Registration and Tracking by using Mean Shift Algorithm

Kavya G[1], Jharna Majumdar[2]

[1]M Tech, Dept of CSE, Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, Karnataka, India

[2]Dean R&D, Professor and Head, Dept of CSE (PG), Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, Karnataka, India

Abstract: ***Object tracking is the problem of determining (estimating) the positions and other relevant information of moving objects like car, ball or human in a video. There are number of application of tracking surveillance, traffic monitoring, robot vision, animations. Registration is the basic step of tracking. This paper proposes mean shift algorithm which is an iterative method, efficient approach to track a non rigid object. Three different kernels are used for weight distribution, namely Uniform, Gaussian, Epanechnikov kernel. To find the Similarity between the consecutive frames Bhattacharyya coefficient is used. The proposed approach also gives efficient results with different size of searching window.***

Keywords: Registration, Object Tracking, Mean Shift, Video, Bhattacharyya Coefficient, kernel density estimation.

1. Introduction

Image Registration is a process of finding the location where optimal matching is obtained by matching a template image called the reference image over the searching region of an input image using a suitable similarity measures. The method although computationally intensive and robust but it is simple, straightforward, requires no priori information about the two images. Image registration has two important requirements, time and accuracy. Tracking is the process of locating object or human over a time using camera [1]. Tracking has two steps target detection and target tracking. Tracking starts by acquiring the initial image and select the object of interest. Acquire next sequence of images locate the target and track the object repeat this until end of frames. The use of object tracking has many vision applications such as automated video surveillance, traffic monitoring, robot vision, animation and security and defense areas. The survey shows huge number of algorithms designed and developed for object tracking [4]. This paper mainly focuses on mean shift algorithm for tracking of object. Mean shift algorithm is an iterative process. The Mean-Shift algorithm tracks by maximizing Bhattacharyya coefficient or minimizing a distance between two probability density functions (pdfs) [5] represented by a reference and candidate histograms. Since the histogram distance (similarity) does not depend on spatial structure of the search window, the method is suitable for deformable and articulated objects.

2. Basic Mean Shift Analysis

The basic idea of mean shift is to find the maximum density in the distribution. In order to illustrate the how mean shift works; For example take any distribution of points ($p_1, p_2, p_3, ..., p_n$) which are spread on 2D space. Each of the points have x and y coordinates. Start with any region of interest around that initial point. Calculate the mean of all points inside the region. So move the initial point to new point that will become new initial point repeat the procedure until it converges to maximum density area. The criterion for convergence is mean of all points is same as initial point. Mean shift vector always points towards direction of maximum density. The General form of mean shift is represented as,

$$m_h(y) = \left[\frac{1}{n_h}\sum_{i=0}^{nh} x_i\right] - y_0 \tag{1}$$

Where, n_h is the total number of points inside region of interest; x_i is the points $x_1, x_2, ..., x_{nh}$; y_0 is the initial point. This is equal weighted mean shift calculation (i.e every points have same weight). The Better way to do this by allotting small weights to each point. It can be represented as,

$$m_h(y) = \left[\frac{\sum_{i=1}^{nh} w_t(y_0)x_i}{\sum_{i=1}^{nh} w_t(y_0)}\right] - y_0 \tag{2}$$

Where, n_h is the number of points inside the kernel; y_0 is the initial estimate; w_t is the weights for each point based on the distance from that point to initial point; h is the kernel radius. Weights are determined by using different kernels such as uniform kernel, Gaussian kernel, Epanechnikov kernel. Mean shift is used for finding modes in a set of samples demonstrate underlining probability distribution (pdfs).

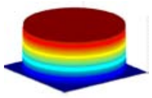
A function with finite number of data points $x_1, x_2, ..., x_i$

$$P(x) = \frac{1}{n}\sum_{i=1}^{n} k(x - x_i) \tag{3}$$

2.1 Uniform Kernel

The uniform kernel function is 1/2, for values between -1 and -1 and zero outside that range. Here $u=(x-x_i)/h$, where h is the window width and x_i are the values of the independent variable in the data, and x is the value of the independent variable for which one seeks an estimate.

