Mean Shift Kalman Object Tracking for Video Surveillance

Shehan Fernando[1], T M J A Cooray[2]

Abstract

In this paper we propose the mean shift Kalman object tracking algorithm for video surveillance which is based on the mean shift algorithm and the Kalman filter. The classical mean shift algorithm for tracking in perfectly maintained conditions constitutes a good tracking method. This was based on color to predict the location of the object in the video frame. However in a real cluttered environment this fails, especially under the presence of noise or occlusions. In order to deal with these problems this method employs a Kalman filter to the classical mean shift algorithm to enhance the chance of tracking accuracy especially when the object disappears from the scene, the algorithm can still track the object after it comes out. The experimental results verifies the ability of the mean shift Kalman object tracking algorithm which can locate the target object correctly even in difficult situations when the target is occluded.

1. Introduction[1]

Object tracking is an important research topic in computer vision. Tracking is an important technique for a variety of applications, ranging from military uses such as anti-aircraft and missiles shelter defense systems to very commercial cases like unmanned autopilot navigation, monitoring and security. The main concern in all the applications is to find the object of interest, which is generally called the target, and then to follow it using vision systems. The goal of object tracking in a video stream is to continuously and reliably determine the position of an object against dynamic scenes with the presence of noise [1].

Further, this is highly utilized in the tasks of automobile driver assistance, vehicle navigation, robotics, human-computer interaction, video surveillance, biometrics, video games and industrial automation applications. Most of these applications require reliable object tracking techniques that meet with real-time constraints. The object tracking in a video sequence can be defined as the dynamic entities that constantly change under the influence of several factors. In the physical world, there are five parameters to be considered: the appearance, illumination, scale, object and movement of the object (slow or fast). The variation of one of these parameters can influence the results of the tracking algorithm. A huge number of tracking methods have been proposed during the past. In a video sequence an object can be defined as anything that is interesting for analysis. For eg. people walking on a road, planes in the sky, cars on the road, hand, face in motion etc.

In recent researches, the form and appearance representations are classified into three families; as the representation by points, representation by bounding boxes and the object silhouettes and contour [2], [3]. These methods for object tracking in video sequence and the feature selection for tracking are applied for many tracking algorithms in the earlier researches. Selecting the right feature plays a critical role in the tracking object in video sequence. The feature selection is closely related to the object representation. The object motion analysis usually focuses on simple characteristics such as texture, color, shape, geometry, etc. In general, many tracking algorithms use a combination of these features. Also, these features are selected manually by the user, depending on the application domain. However, the problem of automatic feature selection has received considerable attention in pattern recognition, namely the detection of objects to achieve tracking in video sequences.

In this project the authors have considered color as the main feature for tracking.

2. Literature Review

The proposed schemes in [4],[5] used the posterior probability distribution over some scene properties of interest based on image observations to improve the functionality of object tracking under real working conditions. Other tracking strategies can also be found as Multiple Hypothesis Tracking [6][7], kernel-based tracking [8][9], and tracking based on optical flow [10].

The Active Shape Model (ASM) has been widely used to recognize and track an object from a video sequence. Since this method is computationally heavy, another complementary research[11] proposed an enhanced ASM and predicted mean-shift algorithm to meet these challenges, which combines the context information and predicts mean-shift to obtain multiangle start shapes for ASM searching; and the best result shape is chosen.

3. Methodology

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 algorithm for object tracking is composed of three modules: selection object module in the first frame of the sequence, the module of Meanshift algorithm and the module of Kalman filter algorithm. The selection module selects the

Shehan Fernando[1] is with the Department of Mechanical Eng., Kotelawala Defence Univ. TMJA Cooray[2] is with the Department of Mathematics., Univ. Moratuwa. (e-mail: shehan117@gmail.com; cooray@math.mrt.ac.lk)

position of the object in the first frame. It consists of extracting the module initialization parameters that are moving through the position, size, width, length of the search window of the object in the first frame of the sequence where the Kalman filter is used to find a smoothed estimation of the object position. The following figure Fig. 1 shows the basic modules in the object tracking algorithm.

