

Research Article

An Adaptive Object Tracking Using Kalman Filter and Probability Product Kernel

Hamd Ait Abdelali, Fedwa Essannouni, Leila Essannouni, and Driss Aboutajdine

Faculty of Sciences Rabat GSCM-LRIT Laboratory Associate Unit to CNRST (URAC 29), Mohammed V University, BP 1014, Rabat, Morocco

Correspondence should be addressed to Hamd Ait Abdelali; hamd.abdelali@gmail.com

Received 13 September 2015; Revised 14 January 2016; Accepted 16 February 2016

Academic Editor: Aiguo Song

Copyright © 2016 Hamd Ait Abdelali et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a new method for object tracking; we use an efficient local search scheme based on the Kalman filter and the probability product kernel (KFPPK) to find the image region with a histogram most similar to the histogram of the tracked target. Experimental results verify the effectiveness of this proposed system.

1. Introduction

Real time object tracking is one of the most active research areas in computer vision. The goal of object tracking is to estimate the locations and motion parameters of the target in a video sequence given the initialized position in the first frame. Research in tracking plays a key role in understanding motion and structure of objects. It finds numerous applications including surveillance [1], human-computer interaction [2], traffic pattern analysis [3], recognition [4], and medical image processing [5], to name a few. Although object tracking has been studied for several decades and numerous tracking algorithms have been proposed for different tasks, it remains to be a very challenging problem. One of the most challenging factors in object tracking is to account for appearance variation of the target object caused by change of illumination, deformation, and pose. In addition, occlusion, motion blur, and camera view angle also pose significant difficulties for algorithms to track target objects. Furthermore, there exists no single tracking method that can be successfully applied to all tasks and situations.

In this paper we present a new method for real time tracking of complex nonrigid objects. This new method successfully coped with camera motion, partial occlusions, and target scale variations. The scale/shape of the object is

approximated by an ellipse and its appearance by histogram-based features derived from local image properties. We use an efficient search scheme (Kalman filter [6, 7], using the probability product kernels [8–11] and integral image [12] as a similarity measure) to find the image region with a histogram most similar to the target of the object tracker. In this work, we address the problem of scale/shape adaptation and orientation changes of the target. The proposed approach (KFPPK) is compared with MKF [13]. In [13] the authors build a color histogram-based visual representation regularized by a spatially smooth isotropic kernel. Using the Bhattacharyya kernel [8] as the similarity measure, a mean shift procedure is performed for object localization by finding the basin of attraction of the local maxima. However, the tracker MKF only considers color information and therefore ignores other useful information such as target scale variations, resulting in the sensitivity to background clutters and occlusions. Extensive experiments are performed to testify the proposed method and validate its robustness and effectiveness to track the scale and orientation changes of the target in real time.

The rest of the paper is organized as follows: Section 2 presents the probability product kernels. Section 3 introduces basic Kalman filter for object tracking. Section 4 presents scale estimation. Section 5 presents the proposed approach and Section 6 the experiment result. Section 7 concludes the paper.

2. Probability Product Kernel

In this paper we define a class of kernels $K_\rho : P \times P \rightarrow \mathbb{R}$ on the space of normalized discrete distributions over some index set Ω . Specifically, we define the general probability product kernel [8, 9] between distributions p and q as

$$K_\rho(p, q) = \sum_k p(k)^\rho q(k)^\rho, \quad (1)$$

where ρ is a parameter and $p(k)$ and $q(k)$ are probability distributions defined on the space Ω , since, for any $p_1, p_2, p_3, \dots, p_N \in \Omega$, the Gram matrix K consisting of elements $K_{ij} = K_\rho(p_i, p_j)$ is positive semidefinite:

$$\sum_i \sum_j \alpha_i \alpha_j K_\rho(p_i, p_j) = \sum_k \left(\sum_i \alpha_i p_i(k)^\rho \right)^2 \geq 0, \quad (2)$$

for $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_N \in \mathbb{R}$. Different ρ values correspond to different types of probability product kernels. For $\rho = 1$, we have

