A Variational Approach to Simultaneous Multi-Object Tracking and Classification

Victor Romero-Cano[1] *, Gabriel Agamennoni[2] and Juan Nieto[1]

[1]*Australian Centre for Field Robotics, The University of Sydney, Sydney, Australia*

[2]*Autonomous Systems Lab, ETH Zürich, Zürich, Switzerland*

Abstract

Object tracking and classification serve as basic components for the different perception tasks of autonomous robots. They provide the robot with the capability of class-aware tracking and richer features for decision-making processes. The joint estimation of class assignments, dynamic states and data associations results in a computationally intractable problem. Therefore, the vast majority of the literature tackles tracking and classification independently. The work presented here proposes a probabilistic model and an inference procedure that render the problem tractable through a structured variational approximation. The framework presented is very generic, and can be used for various tracking applications. It can handle objects with different dynamics, such as cars and pedestrians and it can seamlessly integrate multi-modal features, for example object dynamics and appearance. The method is evaluated and compared to state-of-the-art techniques using the publicly available KITTI dataset.

Keywords

Tracking, Data Association, Classification, Variational Inference, Robotic Perception

1. Introduction

Autonomous robots need to perceive the environment and provide a semantic description of the different moving objects they encounter. This task is usually referred as *dynamic scene understanding* (Buxton 2003). In most scenarios, objects move according to constraints imposed by: 1) the environment (highway, inner city), 2) the type of interactions they are exposed to (cruise, turning) and 3) their nature or class category (car, pedestrian). Reasoning about object classes enables the robot to use context information more selectively for tracking. On one hand, knowledge of object classes can improve tracking. On the other hand, good trajectories serve as features for better classification.

Multi-Object Tracking (MOT) is the procedure that, at a low level, provides information about what objects of interest there are in the environment and their behavioural characteristics (Kaempchen et al. 2009). It allows a perception system to take the input of sensors such as stereo (Vatavu et al. 2014) and RGB-D cameras (Ren et al. 2014), and/or laser (Held et al.

* Corresponding author: e-mail: varomero@acfr.usyd.edu.au

2014), and estimate the trajectories of objects in the environment. The problem of estimating these trajectories is usually approached in terms of object hypotheses that are updated as new observations are available, which introduces one of the most challenging problems in the tracking literature: how do we associate observations to object hypotheses?, also known as the Data Association (DA) problem.

Simultaneous tracking and classification of multiple moving objects with unknown DA is a computationally intractable problem. This is why techniques that classify objects according to their dynamics perform each of the tasks separately, decoupling state estimation from class assignment (Bashir et al. 2007, García-García et al. 2011, Vasquez et al. 2009). Hence, they neglect the natural correlations between object's dynamics, environment and class category. Additionally, most state-of-the-art approaches to multi object tracking either do not assign category labels to tracked objects or obtain them from an independent process usually based on images (Ess et al. 2010).

The joint estimation of object classes, states and data association has several advantages over previous approaches to both object classification and MOT. First of all, it can boost state-of-the-art appearance-based object classification methods (Gavrila and Munder 2006, Wojek et al. 2012), by exploiting motion information and temporal correlations in the data. Secondly, since our approach formulates the problem in terms of a fully probabilistic model, it enables parameter learning. This is in contrast with most MOT approaches, in which the user is expected to empirically set the parameters of the tracker.

The practical advantages of the work presented here are numerous. The system outputs state estimates for all of the objects in the scene and soft assignments of each object to different motion categories or classes. These classes represent motion patterns that we learn in the form of a Mixture of Linear Dynamical Systems (MLDS) (Chan and Vasconcelos 2008). The approach also enables object states/classes to be estimated even from noisy, incomplete and ambiguous measurements. Furthermore, our method utilises classic and efficient statistical estimation techniques such as the Kalman filtering and smoothing recursions (Rauch et al. 1965) as subroutines. The approach is validated with publicly available datasets.

The specific contributions of this paper are:

- a holistic probabilistic graphical model that encapsulates the correlation between object classes and object states while also modelling data association;
- the Expectation-Association (EA) algorithm: a new method for performing data association, state estimation and classification in a joint probabilistic fashion. It addresses both the offline and online cases;
- a method for data association that uses appearance features with dynamic information in a unified probabilistic framework;
- an extensive validation using publicly available data collected in urban environments, and comparisons with state-of-the-art methods.

The underlying model used in this paper was initially presented in (Romero-Cano et al. 2014). In this manuscript we extend our previous work by incorporating to the framework a more effective strategy for initialising association probabilities. This paper also presents a more comprehensive experimental evaluation with 20 stereo-vision sequences from the publicly available KITTI dataset. The paper is organised as follows. Section 2 presents a review of related work. Section 3 introduces our probabilistic model for describing the multi-object tracking and classification problem. In Section 4 we present our variational approximation. Finally, experimental results are presented in Section 5 followed by conclusions and future research directions in Section 6.

2. Related Work

Multi-object tracking is a well-known problem in the robotics community and many publications on the matter have been produced (Choi et al. 2013, Frank et al. 2003, Gu and Veloso 2009, Moosmann and Stiller 2013, Wang et al. 2007). Most of the current approaches to tracking follow a *tracking-by-detection* methodology, where first, objects of interest are detected at each frame (Bajracharya et al. 2009, Romero-Cano and Nieto 2013), second, detections are linked to object hypotheses

across frames, and third, the trajectories of object hypothesis are estimated. Depending on whether detections are assumed to be perfect measurements of the objects state or not, two approaches to tracking exist. The first one defines trajectories in terms of subsets of detections that follow some smoothness constraints (Dicle et al. 2013, Huang et al. 2013). The second one considers objects states as hidden variables that need to be estimated from incomplete and noise observations (Brau et al. 2013, Geiger, Lauer, Wojek, Stiller and Urtasun 2013, Milan et al. 2013, Schumitsch et al. 2006, Segal and Reid 2013). In general, multi-object tracking algorithms estimate the objects states without reasoning about classes. When this reasoning is required, complete tracks obtained from an independent tracking system are typically classified using either similarity-based clustering techniques (Katz et al. 2010) or Hidden Markov Models (Bashir et al. 2007, García-García et al. 2011).

Only few approaches exist that simultaneously perform tracking and classification (Agamennoni et al. 2012, Li 2007, Oh et al. 2007, Rong Li 2007). Rong Li (2007) presented a joint optimisation method that uses information from different sensor modalities in order to jointly estimate objects' states and classes. It allows the user to define the degree of correlation between tracking and classification by means of cost weights for errors in both tasks. However, it assumes known data association, which limits its application to environments where objects appearances are very dissimilar or they have identity markers. Li (2007) introduced the theoretical framework on which (Rong Li 2007) is based. It argues the framework can be applied to solve the data association problem when there is no interest on reasoning about classes. However no results are provided and, it is also not clear how the method can be extended to jointly deal with state estimation, data association and classification.

In (Oh et al. 2007), an approach that performs sampling-based inference on Segmental Switching Linear Dynamic System (S-SLDS) models was presented. Sampling methods can be computationally demanding and thus prohibitively slow (Bishop 2006). They can be slow to converge (Hensman et al. 2012) or fail to converge when not enough data is available in relation to the complexity of the model (Casey et al. 2008). Variational approximations, on the other hand, are guaranteed to converge.

From a theoretical perspective, the work presented in this paper is similar to that by Kanazaki et al. (2007), where a variational method for multi-target tracking with unknown data association is presented. A key difference, however, is that Kanazaki et al. (2007) considers only a single model, common to all of the targets, while our approach utilises a bank of models learnt from data. In other words, our framework produces soft assignments of objects to a set of predefined motion models. Additionally, the results presented by Kanazaki et al. (2007) were obtained from synthetic data only. In contrast, we extensively evaluate our method using stereo images from an urban scenario.

Other approaches to multi-object tracking and classification are based on Random Finite Set (RFS) statistics (Reuter et al. 2013). Under this framework, both the object states and number of objects are modelled as random sets. In (Mahler 2003), the Probability Hypothesis Density (PHD) filter was proposed as an approximation of the multi-object Bayes filter using RFS statistics. Pasha et al. (2009) presented a Gaussian mixture implementation of the PHD filter that allows objects to switch between multiple motion models. Recently, Meissner et al. (2014) extended the work in (Pasha et al. 2009) so that tracked objects can also be classified using features of both, the measurements and the tracked objects. The PHD filter has also been applied to extended targets (Granström et al. 2014), i.e. objects that can emit multiple observations per time step and therefore they are better described by augmented states that consider not only position but also shape descriptors. The Extended Target GM-PHD (ET-GM-PHD) filter (Granström et al. 2012) is a Gaussian Mixture implementation of the PHD filter for extended objects presented by Mahler (2009). The work in (Mahler 2009) requires each possible grouping/partition of the observations be consider in order to update the objects state; which in practice, is computationally prohibitive. Granström et al. (2012) approach the issue by using only the most probable partitions. The main drawback of such an approach is that no explicit reasoning about object identities is performed. Therefore, further post-processing is needed in order to get the individual state trajectories.

