IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. IV (May-Jun. 2014), PP 20-26
www.iosrjournals.org

Video Object Tracking Mechanism

R. Sathya Bharathi
(M.E.II Year, Department of Computer Science and Engineering. K.Ramakrishnan College of Technology Tiruchchirappalli, India)

Abstract: *The video object mechanism is based on the appearance representation of the object. Tracking objects in a video has applications in video surveillance and other some applications. In previous method does not handle the case of full occlusion. The case of full occlusion can be handled by setting an LSK similarity threshold, which stops tracking when the object is lost. The objects can be tracked based on the compression domain and pixel domain. The spatial information of the object that is being tracked is retrieved using Local Steering Kernels. The Color features are extracted using color histograms. The features similarities between a candidate object ROI and the object ROI in the previous frame and the last stored object instance in the object model are measured. For extracting the foreground extended Kalman filter is used. This increases the tracking efficiency of the algorithm.*

Index Terms: *Color histograms, localsteeringkernels, and visualobjecttracking.*

I. Introduction

The object representation methods can be divided into five categories [1]: model-based, appearance-based, contour-based, feature-based, and hybrid. Model-based tracking methods exploit a priori information about the object shape, creating a 2-D or 3-D model for the object [2]. These methods can address the problem of object tracking under illumination variations, changes in the object viewing angle, and partial occlusion. However, their computational cost is heavy, especially when tracking objects with complex 3-Dgeometry. Moreover, they require the implementation of a detailed model for each type of object in the scene, as the models can not be easily generalized. Appearance-based tracking methods use the visual information of the object projection on the image plane, i.e., color, texture, and shape, as well as information on the 2-D object motion [3]. These methods deal with simple object transformations, such as affine ones, including translation and rotation.

However, they are sensitive to illumination changes. Contour-based tracking methods track the object contour by employing shape matching or contour-evolution techniques [4]. Contours can be represented by active models, such as snakes, B -splines, and geodesic active contours, or meshes [5], enabling the tracking of both rigid and non rigid objects. In order to deal with partially occluded objects, tracking algorithms incorporate occlusion detection and estimation techniques. Feature-based methods perform object tracking by tracking a set of feature points, which represent the object [6]. These tracked features are then grouped, according to their associations in the previous frame. These methods perform well in partial occlusion, as well as in tracking very small objects. The major problem of feature-based methods is the correct distinction between the target object and background features. Finally, hybrid methods for object tracking exploit the advantages of the above-mentioned methods, by incorporating two or more tracking methods [7]. Usually, feature-based methods are employed first, for object detection and localization. Then, region-based techniques are used to track its parts. The main disadvantage of these methods is their high computational complexity.

The problem of partial occlusion in appearance-based tracking schemes has been addressed by decomposing the target object in to non overlapping [13] or overlapping [14] fragments, which are tracked separately. The fragments can be selected either manually or randomly. The number and size of the fragments play an important role in tracking performance, as too many or too big fragments result in heavy computational weight and, on the contrary, too few fragments cause the tracker to drift. The new position of the object can be estimated by various voting techniques for the confidence of each fragment, e.g., by the fragment with the maximum confidence, or by selecting the smaller area which contains the entire fragment tracking results. The changes in the object view angle are handled by either multiple hypotheses for the object state [15], or by considering adaptive appearance models [16]. These methods are based on these quintals Monte Carlo method, also known as particle filters [17]. Other approaches employ a hierarchical frame work based on bounded irregular pyramids [18] and an incremental Eigen basis learning framework [19].

Our tracking approach is an appearance based one using both the CHs to describe object color information and the local steering kernel (LSK) object texture descriptors [20]. A preliminary work on visual object tracking based on LSKs was presented in [21]. We first search image regions in a video frame that have high color similarity to the object CH. Once these candidate regions are found, the illumination-invariant LSK descriptors [20] of both the target object and the candidate search region are extracted. LSKs are descriptors of the image salient features. They were first employed as an image denoising and reconstruction technique [22] and later found application in object detection [20]. As an object detector, they were proven to be robust in small scale and orientation changes, as well as small object deformations.

