

1 - IMPLEMENTATION

Mean Shift is a nonparametric technique that constitutes a robust approach to the analysis of complex multi-modal feature spaces. The basic idea behind mean shift algorithm is to climb the gradient of a probability distribution to find the nearest dominant mode. It represents a robust method of finding local extrema in the density distribution of a data set and, besides being used to segment image regions it has been successfully applied to the problem of object tracking in computer vision.

Here, We will present object tracking application of the Mean-shift algorithm. The basic idea behind the Mean-shift algorithm for object tracking is;

First we generate a feature descriptor of a target model as PDF (Probability Density Function), then we try to find candidate in the sequential frames. We define similarity between target and candidate as minimising distance between their PDF. To minimise distance we maximise Bhattacharrya coefficient between two feature vectors. While we are maximising this coefficient mean-shift concept comes in, since likelihood maximisation depends on maximising weights and we state that weights as non-parametric density gradient estimation is equal to mean-shift and we find peaks (modes) for maximisation. The scan for finding likelihood takes places in a pre-defined ROI (region of interest) and we update the ROI every frame when we find the possible candidate.

To define the feature descriptor as PDF of both target and candidates we should explain feature space first. Then we define similarity between target and candidate and give the overview of the object tracking algorithm using Mean-shift. Later, mentioned concepts and mathematical derivations will be given.

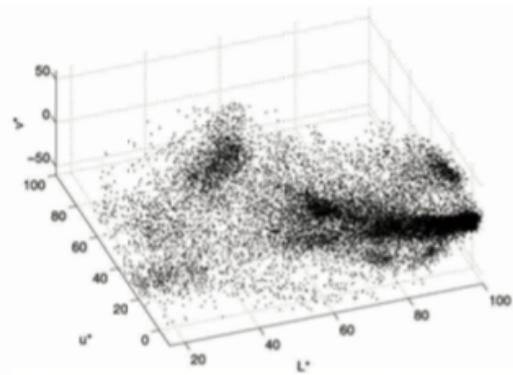
1.1 - Feature space & Density:

We can define feature as a representation of internal structure in another domain of interest."A *feature space* is a mapping of the input obtained through the processing of the data in small subsets at a time. For each subset,a parametric representation of the feature of interest is obtained and the result is mapped into a point in the multidimensional space of the parameter." [1] As we process the input as a content, the change of domain will be application specific. As a result we obtain a mapping between multidimensional space of the parameters. "The nature of the feature space is application dependent. The subsets employed in the mapping can range from individual pixels, as in the color space representation of an image,to a set of quasi-randomly chosen data points,as in the probabilistic Hough transform. Both the advantage and the disadvantage of the feature space paradigm arise from the global nature of the derived representation of the input." [1] Arbitrarily structured feature spaces can be analysed only by nonparametric methods since these methods do not have embedded assumptions.

An example of feature space representation is given below. Density concentrations can be seen easily in the new domain space;



(a)



(b)

Fig. 1. Example of a feature space. (a) A 400×276 color image. (b) Corresponding $L^*u^*v^*$ color space with 110,400 data points.

Image taken from: Dorin Comaniciu & and Peter Meer, "Mean Shift" [1]

"The rationale behind the density estimation-based non-parametric clustering approach is that the feature space can be regarded as the empirical probability density function p.d.f.) of the represented parameter. Dense regions in the feature space thus correspond to local maxima of the p.d.f., that is, to the modes of the unknown density. Once the location of a mode is determined, the cluster associated with it is delineated based on the local structure of the feature space." [1] We will highly benefit from this concept in our algorithm using iterative mean-shift convergence as it's a mathematically proven implementation of finding the modes of the non-parametric density gradient estimation.

1.2 - Object tracking application:

The idea behind object tracking using mean-shift algorithm includes following measurements;

Object tracking application idea

(1) Representing the target in initial frame;

After we chose target modal, we represent it in arbitrary feature space as PDF (Probability Density function). Here we see feature space is a “Quantized Color Space”



(2) Target localisation tracking;

1st frame shows ROI and it's centered target modal. In the sequential 2nd (video frames) we try to find the best candidate in the same ROI. After we find the candidate we update the ROI accordingly as can be seen in the 3rd frame.



(3) PDF representation of target and candidate;

Target and candidate feature space representations in PDFs are shown as m-bin colour histograms.