3. Simultaneous Multi-object Tracking and Classification

In this paper we represent the state trajectory of each object in the scene as a realisation of a Linear Dynamical System (LDS), which is a linear-Gaussian model, widely used for representing time series. We use a mixture of LDSs for representing multiple object categories and extend the model so that data association can be performed. In this section we walk the reader through the components in our model.

3.1. Linear Dynamic Systems

An LDS is a generative model. It represents a sequence of observations as being generated by an underlying hidden Markov process. The hidden process $x^i_{1:T}$ represents the state trajectory of an object. The observations $z^i_{1:T}$ are a linear projection of the states, plus noise. The conditional probability distributions in an LDS are:

$$\begin{aligned} p\left(x^i_t|x^i_{t-1}\right) =& \mathcal{N}\left(x^i_t; Fx^i_{t-1}, Q\right) \\ p\left(z^i_t|x^i_t\right) =& \mathcal{N}\left(z^i_t; Hx^i_t, R\right) \end{aligned} \tag{1}$$

where F is the state transition matrix; Q is the process noise covariance matrix; H is the observation matrix or linear mapping between hidden states and observations; and R is the covariance matrix that describes the noise in the sensor.

The first and second line in Eq. (1) are commonly referred to as the transition and observation models respectively. Under this representation, the state sequence of an object in the scene corresponds to a *Directed Acyclic Graph* (DAG). Inference on this sort of model can be done efficiently and exactly using the sum-product algorithm, which for the LDS model in particular, boils down to a set of forward and backward recursions known as the Kalman filter and smoother with no driving inputs Kschischang et al. (2001).

In the context of our application, observations z_t correspond to noisy measurements of object positions, whereas hidden states are the actual positions, velocities and accelerations. Across this paper we will refer to F, Q, H and R as the model parameters. These parameters encode prior information about the objects dynamics and how the sensor perceives them.

3.2. The Mixture of Linear Dynamic Systems (MLDS)

In most environments, the states of moving objects evolve according to underlying class-dependent dynamics. For instance, in urban scenarios, cars, cyclists and pedestrians share the environment. In our model, co-occurring behaviours are accounted for by augmenting the LDS with a discrete hidden variable s_i. This categorical random variable has a number of values N_s equal to the number of expected classes. With this addition, the transition and observation models for each of the classes become:

$$p\left(x^i_t \mid x^i_{t-1}, s_i = j\right) = \mathcal{N}(F_j x^i_{t-1}, Q_j) \tag{2a}$$

$$p(\mathbf{z^i_t} \mid x^i_t, s_i = j) = \mathcal{N}(H_j x^j_t, R_j) \tag{2b}$$

where $s_i = j$ is the object category that generates trajectory i; and F_j, Q_j, H_j and R_j are the parameters of the jth LDS.

3.3. Data Association

So far, we have introduced a model that allows us to represent both the dynamic (x^i_t) and categorical (s_i) states of object i, given the observation z^i_t. In practice, our sensor provides a set of measurements z_t with no assignment to tracked objects. The problem of assigning observations to objects is known in the literature as *data association*.

Table 1: Terminology used in our model

Index	Symbol	Range
Object	i	$1, \ldots, N_x$
Model	j	$1, \ldots, N_s$
Time step	t	$1, \ldots, T$
Observation	l	$1, \ldots, L_t$
Variable	**Symbol**	**Support**
Class of target i	s^i	$\{1, \ldots, N_s\}$
State of target i at time step t	x_t^i	$\mathbb{R}^m$
Observation l at time step t	z_t^l	$\mathbb{R}^n$
Association of observation l at time step t	a_t^l	$\{1, \ldots, N_x\}$

Consider a sequence of observations $z = (z_1, \ldots z_t, \ldots, z_T)$ with $z_t = (z_t^1 ... z_t^l ... z_t^{L_t})$. These observations are assumed to be generated by N_x different objects. In order to represent the mapping between objects and observations, we define a sequence $a = (a_1, \ldots, a_t, \ldots, a_T)$ of assignment variables, with $a_t = \{a_t^1, \ldots, a_t^l, \ldots, a_t^{L_t}\}$. $a_t^l \in \{1, \ldots, N_x\}$ is a categorical variable that specifies which object is responsible for generating observation z_t^l. If tracked objects are well separated or they have identity makers, the associations are easily obtained and the posterior over objects states can be efficiently calculated. Otherwise, the method will have to account for the ambiguities in the data association process. For this case, estimating the object state x_i requires to calculate the following marginal:

$$p\left(x_i|z\right) = \sum_a p\left(x_i|a, z\right) p\left(a\right). \tag{3}$$

Calculating this marginal requires computations that grow combinatorially with the number of objects and exponentially with time. In order to overcome this computational intractability, approximations are usually made. Section 4 describes our approximation.

3.4. Model Overview

In this section we present our extension of MLDSs in order to deal with unknown data association. Fig. 1 shows the Bayesian network representation of our generative model. The joint probability distribution can be written as:

$$\begin{aligned} p\left(s, x, z, a\right) = & \prod_{i=1}^{N_x} \left[p\left(s_i\right) p\left(x_0^i | s_i\right) \prod_{t=1}^{T_i} p\left(x_t^i | x_{t-1}^i, s_i\right) \right] \\ & \prod_{t=1}^{T} \prod_{l=1}^{L_t} p\left(z_t^l \middle| x_t^{1:N_x}, a_t^l\right) p\left(a_t^l\right). \end{aligned} \tag{4}$$

In a Bayesian network, random variables and their conditional dependencies are represented by means of a DAG. In our graphical model, Θ represents a set of motion models. Each node s_i is a categorical random variable used for indexing 1 of N_s models. x_t^i is a continuous random variable that models the state of object i at time t. z_t^l is the lth observation made a time t. a_t^l is a categorical variable modelling the association between observation l and tracked objects. Finally, N_x is the number of objects in the scene, whereas L_t is the number of observations made at time t. Table 1 summarises the terminology used in the model.

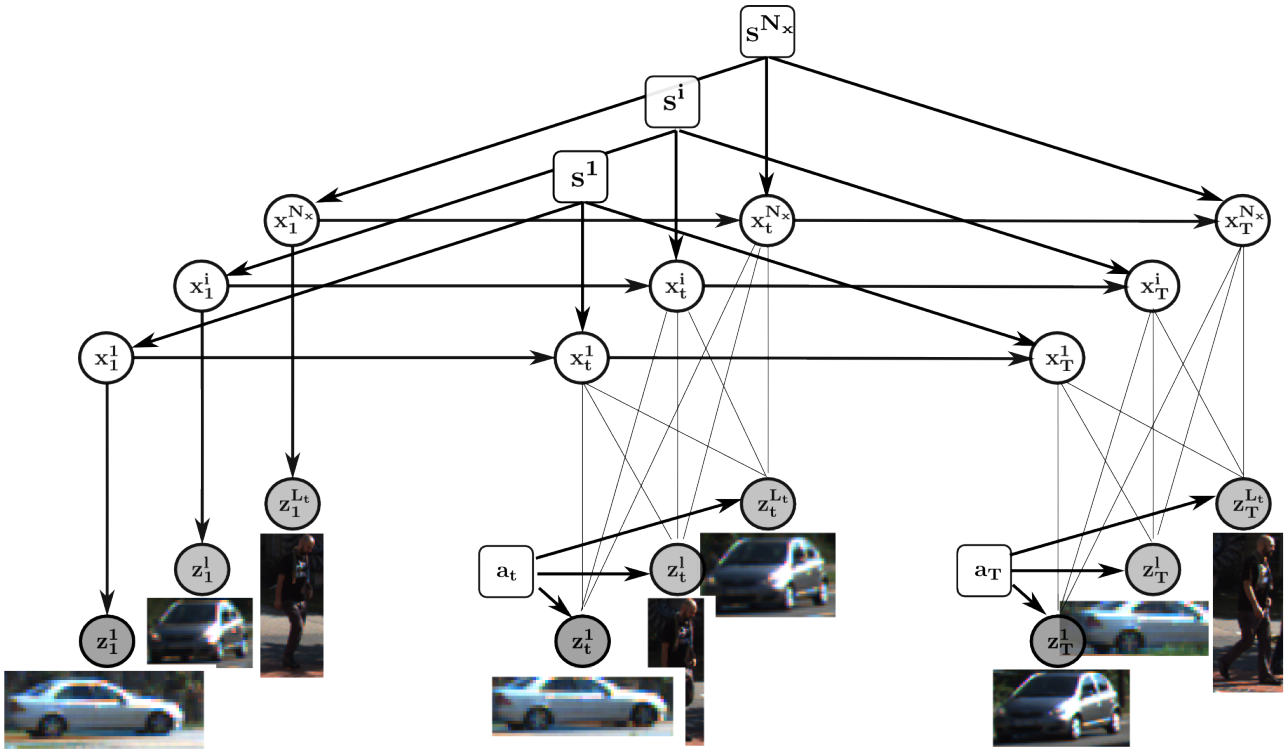


Fig. 1. Our graphical model. Squared and circular nodes represent categorical and continuous random variables respectively. Unfilled nodes indicate hidden variables, while filled nodes are observed.