$$K_1(p, q) = \sum_k p(k)q(k) = IE_p[q(k)] = IE_q[p(k)]. \quad (3)$$

We call this the probability product kernels defined by $K(p, q) = \sum_k p(k)q(k)$. We denote the histogram of target of object tracking T as h_T and the number of pixels inside T as $|T|$, which is also equal to the sum over bins $|T| = \sum_k h_T(k)$. Let q be the normalized version of h_T given by $q = h_T/|T|$, so we can consider q as a discrete distribution, with $\sum_k q(k) = 1$. Let p be the normalized histogram obtained in the frames of the video sequence. For the k -bin of h_T , its value is obtained by counting the pixels that are mapped to the index k :

$$h_T(k) = \sum_{x \in T} \delta[b(x) - k], \quad (4)$$

where $\delta[t]$ is the Kronecker delta with $\delta[t] = 1$ if $t = 0$ and $\delta[t] = 0$ otherwise. The mapping function $b(x)$ maps a pixel x to its corresponding bin index. The computation of the probability product kernels can be expressed as

$$\begin{aligned} K(p, q) &= \sum_k p(k)q(k) \\ &= \sum_k p(k) \left(\frac{1}{|T|} \sum_{x \in T} \delta[b(x) - k] \right) \\ &= \frac{1}{|T|} \sum_{x \in T} \sum_k p(k) \delta[b(x) - k] \\ &= \frac{1}{|T|} \sum_{x \in T} p(b(x)). \end{aligned} \quad (5)$$

Therefore, the computation of the probability product kernels can be done by taking the sum of values $p(b(x))$ within target T . The output of the following algorithm is a support map using integral image to compute the similarity measure between target and candidate region from each frame of the video sequence.

3. Kalman Filter

The Kalman filter has many uses, including applications in control [14], navigation [15], and computer vision. One important field of computer vision is the object tracking. Different movement conditions and occlusions can hinder the vision tracking of an object. In this paper, we present the use of the Kalman filter in the object tracking. We considered using the capacity of the Kalman filter to allow small occlusions and complex movements of objects [16]. Furthermore, the Kalman filter is a framework for predicting a process state and using measurements to correct or “update” these predictions.

3.1. State Prediction. For each time step k , a Kalman filter first makes a prediction \hat{s}_k of the state at this time step:

$$\hat{s}_k = A \times s_{k-1}, \quad (6)$$

where s_{k-1} is a vector representing process state at time $k-1$ and A is a process transition matrix. The Kalman filter concludes the state prediction steps by projecting estimate error covariance P_k^- forward one time step:

$$P_k^- = A \times P_{k-1} \times A^t + W, \quad (7)$$

where P_{k-1} is a matrix representing error covariance in the state prediction at time $k-1$ and W is the process noise covariance (or the uncertainty in our model of the process).

3.2. State Correction. After predicting the state \hat{s}_k (and its error covariance) at time k using the state prediction steps, the Kalman filter next uses measurements to “correct” its prediction during the measurement update steps.

First, the Kalman filter computes a Kalman gain K_k , which is later used to correct the state estimate \hat{s}_k :

$$K_k = P_k^- \times (P_k^- + R_k)^{-1}, \quad (8)$$

where R is measurement noise covariance. Determining R_k for a set of measurements is often difficult. In our contribution we calculated R dynamically from the measurement algorithms state.

Using Kalman gain K_k and measurements z_k from time step k , we can update the state estimate:

$$\hat{s}_k = \hat{s}_k + K_k \times (z_k - \hat{s}_k). \quad (9)$$

Conventionally, the measurements z_k are often derived from sensors. In our approach, measurements z_k are instead the output of the tracking algorithm given the same input: one frame of a streaming video and the most likely x and y coordinates of the target object in this frame (taking the first two dimensions of \hat{s}_k).

The final step of the Kalman filter iteration is to update the error covariance P_k^- into P_k :

$$P_k = (I - K_k) \times P_k^-. \quad (10)$$

The updated error covariance will be significantly decreased if the measurements are accurate (some entries in R_k are low) or only slightly decreased if the measurements are noisy (all of R_k is high).

