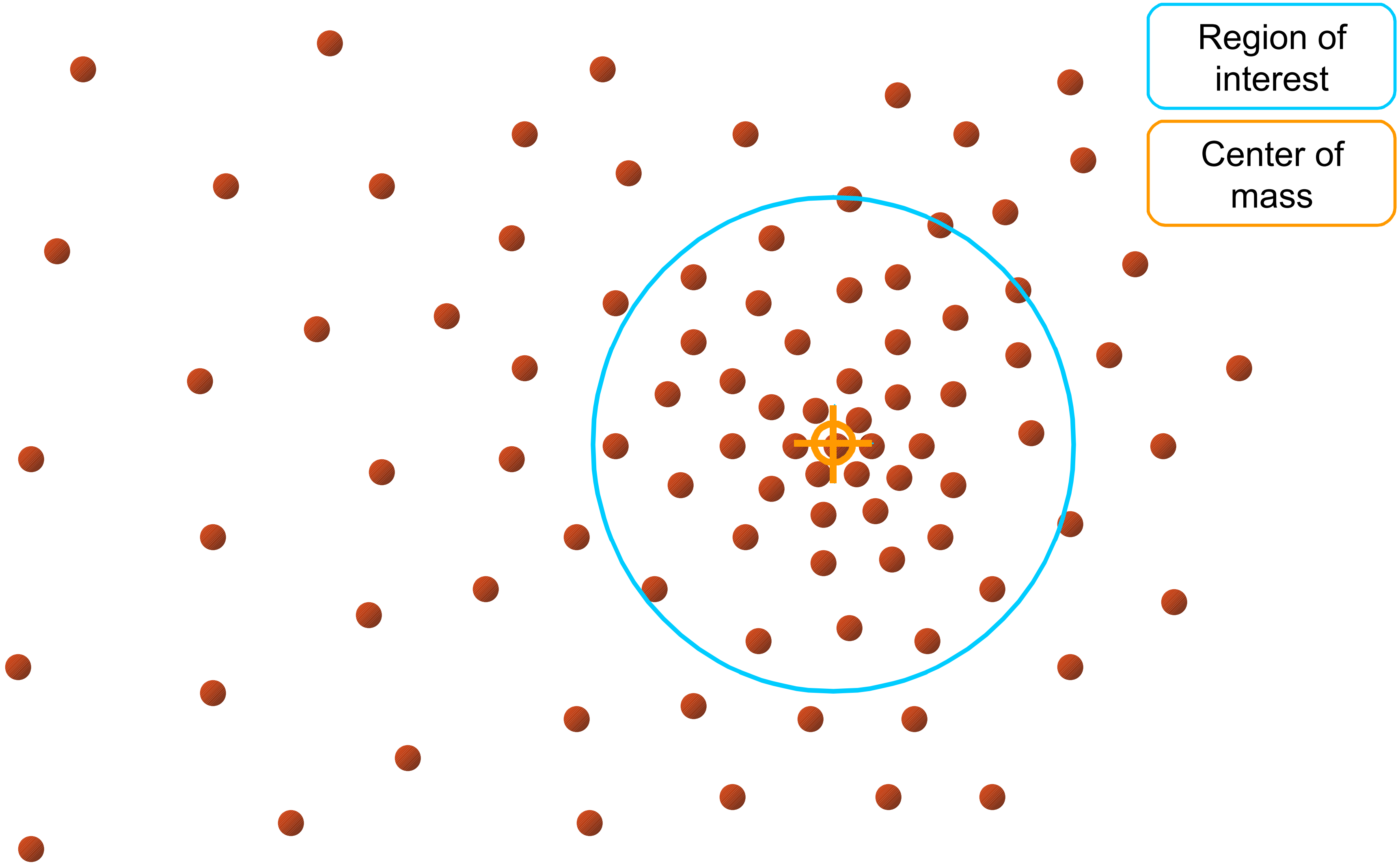


Objective : Find the densest region

Ukrainitz&Sarel, Weizmann

Robert Collins
CSE486, Penn State

Intuitive Description



Objective : Find the densest region

Ukrainitz&Sarel, Weizmann

Using Mean-Shift on Color Models

Two approaches:

1) Create a color "likelihood" image, with pixels weighted by similarity to the desired color (best for unicolored objects)

2) Represent color distribution with a histogram. Use mean-shift to find region that has most similar distribution of colors.