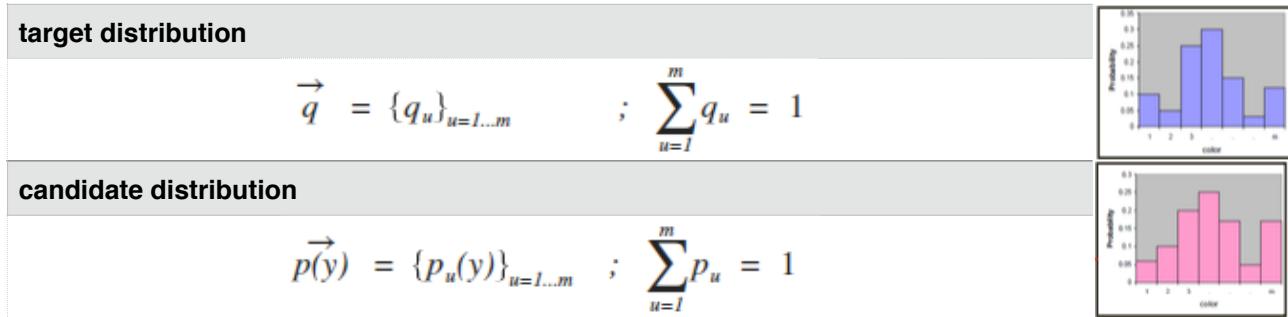
Images are taken from: UCF - Mean Shift Tracking [3]

As shown above target and candidate patches are represented in feature-space as a PDF. Object tracking suitable options for feature-space representations are;

- **intensity**
- **colour**
- **gradient**

Mean-shift object tracking

These PDFs are which the Mean-shift concept will be applied on. If we represent target and candidate in, let's say colour distribution;



The object here is to define a similarity between target and candidate, which is defined with Bhattacharrya coefficient.

So to summarise the idea of object-tracking application is;

- 1 - Select a ROI around the target location in current frame
- 2 - Find the most similar candidate based on the **similarity** function
- 3 - Update ROI in the next frame centering the candidate.

1.3 - Similarity & Bhattacharrya coefficient & Distribution:

- The similarity between object target model feature distribution \mathbf{q} and candidate feature distribution \mathbf{p} are calculated using Bhattacharrya coefficient.
- Distance between object target model \mathbf{q} and candidate \mathbf{p} is defined as square-root of one minus Bhattacharrya coefficient.
- Features probability distribution are calculated by using weighted histograms (Epanechnikov profile).

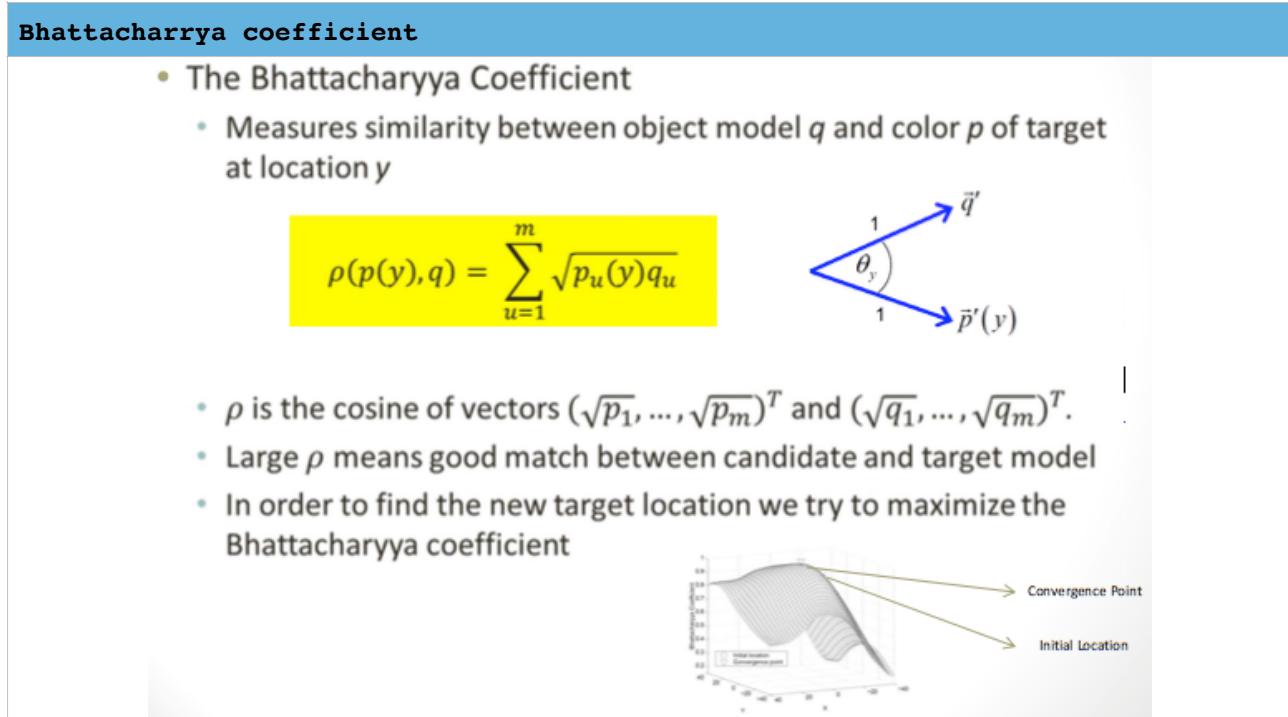


Image taken from: UCF - Mean Shift Tracking [3]

