

Online Viterbi Optimisation for Simple Event Detection in Video

Håkan Ardö
Centre for Mathematical Sciences
Lund University
Lund, Sweden

Kalle Åström, Rikard Berthilsson
Centre for Mathematical Sciences
Lund University
Lund, Sweden

Abstract

In this paper the problem of event detection in image sequences involving situations with multiple objects present is studied. A Hidden Markov Model describing the movements of a varying number of objects as well as their entries and exits is presented. Previously similar models have been used, but in online systems the standard dynamic programming Viterbi algorithm is typically not used to find the global optimum state sequence, as it requires that all past and future observations are available. In this paper we use an extension to the Viterbi algorithm that allows it to operate on infinite time sequences and produce the optimum state sequence with only a finite delay. This makes it possible to use the Viterbi algorithm in real time applications. Restrictions on where objects might enter or exit the scene is placed in the model, which makes the detection of some simple events trivial. We also show that extending the observation model to use several overlapping or non-overlapping cameras is straight forward.

1. Introduction

Detecting simple events, such as a pedestrian entering a shop, or a vehicle turning left at an intersection, can be done by using a robust tracker. By placing restrictions on the tracker, such as vehicles may only appear at the borders of the intersections and not in the middle of the intersections, the event detection will be a simple matter of checking the endpoints of the tracks produced. In this paper we present a tracker that uses online Viterbi optimisation to find the set of tracks with maximum likelihood conforming to restrictions as the one mentioned above.

The problem of tracking moving objects has been studied for a long time, see for example [20, 8]. Two main approaches are commonly used. Either a set of hypothesis are generated and tested against the observed image [10, 16, 12, 3], or methods for detecting objects in single frames, are combined with dynamic models in order to gain robustness [19, 22, 21, 6].

In this paper an online system is presented that tracks a

varying number of multiple moving objects. The entire state space is modelled by a Hidden Markov Model (HMM) [18], where each state represents a configuration of objects in the scene and the state transitions represents object movements as well as the events of objects entering or leaving the scene. The solution is found by optimising the observation likelihood over different state sequences. Results are generated with several frames delay in order to incorporate information from both past and future frames in the optimisation.

1.1. Relation to previous work

A classical solution to generate tracks from object detections is Kalman filtering [13], but since it is a linear model that assumes Gaussian probabilities it often fails in heavy clutter. In those cases particle filtering [10, 16] are often preferred as it allows multiple hypothesis to be maintained.

When several objects are considered, one model for each tracked object is often used. However, the data association problem [20] of deciding which detections should be used to update which models has to be solved. This is done by the MHT [19], for the Kalman filtering framework, where all possible associations are considered. Less likely hypotheses are pruned as the number of possible associations increases exponentially with time. This exponential increase is avoided in the JPDAF [22] by presuming the associations at each time step to be independent from associations of the previous time step. The complexity can be lowered even further by also assuming independence among the different associations at one time step, which is done by the PMHT [21]. Here the data association problem does not have to be solved explicitly in every frame. Instead the probability of each measurement belonging to each model is estimated.

The problems of detecting when objects enter or exit the scene has to be solved separately in the cases above. When using a single model for all objects, as proposed in this paper, neither the data association problem, nor the entry/exit problems has to be solved explicitly in every frame. Instead we optimise over all possible sequences of solutions over time. In [6] the PMHT is extended with the notion of track

visibility to solve the problem of track initiation. However, their system is still based on the creation of candidate tracks that may not be promoted to real tracks, but they will still influence other real tracks.

Extending the particle filter to handle multiple tracks is not straight forward and several versions have been suggested. In [9] a fixed number of objects is assumed, which means that only the data association problem is handled and not the entry and exit problems. The problem of reinitialisation is addressed in [11]. A fixed number of objects is still assumed, but it can abandon the current track if a better candidate track is discovered. In [12] a similar state space to the one proposed in this paper is used. That space is parametrised with a varying number of continuous parameters specifying the state of each object.

