# Model-Free Detection and Tracking of Dynamic Objects with 2D Lidar

Dominic Zeng Wang, Ingmar Posner and Paul Newman

**Abstract**

We present a new approach to detection and tracking of moving objects with a 2D laser scanner for autonomous driving applications. Objects are modelled with a set of rigidly attached sample points along their boundaries whose positions are initialised with and updated by raw laser measurements, thus allowing a nonparametric representation that is capable of representing objects independent of their classes and shapes. Detection and tracking of such object models are handled in a theoretically principled manner as a Bayes filter where the motion states and shape information of all objects are represented as a part of a joint state which includes in addition the pose of the sensor and geometry of the static part of the world. We derive the prediction and observation models for the evolution of the joint state, and describe how the knowledge of the static local background helps identifying dynamic objects from static ones in a principled and straightforward way. Dealing with raw laser points poses a significant challenge to data association. We propose a hierarchical approach, and present a new variant of the well-known Joint Compatibility Branch and Bound (JCBB) algorithm to respect and take advantage of the constraints of the problem introduced through correlations between observations. Finally, we calibrate the system systematically on real world data containing 7.5K labelled object examples and validate on 6K test cases. We demonstrate its performance over an existing industry standard targeted at the same problem domain as well as a classical approach to model-free object tracking.

## I. Introduction

For any mobile robotics system, safe navigation requires reliable perception of the environment. Moving hazards in particular pose significant challenges to the robot's operation. We present in this paper a principled framework for the detection and tracking of dynamic objects using 2D laser scanners that allows a flexible model-free object representation capable of representing objects of any classes and shapes. Our motivation is achieving safe navigation in an autonomous driving scenario in urban areas.

Laser scanners provide direct metric measurements of the environment, 2D scanners are particularly attractive to autonomous driving applications for its light-weight, efficient nature. However, since only a slice sample of the world is available at each scan, it is a challenge to extract critical semantic information from the clutter. This is an acute issue in the urban driving scenario where the measurements returned are often dominated by reflections from background clutter other than the operators (cars, people, cyclists etc.) on the road which pose the threats. In this paper, therefore, we turn to the motion cues and focus on the detection and tracking of *moving* objects independent of their classes and shapes.

The authors are with the Oxford University Mobile Robotics Group {dominic,ingmar,pnewman}@robots.ox.ac.uk


Fig. 1: The RobotCar autonomous driving platform quipped with a SICK LDMRS laser range finder (highlighted). See the online version for colour.

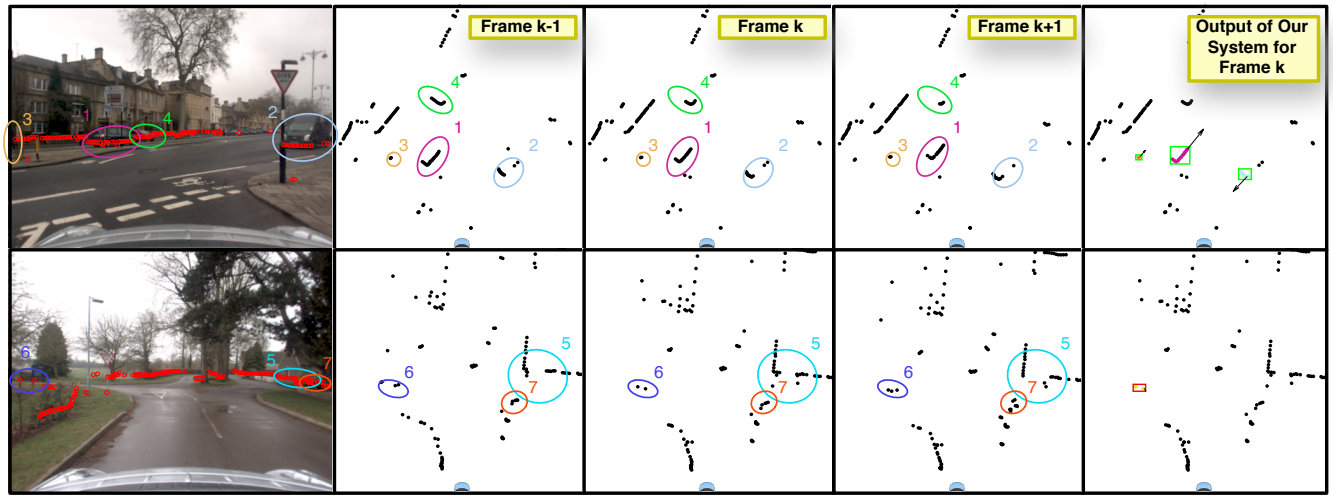



Fig. 2: Examples of real laser scans in an outdoor driving scenario. Each row gives an example at a single instant (Frame $k$). The first column shows the scan data at Frame $k$ reprojected into an image taken by an onboard camera. (The image is to provide some context for visualisation *only*. Our system does not require vision data to operate.) The next three columns show a sequence of consecutive raw laser scans at Frame $k-1$ (left), Frame $k$ (middle) and Frame $k+1$ (right) respectively. The last column shows the output from our proposed system at the instant at Frame $k$ (the middle scan). Here different objects detected are denoted by different colours and highlighted with boxes. A green box is a true detection whilst a red box is a false alarm. Arrows denote velocity estimates. See text for details. Colour version available online.

There are several complicating factors to the successful detection and tracking of moving objects with a 2D laser scanner mounted on a *moving* platform (such as the SICK LDMRS mounted on our research vehicle – a modified Nissan LEAF, Fig. 1). First of all, because the sensor itself is moving, we have uncertainties to its motion, rendering simple frame differencing techniques ineffective. In addition, occlusion and viewpoint changes give the appearance of dynamic behaviours even in a purely static scene. This confusion makes the reliable detection of the *true* dynamic objects difficult without giving a high false alarm rate.

These difficulties are illustrated with real-world data in Fig. 2. What is being shown here are samples of data gathered with our research platform (Fig. 1) as it was driven on the streets of Oxford. The first row gives a scenario of busy traffic, whilst the second row shows an example of laser data in an area where everything is stationary. In the first scenario (Row 1), it can be noted from the sequence of consecutive laser scans that, due to occlusion, it is ambiguous simply judging by

appearance to tell whether an object is moving or not. Here, cars numbered 1 and 2 are traveling at opposite directions, whilst Car Number 4 is stationary. Note how Car Number 4 changes its appearance as Car Number 1 gradually occludes it. Number 3 is a pedestrian walking on the pavement. Even in a situation where there is *no* moving objects in the scene (Row 2), there is still much burden on a dynamic object detector. Because our vehicle is driving through the environment, static structures are being observed from different viewpoints. This gives rise to dynamic-like behaviours in the data. The corner of the building numbered 5 appears to be dynamic, bushes such as the ones circled as 6 and 7 appear to be moving too. Note how Bush Number 7 resembles more in its appearance to a group of walking pedestrians than the *true* walking pedestrian numbered 3 in the first row. These motivating examples illustrate the difficulties in motion detection from a moving 2D scanner. It is challenging even for a human to tell what is moving and what is not from the raw data alone.

Many authors (for example, [Miyasaka et al., 2009] and [Wang et al., 2003]) observe that the problems of sensor pose estimation, map-building and detection and tracking of dynamic objects are closely related to each other. Removal of dynamic objects from the map-building process enhances the quality of the map, while knowledge about the static structure of the environment helps significantly in the successful detection of dynamic objects. Both are in turn tightly coupled with sensor pose estimation because all observations are made relative to the sensor. To this end, our proposed system also estimates a joint state that includes the sensor pose, a local static background that maps the static structure around the sensor and the dynamic states of the tracked moving objects through a Baye's filter. However, we emphasise it is not our interest to map the static part of the environment, only *local* static information is kept and estimated as a part of our state for the purpose of dynamic object detection. This is not a SLAM problem.

The last column of Fig. 2 shows the output of our proposed system to the central frame (the middle column). Despite of the aforementioned difficulties, our system successfully detects the two moving vehicles and the walking pedestrian and rejects ambiguous static structures as moving entities (with one false alarm for Bush Number 6 in the static scene example, the second row).

This paper is a more detailed account of the work previously presented in [Wang et al., 2013]. We structure the paper as follows. In Section II, we review existing approaches to detection and tracking of dynamic objects. Section III states our main contributions in the paper. Section IV presents the core concept of the paper that is our representation of objects. Following which we derive the prediction and observation models for the Bayes filter formulation in Section V. In Section VI, we discuss the challenge posed to data association by our object representation and our solution to it. Finally, in Section VII, we evaluate the performance of the proposed system with real-world data, and show that it outperforms both an industry standard solution that was designed for the same problem domain of object tracking from a moving sensor and a classical approach to model-free tracking based on independent tracking of scan segments. We conclude the paper in Section VIII and discuss about the insights gained and possible improvements for future work.

## II. Related Works

The problem of detection and tracking of multiple manoeuvring targets has been under active research for decades. Early efforts have been focused on the tracking of disjoint point-like targets, and it was soon realised that the challenge lies in

obtaining the correct association between noisy measurements and object tracks [Bar-Shalom et al., 2002].

In mobile robotics applications, however, further complications arise when moving targets are usually buried deep within significant background clutter and their measurement streams evolve in a complex fashion due to their constantly changing appearance. The fact that all observations are made relative to a *moving* sensor adds additional difficulty because static obstacles may also appear dynamic due to occlusion and noise (cf. Fig. 2). To tackle with these difficulties, existing literature on moving object detection and tracking with 2D laser scanners on a moving platform takes broadly two approaches.

The first approach can be summarised as the *model-free* approach, where detection is based on motion cues. The advantage of detecting dynamic obstacles based on motion is that no restriction is placed on the shape or class of the object and no semantic information is needed (i.e. objects are detected regardless whether it is a person or a car etc.). However, the drawback of this class of methods is that only *instantaneously* moving objects are detected, neglecting *potentially* moving objects. Our approach falls within this class of methods, and shares both its strengths and weaknesses.

Examples of model-free dynamic obstacle detection and tracking include systems deployed in the DARPA Urban Challenge [Leonard et al., 2008], [Mertz et al., 2013], which usually function by first segmenting measurements from multiple laser range finders, and then extracting geometric features from the segments. These geometric features are used to compile a list of object hypotheses, dynamic objects are then extracted as objects having a significant manoeuvring speed.

Most related to our work is a body of work that jointly estimates a static map of the environment alongside the detection and tracking of moving objects. Examples include Toyota's tracking system [Miyasaka et al., 2009] and Wang's system [Wang et al., 2003] (later extended by [Montesano et al., 2005] and [Vu et al., 2007]) that combines SLAM with dynamic object tracking. Both approaches take an occupancy grid representation of the environment, and use knowledge of occupancy probabilities from the map to propose likely moving object detections. Biswas et al. [Biswas et al., 2002] also take an occupancy grid representation, and detect non-stationary objects by map differencing. The individual objects are then identified with an EM algorithm. Wolf and Sukhatme [Wolf and Sukhatme, 2005] keep two occupancy grid maps, one for the static part and one for the dynamic part of the environment. However, the focus there is mapping in dynamic environments, dynamic objects are not tracked as separate entities. Yang and Wang [Yang and Wang, 2011] propose a system that jointly estimates the vehicle pose and moving object detections using a variant of RANSAC, and track merging and splits are handled via a decision tree based on spatiotemporal consistency tests. Works by Tipaldi et al. [Tipaldi and Ramos, 2009], [van de Ven et al., 2010] focus on the detection part of the problem, and formulate it under a joint Conditional Random Field (CRF) framework for solving both the data association and moving object detection problems. Finally, the work by Hahnel et al. [Hahnel et al., 2003] is also relevant where the authors formulate an EM algorithm to solve for a set of hidden indicator variables for each laser point to determine whether it is static or dynamic.

The second class of methods takes a *model-based* approach. Here the class of the objects to be detected is known a priori, and objects are first detected based on a parametric model of its shape and then tracked as separate entities. This class of methods has exactly the opposite advantages and drawbacks to the model-free methods. Since objects are detected based on their shape instead of motion, *potentially* moving hazards are also included, yet the class of objects detected is restricted to the specific object class described by the chosen parametric model.

Examples of this class of methods include the work by Granström [Granström, 2012], who detects and tracks rectangular and elliptical targets with a Probability Hypothesis Density (PHD) filter. The line of works by Arras et al. [Arras et al., 2007], [Arras et al., 2008] focus on people detection. They train a boosting classifier to detect legs of people and the detected legs are grouped into individual persons and tracked with a Multi-Hypothesis Tracker (MHT). The work by Schulz et al. [Schulz et al., 2001] is similar in that they also identify legs of people in 2D range scans and apply the Joint Probabilistic Data Association Filter (JPDAF) for robust tracking. However, in their work, only moving persons are considered. Topp and Christensen [Topp and Christensen, 2005] extend the work by removing this restriction, and extends the model for a person to also include people whose legs are not directly visible. Zhao and Thorpe [Zhao and Thorpe, 1998], on the other hand, restrict their attention to vehicles. In their work, an Interactive Multiple Model (IMM) filter is proposed for vehicle tracking that consists of three different motion models to cover a full range of motions for the tracked vehicles. Another example of model-based vehicle detection and tracking is the work presented by Vu and Aycard [Vu and Aycard, 2009], where a box model for vehicles is assumed, and detection and tracking are solved simultaneously by optimising over the best trajectories of the vehicles over a sliding window of laser scans using a Data-driven Markov Chain Monte Carlo (DDMCMC) algorithm.

## III. Contributions

Our main contribution in this paper is the formulation of a unified framework that jointly estimates the pose of the sensor, a continuously updated local static background, and the motion states of dynamic objects, with the focus on reliable detection of moving objects. All three aspects are tightly coupled through a novel joint state representation that allows for objects of arbitrary shapes and sizes to be modelled and tracked. Our state representation is presented in Section IV.

In addition, we propose a hierarchical data association algorithm to assign raw laser measurements to potential state updates, and present a variant of the Joint Compatibility Branch and Bound (JCBB) algorithm [Neira and Tardos, 2001] that is suitable for associating a large number of measurements, and derive an alternative set of recursive update rules based on the triangular form representation of positive definite matrices for its efficient and numerically stable computation. The details of our approach to data association can be found in Section VI.

## IV. An Unusual State Representation

The system we propose is run within a recursive Bayesian framework (implemented as a simple Extended Kalman Filter). In this section, we describe in detail the representation of the system state. In particular, we motivate and describe how dynamic objects are represented to allow objects of any class and any shape to be modelled and tracked.

The motions of dynamic objects can be arbitrary and independent of each other. The sensor, however, does not observe their motions directly but ranges and bearings of points on the surface of the objects. Thus once conditioned on the measurements, motions between different objects become correlated, due to the fact that these observations are taken from a moving sensor.