4. Target Model Update

A target is represented by an ellipsoidal region in the image in order to eliminate the influence of different target dimensions [13]. We will denote a dataset of independent samples by $\chi = \{x_1, \dots, x_N\}$. Let us assume that the probability density function Gaussian $p(x) = \mathcal{N}(x_i, b(x_i), V)$ is a good generative model for our data [17]. Gaussian density function is proposed to automatically determine the number of density functions and their associated parameters, including the pixel locations x_i , covariance V , and location of the target $b(x_i)$. The advantages of using the covariance matrix representation are as follows:

- (i) It is robust to illumination changes, occlusion, and shape deformations.
- (ii) It can capture the intrinsic self-correlation properties of object appearance.
- (iii) It is low dimensional.

Those advantages lead to the computational efficiency:

$$V_0 = \sum_i (x_i - b(x_i))(x_i - b(x_i))^t. \quad (11)$$

We represent an object $O = [o_1, \dots, o_k]^t$ by modeling the spatial-color joint probability distribution of the corresponding region with a Gaussian density function. We define $b(x_i): \mathbb{R}^2 \rightarrow 1, \dots, K$ to be the function that assigns a color value of the pixel at location x_i . We employ the normalized RGB color space as the color feature in our case; we have used a normalized RGB histogram to provide some robustness to illumination change. We can write the features of spatial position and the color as

$$O_k = \sum_{i=1}^{N_{v_0}} \mathcal{N}(x_i, b(x_i), V_0) \delta[b(x_i) - k], \quad (12)$$

where δ is the Kronecker delta function. We use the Gaussian kernel N to rely more on the pixels in the middle of the object and to assign smaller weights to the less reliable pixels at the borders of the objects. We use only the N_{v_0} pixels from a finite neighborhood of the kernel and the pixels further.

The color histogram that describes the appearance of the region is $r_k(b(x), V)$ and the value of the k -bins is calculated by

$$r_k(b(x), V) = \sum_{i=1}^{N_v} \mathcal{N}(x_i, b(x_i), V) \delta[b(x_i) - k]. \quad (13)$$

The weights of the value of k -bins in object are calculated by

$$\omega_i = \sum_{k=1}^K \sqrt{\frac{O_k}{r_k(b(x), V)}} \delta[b(x_i) - k]. \quad (14)$$

The new distribution of color histogram p is calculated by

$$p_i = \frac{\omega_i \mathcal{N}(x_i, b(x_i), V)}{\sum_{i=1}^N \mathcal{N}(x_i, b(x_i), V)}. \quad (15)$$

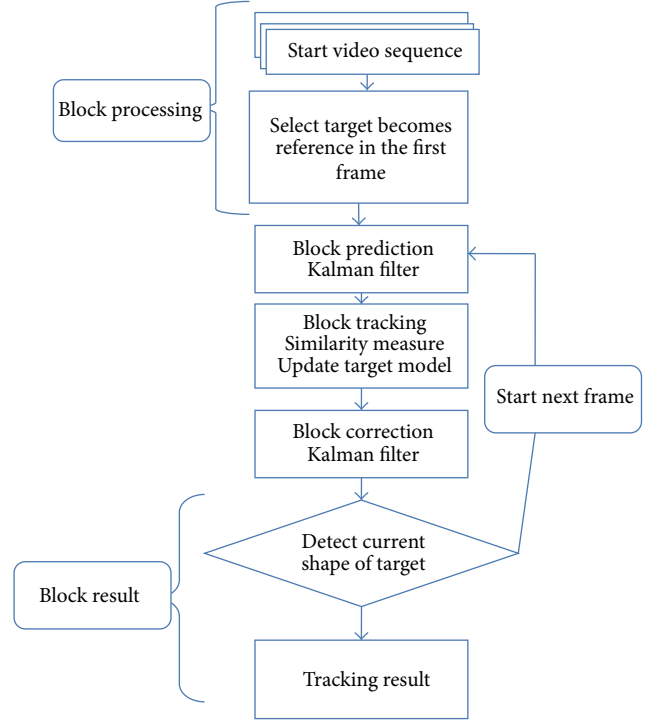


FIGURE 1: Flowchart for the proposed approach (KFPPK).