One problem with the particle filter is that the contribution of previous observations to the current state distribution is represented by the number of particles. This means that the precision of this probability is limited by the number of particles used. This becomes a problem in cases where the numerical differences between different observation likelihoods are big, which is the case for the observation model suggested in this paper. In these situations the particle filter will almost always sample from the most likely particle, and thus becomes a greedy algorithm that does not represent multiple hypotheses anymore.

An alternate approach is to discretize the state space and use an HMM to represent the dynamics. In that case all calculations can be performed with log-likelihoods, and are thus able to represent huge numerical differences.

This has previously been suggested by [8] where a state space is exhaustively searched for an optimum in each frame. However the authors assume a known positions in the previous frame. In another approach [5] the discretizing grid is repositioned in each frame, centred at the current best estimate with its mesh size and directions given as the eigenvalues and eigenvectors of the error covariance. More recently, [4] shows, in the single object case, that a particle filter is capable of matching the performance of an HMM tracker [3] at a fraction of the computational cost. However in [15] it is shown that by placing some restrictions on the HMMs the computational complexity can be reduced from $O(n^2)$ to $O(n)$, and that HMMs with 100,000 states can be used for real time tracking. In both these approaches fairly densely discretized state spaces are used. We show in this work that state spaces discretized more sparsely can be used.

Most real-time HMM-base trackers [15], [14] and [7] do not use the standard Viterbi dynamic programming algorithm [18], which finds the global maximum likelihood state sequence, as this requires the entire set of future observations to be available. Instead they estimate the state posterior distribution given the past observations only. The

particle filter also results in this kind of posterior state distribution, which means that both the particle filter and this kind of HMM trackers suffer from the problem of trying to estimate the single state of the system from this distribution. Later processing stages or data-displays usually requires a single state and not an distribution.

Common ways to do this is to estimate the mean or the maximum (MAP) of this posterior distribution, but this have a few problems:

1. A mean of a multimodal distribution is some value between the modes. The maximum might be a mode that represents a possibility that is later rejected. We propose to instead use optimisation that considers future observation and thereby chooses the correct mode.
2. In the multi object case the varying dimensionality of the states makes the mean value difficult to define. In [12] it is suggested to threshold the likelihood of each object in the configurations being present. Then the mean state of each object for which this likelihood is above some threshold is calculated separately.
3. Restrictions placed in the dynamic model, such as illegal state transactions, are not enforced, and the resulting state sequence might contain illegal state transactions. For the particle filter also restrictions in the prior state distribution might be violated. In [12] for example, the prior probability of two objects overlapping in the 3D scene is set to zero as this is impossible. However the output mean state value may still be a state where two objects overlap, as the two objects may originate from different sets of particles. In our approach impossible states or state transactions will never appear in the results.

In [1] we suggested a novel modification to the Viterbi algorithm [18], that allows it to be used on infinite time sequences and still produce the global optimum. In this paper that method is extended to multiple cameras. The problem with the original Viterbi algorithm is that it assumes that all observations are available before any results can be produced. Our modification allows results to be computed before all observation are received, and still generates the same global optimum state sequence as is done when all observations are available. However, there is a delay of several frames between obtaining an observation and the production of the optimum state for that frame.

In [17] it is suggested to calculate the current state by applying the Viterbi algorithm to a fixed size time slice looking into the future and only storing first state of the solution, but this only gives approximative solutions.

Another problem is that when considering multiple objects, the state space becomes huge. Typically at least some 10000 states is needed for a single object, and to be able to

track N objects simultaneously that means 10000^N states. An exact solution is no longer possible for real time applications. In [1] we present two possible approximations that can be used to compute results in real time. Either use multiple small spatially overlapping HMM's, or only evaluate the M most likely states in each frame.

Also, we show how it is possible, for the later of the two approximations, to assess whether this approximation actually found the global optimum or not. This could be useful in an off line calibration of the approximation parameters.

