Kernel based Approach toward Automatic object Detection and Tracking in Surveillance Systems

Amir Aliabadian, Esmaeil Akbarpour, Mohammad Yosefi

Abstract— ***A modified object-tracking algorithm that uses the flexible Metric Distance Transform kernel and multiple features for the Mean shift procedure is proposed and tested. The Faithful target separation based on RGB joint pdf of the target region and that of a neighborhood surrounding the object is obtained. The non-linear log-likelihood function maps the multimodal object/background distribution as positive values for colors associated with foreground, while negative values are marked for background. This replaces the more usual Epanechnikov kernel (E-kernel), improving target representation and localization without increasing the processing time, minimizing the similarity measure using the Bhattacharya coefficient. The algorithm is tested on several image sequences and shown to achieve robust and reliable frame-rate tracking.***

Index Terms—Modified Object tracking, Distance Transform kernel, Mean Shift, Bhattacharyya coefficient, log-likelihood function maps.

I. INTRODUCTION

Tracking moving objects in video sequences is a central concern in computer vision. Reliable visual tracking is an important elementary task in several computer vision-based applications including, video surveillance and monitoring, sensing and navigation in robotics, key-frame detection, video summarization, and many more. Often the goal is to obtain a record of the trajectory of moving single or multiple targets over time and space. Object tracking in video sequences is a challenging task because of the large amount of data used and the common requirement for real-time computation. One can simplify tracking by imposing constraints on the motion and/or appearance of objects. For example, almost all tracking algorithms assume that the object motion is smooth with no abrupt changes. One can further constrain the object motion to be of constant velocity or constant acceleration based on a priori information. Prior knowledge about the number and the size of objects, or the object appearance and shape, can also be used to simplify the problem. Numerous approaches for object tracking have been proposed. These primarily differ from each other based on the way they approach the following questions: Which Object Representation is suitable for tracking? Which image features should be used? How should the motion, appearance, and shape of the object be modeled? The answers to these questions depend on the context/environment in which the tracking is performed and the end use for which the tracking information is being sought. A large number of tracking methods have been proposed which attempt to answer these questions for a variety of scenarios. Tracking objects can be complex due to target scale variations, loss of information caused by projection of the 3D world on a 2D image, camera motion, partial occlusions, clutter, real-time processing requirements and more. Therefore, it is desirable to ensure that the tracker is as efficient as possible. There are two key steps in video analysis: detection of interesting moving objects, tracking of such objects Frame to frame. The main goal of this proposal is to introduce a new framework in both of object Detection and Object Localization in video sequences.

II. PROCEDURE FOR PAPER SUBMISSION

A. Review Stage

Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. A common approach for object detection is to use information in a single frame. However, some object detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detections. This temporal information is usually in the form of frame differencing, which highlights changing regions in consecutive frames. Given the object regions in the image, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks.

B. Robust Method for Object Background Separation

Faithful object tracking can be achieved if we can separate the target region from the background at each time instant. To achieve this, the RGB based joint pdf of the target region and that of a neighborhood surrounding the object is obtained. This process is illustrated in Fig. 1. The region within the red rectangle is used to obtain the target pdf and the region between the green rectangles is used for obtaining the background pdf. This Work uses Pets2001 Video Sequences for showing the proposed algorithm. The resulting log-likelihood ratio of foreground/background region is used to determine object pixels. The log-likelihood of a pixel considered within the outer bounding rectangle as follow [11]:

Manuscript Received January 25, 2012.

Amir Aliabadian, Electrical and Computer Engineering Department, University of Shomal.Amol. IRAN. (E-mail:a.aliabadian@gamil.com).

Esmaeil Akbarpour, Electrical and Compueter Engineering Dep artement, University of Shomal. Amol. IRAN. (E-mail: Esmaeil akbarpour@yahoo.com).