In order to correctly account for this correlation, the states of the objects and that of the sensor have to be estimated in a single joint distribution. A local static background is also simultaneously estimated as part of the joint state which is essential to distinguishing measurements belonging to dynamic objects from those on static objects. The state therefore consists of three

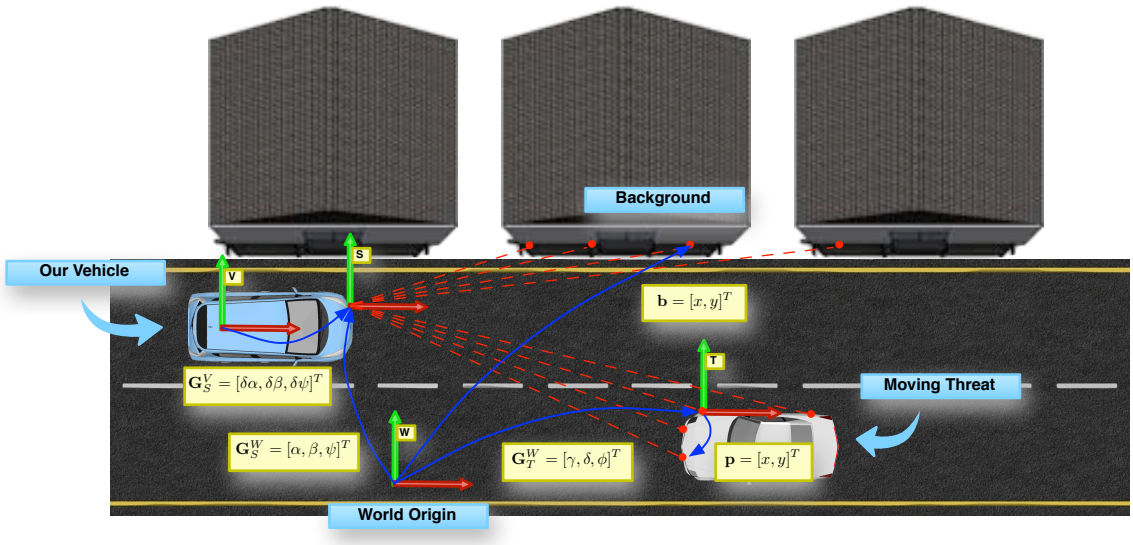



Fig. 3: Illustration of frame conventions and variable definitions. Here $\mathbf{G}_A^B$ denotes an SE2 transform *from* frame A *to* frame B as an $(x, y, \theta)$-triple. Coordinate frames $W$, $V$, $S$ and $T$ denote *world*, *vehicle*, *sensor* and *track* respectively. The sensor and objects' (tracks') motions are referenced to the fixed world origin W, and are denoted by the SE2 transforms $\mathbf{G}_S^W$ and $\mathbf{G}_T^W$. The sensor's extrinsic calibration parameters are denoted by the SE2 transform $\mathbf{G}_S^V$. Boundary points are referenced locally to the track's frame, and are denoted in the illustration by $\mathbf{p}$. Boundary points belonging to the *static* background are referenced globally to the world's frame, and are denoted in the illustration by $\mathbf{b}$. Subscripts are dropped to avoid clutter. See the online version of this article for colour.

parts: the sensor pose, the dynamic objects, and the static map. Fig. 3 illustrates the relationships between the different parts of the joint state, and our object representation, and will be referred to extensively in what follows.

### A. Sensor Pose Representation and Related Matters

The sensor pose (as part of the state vector $\mathbf{x}$) is represented by an SE2 transform $\mathbf{x}_S = \mathbf{G}_S^W = [\alpha, \beta, \psi]^T$ from the sensor's frame of reference to a stationary world frame of reference as depicted in Fig. 3, which is updated by vehicle odometry measurements at the prediction stage, and as part of the measurement update stage with each new laser scan observation.

Since the holonomic constraints apply to the vehicle but not to the sensor directly, and odometry measurements are naturally referenced to the vehicle's frame of reference, the transform between the sensor and vehicle's frames of reference are required. To account for uncertainties in this estimated transform, we include it as part of the state as $\mathbf{x}_C = \mathbf{G}_S^V = [\delta\alpha, \delta\beta, \delta\psi]^T$. This is the SE2 transform that transforms points from the sensor frame into the vehicle frame, also illustrated in Fig. 3.

### B. Model-Free Object Representation

For convenience of description, in what follows, we will also refer to dynamic objects as "tracks", since their motion state is continuously being tracked. Each dynamic object $i$ has its own set of axes $T_i$, and its motion state includes the SE2 transform $\mathbf{G}_{T_i}^W$ from the track's frame $T_i$ to the world frame $W$, and its derivative, i.e. $\mathbf{x}_T^i = [\mathbf{G}_{T_i}^{W\,T}, \dot{\mathbf{G}}_{T_i}^{W\,T}]^T = [\gamma_i, \delta_i, \phi_i, \dot{\gamma}_i, \dot{\delta}_i, \dot{\phi}_i]^T$. This is illustrated by Fig. 3 (the subscript $i$ is dropped to avoid clutter). What is unusual about our representation is however, that *none* of these states are directly observed according to the observation model. Instead, each object has *additional* state

parameters attached, named the "boundary point" coordinates, that are 2D cartesian coordinates represented *locally* to the object's frame of reference. It is these boundary points that are directly observed according to our observation model.

To understand the intuition behind boundary points, consider the case of a moving object being illuminated by the lidar for the first time, for example, in the case illustrated in Fig. 3. The set of raw range and bearing measurements $Z$ is used to initialise a new track with its 6-vector states *plus* boundary points at the locations of the raw measurements in $Z$ but transformed into the object's frame of reference (hence the name "boundary points" because the lidar impinges on the boundary of the object). All subsequent measurements (lidar illuminations) will be taken to be noisy observations of these boundary points on the object.

This model-free representation raises an interesting and central data association question. We must decide whether or not to extend the object's boundary by initialising additional boundary points with new raw lidar measurements or simply associate the laser returns to the existing boundary points as it stands. Furthermore, which of the laser returns belong to the static background and hence have nothing to do with dynamic objects whatsoever? Our approach to data association lies at the heart of this work and is detailed in Section VI.

We make the assumption that dynamic objects observed in the 2D scanning plane of the sensor behave as rigid bodies. This assumption, although it does not hold strictly true due to deformable bodies such as a walking pedestrian, is a close approximation when observations are constrained to the 2D plane. Under this assumption, boundary points stay fixed relative to the object's frame of reference and hence their states have a trivial motion model.

With the introduction of boundary points, each object is thus parameterised with a partial outline of its perimeter allowing objects of arbitrary shapes and dimensions to be modelled under the same representation.

### C. Static Background Representation

The representation for the static part of the state is simply a collection of boundary points as in the case of a dynamic object, except boundary points on the static background are represented with their global 2D cartesian coordinates in the *world*'s reference frame. See Fig. 3 for an illustration.

### D. The Complete State Structure

The complete state $\mathbf{x}$ consists of all parts described above, and is arranged as follows:

$$\mathbf{x} = [\mathbf{x}_S^T, \mathbf{x}_T^T, \mathbf{x}_b^T, \mathbf{x}_p^T, \mathbf{x}_C^T]^T \ , \tag{1}$$

where $\mathbf{x}_S$ is the sensor pose, $\mathbf{x}_T$ the collection of all 6-vector motion states of dynamic objects,

$$\mathbf{x}_T = [{\mathbf{x}_T^1}^T, {\mathbf{x}_T^2}^T, \ldots, {\mathbf{x}_T^{N_T}}^T]^T \ , \tag{2}$$

and $\mathbf{x}_b$ the collection of 2D coordinates of all boundary points belonging to the static part of the state,

$$\mathbf{x}_b = [\mathbf{b}_1^T, \mathbf{b}_2^T, \ldots, \mathbf{b}_{N_b}^T]^T \ , \tag{3}$$

**Algorithm 1** On New Measurement

1: **function** $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ PROCESSMEASUREMENT($\hat{\mathbf{x}}, \mathbf{P}, Z$)
2: **if** $Z$.type = Odometry **then**
3: $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ PROCESSODOMETRYMEASUREMENT($\hat{\mathbf{x}}, \mathbf{P}, Z$)
4: **else**
5: $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ PROCESSLASERMEASUREMENT($\hat{\mathbf{x}}, \mathbf{P}, Z$) ▷ Algorithm 2
6: **end if**
7: **end function**

**Algorithm 2** On New Laser Measurement

1: **function** $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ PROCESSLASERMEASUREMENT($\hat{\mathbf{x}}, \mathbf{P}, Z$)
2: $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ CLEANUPSTATES($\hat{\mathbf{x}}, \mathbf{P}$)
3: $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ FORWARDPREDICT($\hat{\mathbf{x}}, \mathbf{P}$)
4: $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ ASSOCIATEANDUPDATE($\hat{\mathbf{x}}, \mathbf{P}, Z$) ▷ Algorithm 3
5: $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ MERGETRACKS($\hat{\mathbf{x}}, \mathbf{P}$)
6: **end function**

and $\mathbf{x}_p$ the collection of 2D coordinates of all boundary points on all dynamic objects,

$$\mathbf{x}_p = [\mathbf{p}_1^{1T}, \mathbf{p}_2^{1T}, \ldots, \mathbf{p}_{N_p^1}^{1\ T}, \mathbf{p}_1^{2T}, \mathbf{p}_2^{2T}, \ldots, \mathbf{p}_{N_p^2}^{2\ T}, \ldots\ldots, \mathbf{p}_1^{N_T T}, \mathbf{p}_2^{N_T T}, \ldots, \mathbf{p}_{N_p^{N_T}}^{N_T\ T}]^T \ . \tag{4}$$

Finally, $\mathbf{x}_C$ is the vector of the extrinsic calibration parameters of the sensor as described in Section IV-A.

## V. DETECTION AND TRACKING OF DYNAMIC OBJECTS

The mean and covariance of the joint state vector are updated at each iteration according to the standard Bayes filter. The input to our system is a set of odometry measurements and a sequence of range and bearing laser scans. The inclusion of the odometry measurements is necessary because they provide the only *absolute* motion estimates. Without it, it is ambiguous to define what is *static* when all motion estimates are relative to the sensor. In this section, we derive the prediction and observation models for the joint state and present our approach to track initialisation which is key to the separation of dynamic objects from the static background.

Algorithm 1 lists the straightforward procedure carried out at each iteration when a new measurement is received. The mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$ of the joint state are updated differently according to the type of the measurement $Z$ (odometry or laser). In general, odometry measurements arrive at a much higher frequency than laser measurements, they need to be processed very efficiently, and therefore only forward-prediction of the sensor pose state taking the odometry measurement as a noisy control input is carried out in this case. In Section V-A we present the prediction model for this process. Algorithm 2 outlines the sequence of actions executed when a new laser scan is received. First, we do some house-keeping where out-of-date dynamic tracks and boundary points on the static background that have fallen out of the sensor's field of view are dropped. Next, the motion part of all dynamic tracks is forward-predicted according to an appropriate motion model as described in Section V-B, and followed by data association and measurement updates. We derive observation models in Section V-C and defer the discussion of data association until Section VI. Finally, any tracks appear to be static are merged with the map, and adjacent tracks following the same rigid body motion are merged into a single track. The latter is to account for the situation that occasionally a large object is tracked as different "pieces", and this allows for the pieces to be put back into a single

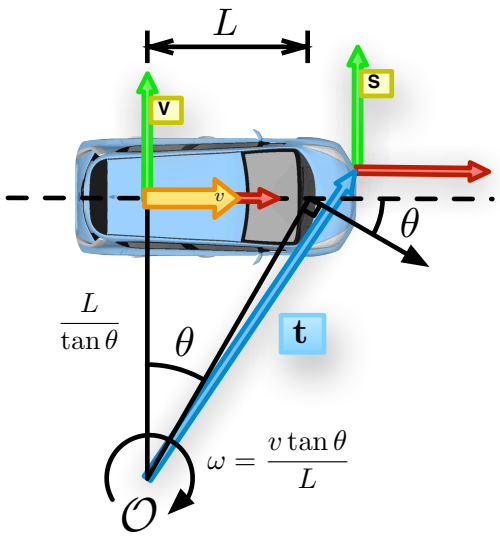



Fig. 4: Illustration of the standard bicycle model. Here $v$ denotes the vehicle's linear speed and $\theta$ the mean angle of the front wheels. The vehicle thus executes a coordinated turn about the point $\mathcal{O}$. Also shown are the vehicle's reference frame $V$ against which the holonomic constraints are specified, and the sensor's reference frame $S$ whose motion is to be predicted. See text for details. Colour version available online.

object. This merging procedure is described in Section V-D.

### *A. Sensor Pose Prediction on Odometry Measurement*

Each odometry measurement contains an estimate to the vehicle speed $v$, and the mean angle of the front wheels $\theta$ as illustrated in Fig. 4. Here we assume the non-holonomic vehicle motion follows a bicycle model. What is nonstandard about our prediction model, however, is that the holonomic constraints of the bicycle model are specified with respect to the *vehicle*'s reference point, whereas it is the motion of the *sensor* that is to be predicted (and is part of the state). We derive the prediction model for the sensor pose in what follows.

Assuming both $v$ and $\theta$ hold constant during the prediction period $\Delta t$, define the uncertain control input as

$$\mathbf{u}(v,\theta) = \begin{bmatrix} \Delta l \\ \Delta\psi \end{bmatrix} = \begin{bmatrix} v\Delta t \\ \omega\Delta t \end{bmatrix} = v\Delta t \begin{bmatrix} 1 \\ \frac{\tan\theta}{L} \end{bmatrix}, \tag{5}$$

where $L$ is the distance between the front and rear wheel axles (see Fig. 4), and $\omega$ the angular speed of the vehicle $\omega = \frac{v\tan\theta}{L}$. In this model, we assume $L$ is a fixed constant. After differentiating Equation 5, we obtain the Jacobian of the control input

$$\mathbf{U}(v,\theta) = \Delta t \begin{bmatrix} 1 & 0 \\ \frac{\tan\theta}{L} & \frac{v\sec^2\theta}{L} \end{bmatrix}. \tag{6}$$

Thus if the measured vehicle state $[\hat{v},\hat{\theta}]^T$ has covariance matrix $\mathbf{V}$,

$$\mathbf{V} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}, \tag{7}$$

the control input has mean $\hat{\mathbf{u}} = \mathbf{u}(\hat{v}, \hat{\theta})$ and covariance $\mathbf{Q} = \mathbf{U}\mathbf{V}\mathbf{U}^T$.

To arrive at a prediction model for the sensor pose $\mathbf{x}_S$, we note that, referring to Fig. 4, the sensor's motion is a simple rotation about the same stationary point $\mathcal{O}$ as that of the vehicle. This is due to the fact that the sensor is attached rigidly to the vehicle.

