

# Learning Higher-Order Markov Models for Object Tracking in Image Sequences

Michael Felsberg and Fredrik Larsson\*  
{mfe, larsson}@isy.liu.se

Department of Electrical Engineering, Linköping University

**Abstract.** This work presents a novel object tracking approach, where the motion model is learned from sets of frame-wise detections with unknown associations. We employ a higher-order Markov model on position space instead of a first-order Markov model on a high-dimensional state-space of object dynamics. Compared to the latter, our approach allows the use of marginal rather than joint distributions, which results in a significant reduction of computation complexity. Densities are represented using a grid-based approach, where the rectangular windows are replaced with estimated smooth Parzen windows sampled at the grid points. This method performs as accurately as particle filter methods with the additional advantage that the prediction and update steps can be learned from empirical data. Our method is compared against standard techniques on image sequences obtained from an RC car following scenario. We show that our approach performs best in most of the sequences. Other potential applications are surveillance from cheap or uncalibrated cameras and image sequence analysis.

## 1 Introduction

Object tracking is a common vision problem that requires temporal processing of visual states. Assume that we want to estimate the position of an object moving in 3D space, given its observed and extracted position (*i.e.* coordinates) in 2D image data, taken from an uncalibrated moving camera. Our focus is on temporal filtering, however, this problem is specific to vision-based tracking since the association problem between visual detections and objects does not exist in many classical sensors, *e.g.*, accelerometers.

The output of the proposed method is 2D trajectories of physical objects. The objects' dynamics are assumed to be unknown and non-linear and the noise terms non-Gaussian. This setting constitutes a hard, weakly-supervised learning problem for the motion and measurement models since no point-to-point correspondences between the observations are available. Once learned, the motion models are applied in a Bayesian tracking framework to extract trajectories from sequences of sets of detections, *i.e.*, also solving the association problem between detections and objects.

The major advantage of our approach compared to other learning methods is that sets of frame-wise detections with unknown correspondences are much easier to extract than strictly corresponding detections or fully stable tracking of object appearances. We

---

\* The research leading to these results has received funding from the European Community's 7th Framework Programme (FP7/2007-2013) under grant agreement n° 215078 DIPLECS.

employ a higher-order Markov model on position space instead of a first-order Markov model on a high-dimensional state-space of object dynamics. Compared to the latter, our approach allows the use of marginal rather than joint distributions, which results in a significant reduction of computation complexity. Densities are represented using a grid-based method, where the rectangular windows are replaced with a smooth Parzen window estimator sampled at the grid points, where *sampling* is meant in the signal processing sense (*i.e.* not stochastic sampling) throughout this paper. This method is as accurate as particle filter methods [1] with the additional advantage that the prediction and update steps can be learned from empirical data. The densities are estimated and processed in the channel representation and thus the employed tracking approach is called channel-based tracking (CBT).

### 1.1 Related Work

Relevant literature research can be found in the area of non-linear, non-Gaussian Bayesian tracking [2, 3]. In Bayesian tracking, the current state of the system is represented as a probability density function of the system's state space. At the time update, this density is propagated through the system model and an estimate for the prior distribution of the system state is obtained. At the measurement update, measurements of the system are used to update the prior distribution, resulting in an estimate of the posterior distribution of the system state.

Gaussian, (non-)linear Bayesian tracking is covered by (extended) Kalman filtering. Common non-Gaussian approaches are particle filters and grid-based methods [2]. Whereas particle filters apply Monte Carlo methods for approximating the relevant density function, grid based methods discretize the state-space, *i.e.*, apply histogram methods for the approximation. In the case of particle filters, densities are propagated through the models by computing the output for individual particles. Grid-based methods use discretized transition maps to propagate the histograms and are closely related to Bayesian occupancy filtering [4].

An extension to grid based methods is to replace the rectangular histogram bins with overlapping, smooth Parzen windows, that are regularly sampled. This method is called *channel-based tracking* [1]. CBT implements Bayesian tracking using channel representations [5] and linear mappings on channel representations, so-called associative networks [6]. The main advantage compared to grid-based methods is the reduction of quantization effects and computational effort. Also, it has been shown that associative networks can be trained from data sets with unknown element-wise correspondence [7].

As pointed out above, channel representations are sampled Parzen window estimators [8], implying that CBT is related to kernel-based prediction for Markov sequences [9]. In the cited work, system models are estimated in a similar way as in CBT, but the difference is that sampled densities make the algorithm much faster. Another way to represent densities in tracking are Gaussian mixtures (*e.g.* [10]) and models based on mixtures can be learned using the EM algorithm, cf. [11], although the latter method is restricted to uni-modal cases (Kalman filter) and therefore disregarded.

