International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 19, No. 3 (2011) 547–558

DOI: 10.1142/S021848851100712X

COVARIANCE TRACKING WITH FORGETTING FACTOR AND RANDOM SAMPLING

XUGUANG ZHANG* and XIAOLI LI

Key Lab of Industrial Computer Control Engineering of Hebei Province,
Institute of Electrical Engineering, Yanshan University, P. R. China
xuguang.zhang78@yahoo.com.cn

MING LIANG

Department of Mechanical Engineering, University of Ottawa,
ON, K1N 6N5, Canada

YANJIE WANG

Changchun Institute of Optics, Fine Mechanics and Physics,
Chinese Academy of Sciences, P. R. China

Received 23 May 2010
Revised 10 December 2010

Covariance matching is an excellent algorithm of target tracking. In this paper, forgetting factor and random sampling methods are proposed to improve the robustness and efficiency of covariance tracking. First, a distance function between covariance matrixes is weighted by using a forgetting factor based on a fuzzy membership function to overcome the disturbances from similar targets. Then a random sampling method is applied to reduce the computing time in covariance matching and to facilitate real-time object tracking. Experiment results show that the algorithm proposed in this paper can effectively mitigate the clutter and occlusion problems at a high computing speed.

Keywords: Covariance tracking; forgetting factor; fuzzy membership function; random sampling.

1. Introduction

Covariance matching is an excellent algorithm of target tracking which can fuse multiple features and provide a global optimal resolution. [9] Target tracking is a key step in various computer vision applications, including surveillance,[1] video compression,[2] driver assistance,[3] perceptual user interfaces,[4] and so on. In most applications, the tracker should be robust to partial occlusions, clutters, changes of object scale and illumination. To meet the above needs, several algorithms have been developed such as contour-based tracking,[5] region-based tracking,[6] and texture-based tracking.[21] Most algorithms of the object tracking are based on a single cue. Thus they are restricted to particular environments such as those are static, controlled or known a priori. So, single cue based method may lost target under clutter background and wide variety of environmental conditions. Combination of multiple features could improve the robustness and generality of object tracking method.

To fuse multiple features for describing a target a natural method is using histograms based on a mean shift algorithm.[7,15] Unfortunately, the histogram-based methods possess several drawbacks:[14] (1) the computation time increases exponentially when fusing several different features through the histograms; (2) the histograms-based representation of target is a statistical model and as such, the spatial arrangement of the feature vector is neglected; and (3) the mean shift tracker requires the object kernels of more overlaps in two consecutive frames, which restrict the application of mean shift tracker.[9]

To overcome the difficulties of the histogram-based methods, a covariance matrix, called covariance descriptor, was proposed by Tuzel to describe a target.[8] Since then, this method has successfully been applied in object detection and object tracking.[16,9] The advantages of the covariance tracking include:[9] (1) it can fuse multiple features and modalities; (2) it is robust against noise and unstable illumination; and (3) it can find a global optimal solution in the matching region. However, the covariance descriptor based global matching can still be disturbed by similar targets and background. In addition, using this method to track a target with large size requires huge computation, making it is very difficult to track target in real-time. To reduce computing time of covariance tracking, an integral image formulation approach was proposed in 11. In this approach the covariance matrix of any candidate region can be obtained efficiently by using a previous iteration. It is noted that this approach can solve the problem of exhaustive search for a target but cannot improve the computing speed of covariance matrix of a candidate target. A random sampling method has been proposed to reduce the computing time of mean shift tracking.[12,13] In this method the computational complexity of mean shift tracking was reduced and the computing time was independent of the size of targets. Therefore, reducing the number of samples of a target, the computing time can be reduced in target tracking.

In this paper, an improved covariance tracking method is proposed. First, an annular search strategy is applied to localize the target. Secondly, the distances between the covariance matrices of target and candidate regions are weighted to overcome the disturbance of a similar target. The weights are determined by a forgetting factor that is computed by a fuzzy function.[17,19] Finally, a random sampling method is applied to reduce the number of samples of a target as so to reduce the computing time of object tracking. The paper is organized as follows. In Section II, we first overview the covariance tracking, including target representation and distance measure of covariance matrices; then, the strategy of an annular search and the distributing weight based on a forgetting factor are presented, followed by the presentation of a random sampling procedure. Experimental results are shown in Section III. The conclusions are drawn in the last section.