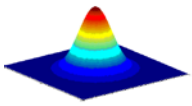
$$K_u(x) = \begin{cases} c \; \| x \| < 1 \\ 0 \; otherwise \end{cases} \tag{4}$$



2.2 Normal Kernel

The Gaussian kernel is this function: $(2PI)^{-.5}exp(-u^2/2)$. Here $u=(x-x_i)/h$, where h is the window width and x_i are the values of the independent variable in the data, and x is the value of the independent variable for which one seeks an estimate. Unlike most kernel functions this one is unbounded on x.

$$K_n(x) = \text{c.exp}\left(-\frac{1}{2} \| x \|^2\right) \tag{5}$$



2.3 Epanechnikov Kernel

The Epanechnikov kernel is this function: $(3/4)(1-u^2)$ for $-1<u<1$ and zero for u outside that range. Here $u=(x-x_i)/h$, where h is the window width and x_i are the values of the independent variable in the data, and x is the value of the scalar independent variable for which one seeks an estimate.

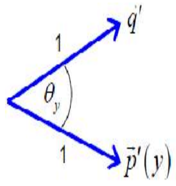
$$K_E(x) = \begin{cases} c\left(1 - \| x_i \|^2\right) \; \| x \| \le 1 \\ 0 \; otherwise \end{cases} \tag{6}$$

3 Bhattcharyya Coefficient

Bhattacharyya coefficient is used for finding the similarities between target in current frame and previous frame. If Bhattacharyya coefficient is maximum then minimizes the dissimilarity. In the proposed method, Bhattacharyya distance measure is applied between target models and candidate in the form of probability distributions. It is represented in the form of metric,

$$\rho(y) = \rho[p(y), q] = \int \sqrt{p_z(y) q_z dz} \tag{7}$$

The feature z representing the color or intensity of the target model is assumed to have a density function q_z, while the target candidate centered at location y has the feature distributed according to $p_z(y)$. The problem is then to and the discrete location y whose associated density $p_z(y)$ is the most similar to the target density q_z. as mean shift is a vector so the $p_z(y)$ and q_z will be in the form of vectors. Bhattacharyya coefficient is measured between two vectors. ρ is the dot product of two vectors or angle between two vectors i.e. $\left(\sqrt{p_1}, \ldots, \sqrt{p_m}\right)^T$ and $\left(\sqrt{q_1}, \ldots, \sqrt{q_m}\right)^T$. Larger the value of ρ means good match is found. In order to find the new target location, try to maximize the Bhattacharyya Coefficient.



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

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

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

4. Proposed Method

Mean shift based tracking has four major step 1) object detection in the first frame, 2) tracking of the object in subsequent frames, 3) extraction of features such as color or grey level intensity or texture or edges represent them in the form of probability density function and 4) formulating decision based upon similarity measure obtained using Bhattacharyya coefficient [2][5]. This Mean shift algorithm iteratively finds the exact match of moving object in sequence of frames. The Mean-Shift algorithm tracks by maximizing Bhattacharyya between two probability density functions (pdfs) [5] represented by a reference and candidate histograms. Since the histogram distance (or, equivalently, similarity) does not depend on spatial structure of the search window, the method is suitable for deformable and articulated objects**.** Histogram does not mean its reserved for color property, it also includes texture or shape (edges). Registration has two important requirements time and accuracy [1]. The experiments are carried out to calculates the time requirement and three accuracy measures (pixel difference, pixel cross correlation, discrete similarity measure). The mean shift vector is updated iteratively towards the convergence where the Bhattacharyya coefficient is maximized. Fig 2 shows flow diagram of mean shift tracking process.

