# A Variational Approach to Simultaneous Tracking and Classification of Multiple Objects

Victor Romero-Cano, Gabriel Agamennoni and Juan Nieto
Australian Centre for Field Robotics
The University of Sydney
Email: varomero,g.agamennoni,j.nieto@acfr.usyd.edu.au

***Abstract*—This paper presents a method for multi-object tracking which provides estimates of the dynamic state of the objects along with class identities. The estimated identities provide information about the objects' behaviour, improving high level reasoning tasks. However, jointly estimating class assignments, dynamic states and data associations results in a computationally intractable problem. This paper proposes a probabilistic model for the multi-object tracking and classification problem, and an inference procedure that renders the problem tractable through a variational approximation. Our framework integrates the efficient Kalman filtering and smoothing recursions into a system that considers the dynamics of the environment to leverage both tracking and classification. The method is evaluated and compared to state-of-the-art techniques using stereo-vision data collected from a moving platform in urban scenarios.**

## I. Introduction

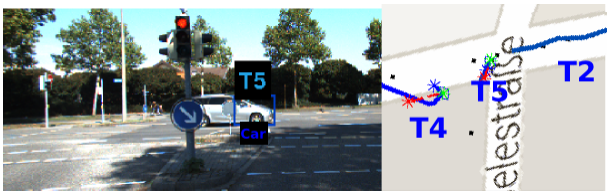
Situational Awareness (SA) is a vital component for mobile systems expected to work in dynamic environments [1]. SA systems are typically designed using three levels of information: *perception, comprehension and projection* [9]. The perception level builds a 'picture' of the environment, the comprehension level provides a 'meaning' to the different elements and the projection level predicts their states in the near future. For this paper, we focus on the first two. The applications that we target are Autonomous Robots (AR) and Advanced Driving Assistance Systems (ADAS). In the context of these applications, Multi-Target Tracking (MTT) is the procedure that, at a low level, provides information about what objects of interest there are in the environment and their behavioural characteristics [14].

Most of the approaches to tracking and classification decouple state estimation from identity assignment [3], [25], [11]. Hence, they neglect the natural correlations between object dynamics and its categories. This paper presents a probabilistic framework that provides joint inference of objects' class and states with unknown data association. Furthermore, this is done by employing classic and efficient statistical estimation techniques such as the Kalman filtering and smoothing recursions [23].

### A. Dynamic scene understanding

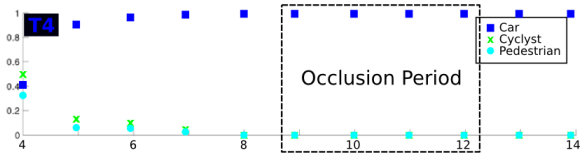
We decompose the dynamic scene understanding task as follows. Firstly, the dynamics of specific object categories is summarised by linear motion models. Secondly, estimates of the objects' kinematic states and their assignment probabilities to the different motion patterns are obtained. Casting dynamic scene understanding under this perspective has several


(a) Frame 6. Objects *T4* and *T2* are correctly classified as *car*.




(b) Frame 10. *T4* is lost due to occlusion and *T5* is initialised.


(c) Frame 14. Our method recovers *T4* by exploiting its consistent dynamics before and after the occlusion.




(d) Class assignment probabilities for object *T4* across time. The top probability consistently corresponds to the class *car*.

Fig. 1: A car tracked and classified through an occlusion. Subfigures *(b)* and *(c)* show the estimated trajectories next to the left stereo-image. Dots, asterisks and solid lines represent observations, per-class state hypothesis and estimated trajectories respectively.

advantages, including the fact that it can boost state-of-the-art appearance-based object classification methods [12], [26], by exploiting motion information and temporal correlations in the data. In addition, it provides a mechanism to exploit prior knowledge about the objects' dynamics.

In this paper, we represent motion patterns using Linear Dynamical Systems (LDS). In our implementations, LDS are learnt from labelled trajectories. Our method simultaneously estimates the kinematic state and the class identity probabilities of a set of test tracks (Fig. 1). Our system outputs state estimates for all of the objects in the scene and soft assignments of each target to the motion categories or classes.

We validate our approach with a publicly available dataset. The data consists of a sequence of stereo images, collected with a camera mounted on top of a car [13]. We present a complete theoretical analysis of our approach, and the steps to apply it to simultaneous multiple-target tracking and classification. The specific contributions of this paper are:

- a model for dynamic scenes and a variational procedure to perform approximate inference on this model;
- the Expectation-Association (EA) algorithm: a method for performing data association, state estimation and class identity assignment in a joint probabilistic fashion. It addresses both the offline and online schemes;
- solving the data association problem by fusing appearance features with dynamic information in a unified probabilistic framework;
- validation of the theoretical approach with real data collected in an urban environment.