4. The Expectation Association Algorithm

Motivated by the efficiency and convergence properties of variational inference methods, we introduce a new deterministic approximation scheme for solving the intractability issue of data association in multi-object tracking and classification. One of the most popular variational methods is the Variational Bayesian Expectation Maximisation (VBEM), proposed by Beal and Ghahramani (2003). The EA algorithm may be regarded as a special case of the VBEM algorithm. Both EA and VBEM are mathematically similar, since they seek to approximate the joint posterior distribution by eliminating the coupling between the variables that make this calculation intractable. They both estimate the posterior distribution over variables of interest in an iterative process which optimises a lower bound of the data likelihood. However they are different in that:

- VBEM decouples state variables from model parameters, whereas EA decouples state and class variables from association variables;
- VBEM is completely general, whereas EA is specialized and performs inference efficiently by exploiting the structure of the problem, and by taking advantage of the well-known Rauch-Tung-Striebel smoothing recursions.

This section starts by motivating our variational approximation. Then we present a structured approximation to the posterior over object classes and states, and data association. The section finalises with a pseudo-code description of the algorithm.

4.1. The Lower Bound

We aim to estimate the posterior $p(s, x, a|z)$ over classes, objects states and associations by maximising a likelihood function. The log-likelihood of the data is obtained by marginalising out the set of hidden variables (s, x, a) given a model

(Θ) and the observations (z):

$$\ln p(z) = \ln \sum_{s,a} \int p\left(s, x, a, z\right) dx. \tag{5}$$

Unfortunately, this integral is both analytically and computationally intractable due to the coupling between variables. By applying the *d-separation criterion* (Bishop 2006, pg. 378) on our model in Fig. 1, it can be seen that, although objects states/classes and associations are marginally independent, conditioning on the observations introduces statistical dependencies between them. As a result of these dependencies, the posterior is a mixture distribution where the number of components increases combinatorially with the number of objects and exponentially with time.

Given that the exact likelihood function is intractable, a lower bound is obtained. Let $q(s, x, a)$ be a probability density function that approximates the exact posterior $p(s, x, a|z)$. By expressing $\ln p(z)$ as

$$\ln \sum_{s,a} \int q\left(s, x, a\right) \frac{p\left(s, x, a, z\right)}{q\left(s, x, a\right)} dx \tag{6}$$

and applying Jensen's inequality (Bishop 2006), we arrive at a lower bound

$$\ln p(z) \geq \sum_{s,a} \int q\left(s, x, a\right) \ln \frac{p\left(s, x, a, z\right)}{q\left(s, x, a\right)} dx = \mathcal{L}\left[q\right]. \tag{7}$$

4.2. The Factorised Approximation

The inequality in Eq. (7) holds for any choice of q. In particular, if $q(s, x, a)$ equals the true posterior $p(s, x, a|z)$, then Eq. (7) becomes an equality. We propose approximating our posterior with a probability density function q that separates classes and states from data associations, so it factorises as follows:

$$q\left(s, x, a\right) = q(s, x)q(a). \tag{8}$$

Given this factorisation, the posterior of interest is approximated as the product of a state/class distribution and an association distribution. The approximate state/class distribution is a mixture distribution whose complexity increases linearly with time and the number of objects, and not exponentially as in the optimal case. Similarly, the calculation of the distribution over associations also become linear in time and in the number of tracked objects.

Our approximation assumes that given the data, state sequences and associations are statistically independent. [1] This does not imply that the state estimates and data associations are decoupled; in fact, they depend on one another via algebraic equalities —see Eqs. (10) to (16). The variational approximation transforms these *statistical* dependencies into *algebraic* constraints. We resolve these constraints by optimizing each of the q factors in turn, i.e. by fixing the state estimates and updating the associations, and vice-versa. As shown in Fig. 2, in the context of our tracking application, object trajectories tend to be temporally coherent, hence, given the data observed up to the current time step, it is possible to recursively estimate states/classes and data association. Fig. 3 shows a synthetic example that illustrates the validity of our assumption.

Unlike the exact posterior, our approximate solution in Eq. (8) is computationally tractable. We can derive the expressions for the factors in Eq. (8) by maximising Eq. (7). As explained by (Bishop 2006, pg. 466), the log of the optimal solution for factor q_j is obtained by considering the log of the joint distribution over all variables and then taking the expectation with respect to all of the other factors q_i for $i \neq j$.

[1] In other words, after having observed the data, any remaining statistical dependencies (e.g. cross-covariances) between state and association variables are not captured by our approximation.

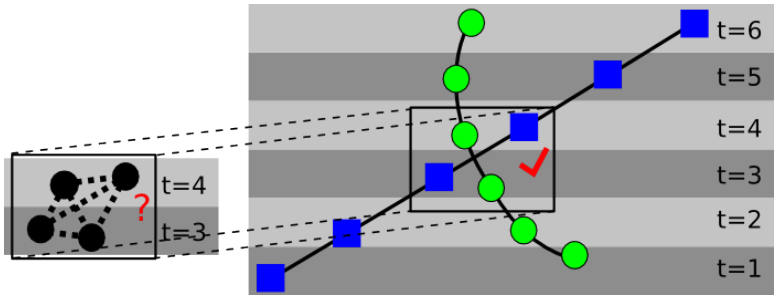


Fig. 2. We assume that once object trajectories are observed, temporal coherence in the observations make the correlation between classes/states and associations negligible, hence we can assume they are statistically independent. On the left hand side of the image, it would be difficult to choose which observations from time $t = 3$ go with which observations at time $t = 4$. In contrast, as shown at the right hand side, by looking at the observations from time $t = 1$ to $t = 6$ this ambiguity is minimised.

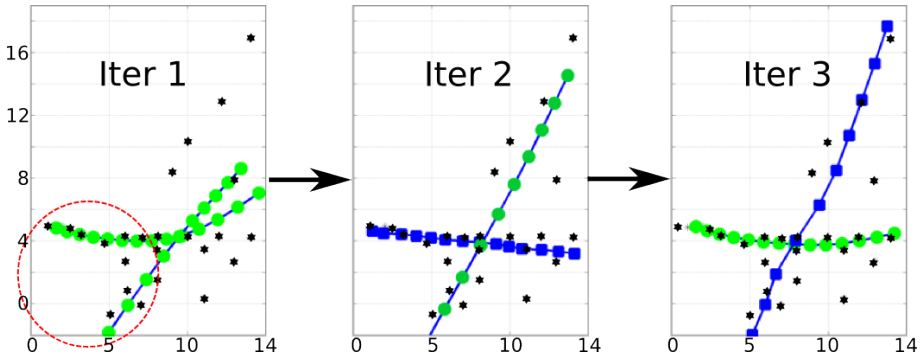


Fig. 3. A Synthetic example of state/class estimation from noisy observations using the EA algorithm. Observations, which are represented by stars, were generated using two different motion models. Filled circles and squares represent the two different object classes, with the true class assignment shown at iteration 3. Those squares/circles that are linked represent one individual object. This example illustrates how the motion history of the objects serves to disambiguate the data association and classify them according to their dynamic behaviour. In iteration 1, two objects are initialised and observations inside the dashed circle are associated to these objects with a high probability. The associations for detections outside the circle are assigned a non-informative uniform distribution. After running the E-step with $q(a)$ initialised as explained before, the resulting trajectories show confident assignments inside the circle and ambiguous ones elsewhere. In iteration 2, we run the E-step using the $q(a)$ updated in the previous iteration. Although the class assignment has not converged to the correct one, object trajectories fit better the structure of the observations. Convergence is reached in the $3rd$ iteration.

Factors $q(s, x)$ and $q(a)$ are updated as explained in Subsections 4.3 and 4.4 respectively, and cycling through them until convergence. We call this iterative process the Expectation-Association (EA) algorithm and introduce its batch version with pseudo-code in Algorithm 1. It will be shown that the initial factorisation in Eq. (8) results in other factorisations across time, and within objects and observations (see Eq. 9 and Eq. 15). These are *induced* factorisations, i.e., they do not concede additional accuracy and are exact given the initial assumption in Eq. (8). Eqs. (10)-(16) make clear the intuition behind our approximation: once a trajectory i has been observed, its states and class assignments can be estimated recursively from a sequence of surrogate observations and noise covariances, which arise from expectations of the data association variables.