Therefore, their incorporation in a tracking frame work results in successful tracking of slowly deformable objects. After discarding the image regions with small CH similarity to the object CH, the new position of the object is selected as the image region, whose LSK representation has the maximum similarity to the one of the target object. As tracking evolves, every time the target object appearance changes, either due to rotation/zooming, or a deformation, or a change i n the view angle, the object model, being a stack containing different instances of the object including information about its scale and 2-D angle, is updated with the representation of the most recent detected object instance. This way, the algorithm is able to cope with changes in object appearance. The final decision on the new tracked object location is determined to be the candidate image region with the maximal average LSK similarity to the detected object instance in the previous frame and the most recent instance in the object model (stack).

II. Lskobjecttracking

The proposed frame work makes the assumption that object translation and deformation between two consecutive video frames is rather small. Each transformation of the object image, i.e., scaling due to zooming or rotation, is considered as an object instance and it is stored in a stack, i.e., a list of object instances (images). The stored object instances comprise the object model [10]. As tracking evolves, the object model is updated with new object instances, incorporating the transformations the object undergoes.

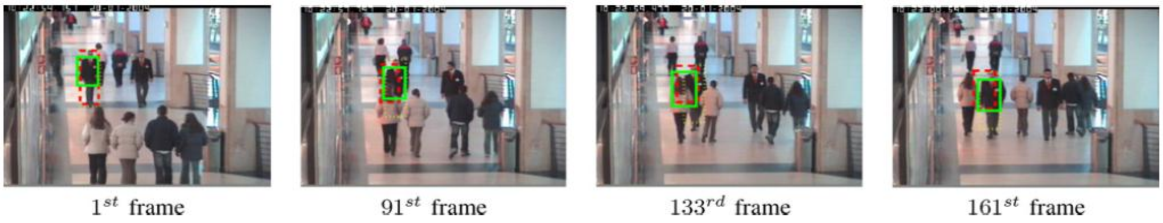
In each new video frame, the new object region of interest (ROI) is searched in a local region around a predicted object position, called search region. These arch region may contain several candidate object ROIs in the new video frame. The algorithm employs spatial information through LSKs [14] and color information through CH for representing both the object instances and the search region. The similarity of the object salient spatial features and CH between a candidate object ROI and the object region in the previous frame and the last updated object instance from the object model (stack) are evaluated. The cosine similarity of the object salient features (i.e., LSK descriptors) is robust to small object appearance changes between two consecutive video frames. In each frame, the patch of the search region with the maximum average LSK similarity to the object image in the previous frame and object instance in the object appearance model is selected as the new object instances.

III. Color Histogram

It is sensitive to changes in illumination and view point changes. After object position prediction and search region selection, the search region is divided into candidate object ROIs. The CHs are compared according to cosine similarity. CH similarity is an indicator of whether the search region patch belongs to the object ROI or the background. The cosine similarity between two histograms **h1, h2**Є R^{256} is given by

$$c(\mathrm{h1}, \mathrm{h2}) = \cos(\theta) = \frac{\langle \mathrm{h1}, \mathrm{h2} \rangle}{\|\mathrm{h1}\| \|\mathrm{h2}\|}$$

Color histograms are the graphical representation of the colors. The color histogram can be built for any kind of color space. The space is divided into an appropriate number of ranges, arranged as a regular grid, each containing many similar color values. A histogram of an image is produced first by discretization of the colors in the image into a number of bins, and counting the number of image pixels in each bin.For digital images, fig(3.2) shows that a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges, that span the image's color space, the set of all possible colors.



1^{st} frame 91^{st} frame 133^{rd} frame 161^{st} frame

Figure 2(a) tracking results of an object with partial occlusion and small variation.

For multi-spectral images, where each pixel is represented by an arbitrary number of measurements (for example, beyond the three measurements in RGB), the color histogram is N-dimensional, with N being the number of measurements taken. Each measurement has its own wavelength range of the light spectrum, some of which may be outside the visible spectrum. If the set of possible color values is sufficiently small, each of those colors may be placed on a range by itself; then the histogram is merely the count of pixels that have each possible color. Most often, the space is divided into an appropriate number of ranges, often arranged as a regular grid, each containing many similar color values. The color histogram may also be represented and displayed as a smooth function defined over the color space that approximates the pixel counts.