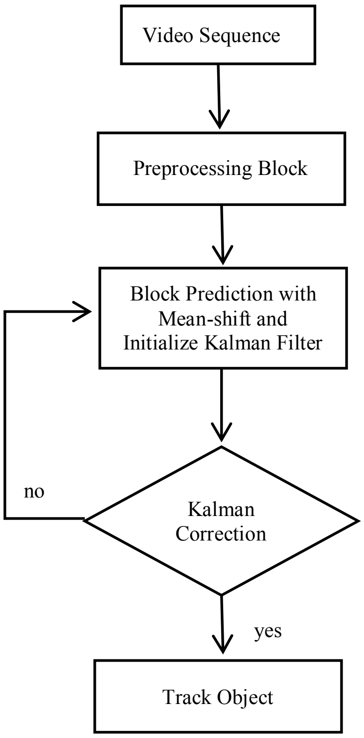


Fig. 1: Flow Chart of the Object Tracking System

4. Tracking with Mean-shift Algorithm

The mean-shift algorithm is a non-parametric method. It provides accurate localization and efficient matching without expensive exhaustive search. This algorithm tries to locate the object by finding the local maximum of a function. The size of the window of search is fixed. It is an iterative process, where it first computes the mean-shift value for the current point position, then moves the point to its mean-shift value as the new position, then computes the mean-shift until it fulfills certain conditions. For a frame, we use the distribution of the levels of grey which gives the description of the shape and we are going to converge on the center of mass of the object calculated by means of moments. The flow chart of mean-shift in Fig. 2 describes the steps of the algorithm. The number of iterations of the convergence of the algorithm is obtained when the subject is followed within the image sequence.

The object target pdf is approximated by a histogram of m bins $\hat{q} = \{\hat{q}_u\}_{u=1,2,3,\ldots m}$, $\sum_{u=1}^{m} \hat{q}_u = 1$ with $\hat{q}_u$being the u-th bin. To form the histogram, only the pixels inside a selected rectangular area surrounding the object are taken into account. The centroid of the rectangle is assumed to be at the origin of the axes. Due to the fact that the rectangle area 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 rectangle to be considered more important.

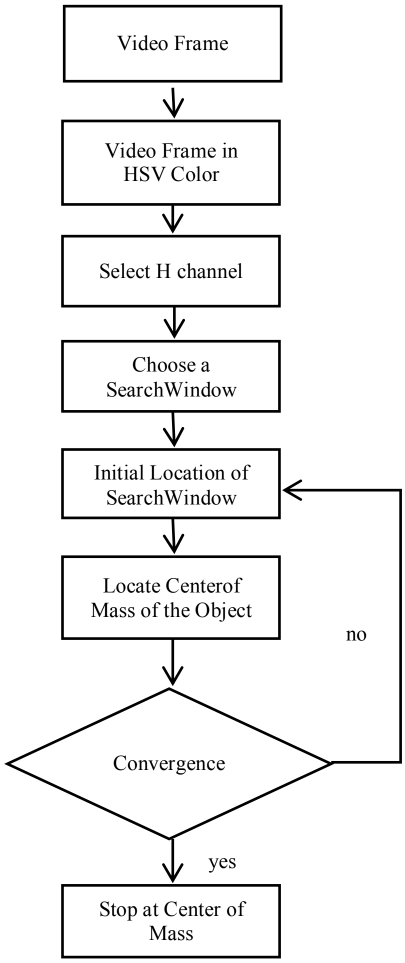


Fig. 2: Flow Chart of Mean-shift Algorithm.