4.3. The Expectation Step (E-step)

The first factor of our variational approximation $q(s, x)$, given the initial approximation, factorises as:

$$q(s, x) = \prod_{i=1}^{Nx} q(x_i|s_i)q(s_i), \tag{9}$$

and is obtained by maximising the lower bound (Eq. (7)) with respect to each of the induced sub-factors $q(s_i, x_i)$.

Algorithm 1 The batch EA algorithm

1: $Models \leftarrow$ Fit MLDS to training trajectories indicative of motion patterns in the environment.
2: $q(a) \leftarrow$ Initialise observation-to-object association probabilities using appearance.
3: **procedure E-step**($Models$, $q(a)$)
4: $\overline{z}_t^i \leftarrow$ Calculate per-object average observation.
5: $\overline{R}_t^{i,j^{-1}} \leftarrow$ Calculate per-object/per-model observation noise covariance.
6: $\sum_{t=2}^{T_i} l_t^{i,j} \leftarrow$ Run per-object/per-model Kalman Filter and obtain innovation log-likelihoods.
7: $q(x|s) \leftarrow$ Run RTS smoother and obtain per-model posterior over object states.
8: $q(s) \leftarrow$ Calculate marginal over class assignments.
9: **end procedure**
10: **procedure A-step**($Models$,$q(s,x)$)
11: $q(a) \leftarrow$ Update the association probabilities.
12: **end procedure**
13: Repeat until convergence.

$$\begin{aligned}\ln q\left(x_{1:T_i}^i | s^i = j\right) = -\frac{1}{2}\Bigg(\sum_{t=1}^{T_i} \left(x_t^i - F^j x_{t-1}^i\right)^T Q^{j^{-1}} \left(x_t^i - F^j x_{t-1}^i\right) \\ + \left(\overline{z}_t^i - H^j x_t^i\right)^T \alpha_t^i R^{j^{-1}} \left(\overline{z}_t^i - H^j x_t^i\right) \Bigg) + \dots\end{aligned} \tag{10}$$

$$\ln q(s^{i,j}) = \ln p(s^{i,j}) + \sum_{t=2}^{T_i} l_t^{i,j} + \dots \tag{11}$$

where

$$\begin{aligned}\alpha_t^i &= \sum_{l=1}^{L_t} \alpha_t^{l,i} \\ \alpha_t^{l,i} &= q(a_t^{l,i}).\end{aligned} \tag{12}$$

From Eq. (10) and Eq. (11) one can see that the optimal $q(s,x)$ is a Gaussian mixture distribution, with one component for each motion pattern. In these equations, "…" represent additive constants. Since the distribution in Eq. (10) has a quadratic form, it can be efficiently calculated using the Kalman filter (KF) and the Rauch-Tung-Striebel (RTS) smoother (Rauch et al. 1965). In this expression, the term $\overline{z}_t^i$, given by

$$\overline{z}_t^i = \frac{\sum_{l=1}^{L_t} \alpha_t^{l,i} z_t^l}{\alpha_t^i}, \tag{13}$$

is a weighted average of the observations with weights proportional to the posterior association probabilities of all the observations and target i. Additionally, Eq. (10) can be seen as an LDS parametrised by $\hat{x}_0^j$, $cov\left(x_0^j\right)$, F^j, Q^j, H^j and the average information matrix

$$\overline{R}_t^{i,j^{-1}} = \alpha_t^i R^{j^{-1}}. \tag{14}$$

Note that the marginal over the class assignment variables in Eq. (11) is obtained by updating the prior over class assignments with the marginal log-likelihood of the data under the model j. This log-likelihood can be obtained as a by-product of the E-step. It is equal to the sum of the innovation log-likelihoods $l_t^{i,j}$, which are computed at each update step. Accumulating these innovation log-likelihoods, after performing filtering with each of the models, allows us to infer

the assignment of targets to motion patterns. Furthermore, since Kalman filtering provides these innovation log-likelihoods each time an observation is processed, evidence about class assignments can be sequentially updated. This is fundamental for applying our framework to online tracking.

4.4. The Association Step (A-step)

The second factor of the factorised approximation is $q(a)$. Its natural logarithm is given by:

$$\ln q(a) = \sum_{t=1}^{T} \sum_{l=1}^{N_z} \sum_{i=1}^{N_x} \ln q\left(a_t^{l,i}\right), \tag{15}$$

where $a_t^{l,i}$ is a categorical random variable over the associations of detection l to object i at time t. We obtain each of the sub-factors in Eq. (15) by maximising Eq. (7) with respect to $q(a_t^{l,i})$.

$$\begin{aligned} \ln q\left(a_t^{l,i}\right) = \ln p\left(a_t^{l,i}\right) - \frac{1}{2}\Bigg(\sum_{j=1}^{N_s} q\left(s^{i,j}\right)\Bigg(\left(z_t^l - H^j \hat{x}_t^{i,j}\right)^T R^{j^{-1}} \left(z_t^l - H^j \hat{x}_t^{i,j}\right) \\ + Tr\left(H^{j^T} R^{j^{-1}} H^j Cov\left(\hat{x}_t^{i,j}\right)\right)\Bigg)\Bigg) + \dots \end{aligned} \tag{16}$$

In these factors, given by Eq. (16), $\hat{x}_t^{i,j}$ is the smoothed state of the targets. Note that $q(a)$ depends on the square of the error between expected and actual observations. Moreover the log-likelihood of assigning object i to observation l at time t decreases when the uncertainty about the state of object i (state covariance) increases. This permits our approach to be robust against spurious observations, even without explicit states to model them as in classical approaches such as JPDAF, MHT or MCMCDA. Since spurious detections typically support a very small portion of the object's trajectory, they tend to have very weak estimated associations, even if these associations were initialised with a high probability. Object trajectories that were initialised due to spurious observations tend to remain short, as they are promptly removed due to their lack of evidential support.

Integrating Appearance and Dynamics A key feature of our formulation is that it allows us to integrate appearance and dynamics when calculating the association between observations and objects. Typically, there are several sources of information to estimate the association between objects and observations. The prior over associations $p(a_t^l)$ (see Eq.(16)) can be calculated, for example, based on appearance features; the inference algorithm then computes the posterior $q(a_t^l)$ by seamless fusing this prior with evidence from the object's dynamics. In this paper a histogram-based appearance model per tracked object is sequentially updated with the image patches obtained from the detections when no association ambiguities (different objects getting together or occlusions) are detected. Then, we calculate the prior over associations by comparing the appearance models against image patches obtained once the ambiguity has finished. This allows our approach to recover objects identities even after object merging or occlusion interactions.

4.5. The Online EA Algorithm

The form of the factors in our approximation allows our method to be implemented in a sequential manner. As shown in Eq. (11), the assignment probabilities are obtained by accumulating the innovation likelihoods of the objects under each of the motion modes. Therefore, when applying our method online, we simply filter each track using each of the mixture components and accumulate their innovation likelihoods so that the class assignment probabilities can be recalculated at each time step. Similarly the association factors can be sequentially updated due to the fact that they are a function of the current object state. The tracking process is summarised in Algorithm 2.

Algorithm 2 The online EA algorithm

Model ← Fit MLDS to training trajectories indicative of motion patterns in the environment
w ← Sliding window provided by the user
EAits ← Number of EA iterations
for $t \leftarrow 1,\ T$ **do**
  $z_t^{1:N_z}$ ← Obtain raw observations
  $q_l(a_{t-w+1:t})$ ← Initialise association probabilities in the sliding window
  for $k \leftarrow 1,$ EAits **do**
    procedure E-step(Model, $q(a_{t-w+1:t})$,$z_t^{1:N_z}$)
      for $i \leftarrow 1,\ N_x$ **do**
        $\mathbf{z^i_{t-w+1:t}}$ ← Calculate weighted observations.
        $\mathbf{R^{i,s_i}_{t-w+1:t}}$ ← Calculate observation noise covariances
        $q_i(x_i|s_i)$ ← Perform filtering
        $\sum l_{1:t}^{i,s_i}$ ← Accumulate innovation log-likelihoods.
        $q_i(x_i|s_i)$ ← Perform smoothing
        $q_i(s_i)$ ← Calculate class assignment probabilities
      end for
    end procedure
    procedure A-step(Model,$q(x, s)$)
      $q(a_{t-w+1:t})$ ← Update the association factors using the estimated states.
    end procedure
  end for
end for