The linear velocity $\mathbf{v}_S$ of the sensor frame origin is more easily obtained in the *vehicle* frame, and transformed into the global world frame later. To do this, note that the vector $\mathbf{t}$ from $\mathcal{O}$ to the origin of $S$ in the *vehicle* frame $V$ is given by

$$\mathbf{t} = \begin{bmatrix} 0 \\ \frac{L}{\tan\theta} \end{bmatrix} + \begin{bmatrix} \delta\alpha \\ \delta\beta \end{bmatrix} . \tag{8}$$

Recall that $\mathbf{x}_C = \mathbf{G}_S^V = [\delta\alpha, \delta\beta, \delta\psi]^T$ is the SE2 transform from the sensor to the vehicle's frame. This gives the absolute velocity of the sensor frame origin (again, in the *vehicle* frame) as

$$\mathbf{v}_S = \omega\mathbf{R}(-\pi/2)\mathbf{t} = \omega \begin{bmatrix} \delta\beta \\ -\delta\alpha \end{bmatrix} + \begin{bmatrix} v \\ 0 \end{bmatrix} , \tag{9}$$

where $\mathbf{R}(\theta)$ is the 2D rotation matrix given an angle of rotation $\theta$. Transforming $\mathbf{v}_S$ to the world frame and assuming small motion within the time step $\Delta t$, the new pose of the sensor $\mathbf{x}'_S$ is given by

$$\mathbf{x}'_S = \mathbf{x}_S + \Delta t \begin{bmatrix} \mathbf{R}(\psi')\mathbf{v}_S \\ -\omega \end{bmatrix} , \tag{10}$$

where $\mathbf{x}_S = [\alpha, \beta, \psi]^T$ is the pose of the sensor before the prediction and $\psi' = \psi - \delta\psi$ is the angle of rotation from the *vehicle* frame to the world frame. Substituting in Equation 9 and rearranging, we arrive at the discrete dynamic model for the sensor pose:

$$\mathbf{x}'_S = \mathbf{f}(\mathbf{x}_S, \mathbf{x}_C, \mathbf{u}) = \begin{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \mathbf{R}(\psi - \delta\psi) \left( \Delta\psi \begin{bmatrix} \delta\beta \\ -\delta\alpha \end{bmatrix} + \begin{bmatrix} \Delta l \\ 0 \end{bmatrix} \right) \\ \psi - \Delta\psi \end{bmatrix} . \tag{11}$$

We can now differentiate Equation 11 to obtain its Jacobian as $\mathbf{J} = [\mathbf{F}\ \mathbf{G}] = [\mathbf{F}_S\ \mathbf{F}_C\ \mathbf{G}]$, where

$$\mathbf{F}_S = \begin{bmatrix} \mathbf{I}_2 & \mathbf{R}(\psi - \delta\psi) \left( \Delta\psi \begin{bmatrix} \delta\alpha \\ \delta\beta \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta l \end{bmatrix} \right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{12}$$

is the Jacobian with respect to $\mathbf{x}_S$, and

$$\mathbf{F}_C = \begin{bmatrix} \Delta\psi\mathbf{R}(\psi - \delta\psi) \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & -\mathbf{R}(\psi - \delta\psi) \left( \Delta\psi \begin{bmatrix} \delta\alpha \\ \delta\beta \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta l \end{bmatrix} \right) \\ \mathbf{0}_{1\times 2} & 0 \end{bmatrix} \tag{13}$$

is the Jacobian with respect to $\mathbf{x}_C$, and

$$\mathbf{G} = \begin{bmatrix} \mathbf{R}(\psi - \delta\psi) \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \mathbf{R}(\psi - \delta\psi) \begin{bmatrix} \delta\beta \\ -\delta\alpha \end{bmatrix} \\ 0 & -1 \end{bmatrix} \tag{14}$$

is the Jacobian with respect to $\mathbf{u}$ respectively, and $\mathbf{F}$ is defined by $\mathbf{F} = [\mathbf{F}_S \ \mathbf{F}_C]$.

Recall that $\hat{\mathbf{x}}$ and $\mathbf{P}$ denote the current mean and covariance of the joint state respectively, it follows that the updates for the complete joint state is then given by

$$\hat{\mathbf{x}}' = [\mathbf{f}^T(\hat{\mathbf{x}}_S, \hat{\mathbf{x}}_C, \hat{\mathbf{u}}), \hat{\mathbf{x}}_r^T, \hat{\mathbf{x}}_C^T]^T \tag{15}$$

for the mean, and

$$\mathbf{P}' = \begin{bmatrix} \mathbf{F}\mathbf{P}_{s,c|s,c}\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T & \mathbf{F}\mathbf{P}_{s,c|r,c} \\ \mathbf{P}_{r,c|s,c}\mathbf{F}^T & \mathbf{P}_{r,c|r,c} \end{bmatrix} \tag{16}$$

for the covariance matrix. Here a subscript of $r$ denotes the remaining states other than $\mathbf{x}_S$ and $\mathbf{x}_C$, and the notation $\mathbf{P}_{s,c|r,c}$ denotes the sub-matrix of $\mathbf{P}$ that is formed by taking the rows belonging to the states $\mathbf{x}_S$ and $\mathbf{x}_C$ and columns belonging to the states $\mathbf{x}_r$ and $\mathbf{x}_C$ and so on. Note that the bottom right block of the covariance matrix is not touched and the top left block involves multiplications of matrices of fixed sizes (at most $6 \times 6$). This computation is therefore dominated by the off-diagonal updates, which is of $O(N)$, thus can be carried out very efficiently.

### *B. Dynamic Object Motion Prediction*

At the prediction step after a new *laser* scan is received, all dynamic tracks are predicted forward according to a generic motion model before being updated with the measurements. A general motion model is desirable in this case because we do not have information regarding to the object class (and hence its expected motion pattern). In an autonomous driving scenario, the sensor itself is constantly moving, this usually results in any particular object instance being observed for only a limited amount of time, rendering object-specific behaviour learning approaches also impractical. Therefore, in this work, we choose the classic constant velocity model. However, in our definition of the constant velocity model, in addition to the conventional linear velocity components, the angular velocity component is also modelled (which also follows a constant *angular* velocity motion decoupled to the linear components). The inclusion of the angular velocity component makes the model adequate at capturing the full range of 2D rigid body motions, which is the best we could hope for in a purely model-free approach.

Our constant velocity model is a linear model given by

$$\mathbf{x}'_T = \mathbf{F}\mathbf{x}_T + \mathbf{v} \ , \tag{17}$$

where $\mathbf{x}_T$ is the dynamic states of the object before the prediction, and $\mathbf{x}'_T$ the predicted states,

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \Delta t\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \ , \tag{18}$$

and $\mathbf{v}$ is a zero mean additive noise with covariance

$$\mathbf{Q} = \begin{bmatrix} \frac{\Delta t^3}{3}\mathbf{V} & \frac{\Delta t^2}{2}\mathbf{V} \\ \frac{\Delta t^2}{2}\mathbf{V} & \Delta t\mathbf{V} \end{bmatrix} . \tag{19}$$

Here $\mathbf{V}$ is the $3 \times 3$ covariance matrix for the zero-mean continuous linear and angular white noise accelerations (see [Bar-Shalom et al., 2002, Chapter 6] for more details).

To extend Equation 17 to multiple tracks, define the matrices

$$\tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{F} & & & \\ & \mathbf{F} & & \\ & & \ddots & \\ & & & \mathbf{F} \end{bmatrix} \text{ and } \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & & & \\ & \mathbf{Q} & & \\ & & \ddots & \\ & & & \mathbf{Q} \end{bmatrix} , \tag{20}$$

where the diagonals have an entry for each dynamic track. The update equations for the joint state mean and covariance become:

$$\hat{\mathbf{x}}' = [\hat{\mathbf{x}}_S^T, (\tilde{\mathbf{F}}\hat{\mathbf{x}}_T)^T, \hat{\mathbf{x}}_r^T]^T , \tag{21}$$

and

$$\mathbf{P}' = \begin{bmatrix} \mathbf{P}_{s|s} & \mathbf{P}_{s|t}\tilde{\mathbf{F}}^T & \mathbf{P}_{s|r} \\ \tilde{\mathbf{F}}\mathbf{P}_{t|s} & \tilde{\mathbf{F}}\mathbf{P}_{t|t}\tilde{\mathbf{F}}^T + \tilde{\mathbf{Q}} & \tilde{\mathbf{F}}\mathbf{P}_{t|r} \\ \mathbf{P}_{r|s} & \mathbf{P}_{r|t}\tilde{\mathbf{F}}^T & \mathbf{P}_{r|r} \end{bmatrix} . \tag{22}$$

Here we follow the same subscripting convention as in Section V-A, and $\mathbf{x}_T$ denotes the collection of dynamic states of *all* dynamic tracks as in Equation 2, and $\mathbf{x}_r$ the remaining states other than $\mathbf{x}_S$ and $\mathbf{x}_T$.

### *C. Observation Models for Raw Laser Measurements*

In this section, we derive observation models for boundary points on the static background and dynamic objects respectively. All variables involved in what follows are defined in Section IV.

First we define the function and its Jacobian

$$\mathbf{u}(x, y) = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}\frac{y}{x} \end{bmatrix} , \ \mathbf{U}(x, y) = \frac{1}{r^2}\begin{bmatrix} rx & ry \\ -y & x \end{bmatrix} , \tag{23}$$

which converts a 2D point from cartesian coordinates into polar coordinates. This function (and its Jacobian) will be used extensively in what follows.

*1) Boundary Points of Static Background:* Each boundary point $j$ on the static background may potentially generate a laser measurement $\mathbf{z} = [r, \theta]^T$, and hence its measurement model is the boundary point's location in polar coordinates in the *sensor*'s frame of reference:

$$\mathbf{h}_j(\mathbf{x}) = \mathbf{u}(\mathbf{g}(\mathbf{x}_S, \mathbf{b}_j)) , \ \mathbf{g}(\mathbf{x}_S, \mathbf{b}_j) = \mathbf{R}^T(\psi)\left(\begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} \alpha \\ \beta \end{bmatrix}\right) . \tag{24}$$

Here $\mathbf{x}_S$ has been defined in Section IV-A, and $\mathbf{b}_j = [x_j, y_j]^T$ is the 2D cartesian coordinates of boundary point $j$ in the world frame as described in Section IV-C and illustrated by Fig. 3.

The Jacobians of $\mathbf{g}$ is given by:

$$\begin{aligned} \mathbf{G}_S(\mathbf{x}_S, \mathbf{b}_j) &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}_S} = \left[ \begin{array}{cc} -\mathbf{R}^T(\psi) & -\mathbf{R}\left(\frac{\pi}{2}\right)\mathbf{g}(\mathbf{x}_S, \mathbf{b}_j) \end{array} \right] , \\ \mathbf{G}_b(\mathbf{x}_S, \mathbf{b}_j) &= \frac{\partial \mathbf{g}}{\partial \mathbf{b}_j} = \mathbf{R}^T(\psi) . \end{aligned} \tag{25}$$

This leads to the overall Jacobians for the measurement model $\mathbf{h}_j$ as:

$$\begin{aligned} \mathbf{H}_S^j(\mathbf{x}) &= \frac{\partial \mathbf{h}_j}{\partial \mathbf{x}_S} = \mathbf{U}(\mathbf{g}(\mathbf{x}_S, \mathbf{b}_j))\mathbf{G}_S(\mathbf{x}_S, \mathbf{b}_j) , \\ \mathbf{H}_b^j(\mathbf{x}) &= \frac{\partial \mathbf{h}_j}{\partial \mathbf{b}_j} = \mathbf{U}(\mathbf{g}(\mathbf{x}_S, \mathbf{b}_j))\mathbf{G}_b(\mathbf{x}_S, \mathbf{b}_j) . \end{aligned} \tag{26}$$

*2) Boundary Points of Dynamic Objects:* Each boundary point $j$ on any dynamic track $i$ may also give rise to a laser measurement, and the measurement model in this case is the 2D polar coordinates of the boundary point in the *sensor*'s frame, and is given by:

$$\mathbf{h}_j^i(\mathbf{x}) = \mathbf{u}(\mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i)) , \ \mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) = \mathbf{R}^T(\psi) \left( \mathbf{R}(\phi_i) \begin{bmatrix} x_j^i \\ y_j^i \end{bmatrix} + \begin{bmatrix} \gamma_i \\ \delta_i \end{bmatrix} - \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right) , \tag{27}$$

where $\mathbf{x}_T^i$ is the dynamic states of track $i$, and $\mathbf{p}_j^i$ the boundary point's cartesian coordinates in the *object*'s frame as discussed in Section IV-B and illustrated by Fig. 3.

To obtain the Jacobians to the measurement model, we again obtain first the Jacobians of $\mathbf{g}$:

$$\begin{aligned} \mathbf{G}_S(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}_S} = \left[ \begin{array}{cc} -\mathbf{R}^T(\psi) & -\mathbf{R}\left(\frac{\pi}{2}\right)\mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) \end{array} \right] , \\ \mathbf{G}_T(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}_T^i} = \begin{bmatrix} \mathbf{R}^T(\psi) & \mathbf{R}^T(\psi)\mathbf{R}(\phi) \begin{bmatrix} -y_j^i \\ x_j^i \end{bmatrix} & \mathbf{0}_{2\times 3} \end{bmatrix} , \\ \mathbf{G}_p(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) &= \frac{\partial \mathbf{g}}{\partial \mathbf{p}_j^i} = \mathbf{R}^T(\psi)\mathbf{R}(\phi) . \end{aligned} \tag{28}$$

Now the Jacobians of $\mathbf{h}_j^i$ follow:

$$\begin{aligned} \mathbf{H}_S^{ij} &= \frac{\partial \mathbf{h}_j^i}{\partial \mathbf{x}_S} = \mathbf{U}(\mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i))\mathbf{G}_S(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) , \\ \mathbf{H}_T^{ij} &= \frac{\partial \mathbf{h}_j^i}{\partial \mathbf{x}_T^i} = \mathbf{U}(\mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i))\mathbf{G}_T(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) , \\ \mathbf{H}_p^{ij} &= \frac{\partial \mathbf{h}_j^i}{\partial \mathbf{p}_j^i} = \mathbf{U}(\mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i))\mathbf{G}_p(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) . \end{aligned} \tag{29}$$

### *D. Track Initialisation and Merging*

The initialisation of new dynamic tracks is non-trivial because we have to ensure that only new *dynamic* objects are initialised into new tracks and static objects are merged with the static background. To this purpose, we apply the technique of constrained initialisation [Williams, 2001], where each new track's motion status is deferred until it has accumulated enough evidence to

make the correct decision. Specifically, a new track is first marked as "tentative" when initialised, and becomes "mature" only if it is continuously being observed for more than a fixed number of frames (otherwise it is dropped). Then it is tested against the static background, and each existing dynamic track in turn for merging. The test and merging are all handled consistently within the same Bayesian filtering framework. If all merging tests fail, it is declared "established" and added to the set of existing dynamic tracks.

In the case of testing against merging with the static background, we are interested in the hypothesis that the track's absolute velocity (linear and angular) is zero given the estimated uncertainty on its motion. Following [Williams, 2001], we take uncertainty into account by introducing a fictitious *perfect* (noiseless) measurement on the track's absolute velocity, and test the validity of a measured value of zero with the standard $\chi^2$ test. Specifically, given a tentative track $T$ with its motion state vector $\mathbf{x}_T = [\gamma, \delta, \phi, \dot{\gamma}, \dot{\delta}, \dot{\phi}]^T$ (in accordance with the notation introduced in Section IV-B). We define a fictitious measurement model

$$\mathbf{h}_1(\mathbf{x}_T) = \begin{bmatrix} \dot{\gamma} \\ \dot{\delta} \\ \dot{\phi} \end{bmatrix} , \tag{30}$$

and consider a measured value of $\hat{\mathbf{z}} = \mathbf{0}$ under the noise-free condition ($\mathbf{R} = \mathbf{0}$). That is to say, given an observer that *perfectly* observes the internal dynamics of the object, what is the chance for it to say they are zero?

The Jacobian to Equation 30 is trivial and is given by

$$\mathbf{H}_T^1 = \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}_T} = [\mathbf{0}_3 \ \mathbf{I}_3] \ . \tag{31}$$

Thus noting that $\mathbf{R} = \mathbf{0}$, the innovation covariance of the fictitious measurement is given by simply $\mathbf{S} = \mathbf{P}_{v_T v_T}$, where $\mathbf{P}_{v_T v_T}$ is the sub-matrix of the joint covariance matrix $\mathbf{P}$ corresponding to the velocity of track $T$.