1.2. Paper overview

The paper is organised as follows. The theory behind hidden Markov models is describe in Section 2. This includes our extensions from [1] to handle infinite time sequences, Section 2.2, and infinite state spaces, Section 2.3. Section 3 describes our proposal of how to use the HMM for object tracking, including single object tracking, Section 3.1, multi object tracking, Section 3.2 and multi camera setup, Section 3.4. Finally Section 4 gives experimental verification.

2. Hidden Markov models

A hidden Markov model is defined [18] as a discrete time stochastic process with a set of states, $S = S_0, \dots, S_N$ and a constant transitional probability distribution $a_{i,j} = p(q_{t+1} = S_j | q_t = S_i)$, where $Q = (q_0, \dots, q_T)$ is a state sequence for the time $t = 0, 1, \dots, T$. The initial state distribution is denoted $\pi = (\pi_0, \dots, \pi_N)$, where $\pi_i = p(q_0 = S_i)$. The state of the process cannot be directly observed, instead some sequence of observation symbols, $O = (O_0, \dots, O_T)$ are measured, and the observation probability distribution, $b_j(O_t) = b_{j,t} = p(O_t | q_t = S_j)$, depends on the current state. The Markov assumption gives that

$$p(q_{t+1} | q_t, q_{t-1}, \dots, q_0) = p(q_{t+1} | q_t) \quad (1)$$

and the probability of the observations satisfies

$$p(O_t | q_t, q_{t-1}, \dots, q_0) = p(O_t | q_t). \quad (2)$$

2.1. Viterbi optimisation

From a hidden Markov model $\lambda = (a_{i,j}, b_j, \pi)$ and an observation sequence, O , the most likely state sequence, $Q^* = \operatorname{argmax}_Q p(Q | \lambda, O) = \operatorname{argmax}_Q p(Q, O | \lambda)$, to produce O can be determined using the classical Viterbi optimisation [18] by defining

$$\delta_t(i) = \max_{q_0, \dots, q_{t-1}} p(q_0, \dots, q_{t-1}, q_t = S_i, O_0, \dots, O_t). \quad (3)$$

For $t = 0$, $\delta_0(i)$ becomes $p(q_0 = S_i, O_0)$, which can be calculated as $\delta_0(i) = \pi_i b_{i,0}$, and for $t > 0$ it follows that $\delta_t(i) = \max_j (\delta_{t-1}(j) a_{j,i}) \cdot b_{i,t}$. By also keeping track of

$\psi_t(i) = \operatorname{argmax}_j (\delta_{t-1}(j) a_{j,i})$ the optimal state sequence can be found by backtracking from $q_T^* = \operatorname{argmax}_i \delta_T(i)$, and letting $q_t^* = \psi_{t+1}(q_{t+1}^*)$ for $t < T$.

2.2. Infinite time sequences

To handle the situations where $T \rightarrow \infty$ consider any given time $t_1 < T$. The observation symbols O_t , for $0 \leq t \leq t_1$, have been measured, and $\delta_t(i)$ as well as $\psi_t(i)$ can be calculated. The optimal state for $t = t_1$ is unknown. Consider instead some set of states, Θ_t , at time t such that the global optimum $q_t^* \in \Theta_t$. For time t_1 this is fulfilled by letting $\Theta_{t_1} = S$, the entire state space. For Θ_t , $t < t_1$, shrinking sets of states can be found by letting Θ_t be the image of Θ_{t+1} under ψ_{t+1} , such that

$$\Theta_t = \{S_i | i = \psi_{t+1}(j) \text{ for some } S_j \in \Theta_{t+1}\}. \quad (4)$$

If the dependencies of the model is sufficiently localised in time, then for some time $t_2 < t_1$, there will be exactly one state $q_{t_2}^*$ in Θ_{t_2} , and the optimal state q_t^* for all $t \leq t_2$ can be obtained by backtracking from $q_{t_2}^*$. No future observations made can alter the optimum state sequence for $t \leq t_2$.

