# EFFICIENT MANAGEMENT OF MULTIPLE AGENT TRACKING THROUGH OBSERVATION HANDLING

J. Gonzàlez[†], D. Rowe[‡] , J. Andrade[†], J.J. Villanueva[‡]

[†] Institut de Robòtica i Informàtica Industrial (UPC – CSIC),
Llorens i Artigas 4-6, 08028, Barcelona, Spain
[‡] Computer Vision Center & Dept. d'Informàtica,
Edifici O, Campus UAB, 08193 Bellaterra, Spain

**ABSTRACT**
Non-supervised multiple-agent tracking is a complex task which demands a structured framework in order to accomplish it. Therefore, this proposal presents a system which is modular and hierarchically organised. It consists in several levels, working in cascade, which are defined according to the different functionalities to be performed. The goal of this work is to implement and experimentally verify a novel image-based algorithm which deals with serious segmentation difficulties, thereby being able to simultaneously perform a reliable tracking of several agents. As a result, agents' trajectories are obtained, as well as quantitative information about their state at any time, such as their speed or size.

**KEY WORDS**
Multiple Agent Tracking, Low Level Tracking, Agent Detection.

## 1 Introduction

Computer-based tracking of multiple agents has become an active research field [6]. This interest is motivated by an increasing number of potential applications, such as smart video surveillance, intelligent user-computer interfaces or the evaluation of human sequences (HSE) [10]. Obtaining robust performances, whilst using non-intrusive technology, is frequently mandatory. Despite this interest, this still constitutes an open problem which is far from been solved, and where serious difficulties should be expected. In open-world applications, the number of agents within the scene may vary over time. In unconstrained environments, the illumination and background-clutter distracters are uncontrolled, affecting the perceived appearance over time. This depends on issues such as the agents' position or orientation and how they face different, varying sources of light.

In the literature, tracking performance is usually based on the results of foreground segmentation, and a subsequent target association. Segmentation can be performed by means of optical flow [4], background subtraction [11], frame differencing, or a combination of these [6]. The association can be accomplished using nearest neighbour techniques, or by means of Data Association Filters (PDAF, JPDAF, MHF), depending on whether several targets and measurements are expected [3]. Usually, a prediction stage is also incorporated, thereby providing better chances of tracking success. Filters such as the Kalman Filter [15], or subsequent extensions such as the EKF [1] or UKF [14] are commonly used. More general dynamics and measurement functions can be dealt with Particle Filters (PF) [2] and further evolutions, such as the UPF [18].

Specifically, Nummiaro et al. [17] use a particle filter based on colour-histogram cues. However, no multiple-target tracking is considered, and it lacks from an independent observation process, since samples are evaluated according to the histograms of the predicted image region. Deutscher et al. [9] present an interesting approach called *annealing particle filter* which aims to reduce the required number of samples. However, pruning hypotheses with lower likelihood could be inappropriate in cluttered environments. They combine edge and intensity measures, but they focused on motion analysis, and thus multiple targets and unconstrained environments are not explored. Contour tracking have been widely explored [12, 16], although this may not be the best approach in crowded scenarios because of the potential multiple occlusions. BraMBLe [13] is an interesting approach to multiple-blob tracking which models both background and foreground using MoG. However, no model update is performed, there is a common foreground model for all targets, and it may require an extremely large number of samples, since one sample contains information about the state of all targets, dramatically increasing the state dimensionality.

Recently, a rather different approach has been introduced [5, 8, 7]. Comaniciu et al. [8] developed an attractive technique called mean-shift tracker. However, their method tracks just one target, initialised by hand and the appearance model was never updated. Collins et al. [7] presented an effective tracker, based also on the mean-shift algorithm, with online selection of discriminative features. It aims to maximise the distinction between the target appearance and its surroundings. Still, it tracks just one target, initialised by hand and which may suffer from model drift. In both cases, just rigid targets are tracked (or rigid regions of them), and since multiple-target tracking is not considered.