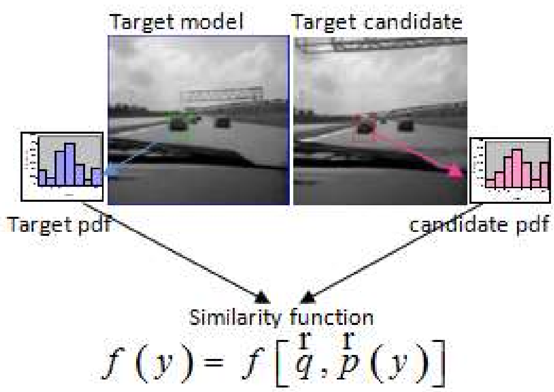


Figure 1: Pdf representation of target and candidate

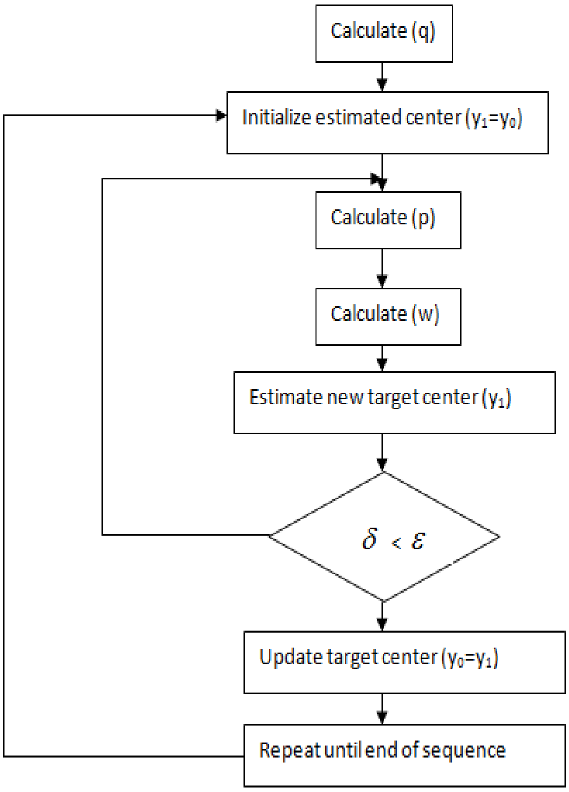


Figure 2: Flow Diagram

4.1Target Model

As mean shift suffers from fixed size search window (i.e. fixed object scale). The target is represented by a rectangular region in the frame. In our approach, experiments with different size of searching window which overcome the fixed size search window problem.

Let $\{x_i^*\}_{i=1....n}$ be the pixel locations of the target model, centered at 0. Pixel at location (x_i^*} the index $b(x_i^*)$ of the histogram bin corresponding to the color of that pixel. The probability of the color u in the target model is derived by employing a convex and monotonic decreasing kernel profile k which assigns a smaller weight to the locations that are farther from the center of the target. The weighting increases the robustness of the estimation, since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or background. The radius of the kernel profile is taken equal to one. It is represented as,

$$\hat{q}_u = c\sum_{i=1}^{n} k\left(\| x_i^* \|^2\right)\delta\left[b\left(x_i^*\right) - u\right] \tag{8}$$

Where δ is the Kronecker delta function. The normalization constant C is derived by imposing the condition $\sum_{u=1}^{n} \hat{q}_u = 1$, from where,

$$c = \frac{1}{\sum_{i=1}^{n} k\left(\| x_i^* \|^2\right)} \tag{9}$$

Since the summation of delta functions for u = 1…. m is equal to one.

4.2 Target candidate

Let $\{x_i^*\}_{i=1....nh}$ be the pixel locations of the target candidate, centered at y in the current frame. Using the same kernel profile k, but with radius h, the probability of the color u in the target candidate is given by,

$$\hat{p}_u(y) = C_h\left(\sum_{i=1}^{n} k\left(\left\| \frac{y - x_i}{h} \right\|^2\right)\right)\delta\left[b\left(x_i^*\right) - u\right] \tag{10}$$

Where, C_h is the normalization constant. The radius of the kernel profile determines the number of pixels (i.e.,the scale) of the target candidate. By imposing the condition that $\sum_{i=1}^{n} \hat{p}_u = 1$ we obtain

$$C_h = \frac{1}{\left(\sum_{i=1}^{n} k\left(\left\| \frac{y - x_i}{h} \right\|^2\right)\right)} \tag{11}$$

Note that C_h does not depend on y, since the pixel locations x_i are organized in a regular lattice, y being one of the lattice nodes. Therefore, C_h can be recalculated for a given kernel and different values of h.

Weights are calculated by this formula,

$$w_i = \sum_{u=1}^{m} \delta\left(b(x_i) - u\right)\sqrt{\frac{\hat{q}_u}{\hat{p}_u(y)}} \tag{12}$$

4.3 Algorithm

Tracking of object or human in the input video is carried out by using mean shift algorithm.