2.3. Infinite state spaces

The problem with using the Viterbi optimisation for large state spaces is that $\delta_t(i)$ has to be calculated and stored for all states i at each time t . By instead only storing the M largest $\delta_t(i)$ and an upper bound, $\delta_{\max}(t)$ on the rest, significantly less work is needed. If M is large enough, the entire optimal state-sequence might be found by backtracking among the stored states. It is also possible to verify if this is the case or not for a given example sequence and a given M . If the global optimum were not found, then M could be increased and the algorithm executed again, or an approximative solution could be found among the stored states. Typically the algorithm is executed off-line for some example sequences to decide how large an M is needed, and then when running live this value is used and approximative solutions are found. The details of this algorithm and a proof of it's correctness is presented in [1].

3. Using HMM for tracking

3.1. Single object tracking

An HMM such as described above can be used for tracking objects in a video sequence produced by a stationary camera. Initially we assume that the world only contains one mobile object and that this object sometimes is visible in the video sequence and sometimes located outside the scene.

The state space of the HMM, denoted S^1 , is constructed from a finite set of grid points $X_i \in \mathbb{R}^2$, $j = 1, \dots, N$ typically spread in a homogeneous grid over the image. The

state S_i represents that the mass centre of the object is at position X_i in the camera coordinate system. A special state S_0 , representing the state when the object is not visible, is also needed.

The observation symbols of this model will be a binary background/foreground image, $O_t : \mathbb{R}^2 \rightarrow \{0, 1\}$, as produced by for example [2]. By analysing the result of the background/foreground segmentation algorithm on a sequence with known background and foreground, the constant probabilities

$$p_{fg} = p(O_t(x) = 1 | x \text{ is a foreground pixel}) \quad (5)$$

and

$$p_{bg} = p(O_t(x) = 0 | x \text{ is a background pixel}) \quad (6)$$

can be calculated. Typically these are well above 1/2, and it is here assumed that they are constant over time and does not depend on x .

The shape of the object when located in state S_i , can be defined as the set of pixels, C_{S_i} , that the object covers when centred in at this position. This shape can be learnt from training data off line. As there is only one object in the world, when the HMM is in state S_i , the pixels in C_{S_i} are foreground pixels and all other pixels are background pixels. The probability, $b_{i,t} = p(O_t | q_t = S_i)$, of this is

$$b_{i,t} = \prod_{x \in C_{S_i}} [O_t(x)p_{fg} + (1 - O_t(x))(1 - p_{fg})] \cdot \prod_{x \notin C_{S_i}} [(1 - O_t(x))p_{bg} + (O_t(x))(1 - p_{bg})] \quad (7)$$

and thereby all parts of the HMM are defined.

3.2. Multi object HMMs

To generalise the one object model in the previous section into two or several objects is straight forward. For the two object case the states become $S_{i,j} \in S^2 = S \times S$ and the shapes, $C_{S_{i,j}} = C_{S_i} \cup C_{S_j}$. The transitional probabilities become $a_{i_1 j_1 i_2 j_2} = a_{i_1 i_2} \cdot a_{j_1 j_2}$.

Solving this model using the Viterbi algorithm above gives the tracks of all objects in the scene, and since there is only one observation in every frame, the background/foreground segmented image, no data association is needed. Also, the model states contain the entry and the exit events, so this solution also gives the optimal entry and exit points.

There is however one problem with this approach. The number of states increases exponentially with the number of objects and in practice an exact solution is only computationally feasible for a small number of objects within a small region of space.