A vision-specific problem in tracking is the associations of observations and objects, in particular in multiple object tracking [12]. Standard solutions are probabilistic multiple-hypothesis tracking (PMHT) [13] and the probabilistic data association filter

(PDAF) [14]. Thus, in our experiments, we have been comparing our approach to a combination of PMHT and a set of Kalman filters, based on an implementation of [15] and [16], and our own implementation of PDAF.

The main novelties of this paper compared to the approach of CBT as defined in [1] is: 1: The multi-dimensional state-space is embedded in a probabilistic formulation (previous work only considered a 1D state and just concatenated channel vectors, leading to sums of densities). 2: The higher-order Markov model for the CBT is embedded into a first order model. This allows to use the Baum-Welch algorithm to learn models from datasets without known associations. 3: The Baum-Welch algorithm has been adapted to using channels. 4: The tracking is applied for visual tracking among multiple objects in a moving camera, and compared to PMHT and PDAF.

## 1.2 Organization of the Paper

After the introduction, the methods required for further reading are introduced: Bayesian tracking, channel representations of densities, and CBT. The novelties of this paper are covered in Section 3: First, probabilistic multi-dimensional formulations for CBT are considered. Second, the CBT method is extended to embed the higher-order Markov model into a first order model and we show that it is sufficient to use the marginals of a higher-order Markov model to track multiple objects. Third, we adapt the Baum-Welch algorithm to the CBT formulation. Fourth, we provide empirical evidence that correspondence-free learning works with the Baum-Welch algorithm applied to the first-order model embedding. In section 4, the whole algorithm is evaluated on image sequences acquired from a RC car. In section 5 we discuss the achieved results.

## 2 Channel-Based Bayesian Tracking

Channel-based tracking (CBT) is a generalization of grid-based methods for implementing non-linear, non-Gaussian Bayesian tracking. Hence we give a brief overview on Bayesian tracking and channel representations before we describe CBT. The material of this section summarizes the material from [1].

### 2.1 Bayesian Tracking

For the introduction of concepts from Bayesian tracking we adopt the notation from [2]. Bayesian tracking is commonly defined in terms of a process model  $\mathbf{f}$  and a measurement model  $\mathbf{h}$ , distorted by i.i.d. noise  $\mathbf{v}$  and  $\mathbf{n}$

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \quad \mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) . \quad (1)$$

The symbol  $\mathbf{x}_k$  denotes the system state at time  $k$  and  $\mathbf{z}_k$  denotes the observation made at time  $k$ . Both models are in general non-linear and time-dependent.

The current state is estimated in two steps. First, given the posterior density of the previous state and all previous observations are known and assuming a Markov process of order one, the prior density of the current state is estimated in the time update as

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} . \quad (2)$$

Second, the posterior is obtained from the measurement update as

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) / \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k . \quad (3)$$

In the case of non-linear problems with multi-modal densities, two approaches for implementing (2) and (3) are commonly used: The particle filter and the grid-based method. Since CBT is a generalization of the grid-based method, we focus on the latter. Grid-based methods assume a discrete state space such that the continuous densities are approximated with histograms. Thus, conditional probabilities of state transitions are replaced with linear mappings. In contrast to [2] where densities were formulated using Dirac distributions weighted with discrete probabilities, we assume band-limited densities and apply sampling theory, since this is more consistent with the formulation of CBT. Sampling the densities  $p(\mathbf{x}_{k_1} | \mathbf{z}_{1:k_2})$  gives us

$$w_{k_1|k_2}^i \triangleq p(\mathbf{x}_{k_1} | \mathbf{z}_{1:k_2}) * \delta(\mathbf{x}^i - \mathbf{x}_{k_2}) \quad k_1, k_2 \in \{k-1, k\} \quad (4)$$

where  $*$  denotes convolution and  $\delta(\mathbf{x}^i - \mathbf{x})$  is the Dirac impulse at  $\mathbf{x}^i$ . Combining (2) and (4) and applying the power theorem gives us

$$w_{k|k-1}^i = \sum_j f_k^{ij} w_{k-1|k-1}^j \quad (5)$$

where  $f_k^{ij} = p(\mathbf{x}_k | \mathbf{x}_{k-1}) * \delta(\mathbf{x}^i - \mathbf{x}_k) * \delta(\mathbf{x}^j - \mathbf{x}_{k-1})$ . Accordingly, combining (3) and (4) results in