The u-th histogram bin can be calculated by the following equation,

$$\hat{q} = C \sum_{i=1}^{n} k(\|x_i^*\|^2)\delta[b(x_i^*) - u] \qquad (1)$$

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

In the next image, the object candidate is inside the same rectangular area with its center at the normalized spatial location $\boldsymbol{y}$. Let $\{\boldsymbol{x}_i\}_{i=1,\ldots,n}$be the normalized pixel coordinates inside the target rectangular area. The pdf of the target candidate is also approximated by an m-bin histogram,

$$\hat{p}_u(\boldsymbol{y}) = C_c \sum_{i=1}^{n} k(\|y - x_i\|^2)\delta[b(x_i) - u] \qquad (2)$$

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

Algorithm 1: Maximizing Bhattacharyya coefficient $\rho(\hat{q}, \hat{p}_u(\boldsymbol{y}))$

Input: The target model $\hat{q} = \{\hat{q}_u\}_{u=1,2,3,\ldots m}$ and its location $\hat{\boldsymbol{y}}_0$ in the previous frame.

1. Initialize the center of the roi in the current frame at $\hat{\boldsymbol{y}}_0$, compute $\{\hat{p}_u(\hat{y}_0)\}_{u=1,\ldots,m}$ using (2) and evaluate $\rho(\hat{q}, \hat{p}_u(\boldsymbol{y_0}))$ using (4).
2. Compute the weights $\{w_i\}_{i=1,2,\ldots,n}$ according to (6).
3. Compute the next location of the target candidate according to (7).
4. Compute $\{\hat{p}_u(\hat{y}_1)\}_{u=1,\ldots,m}$ using (2) and evaluate $\rho(\hat{q}, \hat{p}_u(\boldsymbol{y_1}))$ using (4).
5. If $|\hat{y}_1 - \hat{y}_0| < \epsilon$ Stop.
Otherwise set $\hat{y}_0 \leftarrow \hat{y}_1$ and go to Step 2.

Output: Maximum Bhattacharyya coefficient $\rho(\hat{q}, \hat{p}_u(\boldsymbol{y}))$

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

$$d(\boldsymbol{y}) = \sqrt{1 - \rho(\hat{q}, \hat{p}_u(\boldsymbol{y}))} \tag{3}$$

Where,

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

is the similarity function between $\hat{q}$ and $\hat{p}_u(\boldsymbol{y})$, 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 rectangle center is initialized at a location $\hat{\boldsymbol{y}}_0$ which is the rectangle center in the previous image frame. The probabilities $\{\hat{p}_u(\hat{y}_0)\}_{u=1,\ldots,m}$ are computed and using linear Taylor approximation of (4) around these values:

$$\rho(\hat{q}, \hat{p}_u(\boldsymbol{y}) \approx \frac{1}{2}\sum_{u=1}^{m} \sqrt{\hat{q} \times \hat{p}_u(\boldsymbol{y})} + \frac{C_c}{2}\sum_{i=1}^{n} w_i k(\|y - x_i\|^2) \tag{5}$$

Where,

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \tag{6}$$

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

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

The complete algorithm is summarized in algorithm 1.

In the figure Fig. 3(a), the initialization or selection of the object of interest was performed. After that Fig. 3(b), (c) show how the Mean-shift algorithm tracks the vehicle. It shows that the tracking window is not centered on the object of interest. Later we can observe that this tracking window lost its object. This is also evident by referring to Fig. 3(c).

(a) Initialize ROI

(b) Starts to track object

(c) The mean-shift method merely lost the roi

Fig. 3: Tracking Vehicle with mean-shift

Another drawback of using mean-shift algorithm is that it cannot handle partial or full occlusion.

5. Kalman Filter

In general, we assume that there is a linear process governed by an unknown inner state producing a set of measurements [12] [13] [14]. More specifically, there is a discrete time system and its state at time *n* is given by vector $\boldsymbol{x}_n$. The state in the next time step *n* + 1 is given by

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

where $\boldsymbol{F_{n+1,n}}$is the transition matrix from state $\boldsymbol{x_n}$ to $\boldsymbol{x_{n+1}}$ and $\boldsymbol{w_n}$is white Gaussian noise with zero mean and covariance matrix$Q$.

$$\boldsymbol{w} \sim N(0, \boldsymbol{Q}) \tag{9}$$

The measurement vector $\boldsymbol{z_{n+1}}$ is given by

$$\boldsymbol{z_{n+1}} = \boldsymbol{H_{n+1}x_{n+1}} + \boldsymbol{V_{n+1}} \tag{10}$$

where $\boldsymbol{H_{n+1}}$ is the measurement matrix and $\boldsymbol{V_{n+1}}$ is white Gaussian noise with zero mean and covariance matrix R.