IV. Bayesian Filter

A. Overview

In a dynamic system, the process and measurement model are given by

$$xt= g(xt-1, ut)$$
$$zt= h(xt, vt)$$

where vt and ut are the process and the measurement noise, respectively. The state variable xt(t = 0 . . . n) is characterized by its probability density function estimated from the sequence of measurements zt(t = 1 . . . n). In the sequential Bayesian filtering framework,[10] the conditional density of the state variable given the measurements is propagated through prediction and update stages,

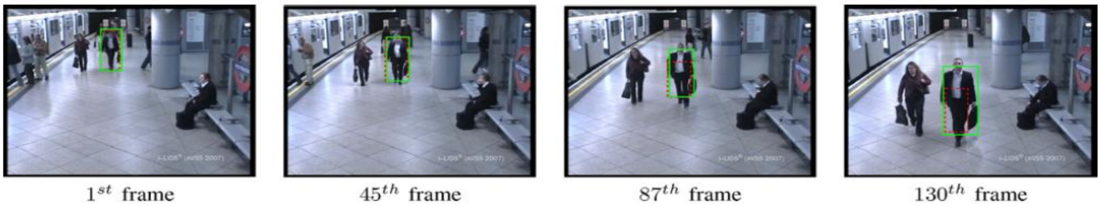
$$p(xt|z1{:}t-1) = \int p(xt|xt-1)p(xt-1|z1{:}t-1)dxt-1$$
$$p(xt|z1{:}t) = \frac{1}{k}p(zt|xt)p(xt|z1{:}t-1)$$

where $k = \int p(xt|xt-1)p(xt-1|z1{:}t-1)dxt$is a normalization constant independent of xt.p(xt−1|z1:t−1) is the prior probability density function (pdf), p(xt|z1:t−1) is the predicted pdf and p(zt|xt) is the measurement likelihood function. The posterior pdf at time step t, p(xt|z1:t), is used as the prior pdf in time step t + 1. At each time step, the conditional distribution of the state variable x given a sequence of measurements z is represented by a Gaussian mixture. Our goal is to retain such a representation through the stages of prediction and update, and to represent the posterior probability in the following step with the same mixture form.

The proposed filtering framework is described as follows. First, unscented transformation (UT) is used to derive a mixture representation of the predicted pdf p(xt|z1:t−1). Second, the density interpolation technique with multi-stage sampling is introduced to approximate the likelihood function with a mixture form. By multiplying two mixture functions, the posterior pdf is obtained through equation. To prevent the number of mix and from growing too large, an algorithm of density approximation based on mode finding is applied to derive a compact representation for the posterior pdf.

B. Lsk descriptors of the search region

LSK value for each image channel is calculated LSKs are local descriptors of the image structure, which exploit both spatial and pixel-value information. They are a nonlinear combination of weighted spacial distances between a pixel of an image and its surrounding.



1^{st} frame 45^{th} frame 87^{th} frame 130^{th} frame

Figure 2(b) tracking results of an object increasing in size.

The distance between an image pixel $\mathbf{p}_i$and its neighboring pixel Pi is measured using a weighted Euclidean distance, which uses as weights the covariance matrix $\mathbf{C}_i$ of the image gradients along x(horizontal) and y(vertical) axes.

$$K_i(p) = \frac{\sqrt{\det(Ci)}}{2\pi}\exp\left\{-\frac{(p_i-p)^T Ci(p_i-p)}{2}\right\},$$

$$i = 1, \dots\dots\dots.., M^2$$

The distance between an image pixel and its neighboring pixel is measured using a weighted Euclidean distance. The resulting projection matrix will then be used for the dimensionality reduction of the LSK descriptors of the search region. Finally the similar LSK descriptors are identified.

C. Extracting the required object of interest

An optimal recursive Bayesian filter for linear functions subjected to Gaussian noise. The kalman filter[15], also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state.

$$\mathrm{K}_t = \mathbf{P}\hat{}_t\, \mathbf{H}^T(\mathbf{HP}\hat{}_t\, \mathbf{H}^T + \mathbf{R})^{-1}$$

$$\mathrm{x}\hat{}_t = \mathbf{x}\hat{}_{t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{Hx}\hat{}_t)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t.$$

The matrix $\mathbf{K}_t$ is called the Kalman gain shown in the fig (3.4) and is chosen such that minimizes the a posteriori error covariance $\mathbf{P}_t$. The object will then be searched in a search region centered at the predicted position **x**t . The size of this region varies according to the expected maximal object velocity, the object size and the reliability of the predicted position.

