Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-42

Robust Mean Shift Object Tracking With Improved Tracking Velocity and Least Localization Errors

Shilpa Wakode[1], Dr. K K Warhade [2], Dr. V M Wadhai [3] , Dr. N.K. Choudhari [4]

***Abstract*— The object tracking algorithms based on men 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 center 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. One more limitation of basic mean shift tracking algorithm is that it does not have an adaptive stop threshold in searching the target procedure. So even at times it gives proper target localization the computation time is much more. To deal with these challenges a new robust mean shift object tracking algorithm with improved tracking velocity and least localization errors is proposed in this paper. This approach includes relocation of the track window on the middle of the target object in every frame using edge based centroid calculation technique and automatic size adjustment of tracking window so that minimum background pixels will be introduced in object model. Also computational speed is improved by limiting the MS iterations count. 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 object tracking, Bhattacharya coefficient**

I. 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] and in 1999 Comaniciu [2] studied the application MS. Comaniciu then in [3][4] used the MS logic to develop kernel based object tracking algorithm. He used Bhattacharyya coefficient to determine similarity measure between object models and object candidate. Many researchers used his method for developing advance tracking algorithms. 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. It is because of object scale variations and due to the fact that MS centroid is not always located at the centre of the target.

Shilpa Wakode
LTCE, Mumbai Univ.
India

Dr. Krishna Warhade
MITCOE, Pune Univ.
India

Dr. Vijay Wadhai
Sinhagad COE, Pune Univ.
India

Dr. Nitin Choudhari
SBCCOE, Nagpur Univ.
India

Comaniciu in [5] proposed MS based object tracking using the automatic update of bandwidth. The 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. Still, there are other aspects which influence localization, such as Taylor approximate expansion formula which is applied in MS, 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 MS 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 method was 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.. Peng in [12] proposed an updating method of object model in MS algorithm.

In MS based 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. Large number of background pixels gives tracking errors or complete tracking failure. Improper target localization and target lost are the two major difficulties observed in basic MS 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.

One more major limitation of basic MS tracking algorithm is that it does not have an automatic stop threshold in searching the target procedure. So even at times it gives proper target localization the computation time is much more. Because of no limit on MS iterations the program may run into endless cycle. As the basic MS tracking algorithm do not have an automatic updating principle for target model, tracking error occurs in case of camera motion, target partial occlusions, and target scale variations. In such cases the difference between target model and true target gets more and more, which ultimately increases searching time or lose

Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-42

tracking target. So in order to improve the tracking velocity, number of iterations should be reduced. Hence in the proposed algorithm, a reasonable searching stop threshold value is set to limit the MS iterations.

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

II. Basic MS Algorithm [4]

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

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

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 y_0. To track target in the next frame its PDF in the next frame is calculated at the same location as

$$\hat{q}_u=C\sum_{i=1}^{n}K\left(\left|\frac{y_0-x_i}{h}\right|^2\right)\delta[b(x_i)-u] \qquad (3)$$

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

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

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.

$$y_1=\frac{\sum_{i=1}^{n_h} x_i\ w_i}{\sum_{i=1}^{n_h} w_i} \qquad (6)$$

Where $w_i=\sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}}\delta[b(x_i)-u]$

III. Robust MS Object Tracking Algorithm with Improved Tracking Velocity and Least Localization Errors

In this paper a new improved object tracking algorithm is proposed to make the original MS tracking more robust. To achieve this, basic MS algorithm is extended with three major steps:

A. *Accurate Centroid Estimation Using Canny Edge Detector*

The basic kernel based MS 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 MS depends upon a many things like target surroundings, noise, shape and size

Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-42

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.

The application of centroid estimation mode is linked with the result of Bhattacharya distance d between target model and candidate. Let T_a is denoted as an upper threshold and T_b as a lower threshold of metric distance d. Here T_a is the distance d value when the difference between target model and target candidate is *a*. T_b is also the distance d value when the difference is *b*. This similarity measure is divided into three stages.

• When $d \leq T_b$ – the basic MS tracking algorithm is working well and proper localization of object is being achieved. So no post processing is required.

• When $T_b \leq d \leq T_a$ - the basic MS tracking algorithm has poor localization still proper tracking can be achieved by application of canny edge detector based centroid estimation technique.

• When $d \geq T_a$ - the basic MS tracking algorithm fails to track the object completely. It may be due to object being tracked is facing full occlusion. So edge detection or centroid estimation at this stage is not reliable.