$$\boldsymbol{V} \sim N(0, \boldsymbol{R}) \tag{11}$$

In equation (10), the measurement $\boldsymbol{z_{n+1}}$ depends only on the current state $\boldsymbol{x_{n+1}}$and the noise vector $\boldsymbol{V_{n+1}}$is independent of the noise $\boldsymbol{w}$.Kalman filter computes the minimum mean-square error estimate of the state $\boldsymbol{x_k}$ given the measurements $\boldsymbol{z_1}, \boldsymbol{z_2}, \dots, \boldsymbol{z_k}$. The solution is a recursive procedure ,which is described in algorithm 2.

The discrete-time process model will be given by

$$\boldsymbol{X}_t = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{X}_{t-1} + N(0, Q) \tag{12}$$

where

$$\boldsymbol{F_{n,n-1}} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

and $N(0, \boldsymbol{Q})$represents anormal distribution for the model with a covariance given by $\boldsymbol{Q}$.

For initialization, the initial positions are set to the initial centers of the region of interest (computed in the very first frame by background subtraction or mouse selection)and

Algorithm 2: Kalman filter

1 Initialization:

$\hat{x}_0 = E[x_0]$

$P_0 = E[x_0 - E[x_0]][x_0 - E[x_0]]^T$

2 Prediction:

$\boldsymbol{\hat{x}_n^-} = \boldsymbol{F_{n,n-1}\hat{x}_{n-1}}$

$\boldsymbol{P_n^-} = \boldsymbol{F_{n,n-1}P_{n-1}F_{n,n-1}^T} + \boldsymbol{Q_n}$Projection of the error covariance ahead

$\boldsymbol{G_n} = \boldsymbol{P_n^- H_n^T}[\boldsymbol{H_n P_n^- H_n^T} + \boldsymbol{R_n}]^{-1}$Compute the Kalman gain

3 Estimation:

$\boldsymbol{\hat{x}_n} = \boldsymbol{\hat{x}_n^-} + \boldsymbol{G_n}(\boldsymbol{z_n} - \boldsymbol{H_n\hat{x}_n^-})$Update estimate with measurement $\boldsymbol{z_n}$

$\boldsymbol{P_n} = (\boldsymbol{I} - \boldsymbol{G_n H_n})\,\boldsymbol{P_n^-}$Update the error covariance

Go to the Prediction step for the next prediction.

6. Tracking with Mean-shift Kalman

In order to increase the accuracy and robustness of the mean-shift tracking algorithm, and to deal with partial or total occlusion, the Kalman filter was introduced for the mean-shift algorithm. The idea is to predict the position of the tracked object in the new frame based on the object's previous motion. 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 or the (estimation).Moreover, in this section, we propose to add a search window in the neighborhood of the previous estimated window, so that the next observation will not be searched in the whole frame but only inside the search window. By this means the tracking procedure was significantly accelerated.

6.1 Methodology

Let the states of the filter be $\boldsymbol{X} = [x_g, y_g, \dot{x}_g, \dot{y}_g]$, where $\dot{x}_g, and\ \dot{y}_g$ represent the velocity in$x_g\ and\ y_g$, respectively.

the velocities are set initialized to an arbitrary value. The discrete-time measurement model is given by:

$$\boldsymbol{Y}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{X}_t + N(0, \boldsymbol{R}) \tag{14}$$

where $N(0, \boldsymbol{R})$ is the zero-centred normal distribution corresponding to the uncertainty in the measurement, with R as the covariance matrix for the measurement.

After the region of interest object is selected, the basic iteration of the mean-shift algorithm is performed for the second frame; however before assigning the new center of the ROI in the new frame (the tracked object), the prediction part of the Kalman filter is computed with the well-known linear Kalman equations. After the prediction step, the value of the displacement is calculated.

