IJICTR International Journal of Information & Communication Technology Research

ITRC

Volume 4- Number 1- December 2011 (29-39)

Efficient Object Tracking Using Optimized K-means Segmentation and Radial Basis Function Neural Networks

Alireza Asvadi
Faculty of Electrical & Computer Engineering
Babol University of Technology
Babol, Iran
asvadi.a@stu.nit.ac.ir

MohammadReza Karami
Faculty of Electrical & Computer Engineering
Babol University of Technology
Babol, Iran
mkarami@nit.ac.ir

Yasser Baleghi
Faculty of Electrical & Computer Engineering
Babol University of Technology
Babol, Iran
y.baleghi@nit.ac.ir

Received: October 5, 2011- Accepted: December 31, 2011

***Abstract*—In this paper, an improved method for object tracking is proposed using Radial Basis Function Neural Networks. Optimized k-means color segmentation is employed for detecting an object in first frame. Next the pixel-based color features (R, G, B) from object is used for representing object color and color features from surrounding background is extracted and extended to develop an extended background model. The object and extended background color features are used to train Radial Basis Function Neural Network. The trained RBFNN is employed to detect object in subsequent frames while mean-shift procedure is used to track object location. The performance of the proposed tracker is tested with many video sequences. The proposed tracker is illustrated to be able to track object and successfully resolve the problems caused by the camera movement, rotation, shape deformation and 3D transformation of the target object. The proposed tracker is suitable for real-time object tracking due to its low computational complexity.**

Keywords-component; computer vision; object tracking; k-means segmentation; radial basis function neural networks; mean shift.

I. INTRODUCTION

Tracking is basic task for several applications of computer vision, e.g., video monitoring systems, video indexing, traffic monitoring, automated surveillance, and so on. The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. Object tracker may also provide the complete region in the image that is occupied by the object at every time instant. Some of important challenges encountered in visual tracking are non-rigid objects, complex object shapes, occlusion, scale/appearance change of the objects and real-time processing. Considerable work has already been done in visual tracking to address the mentioned challenges [1]. There exist many object tracking algorithms in the literature. These algorithms can be divided into three categories [1]: Point Tracking [2, 3], Kernel tracking (e.g. Mean shift [4, 5], KLT tracker [6, 7]), silhouette tracking (e.g. variational methods [8, 9]). These algorithms mainly differ in the way they use image features and model motion, appearance and shape of the object. The first group of tracking methods is point tracking algorithms which use a set of points to represent the object being tracked. After finding the feature points, the point tracking algorithm finds the feature correspondence between frames. In order to do this, the point tracker finds the optimal solution of a cost function minimization problem formed to establish the correspondence. Point trackers are suitable for tracking very small objects which can be represented by a single point [1, 2, 3 and 7]. Kernel tracking is another type of tracking algorithm, which utilize a primitive region from the object to compute the motion of the object frame by frame. The kernel can be a rectangular template or an elliptical shape with an associated histogram and objects are tracked by computing the motion of the kernel in consecutive frames. There are several kernel tracking algorithms in existent and each algorithm is developed to solve one particular problem, such as translation, affine, projective, and estimation. Template matching is the most intuitive approach of kernel tracking because the algorithm finds the region of interest in the current frame and then looks in the next frame until a match is found. The main goal of the kernel trackers is to estimate object motion [1, 4, 5, 6 and 7]. The last group of object tracking methods is silhouette tracking. In some cases, the object is too complex to be represented by a set of points or primitive geometry shapes. Therefore, silhouette tracking is used to find the complete object region in every frame with the help of object models computed from previous frames. Silhouette tracking methods use the information encoded inside the object region. The main difference between distinctive silhouette algorithms is the method to model the objects. The methods to model the objects can be color histograms, edges, or object contours. The most important advantage of tracking silhouettes is their flexibility to handle a large variety of object shapes and find an accurate object region in each frame that can be used in higher level applications like as activity recognition. These methods mostly can be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames [1, 7, 8 and 9]. Many existing algorithms [10-14] segment each video frame to determine the objects; this action can be computationally expensive, and it is not necessary if the goal is to determine the moving objects.

In the last few decades, neural networks have been successfully used in the number of applications such as pattern recognition, remote sensing, dynamic modeling and medicine. The increasing popularity of neural networks in many fields is mainly due to their ability to learn the complex nonlinear mapping between the input-output data and generalize them. As one of the most popular neural network models, radial basis function network attracts lots of attentions on the improvement of its approximation ability as well as the construction of its architecture. Learning-based tracking algorithms were rarely used for general purpose object tracking. This is due to the difficulty in adapting the neural networks for tracking purpose [15]. Bouzenada et al. [16] proposed a global approach for adapting the neural networks for tracking purpose but this method needs to be trained during an offline stage over a set of input and output samples. Ahamed et al. [17] designed an artificial neural network to track a specific airplane, however, the neural network can be trained to track any other object of interest but this is designed to track specific object, which require off-line learning phase. Babu et al. [15] used two neural networks for object tracking, this method use histograms of object region and background region and a threshold for separating object from background which building histograms are computationally expensive and constant threshold is not reliable for every size and kind of object. Asvadi et al. [18] used k-means segmentation and neural network for object tracking but basic k-means is computationally expensive and using neural network for object localization is not reliable alone.

In this paper a fast and straightforward algorithm of object tracking is proposed that uses optimized k-means color segmentation for detecting an object in first frame. Color features from object are used for representing object color and color features from background is extracted and extended. RBF neural network trained by this features to separate object from background in other frames meanwhile mean-shift procedure is used to track object location.