3.3. Calculating $b_{\max}(t)$

To use the method described in Section 2.3, some estimate, $b_{\max}(t)$, as low as possible, has to be found such that

$$b_{\max}(t) \geq b_{i,t} \text{ for } i \notin H_t \quad (8)$$

given the observation symbol O_t . Where H_t is the set of the M largest states stored at time t . The estimate will be derived for any number of objects, e.g. in the state space S^n , for the case when objects are not allowed to overlap. Each state S_i is a subset of the set of possible object centre points $X = \{X_j\}$. S_0 is the empty set. But not all subsets of X has to be evaluated. Consider any state S_i and form $S_j = S_i \cup \{X_k\}$ by adding one object at point X_k . If X_k don't overlap any of the objects present in S_i the likelihood $b_{j,t} = b_{i,t} b_X(k)$, where

$$b_X(k) = \prod_{x \in C_{X_k}} \left(O_t(x) \frac{p_{fg}}{1 - p_{bg}} + (1 - O_t(x)) \frac{1 - p_{fg}}{p_{bg}} \right) \quad (9)$$

and C_{X_k} is the pixels cover by an object centred at X_k . This means that if overlapping objects were not allowed only points X_k where $b_X(k) > 1$ could increase the likelihood, and thus only such points had to be considered. $b_X(k)$ can be computed fast for all k using integral images and then the likelihood of all subsets of the points $\{X_k | b_X(k) > 1\}$ can be evaluated and the maximum $i \notin H_t$ can be found.

3.4. Using multiple cameras

Extending this to multiple overlapping or non-overlapping cameras is straight forward. By calibrating the set of cameras and identifying a common coordinate system for the ground plane, the objects centres, X_k , can be modelled as moving in this common coordinate system. Thereby a single HMM modelling the events on this ground plane can be used. The observations for this model is the images from all the cameras. Using the calibration of the cameras, each centre point can be projected into the shape of the object in each of camera images $C_{X_k}^c$ where $c = 1, 2, \dots$, represents the different cameras. $C_{X_k}^c$ might be the empty set if an object at position X_k is not visible in camera c . By indexing Equation 7 on the camera c , with O_t^c the background/foreground image produced from camera c ,

$$b_{i,t}^c = \prod_{x \in C_{S_i}^c} [O_t^c(x)p_{fg} + (1 - O_t^c(x))(1 - p_{fg})] \cdot \prod_{x \notin C_{S_i}^c} [(1 - O_t^c(x))p_{bg} + (O_t^c(x))(1 - p_{bg})], \quad (10)$$

and the total observation probability

$$b_{i,t} = \prod_c b_{i,t}^c. \quad (11)$$

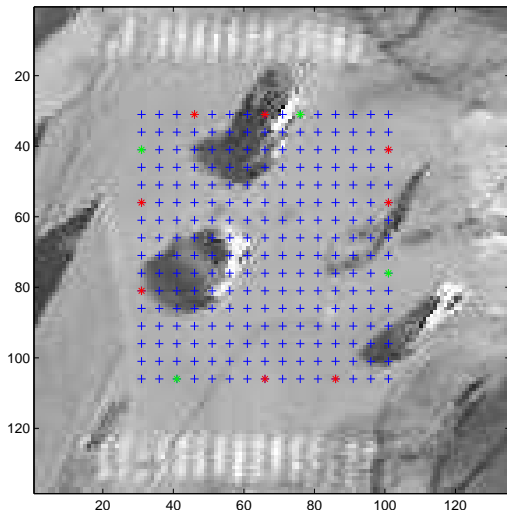


Figure 1: Object centre points in a single model. The red stars are the possible starting points and the green stars are the possible exit points.

