Visual Tracking by Adaptive Kalman Filtering and Mean Shift

Vasileios Karavasilis, Christophoros Nikou, and Aristidis Likas

Department of Computer Science, University of Ioannina,
PO Box1186, 45110 Ioannina, Greece
Ph.: + (30) 26510 08802
{vkaravas,cnikou,arly}@cs.uoi.gr

Abstract. A method for object tracking combining the accuracy of mean shift with the robustness to occlusion of Kalman filtering is proposed. At first, an estimation of the object's position is obtained by the mean shift tracking algorithm and it is treated as the observation for a Kalman filter. Moreover, we propose a dynamic scheme for the Kalman filter as the elements of its state matrix are updated on-line depending on a measure evaluating the quality of the observation. According to this measure, if the target is not occluded the observation contributes to the update equations of the Kalman filter state matrix. Otherwise, the observation is not taken into consideration. Experimental results show significant improvement with respect to the standard mean shift method both in terms of accuracy and execution time.

Keywords: Visual tracking, Kalman filter, mean shift algorithm.

1 Introduction

Tracking is the procedure of generating an inference about motion given a sequence of images. Based on a set of measurements in image frames the object's true position should be estimated. Tracking algorithms may be classified in two categories [1]. The first category is based on filtering and data association, while the second family of methods relies on target representation and localization.

The algorithms based on filtering assume that the moving object has an internal state which may be measured and, by combining the measurements with the model of state evolution, the object's true position is estimated. The first method of that category is the Kalman filter which successfully tracks objects even in the case of occlusion if the assumed type of motion is correctly modeled [2]. This category also includes Condensation [3] algorithm which is more general than Kalman filters, as it does not assume specific type of densities and, using factored sampling, have the ability to predict an object's location under occlusion as well.

On the other hand, tracking algorithms relying on target representation and localization employ a probabilistic model of the object appearance and try to detect this model in consecutive frames of the image sequence. More specifically,

S. Konstantopoulos et al. (Eds.): SETN 2010, LNAI 6040, pp. 153–162, 2010.

color or texture features of the object, masked by an isotropic kernel, are used to create a histogram. Then, the object's position is estimated by minimizing a cost function between the model's histogram and candidate histograms in the next image. A representative method in this category is the mean shift algorithm [1]. Other approaches using multiple kernels [4], Earth Mover's Distance [5] and a Newton style optimization procedure [6] were also proposed.

Combination of Kalman filtering with mean shift have also being proposed in [7,8,9,10,11]. Other works track many objects simultaneously [12]. A Gaussian mixture model (GMM) was used in [13] to represent the object in a joint spatial-color space. Furthermore, the object is represented by its contour [14] and multiple object representations were also combined to make the tracking procedure more robust [15].

In this paper we propose to consider the estimated location of the target obtained by mean shift, as a measurement (observation) of a time-varying Kalman filter in order to address cases presenting occlusions or abrupt motion changes. Hence, the prediction for the object's location is forwarded to a Kalman filter whose state matrix parameters are not constant but they are updated on-line based on recent history of the estimated motion.

The remainder of the paper is organized as follows: in sections 2 and 3, the mean shift algorithm and Kalman filter are respectively reviewed. In section 4, the combination of the algorithms in order to address the problem of occlusion is described. Experimental results are shown in section 5 which are followed by our conclusion in section 6.

2 Background on Mean Shift Tracker

The mean shift [1] is an algorithm trying to locate the object by finding the local maximum of a function. The object target pdf is approximated by a histogram of m bins $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\ldots m}$, $\sum_{u=1}^{m} \hat{q}_u = 1$, with $\hat{q}_u$ being the u-th bin. To form the histogram, only the pixels inside an ellipse surrounding the object are taken into account. The center of the ellipse is assumed to be at the origin of the axes. Due to the fact that the ellipse contains both object pixels and background pixels a kernel with profile $k(x)$, $k : [0, \infty) \rightarrow \Re$ is applied to every pixel to make pixels near the center of the ellipse to be considered more important. To reduce the influence of different length of the ellipse axes on the weights, the pixel locations are normalized by dividing the pixel's coordinates with the ellipse's semi-axes dimension h_x and h_y. Let $\{\mathbf{x}_i^*\}_{i=1\ldots n}$ be the normalized pixel's spatial location. The u-th histogram bin is given by