The rest of the paper is organized as follows: Section 2 describes the proposed algorithm that presents optimized k-means segmentation, background extension, radial basis function neural network and Object localizer. The experimental results are shown in Section 3, and the conclusions and the future prospects are given in Section 4.

II. PROPOSED ALGORITHM

Fig. 1 presents the process of the proposed method. Each block is described briefly in this section. They will be discussed in following sections. In object tracking, first one needs to develop a model for the object of interest from the given initial video frame that starts with selecting object approximately in the first frame manually as it shown in Fig. 2. Next, optimized k-means color segmentation is used for separating the object from background this result in a binary image. The features used for k-means segmentation are simple pixel color based features, which correspond to the values in R-G-B color spaces. In next step the background color feature that are inside the box and surrounding the target object is extended. Background extension is used to improve RBF neural network classification performance when background colors change during tracking. Then, both of the object and extended background features are used to train the RBFNN. Next, trained RBF neural network is used for classifying object and non-object pixels in other frames. Then, the object localizer estimates the target location in the subsequent frames using trained RBFNN and mean-shift procedure.

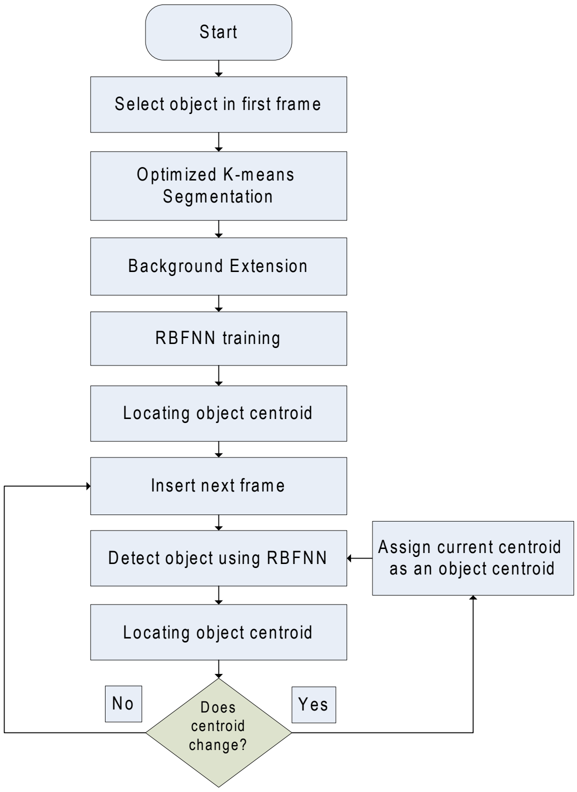

Figure1. system overview

A. *Optimized K-Means Segmentation*

At first frame a box that contains the object is selected. To characterize the target an unsupervised method is needed to separate object from surrounding background. K-means clustering [19] is one of the popular algorithms in clustering and segmentation that is used to separate object from surrounding background. K-means clustering treats each feature point as having a location in space. The basic K-means algorithm then arbitrarily locates, that number of cluster centers in multidimensional measurement space. Each pixel in the image is then assigned to the cluster whose arbitrary mean vector is closest. The procedure continues until there is no significant change in the location of class mean vectors between successive iterations of the algorithms. However, K-means algorithm is very sensitive in initial starting points. K-means generates initial cluster randomly. When random initial starting points close to the final solution, K-means has high possibility to find out the cluster center. Otherwise, it will lead to incorrect clustering results [20].

In this paper, a priori information of object tracking nature exploited to generate initial starting points and make K-means clustering efficient and computationally simple and fast. A priori it is known that the numbers of classes are two and the object is in the center of box and background pixels are around the object and close to border of box. The object color is assumed that has unified color which is most often a valid assumption and it put the object pixels in same cluster.

In this paper, a priori information of object tracking nature exploited to generate initial starting points and make K-means clustering efficient and computationally simple and fast. A priori it is known that the numbers of classes are two and the object is in the center of box and background pixels are around the object and close to border of box. The object color is assumed that has unified color which is most often a valid assumption and it put the object pixels in same cluster.

A brief description of Optimized K-means algorithm that it is used in this paper is as following:

Three color features (R-G-B) are used for representing color space. As it mentioned before the numbers of classes are two. The box pixels are named by $\{x_1, x_2, \ldots, x_n\}$ which $x_i = \{R_i, G_i, B_i\}$.

Step 1: choose 2 initial cluster center z_1, z_2. z_1 is obtained by averaging R-G-B values of four points which are located in 0.05 box width and height distances from the center of box ($z_1 = \frac{1}{4}\sum_{m=1}^{4} x_m$). z_2 is obtained by averaging four points which are in 0.45 box width and height distances from the center of box ($z_2 = \frac{1}{4}\sum_{n=1}^{4} x_n$). The points which are used to obtain z_1 are shown by four red crosses and the points which are used to obtain z_2 are shown by four yellow pluses sign in Fig. 3a.

Step 2: assign point $x_i, i = 1,2,\ldots,n$ to cluster $C_j, j = 1,2$ if $\|x_i - z_j\| < \|x_i - z_p\|, p = 1,2$, and $j \neq p$

Step 3: compute new cluster centers z_i^* by Equation (1):

$$z_i^* = \frac{1}{n_i}\sum_{x_j \in c_i} x_j, \quad j = 1,2 \tag{1}$$

Where n_i is the number of elements belonging to cluster c_i.

Step 4: if $z_i^* = z_i, i = 1,2$ then terminate. Otherwise continue from step 2.

