International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 2, February-2014, pp: (340-347), Impact Factor: 1.252, Available online at: www.erpublications.com

Mean shift based object tracking with accurate centroid estimation and adaptive Kernel bandwidth

ShilpaWakode[1], Dr. Krishna Warhade[2], Dr. Vijay Wadhai[3], Dr. Nitin Choudhari[4]
[1234]Electronics department
[1]R.T.M. Nagpur University, India, [23]MITCOE, Pune, India, [4] SBCCOE, Nagpur, India

Abstract: The object tracking algorithms based on mean shift are good and efficient. But they have limitations like inaccuracy of target localization and sometimes complete tracking failure. These difficulties arises because of the fact that in basic kernel based mean shift tracking algorithm, the centroid is not always at the centre of the target and the size of tracking window remains constant even if there is a major change in the size of object. It causes introduction of large number of background pixels in the object model which give localization errors or complete tracking failure. To deal with these challenges a new robust tracking algorithm based on edge based centroid calculation and automatic kernel bandwidth selection is proposed in this paper. This approach includes relocation of the track window on the middle of the target object in every frame and automatic size adjustment of tracking window so that minimum background pixels will be introduced in object model. The proposed algorithm show good results for almost all the tracking challenges faced by basic mean shift kernel tracking method.

Keywords: Mean Shift (MS), kernel based tracking.

Introduction

In 1975, Fukunaga and Hostetler introduced mean shift (MS) algorithm as a clustering method, which shifts each data to the local maximum of density function. Then in 1995 Cheng [1] represented mean shift in a more generalized way. In 1999 Comaniciu [2] studied the application mean shift. Comaniciu then in [3][4] used the mean shift logic to develop kernel based object tracking algorithm. He used Bhattacharyya coefficient to determine similarity measure between object models and object candidate. Many researchers found his method good for object tracking and later many advance tracking algorithms were developed based on MS logic. However, the target model is not updated during the whole tracking period, which leads to poor localization when the object changes its scale or appearance. Comaniciu in [5] proposed mean shift based object tracking using the automatic update of bandwidth. However MS algorithm has many flaws like the tracking errors or object lost. Introduction of background pixels in object model causes localization error of object tracking. Little background pixels in object model, give good result of object tracking. But in order to let the object contained in object model, some background pixels will certainly get introduced in object model. To minimize the localization errors of object tracking, the background pixels in object model should be made minimum. This problem of object model containing background pixels is studied and discussed in [6]. There in object model, background feature and object feature are integrated using a weight parameter to reduce the localization error of object tracking. The weighted histogram accurately distinguishes between object and background reducing the localization error. Still, there are other aspects which influence localization, such as Taylor approximate expansion formula which is applied in mean shift, and color histogram [7]. Using only histogram for feature description is not sufficient. It may cause false convergence during tracking particularly when similar color modes exist in the target neighborhood [8]. The initialization point for mean shift tracker plays an important role for its convergence to true local maxima. Collins [9] used a "center surround" approach to sample pixels from object and background. His methodwas good for overcoming localization errors. But in case of occlusion, due to incomplete object in object model it gives large tracking errors.In [10] and [11], all the foreground objects are localized by background subtraction and assigned a track index. The occurrences of occlusions are inferred simply by monitoring whether two tracks merge. This method is reliable but it works only with a fixed camera and a known background.Peng in [12] proposed an updating method of object model in mean shift algorithm.

In mean shift based kernel tracking methods the centroid of the target is not necessarily located at the center in all the frames. In every frame tracking window is placed on centroid location. Sometimes the centroid is located near the edges. In that case background pixels will get introduced in object model resulting localization error of object tracking. If number of background pixels is greater than target pixels in an object model then chances of object getting lost in between are very high. To reduce the localization errors of object tracking, it is required to omit the background pixels from object mode. But in order to let the object contained in object model, some background pixels will certainly get introduced in object model. Little background pixels in object model, give good result of object tracking but large number of background pixels give tracking errors or complete tracking failure. Improper target localization and target lost are the two major difficulties observed in basic mean shift kernel tracking method. They can be respectively resolved by relocation of the track window on the middle of the target object and automatic size adjustment of tracking window so that minimum background pixels will be introduced in object model. To achieve this, a new tracking algorithm based on edge based centroid calculation and automatic kernel bandwidth selection is proposed in this paper.

This paper is organized as follows: In Section II the basic kernel based mean shift tracking algorithm is presented. In Section III the proposed tracking algorithm based on edge based centroid calculation and automatic kernel bandwidth selection is discussed. Section IV and V comprises experimental results and concluding remarks respectively.