$$w_{k|k}^i = h_k^i(\mathbf{z}_k) w_{k|k-1}^i / \sum_j h_k^j(\mathbf{z}_k) w_{k|k-1}^j \quad (6)$$

where  $h_k^i(\mathbf{z}_k) = p(\mathbf{z}_k | \mathbf{x}_k) * \delta(\mathbf{x}^i - \mathbf{x}_k)$ . Grid-based methods require the more samples the higher the upper band limit of the pdf, *i.e.*, the wider the characteristic function  $\varphi_{\mathbf{x}}(\mathbf{t}) = E\{\exp(it^T \mathbf{x})\}$ .

## 2.2 Channel Representations of Densities

The channel representation [5, 17] can be considered as a way of sampling continuous densities or, alternatively, as histograms where the bins are replaced with smooth, overlapping basis functions  $b(\mathbf{x})$ , see *e.g.* [18]. Consider a density function  $p(\mathbf{x})$  as a continuous signal that is sampled with a smooth basis function, *e.g.*, a B-spline. It is important to realize here that the sampling takes place in the dimensions of the stochastic variables, not along the time axis  $k$ . It has been shown in the literature that an averaging of a stochastic variable in channel representation is equivalent to the sampled Parzen window (or kernel density) estimator with the channel function as kernel function [8]. For the remainder of this paper it is chosen as [19]

$$b(\mathbf{x}) \triangleq \frac{2a}{\pi} \prod_n \cos^2(ax_n) \quad \text{if } |x_n| < \frac{\pi}{2a}, \quad 0 \quad \text{otherwise.} \quad (7)$$

Here  $a$  determines the relative width, *i.e.*, the sampling density. For the choice of  $a$  the reader is referred to [20]. According to [8], the channel representation reduces the quantization effect compared to ordinary histograms by a factor of up to 20. Switching from histograms to channels allows us to reduce computational load by using fewer bins, to increase the accuracy for the same number of bins, or a mixture of both.

For performing maximum likelihood or maximum a posteriori (MAP) estimation using channels, a suitable algorithm for extracting the maximum of the represented distribution is required. For  $\cos^2$ -channels with a spacing of  $\frac{\pi}{3a}$ , an optimal algorithm in least-squares sense is obtained in the one-dimensional case as [19]

$$\hat{x}_{k_1} = l + \frac{1}{2a} \arg \left[ \sum_{j=l}^{l+2} w_{k_1|k_1}^j \exp(i2a(j-l)) \right]. \quad (8)$$

$N$ -dimensional decoding is obtained by local marginalization in a window of size  $3^N$  and subsequent decoding of the  $N$  marginals. The index  $l$  of the decoding window is chosen using the maximum sum of a consecutive triplet of coefficients:  $l = \arg \max_j (w_{k_1|k_1}^j + w_{k_1|k_1}^{j+1} + w_{k_1|k_1}^{j+2})$ .

### 2.3 Channel-Based Tracking

Channel-based tracking (CBT) is defined by replacing the sampled densities (4) with

$$w_{k_1|k_2}^i \triangleq p(\mathbf{x}_{k_1} | \mathbf{z}_{1:k_2}) * b(\mathbf{x}^i - \mathbf{x}_{k_1}) \quad (9)$$

where  $b(\mathbf{x})$  is the channel basis function (7). The power theorem which has been used to derive (5) and (6) does not hold in general if we sample with channels instead of impulses, because some high-frequency content might be removed. However, if the densities are band-limited from the start, the regularization by the channel basis functions removes no or only little high-frequency content and (5) and (6) can be applied for the channel-based density representations as well.

For what follows, the coefficients of (5) are summarized in the matrix  $\mathbf{F}_k = \{f_k^{ij}\}$  and the coefficients of (6) are summarized in the vector-valued function  $\mathbf{h}_k(\mathbf{z}_k) = \{h_k^j(\mathbf{z}_k)\}$ . Both operators can be learned from a set of training data if both remain stationary and we remove the time index  $k$  (not from  $\mathbf{z}_k$  though):  $\mathbf{F}$  and  $\mathbf{h}(\mathbf{z}_k)$ . The prior and posterior densities are now obtained by

$$\mathbf{w}_{k|k-1} = \mathbf{F} \mathbf{w}_{k-1|k-1}, \quad \mathbf{w}_{k|k} = \mathbf{h}(\mathbf{z}_k) \cdot \mathbf{w}_{k|k-1} / \mathbf{h}^T(\mathbf{z}_k) \mathbf{w}_{k|k-1}, \quad (10)$$

where  $\cdot$  is the element-wise product, *i.e.*, the enumerator remains a vector.