Mean-shift on Weight Images

Ideally, we want an indicator function that returns 1 for pixels on the object we are tracking, and 0 for all other pixels

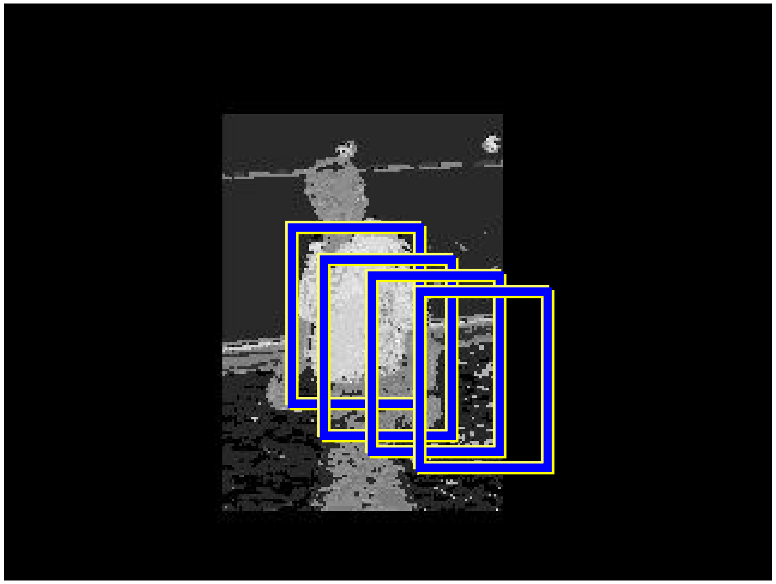
Instead, we compute likelihood maps where the value at a pixel is proportional to the likelihood that the pixel comes from the object we are tracking.

Computation of likelihood can be based on

- color
- texture
- shape (boundary)
- predicted location

Mean-Shift Tracking

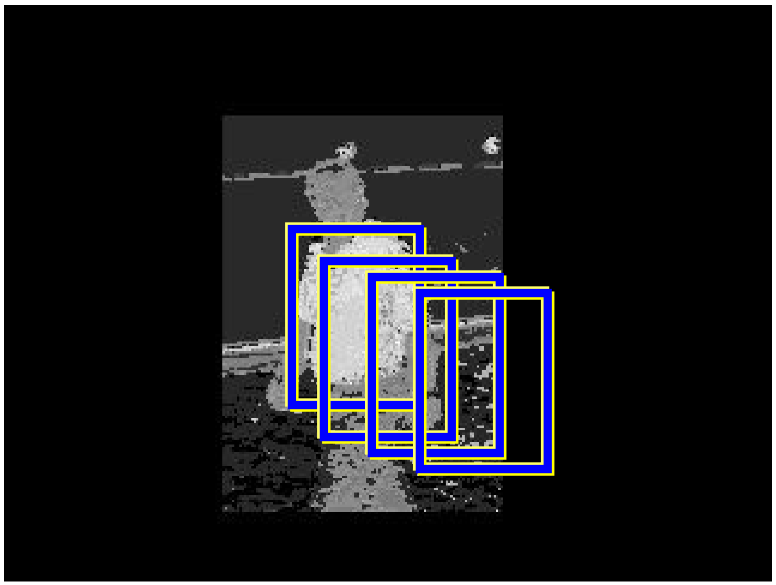
Let pixels form a uniform grid of data points, each with a weight (pixel value) proportional to the "likelihood" that the pixel is on the object we want to track. Perform standard mean-shift algorithm using this weighted set of points.