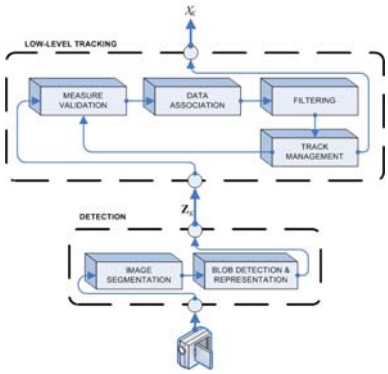



Figure 1. System Architecture for Observation-Measurement Handling

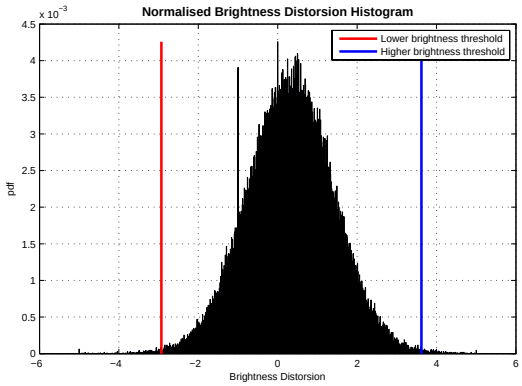



Figure 2. Threshold computation. Thresholds are automatically computed by cumulating histogram values and applying a detection rate.

The remainder of this paper is organized as follows. Section 2 outlines the proposed method, which is is fully described in the next three sections: section 3 details how the segmentation is carried out, and the chosen data representation; section 4 discuss the low-level tracking module; section 5 discusses the experimental results; and section 6 summarises the conclusions, and proposes future-work lines.

## 2 Approach Outline

Reliable target segmentation is critical in order to achieve an accurate feature extraction without considering prior knowledge about the potential targets, specially in dynamic scenes. However, agents who move through cluttered environments require a structured framework to deal with poor detection results. A sketch of this system is shown in Fig. 1.

The lower level performs the target detection task. It consists in two modules. The first one accomplishes the segmentation task, which involves separating image regions that do not belong to the background and extracting them. In order to carry it out, a colour-based background subtraction method is used. Subsequently, the obtained image mask is filtered, and the result is manipulated to obtain representations which can be handled by the low-level tracker. Particularly, an ellipse representation is chosen.

The low-level tracker aims to establish coherent relations of the different targets between frames. In order to accomplish this task, four processed are carried out. In the first place, *gates* are computed, that is, the regions where the observations are expected to appear are specified. This is done according to the target state and the system uncertainties. Subsequently, *data association* is performed. In this stage, correspondences between observations and trackers are set based on a nearest-neighbour decision in the observation space. Afterwards, *filtering* is performed, that is, new target states are estimated according to the associated observations. This is accomplished by a bank of Kalman filters, operating in a state space given by the target estimated centroid, its speed, the axis lengths, the axis length change rates, and the ellipse angle. Finally, the *track-management* module (i) initiates tentative tracks for those observations which are not associated to any existing target; (ii) confirms tracks with enough supporting observations; and (iii) removes low-quality ones. Results are lastly fed back to the measure-validation module.

## 3 Colour Segmentation and Blob Detection

Image segmentation is performed following the method proposed by Horprasert et al. [11], a colour background subtraction approach whose main characteristics are detailed next.

### 3.1 Background model

The background is statistically modelled on a pixel-wise basis, using a window of $N$ frames. During this training period, the mean $\mathbf{E}_i$ and standard deviation $\sigma_i$ of each pixel RGB-colour channel.

Two distortion measures are established: $\alpha$, the brightness distortion, and $CD$, the chromacity distortion. Once each colour-channel value is normalised by their respective standard variation, the brightness distortion is computed by minimising the distance between the current pixel value and the chromacity line. The variation over time of both distortions for each pixel is subsequently computed by means of the Root Mean Square. These values are used as normalising factors so that a single threshold can be set for the whole image, see [11] for details.

Fig 2 shows the normalised brightness distortion histogram for a given frame, as well as the corresponding thresholds.

**(a)** **(b)**

Figure 3. Segmentation and detection examples. **(a)** The segmented foreground pixels are painted on white, while those ones classified as dark foreground are painted on yellow. Shadows are painted on green and highlights on red. **(b)** Detection example: red ellipses represent each target, and yellow lines denote their contour.