Note that for real time constraints in case the process does not terminate at Step 4 normally, then it is executed for a maximum fixed number of iterations [21]. In the presented work this number is chosen 10. In practice the procedure of choosing directed initial cluster centers increases reliability of separating object from surrounding background and moreover it reduces the numbers of k-means iterations that is important for real time constraints. In practice the average number of iterations is about 3. Fig. 3 shows the manually selected object, initial cluster centers and k-means color segmentation result.

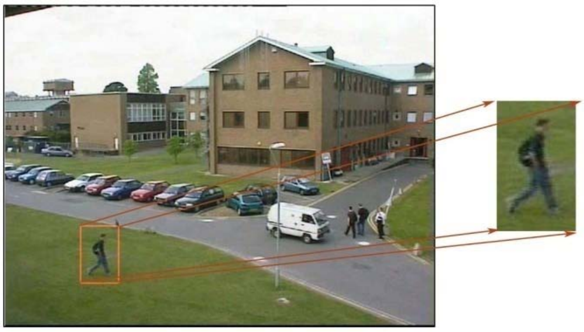

Figure2: selecting the object manually

(a) (b)

Figure3: (a) Manually selected object, green point shows box center, red cross pointes show pixels which their R-G-B color are used for initial object cluster center and yellow plus sign pointes show pixels which their R-G-B color are used for initial background cluster center. (b) Optimized k-means color segmentation result.

B. Background Extension

After segmenting object box in the first frame and detecting an object, color features of object and surrounding background is extracted. The simplest way is to train RBFNN by extracted features of object and surrounding background. Whereas background may be changed in consecutive frames then extracted features of surrounding background from first frame could not be reliable alone. In presented work a new approach is proposed. In addition to extracted features of surrounding background the same numbers of points are distributed randomly in the whole feature space except in object feature points. Both of this surrounding background and distributed points considered as a background features. In presented paper this process is called background extension. Fig. 4 illustrates this process. These points could be distributed randomly or uniformly. Practically both methods had a same result. In this work random distribution of points is used. This extended background features and Object features are used to train RBFNN. The background extension procedure greatly will increase RBFNN classification accuracy in proposed method which will be discussed in next section.

C. Radial Basis Function Neural Network

The purpose of this procedure is to train Radial Basis Function neural network how to segment object. This requires less computational effort over those methods that use segmentation in every frame. R-G-B pixel values of the segmented object and extended background features are used to train the RBF neural network.

A RBFNN is an artificial neural network that uses radial basis functions as activation functions. Fig. 5 shows the structure of the RBFNN. The RBFNN is three layered feed-forward neural network. The first layer is linear and only distributes the input signal, while the next layer is nonlinear and uses Gaussian functions. The third layer linearly combines the Gaussian outputs. Only the tap weights between the hidden layer and the output layer are modified during training. RBFNN have 5 parameters for optimization [22]: 1- The weights between the hidden layer and the output layer. 2- The activation function. 3- The center of activation functions. 4- The distribution of center of activation functions. 5- The number of hidden neurons. The weights between the hidden layer and the output layer are calculated by using Moore-Penrose generalized pseudo-inverse [15, 23]. This algorithm overcomes many issues in traditional gradient algorithms such as stopping criterion, learning rate, number of epochs and local minima. Due to its shorter training time and generalization ability, it is suitable for real-time applications. The radial basis function selected is usually a Gaussian kernel for pattern recognition application [22]. Generally the center and distribution of activation functions should have characteristic similar to data. There are different strategies to specify the center of the radial basis functions like as selecting centers randomly, self-organized selection of centers and supervised selection of centers. The simplest and fastest approach is random selection of centers from the training data set. This is considered to be a sensible approach, provided that the training data are distributed in a representative manner for the problem at hand [22, 24]. In random selection of centers method the width of all Gaussian radial basis function are fixed at:

$$\sigma = \frac{d_{max}}{\sqrt{2m_1}} \tag{2}$$

Where m_1 is the number of centers and d_{max} is the maximum distance between the chosen centers. This formula ensures that the individual radial basis functions are not too peaked or too flat. As mentioned in [21] individually scaled centers with broader widths in areas of lower data density can be used. The lower data density in presented problem is corresponding to extended background features region as it can be seen by green points from Fig. 4b. Random feature points from both object and extended background feature are used as a center of activation functions. The width of Gaussians which their centers are selected from object feature obtained using Equation (2) while the width of Gaussians which their centers are selected from extended background feature are chosen wider for better generalization. In practice, they are chosen 5 times wider than Equation (2) and it is provided satisfactory result. Based on universal approximation theory center and distribution of activation functions are not deterministic if the numbers of hidden neurons being sufficient enough, one can say that the single hidden layer feed-forward network with sufficient number of hidden neurons can approximate any function to any arbitrary level of accuracy.

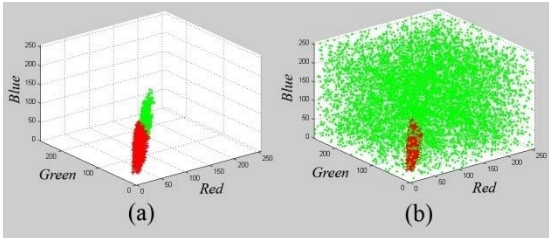

Figure4: (a) Object and background color features in R-G-B color spaces. Green points show object feature, Red Cross pointes show background color features. (b) Object color features and extended background color features. Green points show object feature, Red Cross pointes show extended background color features.