Following approach is used to relocate the track window on the center of the target object when $T_b \leq d \leq T_a$:

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]

$$C = \frac{x_1 + x_2 + \ldots. + x_k}{k} \quad (7)$$

For calculating edge based centroid:

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

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

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

Fig.1 indicates the complete procedure and result of Edge based Centroid Estimation. Initially as shown in (a), the MS based centroid was located somewhere near the edges of the object. There many background pixels were included inside tracking window. This is avoided by calculating the centroid using edge based approach with application of Canny edge detector, as shown in (b) and (c). The new centroid is located at the center of the target and the tracking window is relocated on this new centroid. In (d), it is clearly visible that tracking window includes maximum object pixels only.

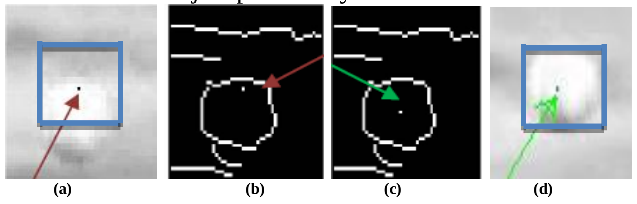

Figure 1. (a) MS based centroid (b) Canny Edge detected Output (c) Adjusted Centroid at Centre of object (d) Final output

B. *Limiting MS Iterations and Updating Target Model*

Let *E is* set as searching stop threshold value. When the distance between the new target candidate and the previous target candidate is smaller than *E*, the iteration for searching newer target candidate stops. The number of iterations can be set as stop threshold value. To avoid endless cycle in real-time object tracking, maximum number of iterations can be set to *N*max. Target model must update in the implementation procedure of a tracking algorithm [14]. But its updating principle must be neither extreme sensitive nor extreme slow so as to avoid wastage of computational time for computing feature representations. Therefore the metric distance d derived from Bhattacharyya coefficient [15] is set as a standard for updating target model. Following approach is used to limit the number of MS iterations and hence increasing the tracking velocity:

1. Given the target model is $\{\hat{q}_u\}_{u=1\ldots m,}$ and its location is $\mathbf{y_0}$ in the previous frame. Initialize the location of the target in the current frame with $\mathbf{y_0}$, compute $\{p_u(y_0)\}_{u=1\ldots m,}$ and evaluate

$$\rho = [p(y_0, q)] = \sum_{u-1}^{m} \sqrt{p_u(y_0) q_u}\,] \quad (8)$$

2. Derive the weights
3. Find the next location of the target candidate y_1
4. Compute $\{p_u(y_1)\}_{u=1..m}$ and evaluate Bhattacharya distance between $p_u(y_1)$ and q
5. If $|y_1 - y_0| < E$ then stop and go to step 7
6. If The maximum iteration number (in the current frame) $\geq$ *N*max, Then go to Step 7, Otherwise set $\mathbf{y_0} = \mathbf{y_1}$ and go to step 3
7. Update the target model by $\{\hat{q}_u\}_{u=1..m} = \{p_u(y_1)\}_{u=1..m}$
8. If no new images then stop otherwise read new image and go to step 1.

Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-42

C. *Adaptive Update of 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. If the size of window does not reduce when the object becomes smaller, then many background pixels will get introduced in the window. Presence of large number of background pixels in object model give tracking errors or complete tracking failure. Also if window size remains constant when the object becomes bigger, 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 [16]. Comaniciu proposed a modified method to update kernel bandwidth [17]. 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 point coordinate position of object and set them as (x_{ul},y_{ul}), (x_{lr},y_{lr}) respectively.
3. Obtain diagonal distance D of the object as

$$D = \sqrt{(x_{lr} - x_{ul})^2 + (y_{lr} - y_{ul})^2} \qquad (9)$$

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[16] as

$$h = (1 + a)\,h \qquad (10)$$

If $D_1 < D_0$ then update kernel bandwidth as

$$h = (1 - a)\,h \qquad (11)$$

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.

D. *Implementation of the Proposed Algorithm*

The implementation flow of the proposed tracking algorithm is as given below:

1. Read the desired number of frames from the video. Convert each frame to gray scale and resize to 256×256 pixel.
2. Read the first frame and select the target manually with location centered at y_0 which will be the center of ellipse also referred as initial point of MS.
3. Depending upon size of object select the bandwidth matrix h = [hx, hy], where hx, hy is the width and height of object respectively. They also denote scale of ellipse in row and column direction respectively.
4. Set the total number of bins for calculating PDF of target to 255. Also set T_a and T_b to be upper and lower threshold of metric distance d.
5. Select Epanechnikov kernel given in (2)
6. To track target in the next frame its PDF $[\hat{q}_u]_{u=1,\ldots,m}$ in the next frame is calculated at the same location using (3).
7. Obtain the PDF $\hat{p}_u\,(y_0)$ of target candidate using (4).
8. To measure the degree of similarity between target and candidate calculate the Bhattacharya coefficient at y_0 and Bhattacharya distance d for same point using (5).