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|\mathbf{x}_i^*\|^2)\delta[b(\mathbf{x}_i^*) - u] \tag{1}$$

where $b : \Re^2 \rightarrow \{1 \ldots m\}$ associates each pixel with each bin in the quantized feature space, δ is the Kronecker delta function and C is a normalization factor such as $\sum_{u=1}^{m} \hat{q}_u = 1$.

In the next image, the object candidate is inside the same ellipse with its center at the normalized spatial location $\mathbf{y}$. Let $\{\mathbf{x}_i\}_{1\ldots n}$ be the normalized pixel coordinates inside the target candidate ellipse. The pdf of the target candidate is also approximated by an m-bin histogram $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\ldots m}$, $\sum_{u=1}^{m} \hat{p}_u(\mathbf{y}) = 1$, with each histogram bin given by

$$\hat{p}_u(\mathbf{y}) = C_c \sum_{i=1}^{n} k\left(\|\mathbf{y} - \mathbf{x}_i\|^2\right) \delta[b(\mathbf{x}_i) - u] \tag{2}$$

where C_c is a normalization factor such as $\sum_{u=1}^{m} \hat{p}_u(\mathbf{y}) = 1$.

The distance between $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}(\mathbf{y})$ is defined as:

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]} \tag{3}$$

where

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u} \tag{4}$$

is the similarity function between $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}(\mathbf{y})$, called Bhattacharyya coefficient.

To locate the object correctly in the image, the distance in (3) must be minimized, which is equivalent to maximize (4). The ellipse center is initialized at a location $\hat{\mathbf{y}}_0$ which is the ellipse center in the previous image frame. The probabilities $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$ are computed and using linear Taylor approximation of (4) around these values:

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{C_c}{2} \sum_{u=1}^{n} w_i k\left(\|\mathbf{y} - \mathbf{x}_i\|^2\right), \tag{5}$$

where

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}} \delta[b(\mathbf{x}_i) - u]. \tag{6}$$

As the first term of (5) is independent of $\mathbf{y}$, the second term of (5) must be maximized. The maximization of this term may be accomplished by employing

Algorithm 1. Maximizing Bhattacharyya coefficient $\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]$

Input: The target model $\{\hat{q}_u\}_{u=1\ldots m}$ and its location $\hat{\mathbf{y}}_0$ in the previous frame.

1. Initialize the center of the ellipse in the current frame at $\hat{\mathbf{y}}_0$,
compute $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$ using (2) and evaluate $\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_0), \hat{\mathbf{q}}]$ using (4).
2. Compute the weights $\{w_i\}_{i=1\ldots n}$ according to (6).
3. Compute the next location of the target candidate according to (7).
4. Compute $\{\hat{p}_u(\hat{\mathbf{y}}_1)\}_{u=1\ldots m}$ using (2) and evaluate $\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1), \hat{\mathbf{q}}]$ using (4).
5. If $\|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_0\| < \epsilon$ Stop.
Otherwise set $\hat{\mathbf{y}}_0 \leftarrow \hat{\mathbf{y}}_1$ and go to Step 2.

the mean shift algorithm [1], which yields the following update:

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n} \mathbf{x}_i w_i g\left(\|\hat{\mathbf{y}}_0 - \mathbf{x}_i\|^2\right)}{\sum_{i=1}^{n} w_i g\left(\|\hat{\mathbf{y}}_0 - \mathbf{x}_i\|^2\right)}, \tag{7}$$

where $g(x) = -k'(x)$ and $k(x)$ is kernel with Epanechnikov profile. The complete algorithm [1] is summarized in algorithm 1.