2. Covariance Tracking

2.1. *Feature vector of a target*

In 8, the feature vector is formed by the position of pixel, color values and the derivatives of intensities. The covariance matrices that fuse the features above have shown an excellent performance in object matching. In this study, the feature vector was defined as a similar way. Consider a color image. The conventional but useful features at point (x,y) contain $R(x,y)$, $G(x,y)$, $B(x,y)$, $I(x,y)$, $I_x(x,y)$, $I_y(x,y)$ and $d(x,y)$. $R(x,y)$, $G(x,y)$, $B(x,y)$ are the color pixel values (red, green and blue), $I(x,y)$ is the intensity feature, $I_x(x,y)$ and $I_y(x,y)$ are the gradients of image I. $d(x,y)$ is a spatial feature given by $d(x,y)=\sqrt{(x'^2+y'^2)}$, where $(x',y')=(x-x_0,y-y_0)$ is the relative coordinate and (x_0,y_0) is the coordinate of the window center. As a result, a vector of seven features is constructed as

$$\mathbf{f}_k = \left[R(x,y) \quad G(x,y) \quad B(x,y) \quad d(x,y) \quad I(x,y) \quad I_x(x,y) \quad I_y(x,y)\right] . \tag{1}$$

2.2. *Construction of covariance matrix*

From M samples, we can extract n features ($f_1, f_2, \ldots, f_n$). The covariance between features i and j can be

$$c_{ij} = \frac{1}{M-1}\sum_{k=1}^{M}(f_{ik}-\bar{f}_i)(f_{jk}-\bar{f}_j) . \tag{2}$$

where f_{ik} and f_{jk} are the features of the kth sample, $\bar{f}_i$ and $\bar{f}_j$ are the averages of the features i and j respectively. Computing the covariance of every two features in the M samples leads to the following n by n covariance matrix

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} .$$

For an $M \times N$ rectangular region R of an image, we can extract the feature vector using formula (1). Then, the covariance matrix $\mathbf{C}_R$ can be computed by

$$\mathbf{C}_R = \frac{1}{MN}\sum_{k=1}^{MN}(\mathbf{f}_k - \boldsymbol{\mu}_R)(\mathbf{f}_k - \boldsymbol{\mu}_R)^T \tag{3}$$

where $\boldsymbol{\mu}_R$ is the average of vector corresponding to the features of the points in region R. The covariance matrix is a symmetric matrix with its diagonal entries representing the variances of the features and the non-diagonal entries representing the correlations of different features.[9]

2.3. *Distances between covariance matrices*

Given an object O and a frame I, the aim of object tracking is to locate O in the current frame. The object O can be represented by a covariance matrix of the features. In the current frame, a new region can be registered depending on a minimum distance of the covariance matrices between the object and the candidate regions. The distance between the covariance matrices can be obtained by a method proposed in 10. The sum of the squared logarithms of the generalized eigenvalues is defined to compute the distance between two covariance matrices. Accordingly the distance measure can be calculated by

$$\rho(\mathbf{C}_i,\mathbf{C}_j)=\sqrt{\sum_{k=1}^{n} ln^2\lambda_k(\mathbf{C}_i,\mathbf{C}_j)} \tag{4}$$

where $\{\lambda_k(\mathbf{C}_i,\mathbf{C}_j)\}$ are the generalized eigenvalues of $\mathbf{C}_i$ and $\mathbf{C}_j$, computed from $|\lambda\mathbf{C}_i-\mathbf{C}_j|=0$. The distance ρ contains the following properties

(1) positivity: $\rho(\mathbf{A},\mathbf{B})\geq 0$, and $\rho(\mathbf{A},\mathbf{B})=0$ only if $\mathbf{A}=\mathbf{B}$.

(2) symmetry: $\rho(\mathbf{A},\mathbf{B})=\rho(\mathbf{B},\mathbf{A})$.