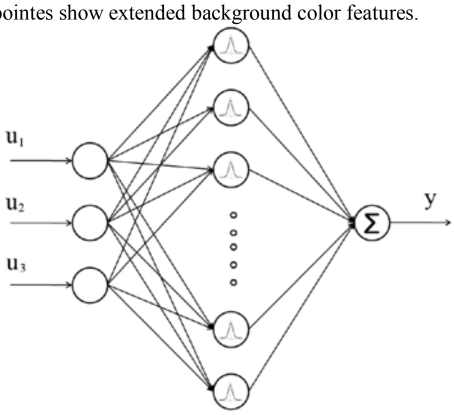

Figure5: The structure of Radial Basis Function Neural Networks

In Figure 5, U is $N \times 3$ dimensional input feature vector which 3 representing R-G-B features and N is number of pixels. Let μ_i and σ_i be the center and width of ith Gaussian hidden neuron and α_i be the interconnection weight between the output neuron and the ith Gaussian neuron. The ith output (y) of RBF Neural Network with K neurons has the following form:

$$y = \sum_{i=1}^{k} \alpha_i \exp\left(-\frac{\|U-\mu_i\|^2}{2{\sigma_i}^2}\right) \tag{3}$$

Equation (3) can be written in matrix form as:

$$Y = \varphi_k \alpha \tag{4}$$

Where

$$\varphi_K(\mu,\sigma,U) = \begin{bmatrix} \varphi_1(\mu_1,\sigma_1,U_1) & \cdots & \varphi_K(\mu_K,\sigma_K,U_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\mu_1,\sigma_1,U_N) & \cdots & \varphi_K(\mu_K,\sigma_K,U_N) \end{bmatrix}$$

Where φ is the Gaussian function. The matrix φ_K (dimension $N \times K$) is called the hidden layer output matrix of the neural network; the ith row of φ is the ith hidden neuron output with respect to inputs $U_1, U_2, \ldots, U_N$. The number of hidden nodes in hidden layer is smaller than the number of the examples available for training for preventing the curse of dimensionality and over fitting. If $\varphi\alpha = T$ then the output weights (α) are calculated by Equation (5):

$$\alpha = \varphi_K^{+} T \tag{5}$$

Where $\varphi_K^{+} = (\varphi_K^{T}\,\varphi_K)^{-1}\,\varphi_K^{T}$ is the Moore–Penrose generalized pseudo-inverse of the hidden layer output matrix [15, 22, 23].

A study on how the classification performance of RBFNN classifier varies against the number of hidden neurons presented, as it is often studied in neural network literature. The first frame of typical tracking scenario is used for training, while the second frame is used for testing. In order to calculate the classification accuracy of the object Equation (6) is used:

$$Object\ Detection\ Rate\ (\%) = \left(\frac{A}{B}\right) \times 100 \tag{6}$$

Where A is the number of correctly detected object pixels; B is the number of object pixels extracted manually. The number of hidden neurons was varied from 9 to 12 and the performance was analyzed and reported in Table 1 for a typical target. The higher generalization accuracy implies better tracking performance. The best generalization performance is achieved by 11 hidden neurons as it can be seen from the table 1. The effect of background extension in RBFNN classifier performance and accuracy is studied. Fig. 6a shows the decision boundary provided by RBFNN classifier in normalized R-G-B color spaces when extracted features of object and surrounding background is used for training. The decision boundary provided by using extracted features of object and extended background is shown in Fig. 6b. By comparing Fig. 6 to Fig. 4, it can be seen whereas background may be changed in consecutive frames then extracted features of surrounding background could not be reliable alone. Hence, background extension is developed to overcome background color change during tracking. As it can be seen in Fig. 6b it is illustrated the background extension modules made the RBFNN classifier model object more accurate. It is expected As long as the object and background is distinguishable in R-G-B color spaces, the RBFNN classifier perform satisfactorily.

D. Object Localization

At first frame object selected approximately and selected region segmented to separate the object from the background more accurately. Object localization for next frame starts at the centroid of segmented object. For next frames in order to find the object pixels, the features are extracted from object rectangle and are tested with RBFNN classifier. For tracking object mean-shift procedure is used.

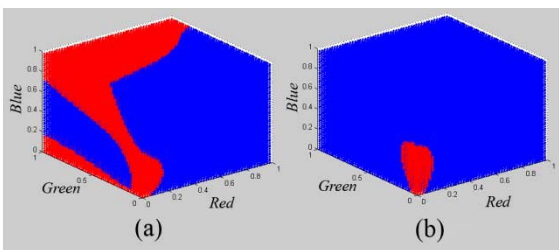

Figure6: (a) The decision boundary of RBFNN classifier when extracted features of object and surrounding background is used for

training in normalized R-G-B color spaces, (b) The decision boundary obtained by using extracted features of object and extended background features. (a) and (b) are corresponding with (a) and (b) in Fig. 4.

TABLE I: Object Detection Rate for first frame (training) and Object Detection Rate for second frame (testing), as the number of hidden neurons (k) increases.

k/ Object detection rate	training	testing
9	76.23%	71.44%
10	76.50%	72.47%
11	77.71%	73.87%
12	78.36%	71.25%

The main idea behind the mean-shift procedure is to treat the points in the spatial space as a probability density function, where the densest regions in the spatial space correspond to the local maxima which is object location [4, 5]. The displacement of the object is given by the shift in centroid of the object pixels. In each iteration center of object rectangle shift to the centroid of detected binary object. The object rectangle is iteratively shifted and tested until object completely placed inside the rectangle (mean-shift convergence). At each iteration object centroid relocates by using Equations (7) and (8):

$$R_{new} = \frac{\sum_{i=0}^{n} r_i}{n} \quad (7)$$

$$C_{new} = \frac{\sum_{i=0}^{n} c_i}{n} \quad (8)$$

Where r_i and c_i show location of each detected object pixels in frame coordinate; R_{new} and C_{new} show the relocated object centroid in each iteration and n is the number of detected object pixels. In presented work, the centroid movement less than a pixel is considered as a complete convergence. Fig. 7 shows mean-shift convergence procedure for a typical object. In practice the average number of mean-shift iterations for a typical scenario is observed 2.67.