The Kalman forward-backward recursions provide our method with the capability of solving data association ambiguities without throwing away evidence in ambiguity areas. By forward propagating the filtering densities, followed by backward propagating the smoothed densities we allow the dynamics of the objects to refine the state estimates and, more importantly, obtain the association between observations and objects as a natural product of the objects' state histories. The approximate state/class distribution for object i in Eq. 10 has the form of an LDS, hence, as mentioned before, this posterior mode can be calculated using the RTS recursions. The resulting update equations for the forward pass are as follows:

$$\begin{aligned}\overline{x}_t^{i,j} &= F^j \overline{x}_{t-1}^{i,j} + K_t\left(\overline{z}_t^i - H^j F^j \overline{x}_{t-1}^{i,j}\right),\\ \overline{V}_t^{i,j} &= \left(I - K_t H^j\right) P_{t-1} \left(I - K_t H^j\right)^T + K_t \mathbf{R_t^{i,j}} K_t^T,\end{aligned} \tag{17}$$

where we have defined

$$\begin{aligned}P_{t-1} &= F^j \overline{V}_{t-1}^{i,j} {F^j}^T + Q^j,\\ K_t &= P_{t-1} {H^j}^T \left(H^j P_{t-1} {H^j}^T + \mathbf{R_t^{i,j}}\right)^{-1}.\end{aligned} \tag{18}$$

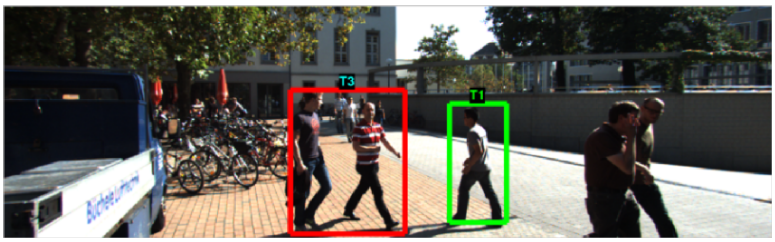
Once the filtering (forward pass) has been done, we calculate the smoothed posterior using the backward recursions:

$$\begin{aligned}\hat{x}_t^{i,j} &= \overline{x}_t^{i,j} + J_t\left(\hat{x}_{t+1}^{i,j} - F^j \overline{x}_t^{i,j}\right)\\ \hat{V}_t^{i,j} &= \overline{V}_t^{i,j} + J_t\left(\hat{V}_{t+1}^{i,j} - P_t\right) J_t^T\end{aligned} \tag{19}$$

where we have defined $J_t = \overline{V}_t^{i,j} {F^j}^T \left(P_t\right)^{-1}$. Using these equations we update the object states in the E-step, and calculate in the A-step, association probabilities that consider this enhanced motion history of the objects. Being able to bootstrap

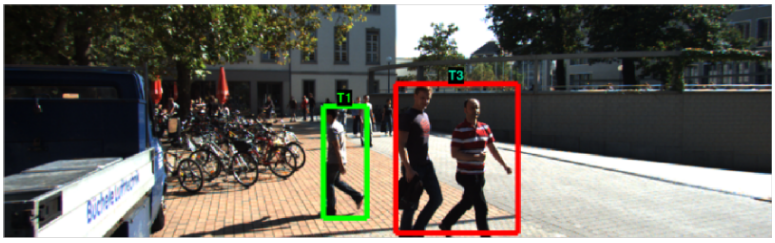
the estimated state trajectories with the estimated data association and vice-versa is particularly important in cases of association ambiguity, i.e. when more than one detection are close to an individual object or when an object gets occluded for a small period of time.

Example 1 - Merging: Consider the case when two objects get close together causing an association ambiguity. Although uninformative for calculating association identities, the observation still provides evidence about the localisation of the objects. Our method makes use of the evidence in ambiguous areas for localisation purposes and recovers the objects identities according to their location history before merging.



(a) Objects $T1$ and $T3$, and their respective detections

(b) Objects $T1$ and $T3$ get close, causing an association ambiguity



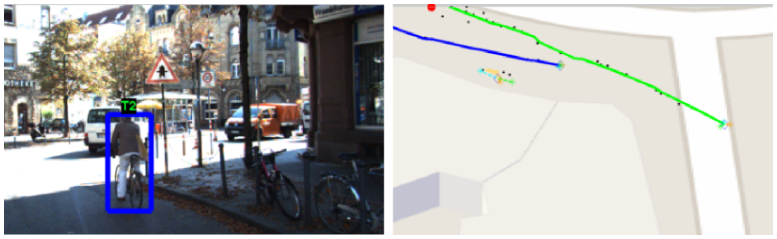
(c) Observations are correctly assigned after the group in (b) splits

Fig. 4. An instance of identity disambiguation carried out by our algorithm

As shown in Fig. 4, although once the grouping occurs (Fig. 4b), there is no detection associated to object $T1$ with a high probability, EA continues estimating the entire object trajectories, and recalculating the data association based on the observations available up to the current time step. This continuous flow of information between trajectory estimation and data association makes it possible to associate $T1$ to the right detection once the grouping is over (Fig. 4c).

Example 2 - Occlusion: A second example, similar in nature to the previous one, is an occlusion. Here, only one of the objects interacting is observed and the occluded object was sufficiently separated before getting occluded so that, no association ambiguity occurs. However, once the occluded object is observed again, it may be difficult to match this observation to its respective track, particularly for vehicles, which suffer drastic appearance changes when they are observed from different perspectives. Our approach predicts the path of the occluded object given all the evidence previous to the occlusion, making it very likely to recover its identity. Fig. 5 shows an instance of this scenario.

(a) Objects $T1$ and $T2$. Detections and identities on the left and trajectories on the right



(b) Object $T1$ is occluded by object $T2$, so the former is not observed

(c) Once the van leaves the occlusion, the online association routine, assigns the new detection to a new object ($T10$)

(d) After smoothing, the detection corresponding to the van is reassigned to object $T1$

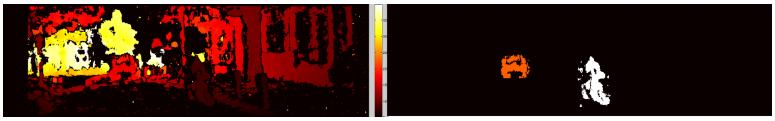
Fig. 5. An instance of identity disambiguation after an occlusion. Images with tracked objects are depicted on the left hand side and object trajectories in a global reference frame on the right (images were cropped to ease visualisation)

5. Experiments

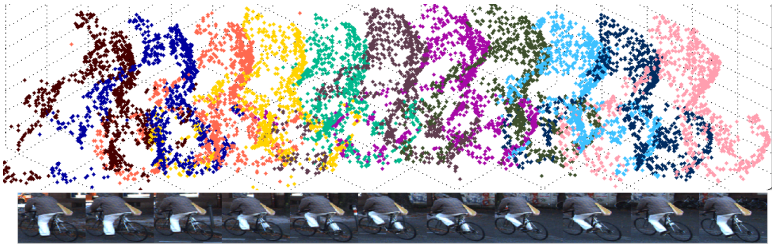
We applied the EA algorithm to online multi-object tracking and classification in urban environments. The dataset utilised has 21 sequences of stereo images that are part of the public KITTI dataset (Geiger, Lenz, Stiller and Urtasun 2013). The dataset contains a significant number of pedestrians and cars interacting in the field of view of a moving platform. The position of the ego-vehicle and ground-truth object detections are provided. Detections also convey ground-truth information about the object class and data association across time. We separated the dataset into training/validation and testing sub-datasets. The testing sub-dataset was selected to be the set of seven sequences with the larger number of interactions between objects. For the testing sub-dataset, detections without any association/class information were used.

The next subsection explains the process of learning the dictionary of motion models. For the quantitative evaluation, we calculate the *Multiple Object Tracking Accuracy* (MOTA) introduced by Bernardin and Stiefelhagen (2008) and the

(a) Stereo pair with detections annotated on left image



(b) Depth image from the stereo processing and individual object segmentations



(c) Sequence of filtered point clouds and image patches

Fig. 6. Features extraction and a training instance.

Mostly-Tracked/Mostly-Lost trajectories (MT/ML) metrics presented by Li et al. (2009), which are evaluation metrics commonly used by the target tracking community. We report these metrics for different state-of-the-art tracking methods and evaluate EA's classification performance.

5.1. Motion Models learning

We learn the parameters of our models using the Expectation Maximisation (EM) algorithm for an MLDS. In our implementation, we use one mixture component per object class. Object classes for the training were chosen to be *Car*, *Cyclist* and *Pedestrian* which, are a subset of all the object categories contained in the KITTI dataset (eight in total). Each training instance consists of a sequence of temporally ordered features extracted from the segmented point cloud of an object in the scene at every time step (see Fig. 6).