The paper is organised as follows. Section II presents a review of related work. Section III describes our probabilistic model for describing dynamic scenes. In Section IV we present our variational approximation with an extension to the online case. Finally, experimental results are presented in Section V followed by conclusions in Section VI.

## II. Related Work

Multi-target tracking is a well-known problem in the robotics community and many publications on the matter have been produced [10], [20], [21], [7]. In general, they estimate the target's states and data associations without an explicit assignment to categories. Complete tracks obtained from an independent tracking system are classified using either similarity-based clustering techniques [16] or Hidden Markov Models [3], [11]. Some exceptions to this trend of separating tracking and classification are [24], [22], [1]. The work proposed here is similar in spirit to these later papers. Namely, [24] applies a method for joint decision and estimation to the problem of tracking and classification in a hypothesis testing framework, but assumes known data association. Conversely, our approach seamlessly infers data association with the objects states.

In [22], 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 [5]. Therefore, we have opted for a variational method, where analytical approximations of the posterior of interest are obtained[1].

From a theoretical perspective, the work presented in this paper is similar to that of [15], where a variational method for multi-target tracking with unknown data association is presented. A key difference, however, is that [15] only considers a single model, common to all the targets. 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 patterns. Additionally, the results presented in [15] were obtained from synthetic data only, whereas we extensively evaluate our method using stereo images from an urban scenario. The work presented in [8] presents an offline tracker that uses dynamics to create object trajectories by associating small sub-tracks when no appearance information is available. Similarly, our work creates object tracks by enforcing smooth trajectories across time. Our method also works online and provides a theoretical method to integrate appearance and dynamic information for solving both the classification and data association problem.

Other approaches to multi-object tracking are based on Random Finite Set (RFS) to represent the objects state. A tracking framework based on the Gaussian Mixture implementation of the Cardinalised Probability Hypothesis Density (GMCPHD) filter [19] was presented in [17]. This work uses appearance to classify tracks. In contrast our framework allows to incorporate dynamic information as well as appearance.

## III. Model Overview

Within our framework, estimating the state of multiple objects boils down to solving three interleaved problems, namely *class identity assignment*, *state estimation* and *data association*.

We propose the generative graphical model presented in Fig. 2. In this model, $\Theta$ represents the model parameters, which are learnt from data; Each node $S^k$ is a categorical random variable used for indexing 1 of $N_s$ models representing different motion patterns. $X_t^k$ is a continuous random variable representing the state of target $k$ at time $t$, and $Z_t$ are the observations at time $t$. Finally, $A_t$ is a set of $N_{z,t}$ categorical variables modelling the association between observations and tracked targets. We further define the following sets of random variables:

$$\begin{aligned} S &= \left[S^1 ... S^{Nx}\right];\ X = \left[X^1_{1:T_1} ... X^{Nx}_{1:T_{Nx}}\right]; \\ A &= [A_1 ... A_T];\quad Z = \left[Z^1_{1:N_{z,1}} ... Z^T_{1:N_{z,T}}\right] \end{aligned} \tag{1}$$

where $N_{z,t}$ is the number of detections at time $t$; $T_k$ is the size of track $k$ and $T$ the size of the entire tracking sequence.

## IV. The Expectation-Association Algorithm

The joint probability distribution for our model (Fig. 2) can be written as:

$$p\left(S, X, A, Z\right) =$$
$$p(S) \prod_{t=1}^{T} p\left(X_t | X_{t-1}, S\right) \prod_{t=1}^{T} p(Z_t | X_t, A_t, S) \prod_{t=1}^{T} p(A_t), \tag{2}$$

[1]Please note that a comparison against sampling methods for approximate inference is not within the scope of this paper.

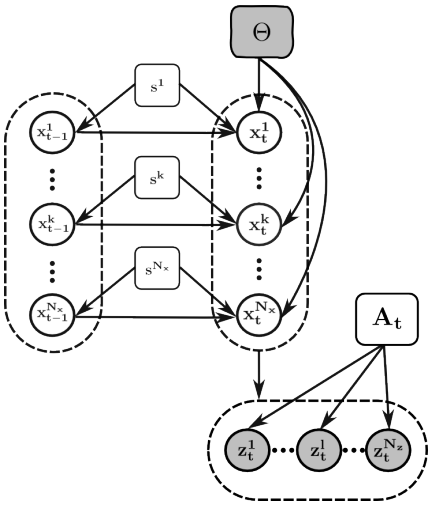



Fig. 2: Two time slices of our Graphical Model. Squared and circular nodes represent categorical and continuous random variables respectively. Unfilled nodes indicate hidden variables, while filled nodes are observed

in which the factors are, from left to right, the prior class assignment probabilities, the state transition and the observation distributions and the prior association probabilities.