III. EXPERIMENTAL RESULTS

To evaluate the accuracy of the proposed method two tests are performed for every scenario. In first test the object trajectory provided by proposed method is compared with ground truth and Mean shift method [5]. The center of the object for the proposed method and ground truth is defined as a centroid of object silhouette. The center of the object in Mean shift method is defined by central point of the object window. In second test the object detection rate is measured by Equation (6) using detected object region by presented method and true object silhouette determined by the ground truth. Both the algorithms assume knowledge of the object position only in the first frame. The object position is estimated in the following frames, using the corresponding algorithm. The ground truth in every image was determined manually. All the examples were carried out with a core 2 Duo 2 GHz processor with 1 GB RAM under MATLAB R2008B. Three test sequences were employed in the evaluation consisting of indoor and outdoor testing situations. Sequence 1 (Fig. 8) shows a person walking from left to right in outdoor environment (PETS 2001 workshop [25]).

Figure7: the iteratively shifting of object rectangle location until convergence.

Figure8. From left to right and up to down shows tracking result for proposed method (dashed red), Mean shift method [5] (dash-dot blue) and ground truth (solid green) for PETS 2001 Sequence. Binary images show the object silhouette obtained by proposed method.

Sequence 2 shows ball sequence available online at [26]. The last sequence is created by a cell phone camera by author. It shows a person walking from right to left in indoor situation when the subject is face tracking.

The first sequence is 200 frames long (frames 850-1050 a scenario from PETS). Video sequences are 576 _ 758 in resolution. The tracking results for frames 850, 900, 950, 975, 990, 1000, 1025, and 1050 are shown in Fig. 8. In this scenario background color change and partial occlusion occurred however the object is tracked accurately. As it is expected the object and background classification is acceptable when the untrained background pixels are tested with RBFNN in later frames. One advantage of the Neural Network is that if an object is temporarily occluded, it will not adversely affect the ultimate object and background classification. It is observed that Mean shift method [5] drifts away from the object of interest when object undergoes partial occlusion whereas proposed method is able to track the object. As it can be seen as long as the object and background is distinguishable in RGB color feature space, the proposed method will provide trusty silhouette of object. The object trajectories of the proposed approach and Mean shift method [5] are compared against the ground truth in Fig. 9. Since the camera is fixed, the trajectory is presented on image plane. For better comparison only important part of frame pane is plotted. The trajectory plot shows that the proposed approach lies close to the ground truth throughout the sequence, whereas performance of Mean shift method [5] degrades and drifts away from the object when the object undergo partial occlusion. Fig. 10 shows the object detection rate of proposed method measured by Equation (6) using ground truth frames. The average object detection rate is 63.60%.

A second example evaluates proposed tracking method for Ball sequence. The sequence has 50 frames of 240 _ 352 pixels. The tracking results for frames 1, 5, 10, 20, 26, 30, 35, 40, 45, and 50 are shown in Fig. 11. Fig. 12 shows the trajectory plot corresponding to the sequence of Fig. 11.

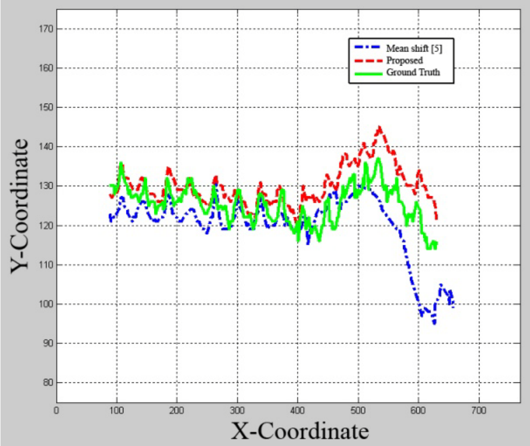

Figure9: Comparison of trajectories for proposed approach and Mean shift method [5] against ground truth for sequence shown in Fig. 8.

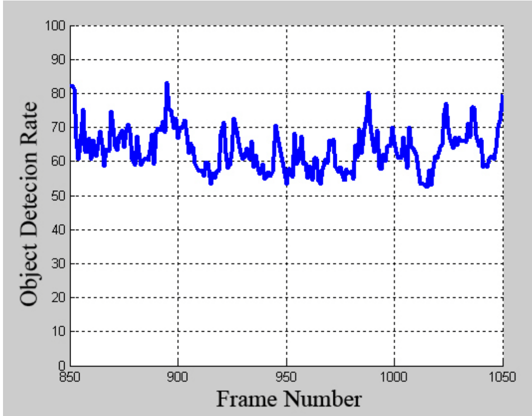

Figure10: Object detection rate of proposed approach for sequence shown in Fig. 8.