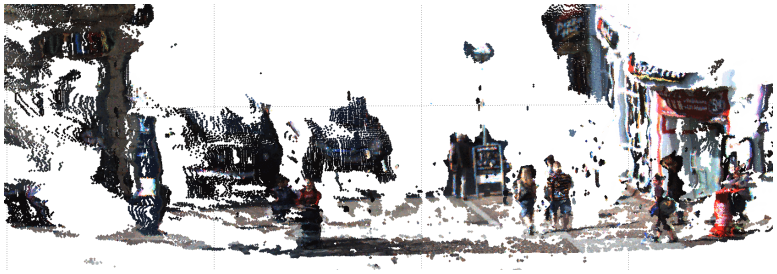
Firstly, a point cloud of the entire scene is obtained by stereo processing the left and right images at time t. This images were obtained from a stereo rig mounted on top of the KITTI ego-vehicle. Secondly, segments of 3D points are extracted from the windows defined by the bounding boxes that accompany the detections. Using the association ground-truth provided with the dataset, point clouds are organised into feature trajectories. Fig. 6 illustrates the process just explained and shows a sequence of point cloud segments corresponding to one training sequence.

After extracting all of the training trajectories, the MLDS is fitted. For this application, the model parameters H and R are shared by all of the motion models due to the fact that only one sensor is used and the model between states and observation features is known.

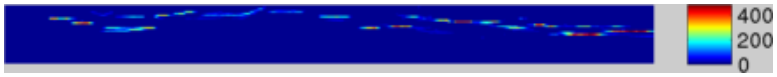
5.2. Object Detection

For detection, we combine 2D detections provided with the dataset and 3D segments obtained from stereo-vision point clouds. At each time step, the point cloud obtained from stereo is projected onto a grid parallel to the ground plane. Subsequently, areas on the grid with high point density are segmented and re-projected to the image plane in order to form well-separated 3D detection candidates. Individual object observations are selected from the 3D detection candidates that overlap with the 2D ones. The detection pipeline is illustrated in Fig. 7.

(a) Stereo pair

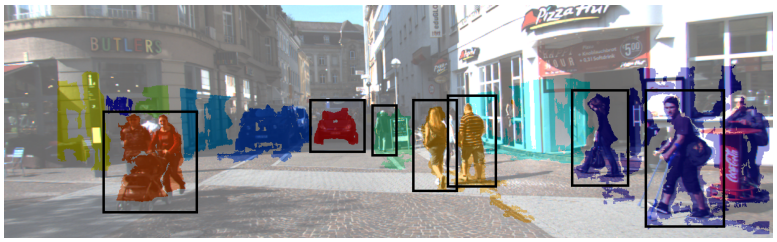


(b) 3D reconstruction obtained from stereo images



(c) Grid with each cell containing the number of 3D points projected on it

(d) Segmented grid with detection candidates



(e) Observations are obtained from 3D detection candidates that are surrounded by bounding boxes (provided with the dataset)

Fig. 7. Object detection pipeline

5.3. Prior Over Association Factors and Tracked Objects Management

Central to our method is the initialisation of the prior distribution $p\left(a_t^{l,i}\right)$ in Eq. (16). Once detections are made, the state of each existent object is time updated and validation gates around them are created. After gating, those detections that were not assigned are initialised as new objects.

In our previous work (Romero-Cano et al. 2014), once an object is updated, the image of its assigned observation is stored. Then the appearance-based prior over associations is obtained by calculating the normalised cross correlation between new observations and object images. In this work, we initialise the prior over associations in a more robust manner. Similar to the work by Morton et al. (2013), we summarise the image information of individual objects before and after association ambiguities using a colour histogram as appearance model. We sequentially update the histogram of object i by averaging the one for the current assigned image patch (detection) and the current appearance model. $p\left(a_t^{l,i}\right)$ is obtained from the histogram-intersection between appearance models and the histograms of the image patches of current detections. Fig. 8 illustrates the process.

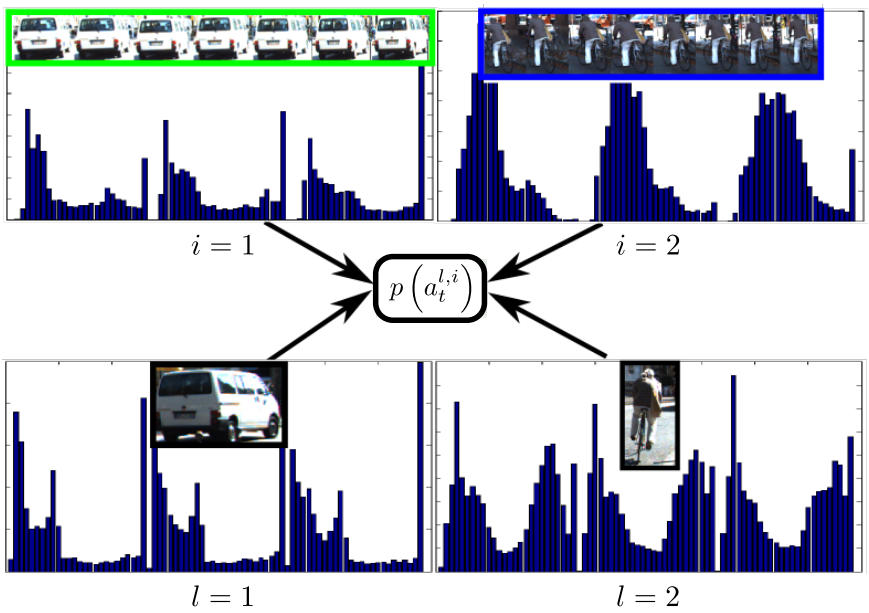


Fig. 8. An appearance-based prior distribution over data associations can be obtained by comparing the colour histogram of tracked objects (indexed by i) and the ones for the current detections (indexed by l). As shown in the top two images, the histogram of each object i summarises its sequence of associated images. The bottom two images show the incoming detection images and their colour histograms.

In terms of object management, we initialise detections that are not inside any of the current object validation gates as new objects. Also, those objects that are not updated during a certain period of time, or leave the field of view of the camera are deleted.

5.4. Performance Evaluation

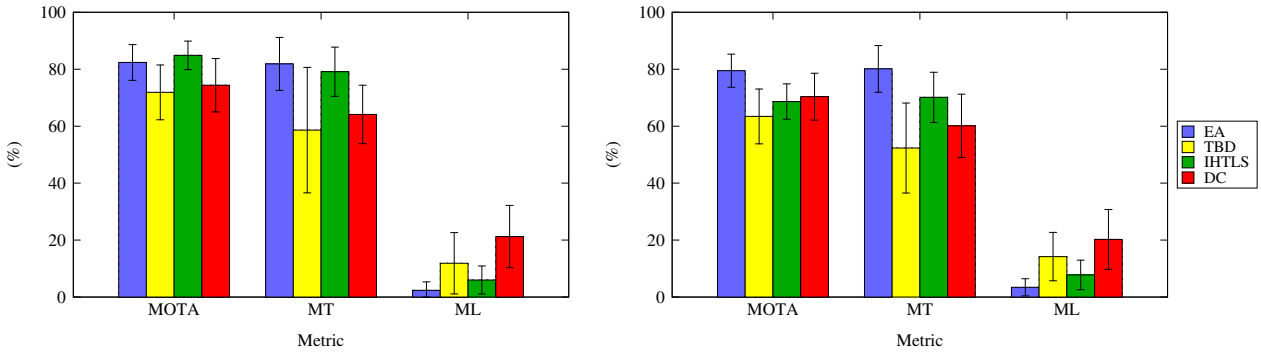
Tracking We compared the EA algorithm to state-of-the-art approaches in multiple objects tracking. We evaluated Discrete-Continuous energy minimization (DC) (Milan et al. 2013) , the Hungarian method for bipartite matching (Geiger, Lauer, Wojek, Stiller and Urtasun 2013) (the authors call their method Tracking By Detection (TBD)) and Iterative Hankel Total Least Squares (IHTLS) (Dicle et al. 2013). For all cases, We used the code provided by the authors.

We first explain the intuition behind the metrics we used for this evaluation. The $MOTA$ metric is calculated as:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}. \tag{23}$$

Where, m_t, fp_t, mme_t and g_t are the number of misses, false positives, mismatches and ground truth objects respectively, for time t. The other two metrics we calculated were MT and ML. They provide the percentage of ground-truth trajectories that were covered by the estimated trajectories for more than 80% and less than 20% in length respectively.

Fig. 9a reports the performance of our approach and that of the compared methods. In average, our method performs better than the others, except for IHTLS. Given how close the MOTA metrics for EA and IHTLS are, we conducted a two-sample t-test for equal means. The null hypothesis is accepted with *p-value* of 0.0165, which means that there is not statistically significant difference between the MOTA metrics for EA and IHTLS. The advantage of our online EA over IHTLS is, however, that our system is sequential and uses past information from small windows (12 frames in our experiments) for performing smoothing. IHTLS is designed to work offline, so it needs information from entire trajectories across all frames in the sequence.