6.2 Example

This methodology was applied to the same video sequence used earlier. In the figure Fig. 4(a), the initialization or selection of the object of interest was performed. After that Fig. 4(b), (c) shows how the Mean-shift Kalman algorithm tracks the vehicle and the tracking window is centered on the object of interest.

Algorithm 3: Mean shift with Kalman filter

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

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

3 Prediction:

$$\widehat{\boldsymbol{x}}_{\boldsymbol{n}}^{-} = \boldsymbol{F}_{\boldsymbol{n,n-1}}\widehat{\boldsymbol{x}}_{\boldsymbol{n-1}}$$
$$\boldsymbol{P}_{\boldsymbol{n}}^{-} = \boldsymbol{F}_{\boldsymbol{n,n-1}}\boldsymbol{P}_{\boldsymbol{n-1}}\boldsymbol{F}_{\boldsymbol{n,n-1}}^{\boldsymbol{T}} + \boldsymbol{Q}_{\boldsymbol{n}}$$
$$\boldsymbol{G}_{\boldsymbol{n}} = \boldsymbol{P}_{\boldsymbol{n}}^{-}\boldsymbol{H}_{\boldsymbol{n}}^{\boldsymbol{T}}[\boldsymbol{H}_{\boldsymbol{n}}\boldsymbol{P}_{\boldsymbol{n}}^{-}\boldsymbol{H}_{\boldsymbol{n}}^{\boldsymbol{T}} + \boldsymbol{R}_{\boldsymbol{n}}]^{-1}$$

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

5 Estimation:

$$\widehat{\boldsymbol{x}}_{\boldsymbol{n}} = \widehat{\boldsymbol{x}}_{\boldsymbol{n}}^{-} + \boldsymbol{G}_{\boldsymbol{n}}(\boldsymbol{z}_{\boldsymbol{n}} - \boldsymbol{H}_{\boldsymbol{n}}\widehat{\boldsymbol{x}}_{\boldsymbol{n}}^{-})$$
$$\boldsymbol{P}_{\boldsymbol{n}} = (\boldsymbol{I} - \boldsymbol{G}_{\boldsymbol{n}}\boldsymbol{H}_{\boldsymbol{n}})\,\boldsymbol{P}_{\boldsymbol{n}}^{-}$$

The output $\widehat{\boldsymbol{x}}_{\boldsymbol{n}}$is the object's new location.

6 Goto the Prediction step for the next iteration.

In this situation, the new center relies completely on the prediction of the Kalman Filter, since no measurement is taken into account.

(a) Initialize roi

(b) Starts to track object

(c) The mean-shift Kalman method centered on the roi

Fig. 4: Tracking Vehicle with mean-shift Kalman algorithm

7. Experimental Results

To evaluate the Mean-shift Kalman method in algorithm 3, we have performed comparisons with the standard mean shift algorithm in algorithm 1. Various test sequences were employed in the evaluation. These sequences consist of mainly outdoor testing situations. Example frames are shown in Fig. 5. Each object is described by its center, in image coordinates, and the size of the rectangle around it.

(a) Initializing the roi with mouse

(b) After 10 frames still the roi is tracked

(c) After 20 frames (on left) using mean-shift the rectangle is moving away from the vehicle, while (on right) the mean-shift Kalman could track the vehicle.

(d) After 60 frames (on left) using mean-shift the rectangle is away from the vehicle, while (on right) the mean-shift Kalman could still track the vehicle successfully.

Fig. 5: Comparing the tracking of a identified vehicle using mean-shift and mean-shift Kalman

Moreover, the mean-shift Kalman algorithm is robust enough to track an object under partial or full occlusion conditions. However, the mean-shift fails to handle such complexities.

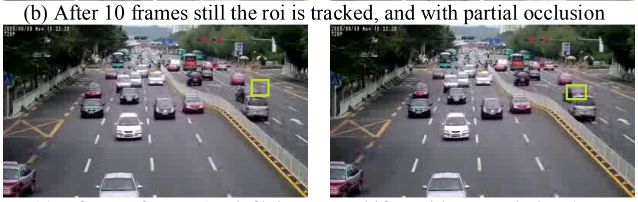
This can be witnessed by analyzing the above test sequences in Fig. 6.