3 Kalman Filter

In general, we assume that there is a linear process governed by an unknown inner state producing a set of measurements. More specifically, there is a discrete time system and its state at time n is given by vector $\mathbf{x}_n$. The state in the next time step $n+1$ is given by

$$\mathbf{x}_{n+1} = \mathbf{F}_{n+1,n}\mathbf{x}_n + \mathbf{w}_{n+1} \tag{8}$$

where $\mathbf{F}_{n+1,n}$ is the transition matrix from state $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$ and $\mathbf{w}_{n+1}$ is white Gaussian noise with zero mean and covariance matrix $\mathbf{Q}_{n+1}$.

The measurement vector $\mathbf{z}_{n+1}$ is given by

$$\mathbf{z}_{n+1} = \mathbf{H}_{n+1}\mathbf{x}_{n+1} + \mathbf{v}_{n+1} \tag{9}$$

where $\mathbf{H}_{n+1}$ is the measurement matrix and $\mathbf{v}_{n+1}$ is white Gaussian noise with zero mean and covariance matrix $\mathbf{R}_{n+1}$.

In equation (9), the measurement $\mathbf{z}_{n+1}$ depends only on the current state $\mathbf{x}_{n+1}$ and the noise vector $\mathbf{v}_{n+1}$ is independent of the noise $\mathbf{w}_{n+1}$.

Kalman filter computes the minimum mean-square error estimate of the state $\mathbf{x}_k$ given the measurements $\mathbf{z}_1, \ldots, \mathbf{z}_k$. The solution is a recursive procedure [2], which is described in algorithm 2.

Algorithm 2. Kalman filter

---

1 Initialization:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0],$$
$$\mathbf{P}_0 = E[(\mathbf{x}_0 - E[\mathbf{x}_0])(\mathbf{x}_0 - E[\mathbf{x}_0])^T].$$

2 Prediction:

$$\hat{\mathbf{x}}_n^- = \mathbf{F}_{n,n-1}\hat{\mathbf{x}}_{n-1},$$
$$\mathbf{P}_n^- = \mathbf{F}_{n,n-1}\mathbf{P}_{n-1}\mathbf{F}_{n,n-1}^T + \mathbf{Q}_n,$$
$$\mathbf{G}_n = \mathbf{P}_n^-\mathbf{H}_n^T[\mathbf{H}_n\mathbf{P}_n^-\mathbf{H}_n^T + \mathbf{R}_n]^{-1}.$$

3 Estimation:

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \mathbf{G}_n(\mathbf{z}_n - \mathbf{H}_n\hat{\mathbf{x}}_n^-),$$
$$\mathbf{P}_n = (\mathbf{I} - \mathbf{G}_n\mathbf{H}_n)\mathbf{P}_n^-.$$

Goto the Prediction step for the next prediction.

---

4 The Proposed Method

The main idea is to find the position of the object with algorithm 1 (considered as the measurement or the observation in Kalman filter terminology) and forward it to algorithm 2 to obtain the current position of the object (estimation). Moreover, in this section, we propose a dynamic scheme for the Kalman filter as the elements of its state matrix are updated on-line depending on a measure evaluating the quality of the observation. By these means the tracking procedure may be significantly accelerated.

We assume that the object is described by its center coordinates (x, y), the ellipse axes are (h_x, h_y) and that the size of the ellipse does not change through time. The state vector $\mathbf{x}_n = [x_n, y_n, 1]^T$ denotes the true position of the center in the image in homogenous coordinates (x_n and y_n are the horizontal and vertical coordinate respectively) and its position varies over time by equation (8). Matrix $\mathbf{F}_{n+1,n}$ is defined as:

$$\mathbf{F}_{n+1,n} = \begin{bmatrix} 1 & 0 & dx_{n+1,n} \\ 0 & 1 & dy_{n+1,n} \\ 0 & 0 & 1 \end{bmatrix}$$

where $dx_{n+1,n}$ $dy_{n+1,n}$ are the horizontal and vertical translations of the object's center. Parameters $dx_{n+1,n}$ $dy_{n+1,n}$ are not constant in time (figure 1), but they are computed dynamically as it will be explained in what follows. The noise vector $\mathbf{w}_{n+1} = [w_{n+1_x}, w_{n+1_y}, 1]^T$ has covariance matrix

$$\mathbf{Q} = \begin{bmatrix} \sigma_{Q_x} & 0 & 0 \\ 0 & \sigma_{Q_y} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where $\sigma_{Q_x} = h_x$ and $\sigma_{Q_y} = h_y$. This means that the assumed noise perturbates the object center inside the ellipse.