We then carry out a validation test on the measurement $\hat{\mathbf{z}} = \mathbf{0}$, to see if it falls within the validation gate, that is, if

$$(\hat{\mathbf{z}} - \mathbf{h}_1(\hat{\mathbf{x}}_T))^T \mathbf{S}^{-1} (\hat{\mathbf{z}} - \mathbf{h}_1(\hat{\mathbf{x}}_T)) \leq \chi^2_{d,\alpha} \ . \tag{32}$$

Here $\hat{\mathbf{x}}_T$ is the current estimate (the mean) of $\mathbf{x}_T$, and $\chi^2_{d,\alpha}$ is the $\chi^2$ validation gate threshold of degree of freedom $d$ ($d = 3$ in this case) and confidence level $\alpha$.

If Equation 32 holds, the hypothesis that this tentative track is stationary is accepted, and the merge proceeds with a formal update to the state estimate as if $\hat{\mathbf{z}}$ were a *real* measurement. This propagates the information gathered with the tentative track so far to the rest of the system and sets its absolute velocity actually to zero. The track can then be safely marginalised out after copying over its boundary points to the static background to complete the merge.

A similar procedure applies to merging tests with an existing dynamic track. In this case, the fictitious measurement applies to the relative motion of the tentative track to the existing track under consideration. Let us denote the dynamic states of the tentative and existing tracks by $\mathbf{x}_T = [\gamma_T, \delta_T, \phi_T, \dot{\gamma}_T, \dot{\delta}_T, \dot{\phi}_T]^T$ and $\mathbf{x}_E = [\gamma_E, \delta_E, \phi_E, \dot{\gamma}_E, \dot{\delta}_E, \dot{\phi}_E]^T$ respectively, then the fictitious measurement we are interested in is the relative motion (both linear and rotational) of the tentative track with respect

to the existing track:

$$\mathbf{h}_2(\mathbf{x}_T, \mathbf{x}_E) = \dot{\mathbf{G}}_T^E = \begin{bmatrix} \mathbf{R}(-\phi_E)\left(\begin{bmatrix} \dot{\gamma}_T \\ \dot{\delta}_T \end{bmatrix} - \begin{bmatrix} \dot{\gamma}_E \\ \dot{\delta}_E \end{bmatrix}\right) - \dot{\phi}_E \mathbf{R}(\tfrac{\pi}{2} - \phi_E)\left(\begin{bmatrix} \gamma_T \\ \delta_T \end{bmatrix} - \begin{bmatrix} \gamma_E \\ \delta_E \end{bmatrix}\right) \\ \dot{\phi}_T - \dot{\phi}_E \end{bmatrix}. \tag{33}$$

Differentiating Equation 33, we obtain its Jacobians

$$\mathbf{H}_T^2 = \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}_T} = \begin{bmatrix} -\dot{\phi}_E \mathbf{R}(\tfrac{\pi}{2} - \phi_E) & \mathbf{0}_{2\times 1} & \mathbf{R}(-\phi_E) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 1 \end{bmatrix}, \tag{34}$$

and

$$\mathbf{H}_E^2 = \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}_E} = \begin{bmatrix} \dot{\phi}_E \mathbf{R}(\tfrac{\pi}{2} - \phi_E) & \mathbf{D}\mathbf{h}_2(\mathbf{x}_T, \mathbf{x}_E) & -\mathbf{R}(-\phi_E) & \mathbf{R}(\tfrac{\pi}{2} - \phi_E)\left(\begin{bmatrix} \gamma_T \\ \delta_T \end{bmatrix} - \begin{bmatrix} \gamma_E \\ \delta_E \end{bmatrix}\right) \\ \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & -1 \end{bmatrix}, \tag{35}$$

where $\mathbf{D}$ is a selection matrix given by

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}. \tag{36}$$

The merging test then proceeds in a similar fashion to the merging test in the static case.

The same merging procedure is also conducted at the end of each processing cycle (Algorithm 2) for testing each existing track against merging with the static background or other existing tracks.

The procedure described in this section is important because it is the *only* means through which the static information in the state is kept up-to-date. It also allows a dynamic object (or a false detection) to transition into the static state. An up-to-date static information is the *only* direct influential factor to successful dynamic object *detection* because successful data association requires an accurate estimate of static boundary points and the classification of each new laser scan into a static part and a dynamic part is handled implicitly by the data association process. Data association is the main subject of our discussion in the next section.

## VI. Hierarchical Data Association

Not all state variables in the joint state are directly observable, for the ones that are, namely boundary points on either the static background or any dynamic object, it is ambiguous which is being observed, which is not, and indeed, whether a new boundary point needs to be initialised. Thus when new laser measurements arrive, it has to be determined for each measurement:

1) If it is an observation on a static object, then
    a) is it an observation of an existing boundary point?
    b) is it an observation of a new boundary point?
2) If it is an observation on an existing dynamic object, then

a) is it an observation of an existing boundary point on the object?

b) is it an observation of a new boundary point on the object?

3) Is it an observation on a new dynamic object?

In addition, in case 2, it has also to be determined to which of the existing dynamic objects the measurement belongs to, and in case 3, how many new tracks need to be initialised.

This data association problem naturally breaks down into two levels. The first level operates at the coarse scale, in which measurements are first divided into clusters, and each cluster is assigned to either the static background, or a dynamic object, or used to initialise a new dynamic track. At the fine level, for each object (or the static background), measurements from the associated clusters are further associated with its existing boundary points or used to initialise new boundary points.

### *A. Coarse Level Data Association*

The measurements in a given laser scan are first divided into a set of clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_{|\mathcal{C}|}\}$. The clusters are then assigned to the static background and dynamic objects recursively with the help of the ICP [Besl and McKay, 1992] algorithm.

We use a simple variant of ICP with outlier rejection. Specifically, to align two point sets $P$ and $Q$, at each iteration, we conduct nearest neighbour search between the two point sets. A point in $P$ is associated to its nearest neighbour in $Q$ if their distance is within a certain threshold, otherwise it is discarded as an outlier for this iteration and become unassociated to any point in $Q$. All *associations* obtained in this way are used to estimate a transform that aligns the point set $P$ to $Q$. The points in $P$ are then updated to their new positions with the estimated transform and the loop continues until convergence. The association upon convergence is taken as the final association, *with outlier rejection* from $P$ to $Q$.

With the ICP procedure defined, the coarse level data association proceeds as follows. First, boundary points on the static background are aligned to the set of measurements $Z$ with ICP, and clusters in $\mathcal{C}$ which contains measurements matched to any boundary points on the static background in this way are associated to the static background, and used to update or initialise new boundary points at the fine level for the static backgournd. Then the associated clusters are removed from $\mathcal{C}$ and a similar procedure follows recursively for each dynamic track. The clusters that remain in $\mathcal{C}$ at the end of this process are thus not associated with any existing track (or the static background), and each cluster will initialise a new tentative dynamic track. This procedure is captured in Algorithm 3.

Coarse level data association makes intuitive sense because first the majority of the measurements belonging to the static part of the environment will be associated with the static boundary points, leaving the outliers being mostly measurements on dynamic objects. The association of clusters instead of raw measurements at this level helps in extending object boundaries because according to Algorithm 3, all measurements in a given cluster will be associated to an object (or the static background) if *any* measurement in it is associated with a boundary point on the object with ICP. Provided the initial clustering is correct and ICP with outlier rejection gives a reasonable performance, the remaining measurements in the cluster can safely be assumed to be previously unobserved boundary points on the same object and be used to extend the object boundary.

ICP is known to perform poorly when the initial misalignment between the two point sets is large. However, if the two point sets start close to aligned, ICP is ideal. This is precisely the case here because data association is conducted *after* model

**Algorithm 3** Coarse Level Data Association

**function** $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ ASSOCIATEANDUPDATE$(\hat{\mathbf{x}}, \mathbf{P}, Z)$
  $\mathcal{C} \leftarrow$ CLUSTERMEASUREMENTS$(Z)$
  $(\hat{\mathbf{x}}, \mathbf{P}, \mathcal{A}) \leftarrow$ ASSOCIATEANDUPDATEWITHSTATIC$(\hat{\mathbf{x}}, \mathbf{P}, \mathcal{C})$ ▷ $\mathcal{A} = \{C \in \mathcal{C} : C \text{ is associated}\}$
  $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{A}$
  **for** $i = 1, 2, \ldots, N_T$ **do**
    $(\hat{\mathbf{x}}, \mathbf{P}, \mathcal{A}) \leftarrow$ ASSOCIATEANDUPDATEWITHDYNAMIC$(\hat{\mathbf{x}}, \mathbf{P}, \mathcal{C}, i)$
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{A}$
  **end for**
  **for all** $C \in \mathcal{C}$ **do**
    $(\hat{\mathbf{x}}, \mathbf{P}) \leftarrow$ INITIALISENEWTRACK$(\hat{\mathbf{x}}, \mathbf{P}, C)$
  **end for**
**end function**

prediction. The boundary points will be in their *predicted* locations instead of their previous locations in the last frame. Given a good motion estimate, the predicted locations of the boundary points will be already close to their actually observed values.

What remains is to ensure that a good set of clusters is produced in the clustering step. The details of our approach to clustering are deferred until Section VI-E where we introduce the proposed EMST-EGBIS clustering algorithm that is designed to produce perceptually coherent clusters.

However, segmentation failure is inevitable in any unsupervised clustering procedure. In such an event, other components of the system such as the merging procedure introduced in Section V-D take over to resolve the issue if it is possible.

### *B. Fine Level Data Association*

Given a set of clusters associated with a certain track (or the static background), the fine level data association must find a matching satisfying certain desirable criteria that assigns measurements contained in the clusters to boundary points on the track. Correct assignment is critical to successful tracking, and, the stability of the system as a whole, due to the fact that correlation is introduced between all pairs of variables in the joint state. In particular, all state variables we would like to infer: the sensor pose, the dynamic states of the tracked objects, are not directly observed.

Joint Compatibility Branch and Bound (JCBB) [Neira and Tardos, 2001] is a well-known data association algorithm that takes into account the correlations between observations. Explained in our nomenclature, an association between the set of measurements and the set of boundary points is called a *feasible* association if:

1) Each measurement is associated to at most one boundary point, and no two measurements are associated to the same boundary point (one-one association).
2) Each matching of a measurement to a boundary point is individually compatible as described below (individual compatibility).
3) The overall data association is jointly compatible as described below (joint compatibility).

To clarify the concepts of individual and joint compatibilities, consider a boundary point whose observation model has the standard form $\mathbf{z}_j = \mathbf{h}_j(\mathbf{x}) + \mathbf{w}_j$. Here $\mathbf{x}$ is the joint state defined in Section IV, and $\mathbf{w}_j$ is the additive zero-mean measurement noise. Thus its innovation covariance matrix is $\mathbf{S}_j = \mathbf{H}_j \mathbf{P} \mathbf{H}_j^T + \mathbf{R}$. Here $\mathbf{H}_j$ is the Jacobian of the function $\mathbf{h}_j$ evaluated at

the current state mean, and $\mathbf{R}$ is the measurement noise covariance matrix (we assume all measurements have the same noise covariance matrix).

*1) Individual Compatibility:* Individual compatibility requires the assigned measurement $\hat{\mathbf{z}}_i$ must fall within a certain confidence region of boundary point $j$'s validation gate, i.e. an assignment of $\hat{\mathbf{z}}_i$ to $\mathbf{z}_j$ is individually compatible if:

$$(\hat{\mathbf{z}}_i - \mathbf{h}_j(\hat{\mathbf{x}}))^T \mathbf{S}_j^{-1} (\hat{\mathbf{z}}_i - \mathbf{h}_j(\hat{\mathbf{x}})) \leq \chi^2_{d,\alpha} \ , \tag{37}$$

where $\chi^2_{d,\alpha}$ is the $\chi^2$ validation gate threshold of degree of freedom $d$ and confidence level $\alpha$. Here, $d$ is the measurement dimension, hence $d = 2$, because each measurement contains a range and a bearing $\hat{\mathbf{z}} = [\hat{r}, \hat{\theta}]^T$.

*2) Joint Compatibility:* Under the assumption of independent observations, the joint observation model of a complete association $\sigma$ is given by

$$\mathbf{h}_\sigma(\mathbf{x}) = [\mathbf{h}^T_{\sigma(1)}(\mathbf{x}), \mathbf{h}^T_{\sigma(2)}(\mathbf{x}), \ldots]^T \ , \tag{38}$$

with the innovation covariance

$$\mathbf{S}_\sigma = \begin{bmatrix} \mathbf{H}_{\sigma(1)}\mathbf{P}{\mathbf{H}_{\sigma(1)}}^T + \mathbf{R} & \mathbf{H}_{\sigma(1)}\mathbf{P}{\mathbf{H}_{\sigma(2)}}^T & \cdots \\ \mathbf{H}_{\sigma(2)}\mathbf{P}{\mathbf{H}_{\sigma(1)}}^T & \mathbf{H}_{\sigma(2)}\mathbf{P}{\mathbf{H}_{\sigma(2)}}^T + \mathbf{R} & \\ \vdots & & \ddots \end{bmatrix} \ , \tag{39}$$

where $\sigma(1)$ denotes the index of the boundary point associated to the first *assigned* measurement and so on. Thus the joint measurement has dimension $N_a d$ if the number of assigned measurements is $N_a$. An association $\sigma$ is jointly compatible if

$$(\hat{\mathbf{z}}_\sigma - \mathbf{h}_\sigma(\hat{\mathbf{x}}))^T \mathbf{S}_\sigma^{-1} (\hat{\mathbf{z}}_\sigma - \mathbf{h}_\sigma(\hat{\mathbf{x}})) \leq \chi^2_{N_a d,\alpha} \ . \tag{40}$$

Here $\hat{\mathbf{z}}_\sigma$ is the collection of the measurements that are *assigned* to some boundary point according to association $\sigma$.

The JCBB algorithm then finds the *feasible* association that has the largest number of assigned measurements $N_a^*$. Since there are in general many feasible associations with $N_a = N_a^*$, the algorithm finds the association $\sigma^*$ that gives the lowest joint Normalised Innovation Squared (jNIS, defined to be the expression to the left of the inequality in Equation 40).

### *C. The JCBB-Refine Algorithm*

Unfortunately, the JCBB algorithm is an exponential algorithm in the number of measurements to be assigned. This means it is not directly applicable to our application domain, since in our case observations are raw laser measurements and number in the 100's.

We introduce the JCBB-Refine algorithm, which instead of aiming to find the optimum assignment $\sigma^*$, we only find a *good* association $\tilde{\sigma}$ that is *feasible*. Of course, there are many *feasible* associations, a *good* association must be measured relative to some gauge. The JCBB-Refine algorithm we propose here takes an initial association $\sigma_0$ as a starting point, and finds a feasible association that has as many assigned measurements and as low a jNIS as possible in a greedy manner while respecting the initial association $\sigma_0$. The initial association $\sigma_0$ can be arbitrary, i.e. it does *not* have to be feasible. In fact, none of the feasibility conditions has to be satisfied.

Given $\sigma_0$, the algorithm first removes assignments that do not comply with individual compatibility (i.e. noncompliant measurements become unassociated with any boundary point), and then removes duplicate assignments with a single pass through the measurements. After these, the resulting association satisfies feasibility conditions 1 and 2. The algorithm then proceeds to iteratively removing the assignment that leads to the most jNIS reduction until condition 3 is satisfied. Starting from this minimal set of assignments that is now feasible, the unassociated measurements are then tried in turn, and assigned to the boundary point (among the boundary points that are individually compatible and yet unassigned) that gives the lowest jNIS if the assignment does not violate joint compatibility. The resulting association is thus guaranteed to remain feasible.

The JCBB-Refine algorithm can be initialised with any sensible starting assignment $\sigma_0$. In our particular application, the assignment as a result of the ICP matching at the coarse level association is a natural starting point. The association after the refinement is then used to update the joint state with the associated measurements, and all unassociated measurements initialise new boundary points to extend the object boundary.