(a) Quantitative evaluation with non-partitioned detections

(b) Quantitative evaluation with partitioned detections

Fig. 9. Quantitative evaluation. Better scores correspond to bigger values of *MOTA* and *MT*, and smaller for *ML*

Robustness against noise: In addition to the potential of real time performance, our approach models the state of the objects using hidden variables, whereas IHTLS uses the raw observations. This is equivalent to assuming that the observations are perfect and complete measurements of the object states. Under our framework, estimation of the data association is done using a smoothed version of the observations, what makes our approach more robust to noise. In order to verify this property under a realistic setting, the detection method was modified so that candidate detections from the grid segmentation were further partitioned and provided to the tracker without any preprocessing (see Fig. 10). We generated these sub-partitions using the Randomized Prim's algorithm (RP) (Manen et al. 2013) for object proposals generation. Sub-partition methodologies are widely used in order to track objects with different geometries or moving closely.

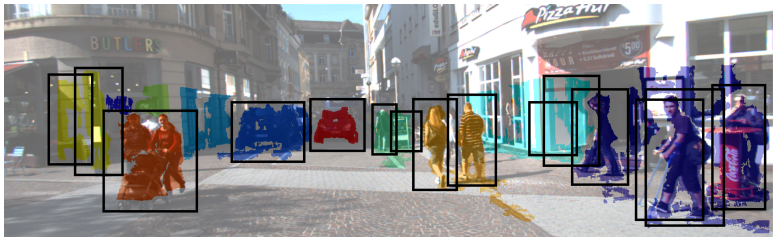


Fig. 10. Sub-partitions on the original detections

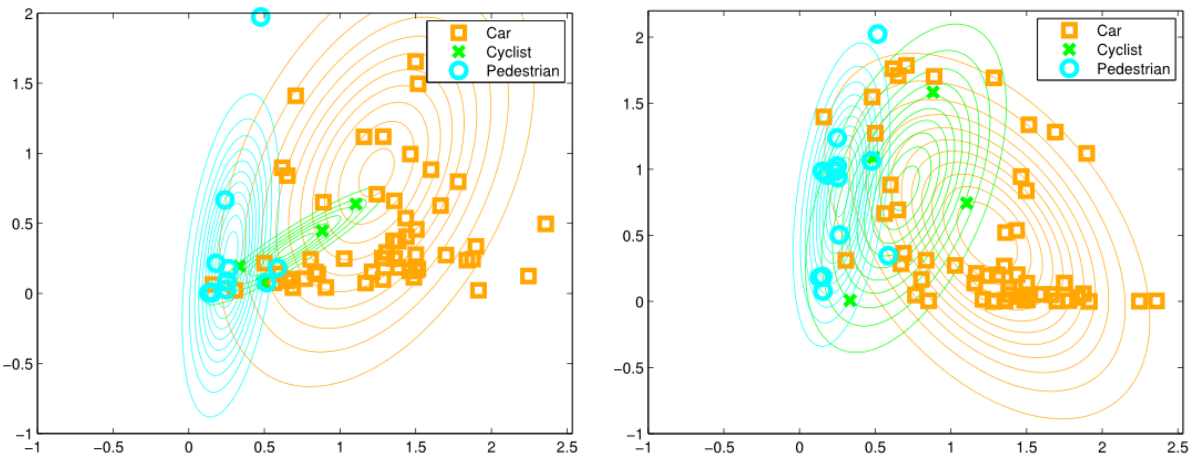
Table 2: Confusion matrix with the classification results across testing sequences (%). The last column shows the number of tracked objects per class

Act.\Pred.	Car	Cyclist	Pedestrian	Total
Car	71	26	3	86
Cyclist	12	82	6	17
Pedestrian	2	5	93	96

The bar graph in Fig. 9b shows how the compared methods have an average drop in performance of 10%, whereas the effect on the performance of EA was less than 5% for all of the metrics. Compared against IHTLS, EA performs as well, with 99.54% confidence (i.e. with a p-value of 0.0046).

Classification Table 2 presents a confusion matrix evaluating the classification performance of the online *EA algorithm*. This table was built by assigning to each object the class label that it took with the highest frequency while it was in the camera's field of view. Values in the main diagonal represent instances of objects to which the correct class category was assigned. They illustrate the descriptive power of our method, obtained by simply accumulating the innovation log-likelihoods under each of the models.

From Table 2 we can observe that there is class overlapping between objects that belong to the class *Cyclist* and instances of both *Car* and *Pedestrian*. To visualise this, we fitted Gaussian density functions to speed variances and heading variances of instances in the training set as a function of their mean velocities and show the contours of these distributions in Fig. 11. It shows how classes *Car* and *Pedestrian* are well separated in this feature space, whereas class *Cyclist* overlaps two classes. This problem can be approached by including extra features.



(a) Speed means versus (x axis) speed variances (y axis)

(b) Speed means versus (x axis) heading variances (y axis)

Fig. 11. The issue of class overlapping. In order to summarise the dynamic behaviour of the training trajectories, we calculated the speed and heading variance of each of these trajectories. We then fit per-class Gaussian distributions to these features and plot de data points along with the contours of the distributions. Note that there is a considerable overlapping between the contours of the class *Cyclist* and the contours of the other two classes

6. Conclusions

This paper develops a new framework for the problem of simultaneous tracking and classification with unknown data association. We proposed a new model for representing the dynamics of multiple objects and introduced the Expectation Association (EA) algorithm, a factorised inference procedure that allows us to estimate object states and classes simultaneously and efficiently. The method fuses objects' appearance and dynamics to solve the data association problem. Furthermore, it is general enough so any sensor modality can potentially be used. The results showed that our online EA

algorithm is more robust to noise than the baseline approaches and achieves state-of-the-art performance even against batch tracking approaches that try to estimate a global solution using the entire observation sequence.

In terms of future directions, ML estimation of model parameters is known to be prone to over-fitting. We are currently working on a regularised procedure for learning the motion models. This will allow us to make learning more robust, and more importantly, it will allow us to safely increase the number of features for describing tracked objects. We are also extending our model in order to incorporate anomalous trajectory detection into the framework.

Further research directions may include integrating detection into the framework. This would require an appearance model more robust than the one we use in the A-step. The popular *tracking-learning-detection* Kalal et al. (2012) does this for single-object tracking. Additionally, the EA algorithm assumes that object classes remain constant. In the current form of the algorithm, if an object changes its dynamic behaviour (for example, a pedestrian who starts riding his bicycle), the assignment probabilities tend to become uniform across the classes involved in the transition and will slowly skew towards the new class as the observations supporting the previous assignment leave the estimation window. In order to quickly account for class switching, change point detection should be included into the framework. Although adding class switching to our model would combinatorially increase the complexity, a variational methodology such as the presented by Agamennoni et al. (2014) could be used to derive an efficient inference procedure. The same kind of methodology could be used to include model parameters as random variables to be inferred, so that objects whose class dynamic model evolves may be accounted for.

Acknowledgment

This work was supported by the Rio Tinto Centre for Mine Automation (RTCMA) and the Australian Centre for Field Robotics (ACFR).

References

Agamennoni, G., Nieto, J. I. and Nebot, E. M. (2012), 'Estimation of Multivehicle Dynamics by Considering Contextual Information', IEEE Transactions on Robotics **28**(4), 855–870.

Agamennoni, G., Worrall, S., Ward, J. R. and Nebot, E. M. (2014), Automated extraction of driver behaviour primitives using Bayesian agglomerative sequence segmentation, in 'IEEE International Conference on Intelligent Transportation Systems', pp. 1449–1455.

Bajracharya, M., Moghaddam, B., Howard, A., Brennan, S. and Matthies, L. H. (2009), 'A Fast Stereo-based System for Detecting and Tracking Pedestrians from a Moving Vehicle', The International Journal of Robotics Research **28**(11-12), 1466–1485.

Bashir, F. I., Khokhar, A. a. and Schonfeld, D. (2007), 'Object trajectory-based activity classification and recognition using hidden Markov models.', IEEE Transactions on Image Processing **16**(7), 1912–9.

Beal, M. J. and Ghahramani, Z. (2003), The Variational Bayesian EM Algorithm for Incomplete Data : with Application to Scoring Graphical Model Structures, in 'Bayesian Statistics', pp. 1–10.

Bernardin, K. and Stiefelhagen, R. (2008), 'Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics', EURASIP Journal on Image and Video Processing **2008**, 1–10.

Bishop, C. M. (2006), Pattern Recognition and Machine Learning.

Brau, E., Guan, J., Simek, K., Pero, L. D., Dawson, C. R. and Barnard, K. (2013), Bayesian 3D tracking from monocular video, in 'International Conference on Computer Vision', pp. 3368–3375.

Buxton, H. (2003), 'Learning and understanding dynamic scene activity: a review', Image and Vision Computing **21**(1), 125–136.

Casey, F., Waterfall, J., Gutenkunst, R., Myers, C. and Sethna, J. (2008), 'Variational method for estimating the rate of convergence of Markov-chain Monte Carlo algorithms', Physical Review E **78**(4), 046704.