We employ algorithm 1 to obtain the measurement vector $\mathbf{z}_{n+1} = [x'_{n+1}, y'_{n+1}]^T$ where x'_{n+1} and y'_{n+1} are the horizontal and vertical coordinates of the ellipse center. In general, these measurements differ from the state variables x_{n+1} and y_{n+1} of vector $\mathbf{x}_{n+1}$ due to the presence of noise $\mathbf{v}_{n+1}$. The relation between measurement $\mathbf{z}_{n+1}$ and state $\mathbf{x}_{n+1}$ is given by (9), where

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and the measurement noise $\mathbf{v}_{n+1} = [v_{n+1_x}, v_{n+1_y}]^T$ has covariance matrix

$$\mathbf{R} = \begin{bmatrix} \sigma_{R_x} & 0 \\ 0 & \sigma_{R_y} \end{bmatrix}$$

where $\sigma_{R_x} = h_x$ and $\sigma_{R_y} = h_y$. This means that the ellipse that actually contains the object and the ellipse we measure in Kalman filtering are overlapping.

The only problem that remains to be solved is the automatic evaluation of $dx_{n+1,n}$ and $dy_{n+1,n}$. Using algorithm 1, we obtain:

– the measurement $\mathbf{z}_{n+1}$,
– the distance between the mixture components of the target model and the target candidate.

The main idea is to use the computed distance to determine if the object was detected or not. This provides a quality measure of the current estimate of the object. If the distance is small then there is a good chance that the object's center is near the predicted center. If this distance is large, then, the target is lost. This distance is embedded in a normalized coefficient:

$$a(\mathbf{y}) = f(d(\mathbf{y})) \tag{10}$$

where $d(\mathbf{y})$ is given by (3) and it is the distance between the model and the candidate histogram at position $\mathbf{y}$ and f is a decreasing function. Experiments have been made with various formulas for f. Relying on the value of $a(\mathbf{y})$ in (10), parameter $\mathbf{d}_{n+1,n} = [dx_{n+1,n}, dy_{n+1,n}]^T$ is automatically updated by:

$$\mathbf{d}_{n+2,n+1} = (1 - a)\mathbf{d}_{n+1,n} + a(\hat{\mathbf{x}}_{n+1} - \hat{\mathbf{x}}_n) \tag{11}$$

where $\hat{\mathbf{x}}_{n+1}$ is the vector containing the estimated values of the horizontal and vertical coordinates of the ellipse center at time $n + 1$. In view of (11), the estimate $\hat{\mathbf{x}}_{n+1}$ contributes to the updates of the displacement $\mathbf{d}_{n+2,n}$ only when the current estimate resembles the source object model, that is when $a(\mathbf{y}) \to 1$. On the other hand ($a(\mathbf{y}) \to 0$), the displacements included in the state matrix $\mathbf{F}_{n+2,n+1}$ remain nearly unchanged, as they were in step $n+1$, considering that the object is occluded. This process has the advantage that matrix $\mathbf{F}_{n+2,n+1}$ incorporating information on the object movement can be updated by the tracking algorithm.