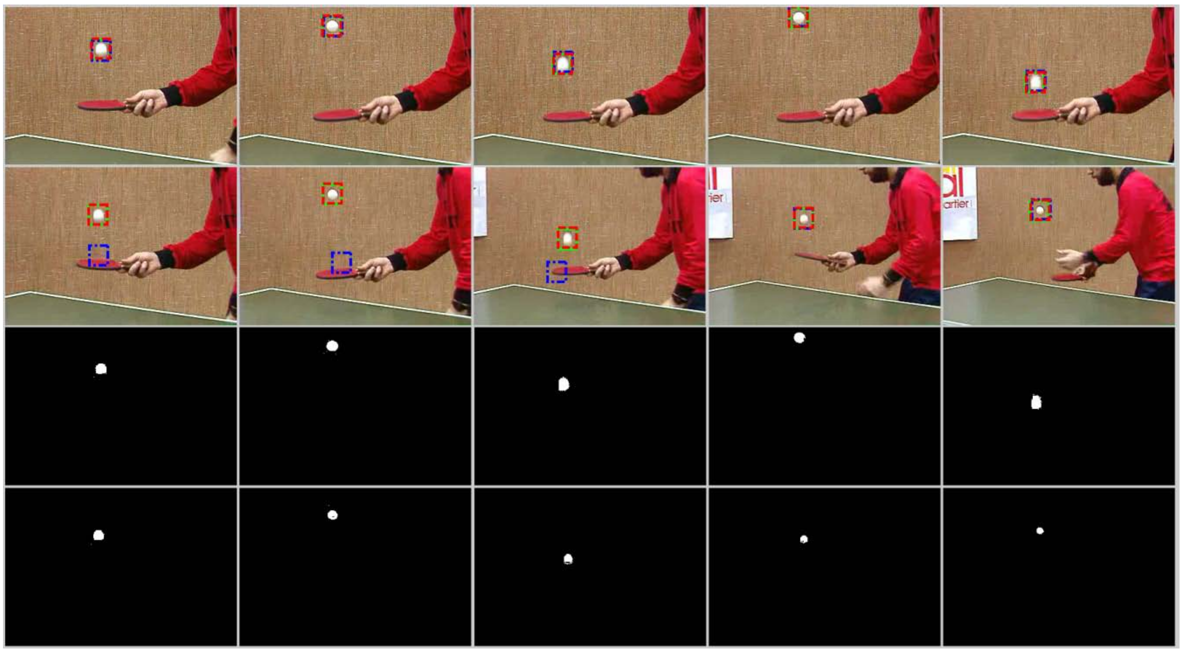

Figure11: From left to right and up to down shows tracking result for proposed method (dashed red), Mean shift method [5] (dash-dot blue) and ground truth (solid green) for Ball Sequence. Binary images show the object silhouette obtained by proposed method.

For better illustration and because of repetitive movement of object the trajectory is shown as the number of frames varies. As it can be seen from Fig. 11 and Fig. 12, Mean shift method [5] has a great drift between frames 29 and 41, whereas the proposed approach slightly drifts during illumination change between frames 11 and 13 and also in frame 27. The both trackers track the object till the end of the sequence. The object silhouettes provided by proposed method also are shown in Fig. 11. The object detection rate of proposed method is shown in Fig. 13 as the number of frames varies. The average object detection rate obtained in this scenario is 69.53%. Zero object detection rates between frames 11 and 13 and also in frame 27 as it mentioned earlier is because of miss detection caused by illumination change and fast movement of ball.

The last example shows a face tracking scenario. This time the face of moving person is segmented and tracked. Note that the camera is moving. The original sequence has 211 frames of 480 _ 640 pixels. In this work, frames between 130 and 165 are used as a scenario. The tracking results for frames 130, 135, 140, 145, 150, 155, 160 and 165 are shown in Fig. 14. The object trajectories of the proposed approach and Mean shift method [5] are compared against the ground truth in Fig. 15. Because of camera movement the trajectory is shown along with number of frames. The trajectory plot shows that both methods remain close to the ground truth throughout the sequence. Fig. 16 shows the object detection rate as the number of frames varies. The average object detection rate is 59.23%.

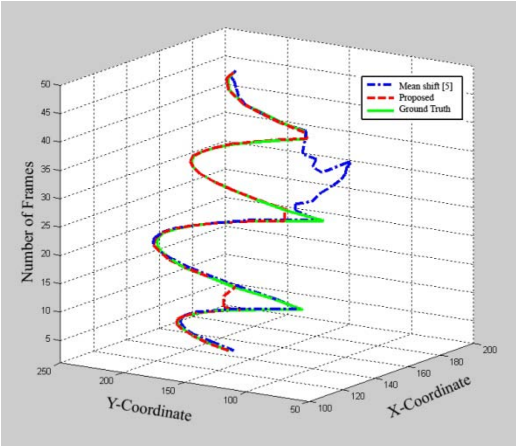

Figure12: Comparison of trajectories for proposed method and Mean shift method [5] against ground truth for sequence shown in Fig. 11.

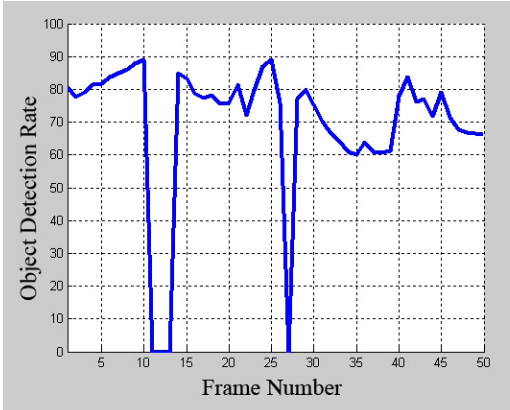

Figure13: Object detection rate of proposed approach for sequence shown in Fig. 11.