Input: Video V, contains N frames
Output: Object detection and tracking
Begin

Read Video V;

Step 1 For loop of each video frame k = 1 to N

1.1 Read video frame Vk and Vk+1
1.2 Pick a target area (centre) y_0

Step2 Compute a Normalized weighted 2-D Histogram distribution for area to get the target by using equation (8).

Step3 Loop

3.1 Get next frame.
3.2 Initialize the location of the target in the current frame with y_0
3.3 From the current centre compute the candidate target normalized weighted 2-D Histogram by using equation (10)
3.4 Compute Bhattacharya distance between target and candidate by using equation (7)
3.5 Derive the weights $\{w_i\}_{i=1......nh}$ by using equation (12)
3.6 Based on the mean shift vector, derive the new location of the target,

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\| \frac{\hat{y}_0 - \hat{x}_i}{h} \right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\| \frac{\hat{y}_0 - \hat{x}_i}{h} \right\|^2\right)} \quad (13)$$

Update $\{p_u(\hat{y}_1)\}_{u=1\ldots m,}$ and evaluate

$$\delta[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y_1)\hat{q}_u}$$

3.7 While

$$\delta[\hat{p}(\hat{y}_1), \hat{q}] < \delta[\hat{p}(\hat{y}_0), \hat{q}]$$

$$\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1), \text{ do}$$

3.8 If

$\| \hat{y}_1 - \hat{y}_0 \| < \varepsilon$ stop,

Otherwise set $\hat{y}_0 \leftarrow \hat{y}_1$ and go to step1.

End loop

Repeat this till end of frames.

End loop

End of algorithm

5. Accuracy Measure

In order to find the good match between target and candidate propose approach uses Bhattacharyya coefficient. Accuracy of match found can be achieved by three different accuracy measure algorithms [17].

a. Discrete Approximate Match Method of
b. Pixel difference accuracy match
c. Pixel cross correlation accuracy match Accuracy

5.1 Discrete Approximate Match Method of Accuracy

INPUT: Two matched images
OUTPUT: The accuracy of match.

1) Let 'A' be the reference sub-image and 'B' be the matched image.
2) For each position (i,j) in A
3) Find whether there exists a similar point in an area of 3 X 3 around the corresponding position (i,j) in B.
4) Find the total number of match positions.
5) Find percentage of accuracy by, dividing the value of total match by total number of pixels.

$$Acc = \frac{total\ match}{N} * 100$$

Where N is the number of pixels.

5.2 Pixel Difference Method of Accuracy

INPUT : Two matched images
OUTPUT: The accuracy of match.
STEPS –

1) Let 'A' be the reference sub-image and 'B' be the matched image.
2) For each position (i,j) in A
3) Find the difference in pixel value between the reference image and the matched image

$$abs(A_{(i,j)} - B_{(i,j)})$$

4) Find the sum of pixel difference in both images.

$$Sum = \sum_{a,b} abs\,(A - B)$$

5) Find percentage of accuracy, by subtracting the sum of difference by the total number of possible values.

$$Acc = \frac{(255N - sum)}{255\ N} * 100$$

Where N is the number of pixels.

5.3 Pixel Cross Correlation Method of Accuracy

INPUT : Two matched images
OUTPUT: The accuracy of match.
STEPS –

1) Let 'I1' be the reference sub-image and 'I2' be the matched image.
2) For each position (i,j) in I1 and I2
3) Find the sum of squared difference between the pixel values in reference and matched image.

$$\sum (I1 - I2)^2$$

4) Find the percentage of difference in pixel value by,

$$PCC = \frac{\sum (I1 - I2)^2}{\sqrt{\sum I1} * \sqrt{\sum I2}}$$

5) The accuracy of registration is found by subtracting the percentage difference by 100

Acc = 100 – PCC

6. Results And Analysis

The proposed approach is implemented on Windows 7 in Microsoft Visual Studio platform using VC++ language for programming. In our experiments we consider the real time videos taken with different human at various locations and car moving video.