### *D. Recursive Updates in Triangular Form*

It is shown [Neira and Tardos, 2001] that the innovation covariance matrix $\mathbf{S}$, its inverse, and the jNIS can be computed recursively as hypotheses are being tested. However in its direct form, the recursion suffers from numerical stability issues when the number of measurements becomes large because both $\mathbf{S}$ and $\mathbf{S}^{-1}$ have to be maintained to be symmetric and positive definite. We show the same computation can be achieved in the triangular form, which is a numerically stable representation for positive definite matrices.

To begin with, at step $k$, assume a decomposition for $\mathbf{S}_k$ is given such that $\mathbf{S}_k = \mathbf{U}_k^T\mathbf{U}_k$ for some upper triangular matrix $\mathbf{U}_k$, for example through Cholesky decomposition. Then its inverse is given by $\mathbf{S}_k^{-1} = \mathbf{U}_k^{-1}(\mathbf{U}_k^{-1})^T$. If we define a new matrix $\mathbf{G}_k = \mathbf{U}_k^{-1}$ so that $\mathbf{S}_k^{-1} = \mathbf{G}_k\mathbf{G}_k^T$, we obtain a decomposition also for $\mathbf{S}_k^{-1}$. In particular, $\mathbf{G}_k$ is also upper triangular.

Now the next iteration selects a new boundary point to be assigned to a measurement expanding the innovation covariance matrix (with reference to Equation 39) to

$$\mathbf{S}_{k+1} = \begin{bmatrix} \mathbf{S}_k & \mathbf{W}_k^T \\ \mathbf{W}_k & \mathbf{N}_k \end{bmatrix} . \tag{41}$$

If we now define a new upper triangular matrix

$$\mathbf{U}_{k+1} = \begin{bmatrix} \mathbf{U}_k & \mathbf{R}_k^T \\ \mathbf{0} & \mathbf{F}_k \end{bmatrix} , \tag{42}$$

where $\mathbf{R}_k = \mathbf{W}_k\mathbf{G}_k$ and $\mathbf{F}_k = \text{chol}(\mathbf{N}_k - \mathbf{R}_k\mathbf{R}_k^T)$. It is then straightforward to verify that $\mathbf{S}_{k+1} = \mathbf{U}_{k+1}^T\mathbf{U}_{k+1}$ by direct evaluation.

Equation 42 establishes a recursion on the upper triangular matrix $\mathbf{U}_k$. A similar recursion on $\mathbf{G}_k$ can be obtained by explicitly inverting Equation 42. Note that Equation 42 expresses $\mathbf{U}_{k+1}$ in block form, hence we can apply the standard

formula for matrix inversion in block form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix} . \tag{43}$$

This leads to

$$\mathbf{G}_{k+1} = \mathbf{U}_{k+1}^{-1} = \begin{bmatrix} \mathbf{U}_k & \mathbf{R}_k^T \\ \mathbf{0} & \mathbf{F}_k \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{U}_k^{-1} & -\mathbf{U}_k^{-1}\mathbf{R}_k^T\mathbf{F}_k^{-1} \\ \mathbf{0} & \mathbf{F}_k^{-1} \end{bmatrix} . \tag{44}$$

Noting $\mathbf{G}_k = \mathbf{U}_k^{-1}$ and define a new matrix $\mathbf{M}_k = \mathbf{F}_k^{-1}$ for notational convenience, we obtain the result

$$\mathbf{G}_{k+1} = \begin{bmatrix} \mathbf{G}_k & -\mathbf{G}_k\mathbf{R}_k^T\mathbf{M}_k \\ \mathbf{0} & \mathbf{M}_k \end{bmatrix} . \tag{45}$$

This gives a recursion on the decomposition of the inverse of the covariance matrix $\mathbf{S}_k^{-1}$.

At any stage of the recursion, if the innovation covariance matrix $\mathbf{S}_k$ or its inverse is needed, it can be obtained by a straightforward evaluation $\mathbf{S}_k = \mathbf{U}_k^T\mathbf{U}_k$ or $\mathbf{S}_k^{-1} = \mathbf{G}_k\mathbf{G}_k^T$. Thus by maintaining a different recursion on $\mathbf{U}_k$ and $\mathbf{G}_k$, we keep an *implicit* representation for the innovation covariance matrix and its inverse. This triangular form does not suffer from numerical stability issues as keeping an explicit recursion for $\mathbf{S}_k$ and $\mathbf{S}_k^{-1}$ does because $\mathbf{U}_k$ and $\mathbf{G}_k$ are maintained to be upper triangular by definition and this automatically guarantees $\mathbf{S}_k$ and $\mathbf{S}_k^{-1}$ to be symmetric and positive definite.

The innovation $\boldsymbol{\nu}_k$ also needs to be expanded with the newly assigned measurement:

$$\boldsymbol{\nu}_{k+1} = \begin{bmatrix} \boldsymbol{\nu}_k \\ \tilde{\boldsymbol{\nu}}_k \end{bmatrix} , \tilde{\boldsymbol{\nu}}_k = \hat{\mathbf{z}}_k - \mathbf{h}_k(\hat{\mathbf{x}}) , \tag{46}$$

where $\hat{\mathbf{z}}_k$ is the newly assigned measurement, and $\mathbf{h}_k(\hat{\mathbf{x}})$ the predicted measurement given by the measurement model of the associated boundary point (as presented in Section V-C).

Given our representation of the innovation covariance matrix and its inverse in the triangular form, a simpler formula for the jNIS can be obtained if we keep track of an alternative vector $\boldsymbol{\xi}_k = \mathbf{G}_k^T\boldsymbol{\nu}_k$ instead of $\boldsymbol{\nu}_k$. The recursion for $\boldsymbol{\xi}_k$ can be established by substituting Equations 45 and 46 into $\boldsymbol{\xi}_{k+1} = \mathbf{G}_{k+1}^T\boldsymbol{\nu}_{k+1}$, so we have

$$\xi_{k+1} = \begin{bmatrix} \mathbf{G}_k & -\mathbf{G}_k\mathbf{R}_k^T\mathbf{M}_k \\ \mathbf{0} & \mathbf{M}_k \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\nu}_k \\ \tilde{\boldsymbol{\nu}}_k \end{bmatrix} = \begin{bmatrix} \mathbf{G}_k^T\boldsymbol{\nu}_k \\ \mathbf{M}_k^T(\tilde{\boldsymbol{\nu}}_k - \mathbf{R}_k\mathbf{G}_k^T\boldsymbol{\nu}_k) \end{bmatrix} . \tag{47}$$

Noting $\boldsymbol{\xi}_k = \mathbf{G}_k^T\boldsymbol{\nu}_k$ and define $\boldsymbol{\mu}_k = \mathbf{M}_k^T(\tilde{\boldsymbol{\nu}}_k - \mathbf{R}_k\mathbf{G}_k^T\boldsymbol{\nu}_k) = \mathbf{M}_k^T(\tilde{\boldsymbol{\nu}}_k - \mathbf{R}_k\boldsymbol{\xi}_k)$ we arrive at

$$\xi_{k+1} = \begin{bmatrix} \boldsymbol{\xi}_k \\ \boldsymbol{\mu}_k \end{bmatrix} . \tag{48}$$

Now with $\boldsymbol{\xi}_k$ defined, the jNIS at each iteration $k$ has a remarkably simple form

$$\text{jNIS}_k = \boldsymbol{\nu}_k^T\mathbf{S}_k^{-1}\boldsymbol{\nu}_k = \boldsymbol{\nu}_k^T\mathbf{G}_k\mathbf{G}_k^T\boldsymbol{\nu}_k = \boldsymbol{\xi}_k^T\boldsymbol{\xi}_k , \tag{49}$$

a recursion for the jNIS therefore follows naturally from the recursion for $\boldsymbol{\xi}_k$ (Equation 48):

$$\text{jNIS}_{k+1} = \text{jNIS}_k + \boldsymbol{\mu}_k^T \boldsymbol{\mu}_k \ . \tag{50}$$

Finally, we note that these recursive update rules based on an *implicit* triangular form representation of the innovation covariance matrix and its inverse have the same computational complexity as the explicit recursions introduced in [Neira and Tardos, 2001], but are more numerically stable.

### *E. EMST-EGBIS Clustering*

Euclidean Minimum Spanning Tree (EMST) based clustering algorithms have a long history [Zahn, 1971], [Asano et al., 1988], [Wang et al., 2012], and is known for being capable of detecting clusters with irregular boundaries [Zahn, 1971]. We propose a novel variant of EMST-based clustering algorithms in this section, which is efficient, and particularly suitable for 2D range-bearing measurements where point densities vary significantly at different distances from the sensor and across different scene geometries.

EGBIS [Felzenszwalb and Huttenlocher, 2004] is a popular graph-based segmentation algorithm, that is effective at producing perceptually coherent segments over a wide range of variation in the dissimilarity measure across the global graph structure. However, the graph structures over unstructured application domains such as laser point clouds are usually not straightforward to define.

Our EMST-EGBIS algorithm combines the strengths of both algorithms. Specifically, we first compute the EMST over the collection of points, and take the obtained EMST as the input graph structure to EGBIS to compute the clusters. Edge weights of the EMST (Euclidean distances between points) are taken directly as the dissimilarity measure. We apply EMST-EGBIS to each incoming laser scan to obtain measurement clusters to associate to object tracks (and the static background) at the coarse level of data association (cf. Section VI-A).

## VII. System Evaluation

In this section, we evaluate the proposed system, both quantitatively and qualitatively, and compare its performance against two benchmarking object tracking approaches, of which one is an industrial standard solution. We note there exists a large body of work on similar application domains (for example, [Miyasaka et al., 2009], [Mertz et al., 2013], [Wang et al., 2003]), however it is often difficult to obtain a fair quantitative comparison to the methods due to either a lack of quantitative results or difficulty of a direct comparison using a common dataset. This motivates the comparison to a commercial product – the only one we are aware of which identifies and tracks dynamic obstacles. In addition, we also place the proposed system in comparison to a classical tracking solution where each object is tracked with independent Kalman filters with greedy data association (details to follow in the next section).

### *A. Experiment Setup*

Our experiment platform is a modified Nissan Leaf that is equipped with a SICK LDMRS laser scanner, which is a scanner targeted at object tracking applications on automotive platforms. It scans the environment in four vertically separated scanning

planes at 12.5Hz and produces native object tracking information at the same time. Odometry information is provided internally as part of the vehicle state at 100Hz. Fig. 1 shows our experiment platform together with the sensor setup.

The main benefit of the multi-layered architecture of the SICK LDMRS scanner is that it can compensate for the vehicle pitch appropriately. To take advantage of the multi-layered scanner, we follow a standard procedure to remove ground strike measurements given multiple scanning layers described in [Leonard et al., 2008]. Specifically, a grid is first built over the 2D scanning plane, measurements from each scanning layer are projected down to the scanning plane and associated with the grid cell that the measurement falls into. Then each occupied grid cell is taken in turn. Where a cell contains measurements from more than one scanning layers the measurements contained in that cell will be retained, otherwise they are discarded. Finally, a single 2D polar scan free of ground strikes is generated by taking all measurements that are retained, collecting range measurements corresponding to each discrete scan angle from this retained set, and generate a new range value for the scan angle by averaging. The resulting 2D scan has a number of measurements in the same order as any single scan layer. The intuition behind this technique is that obstacles are assumed to extrude out from the ground thus measurements from multiple layers when projected down to the scanning plane tend to be close. On the other hand, a ground strike due to vehicle pitching does not exhibit this behaviour because it is equivalent to measuring a ramp. In all that follows, we conduct experiments using this synthesised scan from all scanning layers.

We note that this procedure of combining the multiple layers of the LDMRS scanner is solely for the purpose of removing false measurements due to ground strikes. Our proposed tracking framework is not dependent on a multi-layered laser scanner. The framework applies to any 2D laser scan. In fact, from our experience, in the case of few ground strikes, using only a single layer from the LDMRS scanner produces similar tracking performance.

In addition to the LDMRS's native detection system, we compare the proposed system with a classic model-free tracking approach. For this second baseline, the laser scan is first clustered with the EMST-EGBIS algorithm described in Section VI-E. Then each cluster centroid is tracked with an independent Kalman filter under the constant velocity model. Data association in this case is done by greedily assigning each observed cluster centroid to the first object track for which the observation falls within its validation gate. A new object track is initialised if a cluster cannot be associated with any existing object track in this way.

To evaluate the proposed system against the baselines, both quantitatively and qualitatively, we collected data of busy traffic at the centre of Oxford containing a variety of dynamic objects including pedestrians, cars, bicyclists, buses, trucks, motorcycles and so on, and extracted two busy sections of the log right at the centre of the city for evaluation. One dataset is used to find the best-performing parameter set, and is hence named the *training* set, and the other, the test set, is used to obtain unbiased test results running under the trained parameter set for fair comparison. Both datasets are hand-labelled to provide ground truth for quantitative evaluation. Fig. 5 presents sample frames from the training and testing datasets respectively to illustrate the complexity of the datasets, and Table I lists the relevant statistics of the datasets.

Note that, from Fig. 5, a group of pedestrians is labelled in the datasets as a single dynamic object provided the group has a common heading. Taking a model-free approach, our goal here is to identify the dynamic hazard, to be able to describe and predict its motion and estimate its extend. All of these requirements can be satisfied by treating the group as a single object. In