$$\hat{\rho}(y_0) = \hat{\rho}[\hat{p}(y_0), \hat{q}] = \sum_{u=1}^{m} \sqrt{(\hat{p}_u(y_0)\hat{q}_u\,)}$$

9. From the PDF of target $\hat{q}_u$ and candidate $\hat{p}_u\,(y_0)$ calculate the respective weights $[w_i]_{u=1,..m}$ for each point in the kernel.
10. Obtain new target location y_1 using (6) and then again obtain PDF of target candidate at y_1 i.e. $\hat{p}_u\,(y_1)$.
11. Evaluate the Bhattacharya coefficient at y_1 i.e. $\hat{\rho}(y_1)$ using $\hat{p}_u\,(y_1)$ and hence Bhattacharya distance d at y_1 .
12. If $|y_1 - y_0| < E$ then stop and go to step 14
13. If The maximum iteration number (in the current frame) ≥ Nmax, Then set $\mathbf{y_1} = \mathbf{y_1}$and go to step 14, Otherwise set $\mathbf{y_0} = \mathbf{y_1}$ and go to step 10.
14. If $T_b \le$ d $\le T_a$, place the target window at new centroid and go to step 16.
15. Perform Canny Edge detection and determine the exact centroid of the object i.e. new y_1. Relocate the tracking window at new centroid and go to step 11.
16. Calculate diagonal distance D using (9). Set diagonal distance of the object of previous and current frame as D_0 and D_1 respectively
17. If $D_1 > D_0$ then update h = (1 + a) h
 If $D_1 < D_0$ then update h = (1 - a) h
 If $D_1 = D_0$ then no need to update h.
18. Update the target model by $\{\hat{q}_u\}_{u=1..m} = \{p_u\,(y_1)\}_{u=1..m}$
19. If no new images then stop otherwise read new image and go to step 1. Read the next frame so that current frame = next frame and repeat the procedure to track the object in subsequent frames.

IV. 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. Fig. 2 and 3 gives tracking results of basic kernel based MS tracking algorithm and proposed algorithm respectively.

Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-42

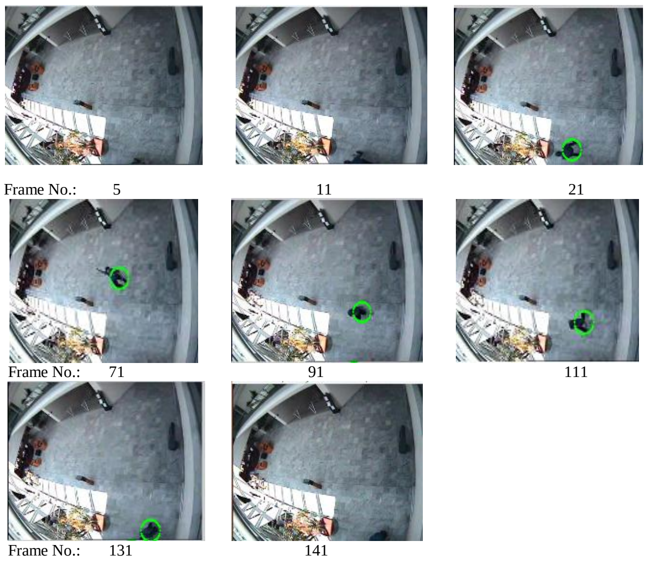

Figure 2. Tracking Results using basic kernel based MS algorithm

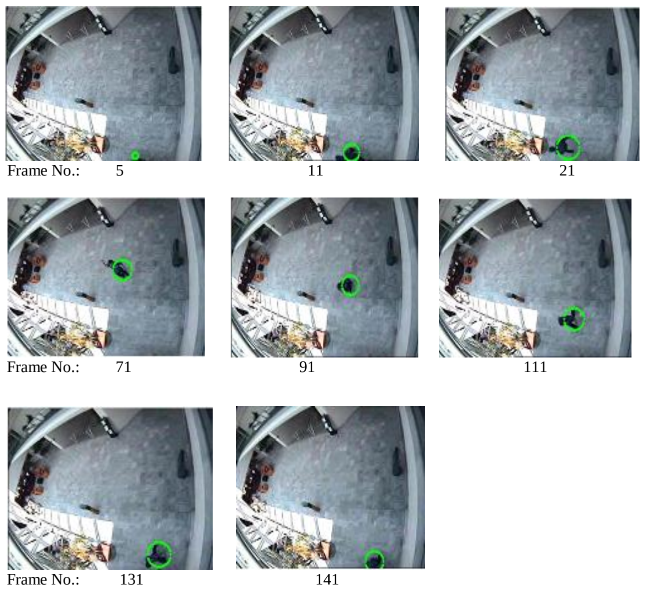

Figure 3. Tracking Results using Proposed Algorithm

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. 3 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.