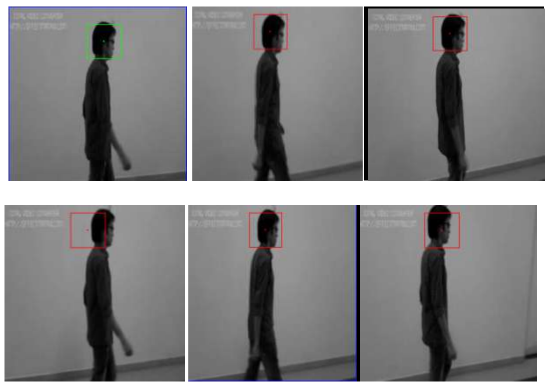


Figure 3: Output of mean shift tracking by using Epanechnikov kernel; searching window size 50X50

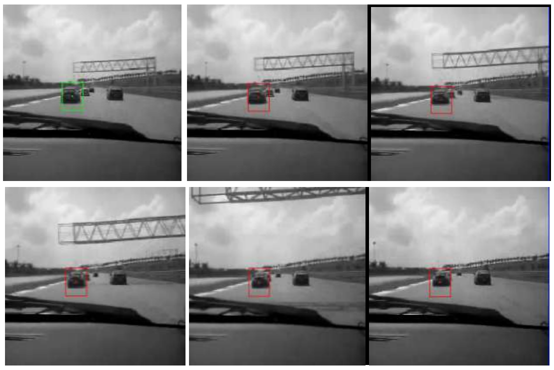


Figure 4: Output of mean shift tracking by using Gaussian kernel; searching window size 40X30

Figure 5: Output of mean shift by using normal kernel; searching window size 30X30

Figure 6: Output of mean shift on color feature by using Epanechnikov kernel

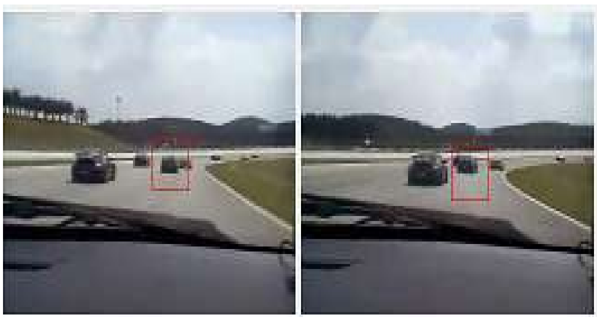


Figure 7: Output of mean shift on gaussian kernel

Figure 8: Output of mean shift on uniform kernel

	Time	Accuracy			TH	TW
		PDM	PCCM	DAMM		
ball	1.025000	98.850105	99.597626	100.00000	30	30
	5.835000	100.00000	100.00000	100.00000	50	40
	4.776000	98.75801	98.564000	99.809000	60	30
car	2.621000	94.84444	96.796211	98.241648	30	30
	6.259000	95.263618	93.307196	92.156987	40	60
	3.858000	96.868629	97.623466	91.369545	50	30
human	4.969000	98.054031	99.356872	89.648953	50	50
	3.029000	98.517647	98.883415	98.064586	30	30
	4.961000	93.987466	98.346657	99.675000	60	30

Table 1: shows the measures of Epanechnikov kernel

	Time	Accuracy			TH	TW
		PDM	PCCM	DAMM		
ball	8.705000	98.850105	99.597626	97.394678	30	30
	2.385000	98.97092	99.99079	98.23878	50	50
	5.531000	98.75801	97.564000	99.809000	60	30
car	7.986000	95.84444	97.796211	98.241648	20	30
	9.04000	94.263618	94.307196	97.156987	30	30
	1.985800	95.868629	97.623466	99.369545	50	40
human	9.676900	98.054031	99.356872	89.648953	50	50
	5.290000	98.517647	98.883415	98.064586	30	30
	6.553000	93.987466	98.346657	99.675000	40	30

Table 2: shows the measures of Gaussian kernel

	Time	Accuracy			TH	TW
		PDM	PCCM	DAMM		
ball	2.605000	98.850105	99.597626	99.774500	30	30
	3.330000	98.579000	99.810600	99.123400	40	40
	4.120000	98.748501	99.840500	99.809000	50	40
car	2.019000	94.458802	94.546782	94.241648	30	30
	6.259000	95.263618	93.307196	92.156987	40	60
	3.858000	96.868629	97.623466	91.369545	50	30
human	2.371000	98.054031	98.356872	96.542310	30	30
	4.242000	98.517647	97.883415	97.064586	50	30
	5.496000	98.885466	98.567857	98.358760	60	40