Mohammad Yosefi, Electrical Engineering Department, Shahrood University of Technology, Semnan, IRAN. (E-mail: yoosefimohammad 398@gmail.com).

$$F_{\text{i}} = \text{Log}\frac{max\{\widehat{M}_u,\gamma\}}{max\{\widehat{N}_u,\gamma\}} \qquad (1)$$

Where, $\widehat{\text{M}}_{\text{u}}$ is the color weighting histogram belonging to the target and $\widehat{\text{N}}_{\text{u}}$is normal color histogram of background respectively; and γ is a small non-zero value to avoid numerical instability.

$$\hat{M}_u = C_s \sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)\delta\left[b\left(x_i^*\right)-u\right] \qquad (2)$$

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

Where b function relates corresponding index to each pixel and K is a convex uniformly decreasing and isotropic function.$\text{C}_\text{s}$is Constant and δ is Kronecker delta function.

The Main goal behind the use of kernel is maximize the pixels values in order to separate two classes as best. Furthermore the intrinsic descent spatial behavior of Kernel causes the Value of adjacent pixels near to Boundary seems less noticeable. The non-linear log-likelihood function maps the multimodal object/background distribution as positive values for colors associated with foreground, while negative values are marked for background. The weighting factor M_i is obtained as:

Ui= {1 if Fi > T_0 , 0 otherwise (3)

Where τ_0 is the threshold to decide on the most reliable target pixels. Once the target is localized by user interaction or detection in the first frame, the likelihood map of the object/background is obtained using (3).Then we final mask obtained after Morphological operations. The outer rectangle is chosen in order to have comparable number of pixels from object rectangle as well as background region. If we take a larger rectangle, then far away pixels that are similar to object could weaken the object model. Especially in scenarios with background-clutter, the immediate background pixels play a major role in distinguishing the object, than the farther background pixels. We use outer rectangle with area equal to two times the target rectangle area so that the number of background pixels is approximately the same as the number of pixels.

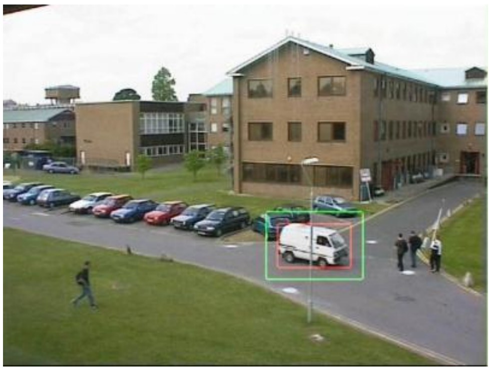
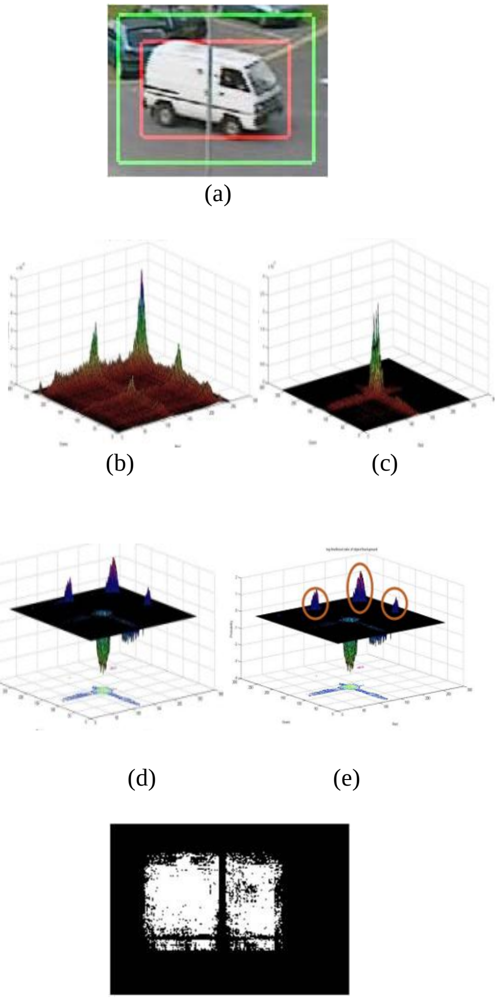


Fig.1.Object and Background is bounded by red and green rectangular respectively.