We would like to estimate the posterior $p(S, X, A|Z)$ over classes, targets states and associations by maximising the likelihood of the data. The log likelihood of the data is obtained by marginalising out the set of hidden variables $(S, X, A)$ given a model $(\Theta)$ and a set of data $(Z)$:

$$\ln p(Z) = \ln \sum_{S,A} \int p\left(S, X, A, Z\right) dX \tag{3}$$

Unfortunately, this integral is both analytically and computationally intractable due to the coupling between variables.

To get a better insight into this issue, 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

$$\ln \sum_{S,A} \int q\left(S, X, A\right) \frac{p\left(S, X, A, Z\right)}{q\left(S, X, A\right)} dX \tag{4}$$

and applying Jensen's inequality [5] we arrive at a lower bound

$$\ln p(Z) \geq \sum_{S,A} \int q\left(S, X, A\right) \ln \frac{p\left(S, X, A, Z\right)}{q\left(S, X, A\right)} dX = \mathcal{L}\left[q\right] \tag{5}$$

on the marginal log-likelihood (Eq. (3)) of the target observations under our model. This inequality holds for any choice of $q$. In particular, if $q(S, X, A)$ equals the true posterior $p(S, X, A|Z)$, then Eq. (5) becomes an equality. By using the *d-separation criterion* [5] it can be seen that, although multi-target's states and associations are marginally independent, they become conditionally dependent given the observation sequence. As a result of these dependencies, the posterior is a mixture distribution where the number of components increases combinatorially with the number of targets and exponentially with time.

Since the exact posterior is computationally intractable, we propose approximating it with a probability density function $q$ that separates classes and states from data associations. Therefore it factorises as follows:

$$q\left(S, X, A\right) = q(S, X)q(A). \tag{6}$$

This factorisation does not imply that we are ignoring the interactions between states/classes and associations. It rather means that our approximation does not capture any ambiguities that remain once the entire sequence of data has been observed. In the context of our application, object trajectories tend to be temporally coherent, so that, given the data in a reasonably big estimation window, the estates and class identity of the objects can be estimated independently of the data association (See Fig. 3).

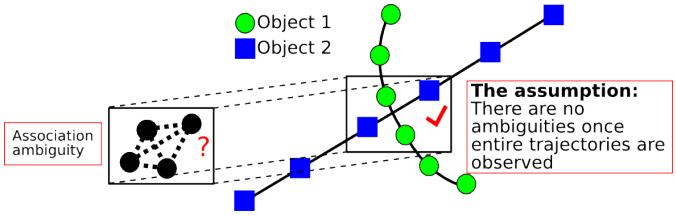



Fig. 3: An intuitive explanation of our variational approximation. We assume that, once entire trajectories are observed, there are no ambiguities about the association between object states and observations.

Unlike the exact posterior, our approximate solution in Eq. (6) is computationally tractable. We derive the expressions for the factors in Eq. (6) by maximising Eq. (5). As shown in [5], 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 $i \neq j$.

Factors $q(S, X)$ and $q(A)$ are updated as explained in Subsections IV-A and IV-B respectively, cycling through them until convergence. We call this iterative process the *Expectation-Association (EA)* algorithm and introduce its batch version with the pseudo-code in Algorithm 1. Eqs. (9)-(11) make clear the intuition behind our approximation: Once a complete track $k$ has been observed, its state, class assignment and associations to observations can be estimated independently.

### A. The Expectation Step (E-step)

The first factor of our variational factorisation is $q(S, X)$. By maximising the lower bound (Eq. (5)) we obtain an expression for the $q$ factor that relates motion patterns to target states. From Eq. (9) and Eq. (10) one can see that the optimal $q(S, X)$ is a Gaussian mixture distribution, with one component for each motion pattern.