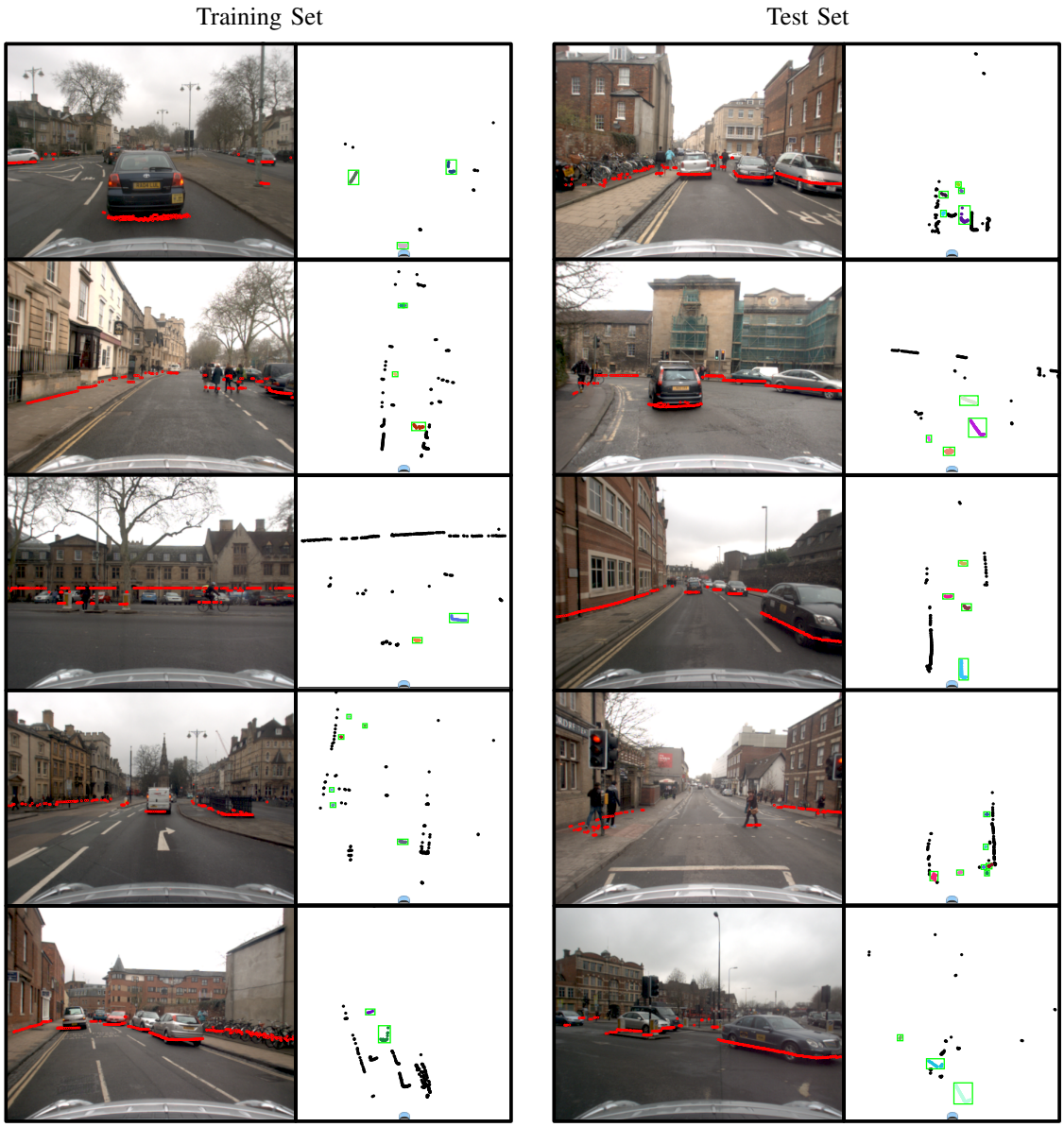



Fig. 5: Example frames from the datasets used for training and testing, demonstrating the variety and complexity of dynamic scenarios covered by this challenging dataset. For each column, similar to Fig. 2, the left panel shows the image from the onboard camera *for visualisation only*, with laser scan points projected into the image plane, and the right panel shows the corresponding laser scan with *ground truth* labels. Each ground truth object is highlighted by a green box. The left column shows five example challenging scenarios from the training set. From top to bottom: (1) a complex junction, (2) an area frequently traversed by pedestrians and groups of pedestrians, (3) a T-junction, also showing a cyclist travelling orthogonal to the ego-vehicle heading, and a pedestrian waiting for crossing (note in this scenario the pedestrian is *stationary* hence is *not* included in the ground truth labelling, (4) a situation where a large number of pedestrians can be observed by the laser at a far distance, (5) a car manoeuvring from its parked location. Similarly, the right column gives five example challenging situations from the test set. From top to bottom: (1) a narrow street with pedestrians and manoeuvring cars, (2) a complex junction with oncoming cars turning at different rates (relative to the ego-vehicle motion), (3) a narrow road with significant oncoming traffic, (4) a busy pedestrian crossing, (5) a wide turn resulting in a large relative motion to oncoming vehicles. Colour version available online.

TABLE I: Details of the training and test datasets. Here each count of an "object" is a single observation of an object instance in a single laser scan.

| Dataset | No. Laser Frames | Duration (min) | Drive Length (km) | No. Objects |
|---|---|---|---|---|
| Training | 3508 | 4.68 | 1.04 | 7517 |
| Test | 2151 | 2.87 | 0.82 | 5928 |

fact, the semantic description that this single "object" is actually composed of several "pedestrians" is neither achievable nor necessary given a completely model-free approach. Also, more specifically, this imposes unrealistic demands to the clustering algorithm where the cluster boundaries in this case are really only defined semantically. Individuals who split from a group are labelled as separate objects as soon as their motion differs from the common motion of the group.

### *B. Evaluation Metric and System Training*

We evaluate the system's ability to detect dynamic objects against the ground truth using the standard Precision and Recall metrics. Specifically, Precision and Recall are computed over the detected object boxes against the hand-labelled ground truth object boxes using the overlapping criterion as is commonly used in the computer vision community [Everingham et al., 2010]. An object box is marked as a true detection if it overlaps with a ground truth object box by more than a fixed percentage threshold. In all our results, we use $0.5$ as the percentage overlap threshold. And a detection is matched to at most one ground truth object, and multiple detections of the same ground truth object are treated as false positives.

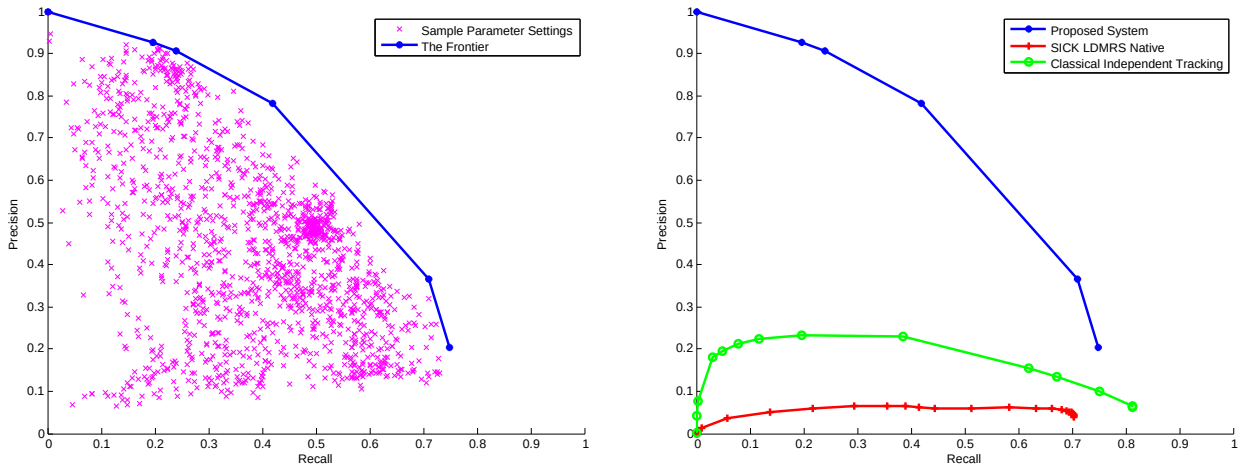
To train the system for best performing parameter sets, we follow an approach similar to that described in [Gavrila and Munder, 2007] as follows: both Precision $P$ and Recall $R$ are functions of system parameters, thus if the number of system parameters exceeds one, the set of all feasible $(R, P)$ pairs will in general occupy a continuous 2D space in the $R$-$P$ plane. The best parameters are then the parameters that give rise to the $(R, P)$ pairs at the *frontier* of the feasible region (conceptually corresponds to the top-right boundary of the feasible region, see Fig. 6(a) for an example).

Formally, the 2D feasible region parameterised by the set of all possible parameters $\mathcal{P}$ is given by $\mathcal{F} = \{(R(\mathbf{p}), P(\mathbf{p})) : \mathbf{p} \in \mathcal{P}\}$, the frontier parameter set of $\mathcal{F}$ is given by $F = \{\mathbf{q} \in \mathcal{P} : \forall \mathbf{p} \in \mathcal{P},\ R(\mathbf{p}) \leq R(\mathbf{q}) \text{ or } P(\mathbf{p}) \leq P(\mathbf{q})\}$. In other words, a parameter set is in $F$ if and only if it is not possible to achieve both a higher Precision and a higher Recall.

To find this frontier parameter set, we apply a Bayesian parameter tuning algorithm developed by Snoek et al. [Snoek et al., 2012] to bias the search in the high-dimensional parameter space to look for satisfactory parameter settings, and obtain an approximation to the frontier parameter set by finding the upper part of the convex hull of the obtained $(R, P)$ scatter plot. Fig. 6(a) shows the obtained $1803$ sample parameter settings with the algorithm, and the the blue curve shows the extracted frontier.

Since the SICK LDMRS native tracking system clusters each incoming scan and keeps track of every cluster, it makes no distinction between static and dynamic objects. To compare the systems under the same setting, we take tracks with estimated speeds higher than a given threshold to be the detected dynamic objects. It would be desirable to be able to fine-tune the parameters of the LDMRS's native tracking system. However, the most critical parameters are fixed internally to the sensor, and modifications are unfortunately not feasible.

The independent tracking baseline is similar in the respect that it also does not distinguish between static and dynamic objects. Hence thresholding on the object's absolute speed is also used to find dynamic objects to compare with the proposed




(a) (b)

Fig. 6: (a) Scatter plot of the obtained 1803 sample parameter settings using [Snoek et al., 2012] with the estimated frontier overlaid. (b) Precision-Recall tuning curves for the proposed system, the SICK LDMRS native tracking system, and the independent tracking baseline. See the online version for colour.

system. However, in the case of this baseline, we have control over the internal parameters, thus we apply the same Bayesian parameter tuning technique also to the classical independent tracking baseline to find the best internal parameter setups as well as the best speed threshold.

Fig. 6(b) presents the Precision-Recall curves for the proposed and the two baseline systems for comparison. The curve of the LDMRS's native system is generated by varying the speed threshold as described above, whereas for the classical independent tracking baseline, shown in the plot is the extracted frontier after Bayesian parameter selection including both the internal parameters and the speed threshold. As can be seen, the proposed system outperforms both baseline systems by a significant margin. This is somewhat expected, since both the LDMRS's native system and the independent tracking approach track only the cluster centroids, which are not stable reference points on the objects to track due to occlusions and dependency on the sensor viewpoint. On the other hand, the proposed system enforces each track's frame of reference to be attached rigidly to the object, and dynamic objects are explicitly handled differently to static ones. The somewhat bizarre behaviour of the Precision-Recall curves for the two baseline systems at the low Recall end is an artefact of the fact that at a speed threshold that is too high, all the sensible speed estimates are below the threshold while there exists false positives with some speed higher than the threshold. In this scenario, one obtains zero Precision *and* zero Recall.

The performance of the LDMRS's native detection system according to the Precision-Recall curves compares inconceivably poorly even against the simple independent tracking baseline. The reason for this apparently poor performance may be explained by the inability to optimise for the system's internal parameters. To see how much difference it makes, Fig. 7 plots the results for the *classical independent tracking* baseline (whose internal parameters we have total control of) in common axes with a Precision-Recall curve for the same baseline generated by fixing the internal parameters to a hand-tuned set of values and only varying the speed threshold, simulating the generation of the Precision-Recall curve in the case of the LDMRS's native system. As can be noted, the Bayesian optimisation algorithm is able to quickly discover different parameter settings that are better than the hand-tuned values, resulting in a much better achievable performance.

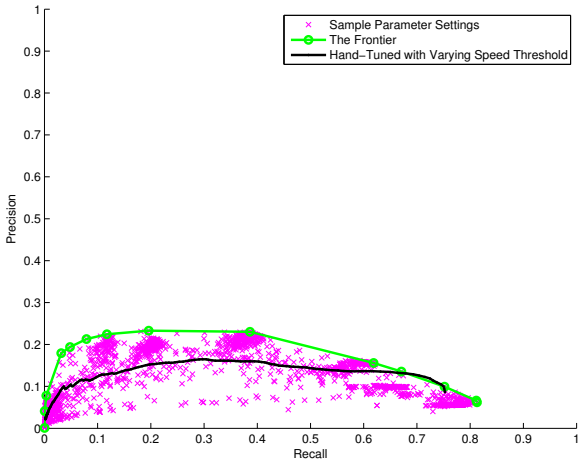



Fig. 7: An experiment to provide insights to the importance of Bayesian parameter tuning. All results are obtained with the classic independent tracking baseline. Shown on common axes are: a scatter of $(R, P)$ pairs of the 1749 sample parameter settings evaluated by the Bayesian parameter tuning algorithm [Snoek et al., 2012], the extracted frontier based on the sampled parameter settings, and a Precision-Recall curve generated by varying *only* the speed threshold fixing all the internal parameters to a set of hand-tuned values (similar to how the Precision-Recall curve for the LDMRS's native tracking system in Fig. 6(b) is obtained). Colour version online.

TABLE II: Quantitative evaluation of the three systems on the test dataset using the parameters selected with the help of the training set (see text for details). Performance is measured with the standard Precision and Recall metrics, as well as the corresponding $F_1$-measures for a comparison of overall performance in both Precision and Recall. The best performing system with respect to a specific metric is highlighted in bold.

| | Proposed System | SICK LDMRS Native | Classical Independent Tracking |
|---|---|---|---|
| Precision | **0.45** | 0.07 | 0.14 |
| Recall | 0.39 | **0.70** | 0.23 |
| $F_1$-measure | **0.42** | 0.13 | 0.18 |

### *C. Test Case Performance : a Quantitative Evaluation*

Given a range of operating points along the Precision-Recall curve, we choose empirically a single parameter setting that achieves the best balanced performance from Fig. 6(b) for each system. Specifically we choose the parameter setting that gives $R = 0.53$ and $P = 0.57$ for the proposed system, the speed threshold that achieves $R = 0.69$ and $P = 0.05$ for the LDMRS's native system, and the parameter setting (including the speed threshold) that gives $R = 0.39$ and $P = 0.23$ for the classical independent tracking baseline. All experiments that follow report metrics evaluated on the *test* dataset using these chosen operating points.

Table II lists performance metrics evaluated on the test set using the chosen parameter settings for each system. Here in addition to Precision and Recall we also report the standard $F_1$-measure as an overall performance measure taking account of both Precision and Recall. The trend observed during system training (cf. Fig. 6(b)) remains also in the test case. Judging by the $F_1$-measure, the classical independent tracking baseline outperforms the SICK LDMRS's native tracking system, while our proposed system outperforms both by a significant margin. Note however, the absolute figures differ from the training case. In particular, with the selected speed threshold, the LDMRS's native tracking system performs slightly better on the test set than on the training set (a $F_1$ value on the training set of $0.09$ from $R = 0.69$ and $P = 0.05$ compared with $0.13$ on the test