### 3.2 Image segmentation

Pixels are classified into five categories, depending on their chromacity and brightness distortion. For each frame, both normalised pixel distortions are computed. Those pixels whose chromacity distortion is higher than expected (that is, over the chromacity threshold) are marked as foreground. Those which are not, if the brightness distortion is more negative than the dark threshold, are marked as dark foreground. The rest are classified as highlight, if the brightness distortion is higher that the upper distortion threshold; or shadows, if the brightness distortion is lower than the lower distortion threshold. If none of these conditions hold, the pixel is classified as normal background. An example of foreground segmentation is show in Fig 3.(a).

### 3.3 Blob detection

Once the current image has been segmented into the aforementioned five categories, blobs that may correspond to agents are detected. First, both foreground and dark-foreground maps are fused. Then, majority, opening and closing morphological operations are applied. Finally, a minimum-area filter is used. The surviving pixels are grouped into blobs. Each blob is labelled, their contours are extracted and an ellipse representation —which keeps the blob first and second moments— is computed. Thus, the *j*-observed blob at time *t* is given by the vector $\mathbf{z}_j^t = \left(x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t\right)$, where $x_j^t, y_j^t$ represent the ellipse centroid, $h_j^t, w_j^t$ are the major and minor axes, respectively, and the $\theta_j^t$ gives the angle between the abscissa axis and the ellipse major one. Fig 3.(b) shows an example of target detection.

## 4 Low-level blob tracker

Given successive target detections, the observations must be associated, thereby establishing correspondences between frames. The target state is then estimated by filtering the sequence of noisy measures.

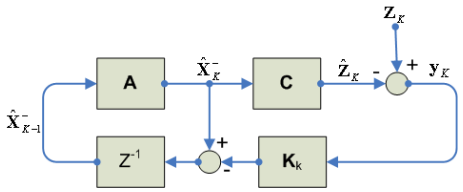



Figure 4. Recursive Kalman filter.

### 4.1 State-space model

In this work, it is assumed that human beings move slowly enough compared to the frame rate. Since their long-run dynamics are hardly predictable, a first-order dynamic model is adopted. This assumption holds in most HSE applications. The observation vector at time $t$ is given by the blob detection module. The target state is defined by $\mathbf{x}_j^t = \left(x_j^t, \dot{x}_j^t, y_j^t, \dot{y}_j^t, h_j^t, \dot{h}_j^t, w_j^t, \dot{w}_j^t, \theta_j^t\right)$, which defines a state variable for every observation one and adds the target speed and the size change rate. Thus, the model considered is given by a constant-speed approach where the acceleration is modelled as *White Additive Gaussian Noise* (WAGN) —except for the angle variable, whose speed is modelled as noise.

### 4.2 Linear Filters

We begin our experiments by including a linear filter, but the inclusion of more complex filters would be straightforward. To start with, the Kalman filter [15] implements a recursive algorithm which works in a prediction-correction way, estimating the system state from noisy measures. The estimator is optimal in the sense that it minimises the steady-state error covariance. However, strong assumptions are required: the transition model must be linear Gaussian, and the sensor model must be Gaussian. Nevertheless, albeit these conditions rarely exist, the filter still works reasonably well for many applications.

It works in two steps which are recursively performed (a block diagram is shown in Fig 4) . In the first one a prediction is made: the expectation and covariance are propagated according to the the dynamic model, thereby obtaining the temporal prior:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1}, \quad (1)$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}. \quad (2)$$

After obtaining the new measurement $\mathbf{z}_k$, the second step is carried out, and values are updated according to the observation likelihood:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{y}_k, \quad (3)$$
$$\mathbf{P}_k = \mathbf{I} - \mathbf{K}_k \mathbf{C} \mathbf{P}_k^-, \quad (4)$$

where:

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_k^-, \quad (5)$$

is called the *innovation or the residual*,

$$\mathbf{S}_k = \mathbf{C}\mathbf{P}_k^- \mathbf{C}^T + \mathbf{R}, \quad (6)$$

is called the *innovation covariance*, and

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}^T \mathbf{S}_k^{-1}, \quad (7)$$

is known as the *Kalman gain.*

### 4.3 Measure validation

In a multiple-target tracking scenario, numerous observations may be obtained at every sampling period. In this case, some observations could have been generated by clutter or noise processes, and several observations might correspond to the same target with a given probability. Measure validation consists in establishing the regions or *gates* where the target observations are expected, in agreement with the target state and the system uncertainties.