Basic Mean Shift Algorithm

Kernel based MS algorithm [4] is broadly classified into two components viz. target model representation and candidate model representation.

A. Target Model

The target model is represented in its feature space by its probability density function (PDF), which is calculated using kernel density estimation given by

$$f(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) \qquad (1)$$

Where h is the bandwidth of the kernel and x_i is the center of the d dimensional kernel while n is total number of points in the kernel. Kernel density can be determined with the application of Epanechnikov kernel [3] which is defined as

$$K_E = \begin{cases} \frac{1}{2} C_d^{-1}(d+2)\left(1-|x|^2\right) if\ |x| \le 1 \\ 0 \qquad \text{otherwise} \end{cases} \qquad (2)$$

Where C_d is the volume of the d-dimensional space. The target is selected manually in the first frame and its PDF is calculated by considering its location centered at x_0. To track target in the next frame its PDF in the next frame is calculated at the same location as

$$\hat{q}_u = C\sum_{i=1}^{n} K\left(\left|\frac{x_0 - x_i}{h}\right|^2\right)\delta[b(x_i) - u] \qquad (3)$$

B. Candidate Model

The candidate is the area containing the moving object in the subsequent frames. Candidate model can be described as the probability density distribution of the pixel's feature value in the candidate area centered at y. The PDF of the target candidate is calculated as

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n} K\left[\left|\frac{y-x_i}{h}\right|^2\right]\delta\left[b(x_i) - u\right] \qquad (4)$$

Where
$$C_h = \frac{1}{\sum_{i=1}^{n_h} K\left[\left|\frac{y-x_i}{h}\right|^2\right]}$$

and$u = 1 \ldots m$. Here m is the number of bins used for the calculation of PDF for target representation, h is the bandwidth of the kernel and $x_i$is the center of the d dimensional kernel. While n is total number of points in the kernel and $\delta\left[b\left(x_i\right) - u\right]$is Kroneckor delta function. $b(x_i)$ is image feature value at spatial location $x_i$and C is the normalization constant. Bhattacharya coefficient is used to derive the similarity or correlation between the target model and target candidate. It is specified in the form of a distance given by

$$d = \sqrt{1-\hat{\rho}(y)} \tag{5}$$

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

The term $\hat{\rho}(y)$ is referred as Bhattacharya coefficient. New target location y_1 in current frame is found by iteratively proceeding towards the maxima in the neighborhood. The new target location y_1 is obtained by recursively traveling from its initial location y_0 using following relation, where w_i are the respective weights

$$y = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \tag{6}$$

Mean Shift Algorithm with Accurate Centroid Estimation and Automatic Kernel Bandwidth Update

A. Accurate Centroid Estimation

The basic kernel based MSE rely only on image spectral features which results in poor localization. To avoid that in this approach object structure information is integrated into image histogram to get combine effect of both spectral and gray level features. The accuracy of mean shift depends upon a many things like target surroundings, noise, shape and size modifications etc. Because of this the track window around the target fails to be at the location it is supposed to be. This problem is overcome by the accurate centroid estimation of the target object and relocation of the track window on the middle of the target object [13]. This post processing step makes tracking robust even for convex shaped objects. Following approach is used to relocate the track window on the center of the target object:

1) Canny edge detector should be applied for edge detection of that image.
2) Image should be binarized using proper threshold
3) The centroid of a finite set of points $x_1 + x_2 + \ldots\ldots + x_k$ should be calculated as [13]

$$C = \frac{x_1 + x_2 + \ldots. + x_k}{k} \tag{7}$$

For calculating edge based centroid:

a) First centre column number should be obtained using above formula considering the position of edge on the right and left of the centre point provided by mean shift along horizontal axis.

b) Using the new column number as the centre, the same step for the edges above and below the centre point must be repeated with adjusting the row number.Thus row and column centre obtained using above logic gives new centroid position.

4) The track window must be placed on the new center point calculated using above approach.

B. Automatic Update of the Kernel Bandwidth

In basic kernel based MS algorithm, the size of tracking window remains constant even if there is major change in the size of object. For robust tracking, if the object becomes smaller, the size of window should get smaller accordingly and vice a versa for the object turn out to be bigger. In first case when object becomes smaller, if the size of window does not reduce then many background pixels will get introduced in the window. This will influence the object model distribution. Presence of large number of background pixels in object model give tracking errors or complete tracking failure. In later case when the object becomes bigger, if window remains constant then the candidate object model contains only part of target pixels not pixels of complete target. Then in this case the tracking window can track near the area of the object which in turn will give localization errors [14]. Comaniciu proposed a modified method to update kernel bandwidth [2]. But his method was applicable only if object size becomes smaller. It fails in case of growing objects.