8. Limitations of the Algorithm

The experiments showed that this algorithm is quite sensitive to the initial conditions, that is, if the bounding box is initially not located in a good place (or, equivalently, if the initial target is not precisely determined), it will go to a wrong place after some movement of the object. However, this displacement will not be very big and tracking will be still possible, although less accurate. Second, in case that the difference between the background and the object is small in the sense of illumination and contrast, it is hard to track the object with the color model that was defined. One of the possible ways is to try to adapt the color space according to the case of study, if high accuracy is needed. But in case there is a good difference between the object and its background, the algorithm works well due to much redundant information. The main limitation of simple mean-shift tracking is the case of total occlusion. The proposed way to solve this problem in this paper was to use the Kalman filter to predict the position of the tracked object when it is partially or totally occluded.

(a) Initializing the roi with mouse

(b) After 10 frames still the roi is tracked, and with partial occlusion



(c) After 20 frames (on left)the mean-shift could not track, but (on right)the mean-shift Kalman could track the object.

(d) After 40 frames (on left)the mean-shift has merely lost the object, but (on right)the mean-shift Kalman could still track the object.

(e) After 80 frames (on left)the mean-shift has lost the object, however (on right)the mean-shift Kalman was still able to track the object.

Fig. 6: Comparing the tracking of a identified vehicle using mean-shift and mean-shift Kalman under partial and full occlusion

9. 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 a Kalman filter. As a future enhancement the authors propose the filter's state matrix to be automatically updated for abrupt motion changes which means an adaptive Kalman filter. Future work consists in considering tracking of multiple targets.

References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,vol. 34, no. 3, pp. 334–352, 2004.

[2] N.A Gmez, "A Probabilistic Integrated Object Recognition and Tracking Framework for Video Sequences". Universitat Rovira I Virgili, PHDthesis, Espagne, 2009.

[3] O. S. Afzal, and N. Vaswani, and A. Tannenbaum ,A. and Yezzi, " ObjectTracking: A Survey". IEEE, no. 6, pp. 4244–4249. 2009.

[4] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," International journal of computer vision,vol. 29, no. 1, pp. 5–28, 1998.

[5] K. Choo and D. Fleet, "People tracking using hybrid montecarlo filtering," in Proceedings of the Eighth IEEE International Conferenceon Computer Vision, vol. 2. IEEE, 2001, pp. 321–328.

[6] D. Reid, "An algorithm for tracking multiple targets," Automatic Control, IEEE Transactions on, vol. 24, no. 6, pp. 843–854, 1979.

[7] Multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," Pattern Analysis and Machine Intelligence,IEEE Transactions on, vol. 18, no. 2, pp. 138–150, 1996.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking,"IEEE Transactions on Pattern Analysis and Machine Intelligence,pp. 564–575, 2003.

[9] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearancemodels for visual tracking," IEEE Transactions on Pattern Analysisand Machine Intelligence, pp. 1296–1311, 2003.

[10] B. Lucas, T. Kanade et al., "An iterative image registration Technique with an application to stereo vision," in International joint conference on artificial intelligence, vol. 3. Citeseer, 1981, pp. 674–679.

[11] B. Pu, S. Liang, Y. Xie, Z. Yi, and P. Heng, "Video facial feature tracking with enhanced asm and predicted meanshift," in 2010 Second International Conference on Computer Modeling and Simulation.IEEE, 2010, pp. 151–155.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35-45, 1960.

[13] An Introduction to the Kalman Filter Greg Welch, Gary Bishop SIGGRAPH 2001.

[14] Visual Tracking by Adaptive Kalman Filtering and Mean Shift, Vasileios Karavasilis, Christophoros Nikou, and Aristidis Likas.

[15] Cuevas, E., Zaldivar, D., Rojas, R.: Kalman filter for vision tracking. Technical Report B 05-12, Freier-Universitat Berlin, Institut fur Informatik (2005).