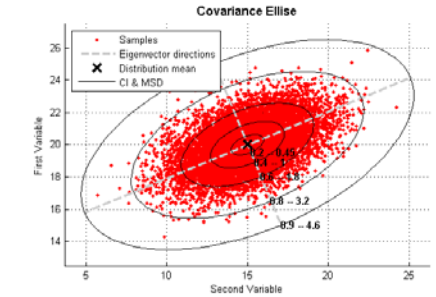
The gating process works as follows. Each target expected observation is predicted according to the system dynamics:

$$\hat{\mathbf{z}}_k = \mathbf{C}\mathbf{A}\hat{\mathbf{x}}_{k-1}. \quad (8)$$

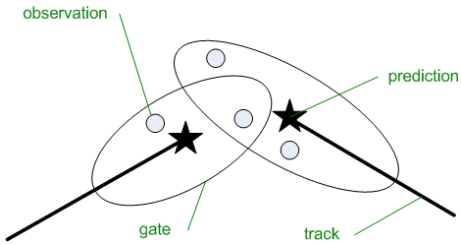
The predicted error covariance is computed according to Eq. (2). Subsequently, the innovation covariance matrix $\mathbf{S}_k$ defines an ellipsoid in the observation space whose axes are given by the covariance matrix eigenvectors, and the axis length —for the ellipsoid with unit Mahalanobis radius— is given by the the square root of corresponding eigenvalues. A particular Mahalanobis radius defines an ellipsoid, centred at the mean of the distribution, which encloses a probability mass given by the confidence interval associated with the ellipsoid, see Fig 5.(a). The Mahalanobis Squared Distance (MSD) of a *d*-dimensional Gaussian observation variable is given by:

$$d^2_{Mahal} = (\mathbf{z}_k - \hat{\mathbf{z}}_k)\, \mathbf{S}_k^{-1}\, (\mathbf{z}_k - \hat{\mathbf{z}}_k)^T, \quad (9)$$

and it is distributed according to a Chi-squared distribution with *d* degrees of freedom:




**(a)**




**(b)**

Figure 5. **(a)** Innovation Covariance Ellipsoid. **(b)** Observation association.

$$d^2_{Mahal} \sim \chi^2_d. \quad (10)$$

Thus, the Mahalanobis radius corresponding to the ellipsoid with a given confidence interval can be computed by evaluating the inverse of the cumulative distribution function of the Chi-squared distribution. This means that measures can be validated for a given confidence interval by calculating the MSD between the predicted observation and the actual one, and comparing this value with the Mahalanobis radius for this confidence interval —which is obtained from the inverse of the cumulative distribution function of the Chi-squared distribution.

### 4.4 Data Association and Filtering

Measures are associated to the nearest neighbour tracker in whose gate they lie, see Fig 5.(b). A more complicated data association method, such as PDAF or JPDAF, is not considered to be necessary since observations are usually just within one target gate. This is intrinsic to the segmentation method: if two targets are so close in the observation space as to introduce ambiguity in the data association process, the segmentation module is likely to segment just one blob corresponding to the group formed by both targets. This issue is addressed at the event-management section.

A bank of Kalman filters is implemented to estimate the state of all targets detected within the scene. As a special case, if no observation is associated to a particular target, its state is estimated using a Kalman Gain equal to zero, i.e. it is just propagated according to the dynamic model.

### 4.5 Track Management

This module manages the target tracks by instantiating, confirming and removing them. This is done according to the values of two indicators: the square root of the covariance matrix determinant and the observation Mahalanobis Square Distance. The first one, the square root of the covariance matrix determinant, is related to the track uncertainty. The determinant is given by the product of the matrix eigenvalues, which correspond to the variance of the dimensions given by the eigenvectors. The innovation covariance matrix $\mathbf{S}_k$ is calculated recursively, which depends just on the system matrices $\mathbf{A}, \mathbf{C}, \mathbf{Q}$ and $\mathbf{R}$. That means that while an observation is associated, the determinant of the innovation covariance matrix will decrease to its asymptotic value, and the time taken depends only on the system dynamics and the uncertainties given by the covariance matrices. That is to say, it does not depend on the observation MSD. However, it is a good indicator of how many observations have been associated and whether there have been frames without any observation. This is done without the need of setting thresholds and specifying cases: it is intrinsic to the behaviour given by the system dynamics.