The covariance matrix update can be used to approximate the new shape of the object and it is calculated by

$$V = \beta \sum_{i=1}^N p_i (x_i, b(x_i))(x_i, b(x_i))^t, \quad (16)$$

and we should use $\beta = 1.5$. The correct value for β depends on the noise that is present in the image sequence.

5. Proposed Approach

To ensure good organization of the progress of the work, we used the benefits of modular design in our implemented approach using MATLAB. The goal of object tracking is to generate the trajectory of an object over time by discovering its exact position in every frame of the video sequence. The steps of the object tracking system are shown in Figure 1.

The proposed approach (KFPPK) for object tracking is composed of five blocks named block processing, block prediction, block tracking, block correction, and block result. The functions of these blocks are as follows.

Block Processing. The algorithm starts the video sequence and converts it into images processing for extracting color information of images and the target of the object tracking.

Block Prediction. This step attempts to evaluate how the state of the target will change by feeding it through a state prediction of the Kalman filter. The state prediction serves two purposes: the time update equations are responsible for projecting forward (in time) the current state and the error

covariance estimates for obtaining the a priori estimate for the next time step.

Block Tracking. In this block we use the probability product kernels combined with integral image as a similarity measure to find the image region with a histogram most similar to the target of the object tracker. Furthermore, we estimate the scale/shape and orientation of the object tracker using target model update.

Block Correction. The block correction update equations are responsible for the feedback. That is used for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be considered as predictor equations, while the measurement update equations can be thought of as corrector equations based on block tracking.

Block Result. Tracking trajectory of the object is done on the basis of the region properties of the object such as scale/shape and centroid.

The algorithm of the proposed approach (KFPPK) can be explained as follows.

Step 1. Start video sequence and select the target of the object tracker in the first frame.

Step 2. In this stage we use state prediction of Kalman filter to estimate how the state of a target will change by feeding it through a current state and error covariance estimates to obtain a priori estimate for the next time step. This step uses (1) and (2).

Step 3. Calculate the similarity measure ($p(b(x))$) between the target model and candidate regions using (5) and estimate the shape and orientation of the object tracker using (13).

Step 4. Correct and update equations into a priori estimate to obtain an improved a posteriori estimate, using (9) and (10).

Step 5. Draw trajectory. And go to Step 2 in the next frame.

6. Experimental Results

This section mainly shows the tracking results using the proposed approach (KFPPK). Experiments on real video sequences have been carried out to evaluate the performance of the proposed algorithm. In these experiments, we compared our system with existing algorithms MKF [13]. The proposed algorithm achieves good estimation accuracy of the scale and orientation of the object in the video sequence. We used different sequences; each has its own characteristics but the use of a single object in movement is a commonality between these different sequences. We then set up experiments to list the estimated width, height, trajectory, and orientation of the object.

In this work, we selected normalized RGB color space as the feature space and it was quantised into $16 \times 16 \times 16$ bins to compare between algorithms. One synthetic video sequence and one real video sequence are used in the experiments.

TABLE 1: The average time by different methods on the videos sequences.

Methods/sequences	KFPPK	MKF [13]
Ellipse (77 frames)	0.07 s	0.48 s
Occlusion (193 frames)	0.08 s	0.56 s

We first use a synthetic ellipse sequence (where the resolution is 352×240 , the frame rate is 25 fps, and the number of frames is 77) which is used to verify the efficiency of the KFPPK. As shown in Figure 2, the external ellipses represent the target candidate regions, which are used to estimate the real targets, that is, the inner ellipses. The experimental results show that the KFPPK is reliable for estimating the mean position and trajectory (as shown in Figure 3) of the ellipse with scale and orientation changes. Meanwhile, the results by the MKF [13] algorithm are not as performant as the proposed approach KFPPK because of significant scale and orientation changes of the object tracker.