Since the distribution in Eq. (9) has a quadratic form, it is efficiently calculated using the Kalman filter (KF) and the Rauch-Tung-Striebel (RTS) smoother [23]. In this expression, the term $\overline{Z}_t^k$, given by

**Algorithm 1** The batch EA algorithm

1: $Models \leftarrow$ Fit LDSs to training trajectories indicative of motion patterns in the environment.
2: $q(A) \leftarrow$ Initialise observation-to-target association probabilities.
3: **procedure E-STEP**($Models$, $q(A)$)
4: $\quad \overline{Z}_t^k \leftarrow$ Calculate per-target average observation.
5: $\quad \overline{R}_t^{k,j^{-1}} \leftarrow$ Calculate per-target/per-model observation noise covariance.
6: $\quad \sum_{t=2}^{T_k} l_t^{k,j} \leftarrow$ Perform filtering and obtain innovation log-likelihoods.
7: $\quad q(X|S) \leftarrow$ Run RTS smoother and obtain per-model posterior over targets states.
8: $\quad q(S) \leftarrow$ Calculate marginal over class assignments.
9: **end procedure**
10: **procedure A-STEP**($Models$,$q(S, X)$)
11: $\quad q(A) \leftarrow$ Update the association probabilities.
12: **end procedure**
13: Repeat until convergence.

$$\overline{Z}_t^k = \frac{\sum_{l=1}^{N_z} q\left(a_t^{l,k}\right) Z_t^l}{\sum_{l=1}^{N_z} q\left(a_t^{l,k}\right)}, \tag{7}$$

is a weighted average of the observations with weights proportional to the posterior association probabilities of all the observations and target $k$. Additionally, Eq. (9) can be seen as an LDS parametrised by $X_0^j$, $Cov(X_0^j)$, $F^j$, $Q^j$, $H^j$ and the average observation noise covariance matrix

$$\overline{R}_t^{k,j^{-1}} = \sum_{l=1}^{N_z} q\left(a_t^{l,k}\right) R^{j^{-1}}. \tag{8}$$

Note that the marginal over the assignment variables in Eq. (10) 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 $l_t^{k,j}$ computed during each correction or update step by the Kalman filter on the $j$th LDS.

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.

### B. The Association Step (A-step)

The second factor of our variational factorisation is $q(A)$. The initial factorisation in Eq. (6) results in other factorisations across time and within observations. Note that these are *induced* factorisations, i.e., they do not concede additional accuracy and are exact given the initial assumption in Eq. (6). Therefore, the natural logarithm of $q(A)$ can be expressed as:

$$\ln q(A) = \sum_{t=1}^{T} \sum_{l=1}^{Nz} \ln q\left(a_t^l\right), \tag{12}$$

where $a_t^l$ is a categorical random variable over the associations of detection $l$ to the tracked objects at time $t$. We obtain each of the sub-factors in Eq. (12) by maximising Eq. (5) with respect to $q(a_t^l)$ as explained at the beginning of Section IV. In these factors, given by Eq. (11), $\hat{X}_t^{k,j}$ 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 target $k$ to observation $l$ at time $t$ decreases when the uncertainty about the state of target $k$ (state covariance) increases.

Another advantage of our formulation is that it allows us to integrate appearance and dynamics when calculating the association between observations and targets. In most cases, there are several sources of information about the association between targets and observations. The prior over associations $q(a_t^l)$ can be calculated, for example, based on appearance features; the inference algorithm then computes the posterior by seamless fusing this prior with evidence from the target's dynamics.

### C. Innovations Accumulation: The Online EA Algorithm

Although our model reasons on entire trajectories, the form of the factors in our approximation lend themselves to an

$$\begin{aligned} \ln q\left(X_{1:T}^k | S^k = j\right) \propto -\frac{1}{2}\Bigg( & \sum_{t=1}^{T_k} \left(X_t^k - F^j X_{t-1}^k\right)^T Q^{j^{-1}} \left(X_t^k - F^j X_{t-1}^k\right) \\ & + \left(\bar{s}_t^k - H^j X_t^k\right)^T \sum_{l=1}^{N_z} q(a_t^{l,j}) R^{j^{-1}} \left(\overline{Z}_t^k - H^j X_t^k\right) \Bigg) \end{aligned} \tag{9}$$

$$\ln q(S^{k,j}) \propto \ln p(S^{k,j}) + \sum_{t=2}^{T_k} l_t^{k,j} \tag{10}$$

$$\begin{aligned} \ln q\left(a_t^{l,k}\right) \propto \ln p\left(a_t^{l,k}\right) - \frac{1}{2}\Bigg( & \sum_{j=1}^{N_s} q\left(S^{k,j}\right) \Bigg( \left(Z_t^l - H^j \hat{X}_t^{k,j}\right)^T R^{j^{-1}} \left(Z_t^l - H^j \hat{X}_t^{k,j}\right) \\ & + Tr\left(H^{j^T} R^{j^{-1}} H^j Cov\left(\hat{X}_t^{k,j}\right)\right)\Bigg)\Bigg) \end{aligned} \tag{11}$$

online estimation algorithm. As shown in Eq. (10), the assignment probabilities are obtained by accumulating the innovation likelihoods of the targets under each of the models. Therefore, when applying our method online, we simply filter each track using each of the models and accumulate their innovation likelihoods so that the class assignment probabilities can be recalculated at each time step.