Given feature distributions for target and candidate \mathbf{q} and \mathbf{p} , the distance is defined as;

$$d(\mathbf{y}) = \sqrt{1-\rho(\mathbf{y})} ; \rho(\mathbf{y}) \rightarrow \text{Bhattacharya coef.}$$

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

While we minimise the distance to find best candidate we actually aim to maximise Bhattacharrya coef. to maximise the likelihood. Analytical expression for maximising Bhattacharrya coef. benefits from Taylor series. To expression for expanding Taylor series around initial estimate \mathbf{y}_0 is given below;

Distance is given as;

$$d(\mathbf{y}) = \sqrt{1-\rho(\mathbf{y})} ; \rho(\mathbf{y}) \rightarrow \text{Bhattacharya coef.}$$

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

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

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

Maximizing *Bhattacharya coef.* can be obtained by maximizing the second term since first term is constant for given \mathbf{y}_0 initial.

Here \hat{p}_u actually corresponds to distribution of the particular value in the PDF representation, and it's relation can be given as;

$$p(u) = C \sum_{x_i \in S}^{n_s} k(\|x_i\|^2) \delta[S(x_i) - u] ; k \rightarrow \text{profile}$$

The second term gives us;

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

h : radius of sphere

C_h : normalization constant

$S(x_i)$: feature level at x

\mathbf{y} : kernel center

m : number of binds

Maximisation of the likelihood of distributions actually corresponds to maximising weights expressed in a relation of target and candidate distribution.

"Thus, to minimise the distance (6), the second term in (9) has to be maximised, the first term being independent of y . Observe that the second term represents the density estimate computed with kernel profile $k(x)$ at y in the current frame, with the data being weighted by w_i (10). The mode of this density in the local neighbourhood is the sought maximum that can be found employing the mean shift procedure. In this procedure, the kernel is recursively moved from the current location y_0 to the new location y_1 ..."[2]

It's the intuitive idea of the mean-shift tracking since iteratively generating the mean-shift vector equals to finding modes (peaks) of the non-parametric density gradient estimation;

And as found from the second term above, here maximization of the likelihood of target and candidate depends on maximizing weights given as;

$$w_i = \sum_{u=1}^m \delta[S(\mathbf{x}_i) - u] \sqrt{\frac{q_u}{\hat{p}_u(y_0)}} \quad ; \quad 0 \leq w_i \leq 1$$

Provided with w_i mean-shift vector can be written as;