The second test is an occlusion sequence (where the resolution is 320×240 , the frame rate is 25 fps, and there are 193 frames) which is used to verify the efficiency of the KFPPK on a more complex sequence. The object exhibits large scale changes with partial occlusion. We can see the results by the proposed approach (KFPPK) and the MKF [13] in Figure 4. We observe that the KFPPK works much better in estimating the scale and orientation of the target, especially when occlusion occurs.

Table 1 lists the average time on the videos sequences. We notice that our proposed approach (KFPPK) has a better average time of execution than MKF [13] algorithm.

6.1. Tracking Performance Evaluation. The main objective in this section is to evaluate the performance of our method using Receiver Operator Characteristic (ROC) curves. The ROC curves show how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. ROC curves are defined via two major rates expressed as follows:

$$\begin{aligned} \text{True Positive Rate} &= \frac{TP}{TP + FN}, \\ \text{False Positive Rate} &= \frac{FP}{FP + TN}, \end{aligned} \quad (17)$$

where TP (true positives) are examples correctly labeled as positives, FP (false positives) refer to negative examples incorrectly labeled as positive, TN (true negatives) correspond to negatives correctly labeled as negative, and FN (false negatives) refer to positive examples incorrectly labeled as negative.

We studied the ROC curves of the proposed algorithm and compared them with the MKF [13] algorithm. Figure 5 compares the performance of the algorithms using ROC curves for each video sequence. The KFPPK algorithm has a higher performance in regard to the MKF [13] algorithm.

The experimental results demonstrate that the KFPPK is robust in the tracking of the trajectory of objects in different