(3) triangle inequality: $\rho(\mathbf{A},\mathbf{B})+\rho(\mathbf{A},\mathbf{C})\geq\rho(\mathbf{B},\mathbf{C})$.

Based on the distance measure, the candidate region may be considered as the tracking target if the distance between the object and this candidate region is the smallest.

2.4. *Forgetting factor for weight estimation*

In a task of target tracking, the presence of similar objects could result in a false localization. In the tradition tracking method, the searching window is from the lower left corner to upper right corner of the searching window. In this study, a new method is proposed to reduce the disturbance of similar targets. First, we search target according to an annular strategy. Just as Fig. 1 shows, we distribute different priorities to each candidate point by the distance between this position and the center of tracking window. Therefore, the candidate points closer to the window center will be searched earlier. Secondly, we impose different weights to candidate points by a forgetting factor. During the process of target tracking, the center of window represents the target location at the previous frame. Considering the displacement of a target is very small between two consecutive frames, a simple intuition can be occurred: the further a candidate position from the window center, the lower likelihood the position is a target (i.e., the larger forgetting to the previous frame for this position). Based on this priori intuition, we allocate different weights called forgetting factors to the candidate locations according to the distance between these locations and the window center. Therefore, we can weaken the disturbance by the false similar targets. Allocating a weight to each candidate point can be achieved by formula (5)

$$\rho' = F\rho \tag{5}$$

where F is the forgetting factor ($0 < F < 1$).[18] The forgetting factor of a candidate point is determined by the distance between the candidate point and the center position of the search window, i.e.,

$$F=\beta\cdot\mu_{\tilde{A}}(d)+1 \tag{6}$$

where β is a constant, d is the distance between the candidate point and the center of the search window, $\mu_{\tilde{A}}(d)$ is the membership degree [17] which can be computed by

$$\mu_{\tilde{A}}(d)=\begin{cases} 1 & if(d=0) \\ (d_{\max}-d)/d_{\max} & if(0<d<d_{\max}) \\ 0 & if(d=d_{\max}) \end{cases} \tag{7}$$

with $d_{\max}$= Max(d).

In this study, we specify $-1 \le \beta \le 0$ such that the forgetting factor F is limited in the range of [0, 1]. If the value of β is small (*e.g.*, –0.9), the forgetting effect is high and the disturbance from a similar target is small. Consequently, target matching has more chance to settle at the center of window, which is not suited to deal with the target moving at a high speed. If the value of β is large (such as –0.1), the forgetting effect diminishes and the disturbance from similar targets becomes more severe. This could cause faulty target matching. For the above reasons, β is set at –0.5 in our experiments to balance the above effects.

6	5	4	5	6
5	3	2	3	5
4	2	1	2	4
5	3	2	3	5
6	5	4	5	6

Fig. 1. Different priorities defined by annular search strategy.

2.5. *Covariance tracking with uniform random sampling*

To improve the efficiency of the covariance tracking, a random sampling method is proposed to reduce the computation of the covariance matrix in this study. The conventional method for computing a covariance matrix needs all the pixels of an image region R (target image or the candidate region) to participate in the calculation. The computation of the covariance matrix increases with the size of the target. As for random sampling, fewer pixels are picked up randomly from the image region for the computing of covariance matrix. For a pixel in an image region, whether it is selected for computing the covariance matrix follows a probability distribution. Therefore, we may use fewer samples to present a big target by using the random sampling method. If we fix the number of random samples the computational complexity of the covariance matrix would not be related to the object size. In this work, a simple uniform distribution is used to decide which pixels are selected as samples. For uniform distribution, the probability density function of random variable x on an interval (a, b) and elsewhere is

$$\varphi(x) = \begin{cases} \dfrac{1}{b-a} & a < x < b \\ 0 & other \end{cases} \tag{8}$$

In such a way, the probability of each pixel which is selected to calculate the covariance matrix is equal. During tracking, a fixed number of samples should be set.

Therefore, the computing speed is depended on the number of samples which is independent of the size of target.