Nevertheless, the quality of the observation must be taken into account, and therefore, the MSD of each target associated observation is evaluated and compared with a pre-defined confidence value. The MSD, seen as the Mahalanobis radius of the ellipsoid, is used to mark those observations beyond a given ellipsoid variance.

## 5 Experimental Results

The performance of the algorithm has been tested using sequences taken from the PETS 2001 Test case Scenarios [1], recorded in an outdoor scene. A track is instantiated every time an observation remains orphan: when $|\mathbf{S}|^{\frac{1}{2}}$ is below a certain percentage of its asymptotic value, and the MSD is lower than a given ellipsoid variance, the track is confirmed. If $|\mathbf{S}|^{\frac{1}{2}}$ grows beyond a value which corresponds to a certain number of consecutive frames without observation, the track is deleted, and the Kalman filter removed. These behaviours can be seen in Fig 6: at frame 354 a target starts entering the scene, an observation is received and a tracker is instantiated; while new observations are associated, the determinant indicator decreases. However, at frame 357 a major change happened —because the target has completely entered the visual field— the MSD is higher enough so that the observation is not associated to the existing tracker, a new track is instantiated —the observation is far beyond the gate boundary— and the previous process is repeated. Lastly, at frame 367 there are several segmentation errors, but as measurements still lie within the gate, there is not an instantiation of a new tracker, that is, they are assigned to the existing tracker. Fig. 7 shows the sequence result of the tracking process.

[1] International Workshops Performance on Evaluation of Tracking and Surveillance at http://peipa.essex.ac.uk/ipa/pix/pets/

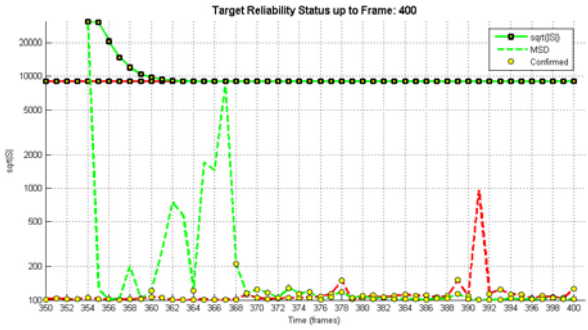



Figure 6. Track management .Tracks are confirmed when both $|\mathbf{S}|^{\frac{1}{2}}$ and MSD are low enough. Tracks with high $|\mathbf{S}|^{\frac{1}{2}}$ are removed.

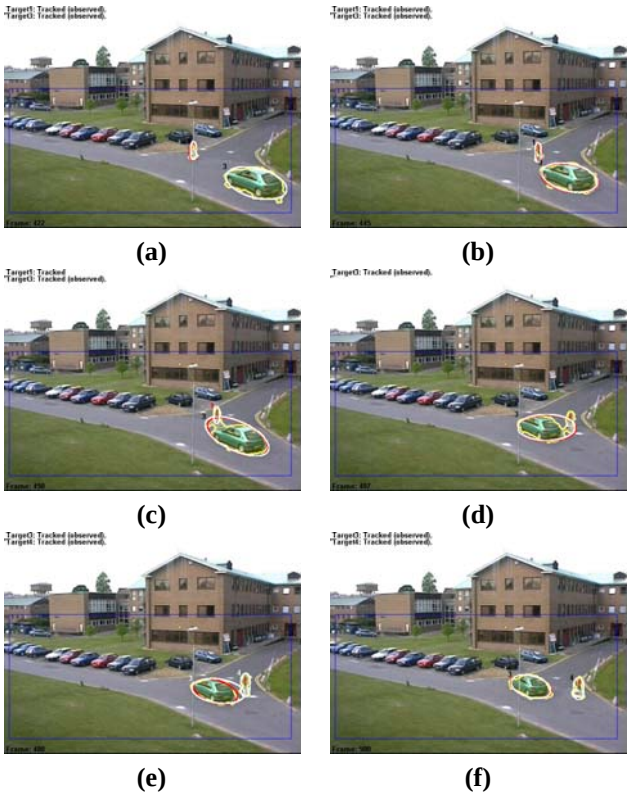



Figure 7. Experimental Results. See text for details.