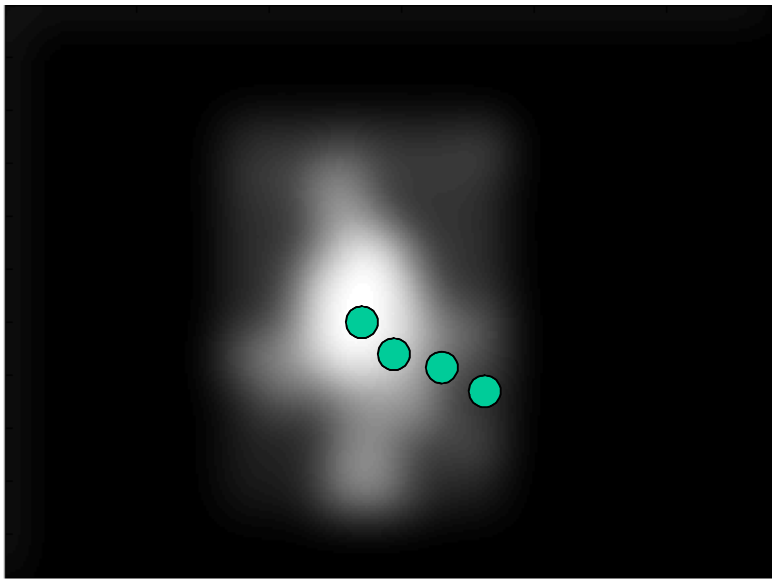


$$\Delta \mathbf{x} = \frac{\sum_a \mathbf{K(a-x)\ w(a)\ (a-x)}}{\sum_a \mathbf{K(a-x)\ w(a)}}$$

Nice Property

Running mean-shift with kernel K on weight image w is equivalent to performing gradient ascent in a (virtual) image formed by convolving w with some "shadow" kernel H.





Note: mode we are looking for is mode of location (x,y) likelihood, NOT mode of the color distribution!

Example: Face Tracking using Mean -Shift

Gray Bradski, "Computer Vision Face Tracking for use in a Perceptual User Interface," ***IEEE Workshop On Applications of Computer Vision*****, Princeton, NJ, 1998, pp.214-219.**

Figure 7: Orientation of the flesh probability distribution marked on the source video image

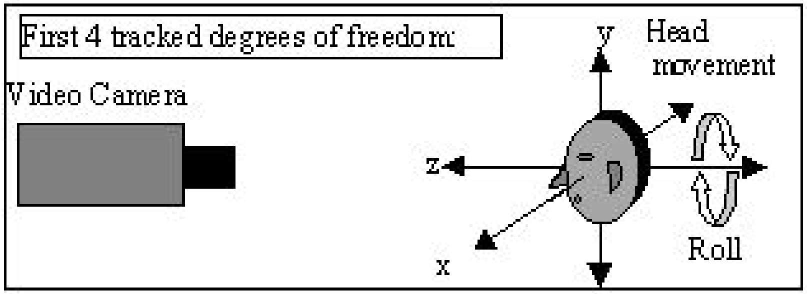
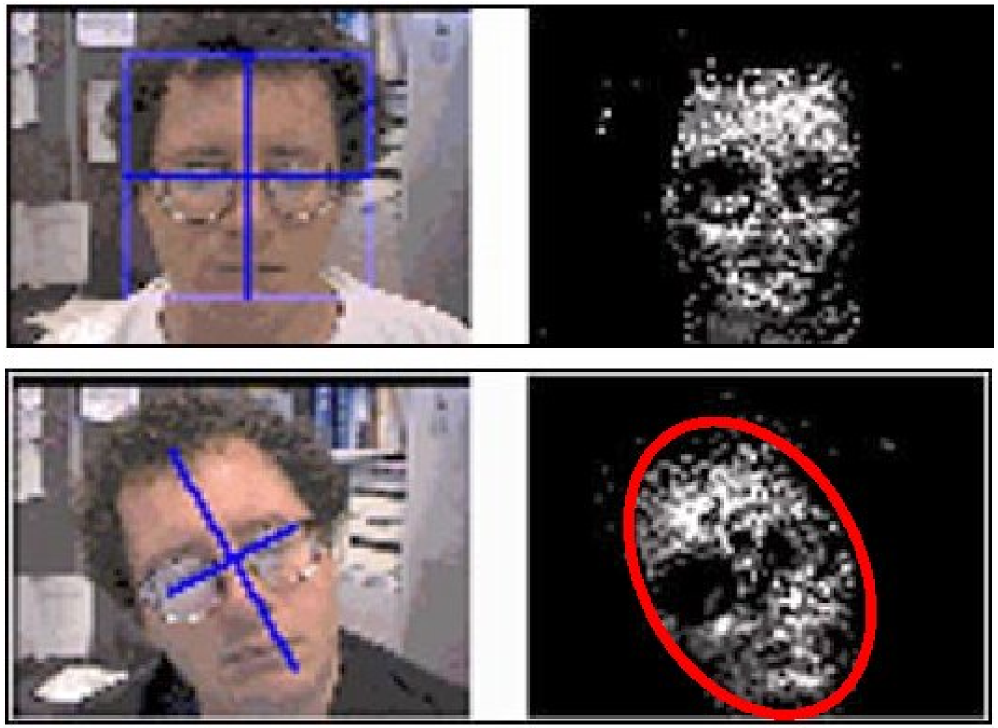