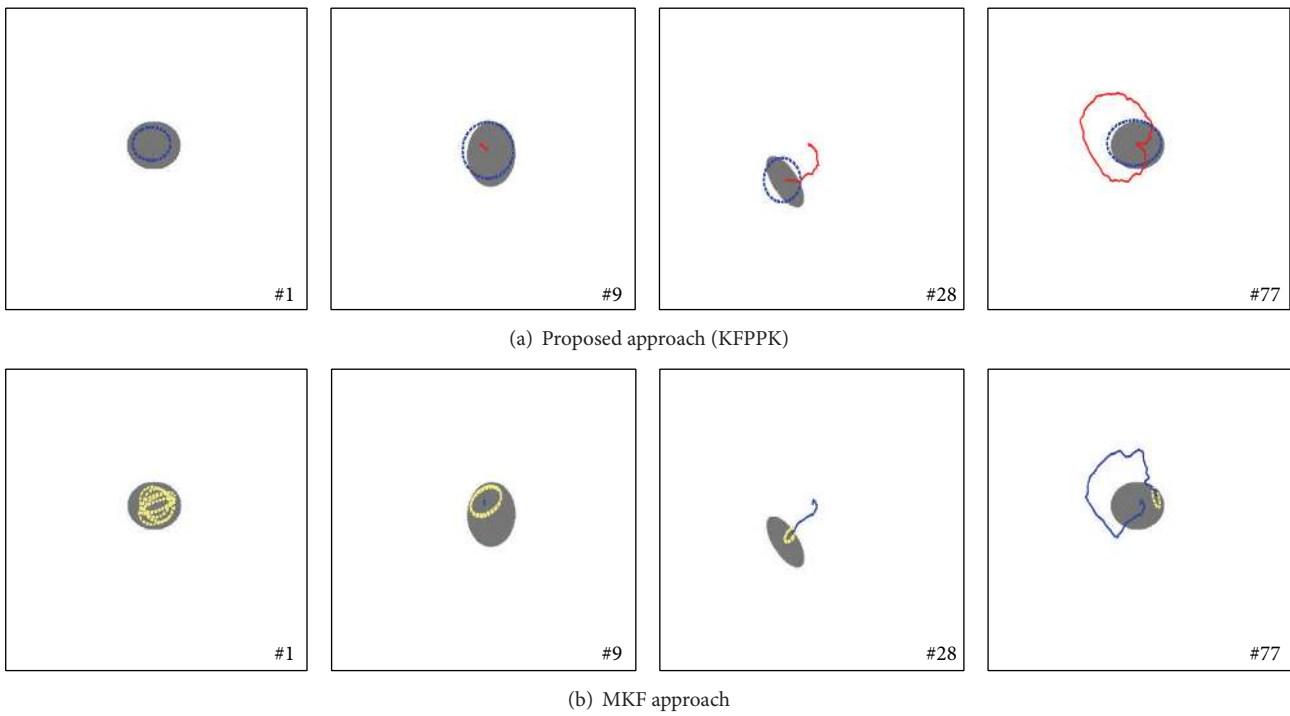


FIGURE 2: Tracking results of the synthetic ellipse sequence by different tracking algorithms. Frames 1, 9, 28, and 77 are displayed.

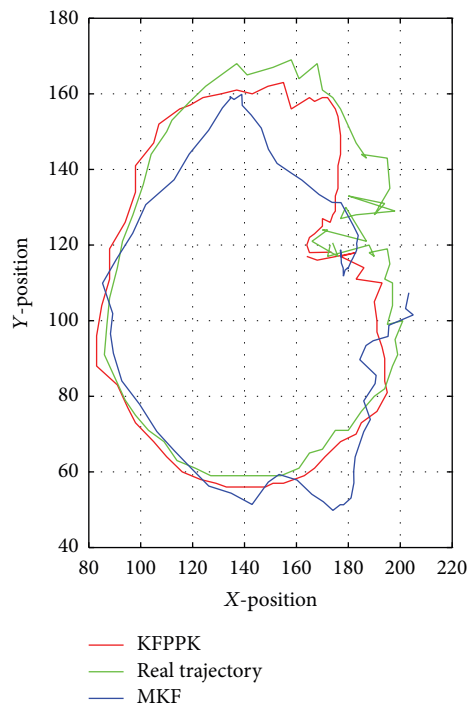


FIGURE 3: Trajectory results in the synthetic ellipse video sequence by the proposed approach KFPPK and the MKF tracking algorithms. We can see the KFPPK is closer to the real trajectory than the MKF tracking algorithm.



FIGURE 4: Tracking results of the occlusion sequence by different tracking algorithms. Frames 1, 31, 35, 44, 148, and 193 are displayed.

situations (scale variation, pose, rotation, and occlusion). It can be seen that the KFPPK achieves good estimation accuracy in real time of the scale and orientation of the target.

7. Conclusion

In this paper, the proposed approach (KFPPK) has been presented for tracking a single moving object in the video

sequence using color information. In this approach we combine the Kalman filter and probability product kernel as a similarity measure using integral image to compute the histograms of all possible target regions of object tracking in the video sequence. The new KFPPK has been compared with the state-of-the-art algorithm on a very large dataset of tracking sequences and it exceeds in performance in the processing speed. The extensive experiments are performed

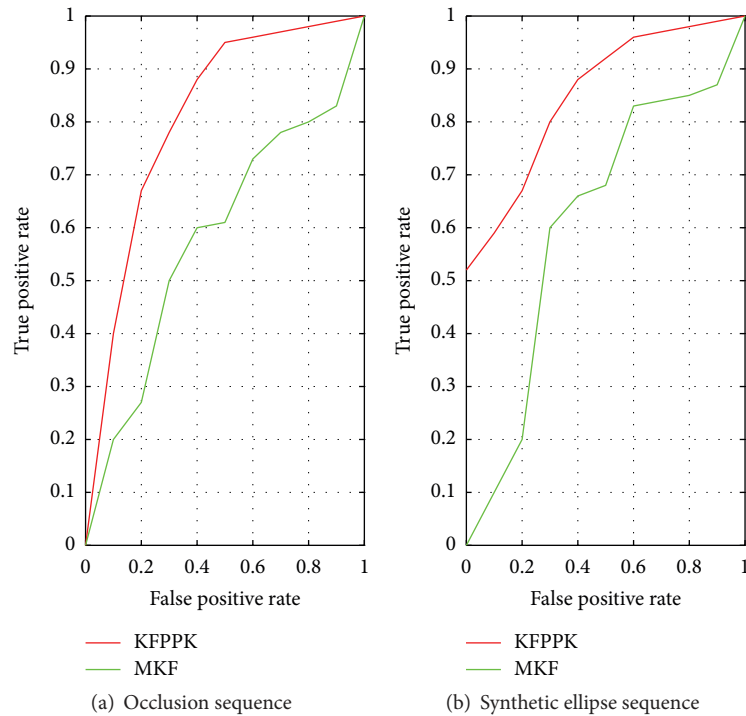


FIGURE 5: ROC curves by different algorithms on the different videos sequences.