3. Experiments and Results

The proposed method has been used to test several complex video sequences. Some interesting cases are presented. In all experiments, the feature vectors are calculated using Eq. (1) and the image size is 320×240 pixels.

Figure 2 shows some frames of four moving targets marked by red rectangles. For the *Black Car* target (the first row of the figure), both the camera and object are moving, the scale and appearance of the target change over time. In the *Lab* sequence (the second row of the figure), the target is non-rigid and the appearances of target change heavily over time. In the *Soccer* sequence (the third row of the figure), the target is non-rigid and jerky movements are included. In the *OneShopOneWait2cor* sequence (the last row of the figure, from CAVIAR database[21]), the target is also non-rigid and is blocked by another person. The proposed method can track the targets well in all the four cases, i.e., the *Black car*, *Lab, Soccer* and *OneShopOneWait2cor* sequences.

In Fig. 3 the *Intersection* sequence is shown here to evaluate the performance of forgetting factor. In this sequence, there are similar cars around the target, a white car. In order to show the robustness of forgetting factor, we did not use random sampling method in this sequence. The searching strategy without forgetting factor fails to track the target in this case and a wrong vehicle is tracked (see Fig. 3 (Top)). The new method proposed in this study can track the target correctly (see Fig. 3 (Bottom)). Here, the annular strategy was used to substitute for the traditional strategy because the result of traditional strategy is too poor to present. The tracking errors with the proposed strategy and the annular strategy without forgetting factor are plotted in Fig. 4. The ground truths are marked every five frames by hand. The annular strategy without forgetting factor lost target in frame 21, then tracking the wrong target through the sequence.

To test the efficiency of the random sampling, the *Walk2* sequence (Fig. 5) and *OneStopMoveEnter1cor* sequence (Fig. 6) (from CAVIAR database[21]) are employed. In order to show the effect of random sampling, we tracked targets without using forgetting factor in these sequences. In these experiments, 10, 20, 50, 100 samples are randomly extracted for building the model of the candidate and target images. The tracker loses the target for the case of 10 samples. With 20 samples, the tracker can track the target successfully but the precision is low. When 50 samples are used, the stability of the tracker is improved in comparison with the case of 20 samples. When the number of samples increases to 100, stable target tracking can be achieved. In order to testify the effect of random sampling, the tracking error of *Walk2* and *OneStopMoveEnter1cor* sequences are showed in Fig. 7 and Fig. 8. Here we use the tracking error of full sampling as the ground truth. The curves of tracking errors show that the tracking accuracy will increase with the number of samples. Comparing with full sampling the tracking errors are about 10 pixels by using 100 samples.

Fig. 2. The test results of proposed method in four different sequences. In the *Black Car* sequence, the scale and appearance of the object change over time due to the moving camera and object. In the *Lab*, *Soccer* and *OneShopOneWait2cor* sequences, the objects are non-rigid. The appearances of the target change drastically in the *Lab* sequence. The object is blocked in the *OneShopOneWait2cor* sequence.

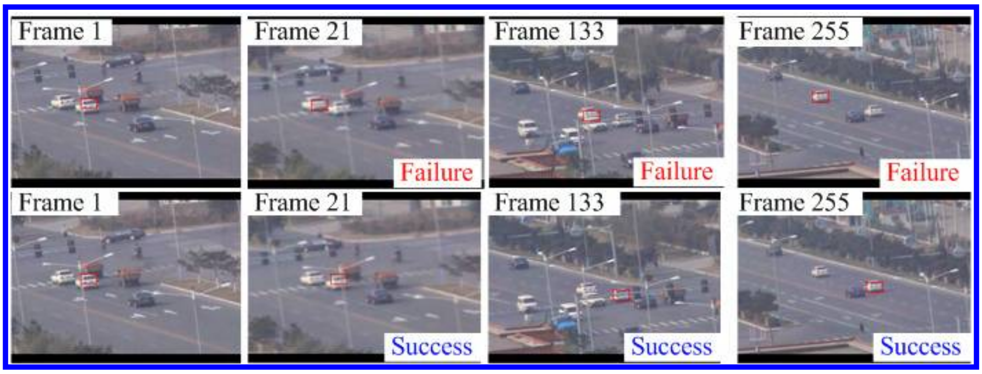

Fig. 3. The test result of the *Intersection* sequence. Upper, the result obtained with the annular strategy (without forgetting factor); bottom, the result with weight of a forgetting factor.