In [1] the system model  $\mathbf{f}$  is learned by estimating the matrix  $\mathbf{F}$  from the covariance of the state channel vector. Since the model matrix corresponds to the conditional pdf and not to the joint pdf, the covariance is normalized with the marginal distribution for the previous state (see also [21], plugging (3.3) into (2.7))

$$\hat{\mathbf{F}} = \sum_{k=1}^{K_{\max}} \mathbf{w}_{k|k} \mathbf{w}_{k-1|k-1}^T / \mathbf{1} \sum_{k=1}^{K_{\max}} \mathbf{w}_{k-1|k-1}^T \quad (11)$$

where  $\mathbf{1}$  denotes a one-vector of suitable size and the quotient is evaluated point-wise. For the initial time step, no posterior of the previous state is available and the time update cannot be computed using the model matrix above. Instead, the empirical distribution for the initial state is stored and used as  $\mathbf{w}_{0|0}$ .

The measurement model  $\mathbf{h}(\mathbf{z}_k)$  and its estimation is not considered in more detail here, since our experiments are restricted to the case where  $\mathbf{z}_k$  are noisy and cluttered observations of  $\mathbf{w}_k$ . A more general case of observation models has been considered in [1]. In summary the algorithm is just iterating (10) and (8) over  $k$ .

### 3 Learning Higher-Order Markov Models

In this section, we generalize (10) for multi-dimensional input.<sup>1</sup> In a next step, we show why higher-order marginalized Markov models are suitable for tracking multiple objects. We describe further how they can be embedded in a first-order model in order to apply standard algorithms like the Baum-Welch algorithm [22]. Finally, we explain how these models can be learned from data without point-wise correspondences.

#### 3.1 Multi-Dimensional Case

Consider the conditional density of a certain state dimension  $m$  given  $N$  previous states and apply Bayes' theorem:

$$p(x_k^m | x_{k-1}^1, \dots, x_{k-1}^N) = p(x_{k-1}^1, \dots, x_{k-1}^N | x_k^m) p(x_k^m) / p(x_{k-1}^1, \dots, x_{k-1}^N). \quad (12)$$

Since channel representations of densities are closely related to robust statistics [8] and since robust matching of states allows to assume mutual independence of the old states  $x_{k-1}^1, \dots, x_{k-1}^N$  [23], we obtain

$$p(x_k^m | x_{k-1}^1, \dots, x_{k-1}^N) = \prod_{n=1}^N (p(x_{k-1}^n | x_k^m) / p(x_{k-1}^n)) p(x_k^m) \quad (13)$$

and applying Bayes' theorem once more results in

$$p(x_k^m | x_{k-1}^1, \dots, x_{k-1}^N) = p(x_k^m)^{1-N} \prod_{n=1}^N p(x_k^m | x_{k-1}^n). \quad (14)$$

Note that the new states  $x_k^m$  still depend on *all* old states but these conditional densities are computed by separable products of pairwise conditional densities and a proper normalization. This factorization is of central importance to avoid a combinatorial explosion while producing only a small approximation error.

A practical problem is, however, that the densities are represented by channels and repeatedly multiplying these representations will lead to extensive low-pass filtering of the true densities. Their product might not even be a valid channel vector!

Considering the basis functions (7) more in detail, it turns out that taking the square-root of the product of channel vectors is a good approximation of the channel representation of the corresponding density function product

$$(p_1 p_2) * b(\mathbf{x}^i) \approx \sqrt{(p_1 * b(\mathbf{x}^i))(p_2 * b(\mathbf{x}^i))} \quad (15)$$

and whenever multiplying densities in channel representation, we applied a square-root to the factors. This product is directly related to the Bhattacharyya coefficient [24].

<sup>1</sup> Note that the method proposed in [1], namely to concatenate channel vectors, is not correct in full generality.

### 3.2 Higher-Order Markov Models for Tracking

In order to have low computational complexity during runtime and a moderate number of learning samples during training, joint densities of high-dimensional state spaces should be avoided. Replacing joint densities with the respective marginals is possible in case of statistical independence, but in the practically relevant case of tracking multiple objects, using marginals means to mix up properties from different objects. Instead of having two objects with two different properties each, one ends up with four objects: The correct ones and two ghost objects with mixed properties, see below.