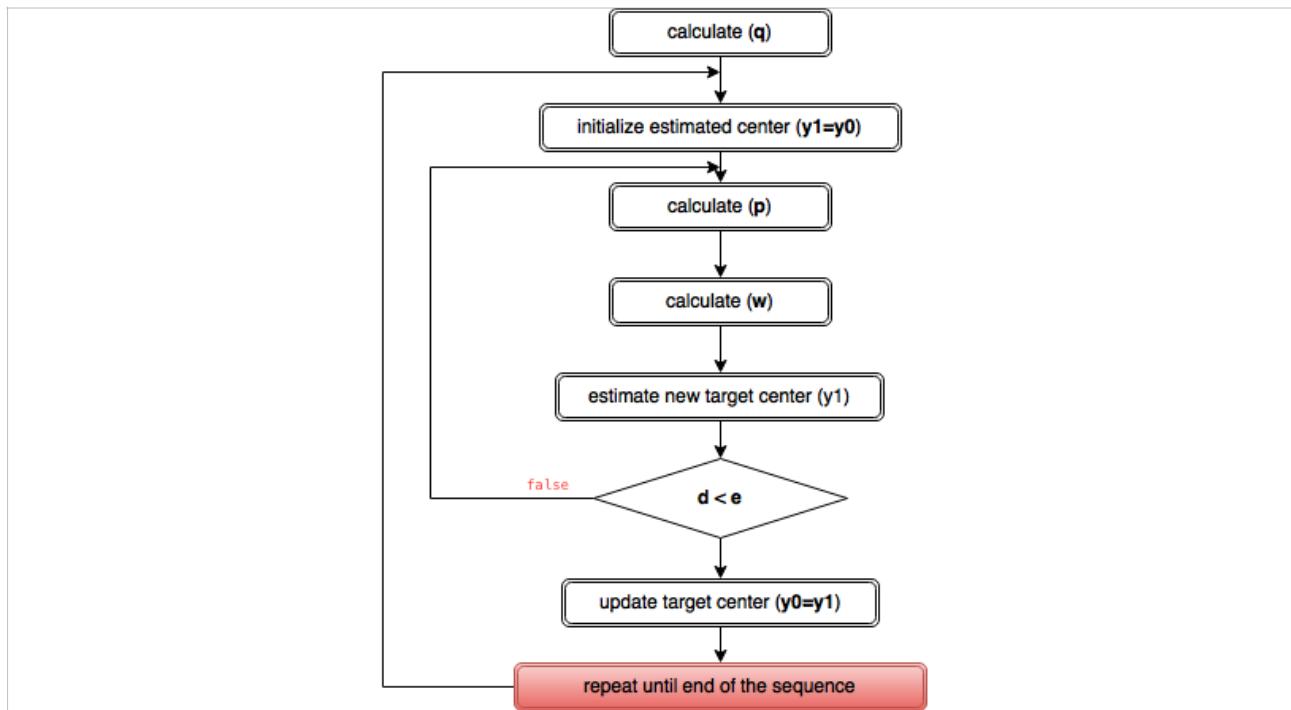
$$M_h(\mathbf{y}_0) = \frac{\sum_{i=1}^{n_s} w_i(\mathbf{y}_0) \mathbf{x}_i}{\sum_{i=1}^{n_s} w_i(\mathbf{y}_0)} - \mathbf{y}_0$$

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

Following on our derivations, essentially to find out where the best candidate, based on the contribution of each of the pixel in the window \mathbf{x}_i , where its contribution with weight w_i calculated as the likelihood of the target and candidate feature distributions, we iteratively shift until we find the mode of the similarity.

1.4 - Algorithm:

Mean-shift concept and the density gradient estimation will be given in the following section, but first let's look the overview of the object tracking algorithm. Here the mean-shift iteration continues until the distance d between target and candidate distributions is below a defined delta-error ϵ .



1.5 - Mean-shift & density gradient estimation:

Mean-shift is defined as; given ROI with starting from initial location at every iteration shifting occurs to the densest point until shift is zero or below pre-defined epsilon value. Every-time shift occurs, ROI is updated accordingly in an iterative fashion.

Mean-shift idea	
Starting with; <ul style="list-style-type: none"> - an arbitrary ROI - initial location 	<p>The diagram shows a collection of red dots representing a 'Region of interest'. A blue circle, labeled 'Center of mass', encloses a cluster of dots. A yellow arrow points from the center of the blue circle to a specific dot within it, labeled 'Mean Shift vector'. The text 'Objective : Find the densest region' and 'Distribution of identical billiard balls' is overlaid on the diagram.</p>

Images are taken from: UCF - Mean Shift Tracking [3]

The equation for mean-shift vector is given below;

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

- Data points and approximate location of the mean of this data are given.
- Estimating the exact location of the mean of the data by determining the shift vector from the initial mean (iteratively until mean-shift vector $< \epsilon$).

Mean-shift vector always point towards to the direction of the max. increase in the density.

Mean-shift vector can be given in a weighted fashion. Weights are determined using kernels (masks). For instance following kernels may considered;

- Uniform
- Gaussian
- Epanechnikov

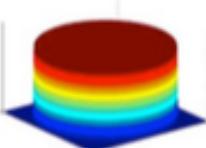
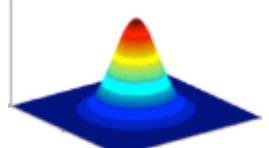
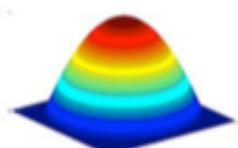
$$M_h(\mathbf{y}_0) = \frac{\sum_{i=1}^{n_x} w_i(\mathbf{y}_0) \mathbf{x}_i}{\sum_{i=1}^{n_x} w_i(\mathbf{y}_0)} - \mathbf{y}_0$$