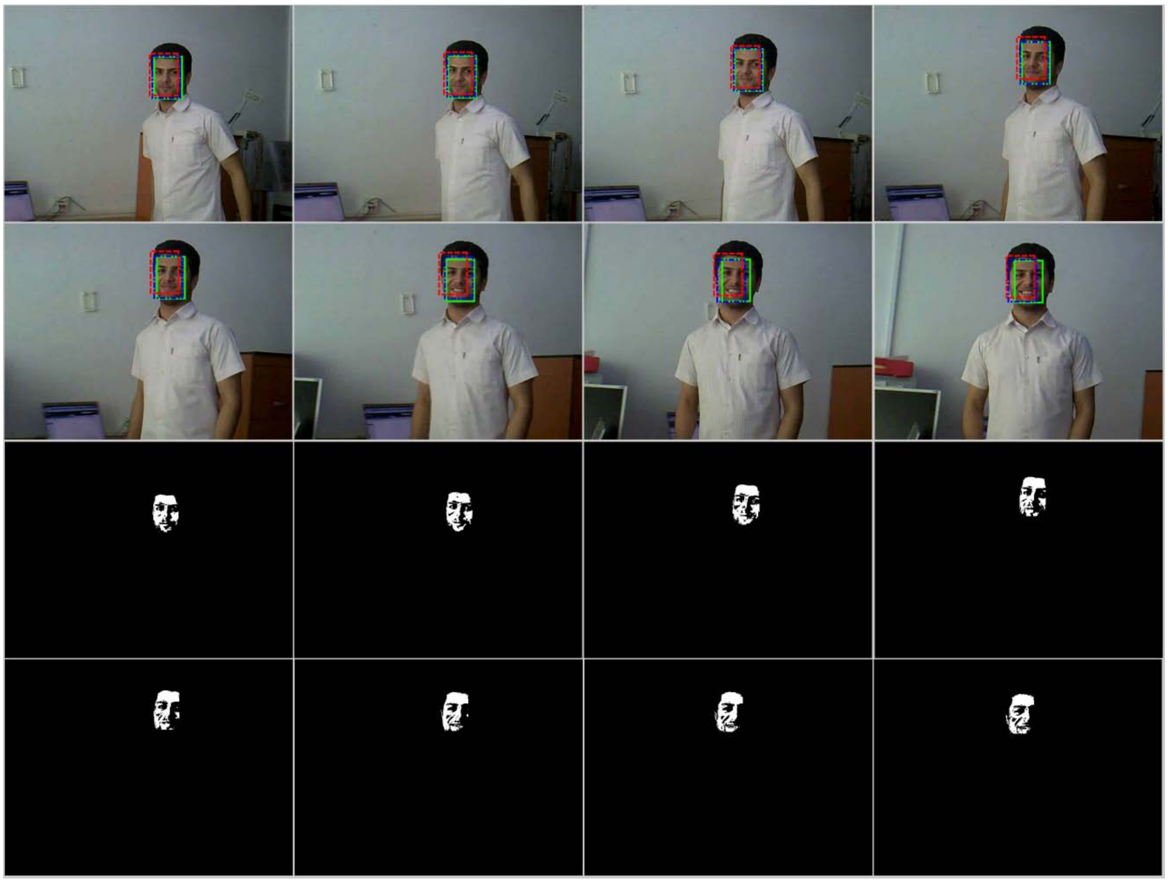

Figure14: From left to right and up to down shows tracking result for proposed method (dashed red), Mean shift method [5] (dash-dot blue) and ground truth (solid green) for face tracking Sequence. Binary images show the object silhouette obtained by proposed method.

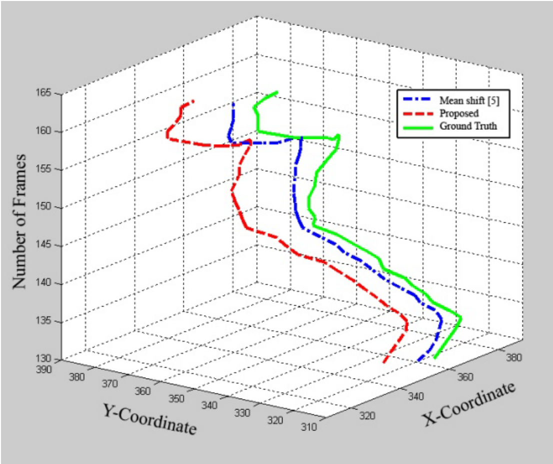

Figure15: Comparison of trajectories for proposed method and Mean shift method [5] against ground truth for sequence shown in Fig. 14.

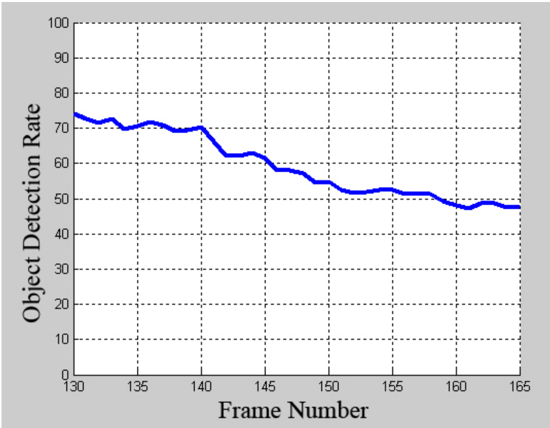

Figure16: Object detection rate of proposed approach for sequence shown in Fig. 14.

The object detection rate reduce as the number of frames increases that is because of growing shadow in right side of face produced by movement of person to the source of light. As it can be seen the proposed algorithm provides the satisfactory object silhouette and tracks the object till the end of the sequence.

IV. CONCLUSIONS

In this paper, an improved method of object tracking using optimized k-means color segmentation, radial basis function neural network and mean-shift procedure proposed. The k-means color segmentation is used to detect object in the initial frame by using R-G-B color features. Background extension is used to improve RBF neural network performance when background colors change during tracking. RBF neural network is trained by object features and extended background features. Trained RBF neural network is used for classifying object and non-object pixels in other frames. Object localization is achieved by detecting object in each of the frames by using RBF neural network together with mean-shift procedure for tracking object location.