The approach to drastically reduce dimensionality is no option either. The state space should contain sufficiently many degrees of freedom to describe the observed phenomena. Typical choices for tracking are first or second order Euclidean motion states or higher-order Markov models, albeit less frequently used. Euclidean motion states appear more attractive when the system model is to be engineered, since we are used to thinking in physical models. However, this is not relevant when it comes to learning systems. Contrary, learning systems are easier to design when the in-data lives in the same space and hence we consider  $n$ -tuples of positions instead of motion states.

Actually, higher-order Markov models have an important advantage compared to motion states: In case of several moving objects, it is important to have correspondence between the different dimensions of the motion state, *i.e.* to bind it to a particular physical object. Otherwise one ends up with a grid of possible (and plausible) ghost states. The same happens if *e.g.* a second-order Markov model of position is used (which corresponds to position and velocity) and correspondence between the consecutive states is lost. However, depending on the absolute placement of the states, the ghost states quickly diverge into regions outside an area which can be assumed to be plausible. Furthermore, it is very unlikely that there will be consistent measurements far away from the correct locations, *i.e.*, the wrong hypotheses will never get support from measurements. In expectation value sense, all wrong hypotheses will vanish, which is a direct consequence of the proof in [7]. Hence, if joint densities are no option due to computational complexity, the higher-order Markov model is more suitable for multi-object tracking than motion state spaces. Using higher-order Markov models has already been proposed in [1], however not in a proper product formulation as derived in Sect. 3.1.

### 3.3 Embedding $n$ D Higher-Order Models

Higher-order Markov models depend on more states than just the previous one. In order to make use of the Markov property,  $n$  consecutive states need to be embedded in a larger state vector. However, as shown in Sect. 3.1, we have to multiply all densities obtained from different dimensions and time-steps according to (14), *i.e.*, we may not propagate the new state vectors through a linear mapping. Instead, we obtain

$$\mathbf{w}_{k|k-1}^m = (\mathbf{w}_k^m)^{1-N/2} \prod_{n=1}^N \sqrt{\mathbf{F}_n^m \mathbf{w}_{k-1|k-1}^n} . \quad (16)$$

What remains is how to learn the models  $\mathbf{F}_n^m$ ,  $\mathbf{h}$  and the prior. Note that due to the separable product in (16), all linear models can be learned separately by the Baum-Welch algorithm [22] and we omit the indices  $n$  and  $m$  in what follows.

In its initial formulation, the Baum-Welch algorithm is supposed to work on discrete state spaces. It can thus be applied to grid-based methods, but it has to be modified according to Sect. 3.1 for being applicable to channel vectors. Hence, all products of densities occurring in the Baum-Welch algorithm are replaced with square-root products of channel vectors. The  $\alpha$ -factor from the algorithm is identical to the update equations (10) ( $\alpha_k = \mathbf{w}_{k|k}$ ), which are modified accordingly. The  $\beta$ -factor from the algorithm is computed by propagating backwards through the measurement model and the system model

$$\beta_k = (\mathbf{F}^T(\mathbf{h}(\mathbf{z}_{k+1}) \cdot \beta_{k+1}))^T \quad (17)$$

where  $\cdot$  denotes the element-wise product. Again, all products are replaced with square-root products in the case of channels.

The computation for the system model update is straightforward, given that the factors  $\alpha$  and  $\beta$  are known

$$\mathbf{F} \leftarrow \frac{1}{N} \sum_k \beta_{k+1} \cdot \mathbf{h}(\mathbf{z}_{k+1}) \cdot (\mathbf{F}\alpha_k) \quad (18)$$

for a suitable normalization  $N$ . The measurement model is implemented as a mapping to measurement channels, *i.e.* a matrix as well, and it is updated as

$$\mathbf{h}(\mathbf{z}) \leftarrow \frac{1}{N} \sum_k \mathbf{z}_k (\alpha_k \cdot \beta_k) \quad (19)$$