n_x : num. points in the kernel

\mathbf{y}_0 : initial mean location

\mathbf{x}_i : data points

h : kernel radius

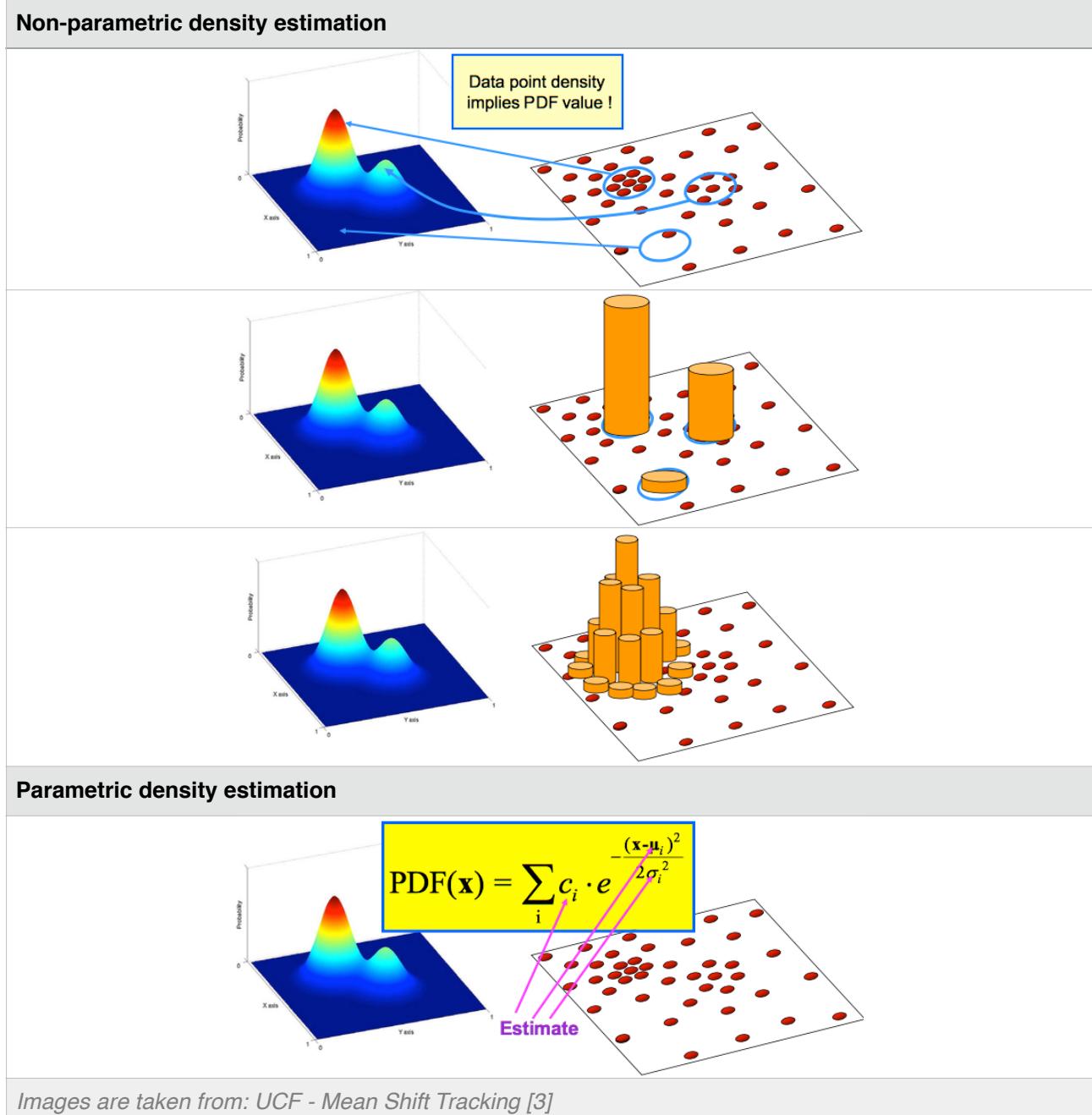
parameterized kernels		
Uniform	Gaussian	Epanechnikov
$K_U(\mathbf{x}) = \begin{cases} c & \ \mathbf{x}\ \leq 1 \\ 0 & \text{otherwise} \end{cases}$	$K_N(\mathbf{x}) = c \exp\left(-\frac{1}{2} \ \mathbf{x}\ ^2\right)$	$K_E(\mathbf{x}) = \begin{cases} c(1 - \ \mathbf{x}\ ^2) & \ \mathbf{x}\ \leq 1 \\ 0 & \text{otherwise} \end{cases}$
		

$P(x) = \frac{1}{n} \sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i)$	Probability distribution; function of some finite number of data points $\mathbf{x}_1 \dots \mathbf{x}_n$ defined as sum of kernel distances
----------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------