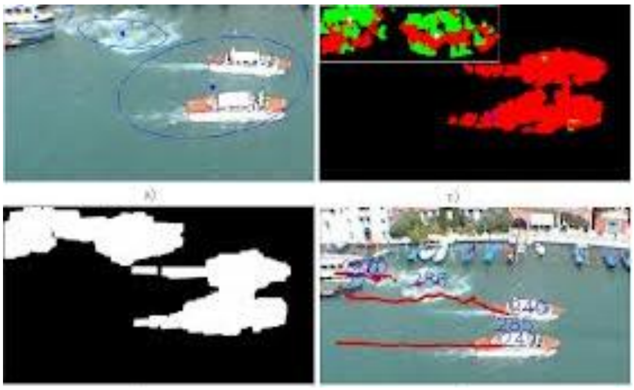


Figure 2(c) kalman filter tracking

If the object moves fast, or it moves in a non smooth trajectory, or it is large and user are not confident on the prediction, user select a large search region. If the object ROI size is $Q_1 \times Q_2$ pixels, then the search region size is set to $R_1 \times R_2 = 2Q_1 \times 2Q_2$pixels. The object ROI dimensions $Q_1 \times Q_2$are selected to be small to increase tracking speed but large enough in order to preserve the object salient features. Typical values of $Q_1 \times Q_2$ are around 30 × 30 p.

V. Experimental Results

A. Experimental Setup

The effectiveness of the proposed tracking scheme was tested in a number of videos coming from various applications. The initialization in each video was performed through a training-free object detection algorithm over the entire first video frame [20]. The search region size is R1 ×R2 = 2Q1 ×2Q2,where Q1 × Q2 are the downscaled object dimensions, which are selected for each experiment table 1.

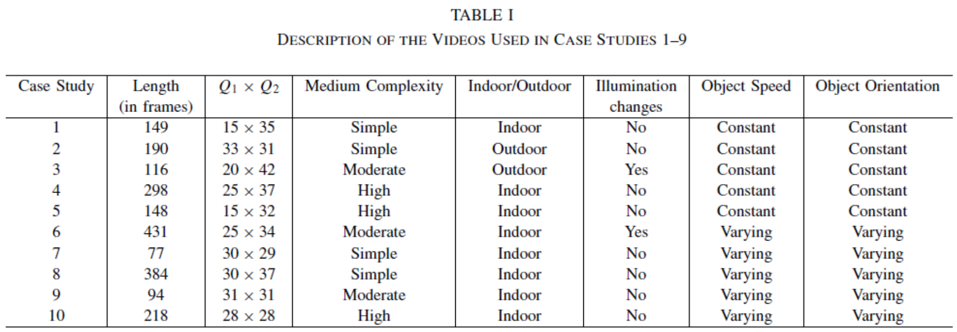


TABLE I
DESCRIPTION OF THE VIDEOS USED IN CASE STUDIES 1–9

Case Study	Length (in frames)	$Q_1 \times Q_2$	Medium Complexity	Indoor/Outdoor	Illumination changes	Object Speed	Object Orientation
1	149	15 × 35	Simple	Indoor	No	Constant	Constant
2	190	33 × 31	Simple	Outdoor	No	Constant	Constant
3	116	20 × 42	Moderate	Outdoor	Yes	Constant	Constant
4	298	25 × 37	High	Indoor	No	Constant	Constant
5	148	15 × 32	High	Indoor	No	Constant	Constant
6	431	25 × 34	Moderate	Indoor	Yes	Varying	Varying
7	77	30 × 29	Simple	Indoor	No	Varying	Varying
8	384	30 × 37	Simple	Indoor	No	Varying	Varying
9	94	31 × 31	Moderate	Indoor	No	Varying	Varying
10	218	28 × 28	High	Indoor	No	Varying	Varying

Figure 3(a) Description of the videos used in case studies

The window size for the LSK feature extraction is 3 × 3pixels. The rotation step is 10 deg, except for some cases, here we do not expect 2-D rotation of the tracked object and we set it equal to zero (e.g., when we track people by using surveillance cameras). The scale step is set to 10%. The threshold for the model update T is zero, which means that, every time the similarity value decreases, we search for possible scale and rotation of the object. Finally, the noise covariance matrix **Q** was set to the identity matrix **Q** = **I** εR4×4 and the value of the measurement noise covariance matrix **R** was set to the identity matrix **R** = **I** εR2×2. This initialization was proven to provide good tracking results.