4. Experiments

4.1. Traffic

For automatic analysis of traffic surveillance the first step is often to extract the trajectories of the vehicles in the scene. The infinite state space method described in Section 2.3 with $M = 20$ were tested on this problem by analysing a 5 minutes surveillance video, acquired by a Axis 2120 web camera, Figure 1. The sequence contained in total 40 vehicles, including some larger than the normal car and one bus. The shape of the objects is assumed to be squares approximately the size of the cars. The full tracks of all the 40 vehicles where correctly extracted. These tracks could be used to count the number of vehicles turning left/right or going straight through the intersection, by simply checking their start and end-points which are guaranteed to be one of the start/end points marked in Figure 1. In addition to those correct tracks 5 more were discovered. Two due to groups of bicycles, and 3 due to larger vehicles, including the bus. The main reason for those failures is the assumption that all objects in the scene have the same shape, which is not the case here. A solution to that would be to have several different object types of different sizes in the model. This run were made in matlab at 0.78 fps (0.75 if $b_{\max}(t)$ also were calculated to evaluate whether the global maximum were reached). Tests with varying M showed that the global maximum is found with as small values as $M = 20$. This might seem unbelievable small, but note that this is the 20 most likely states give the entire history, not 20 random samples as compared to the particle filter. Also, object centres are quite sparsely sampled, keeping the number of possibilities low, and at each time interval typically some

200-300 states (all states reachable from the stored 20) are evaluated and compared to find the top 20.

4.2. Occlusion

To test how well the system could handle occlusions, another test were performed using a camera overlooking a corridor with significant perspective effects and occlusions, cf. Figure 2. The camera where calibrated and the world coordinate system registered to a blueprint of corridor. The set of possible object centre point X_k , were generated as an regular grid in the blueprint. For each of the centre points the region of the image a pedestrian located at that point would cover could be calculated from the calibration. Entry and exit points were placed around the doors and along the bottom half of the image.

The sequence is 1:45 min and contains 9 events of people walking through the scene in different ways. All of them were correctly detected. But because of noise from the opening and closing of the doors and reflections a few short erroneously tracks were generated between the entry and exit points belonging to the same door, but they were all easily filter out afterwards by removing all short tracks starting and ending at the same door. No other errors were made.

5. Conclusions

In this paper we have proposed a multi HMM model to model multiple targets in one single model with the advantage of solving all the problems of a multi target tracker by a Viterbi optimisation. This includes track initiation and termination as well as the model updating and data association problems. Furthermore two extensions to the standard Viterbi optimisation are used that allows the method to be used in real-time applications with infinite time sequences and infinite state spaces. The later extension only gives approximative solutions in the general case, but can also determine if an exact solution were found.

References

- [1] H. Ardö, R. Berthilsson, and K. Åström. Real time viterbi optimization of hidden markov models for multi target tracking. In *IEEE Workshop on Motion and Video Computing*, 2007.
- [2] Håkan Ardö and Rikard Berthilsson. Adaptive background estimation using intensity independent features. *Proc. British Machine Vision Conference*, 03, 2006.
- [3] Marcelo G.S. Bruno and Jose M.F. Moura. Multiframe detector/tracker: Optimal performance. *IEEE Transactions on Aerospace and Electronic Systems*, 37(3):925–946, 2001.