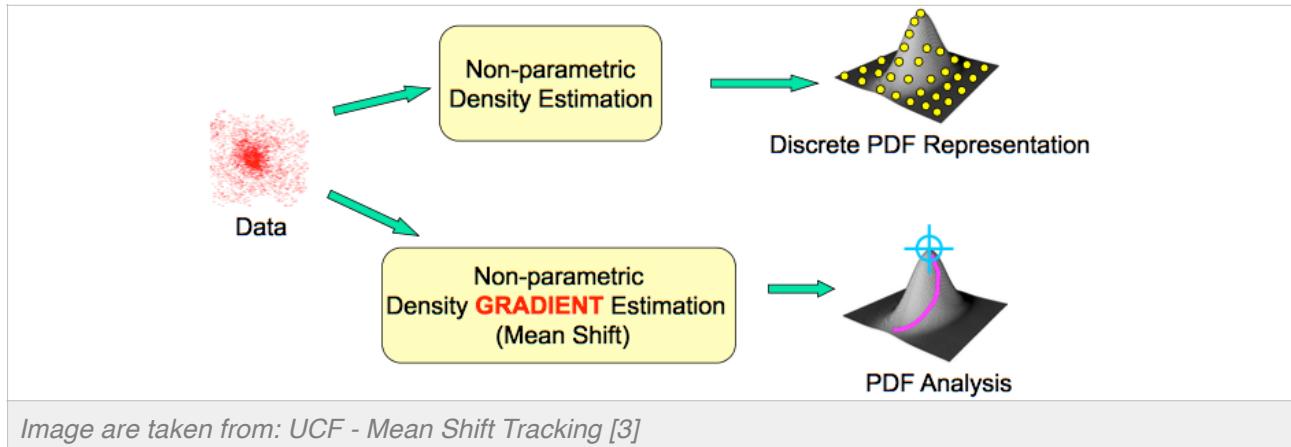
Mean-shift Properties;

- Mean-shift vector has the direction of the gradient of the density estimate.
- It is computed iteratively for obtaining the max. density in the local neighbourhood.

For a given arbitrary data set, explaining its probability distribution in analytic form is not a possible case always. When the probability distribution can be defined with a closed form, it's discrete PDF representation is given with a non-parametric density estimation. Non-parametric density estimation requires memory space for storing all the data analysed while it's not the case when distribution is manifested in a closed analytic form. Meanwhile in parametric estimation the assumption is the data points sampled manifest an underlying PDF.



Here mean-shift is a tool for finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in a region \mathbf{R} .



The relation for mean-shift to density gradient estimation is given below, which states gradient of the density estimate is equal to mean-shift;

A profile is defines for a radially symmetric kernel

Radially symmetric kernel: $K(\mathbf{x}) = ck(\|\mathbf{x}\|^2)$; $k \rightarrow \text{profile}$

$$P(\mathbf{x}) = \frac{1}{n} \sum_i K(\mathbf{x}-\mathbf{x}_i) = \frac{1}{n} c \sum_i k(\|\mathbf{x}-\mathbf{x}_i\|^2)$$

Gradient of the density estimate is equal to mean-shift;

$$\begin{aligned} P(\mathbf{x}) &= \frac{1}{n} c \sum_i k(\|\mathbf{x}-\mathbf{x}_i\|^2) \\ \nabla P(\mathbf{x}) &= \frac{1}{n} c \sum_i \nabla k(\|\mathbf{x}-\mathbf{x}_i\|^2) \\ &= \frac{1}{n} 2c \sum_i (\mathbf{x}-\mathbf{x}_i) \hat{k}(\|\mathbf{x}-\mathbf{x}_i\|^2) \\ &= \frac{1}{n} 2c \sum_i \mathbf{x}_i g(\|\mathbf{x}-\mathbf{x}_i\|^2) - \frac{1}{n} 2c \sum_i \mathbf{x} g(\|\mathbf{x}-\mathbf{x}_i\|^2) ; g(x) = -\hat{k}(x) \\ &= \frac{1}{n} 2c \sum_i g(\|\mathbf{x}-\mathbf{x}_i\|^2) \left[\frac{\sum_i \mathbf{x}_i g(\|\mathbf{x}-\mathbf{x}_i\|^2)}{\sum_i g(\|\mathbf{x}-\mathbf{x}_i\|^2)} - \mathbf{x} \right] \\ &= \frac{1}{n} 2c \sum_i g \left[\frac{\sum_i \mathbf{x}_i g_i}{\sum_i g_i} - \mathbf{x} \right] \quad ; g_i = (\|\mathbf{x}-\mathbf{x}_i\|^2) \end{aligned}$$