Several considerations must be taken into account according to the results shown in Fig. 6. In the first place, depending on the system matrices, the time needed to reach a value close to the asymptotic value of the determinant may considerably vary. Thus, if $|\mathbf{Q}|$ grows, the dynamics are less reliable, the Kalman Gain grows, the state variables are more affected by the observation values, and the convergence is faster. On the other hand, if $|\mathbf{R}|$ grows, the

measure is less reliable, the Kalman gain decreases, the predicted values are less affected by the current observation, and the convergence is slower.

Secondly, if a target shape or position abruptly changes, the observation may lie outside the tracker gate. In this case, a new Kalman filter is instantiated, and both, the old and the new one are now competing for the observations. Therefore, data association after occlusion should be handled using additional, high level processing.

## 6 Conclusions

In this work a principle and structured system is presented in an attempt to take a step towards solving the numerous difficulties which appear in unconstrained tracking applications. We use a structured framework in order to accomplish it: this proposal has presented a system which is modular and hierarchically organised. It consists in two levels, working in cascade, which are defined according to the different functionalities to be performed. A robust tracking is achieved in a non-friendly environment. The method is adaptive is the sense of number of targets.

Future research will be done in every module. Thus, a recursive background model adaptation, or a multimodal pixel modelling —which copes with background in motion— would be an interesting segmentation module enhancement. Target representation can be improved by adopting a multi-layer approach which allows to taken into account deposited and removed background objects. This can enhance agent tracking during long-term occlusions. In addition, targets should be classified by distinguishing among people, vehicles and other objects in motion. Finally, initialisation should include a group segmentation method so that agents who enter the scene together could be segmented and independently tracked.

## Acknowledgements

This work has been supported by EC grant IST-027110 for the HERMES project and by the Spanish MEC under projects TIC2003-08865 and DPI-2004-5414. J. Andrade and J. Gonzàlez also acknowledge the support of Juan de la Cierva Postdoctoral fellowships from the Spanish MEC.

## References

[1] B. Anderson and J. Moore. *Prentice Hall*. Optimal Filtering, 1979.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on PFs for On-line Non-linear/Non-Gaussian Bayesian Tracking. *SP*, 50(2):174–188, 2002.

[3] Y. Bar-Shalom and T. Fortran. *Tracking and Data Association*. A. Press, 1988.

[4] C. Bregler. Learning and Recognising Human Dynamics in Video Sequences. In *CVPR, Puerto Rico*, pages 568–574. IEEE, 1997.

[5] R. Collins. Mean-shift Blob Tracking through Scale Space. In *CVPR, Madison, WI, USA*, volume 2, pages 234–240. IEEE, 2003.

[6] R. Collins, A. Lipton, and T. Kanade. A System for Video Surveillance and Monitoring. In *8th ITMRRS, Pittsburgh, USA*, pages 1–15. ANS, 1999.

[7] R. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. *PAMI*, 27(10):1631–1643, 2005.

[8] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *PAMI*, 25(5):564–577, 2003.

[9] J. Deutscher and I. Reid. Articulated Body Motion Capture by Stochastic Search. *IJCV*, 61(2):185–205, 2005.

[10] J. Gonzàlez. *Human Sequence Evaluation: The Keyframe Approach*. PhD thesis, UAB, Spain, 2004.

[11] T. Horprasert, D. Harwood, and L. Davis. A Robust Background Subtraction and Shadow Detection. In *4th ACCV, Taipei, Taiwan*, volume 1, pages 983–988, 2000.

[12] M. Isard and A. Blake. A Mixed State Condensation Tracker with Automatic Model Switching. In *6th ICCV, Bombay, India*, pages 107–112. IEEE, 1998.

[13] M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada*, volume 2, pages 34–41. IEEE, 2001.

[14] S. Julier and J. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. In *11th AeroSense, Orlando, Florida*, volume 3068, pages 182–193, 1997.

[15] R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *ASME–Journal of Basic Engineering*, 82(D):35–45, 1960.

[16] J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *IJCV*, 39(1):57–71, 2000.

[17] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *IVC*, 21(1):99–110, 2003.

[18] E. Wan and R. van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *AS-SPCC, Lake Louise, Canada*, pages 153–158. IEEE, 2000.