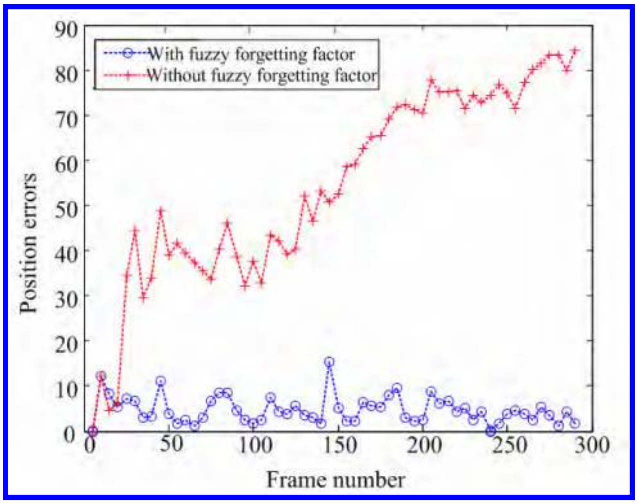

Fig. 4. Tracking errors comparison of proposed strategy (weighted by forgetting factor) and annular strategy without forgetting factor for the Intersection sequence.

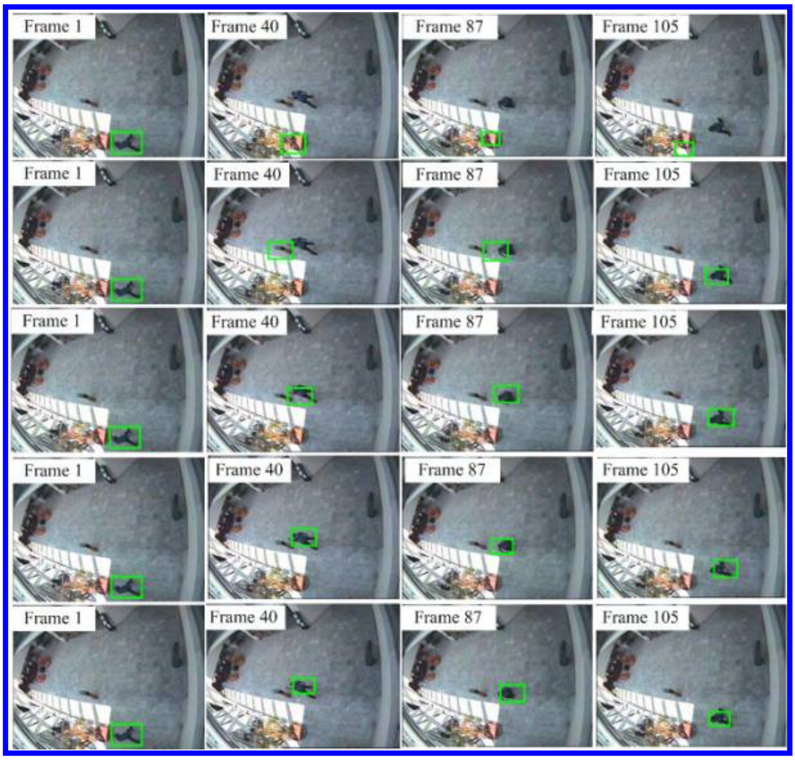

Fig. 5. Covariance tracking for *Walk2* sequence with random sampling. First row: only 10 random samples are picked from each of the candidate and model images, 20 samples in the second row, 50 samples in the third row , 100 samples in the fourth row and full samples in the last row.

Fig. 6. Covariance tracking for *OneStopMoveEnter1cor* sequence with random sampling. First row: only 10 random samples are picked from each of the candidate and model images, 20 samples in the second row, 50 samples in the third row , 100 samples in the fourth row and full samples in the last row.