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

$$\nabla P(\mathbf{x}) = \frac{c}{n} \sum_i g_i \times m(\mathbf{x})$$

$$m(\mathbf{x}) = \frac{\nabla P(\mathbf{x})}{\frac{c}{n} \sum_i g_i} \quad (\text{mean-shift})$$

Mean-shift mode detection can stuck in saddle points. To overcome mode detection staying local plateaus while trying find peaks we use update mean-shift procedure.[3];

- Find all modes using the simple mean-shift procedure
- Prune modes by perturbing them (find saddle points and plateaus)
- Prune nearby - take the highest in the window

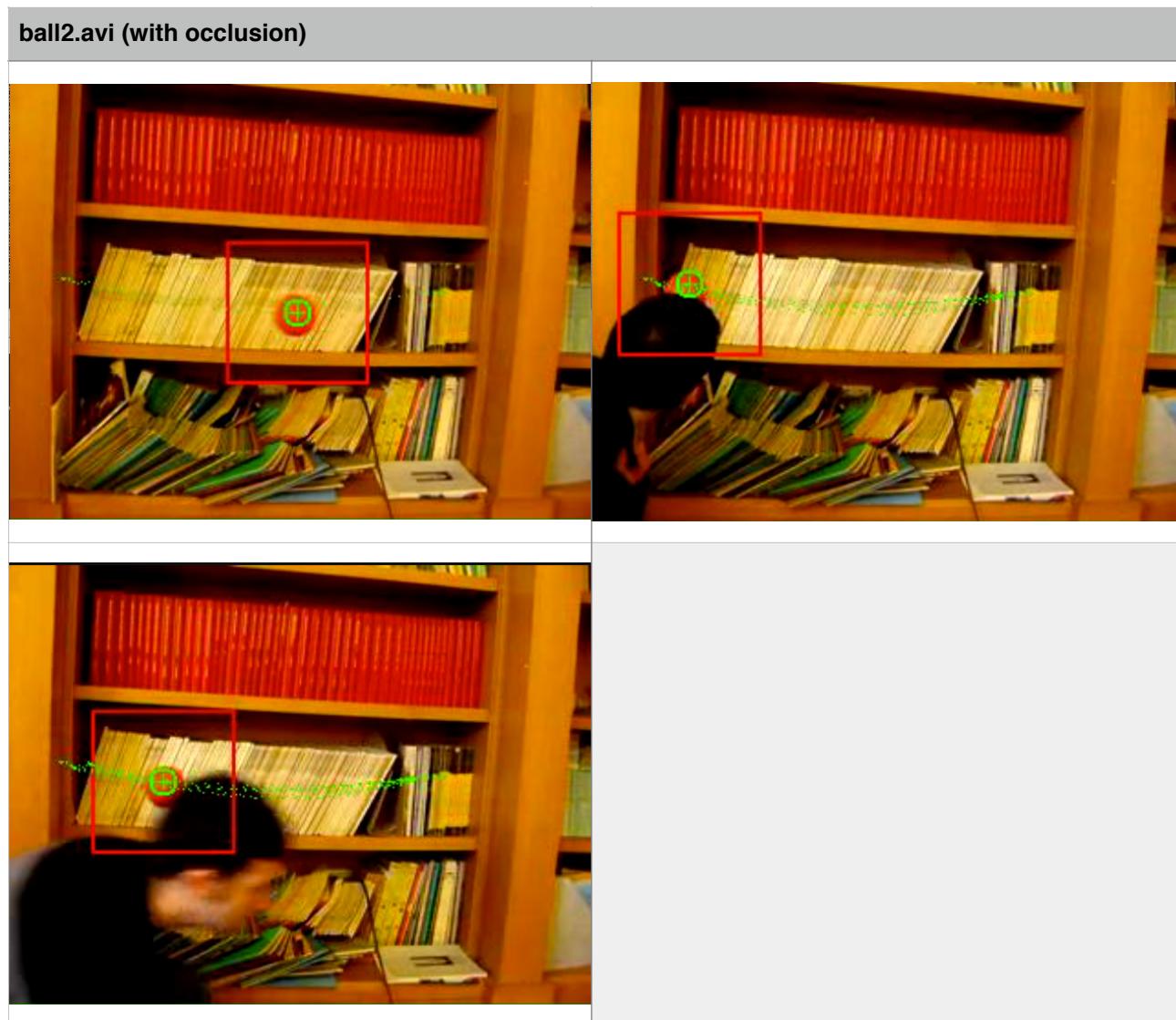
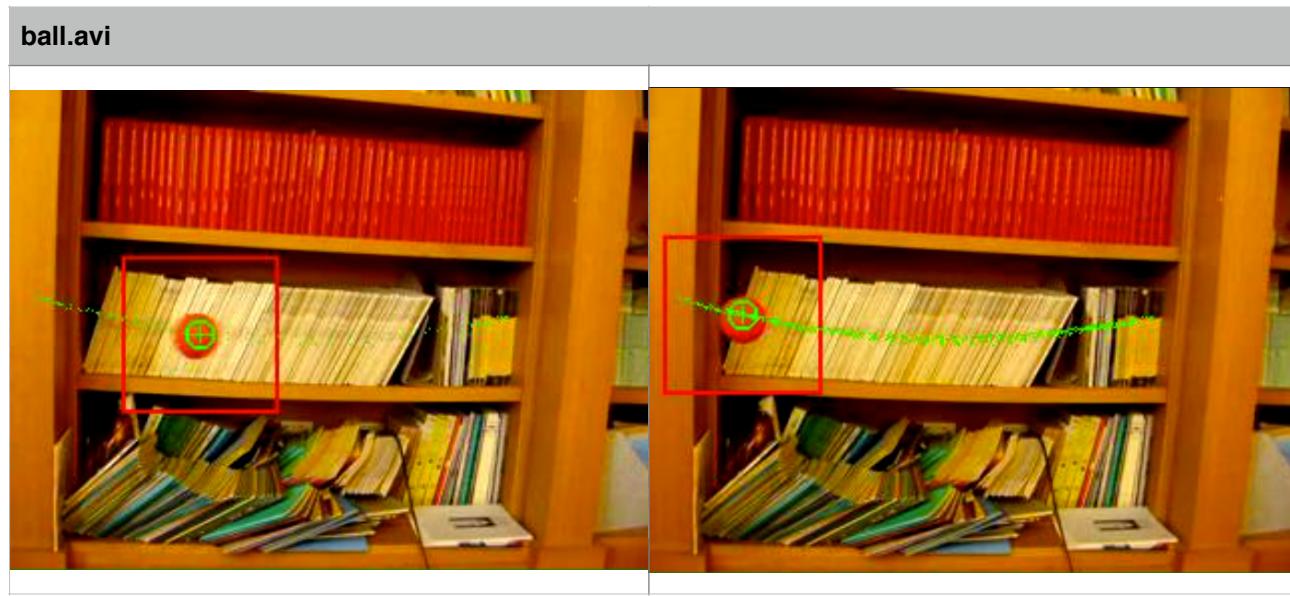
We run the process in parallel with multiple initial estimates of the mean for mean-shift in real modality analysis. Hopefully, we will have more than average same convergence to the densest location. More estimation will converge to the right densest region.