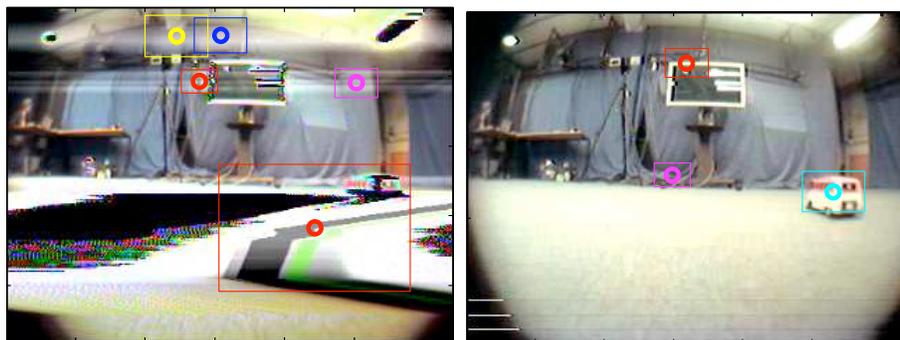
In the tracking *evaluation*, the  $\beta$ -factor has not been used, due to its anti-causal computation.

### 3.4 Correspondence-Free Learning

It has been shown that certain types of learning algorithms on channel representations do not require element-wise correspondence for learning and that learning becomes even faster if sets of samples are shown simultaneously [7]. This scenario is exactly the one that we meet when we try to learn associations of observations and objects: Detections might fail or might be irrelevant. Consider *e.g.* Fig. 1 for the type of learning problem that we face: We get detection results without correspondence, including drop-outs, outliers, and displacements. From these detections we train a system model and a measurement model. Using these models, we track individual cars through the sequence.

The correspondence-free learning has been shown in [7] by proving equivalence to a stochastic gradient descent on the learning problem with correspondence. The central question here is, whether the Baum-Welch algorithm will lead to a similar result, *i.e.*, whether the expectation of the algorithm will be the solution of the learning problem with correspondence. We will not give a formal proof here, but in the subsequent section, we will give empirical evidence that this is the case.

The Baum-Welch algorithm is initialized with the covariance-based estimates that were used in [1]. In our experiments, the algorithm converged after four iterations. The fact that the algorithm found a model capable of tracking an individual object empirically supports that correspondence-free learning also works in this scenario. Thus, CBT using Baum-Welch gives a solution for the *hard association problem* of detections and objects without additional optimization of discrete assignment problems.



**Fig. 1.** Two consecutive frames, 321 (left) and 322 (right), from the first RC car (rightmost box) sequence with detections (boxes).

## 4 Experiments

We evaluated the proposed method in comparison to PMHT [13] and to PDAF [14]. Both methods are extensions to the Kalman filter, which try to overcome the problem of object tracking in the presence of multiple, false and/or missing measurements. PMHT uses a sliding window that incorporates previous, current and future measurements and applies the expectation-maximization approach to get the new state estimates. PDAF makes probabilistic associations between target and measurements by *combined innovation*. That is, PDAF makes a weighted update based on all measurements where the weights are given by the probability for the respective measurement given the current prediction. The experimental setup is illustrated in Fig. 1: From (partly very low quality) image sequences, each consisting of several hundred frames showing the front view, we detect cars. These detections are indicated by the colored boxes.

Note that we do not use visual tracking of vehicles by *e.g.* least-squares, since the cars might change their appearance significantly from different views, due to shadows, highlights, change of aspect etc. For detection we use a real-time learned cascade classifier [25]<sup>2</sup>. The input to our tracking algorithm is an unsorted list of coordinates with corresponding likelihood-ratios for each frame. We trained the system in leave-one-out manner on the detections from all sequences except for the respective evaluation sequence. We were only interested in the other RC car in the sequences, *i.e.*, we were only interested in the primary occurring trajectory.

The parameters were chosen as follows. For the PMHT method, we obtained most stable results with a motion model including velocity but with constant size; size-change estimates did not improve the results. For the cost function of an association, we chose distance instead of probability, since the latter did not give reasonable results for large parts of the trajectories. The PMHT implementation is based on the implementation<sup>3</sup> of [15] and [16]. For the PDAF method, we also obtained the most stable results with the

<sup>2</sup> We would like to thank our project partners from Prague (J. Matas, T. Werner and A. Shekhovtsov) for providing the detections.

<sup>3</sup> <http://www.anc.ed.ac.uk/demos/tracker/>

model described above. We computed the weights of each measurement as the probabilities for the respective measurement given the prediction based on previous timesteps. For the CBT method, we chose 20 channels per state vector dimension and a model of order 3.