In order to clarify the impact of parameter $a(\mathbf{y})$ in (10), a representative schema is shown in figure 1. The ellipses with the time steps on top of them show the position of the object in the respective frame. The dash lines show the iterations of the mean shift for one frame. The solid arrows show the displacements due to the state matrix $\mathbf{F}_{n+1,n}$. In frame 4, we assume that an occlusion takes place, so the object is lost. The first row, indicated by *MS* shows the results for mean shift while the second row (*MS-K*) show the enhanced mean shift with Kalman filter. Using mean shift, the object is successfully tracked in frames $n = 2$ and $n = 3$, but due to occlusion it is lost after frame $n = 4$. On the other hand, in row 2, the initial state matrix $\mathbf{F}_{2,1}$ has $dx_{2,1} = 0$ and $dy_{2,1} = 0$. The object is tracked using mean shift at time $n = 2$. Assuming that $a(\mathbf{y}) \to 1$ the state matrix is updated and $\mathbf{F}_{3,2}$ has $dx_{3,2} = d_1$ and $dy_{3,2} = 0$. The starting position y_0^3 at time $n = 3$ will not be the same as the end position y^2 of time $n = 2$, but using matrix $\mathbf{F}_{3,2}$ the initial position will be more close to the true object. So the number of iterations of the mean shift is significantly reduced. The state matrix $\mathbf{F}_{4,3}$ is updated by $dx_{4,3} = d_2$ and $dy_{4,3} = 0$. Using this state matrix, the object is assumed to be in position y_0^4 at time $n = 4$, and because mean shift can not find the object (occlusion) the end position is $y^4 = y_0^4$ leaving $\mathbf{F}_{5,4} = \mathbf{F}_{4,3}$ because $a(\mathbf{y}) \to 0$. In the last frame ($n = 5$) by using state matrix

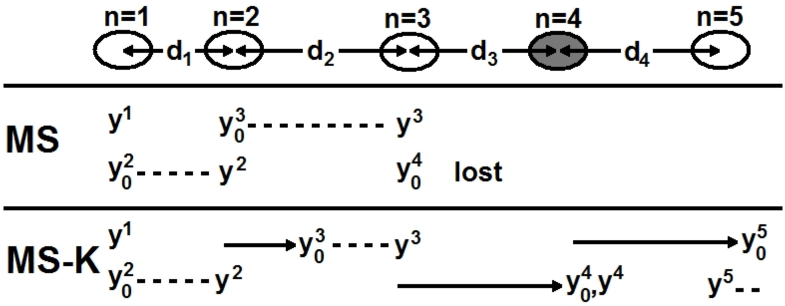

Fig. 1. The displacements d_1, d_2, d_3 and d_4 are different. The first row (MS) shows the displacements using only mean shift. In the second row (MS-K) a Kalman filter is combined with mean shift. The dots represent distinct iterations of mean shift in a single frame. The solid arrows show the displacement predicted by the Kalman filter. In frame $n = 4$, an occlusion takes place and the object is lost by the mean shift, while using Kalman filter the tracking is successful.

$\mathbf{F}_{5,4}$ the initial position y_0^5 bypass the object, but due to mean shift (which is moves its center backwards) the object is successfully located.

Algorithm 3 presents the mean shift with Kalman tracking algorithm with occlusion handling.

5 Experimental Results

To evaluate the proposed algorithm 3, we have performed comparisons with the standard mean shift algorithm [1]. Various test sequences were employed in the evaluation. These sequences consist of outdoor and indoor testing situations. Representative frames are shown in figure 2. Each object is described by its center, in image coordinates, and the size of the ellipse around it (the ellipse has axes parallel to the image axes). The ground truth in every image was determined manually. In all tests, the number of histogram bins for the mean shift algorithms was 16 as suggested in [1]. The examples were carried out with a core 2 Duo 1.66 GHz processor with 2GB RAM under Matlab.

To estimate the accuracy of the compared algorithms we measure the normalized Euclidean distance between the true center ($\mathbf{c}$) of the object (as determined by the ground truth) and the estimated location of the ellipse center ($\hat{\mathbf{c}}$). The normalized Euclidean distance is defined by

$$NED\left(\mathbf{c}, \hat{\mathbf{c}}\right) = \sqrt{\left(\frac{\mathbf{c}_x - \hat{\mathbf{c}}_x}{h_x}\right)^2 + \left(\frac{\mathbf{c}_y - \hat{\mathbf{c}}_y}{h_y}\right)^2} \tag{12}$$

where we recall that h_x and h_y are the ellipse dimensions. This implies that if $NED\left(\mathbf{c}, \hat{\mathbf{c}}\right) < 1$, then the estimated ellipse center $\hat{\mathbf{c}}$ is inside the ground truth ellipse. By these means, the image size and the ellipse dimensions do not influence the relative distance between ($\mathbf{c}$) and ($\hat{\mathbf{c}}$).