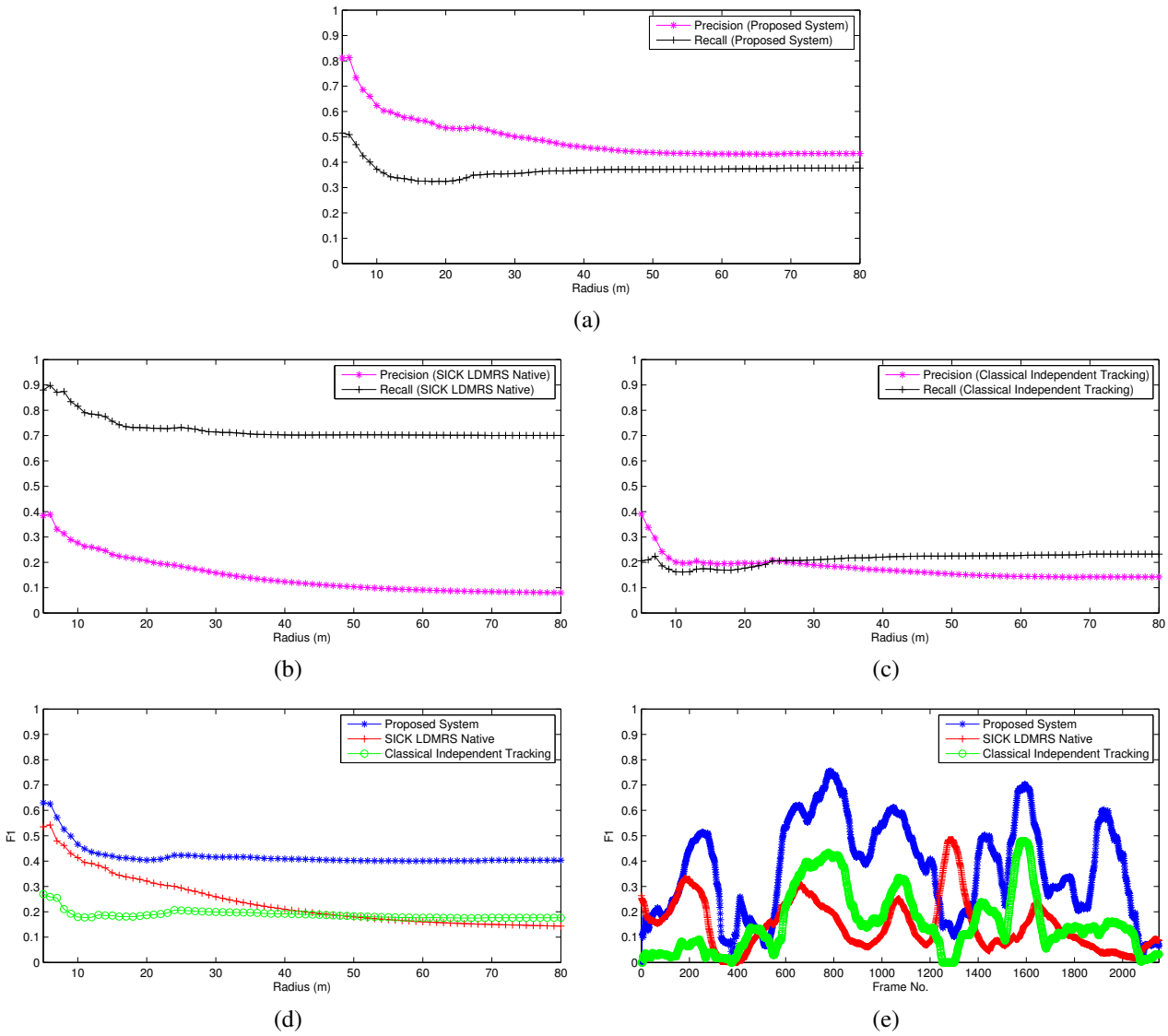



Fig. 8: (a) Precision and Recall versus operating radius for the proposed system. (b) Precision and Recall versus operating radius for the SICK LDMRS's native system. (b) Precision and Recall versus operating radius for the independent tracking baseline. (d) $F_1$-measure versus operating radius for all three systems. (e) $F_1$-measure over past 100 frames versus frame number for the three systems. Colour version online.

set), emphasising further the importance of using a different test set for unbiased comparison.

Fig. 8(a-c) show performance metrics for the three systems on the test dataset as the detection range is varied. All three systems show a decreasing trend on both Precision and Recall as the detection radius increases. The independent tracking baseline shows an interesting trend on its Recall, which after an initial drop increases back up at further distances. This is likely due to the fact that objects far away tend to appear as well-separated clusters of very small sizes – a situation particularly suited to simple tracking techniques that track only the cluster centroids such as the independent tracking approach here.

Fig. 8(d) places the systems under common axes using the $F_1$-measure for comparison. From the figure, although the close-range performances of the proposed system and the LDMRS's native tracking system are similar (with the proposed system slightly outperforming), the difference is significant from 20m onwards. Interestingly, the classical independent tracking baseline exhibit close to uniform performance over different detection ranges. Although performing worse than the LDMRS's native

tracking system at close ranges, the independent tracking baseline performs (albeit slightly) better at further ranges, agreeing to the trend observed during training (cf. Fig. 6(b)).

Fig. 8(e) compares the instantaneous performance at each frame of the three systems. $F_1$-measures are evaluated at each frame based on detections of the past $100$ frames for each system, and results are plotted against the frame number. While the proposed system outperforms the LDMRS at most frames, there are occasional performance drops. Closer inspection into the dataset reveals that around Frame $400$ there exists a period of driving with very few number of dynamic objects present, hence the apparent low performance from both systems. However, near to Frame $1300$, many walking pedestrians close to background clutter are present which are missed out by the proposed system due to segmentation failure. The LDMRS performs better in this scenario but in sacrifice of Precision. The performance of the independent tracking baseline follows approximately the same trend over time as that of the proposed system, albeit at a lower quality, suggesting the performance of any model-free tracking approach is heavily influenced by scene complexity. We take a closer look into the instantaneous performance of the proposed system on the test dataset in the next section, and identify some interesting cases, both the challenging but successful ones, and the common failure modes, together with cases where the performance of the proposed system differs from those of the two baseline systems and discuss about possible reasons for it being so.

### *D. Test Case Performance : a Qualitative Evaluation*

Continuing with the setup from the previous section, we take a closer look behind the numbers in this section. All qualitative results presented in this section are obtained by running the respective system (our proposed system and the two baselines for comparison) on the *test* dataset using the parameter set selected during the training phase (cf. Section VII-B) for unbiased evaluation.

Fig. 9 presents some examples of situations where the proposed system is able to successfully detect and track the dynamic objects in the scene, despite of the complexity of a real-life driving scenario in a busy town centre. The top left example shows the performance of the proposed system during busy oncoming traffic. All oncoming and leading cars are tracked successfully. Although two pedestrians can be observed on the left, they are not tracked because they are stationary. Here we note again that, in our proposed model-free approach to moving object detection and tracking, only *instantaneously* moving objects are detected and tracked, for example, parked or instantaneously stationary cars are also not included. The top right example gives a complex situation where the system has been successful in tracking a manoeuvring vehicle, two walking pedestrians (one is pushing a bicycle) as a group and another car travelling in the far field. Some pedestrians walking along walls on the right are missed due to segmentation failure (under-segmentation with the wall). In the bottom left example, our vehicle itself is turning giving rise to a large relative motion to the tracked objects. Here a bicyclist (highlighted), together with a vehicle behind, are tracked successfully. The far field “miss-detections”, upon closer look, may actually correspond to unlabelled moving pedestrians at a large distance. The bottom right example shows an interesting situation where a group of pedestrians (highlighted) are successfully tracked however the corresponding ground truth label is missing. This suggests that imperfect ground truth labelling may also contribute, to a certain degree, to a possibly underestimated system performance. Note the car on the left in this example is parked.

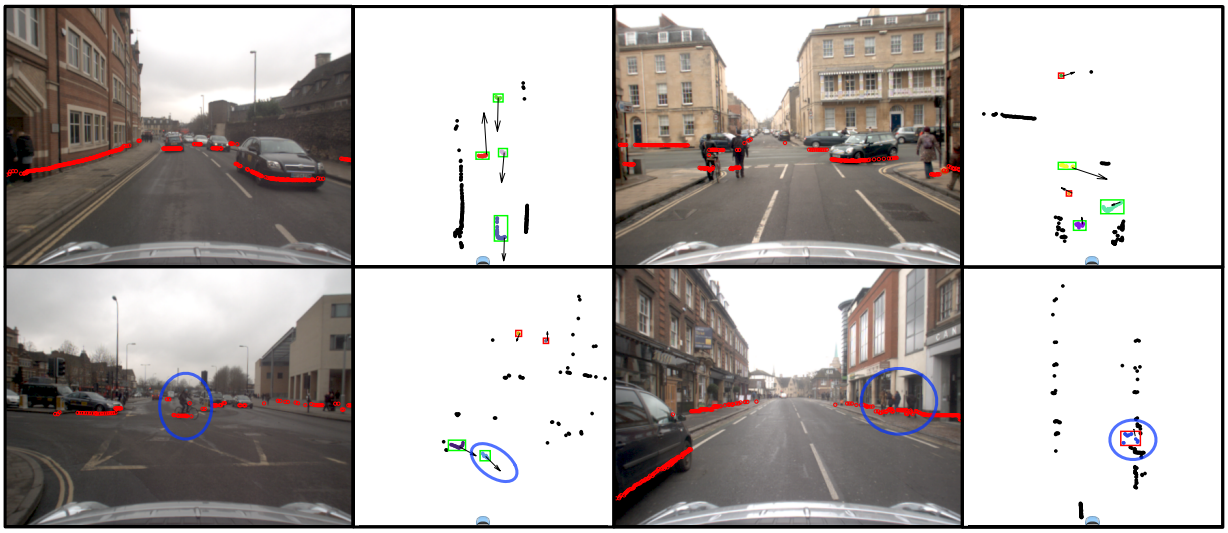
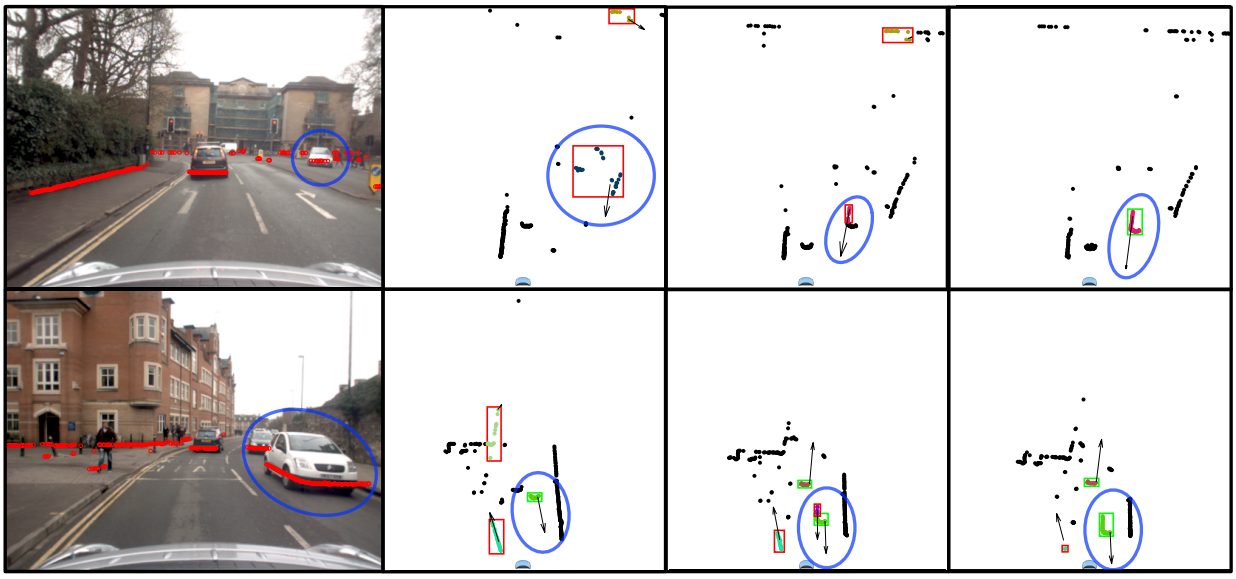


Fig. 9: Examples of cases where the proposed system is able to successfully detect and track dynamic objects under challenging situations. As in Fig. 2, green boxes represent true detections according to the ground truth labels, and red boxes denote false alarms. Blue ellipses highlight interesting situations in the scene. Images are provided for visualisation *only*. See text for details. Colour version available online.

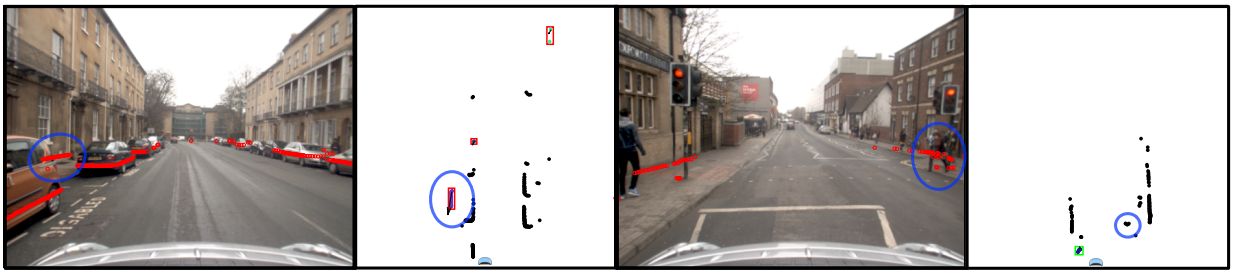
Fig. 10 studies some common failure modes of the proposed system. We divide this study into two different cases: a recoverable case, where despite initial tracking failure, the system subsequently is able to recover from the incorrect states, and an unrecoverable case, where an object is erroneously tracked or missed until it moves out of the field-of-view of the sensor.

Fig. 10(a) gives two common causes of recoverable failure. The top row shows the situation where some measurements on the static background get erroneously associated with a moving vehicle being tracked, due to under-segmentation (left laser scan). However, the hierarchical data association procedure (cf. Section VI) means that in the next iteration, this unreasonably large "object" is likely to be associated with more measurements on the static background. This in turn results in its estimated motion to drop, until eventually low enough to be merged with the static background (cf. Section V-D). Then because of motion inconsistency, soon a new object track will be initialised on the same vehicle (middle laser scan), and old boundary points that are supposed to belong to the tracked vehicle but falsely merged with the static background previously will expire because they are no longer observed. After that the system successfully recovers and keeps good track of the vehicle (the right scan). The bottom row, on the other hand, shows a case of recoverable failure due to *over*-segmentation. Because of viewpoint changes, a car is first erroneously tracked as two separate segments (the middle scan). But thanks to the merging procedure (cf. Section V-D), the system soon realises these segments should belong to the same object, and the car is subsequently tracked successfully as a single entity (the right scan). Although during a recoverable failure, the system eventually corrects its mistakes, there is nonetheless a transient time during which tracking is incorrect. Most of recoverable failures are due to segmentation errors. Despite our efforts in obtaining perceptually coherent segments in introducing the EMST-EGBIS algorithm in Section VI-E, segmentation errors are inevitable. One possible way to improve on this matter is to introduce semantics into the framework, to actively *look for* certain known types of cluster boundaries. We will return to this point in our discussions in Section VIII.

Fig. 10(b) presents two common cases of unrecoverable failure. The inclusion of a static background in our model helps



(a)



(b)

Fig. 10: Common failure modes of the proposed system. (a) The recoverable cases. From left to right: an image with projected laser data to provide some context *only*, and tracking results from the proposed system showing the evolution of the system's perception over time. (b) The unrecoverable cases. In particular, the case on the right is potentially dangerous. In both (a-b), as with other figures, green boxes denote true detections, red boxes denote false alarms. Blue ellipses highlight regions of interest. See text for details. Colour version available online.

significantly in resolving ambiguities resulting from viewpoint changes or occlusion (this will be demonstrated later qualitatively in this section). However, in regions beyond the current extend of the estimated static background, false detections may still arise when a static structure appears dynamic due to sensor motion, such as the short section of the wall visible only through inbetween the two parked cars (highlighted) in the left example in Fig. 10(b). This false detection may not be as severe because the erroneous "dynamic" object appears to travel parallel and in reverse direction as our own vehicle. The failure case in the example on the right however, is very *critical*. In this case, it is possible that sometimes motion of slow objects such as pedestrians is below the threshold for detection. When this happens, if the pedestrian remains walking in a low speed, they may be missed entirely, albeit crossing *right in front of the vehicle*. This situation exposes a weak point of purely model-free approaches to moving object detection (such as the proposed system here). Because of a lack of any semantic interpretation, the situation is possible to occur no matter what system parameters are actually used, provided a pedestrian walks *slowly enough*. Again, one possible solution is to bring in model-based elements in aiding difficult situations such as a slow walking pedestrian. We include this possible extension in our discussions in Section VIII.