Figure 8: First four head tracked degrees of freedom: X, Y, Z location, and head roll

Robert Collins
CSE486, Penn State

Bradski's CamShift



**X,Y location of mode found by mean-shift.
Z, Roll angle determined by fitting an ellipse
to the mode found by mean-shift algorithm.**

CamShift Results

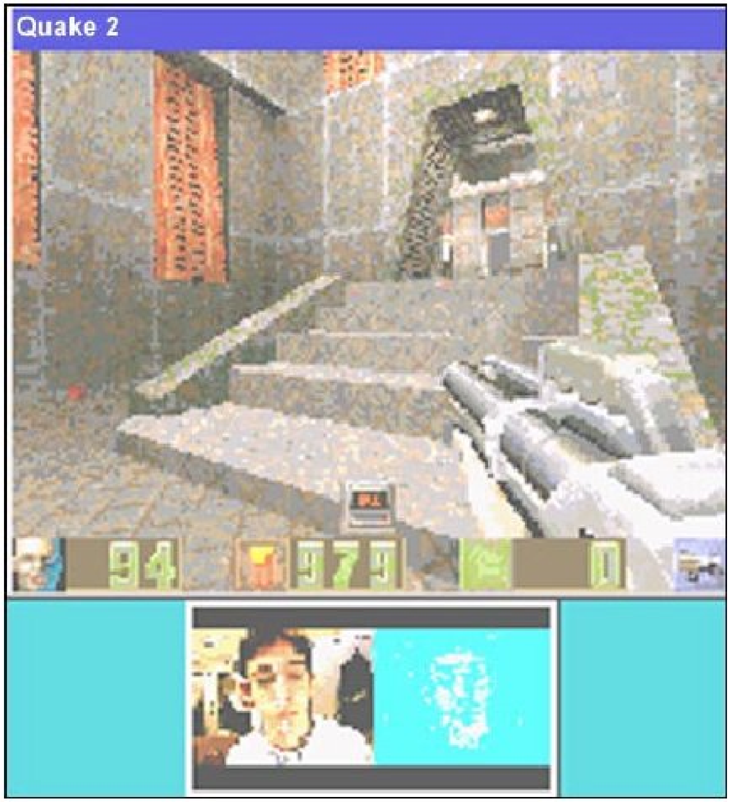
From Gary Bradski

Fast motion

Distractors

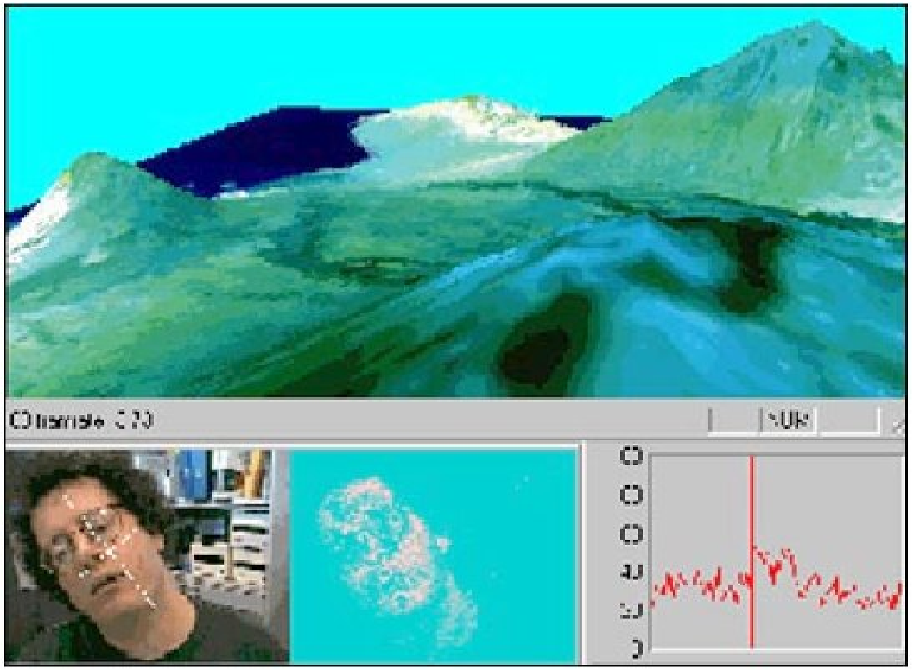
Robert Collins
CSE486, Penn State

CamShift Applications



Quake interface

Flight simulator



Using Mean-Shift on Color Models

Two approaches:

1) Create a color "likelihood" image, with pixels weighted by similarity to the desired color (best for unicolored objects)

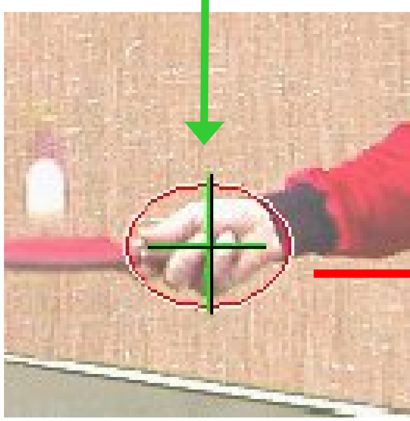
2) Represent color distribution with a histogram. Use mean-shift to find region that has most similar distribution of colors.

Robert Collins
CSE486, Penn State

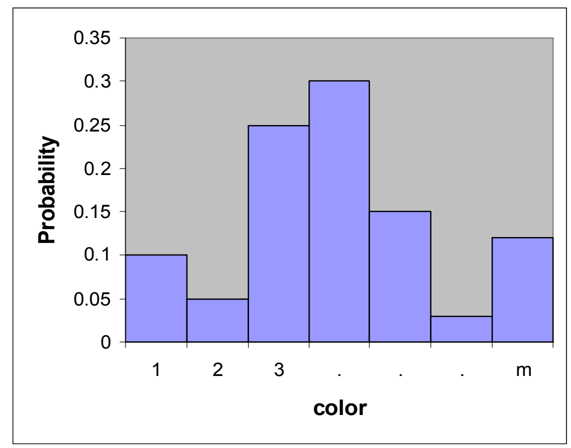
Mean-Shift Object Tracking

Target Representation

Choose a reference target model → Choose a feature space → Represent the model by its PDF in the feature space



Quantized Color Space

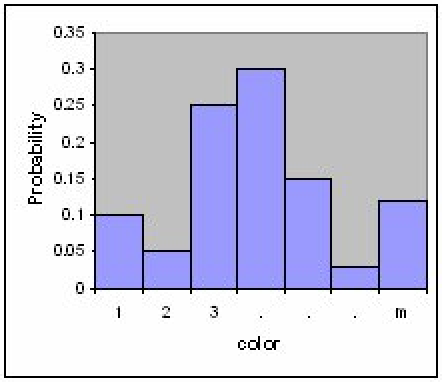


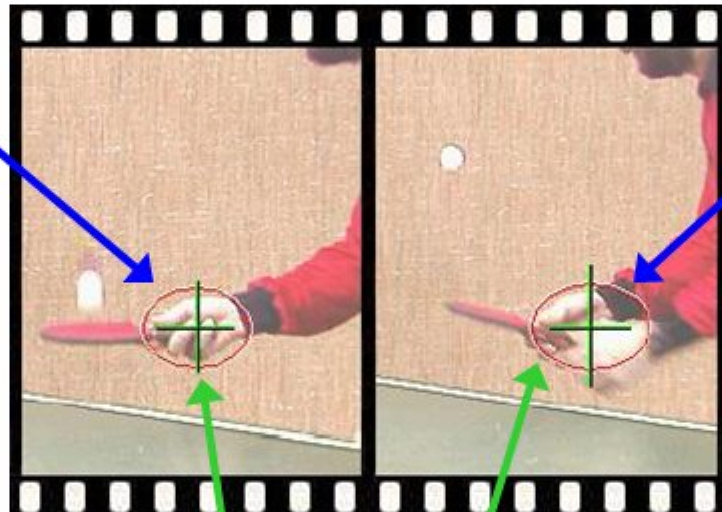
Kernel Based Object Tracking, by Comaniniu, Ramesh, Meer