(a)

(b) (c)

(d) (e)

(f)

Fig.2. Depicts of Proposed Method :(a) Boundary of Target and Background. (b)Weighted color histogram of Foreground. (c) Background color histogram. (d) F_i Mapping. (e) Boundary regions are selected by $\text{U}_\text{i}$and (f) Mask obtained after morphological operation.

III. OBJECT TRACKING

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. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object Regions in every frame are obtained by means of an object detection algorithm, and then the tracker correspond objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. In either tracking approach, the objects are represented using the shape and/or appearance models. The model selected to represent object shape limits the type of motion or deformation it can undergo. For example, if an object is represented as a point, then only a translational model can be used. In the case where a geometric shape representation like an ellipse is used for the object, parametric motion models like affine or projective transformations are appropriate. These representations can approximate the motion of rigid objects in the scene. For a non-rigid object, silhouette or contour is the most descriptive representation and both parametric and nonparametric models can be used to specify their motion.

IV. PROPOSED METHOD FOR TARGET LOCALIZATION

Mean Shift (MS) is nonparametric statistical methods are based on Target Localization and Representation to find the nearest mode of a point sample distribution, that has been adopted as an efficient technique for image segmentation and object tracking. In basic method which uses usual Epanechnikov kernel (E-kernel), the feature histogram-based target representations are regularized by spatial masking with an isotropic kernel. The basic mean shift algorithm relies on color cues. In this algorithm [2] the new initial center of target in every frame estimated by:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)} \qquad (4)$$

(5)

Where

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

$\hat{p}_u(y)$ Is metric estimated model of candidates based on color histogram and $\hat{q}_u$ is Target model, which can be calculated in the first frame or be updated in the next frames.

The final target localization algorithm is described as follow:

1- Calculate the target model $\left\{\hat{q}_u\right\}_{u=1...m}$ and its position $\hat{y}_0$ in first frame.

2- Calculate $\left\{\hat{p}_u\left(\hat{y}_0\right)\right\}_{u=1...m}$ and compute $\rho\left[\hat{p}_u\left(\hat{y}_0\right), \hat{q}_u\right]$ in current frame.

ρ Is Similarity measure obtained by:

$$\hat{\rho}(y) \equiv \rho\left[\hat{p}(y), \hat{q}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\, \hat{q}_u} \qquad (6)$$

3- Obtain the weight w_i using (5)

4- Finding the new target location using (4)

5- If $\left\|\hat{y}_1 - \hat{y}_0\right\| \langle \varepsilon$ Stop otherwise $\hat{y}_0 \leftarrow \hat{y}_1$ and go to step1.

Color is a meaningful part of many tracking algorithms. Since color histograms are robust to partial occlusion, scale and rotation invariant, the resulting algorithm can efficiently and successfully handle non-rigid deformation of the target and rapidly changing dynamics in complex unknown background. The benefit of color is that it is a weak model and is therefore unrestrictive about the type of objects being tracked. The main difficulty for tracking with color alone occurs when the region around the target contains objects with similar color. When the region is cluttered in this way a single cue does not provide reliable performance because it fails to fully model the target. Therefore one challenge is to find solution of this problem caused to make a robust algorithm. Rarely texture and edge features have been widely used for video based tracking purposes. Furthermore, they have not been applied to tracking with mean shift technique. Some another recent research has been concentrated on cue-selection approach which in most case the present visual cues was evaluated by using Potential filtering applications in video sequences. In continue we aim at to develop idea by expanding algorithm based on mentioned multiple features. We are going to show that color, texture and edge complete each other and provide reliable performance.