Chan, A. B. and Vasconcelos, N. (2008), 'Modeling, clustering, and segmenting video with mixtures of dynamic textures.', IEEE transactions on pattern analysis and machine intelligence **30**(5), 909–26.

Choi, W., Pantofaru, C. and Savarese, S. (2013), 'A General Framework for Tracking Multiple People from a Moving Camera.', IEEE transactions on pattern analysis and machine intelligence **35**(7), 1577–91.

Dicle, C., Camps, O. I. and Sznaier, M. (2013), The Way They Move : Tracking Multiple Targets with Similar Appearance, in 'IEEE International Conference on Computer Vision', pp. 2304–2311.

Ess, A., Schindler, K., Leibe, B. and Van Gool, L. (2010), 'Object Detection and Tracking for Autonomous Navigation in Dynamic Environments', The International Journal of Robotics Research **29**(14), 1707–1725.

Frank, O., Nieto, J., Guivant, J. and Scheding, S. (2003), Multiple Target Tracking using Sequential Monte Carlo Methods and Statistical Data Association, in 'IEEE/RSJ International Conference on Intelligent Robots and Systems', Vol. 00, pp. 2718–2723.

García-García, D., Parrado-Hernández, E. and Diaz-de Maria, F. (2011), 'State-space dynamics distance for clustering sequential data', Pattern Recognition **44**(5), 1014–1022.

Gavrila, D. M. and Munder, S. (2006), 'Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle', International Journal of Computer Vision **73**(1), 41–59.

Geiger, A., Lauer, M., Wojek, C., Stiller, C. and Urtasun, R. (2013), '3D Traffic Scene Understanding from Movable Platforms.', IEEE transactions on pattern analysis and machine intelligence pp. 1–14.

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013), 'Vision meets Robotics: The KITTI Dataset', The International Journal of Robotics Research .

Granström, K., Lundquist, C., Gustafsson, F. and Orguner, U. (2014), 'Random Set Methods: Estimation of Multiple Extended Objects', IEEE Robotics & Automation Magazine (June), 73–82.

Granström, K., Lundquist, C. and Orguner, O. (2012), 'Extended Target Tracking using a Gaussian-Mixture PHD Filter', IEEE Transactions on Aerospace and Electronic Systems **48**(4), 3268–3286.

Gu, Y. and Veloso, M. (2009), 'Effective Multi-Model Motion Tracking using Action Models', The International Journal of Robotics Research **28**(1), 3–19.

Held, D., Levinson, J., Thrun, S. and Savarese, S. (2014), Combining 3D Shape , Color , and Motion for Robust Anytime Tracking, in 'Robotics: Science and Systems'.

Hensman, J., Rattray, M. and Lawrence, N. (2012), Fast Variational Inference in the Conjugate Exponential Family., in 'NIPS', pp. 1–9.

Huang, C., Li, Y. and Nevatia, R. (2013), 'Multiple target tracking by learning-based hierarchical association of detection responses.', IEEE transactions on pattern analysis and machine intelligence **35**(4), 898–910.

Kaempchen, N., Schiele, B. and Dietmayer, K. (2009), 'Situation Assessment of an Autonomous Emergency Brake for Arbitrary Vehicle-to-Vehicle Collision Scenarios', IEEE Transactions on Intelligent Transportation Systems **10**(4), 678–687.

Kalal, Z., Mikolajczyk, K. and Matas, J. (2012), 'Tracking-Learning-Detection.', IEEE transactions on pattern analysis and machine intelligence **34**(7), 1409–1422.

Kanazaki, H., Yairi, T., Machida, K., Kondo, K. and Matsukawa, Y. (2007), 'Variational Bayes Data Association Filter', 3rd International Conference on Intelligent Sensors, Sensor Networks and Information pp. 401–406.

Katz, R., Nieto, J. and Nebot, E. (2010), 'Unsupervised Classification of Dynamic Obstacles in Urban Environments', Journal of Field Robotics **27**(4), 450–472.

Kschischang, F. R., Member, S., Frey, B. J. and Loeliger, H.-a. (2001), 'Factor Graphs and the Sum-Product Algorithm', IEEE Transactions on Information Theory **47**(2), 498–519.

Li, X. R. (2007), Optimal Bayes Joint Decision and Estimation, in 'International Conference on Information Fusion', Vol. 3950.

Li, Y., Huang, C. and Nevatia, R. (2009), Learning to Associate : HybridBoosted Multi-Target Tracker for Crowded Scene, in 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 2953–2960.

Mahler, R. (2003), 'Multitarget bayes filtering via first-order multitarget moments', IEEE Transactions on Aerospace and Electronic Systems **39**(4), 1152–1178.

Mahler, R. (2009), PHD filters for nonstandard targets, I: Extended targets, in 'IEEE International Conference on Information Fusion', pp. 448–452.

Manen, S., Guillaumin, M. and Gool, L. V. (2013), Prime Object Proposals with Randomized Prim âŁ™ s Algorithm, in 'International Conference on Computer Vision', pp. 4321–4328.

Meissner, D., Reuter, S., Strigel, E. and Dietmayer, K. (2014), 'Intersection-Based Road User Tracking Using a Classifying Multiple-Model PHD Filter', IEEE Intelligent Transportation Systems Magazine **6**(April 2014), 21–33.

Milan, A., Schindler, K. and Roth, S. (2013), 'Detection- and Trajectory-Level Exclusion in Multiple Object Tracking', IEEE Conference on Computer Vision and Pattern Recognition pp. 3682–3689.

Moosmann, F. and Stiller, C. (2013), Joint Self-Localization and Tracking of Generic Objects in 3D Range Data, in 'IEEE International Conference on Robotics and Automation', pp. 1138–1144.

Morton, P., Douillard, B. and Underwood, J. (2013), 'Multi-sensor identity tracking with event graphs', IEEE International Conference on Robotics and Automation pp. 4742–4748.

Oh, S. M., Rehg, J. M., Balch, T. and Dellaert, F. (2007), 'Learning and Inferring Motion Patterns using Parametric Segmental Switching Linear Dynamic Systems', International Journal of Computer Vision **77**(1-3), 103–124.

Pasha, S. A., Vo, B.-N., Tuan, H. D. and Ma, W.-K. (2009), 'A Gaussian Mixture PHD Filter for Jump Markov System Models', IEEE Transactions on Aerospace and Electronic Systems **45**(3), 919–936.

Rauch, H. E., Striebel, C. T. and Tung, F. (1965), 'Maximum likelihood estimates of linear dynamic systems', Journal of American Institute of Aeronautics and Astronautics **3**(8), 1445–1450.

Ren, C. Y., Prisacariu, V., Kaehler, O., Reid, I. and Murray, D. (2014), '3D Tracking of Multiple Objects with Identical Appearance Using RGB-D Input', 2nd International Conference on 3D Vision pp. 47–54.

Reuter, S., Wilking, B., Wiest, J., Munz, M. and Dietmayer, K. (2013), 'Real-Time Multi-Object Tracking using Random Finite Sets', IEEE Transactions on Aerospace and Electronic Systems **49**(4), 2666–2678.

Romero-Cano, V., Agamennoni, G. and Nieto, J. (2014), A Variational Approach to Simultaneous Tracking and Classification of Multiple Objects, in 'International Conference on Information Fusion', number d.

Romero-Cano, V. and Nieto, J. I. (2013), Stereo-based Motion Detection and Tracking from a Moving Platform, in 'Intelligent Vehicles Symposium', pp. 499–504.

Rong Li, X. (2007), 'Joint tracking and classification based on bayes joint decision and estimation', 10th International Conference on Information Fusion pp. 1–8.

Schumitsch, B., Thrun, S., Guibas, L. and Olukotun, K. (2006), The Identity Management Kalman Filter (IMKF), in 'Robotics: Science and Systems'.

Segal, A. V. and Reid, I. (2013), Latent Data Association : Bayesian Model Selection for Multi-target Tracking, in 'IEEE International Conference on Computer Vision', pp. 2904–2911.

Vasquez, D., Fraichard, T. and Laugier, C. (2009), 'Growing Hidden Markov Models: An Incremental Tool for Learning and Predicting Human and Vehicle Motion', The International Journal of Robotics Research **28**(11-12), 1486–1506.

Vatavu, A., Danescu, R. and Nedevschi, S. (2014), 'Stereovision-Based Multiple Object Tracking in Traffic Scenarios Using Free-Form Obstacle Delimiters and Particle Filters', IEEE Transactions on Intelligent Transportation Systems pp. 1–14.

Wang, C.-C., Thorpe, C., Thrun, S., Hebert, M. and Durrant-Whyte, H. (2007), 'Simultaneous Localization, Mapping and Moving Object Tracking', The International Journal of Robotics Research **26**(9), 889–916.

Wojek, C., Walk, S., Roth, S., Schindler, K. and Schiele, B. (2012), 'Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes.', IEEE transactions on pattern analysis and machine intelligence .