Mean-shift mode detection convergence;

- Automatic converge speed (mean-shift vector size depends on gradient)
- Near maxima -> the steps are small and refined
- Converge 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

2 – ANALYSIS

The results of the MATLAB implementation using colour of histograms for object distributions on test videos can be seen below;



Mean-shift object tracking

It should be noted the decision about feature distribution for mean-shift algorithm is important since clustering likelihood estimation will be based on this decision after transforming to new domain space.

The implementation gives good results, to be specific;

- There is no huge occlusion happening in the video frames (except some minor occlusion in **ball2.avi** video), so mean-shift vectors never lose the original target model.

- There is a direct separation between object modal and the background. This sparsity make it easy to estimate distances between target and candidate and the find the best similarities.

The idea for mean-shift algorithm was;

(1) generating two feature vectors as PDF of feature distributions after we transform to feature space

(2) Minimise distance between vectors -> Find the modes by maximising distributions

(3) mean-shift (to dense region) by applying the algorithm.

The first consideration about the implementation is scale related issues. If the objects moves too fast between frames or if fps of video is not enough to track the object we will have problems since the size the ROI is constant.

Secondly, and more importantly if the size of the target object changes like maybe the camera zooms in/zooms out we will have problem again since kernel radius (**bandwidth**) defined as **h** is constant too. "The influence of the bandwidth parameter **h** was assessed empirically in through a simple image segmentation task. In a more rigorous approach, however, four different techniques for bandwidth selection can be considered." [1] More about bandwidth selection and adaptation techniques can be gathered from *Dorin Comaniciu & and Peter Meer, "Mean Shift"* [1] and other follow-up works.

REFERENCES

[1] Dorin Comaniciu & and Peter Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis"; IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002

[2] Dorin Comaniciu & Visvanathan Ramesh & Peter Meer, "Kernel-Based Object Tracking"; IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 5, MAY 2003

[3] UCF - Center for Research in Computer Vision - Mean Shift Tracking; <http://crcv.ucf.edu/courses/CAP5415/Fall2012/Lecture-11-MeanShiftTracking.pdf>