Following approach can be used for adaptive update of kernel bandwidth:

1) Apply Canny edge detector for detecting object edges.
2) Determine the upper left and lower right corner pointcoordinate position of object and set them as (x_0,y_0) , (x_1,y_1) respectively.

3) Obtain diagonal distance D of the object as

$$D = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (8)$$

4) Set diagonal distance of the object of previous and current frame as D_0 and D_1 respectively, and h to be the kernel bandwidth.

5) If $D_1 > D_0$ then update kernel bandwidth[14] as

$$h = (1 + a)\, h \quad (9)$$

If $D_1 < D_0$ then update kernel bandwidth as

$$h = (1 - a)\, h \quad (10)$$

If $D_1 = D_0$ then no need to update kernel bandwidth.

Here coefficient a is the scaling factor, whose value can be generally chosen as 0.1.

C. The Proposed Algorithm Flowchart

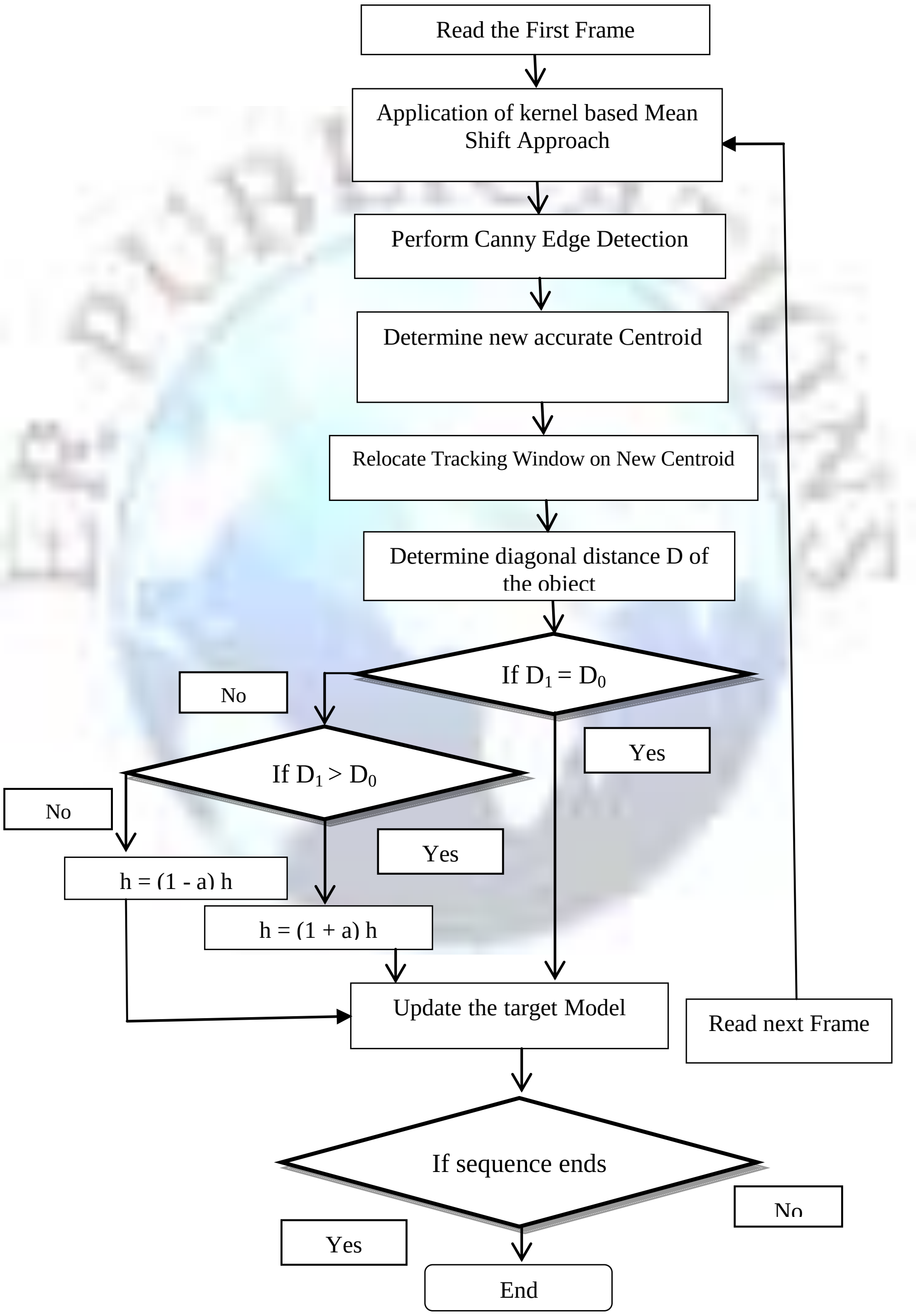

Figure 1: Proposed Algorithm Flowchart

Fig. 1 illustrates the complete flowchart of the proposed algorithm. As per the proposed algorithm, first basic kernel based mean shift algorithm is applied to incoming frames. Then edges are detected using canny edge detector. Using

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 2, February-2014, pp: (340-347), Impact Factor: 1.252, Available online at: www.erpublications.com

accurate centroid estimation discussed above new centroid is calculated which is located at the center of the object. Then tracking window is placed on this new centroid. Then automatic kernel bandwidth update approach is applied if there is any change in the size of object. The process is continued till last frame.

Experimental Results

The experiments are carried out on video clips from standard PETS Videos Database and videos from movies. Programming is done in MATLAB R2010a. System used for programming is with Intel Core 3, 4GB RAM, Window 8.

Frame No.: 5 11 21

Frame No.: 71 91 111

Frame No.: 131 141

Figure 2: Tracking Results using Proposed Algorithm

Frame No.: 5 11 21

Frame No.: 71 91 111

Frame No.: 131 141

Fig. 3: Tracking Results using basic kernel based MS Algorithm

Fig. 2 and 3 gives tracking results of proposed algorithm and basic kernel based MS tracking algorithm respectively.