Algorithm 3. Mean shift with Kalman filter

1 Initialization: $\hat{\mathbf{x}}_0 \leftarrow$ initial object location:

$$\mathbf{P}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} h_x & 0 & 0 \\ 0 & h_y & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} h_x & 0 & 0 \\ 0 & h_y & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{F}_{1,0} = \mathbf{I}_{3\times 3}.$$

2 Compute the initial histogram q in the first frame as described in (1).

3 Prediction:

$$\hat{\mathbf{x}}_n^- = \mathbf{F}_{n,n-1}\hat{\mathbf{x}}_{n-1},$$
$$\mathbf{P}_n^- = \mathbf{F}_{n,n-1}\mathbf{P}_{n-1}\mathbf{F}_{n,n-1}^T + \mathbf{Q},$$
$$\mathbf{G}_n = \mathbf{P}_n^- \mathbf{H}_n^T[\mathbf{H}_n\mathbf{P}_n^-\mathbf{H}_n^T + \mathbf{R}]^{-1}.$$

4 Measurement: Compute the new center ($\mathbf{z}_n$), $p(\mathbf{y})$ and the distance between q and p using algorithm 1.

5 Estimation:

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \mathbf{G}_n(\mathbf{z}_n - \mathbf{H}_n\hat{\mathbf{x}}_n^-),$$
$$\mathbf{P}_n = (\mathbf{I} - \mathbf{G}_n\mathbf{H}_n)\mathbf{P}_n^-.$$

The output $\hat{\mathbf{x}}_n$ is the object's new location.

6 Update the elements of $\mathbf{F}_n$ using (11).

7 Goto the Prediction step for the next iteration.

Fig. 2. Representative frames of the image sequences. In the first sequence (*walk 1*) a woman is walking from the center of the image to the right. In the second sequence (*walk 2*), a man is walking from the left side to the right and backward. In the third sequence (*car 1*) the car is moving from the left to the right an backward. In the forth sequence (*car 2*) the car is moving from the left to the right and occlusion takes place in the right side of the image.

The proposed algorithm is tested using three different functions for f in (10), $f_1(x) = 1 - x$, $f_2(x) = 1 - \sqrt[10]{x}$ and $f_3(x) = e^{-10x}$. Tables 1 and 2 summarize the comparisons. As it can be seen, the mean shift with Kalman filter has better performance in terms of accuracy and execution time than the original algorithm. Moreover, the type of function f in (10) is not critical for the performance of the algorithm in ordinary tracking scenarios.

In figure 3, an example with occlusion is presented. As the car is moving it is occluded by trees at the right side of the image. The standard mean shift misses the object. On the other hand, the proposed algorithm successfully tracks the

Table 1. Tracking accuracy. The average normalized Euclidean distance between the true object center and the estimated object center is presented for the compared methods.

Sequence	Frames	MS	MS-K$_{f_1}$	MS-K$_{f_2}$	MS-K$_{f_3}$
Walk1	80	0.381	0.183	0.206	0.215
Walk2	156	0.229	0.365	0.404	0.398
Car1	183	0.303	0.226	0.321	0.286
Car2	72	lost	0.391	0.452	0.365

Table 2. Execution times for the compared methods (sec/frame)

Sequence	Frames	MS	MS-K$_{f_1}$	MS-K$_{f_2}$	MS-K$_{f_3}$
Walk1	80	7.983	4.004	3.827	3.858
Walk2	156	8.369	4.302	6.432	6.141
Car1	183	10.39	5.958	7.736	6.831
Car2	72	2.262	1.387	1.775	1.708