Table 3: Shows the measures of Uniform Kernel

It is observed from table 1 that the Epanechnikov kernel takes less time to track. From table 2 we observe that Gaussian kernel takes more time to track when compared to

Uniform and Epanechnikov kernel. From table 3 it is observed that it takes less time to track when compared with Gaussian kernel. It is observed that from all the tables Epanechnikov kernel gives maximum accuracy when compared to other kernels. Finally from all the tables we can observe that the Epanechnikov kernel will give the best results.

7 Conclusion and Future Work

The proposed approach efficiently tracks the object or human. Mean shift algorithm for tracking works fine with different size of searching window. This method includes color and grey level intensity values. Proposed algorithm has three different kernels which show satisfactory results. Epanechnikov kernel will gives best results. The edge feature is used get the more appropriate result to the proposed method. There are many types of kernels like triangular, tricube, logistic can be used for distribution.

8 Acknowledgements

The authors acknowledge Prof N R Shetty, Director, Nitte Meenakshi Institute of Technology and Dr H C Nagaraj, Principal, Nitte Meenakshi Institute of Technology for providing the support and infrastructure to carry out the research.

References

[1] "Image registration and target tracking" Jharna Majumdhar,Y Dilip ADE
[2] "Real-Time Tracking of Non-Rigid Objects using Mean Shift", Dorin Comaniciu, Visvanathan Ramesh, PeterMee, IEEE CONFERENCE, 2000
[3] "Mean shift: Robust approach towards feature `space analysis", Dorin comaniciu, Peter meer, IEE TRANSACTION Patren matching and machine intelligence, may 2002
[4] "Image registration methods: a survey", Barbara Zitova, Jan Flusser, Image and Vision Computing 21 (2003)
[5] "Kernel-Based Object Tracking" Dorin Comaniciu, Senior Member, IEEE, Visvanathan Ramesh, Member, IEEE, and Peter Meer, IEEE transactions on pattern analysis and machine intelligence, vol. 25, no. 5, may 2003
[6] "Hybrid particle filter and mean shift tracker with adaptive transition model Emilio Maggio and Andrea Cavallaro, IEEE international conference, 2005
[7] "Efficient Mean-Shift Tracking via a New Similarity Measure", Changjiang Yang, Ramani Duraiswami and Larry Davis, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2005
[8] "Structured Combination of Particle Filter and Kernel Mean Shift Tracking A. Naeem, S. Mills, and T. Pridmore, IEEE conference, 2006
[9] "Real-time hand tracking using a mean shift embedded particle filter", Caifeng Shana, Tieniu Tan,YuchengWei, Pattern Recognition 40 (2007).
[10] "Managing Particle Spread via Hybrid Particle Filter/Kernel Mean Shift Tracking" Asad aeem, Tony Pridmore and Steven Mills, British Machine Vision Association, 2007
[11] "Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection', Alper Yilmaz, IEEE 2007
[12] "Adaptive Mean-Shift Tracking with Auxiliary Particles", Junqiu Wang, Yasushi Yagi, IEEE TRANSACTION 2009
[13] "Kernel-Based Object Tracking via Particle Filter and Mean Shift Algorithm" Y.S. Chia W.Y. Kow W.L. Khong A. Kiring K.T.K. Teo , 11th International Conference on Hybrid Intelligent Systems (HIS), 2011
[14] "Robust Scale-adaptive Mean-Shift for Tracking", Tomas Vojir1, Jana Noskova2, and Jiri Matas1, SCIA 2012
[15] "Constructing re-substitution entropy estimator with discontinuous kernels", Liyuan Xu ; Su-Juan Wang ; Yu-Lin He, IEEE July 2013
[16] "Performance Evaluation of Various Functions for Kernel Density Estimation", Youngsung Soh, Yongsuk Hae, Aamer Mehmood, Raja Hadi Ashraf, and Intaek Kim, OJAppS March 2013
[17] "Human Tracking using Particle Filter", Jharna Majumdar, Kiran S, IJCA 2013.