V. ESTIMATION OF TARGET LOCATION USING MULTIPLE FEATURES

We search a region in new frame on which is most similar to target. We search our object in new frame around the target location in previous frame and estimate most similarities as new location of the target because target motion in two sequential frames is not considerable. We define similarity criteria for comparison of two Histograms. In this Work we suggest Bhattacharya distance for each feature independently, which can be defined as below:

$$\hat{\rho}_c(y) \equiv \rho\left[\hat{p}(y), \hat{q}\right] = \sum_{u_c=1}^{m_c} \sqrt{\hat{p}_{u_c}(y)\, \hat{q}_{u_c}} \qquad (7)$$

$$\hat{\rho}_e(y) \equiv \rho\left[\hat{p}(y), \hat{q}\right] = \sum_{u_e=1}^{m_e} \sqrt{\hat{p}_{u_e}(y)\, \hat{q}_{u_e}} \qquad (8)$$

$$\hat{\rho}_t(y) \equiv \rho\left[\hat{p}(y), \hat{q}\right] = \sum_{u_t=1}^{m_t} \sqrt{\hat{p}_{u_t}(y)\, \hat{q}_{u_t}} \qquad (9)$$

A method is presented here which takes account of the Bhattacharyya distance (6) to give some significance to each feature based on the current frame. Using the smallest value of the distance measure $\hat{\rho}(y)$ for each feature the weight for each feature f is determined by: f=1, 2… F

$$\hat{\varepsilon}_f = \frac{1}{\hat{\rho}_{f\min}} \qquad (10)$$

The weights are then normalized such that $\sum_{f=1}^{F} \varepsilon_f = 1$.

$$\varepsilon_f = \frac{\hat{\varepsilon}_f}{\Sigma_{f=1}^{F} \hat{\varepsilon}_f} \qquad (11)$$

In this way, the new location of the target in current frame for each feature space is calculated independently with (4) then final location of target in that iteration is given by:

$$\hat{y}_{1multiple} = \varepsilon_c \, \hat{y}_{1c} + \varepsilon_e \, \hat{y}_{1e} + \varepsilon_t \, \hat{y}_{1t} \qquad (12)$$

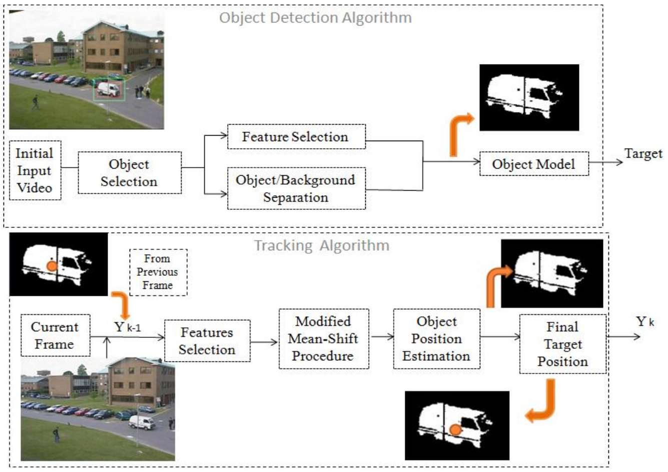
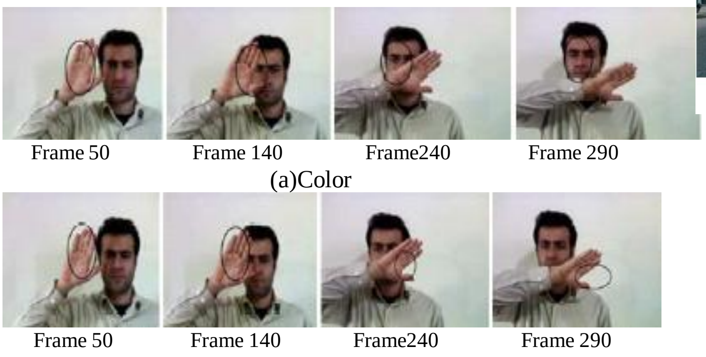