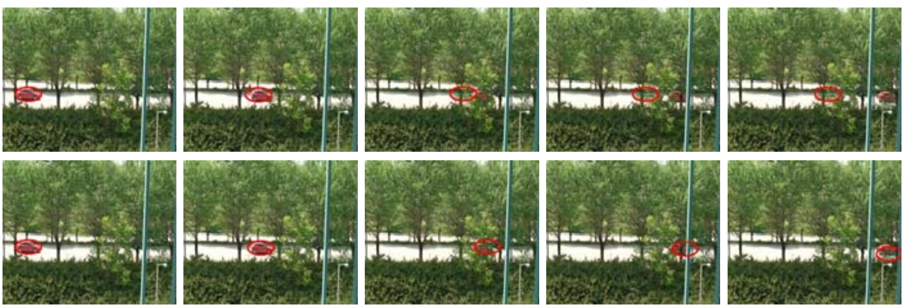

Fig. 3. *Car 2* Frames of a moving car with occlusions. The first row shows the result of the original mean shift. The second row shows the results of mean shift with Kalman filtering using $f_1(x) = 1 - x$.

car. This happens because the state matrix $\mathbf{F}_{n+1,n}$ remains unchanged if the distance (3) is not small enough.

6 Conclusion

We have proposed a method combining mean shift with Kalman filter for object tracking in long image sequences. Using mean shift, we obtain an estimation of the object's location which is then forwarded as an observation to an adaptive Kalman filter. The filter's state matrix is automatically updated with respect to a quality indicator concerning the obtained measurement. This on-line update of the filter's parameters may lead to a significant improvement in accuracy and computation time. Consequently, abrupt motion changes and partial or total occlusions may be successfully addressed. Future work consists in considering tracking of multiple targets.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5), 564–577 (2003)
2. Cuevas, E., Zaldivar, D., Rojas, R.: Kalman filter for vision tracking. Technical Report B 05-12, Freier Universitat Berlin, Institut fur Informatik (2005)
3. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)
4. Fan, Z., Yang, M., Wu, Y.: Multiple collaborative kernel tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(7), 1268–1273 (2007)
5. Zhao, Q., Tao, H.: Differential Earth Mover's Distance with its application to visual tracking. To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence 32(2), 274–287 (2010)
6. Hager, G.D., Dewan, M., Stewart, C.V.: Multiple kernel tracking with SSD. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004), vol. 1, pp. 790–797 (2004)
7. Zhu, Z., Ji, Q., Fujimura, K., Lee, K.: Combining kalman filtering and mean shift for real time eye tracking under active ir illumination. In: Proceedings of 16th International Conference on Pattern Recognition (ICPR 2004), vol. 4, p. 40318 (2002)
8. Babu, R.V., Pérez, P., Bouthemy, P.: Robust tracking with motion estimation and local kernel-based color modeling. Image and Vision Computing 25(8), 1205–1216 (2007)
9. Qi, Y., Jing, Z., Hu, S., Zhao, H.: New method for dynamic bias estimation: Gaussian mean shift registration. Optical Engineering 47(2), 26401 (2008)
10. Lu, H., Zhang, R., Chen, Y.W.: Head detection and tracking by mean-shift and kalman filter. In: Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control (ICICIC 2008), p. 357 (2008)
11. Zhao, J., Qiao, W., Men, G.Z.: An approach based on mean shift and kalman filter for target tracking under occlusion. In: International Conference on Machine Learning and Cybernetics, vol. 4(12–15), pp. 2058–2062 (2009)
12. Bugeau, A., Perez, P.: Track and cut: Simultaneous tracking and segmentation of multiple objects with graph cuts. EURASIP Journal on Image and Video Processing 2008, ID:317278 (2008)
13. Wang, H., Suter, D., Schindler, K.: Effective appearance model and similarity measure for particle filtering and visual tracking. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 606–618. Springer, Heidelberg (2006)
14. Yilmaz, A., Li, X., Shah, M.: Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(11), 1531–1536 (2004)
15. Moreno-Noguer, F., Sanfeliu, A., Samaras, D.: Dependent multiple cue integration for robust tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(4), 670–685 (2008)