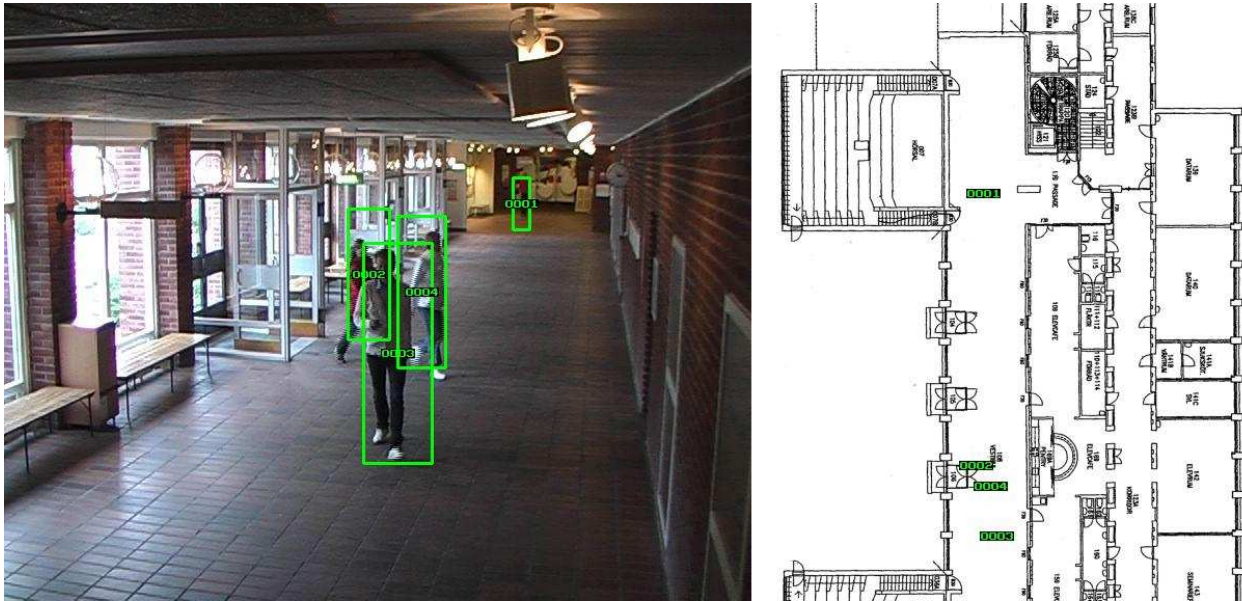


Figure 2: Result from corridor tracking example. 4 pedestrians are present and correctly detected. Each is marked in the blueprint with their id at their centre position as well as in the image with a box representing areas that should be foreground.

[4] M.G.S. Bruno. Sequential importance sampling filtering for target tracking in image sequences. *IEEE Signal Processing Letters*, 10(8):246–249, 2003.

[5] R. S. Bucy and K. D. Senne. Digital synthesis of non-linear filters. *Automatica*, 7:287–298, 1971.

[6] S. J. Davey, D. A. Gray, and S. B. Colegrove. A markov model for initiating tracks with the probabilistic multi-hypothesis tracker. *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, 1:735–742 vol.1, 2002.

[7] Mei Han, Wei Xu, Hai Tao, and Yihong Gong. An algorithm for multiple object trajectory tracking. *cvpr*, 01:864–871, 2004.

[8] David Hogg. Model-based vision: a program to see a walking person. *Image and vision computing*, 1(1):5–20, 1983.

[9] C. Hue, J.-P. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(3):791–812, 2002.

[10] M. Isard and A. Blake. A visual tracking by stochastic propagation of conditional density. In *4th European Conf. Computer Vision*, 1996.

[11] Michael Isard and Andrew Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. *Lecture Notes in Computer Science*, 1406:893–908, 1998.

[12] Michael Isard and John MacCormick. Bramble: A bayesian multiple-blob tracker. In *ICCV*, pages 34–41, 2001.

[13] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82-D:35–45, 1960.

[14] Jien Kato, T. Watanabe, S. Joga, Ying Liu, and H. Hase. An hmm/mrf-based stochastic framework for robust vehicle tracking. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):142–154, 2004.

[15] Javier Movellan, John Hershey, and Josh Susskind. Real-time video tracking using convolution hmms. In *CVPR*, 2004.

[16] Gordon N.J., Salmond D.J., and Smith A.F.M. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.

[17] Maurizio Pilu. Video stabilization as a variational problem and numerical solution with the viterbi method. *cvpr*, 01:625–630, 2004.

[18] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[19] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transaction on Automatic Control*, 24(6):843–854, 1979.

[20] R. W. Sittler. An optimal data association problem in surveillance theory. *IEEE Transactions on Aerospace and Electronic Systems*, MIL-8(Apr):125–139, 1964.

[21] R. L. Streit and T. E. Luginbuhl. Probabilistic multi-hypothesis tracking. Technical Report 10428, NUWC, Newport, RI, 1995.

[22] T. Fortmann Y. Bar-Shalom and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In *Conf. on Information Sciences and Systems*, 1980.