Fig.3.Proposed Framework For object Detection and Localization

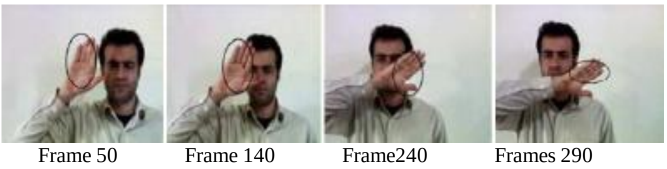
VI. EXPREMENTAL RESULTS

We implemented the proposed algorithm on several image sequences. The evaluation of the modified mean shift object tracking in comparison with the basic radials symmetric E-kernel is presented, too. We track moving objects, a static object with a moving camera and a combination of the two. All the tests were carried out on a Pentium 4 CPU 3.60 GHz with 1GB RAM. The first sequence includes hand tracking. In a hand-tracking scenario the Mean Shift with color features (Fig. 4(a)) fails in the frame of 260 while the modified Mean Shift (Fig. 4(c)) is successful in that frame. As can be seen in (fig. 4(a) and 5(b) -frames 290) the mean shift tracker with color and edge features are unable to track successfully, while the modified. Mean shift with combined color and edge features (Fig. 4(c)) successfully tracks the hand through the entire sequence.



(a)Color

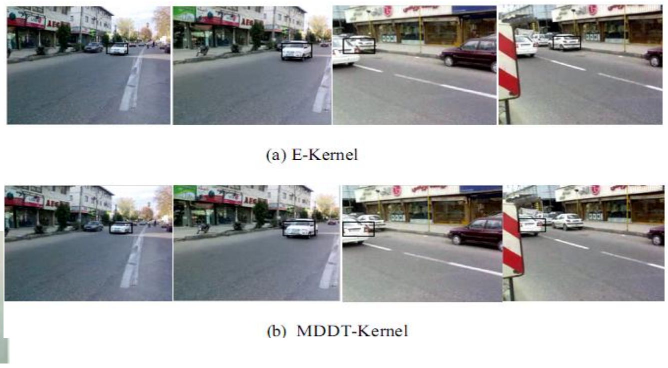
(b)Edge



(C)Color & Edge

Fig.4. Hand tracking scenario. The Mean Shift with color features (Fig. 4(a)) fails in the frame of 240 while the Mean Shift with multiple features (Fig. 4(c)) is successful in that frame. As can be seen in (fig. 4(a) and 4(b)-frames 290) the mean shift tracker is unable to track successfully as the hand is moved in front of the face. The mean shift with combined color and edge features (Fig.4(c)) successfully tracks the hand through the entire sequence.

In the second experiment, we compare the tracking of a moving car in a video sequence that includes 450 frames of 320´ 480 pixels, comparing the normal E-kernel with the MDDT kernel. The simulation used one frame for each 5 frames. The target location was initialized by a rectangular region (shown) of size 86 ´41pixels. Fig.5 (a) and (b) show some examples, frames 1, 35, 60 and 75, from the whole sequence. In frame 60 some of the original car is still contained within the window, but after the 75nd frame, the car is lost completely in Fig.5 (a), as the tracker finally latches on to another crossing car. This demonstrates that the inclusion of the background of the tracked car (in this case another car) includes pixels that are similar in color space, so that the algorithm fails to identify the correct distribution in succeeding frames and hence follows the wrong target. Fig.6 shows the value of iteration computed for each frame. In Fig.7 distance function, which calculated by the Bhattacharyya coefficient, is presented. The peak in the E-kernel data is 0.636 which increases to the 0.7 in MDDT. Fig.8 shows the similarity surfaces made by candidate models in frame 35 with E-kernel and MDDT kernel, (a) and (b) respectively. Initial point is center of target model in frame of 33 and extend of simulation is 60×60.



(a) E-Kernel

(b) MDDT-Kernel

Fig.5. Tracking the crossing car

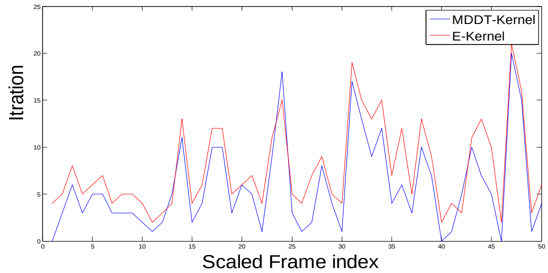


Fig.6. Iteration Value for selected Frames.