Regarding the A-step, the association factors are updated using entire smoothed trajectories. Nevertheless, we use the filtered states as in [15], but also perform fixed-lag smoothing. Even though we are maximising a local version of the lower bound (from time 1 to $t$), i.e., with respect to a subset of all of the random variables of the batch case, the algorithm still converges. The entire tracking process is summarised in Algorithm 2.

**Algorithm 2** The online EA algorithm

$Models \leftarrow$ Fit LDSs to training trajectories indicative of motion patterns in the environment
**for** $t \leftarrow 1, T$ **do**
  $q(A_t) \leftarrow$Initialise association probabilities
  $\overline{Z}_t^k \leftarrow$ Calculate per-target average observation.
  $\overline{R}_t^{k,j^{-1}} \leftarrow$ Calculate per-target/per-model observation noise covariance
  **procedure E-STEP**($Models$,$q(A_t)$,$\overline{Z}_t^k$,$\overline{R}_t^{k,j^{-1}}$)
    $q(X_t|S_t) \leftarrow$ Perform filtering
    $\sum l_{1:t}^{k,j} \leftarrow$ Accumulate innovation log-likelihoods.
    $q(S_t) \leftarrow$ Calculate class assignment probabilities
  **end procedure**
  **procedure A-STEP**($Models$,$q(S, X_t)$)
    $q(A_t) \leftarrow$ Update the association factors using the filtered states.
  **end procedure**
  Perform **E-step** with the updated associations.
  Perform **Fixed-lag Smoothing**.
**end for**

## V. Experimental Results

We applied the online version of the proposed *EA algorithm* to stereo vision-based tracking and evaluated its performance on the public KITTI dataset [13]. The dataset, which is conformed by a total of 20 sequences, 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 labels with ground-truth detections (bounding boxes) of the objects in the scene are provided. Detections also convey ground-truth information about the object class and data association across time. We selected a set of 15 trajectories per object category from sequences $S02 - S09$ for training. These sequences have an average size of 300 frames. Sequences $S11$ and $S19$ were utilised for testing purposes, each of which are 372 and 1059 frames long respectively.

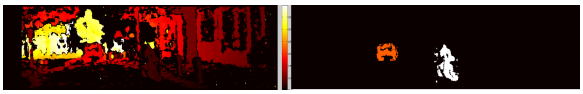
Subsection V-A explains the process by which the dictionary of motion models is obtained. Then, for the quantitative evaluation, we calculate the *Multiple Object Tracking Accuracy (MOTA)* [4] and *Mostly-Tracked (MT) / Mostly-Lost (ML) trajectories* [18] metrics, which are evaluation metrics commonly used by the target tracking community. In the KITTI benchmark [13] these same metrics were reported for some state-of-the-art approaches to MTT. Finally, we report on the classification and capabilities of the method.

### A. Training

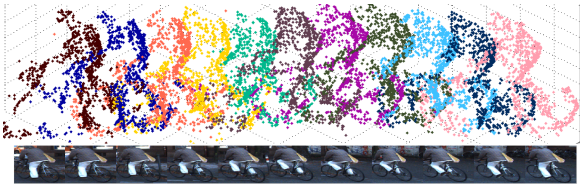
Object classes for the training were chosen to be *Car*, *Cyclist* and *Pedestrian* which, are a subset of all the object categories contained in the KITTI dataset (eight in total). Each training instance consists of a sequence of temporally ordered features extracted from the segmented point cloud of an object in the scene at every time step (See Fig. 4).

Firstly, a point cloud of the entire scene is obtained by stereo processing the left and right images at time t. Secondly, segments of 3D points are extracted from the windows defined by the bounding boxes that accompany the detections. Segments are further filtered by organizing them into an Octree and deleting points that fall into bins with a depth of 1. We found that this procedure cleans sparse and disconnected sets of points that in a segment usually correspond to noise. Using the association and object category ground truth provided with the dataset, point clouds are organised into feature trajectories. Fig. 4 illustrates the process just explained and shows a sequence of point cloud segments corresponding to one training sequence.