In case of basic kernel based MS algorithm, tracking failure is observed when complete object is not visible in the scene. In initial frames when the target is partially entered in the scene as well at the exit of target from the scene, the basic kernel based MS algorithm fails to track the object properly rather object is lost in few frames. Here in this case since the centroid of the target is not exactly located at the center and the kernel bandwidth does not adaptively changes with the size of target, basic kernel based MS tracking algorithm faces localization errors. As in proposed algorithm, accurate centroid calculation and automatic update of kernel bandwidth is achieved target is exactly tracked even if it is partially appeared in the scene. Fig. 2 gives tracking results of proposed algorithm where it is visible that the size of searching window changes with the size of target and target is accurately tracked till last frame.

(a) Original Scene (b) Tracked Output

Fig. 4: Multiple target tracking using proposed approach

Experiments are carried out to track multiple targets using proposed approach but the results are not much satisfactory for multiple targets. Fig. 4 illustrates an original scene from a video and its tracked output using proposed approach. In figure (b) tracking window size for each object is in accordance with their respective size.

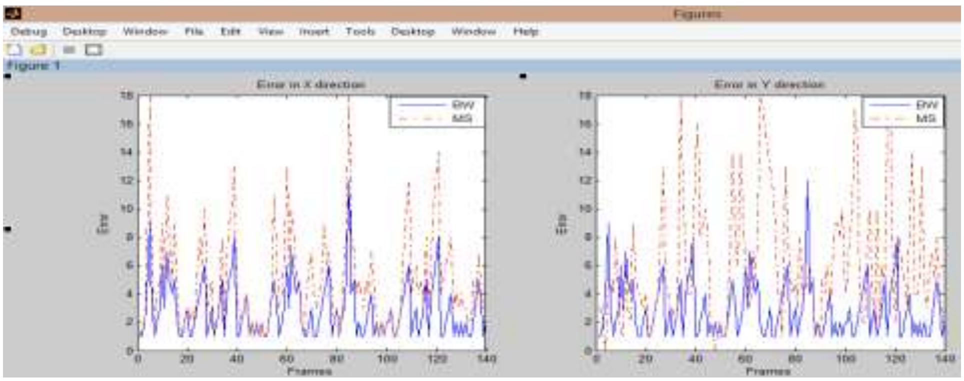

Figure 5: The error of object location in each frame along xdirection and ydirection

The comparison of experimental results between proposed algorithm and basic kernel based MS tracking algorithm is shown in Figure 5. In Figure 5, BW denotes proposed Automatic kernel bandwidth update method, and MS denotes basic Kernel based MS tracking algorithm. In order to show the comparison of results in more details, the following formula is used to compute error of tracking location in i^{th} frame:

$$error_i = \left| T_i - C_i \right| \quad (11)$$

Where Ti is the tracking location in i^{th} frame, C_i is the accurate location of object in i^{th} frame. The average error in tracking the object is calculated using following formula:

$$\text{Average error} = \sum_{i=1}^{N} (error_i/N) \quad (12)$$

Where, Nis total number of frames. Equation (11) and (12) are used to compute the errors of object location in each frameand the average error along *x* and *y* direction. Here $C_i$is decided manually. This method is used here for locating the object and it gives some error for each location. But the average error is less in the proposed automatic kernel bandwidth update method as compared to basic Kernel based MS tracking algorithm. A quantitative evaluation of test results between the two methods is given in Table 1.