All the above tests were performed on a 2.8GHz PC. The program was coded using Visual C++. The computing times of the employed method are listed in Table 1. From the table, we can see that the computing time with random sampling is only 40 ms for the target of size 64×64 pixels comparing to 138 ms for the same target without random sampling. It should be noted that in this test the computing time of covariance matrix is not related to the target size due to the application of the random sampling. In particular, in order to improve the readability of the program some operations such as transferring the images of candidate regions in the memory were performed. Therefore the processing time increases slightly with the size of target in our experiments.

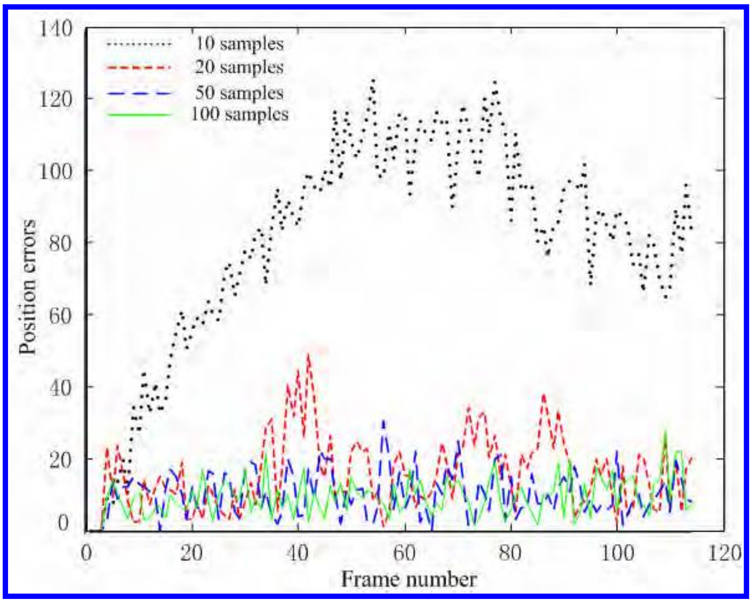

Fig. 7. Tracking errors comparison of different random samples of *Walk2* sequence.

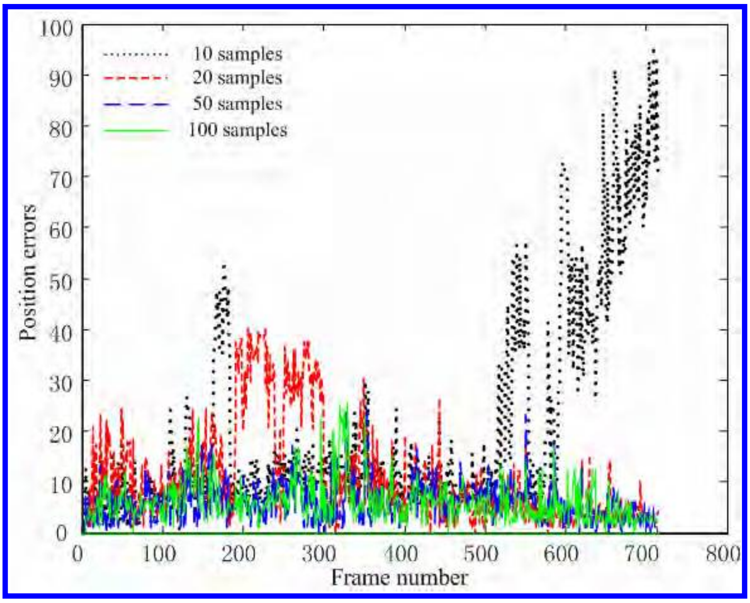

Fig. 8. Tracking errors comparison of different random samples of *OneStopMoveEnter1cor* sequence.

Table 1. Computing times of covariance tracking with or without random sampling.