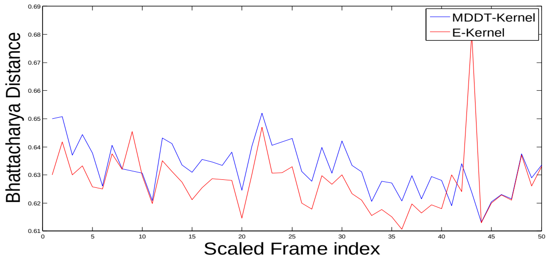
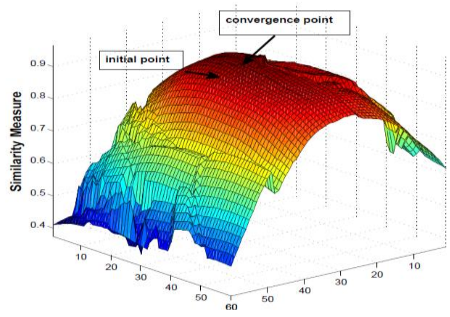
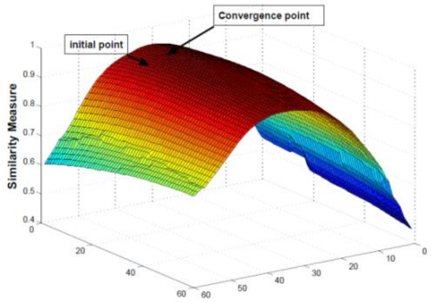


Fig.7. The Bhattacharya distance values, calculated by the Bhattacharyya coefficient.



(a)



(b)

Fig.8. The similarity surfaces (values of the Bhattacharyya coefficient) for frame 35. The initial points and convergence points are shown. (a) The result from the E-kernel. (b) The result from the MDDT-kernel.

In terms of complexity, computed from 20 executions of the program, the average selected frames per second of the MDDT kernel and the E-kernel are 15.76 and 17.57 respectively. The maximum numbers of iterations within a single frame are 14 and 20, respectively. The average times per frame are roughly comparable because although the speed of convergence is quicker with the MDDT-kernel, additional processing is required to segment the target window, in order to get more robust and accurate tracking. From Table 1, which shows quantitative results, the MDDT kernel algorithm needs on average only 8.2 iterations to converge to the optimal result, but the E-kernel needs 14.4 iterations on average, the greatly reduced number of iterations balances the greater complexity of computing the MDDT kernel, so the processing speed per frame is comparable.

Table1. Comparison results of MDDT and E-kernel method

Method	Average iteration s	CPU time (sec./frame)		
		max	min	mean
E-kernel	14.4	0.4453	0.3534	0.3993
Proposed Method	8.2	0.3688	0.2914	0.3301

VII. CONCLUSION

We have described the implementation of a scaling, normalized Metric distance kernel as a weighting and constraining function applied to the mean shift tracking algorithm that maximizes the similarity between model and candidate distributions in multiple features space. In comparison with the E-kernel, used as an exemplar of a radially symmetric function, application of the MDDT-kernel can achieve better results because it can reject false nodes that are caused by the inclusion of changing background pixels. Using multi-features make similarity surfaces more convergence. The processing time is sufficiently small for real time operation, as the added cost of foreground-background separation is offset by the more rapid finding of the correct mode. The results presented on a number of video sequences show that the MDDT-kernel algorithm with multiple features performs well in terms of improved stability, accuracy and robustness on camera motion and partial occlusions.