Table 1: Comparative Results

Comparison for tracking	Basic MS tracking algorithm	Proposed Edge based centroid estimation with automatic kernel bandwidth update method
Avg. error in *x* direction	4.864	2.85
Avg. error in *y* direction	6.214	3.05
Avg. iterations	27.55	22.49
Avg. Bhattacharya coefficient value	0.947	0.948

As shown in Table 1 the average Bhattacharyya coefficient value vary little in edge based method comparing to original KBOT, but when occlusion occur, the Bhattacharyya coefficient value in edge based method is larger than that in KBOT. The original KBOT was having localization issues while handling the same video with improved centroid estimated KBOT implementation the algorithm was successful in tracking the target with improved localization.

Conclusion/Results

The two major challenges improper target localization and loss of target track faced by basic mean shift kernel tracking method, are addressed in this paper. A new tracking algorithm based on accurate centroid calculation and automatic kernel bandwidth selection is proposed in this paper. This approach includes relocation of the track window on the middle of the target object in every frame and automatic size adjustment of tracking window so that minimum background pixels will be introduced in object model. The proposed algorithm show good results for almost all the tracking challenges faced by basic mean shift kernel tracking method. But its performance is poor if there is no proper distinction between target and background. In future work can be done towards enhancing the algorithm to overcome this defect and extend it for accurate multiple object tracking.

References

[1]. Y. Cheng, "Mean shift, mode seeking, and clustering," IEEE Trans. on pattern analysis and machine intelligence, Vol.17, No.8, 1995, pp.790-799.
[2]. D. Comaniciu, and P. Meer, "Mean shift analysis and applications," In: Proc. of the IEEE Int'l Conf. on Computer Vision, 1999, pp.1197-1203.
[3]. D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," Proc. of IEEE Conf. on Comp. Vision and Pattern Recog., 2000, pp.142-149.
[4]. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, 2003 pp. 564-575.
[5]. D. Comaniciu, "An Algorithm for Data-Driven Bandwidth Selection", in IEEE Trans. on pattern analysis and machine intelligence. Vol.25, No.2, 2003, pp.281-288.
[6]. Z.Wen, Z. Cai, " A Robust Object Tracking Approach using Mean Shift", Third International Conference on Natural Computation (ICNC 2007), Sep 2007.

[7]. A. Lehuger, P. Lechat and P. Perez, "An Adaptive Mixture Color Model for Robust Visual Tracking", in Proc. IEEE Int. Conf. on Image Process, Oct 2006,pp. 573 – 576.
[8]. Xu Dong, Y. Wang and Jinwen, "Applying a New Spatial Color Histogram in Mean-Shift Based Tracking Algorithm", Image and Vision Comp., Univ. of Otago, New Zealand, 2005.
[9]. R. T. Collins, Y Liu and M Leordeanu, "Online Selection of Discriminative Tracking Features", IEEE Trans. on pattern analysis and machine intelligence, Vol. 27, No. 10, 2005, pp. 1631-1643.
[10]. A. Senior, "Appearance Models for Occlusion Handling" , J. Image Vis. Comput., Vol. 24, No. 11, 2006, pp. 1233-1243.
[11]. A. W. Senior, "Tracking with Probabilistic Appearance Models", inECCV Workshop on Perform. Eval. Tracking Surveillance Syst., Jun 2002, pp. 48-55.
[12]. N. Peng, S,Yang J, Liu Z., "Mean Shift Blob Tracking With Kernel Histogram Filtering and Hypothesis Testing", Pattern Recognition Letters, Vol. 26, No.5, 2005, pp. 605-614.
[13]. R. Mehmood, M. Ali, I. Taj "Applying Centroid Based Adjustment to Kernel Based Object Tracking for Improving Localization", IEEE, 2009.
[14]. Le Zhang, D. Zhang, Yixin Su, Fei Long, "Adaptive Kernel Bandwidth Object Tracking Based on Mean Shift Algorithm", Proc. of IEEE Conference on Itelligent Control and Information Processing (ICICIP), 2013, pp. 413-416.