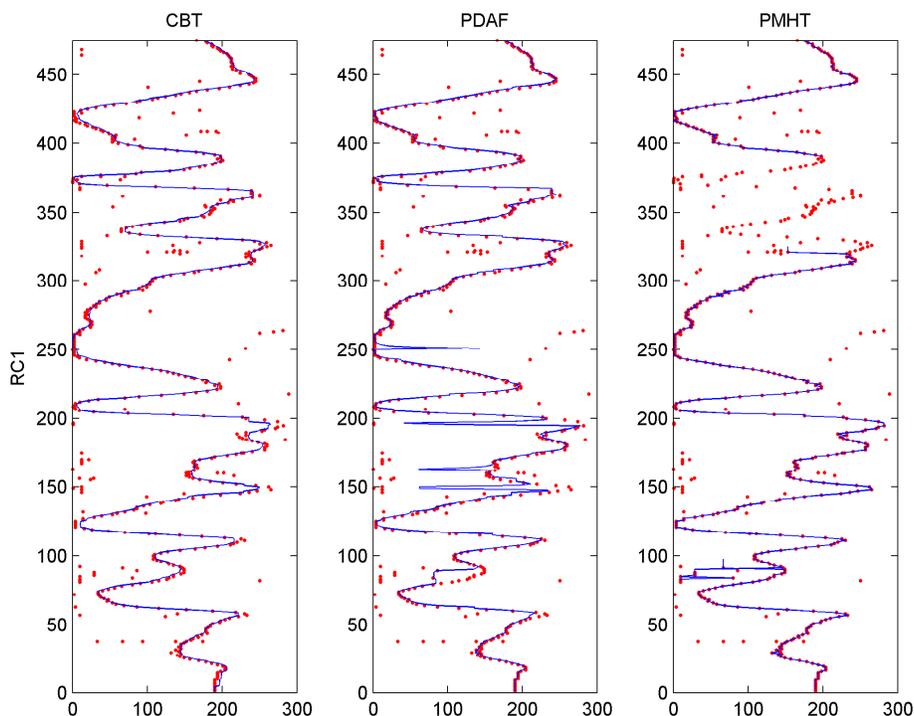
After some optimization, all three methods delivered reasonable trajectories in all test cases. However, for PMHT and PDAF the association of detections to the correct object sometimes failed, *e.g.*, for PMHT around frame 80 in RC1 and for PDAF around frame 150 in RC1. Figure 2 shows the obtained result for this sequence. The initial detections are indicated as red crosses, and the obtained results as blue curves. The accuracy of PMHT is very good in all segments. However, spurious trajectories had previously been removed by thresholding the length of the trajectories. PDAF shows slightly worse accuracy but on the other hand suffers less from completely losing track of the object. Notice that CBT does not lose track of the object at all and also shows the best average accuracy in most cases.

The root mean squared error (RMSE) for the three methods compared to manually labeled ground truth are shown in table 1. In order to make the comparison fair for PMHT, which is designed to track multiple objects instead of a single object in clutter, we always choose the reported object that is closest to the ground truth. This means that we ignore each time PMHT loses track of the object, as long as it reintroduces a new object at the correct position. We also have included the performance of the pure detector in the last column. For the detector we used the most likely measurement, given by the log-likelihood, when there were multiple reported detections.

All results are shown for three different outlier rejection strategies; no rejection, rejection for outliers larger than 20 and larger than 5. We can see that the performance of PMHT is slightly better than CBT when we almost disregard outliers, see RCX(5). CBT is performing best when we put some weight to outliers, RCX(20), and there is an even greater difference when no thresholding is done, *c.f.* RCX( $\infty$ ). Note that when no target is reported at all, *e.g.* PMHT frame 350 in RC1, that frame did not contribute to the RMSE for that method. The CBT kept track of the car in front in nearly all cases, until it got out of view. No association problems occurred. Again, the estimates are very accurate in comparison to the localization of the original detections and in comparison to the other two methods.

**Table 1.** The RMSE of each method compared to manually labeled ground truth. The number in the paranthesis denotes the maximum deviation that was used. If the actual deviation was larger it was replaced with this value.

	CBT	PMHT	PDAF	Detector	CBT	PMHT	PDAF	Detector	
RC1 ( $\infty$ )	<b>7.3</b>	13.1	18.4	40.43	RC2 ( $\infty$ )	<b>6.7</b>	15.6	23.3	42.72
RC1 (20)	<b>6.6</b>	8.9	9.1	10.54	RC2 (20)	<b>6.5</b>	7.1	8.5	10.58
RC1 (5)	3.9	<b>3.2</b>	4.0	4.17	RC2 (5)	3.9	<b>3.5</b>	3.9	4.23



**Fig. 2.** Result on RC car sequences. Left/center/right column shows the result of CBT/PDAF/PMHT on RC sequence 1. The red crosses indicate the detections and the blue curve is the result obtained by each method. We have only plotted the result for the x-axis.