B. Qualitative Evaluation

The performance of the proposed tracker is compared with two other state-of-the-art trackers found in the literature: one that incorporates an appearance based object representation (i.e., image intensity) with particle filters (called PF tracker) [16] and another one that performs object tracking by dividing the object of interest in smaller fragments (called FT tracker) [13]. The FT tracker is publicly available at the authors' site. In order to have a better understanding of the optimal tracking results for each tracking scheme, each tracker is initialized with a different object ROI that fits best to the characteristics of the tracking algorithm. For example, the PF tracker is initialized inside the tracked object, while the FT tracker and the CH-LSK trackers are initialized in such a way that the initial object ROIs contains the object boundary and a small amount of background.

A summary of the main characteristics of the videos used in the experiments is shown in Table I. a) Case studies 1 and 2: In the first two experiments, we test the performance of the proposed tracking scheme under variations of the object scale. In the first experiment, the purpose is to track a person in the video from the i-LIDS bag and vehicle detection challengedataset,1 while, in the second experiment, we aim to track a car in the video AVSS−PV−Easy−Divx.avi from the i-LIDS bag and vehicle detection challenge dataset.

The tracking results for the proposed CH-LSK tracker and the PF tracker are illustrated in Figs. 1 and 2 for the first and second experiment, respectively. We note that the FT tracker does not take into account changes in scale. Therefore, it was not used in these experiments. We can observe in Fig. 1 that the CH-LSK tracker is successful in tracking the change in the object image size, as opposed to PF tracker, which keeps an almost constant size for the tracked object. On the other hand, both trackers have a similar performance in keeping track of the decreasing size of the car in the second experiment (Fig. 2).b) Case study 3: In this experiment we test the performance of the proposed algorithm in the video of the PETS2001 dataset, 2 which depicts a car moving in a circular trajectory in an Omni directional camera. The experimental results are shown in Fig. 3 for the proposed CH-LSK tracker and the PF tracker.

Again, the FT tracker is not used, as it does not handle rotational motion. The proposed tracker and the PFtracker have the same initialization. The PF tracker loses the object very quickly, while the CH-LSK tracker follows better both the rotation and scale changes of the car. This means that the proposed tracker is more robust in rotation changes. c) Case studies 4 and 5: This case study deals with the problem of object tracking in a video with partial occlusion and small scale variations. We conducted two experiments. In the first experiment, the video used is OneStopMoveEnter1cor.avi from the CAVIAR dataset.3 The object of interest (a man) is partially occluded and his ROI size increases as he moves toward the camera. The tracking results are depicted in Fig. 4. The PF tracker, in the beginning of the video, tracks the full body of the man, but, as tracking evolves, the tracking area gets smaller resulting, after 160 frames, to track only man's torso.

The FT tracker is able to handle partial occlusion and tracks the full body of the man successfully. The CH-LSK tracker tracks successfully only man's torso, as the bounding box contains a significant amount of background area, which affects tracking performance. In the second experiment, the object of interest is the man in the WalkByShop1cor.avi video of the CAVIAR dataset. In this video, more than 75% of the object area is occluded. The tracking results are shown in Fig. 5. PF tracker stops tracking the man, when he walks behind the first person in the foreground (from the right).

FT tracker is able to handle the first occlusion but, due to the fact that it cannot follow the person’s change in scale, it stops tracking the man when he walks behind the second person in the foreground (from the right). On the other hand, the proposed tracker is able to track the man throughout the video. d) Case study 6: In this case study, we test the performance of the proposed CH-LSK tracker in a video with strong changes in illumination conditions. More precisely, we track the face of a person, who moves in a room with half of the lights switched on and which is switched off after awhile. Snapshots of the tracking results are depicted in Fig. 6.