Mean-Shift Object Tracking

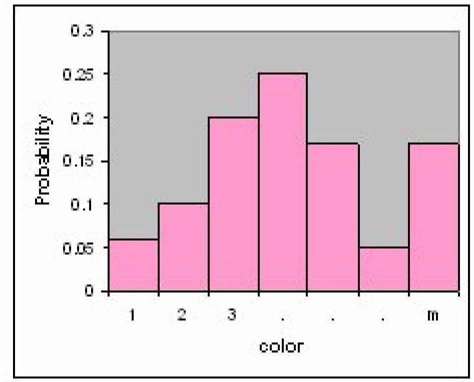
PDF Representation

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:

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

Ukrainitz&Sarel, Weizmann

Comparing Color Distributions

Bhattacharya Distance:

Given an n-bucket model histogram $\{m_i \mid i=1,\ldots,n\}$ and data histogram $\{d_i \mid i=1,\ldots,n\}$, we follow Comanesciu, Ramesh and Meer * to use the distance function:

$$\underbrace{\Delta(m,d) = \sqrt{1-\sum_{i=1}^{n}\sqrt{m_i \times d_i}}}_{\text{Similarity Function}}$$

Similarity Function

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

Why?

1) it shares optimality properties with the notion of Bayes error
2) it imposes a metric structure
3) it is relatively invariant to object size (number of pixels)
4) it is valid for arbitrary distributions (not just Gaussian ones)

*Dorin Comanesciu, V. Ramesh and Peter Meer, "Real-time Tracking of Non-Rigid Objects using Mean Shift," IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, 2000 (best paper award).

Glossing over the Details

Spatial smoothing of similarity function by introducing a spatial kernel (Gaussian, box filter)

Take derivative of similarity with respect to colors. This tells what colors we need more/less of to make current hist more similar to reference hist.

Result is weighted mean shift we used before. However, the color weights are now computed "on-the-fly", and change from one iteration to the next.

Mean-Shift Object Tracking

Results

From Comaniciu, Ramesh, Meer

Feature space: 16×16×16 quantized RGB
Target: manually selected on 1st frame
Average mean-shift iterations: 4

Mean-Shift Object Tracking

Results

Partial occlusion

Distraction

Motion blur

Ukrainitz&Sarel, Weizmann

Robert Collins
CSE486, Penn State

Mean-Shift Object Tracking

Results

From Comaniciu, Ramesh, Meer

Ukrainitz&Sarel, Weizmann

Mean-Shift Object Tracking

Results

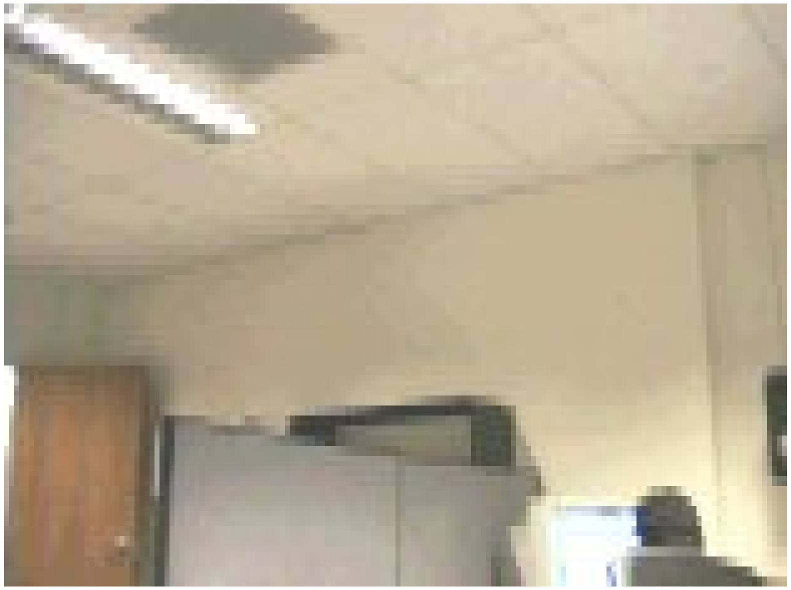
From Comaniciu, Ramesh, Meer

Feature space: 128×128 quantized RG

Mean-Shift Object Tracking

Results

The man himself…



From Comaniciu, Ramesh, Meer

Feature space: 128×128 quantized RG