REFRENCES

1. COMANICIU, D. AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. IEEETrans. Patt. Analy. Mach. Intell. 24, 5, 603–619.
2. COMANICIU, D., RAMESH, V., AND MEER, P. 2003. Kernel-based object tracking. IEEE Trans. Patt. Analy. Mach.Intell. 25, 564–575.
3. JEPSON, A., FLEET, D., AND ELMARAGHI, T. 2003. Robust online appearance models for visual tracking. IEEETrans. Patt. Analy. Mach. Intell. 25, 10, 1296–1311.
4. KANG, J., COHEN, I., ANDMEDIONI, G. 2003. Continuous tracking within and across camera streams. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 267–272.
5. KANG, J., COHEN, I., AND MEDIONI, G. 2004. Object reacquisition using geometric invariant appearance model. In International Conference on Pattern Recognition (ICPR). 759–762.
6. KHAN, S. AND SHAH, M. 2003. Consistent labeling of tracked objects in multiple cameras with over lapping fields of view. IEEE Trans. Patt. Analy. Mach. Intell. 25, 10, 1355–1360.
7. LOWE, D. 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60, 2, 91–110.
8. COLLINS, R. AND LIU, Y. 2003. On-line selection of discriminative tracking features. In IEEE International Conference on Computer Vision (ICCV). 346–352.

9. SATO, K. AND AGGARWAL, J. 2004. Temporal spatio-velocity transform and its application to tracking and interaction. Comput. Vision Image Understand. 96, 2, 100–128.
10. SERBY, D., KOLLER-MEIER, S., AND GOOL, L. V. 2004. Probabilistic object tracking using multiple features. In IEEE International Conference of Pattern Recognition (ICPR). 184–187.
11. Jeakar, J. And Venkatesh, R.,2008. Robust object tracking with background-weighted local kernels. Computer Vision and Image Understanding.296-307.
12. Babaiian,A. And Bayesteh, R., 2008. Target Tracking Using Wavelet Features and RVM Classifier. Fourth International Conference on Natural Computation. 575-578.
13. Venkatesh,R. And Suresh, S., 2010. Online adaptive radial basis function networks for robust object tracking. Computer Vision and Image Understanding.297-310.
14. Yu, J. And Tan,J. 2009. Object density-based image segmentation and its applications in biomedical image analysis. Computer methods and programs in biomedicine.193-204.
15. Babaiian,A. And Rastegar, S.2009. Modify Kernel Tracking Using an Efficient Color Model and Active Contour. 41st Southeastern Symposium on System Theory University of Tennessee Space Institute. 59-63.
16. Rastegar,S. And Babaiian, A.2009. Airplane Detection and Tracking Using Wavelet Features and SVM Classifier. 41st Southeastern Symposium on System Theory University of Tennessee Space Institute. 64-67.
17. Li,Q. And Qu, W.2010: Real-time interactive multi-target tracking using kernel-based trackers. The International Conference on Image Processing (ICIP): 689-692.

AUTHOR PROFILE

Amir Aliabadian: Amir Aliabadian was born in Babol, Iran, in 1982.He received his B.Sc.degree in Electrical Engineering from the Mazandaran University and his M.Sc.degree in telecommunication system in Emam Hossein University, Iran, in 2004 and 2007, respectively. Now, He is Faculty member of Electronic and Computer Engineering Department of Shomal University, Amol, Iran.(Email:a.aliabadian@gmail.com).

Esmaeil AkbarPour: Esmaeil Akbarpour was born in Amol,Iran,in 1979. He Received his Both B.Sc and M.sc.degree in Electrical Engineering from the Amrkabir University. Now, He is Faculty member of Electronic and Computer Engineering Department of Shomal University, Amol, Iran. His special interest is Image processing and intelligent systems. (Email:Esmaeilakbarpour@yahoo.com).

Mohammad Yosefi: Mohammad Yosefi was born in Gorgan, Iran, in 1984. He received his B.Sc.degree in Electrical Engineering from the Nooshirvani University of Technology and his M.Sc.degree in electrical Engineering in Shahrood University, Iran, in 2011 and 2007, respectively. His special interest is Robotic and Intelligent Control Systems. (Email: yoosefimohammad 398@ gmail.com).