(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. 4: Features extraction and a training instance.

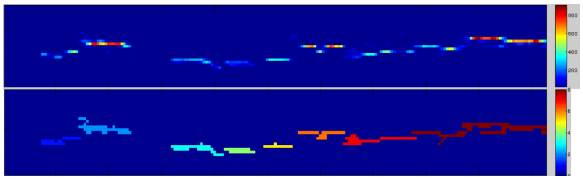
After extracting all of the training trajectories, a set of LDS models per object category are fitted by means of the EM algorithm. From those models, the matrices $F$ and $Q$ that parametrise the state transition probabilities and the prior over initial states ($X_0$) are learnt. In this application, 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.

### B. Object detection

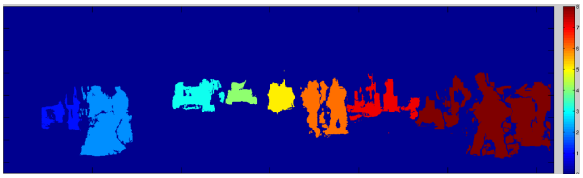
During tracking, object detections are obtained at every time step. At time $t$, the entire scene is segmented into coarse semantic categories following [6]. One of these categories

is the category *object* which highlights the regions on the image where objects might be. We then project 3D points corresponding to pixels labelled as *object* onto a polar grid parallel to the ground plane. Subsequently, areas on the grid with high point density are segmented and re-projected to the image plane, resulting in individual object detections. The detection subsystem is depicted in Fig. 5.

(a) Coarse semantic segmentation. Regions on the image belonging to the category *object* are highlighted.



(b) Individual detections obtained by projecting points to grid on the ground plane.



(c) Detections on the image plane.

Fig. 5: Object detection generation during tracking.

### C. Prior Over Association Factors and Tracked Objects Management

Central to our method is the initialisation of the prior distribution $p\left(a_t^{l,k}\right)$ in Eq. (11). It provides a principled manner to include spatial knowledge about the association between detection $l$ and object $k$ at time $t$. For our application, this prior is calculated making use of the rich information that is available from stereo-vision. Once detections are made, the state of each existent object is time updated and validation gates around them are created.

Both the point clouds and image patches of those detections that are inside at least one of the validation gates are compared against point clouds and image patches associated with the tracked objects. For point clouds, we calculated the association distance as the number of points of the detections that fall inside the convex hulls generated by the tracked objects. Note that overlapping (in a global reference frame) is a safe assumption given a camera's frame rate of $10fps$, and the fact that objects in urban environments are constrained to low velocities. For the image patches, we use the correlation coefficient as the similarity metric. Then, these two similarity matrices are averaged. Detections that were not assigned are initialised as new objects. Finally, the prior over the data association at time $t$ is calculated by normalising the similarities between assigned objects and assigned detections. In the paper we defined $N_z$ as the number of assigned detections and $N_x$ as the number of objects. Objects that were not updated during a certain period of time or that leave the field of view of the camera are deleted.

### D. Performance Evaluation

In order to compare the performance of our approach with state-of-the-art methods (*Discrete-Continuous optimization (DC)* [2] and *Tracking By Detection (TBD)* [27]), we use the $MOTA$ metric, which is calculated as:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \qquad (13)$$

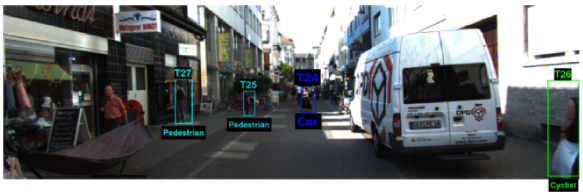
in which $m_t$, $fp_t$, $mme_t$ and $g_t$ are the number of misses, false positives, mismatches and ground truth objects respectively, for time $t$. The other two metrics we calculated were $MT$ and $ML$. They provide the percentage of ground-truth trajectories that were covered by the estimated trajectories for more than $80\%$ and less than $20\%$ in length respectively. Table I reports the performance of our approach and that of state-of-the-art methods. It shows that the *EA algorithm* outperforms the two other methods in all of the metrics considered, with the exception of the runtime. Note that our implementation has not been optimised.