We notice that the proposed CH-LSK tracker is robust to illumination variations, as it tracks the person’s face, either in the case where the illumination change is gradual, i.e., when the person moves slowly toward the lit part of the room, or in the case where the illumination change is abrupt, i.e., when the lights are switched off. The PF tracker has similar behavior to the CH-LSK tracker, while the FT tracker is not able to handle the gradual illumination change: when the person walks in the lit part of the room it drifts to the person’s t-shirt, which has more similar color to the person’s face in the previous frames. Visual object tracking can be employed in human activity recognition systems such as eating and drinking, by analyzing the trajectories of the employed auxiliary utensils, e.g., glass, spoons, forks, as shown in Fig. 7. A drinking activity can be recognized by the trajectory of the glass or by the distance between the glass and the face, as shown in Figs. 7–9. In eating activity recognition, the tracked object can be other kitchen utensils (e.g., fork, knife, and spoon) or bare human hands.

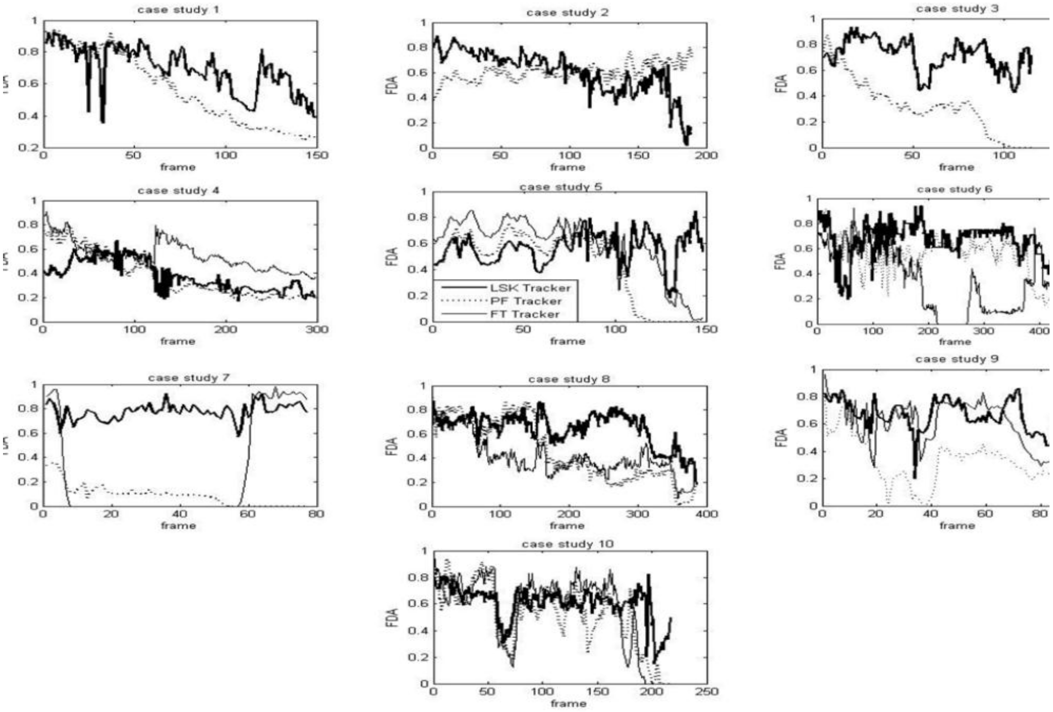


Figure3(b). Frame detection accuracy of the proposed CH-LSK tracker, the particle filter tracker and the fragments-based tracker for the videos in case studies 1-10.

In the following experiments, we test the performance of the algorithm in tracking objects which can be used in this framework. The test videos were recorded in AIIA laboratory and are included in the MOBISERV-AIIA eating and drinking activity database, which is employed in a nutrition support system designed to prevent dehydration and underfeeding of patients suffering from dementia’s) Case studies 7–9: In these case studies, we compare the performance of the three trackers when tracking a glass or hands during eating/drinking activity. The video depicts a person as he takes one sip from the glass. The experimental results are shown in Fig. 7. The PF tracker cannot keep up with the orientation change of the glass and loses track of it during sipping.