TARGET SIZE	WITHOUT RANDOM SAMPLING	WITH RANDOM SAMPLING
12×12 pixels	15ms	15ms
12×24 pixels	22ms	21ms
24×24 pixels	33ms	22ms
36×24 pixels	36ms	23ms
36×36 pixels	50ms	24ms
48×36 pixels	63ms	26ms
64×36 pixels	82ms	30ms
64×64 pixels	138ms	40ms

4. Conclusion

In this study, the forgetting factor and random sampling are used to improve the performance of covariance tracking. The efficiency and robustness of this method are tested by some practical cases. The different weights determined by the distances between the candidate points and the center position of the searching window can reduce the disturbance derived from other similar targets. The random sampling can enhance the processing speed of the covariance tracking significantly , which allow this method to track large target in real time systems.

Acknowledgments

This work was supported by the project of National Natural Science Foundation of China (No. 60805045) and National Science Fund for Distinguished Young Scholars of China (No. 61025019). Authors would like to thank Prof. F. Porikli for the discussion on the covariance tracking with random sampling.

References

1. V. Kettnaker and R. Zabih, Bayesian multi-camera surveillance, *IEEE Conf. Computer Vision and Pattern Recognition* (1999) 253−259.
2. D. Bue, D. Comaniciu, V. Ramesh and C. Regazzoni, Smart cameras with real-time video object generation, *IEEE Conf. Image Processing,* Vol. III (2002), pp. 429−432.
3. U. Handmann, T. Kalinke, C. Tzomakas, M. Werner and W. Von Seelen, Computer vision for driver assistance systems, *SPIE*, *Session Enhanced and Synthetic Vision* 3364 (1998) 136−147.
4. G. R. Bradski, Computer vision face tracking as a component of a perceptual user interface, *IEEE Workshop Applications of Computer Vision* (1998) 214−219.
5. J. C. Nascimento and J. S. Marques, Robust shape tracking in the presence of cluttered background. *IEEE Transactions on Multimedia* **6**(6) (2004) 852−861.
6. T. Liu and H. Chen. Real-time tracking using trust-region methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(3) (2004) 397−402.

7. D. Comaniciu, V. Ramesh and P. Meer, Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **25** (2003) 564–577.
8. O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. *European Conf. Computer Vision,* Graz, Austria, **2** (2006) 589−600.
9. F. Porikli, O. Tuzel, P. Meer. Covariance tracking using model update based on Lie algebra. *IEEE Conf. Computer Vision and Pattern Recognition*, New York, **1** (2006) 728−735.
10. W. Förstner and B. Moonen. A metric for covariance matrices. *Technical report, Dept. of Geodesy and Geoinformatics* (Stuttgart University, 1999).
11. F. Porikli and O. Tuzel, Fast construction of covariance matrices for arbitrary size image windows, *IEEE Conf. Image Processing* (2006) 1581−1584.
12. A. P. Leung and S. Gong, Mean shift tracking with random sampling, *British Machine Vision Conference* (2006) 729−738.
13. P. Leung and S. Gong, Optimizing distribution-based matching by random subsampling. *IEEE Conf. Computer Vision and Pattern Recognition* (2007) 1−8.
14. F. Porikli, O. Tuzel and P. Meer, Covariance tracking using model update based means on Riemannian manifolds, *IEEE Conf. Computer Vision and Pattern Recognition*, New York, NY, 2007.
15. R. T. Collins, Mean-shift blob tracking through scale space, *IEEE Conf. Computer Vision and Pattern Recognition* (2003) 234–240.
16. F. Porikli and T. Kocak, Robust license plate detection using covariance descriptor in a neural network framework, *IEEE Conf. Advanced Video and Signal Based Surveilance* **107** (2006).
17. L. A. Zadeh, Fuzzy sets, *Information and Control* **8** (1965) 338−353.
18. G. Kim and N. I. Cho, Time varying forgetting factor for the noise estimation in multi-channel noise reduction. *Applied Acoustics* **69**(8) (2008) 751−754.
19. H. Liu, A Fuzzy Qualitative framework for connecting robot qualitative and quantitative representations, *IEEE Transactions on Fuzzy Systems* **16**(6) (2008) 1522−1530.
20. Caviar Dataset: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.
21. J. Caban, A. Joshi and P. Rheingans, Texture-based feature tracking for effective time-varying data visualization, *IEEE Transactions on Visualization and Computer Graphics* **13(**6) (2007).