TABLE I: Qualitative evaluation according to the *MOTA* [4] and *Mostly-Tracked/Partly-Tracked/Mostly-Lost* [18] metrics. Better scores correspond to bigger values of *MOTA* and *MT* and smaller for *ML* and *Runtime*.

| Sequence S11 | | | | |
|---|---|---|---|---|
| **Method** | **MOTA** | **MT** | **ML** | **Runtime** |
| Our EA | **89.15 %** | **90.38 %** | **1.01 %** | 0.1 s |
| DC [2] | 87.10 % | 76.52 % | 17.42 % | 0.1 s |
| TBD [27] | 88.73 % | 82.69 % | 3.85 % | 1 s |
| Sequence S19 | | | | |
| **Method** | **MOTA** | **MT** | **ML** | **Runtime** |
| Our EA | **77.63 %** | **85.87 %** | **1.08 %** | 0.1 s |
| DC [2] | 73.00 % | 65.62 % | 19.38 % | 0.1 s |
| TBD [27] | 52.45 % | 47.17 % | 14.13 % | 1 s |

Fig. 6 shows an output of our data association and pattern identity estimation procedures. We identify objects as $Tx$ where $x$ is the target's identity. Notice that in spite of objects *T25* and *T32* being occluded by object *T24*, their identities are still correctly estimated. Note also that the temporary ambiguity about the class of object *T24* during frames 124 to 130 is resolved a long time before the object leaves the field of view of the ego-vehicle.

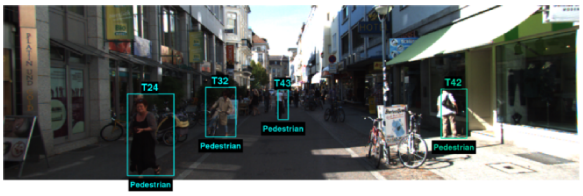
*1) Class Assignment:* Table II presents a confusion matrix evaluating the classification performance of the online *EA algorithm*. This table was built by assigning to each target the class label it took at the moment it left the camera's field of view. Values in the main diagonal represent instances of targets to which the correct class category was assigned. They showcase the descriptive power of our method, obtained by simply accumulating the innovation log-likelihoods under each of the models. Since a non-representative number of instances of the class *Cyclist* were present in the datasets, only



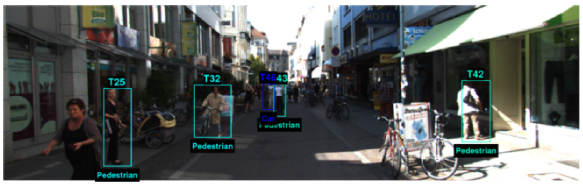

(a) Frame 123: object *T24* is misclassified as *Car*


(b) Frame 130: object *T24* is correctly classified.

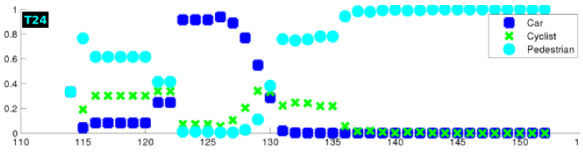

(c) Frame 141: object *T24* occludes object *T32*.




(d) Frame 146: Object *T32* is recovered and object *T24* occludes *25*.

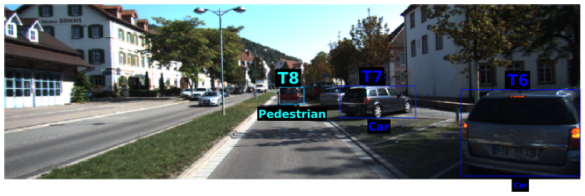



(e) Frame 152: Object *T25* is recovered.

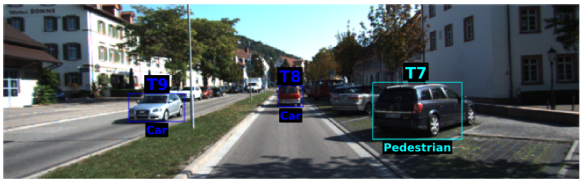



(f) Class assignment probabilities for object *T24*.

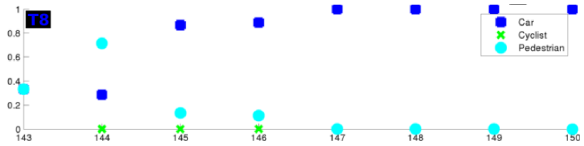
Fig. 6: Class identity of object *T24* is disambiguated after collecting evidence. Objects *T25* and *T32* are tracked and classified despite of being occluded.




(a) Object *T8* (middle of the image) at frame 144.