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 Fig. 4(b) tracking window size for each object is in accordance with their respective size.

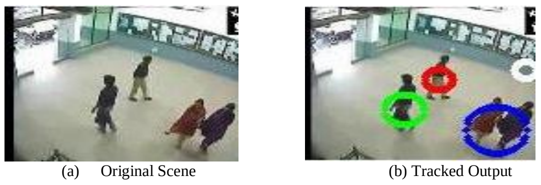

Figure 4. Multiple target tracking using proposed approach

The comparison of experimental results between proposed algorithm and basic kernel based MS tracking algorithm is shown in Fig. 5. In Fig. 5, BW denotes proposed Automatic kernel bandwidth update method, and MS denotes basic MS tracking algorithm.

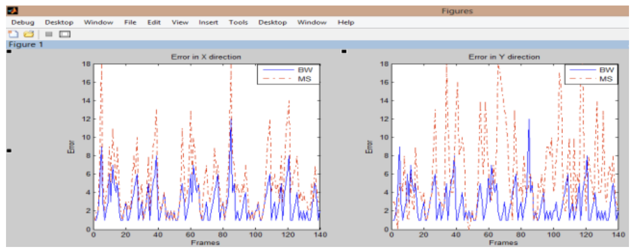

Figure 5. The error of object location in each frame along x direction and y direction

In order to show the comparison of results in more details, the following formula is used to compute error of tracking location in ith frame:

$$\text{error}_i = \left| T_i - C_i \right| \qquad (12)$$

Where Ti is the tracking location in ith frame, C_i is the accurate location of object in ith frame. Equation (12) and (13) are used to compute the errors of object location in each frame and the average error along *x* and *y* direction.

The average error in tracking the object is calculated using following formula:

$$\text{Average error} = \sum_{i=1}^{N} (\text{error}_i / N) \qquad (13)$$

Where, N is total number of frames. 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. A quantitative evaluation of two methods based on test results between the two methods is given in Table I.

From Table I it is observed that, the average Bhattacharyya coefficient value vary little in proposed method comparing to basic MS, but when occlusion occurs, the Bhattacharyya coefficient value in proposed method is larger than that in basic MS. Tracking error is very less in proposed method as compared to the basic MS, this ensures improved target localization. Also number of iterations is reduced in proposed method which improves the tracking velocity.

Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-42

Table I
Comparative Results

Comparison for tracking	**Basic MS tracking**	**Proposed 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

V. Conclusion

The three major challenges improper target localization, loss of target track and endless MS iterations faced by basic MS kernel tracking method, are addressed in this paper. A new tracking algorithm based on accurate centroid calculation, automatic kernel bandwidth selection and limiting MS iterations 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. Also to improve tracking velocity a stop threshold is introduced for limiting MS iterations. The proposed algorithm show good results for almost all the tracking challenges faced by basic MS 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, “MS, 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, “MS analysis and applications,” *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 MS,” *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 MS”, *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*”, ECCV Workshop on Perform. Eval. Tracking Surveillance Syst.*, Jun 2002, pp. 48-55.

[12]. N. Peng, S,Yang J, Liu Z., “MS 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]. M. Weng, H. Mingyi, and Y. Zhang, “An Adaptive Implementation of the Kernel-Based Object Tracking Method”, IEEE International Conference on Innovative Computing, Information and Control (ICICIC'06), 2006.

[15]. F. Aherne, N. Thacker, and P. Rockett, “The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data,” Kybernetika, 1998, vol. 34, no. 4, pp. 363-368.

[16]. Le Zhang, D. Zhang, Yixin Su, Fei Long, “Adaptive Kernel Bandwidth Object Tracking MS Algorithm”, Proc. of IEEE Conference on Itelligent Control and Information Processing (ICICIP),2013, pp. 413-416.

[17]. D. .Comaniciu, P. Meer, “MS analysis and applications”, in: Proc. of the IEEE Int’l Conf. on Computer Vision, 1999, pp.1197-1203.

About Author (s):

	Shillpa M. Wakode Pursuing Ph.D. in R.T.M.N.U. Designation: Assistant Professor, L.T.C.E., Navi Mumbai Research Area: Object tracking for video surveillance applications
	Dr. Krishna K. Warhade Ph.D. , IIT Bombay. Designation: Professor, M.I.T.C.O.E., Pune Research Area: Video Segmentation, Processing and Tracking
	Dr. Vijay M. Wadhai Ph.D.,Amravati University Designation: Principal and Professor, Sinhagad COE., Pune Research Area:
	Dr. Nitin K. Choudhari Ph.D.Jamia,New Delhi Designation: Principal and Professor, S.B.C.C.O.E., Nagpur Research Area: Signal Processing