to testify the KFPPK and validate its robustness to the scale and orientation changes of the target in real time. This implemented system can be applied to any computer vision application for moving object detection and tracking.

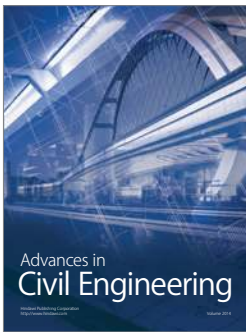
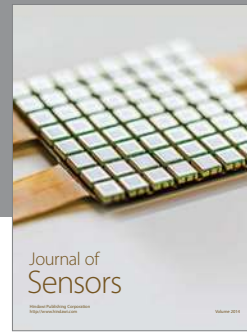
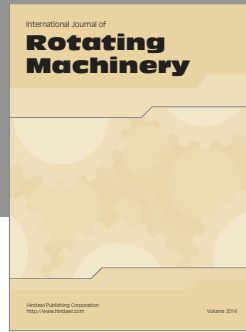
Conflict of Interests

The authors declare that they have no competing interests.

References

- [1] I. Haritaoglu, D. Harwood, and L. Davis, "W4s: a real-time system for detecting and tracking people," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 962–968, June 1998.
- [2] M. de La Gorce, N. Paragios, and D. J. Fleet, "Model-based hand tracking with texture, shading and self-occlusions," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, Anchorage, Alaska, USA, June 2008.
- [3] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, 2006.
- [4] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [5] X. S. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 115–129, 2005.
- [6] T. Jebara and R. Kondor, "Bhattacharyya and expected likelihood kernels," in *Learning Theory and Kernel Machines*, pp. 57–71, Springer, Berlin, Germany, 2003.
- [7] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *The Journal of Machine Learning Research*, vol. 5, pp. 819–844, 2004.
- [8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [9] A. Salhi and A. Y. Jammoussi, "Object tracking system using Camshift, Meanshift and Kalman filter," *World Academy of Science, Engineering & Technology*, vol. 64, pp. 674–679, 2012.
- [10] H. A. Abdelali, F. Essannouni, L. Essannouni, and D. Aboutajdine, "Algorithm for moving object detection and tracking in video sequence using color feature," in *Proceedings of the 2nd World Conference on Complex Systems (WCCS '14)*, pp. 690–693, Agadir, Morocco, November 2014.
- [11] H. A. Abdelali, F. Essannouni, L. Essannouni, and D. Aboutajdine, "A new moving object tracking method using particle filter and probability product kernel," in *Proceedings of the Intelligent Systems and Computer Vision (ISCV '15)*, pp. 1–6, IEEE, Fez, Morocco, March 2015.
- [12] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [13] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
- [14] M. Ghanai and K. Chafaa, *Kalman Filter in Control and Modeling*, InTech, 2009.
- [15] X. Ning and J. Fang, "An autonomous celestial navigation method for LEO satellite based on unscented Kalman filter and

- information fusion,” *Aerospace Science and Technology*, vol. 11, no. 2-3, pp. 222–228, 2007.
- [16] E. Cuevas, D. Zaldivar, and R. Rojas, “Vision tracking prediction,” in *Proceedings of the International Congress on Computer Sciences, Biomedical, Engineering and Electronics (CONCIBE '05)*, Guadalajara, México, October 2005.
- [17] H. Wang, D. Suter, K. Schindler, and C. Shen, “Adaptive object tracking based on an effective appearance filter,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1661–1667, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