Proposed method is able to find an accurate object region in each frame that can be used in high level vision like as action recognition. Since the optimized k-means color segmentation and trained RBF neural network incurs very low computational load, the proposed algorithm is suitable for real-time applications. In this paper the object color is assumed homogenous. The k-means clustering or other clustering methods can be more improved for segmenting multi color objects in first frame. In addition to R-G-B color feature, gradient and texture-based features can be utilized for better object and background discrimination. As it shown presented method can successfully handle partial occlusion of object but for handling full occlusion of object the proposed method can be combined with prediction methods (e.g. kalman filter). For a specific application a priori information of that application domain can be used to adapt and improve presented method.

ACKNOWLEDGMENT

Alireza Asvadi thanks Dr. Reza Ghaderi for stimulating discussions on the subject and Seyed Amir Mohammadi that has shared the code of mean-shift tracking.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Computing Surveys, Vol. 38, No. 4, pp. 1–45, December 2006.

[2] A C. Veenman, M. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 23, No. 1, pp. 54–72, December 2001.

[3] W. Zeng, G. Zhu, Y. Li, "Point Matching Estimation for Moving Object Tracking Based on Kalman Filter," in: Proc. ICIS 2009, pp. 1115–1119.

[4] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in: Proc. ICCV 1999, pp. 1197.

[5] D. Comaniciu, V. Ramesh and P. Meer. Kernel-based object tracking. IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 25, pp. 564–575, 2003.

[6] C. Tomasi and J. Shi, "Good features to track," in: Proc. CVPR 1994, pp. 593–600.

[7] B. Han, C. Paulson, T. Lu, D. Wu, J. Li, "Tracking of multiple objects under partial occlusion," in: Proc. SPIE 2009, pp. 13–17.

[8] A. Yilmaz, X. Li, and M. Shah, "Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras," IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 26, No. 11, pp. 1–6, November 2004.

[9] M. Bertalmo, G. Sapiro, and G. Randall, "Morphing active contours, " IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 22, No. 7, pp. 733–737, 2000.

[10] S. Avidan, "Ensemble Tracking," IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 29, No. 2, pp. 261–271, February 2007.

[11] A. Bugeau, P. Pérez, "Detection And Segmentation Of Moving Objects In Complex Scenes," Computer Vision And Image Understanding, Vol. 113, No. 4, pp. 459–476, 2009.

[12] T. Brox, M. Rousson, R. Deriche, J. Weickert, "Colour, texture, and motion in level set based segmentation and tracking," Image and Vision Computing, Vol. 28, No. 3, pp. 376–390, March 2010.

[13] R.Collins, Y.Liu and M.Leordeanu, "On-Line Selection of Discriminative Tracking Features," IEEE Transactions On Pattern Analysis And Machine Intelligence , Vol. 27, No. 10, pp.1631–1643, October 2005.

[14] R.Collins, Y.Liu, "On-Line Selection of Discriminative Tracking Features," in: Proc. ICCV 2003 , pp.346–352.

[15] R.V. Babu, S. Suresh and A. Makur, "Online adaptive radial basis function networks for robust object tracking," Computer

Vision And Image Understanding, Vol. 114, No. 3, pp. 297–310, 2010.

[16] M. Bouzenada, M.C. Batouche, Z. Telli, "Neural network for object tracking," Information Technology Journal, Vol. 6, No. 4, pp. 526–533, 2007.

[17] J. Ahamed, M.N. Jafri, J. Ahamad, M.I. Khan, "Design and implementation of a neural network for real-time object tracking," in Proceedings of World Academy of Science, Engineering and Technology, vol. 6, 2005, pp. 209–212.

[18] A. Asvadi, M. Karami-Mollaie, Y. Baleghi, H. Seyyedi-Andi, "Improved Object Tracking Using Radial Basis Function Neural Networks," in: Proceedings of 7th Iranian Machine Vision and Image Processing, November 2011.

[19] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[20] Y.M. Cheung, "K*-Means: A new generalized k-means clustering algorithm," Pattern Recognition Letters, Vol. 24, No. 15, pp. 2883–2893, 2003.

[21] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," Pattern Recognition, Vol. 33, No. 9, pp. 1455–1465, 2000.

[22] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Pearson, 1999, pp. 278–340.

[23] R. Penrose, "A generalized inverse for matrices," in: Proceedings of the Cambridge Philosophical Society, 1955, pp. 406–413.

[24] D. Lowe, "Adaptive radial basis function nonlinearities, and the problem of generalisation," in: Proceedings of First IEE International Conference on Artificial Neural Networks, 1989, pp. 171–175.

[25] http://www.cvg.rdg.ac.uk/slides/pets.html

[26] http://media.xiph.org/video/derf/

Alireza Asvadi was born in 1984 in Sari (Iran). He received his B.Sc. degree in electrical engineering in 2008 from University of Isfahan, Isfahan, Iran. He is currently working toward the M.Sc. degree in the Department of Electrical & Computer Engineering, Babol (Noushirvani) University of Technology, Babol, Iran. His research interests include the field of computer vision, with an emphasis on object tracking.

Mohammad Reza Karami received the B.Sc. degree in electronical engineering in 1992, M.Sc. degree in Signal Processing in 1994, and the Ph.D. degree in Biomedical Engineering from I.N.P.L d'Nancy of France. He is now the Associate at the Department of Electrical & Computer Engineering, Babol University of Technology. Since 1998 his research is in signal and speech processing. He has published 80 articles in journals and conferences.

Yasser Baleghi is an Assistant Professor of Electronic Engineering at Babol University of Technology. He holds a Ph.D. degree in Electronic Engineering from Iran University of Science & Technology. His research interests are evolvable and adaptive hardware, image processing and fault tolerant system design.