## 5 Conclusion

We have extended the framework of CBT in several ways. The multi-dimensional case has been re-formulated in a sound probabilistic approach. The previously suggested higher-order Markov model has been embedded into a first-order model, allowing to apply the Baum-Welch algorithm for learning the system model and the measurement model for the tracking. The learning algorithm itself has been extended and it has been shown to work on weakly labeled data. The association problem of observations and objects is solved without additional discrete optimization steps.

The resulting tracking algorithm has been shown to extracting individual objects from noisy detections of multiple objects and compares favorably with existing techniques. We have discussed the advantages of using marginals of higher-order Markov models compared to motion states. As a result of working on marginals, the algorithm runs in full real-time. The proposed method shows best accuracy and robustness in most of the evaluated sequences.

Potential application areas are visual surveillance from cheap and/or uncalibrated cameras and image sequence analysis of objects with unknown system models.

## References

1. Felsberg, M., Larsson, F.: Learning Bayesian tracking for motion estimation. In: International Workshop on Machine Learning for Vision-based Motion Analysis. (2008)
2. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Sig. P.* **50** (2002) 174–188
3. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision* **29** (1998) 5–28
4. Coué, C., Fraichard, T., Bessière, P., Mazer, E.: Using Bayesian programming for multi-sensor multitarget tracking in automotive applications. In: ICRA. (2003)
5. Granlund, G.H.: An Associative Perception-Action Structure Using a Localized Space Variant Information Representation. In: Proceedings of the AFPAC Workshop. (2000)
6. Johansson, B., et al.: The application of an oblique-projected landweber method to a model of supervised learning. *Mathematical and Computer Modelling* **43** (2006) 892–909
7. Jonsson, E., Felsberg, M.: Correspondence-free associative learning. In: ICPR. (2006)
8. Felsberg, M., Forssén, P.E., Schar, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. *PAMI* **28** (2006) 209–222
9. Georgiev, A.A.: Nonparametric system identification by kernel methods. *IEEE Trans. on Automatic Control* **29** (1984)
10. Han, B., Joo, S.W., Davis, L.S.: Probabilistic fusion tracking using mixture kernel-based Bayesian filtering. In: IEEE Int. Conf. on Computer Vision. (2007)
11. North, B., Blake, A.: Learning dynamical models using expectation-maximisation. In: ICCV. (98)
12. Ardö, H., Åström, K., Berthilsson, R.: Real time viterbi optimization of hidden markov models for multi target tracking. In: Proceedings of the WMVC. (2007)
13. Streit, R.L., Luginbuhl, T.E.: Probabilistic multi-hypothesis tracking. Technical report, 10, NUWC-NPT (1995)
14. Shalom, B.Y., Tse, E.: Tracking in a cluttered environment with probabilistic data association. *Automatica* **11** (1975) 451–460
15. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis and Machine Intell.* **22** (2000) 747–757
16. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* **38** (1987) 325–340
17. Snippe, H.P., Koenderink, J.J.: Discrimination thresholds for channel-coded systems. *Biological Cybernetics* **66** (1992) 543–551
18. Pampalk, E., Rauber, A., Merkl, D.: Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In: ICANN’02, Madrid, Spain, Springer (2002) 871–876
19. Forssén, P.E.: Low and Medium Level Vision using Channel Representations. PhD thesis, Linköping University, Sweden (2004)
20. Felsberg, M.: Spatio-featural scale-space. In: International Conference on Scale Space Methods and Variational Methods in Computer Vision. Volume 5567 of LNCS. (2009)
21. Yakowitz, S.J.: Nonparametric density estimation, prediction, and regression for markov sequences. *Journal of the American Statistical Association* **80** (1985)
22. Baum, L.E., et al.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.* **41** (1970) 164–171
23. Rao, R.P.N.: An optimal estimation approach to visual perception and learning. *Vision Research* **39** (1999) 1963–1989
24. Therrien, C.W.: Decision, estimation, and classification: an introduction into pattern recognition and related topics. John Wiley & Sons, Inc. (1989)
25. Sochman, J., Matas, J.: Waldboost - learning for time constrained sequential detection. In: Proc. Conf. Computer Vision and Pattern Recognition. Volume 2. (2005) 150–157