(b) Object *T8* at frame 150.




(c) Class assignment probabilities for object *T8*.

Fig. 7: A classification example. Notice the classification ambiguity about the class identity of object *T8*, which is resolved after frame 145 when further evidence about its dynamics is collected. See text for more details.

classification results for the classes *Car* and *Pedestrian* are reported.

Fig. 7 shows a scene with moving and static cars. In this case, the classification ambiguity about the class identity of object *T8* is resolved after frame 145 when further evidence about its dynamics is collected. Since only moving objects were used to learn the models, *T7* is misclassified as *Pedestrian*. Nevertheless, the object is correctly tracked. In general, misclassifications are mainly due to non-linearities in the motion of targets and unexpected inaccuracies in the navsolution of the ego-vehicle.

TABLE II: Confusion matrix with the classification results

| Act.\Pred. | Car | Pedestrian | Total |
|---|---|---|---|
| **Car** | 13 | 1 | 14 |
| **Pedestrian** | 0 | 53 | 53 |

## VI. Conclusions

We have presented a new framework for the problem of simultaneous tracking and classification with unknown data association. We proposed a model for representing the dynamics of multiple objects and a factorised inference procedure that allows us to estimate their state and class identity independently and efficiently. Our method fuses objects' appearance and dynamics to estimate the associations between tracked objects and observations. Furthermore, it is general enough so any sensor modality can potentially be used.

Since the moving objects in the test datasets have a short life span, we can safely assume that their motion can be modelled by LDS models. In more general scenarios, a single object switches between several motion modes, giving place to non-linear trajectories. Current work focuses on expanding the algorithm to incorporate switching LDS models.

## ACKNOWLEDGMENT

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

## REFERENCES

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

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1926–1933, June 2012.

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

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

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. 2006.

[6] C. Cadena and J. Kosecka. Recursive Inference for Prediction of Objects in Urban Environments. In *International Symposium on Robotics Research*, pages 1–16, 2013.

[7] W. Choi, C. Pantofaru, and S. Savarese. A general framework for tracking multiple people from a moving camera. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1577–91, July 2013.

[8] C. Dicle, O. I. Camps, and M. Sznaier. The Way They Move : Tracking Multiple Targets with Similar Appearance. In *IEEE International Conference on Computer Vision*, pages 2304–2311, 2013.

[9] M. R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[10] O. Frank, J. Nieto, J. Guivant, and S. Scheding. Multiple Target Tracking using Sequential Monte Carlo Methods and Statistical Data Association. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 00, pages 2718–2723, 2003.

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

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

[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, Aug. 2013.

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

[15] H. Kanazaki, T. Yairi, K. Machida, K. Kondo, and Y. Matsukawa. Variational Bayes Data Association Filter. *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 401–406, 2007.

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

[17] L. Lamard, J. P. Boyer, I. Pascal, and R. Sas. Multi target tracking with CPHD filter based on asynchronous sensors. In *International Conference on Information Fusion*, pages 892–898, 2013.

[18] Y. Li, C. Huang, and R. Nevatia. Learning to Associate : HybridBoosted Multi-Target Tracker for Crowded Scene. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2953–2960, 2009.

[19] R. Mahler and L. Martin. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4), 2007.

[20] D. Mitzel and B. Leibe. Taking Mobile Multi-object Tracking to the Next Level : People , Unknown Objects , and Carried Items. In *ECCV*, pages 566–579, 2012.

[21] F. Moosmann and C. Stiller. Joint Self-Localization and Tracking of Generic Objects in 3D Range Data. In *IEEE International Conference on Robotics and Automation*, pages 1138–1144, 2013.

[22] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and Inferring Motion Patterns using Parametric Segmental Switching Linear Dynamic Systems. *International Journal of Computer Vision*, 77(1-3):103–124, July 2007.

[23] H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *Journal of American Institute of Aeronautics and Astronautics*, 3(8):1445–1450, Aug. 1965.

[24] X. Rong Li. Joint tracking and classification based on bayes joint decision and estimation. *10th International Conference on Information Fusion*, pages 1–8, 2007.

[25] D. Vasquez, T. Fraichard, and C. Laugier. 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, Aug. 2009.

[26] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes. *IEEE transactions on pattern analysis and machine intelligence*, Aug. 2012.

[27] H. Zhang, A. Geiger, T. Mpi, and R. Urtasun. Understanding High-Level Semantics by Modeling Traffic Patterns. In *IEEE International Conference on Computer Vision*, pages 3056–3063, 2013.