The FT tracker loses track of the glass when moving the glass up/down between the table and the mouth, but coincidentally finds the object when it is setback on the table, because its final position is very close to its original one. The CH-LSK tracker is successful in tracking the glass throughout the duration of the video. Furthermore, in this experiment, we test the performance of the three trackers in tracking human hands during eating. In the video hands.avi, the person cuts food with a knife and then eats with a fork. Generally, hand tracking is a difficult task, because hands are articulated objects and they constantly change shape. Fig. 8 shows the results of the three trackers. The PF tracker and the FT tracker keep tracking the right hand, but stop tracking the left hand, which performs more complicated movements.

The proposed tracker handles successfully the movement of both hands. In case study 9, we test the performance of the tracker in tracking a glass in an activity which is not drinking. In the video glass.avi the person enters the scene and sets the glass on the table. The glass of interest changes size and rotation and it is partially occluded by the hands. The experimental results are shown in Fig. 9. We notice that all trackers are successful in tracking part of the glass in the whole duration of the video. However, only the CH-LSK tracker is able to track the change in the object size. f) Case study 10: In this case study, we test the performance of the algorithm in a complex scenario for face tracking. In the video face.avi, a face constantly changes orientation and the hands occlude part of the face. The results are shown in Fig. 10. In the entire duration of the video, the CH-LSK tracker tracks the facial area better than either the PF tracker or the FT tracker, which drifts upward when the person lowers the head and stops tracking the head, when it is shifted to a profile view.

VI. Conclusion

In this paper, we proposed a new Bayesian filtering framework where analytic representations are used to approximate relevant density functions. Density approximation and interpolation technique are introduced for density propagation. Various simulations and tests on object tracking in real videos show the effectiveness of our density approximation methods and the kernel-based Bayesian filtering. By maintaining analytic representations of the density functions, we can sample in the state space more effectively and more efficiently. This advantage is significant for high dimensional problems. In addition, the approximation error can be monitored and analyzed. Our future work is focused on analyzing the approximation error in the posterior distribution and its propagation over time.

References

[1]. A. Cavallaro, O. Steiger, and T. Ebrahimi, “Tracking video objects in cluttered background,” IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 4, pp. 575–584, Apr. 2005.

[2]. D. Roller, K. Daniilidis, and H. H. Nagel, “Model-based object tracking in monocular image sequences of road traffic scenes,” Int. J. Comput. Vision, vol. 10, pp. 257–281, Mar. 1993.

[3]. C. Yang, R. Duraiswami, and L. Davis, “Efficient mean-shift tracking via a new similarity measure,” in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn., vol. 1, Jun. 2005, pp. 176–183.

[4] A. Yilmaz, X. Li, and M. Shah, “Contour-based object tracking with occlusion handling in video acquired using mobile cameras,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 11, pp. 1531–1536, Nov.2004.

[5] Y. Wang and O. Lee, “Active mesh—a feature seeking and tracking image sequence representation scheme,” IEEE Trans. Image Process., vol. 3, no. 5, pp. 610–624, Sep. 1994.

[6] L. Fan, M. Riihimaki, and I. Kunttu, “A feature-based object tracking approach for realtime image processing on mobile devices,” in Proc. 17th IEEE ICIP, Sep. 2010, pp. 3921–3924.

[7] L.-Q. Xu and P. Puig, “A hybrid blob- and appearance-based framework for multi-object tracking through complex occlusions,” in Proc. 2nd Joint IEEE Int. Workshop Visual Surveillance Perform. Evaluation Tracking Surveillance, Oct. 2005, pp. 73–80.

[8] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Real-time foreground-background segmentation using codebook model,” Real- Time Imaging, vol. 11, pp. 172–185, Jun. 2005.

[9] M. Piccardi, “Background subtraction techniques: A review,” in Proc. IEEE Int. Conf. Syst., Man Cybern., vol. 4, Oct. 2004, pp. 3099–3104.

[10] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 5, pp. 564–577, May 2003.

[11] S. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogni., vol. 2. Jun. 2005, pp. 1158–1163.

[12] D. Comaniciu and P. Mee, “Mean shift: A robust approach toward feature space analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603–619, May 2002.

[13] A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments-based tracking using the integral histogram,” in Proc. IEEE Conf. CVPR, Sep. 2006,pp. 798–805.