Fig. 11 gives three common cases where the performance of the proposed system tends to differ from that of the two

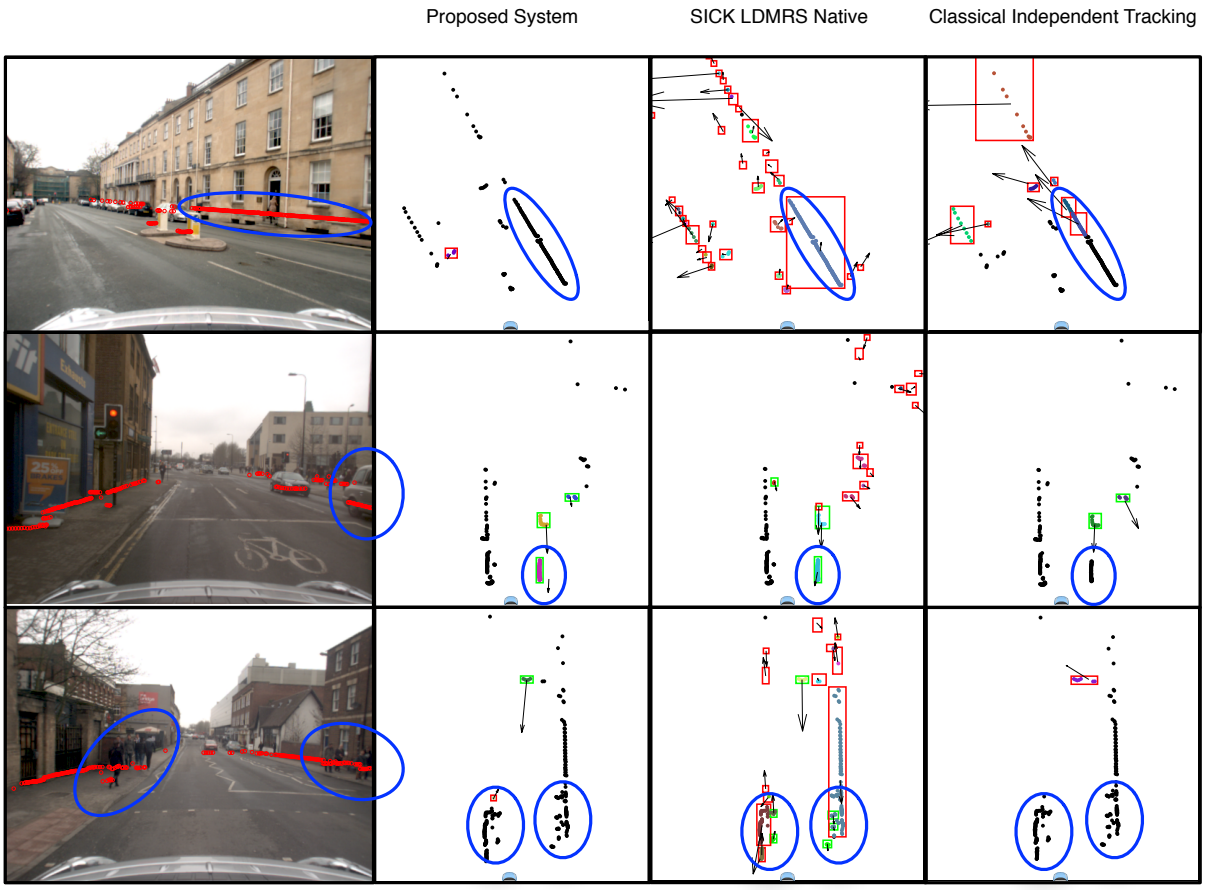



Fig. 11: Three cases of differing performance over the three systems evaluated quantitatively in Section VII-C. Each row is structured from left to right: the camera image for visualisation *only*, the tracking result of the proposed system, the tracking result of the SICK LDMRS's native system, and that of the classical independent tracking baseline. As with other figures, green boxes represent true detections, red boxes indicate false alarms. Interesting areas are highlighted with blue ellipses. See text for details. Colour version available online.

baselines we compared with quantitatively in Section VII-C. The first example (the top row), demonstrates the importance of keeping an estimate of a local static background as part of the state. As can be seen from the figure, both baselines give false detections around the section of the wall on the right, due to either uncertain motion estimates (LDMRS native tracking, middle scan), or data association failure (independent tracking, right scan). The proposed system, however, suffers from none of these problems, because as soon as the measurements belonging to the wall are assigned to the static background, they are *known* to be stationary. The second example (the middle row), shows different system responses in the event of viewpoint changes. A car (highlighted) traveling towards our vehicle in the opposite direction is being tracked successfully until it gradually goes out of view. The appearance of the car thus changes significantly as it moves past the sensor. Both baseline systems, because they all track only the cluster centroids, do not have a stable reference tracking point on the car, thus the motion estimates they produce are incorrect (in the case of the independent tracking, it is believed that the car has stopped). Our proposed system, on the other hand, tracks a fixed reference point rigidly attached to the object due to our special object representation (cf. Section IV-B), hence is able to maintain correct tracking on the car (left laser scan). The third example (bottom row) examines in detail the discrepancy of performance of the three systems around Frame 1300 observed in Fig. 8(e) earlier. As may be noted from the

detection results shown, due to noisy sensor measurements and segmentation errors, the two groups of pedestrians (highlighted) on pavements either side of the road are largely missed by both the proposed, and independent tracking systems. The LDMRS's native tracking system, however, is able to detect and track the majority of the pedestrians with success, albeit at the sacrifice of a large number of false positives. The fact that both the proposed system and the independent tracking baseline make the same mistakes in this case may suggest that the error is due to either a common mistake made by the clustering algorithm (both approaches use the EMST-EGBIS clustering algorithm described in Section VI-E), or the way multiple scanning layers are fused into a single scan described in Section VII-A. The mechanisms the LDMRS's native tracking method relies on to fuse information from multiple scanning layers are not known to us. Some specifics of the information fusion mechanism may have proven successful in this particular situation.

### *E. Timing*

In this section, we take an empirical analysis of the computational efficiency of the proposed system. The parameter training procedure followed in Section VII-B is excellent in investigating the full potential of a system in terms of system performance. However, different parameter settings inevitably give rise to different computational requirements. For example, a larger maturity threshold (cf. Section V-D) means a larger number of tentative tracks may be kept around in the system before they are merged with the static background or existing object tracks, increasing the load of the system. The best performing parameter set is not necessarily the most efficient parameter set. As a compromise, we hand-tuned the system parameters by visual inspection using *only* the training dataset for a reasonable performance and at the same time a satisfactory computational time. We made sure during this hand-tuning process the test set is completely hidden away from us so that any performance and timing measures remain unbiased. The resulting parameter set gives a Precision of $0.48$ and a Recall of $0.37$ evaluated on the *test* dataset, corresponding to an $F_1$-measure of $0.42$. These figures compare well with quoted values in Table II using the best performing parameter set selected during the training phase.

Fig. 12 then shows timing results obtained with this hand-tuned parameter set on the test dataset as statistics over processing times for the laser and odometry measurement types respectively. The results are generated using our current prototype implementation in MATLAB on a MacBook Pro equipped with a quad-core 2GHz Intel i7 CPU and 8GB of RAM. Fig. 12(a) shows a plot of the time taken per frame over the entire test sequence (in milliseconds) in the case of laser measurements, together with a rough measure of scene complexity at each time instant. Scene complexity here is measured by the number of ground truth objects present at any given instant. Some degree of correlation may be observed between computation time per frame and scene complexity, especially during the first half of the sequence (up to around Frame $1200$). Of course, we note here computation time per frame should not completely depend on scene complexity measured by the number of dynamic objects present, because at each laser frame the static background also needs to be updated. The average time taken per frame evaluates to 336ms, corresponding to a frame rate of around 3Hz if buffering is used. Fig. 12(b) shows a similar time plot for odometry measurements. Update times taken on each odometry measurement appear much more stationary compared with those of the laser measurements. This makes intuitive sense because odometry updates do not depend on scene complexity. The average time taken per odometry update evaluates to a mere $0.52$ms, confirming the theoretical insight gained in Section V-A.

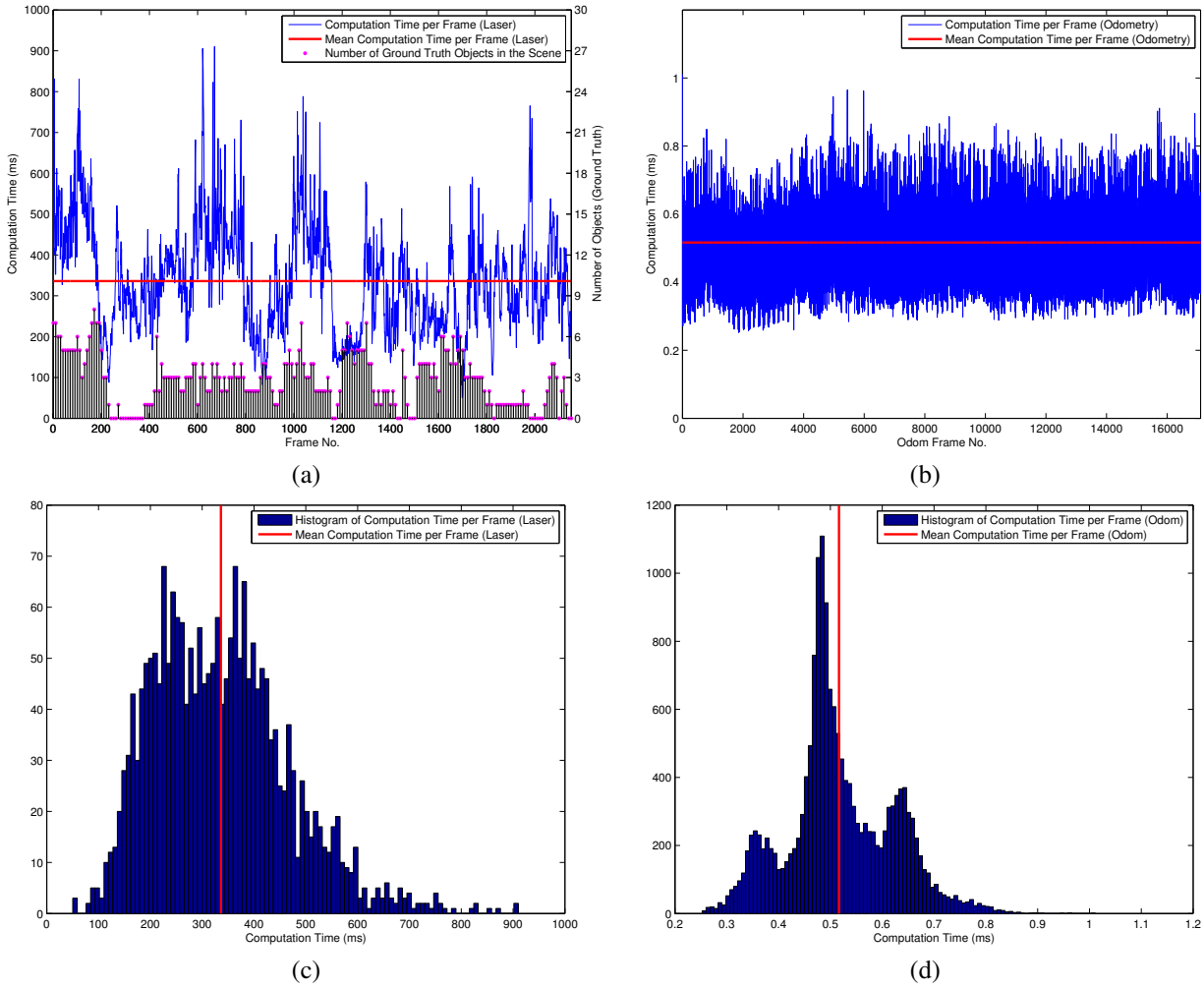



Fig. 12: (a) Computation time per frame for each laser frame. Also shown on the same graph is a simple measure of scene complexity given by the number of ground truth objects present at each frame. There exists a certain degree of correlation between computation time and scene complexity. (b) Computation time per frame for each odometry measurement frame. (c) A histogram of computation time per frame in the case of laser measurements. (d) A similar histogram for odometry measurements. In (a-d), the mean computation time per frame is indicated by either a horizontal line (time plots) or a vertical line (histograms). See the online version for colour.

Fig. 12(c) and Fig. 12(d) look at the distribution of computation times laid out as histograms for the laser and odometry measurement types respectively. The distribution for the laser measurement type is unimodal and has a tail in the high value end, while the distribution of computation time in the case of odometry updates exhibit a peculiar multimodal shape which may correspond to the "spikes" at both the high and low ends of the plot in Fig. 12(b). Finding a theoretical justification for this form of the distribution is, however, challenging, yet it is not of the primary concern for all practical purposes.

## VIII. Conclusion and Discussions

In this paper, we presented a unified Bayesian framework for jointly estimating the sensor pose, a local static background and dynamic states of moving objects. The main focus of our work is on the detection and tracking of moving objects independent of classes and shapes. We described our model-free representation of objects using boundary points initialised with raw laser measurements, and derived their observation models. The dynamics of the moving objects are inferred as hidden variables

under a rigid body constraint, making the quality of the data association algorithm critical to the system's correct operation.

Therefore, within the same unified framework, we proposed a novel two-level data association algorithm that takes benefits of both the density of observations and strong correlations between them. A new variant of the JCBB [Neira and Tardos, 2001] algorithm was suggested to tackle with large numbers of measurements, and a solution to numerical stability issues under such scenarios was also presented.

Finally, the proposed system was tuned systematically on real-world data against hand-labelled ground truth, and both quantitative and qualitative evaluations demonstrated the system's superior performance over an existing industry standard also targeted at object tracking for automotive applications and a classical independent model-free tracking approach.

There are several advantages in taking a model-free approach over a model-based approach to object tracking. For example, dynamic objects are identified and tracked independent of their shapes and classes, lifting the design efforts required in model-based approaches to treat each object class separately. Also, unexpected object classes are covered in the same framework with no difference to other familiar object classes, thus providing full situational awareness.

Despite these advantages, a purely model-free approach does lack semantic interpretation to the objects it tracks, giving difficulties in many folds, some of which we have already encountered in Section VII-D. There is still room for improvements over the proposed model-free framework. In the future, we aim to extend our model-free framework with model-based elements similar to the approach taken by [Vu and Aycard, 2009] to bring the best of both worlds. Specifically, model-based elements may be designed as "plugins" to the existing model-free framework serving only to improve its performance. Individual object classes (for example, the familiar object classes cars, pedestrians and bicyclists) can be added one at a time or whenever a corresponding "plugin" is available. Each of such modules provides additional semantic interpretation for the objects the model-free tracker is currently tracking, aiding it in many fronts. First, given semantic interpretations, we may bias the output of the EMST-EGBIS algorithm to correctly segment objects that are known to the model-based modules, reducing segmentation errors. Second, once an object of a given class is recognised, it may be flagged as a *potential* dynamic object even if it is not actually moving or moving in a low speed. Finally, additional semantic knowledge will enable us to apply more sophisticated motion models, or even multiple models (e.g. the Interacting Multiple Model (IMM) filter [Zhao and Thorpe, 1998]) to objects whose class is known with confidence to better account for all complexities of motion the object may exhibit. This may include, for example, in the case of pedestrians, a slow motion model to help with the situation encountered in Section VII-D. When an unexpected object is observed, or the object class may not be determined with confidence, the system falls back to model-free tracking, still maintaining a full situational awareness.

## ACKNOWLEDGMENT

This work is supported by the Clarendon Fund. Paul Newman is supported by an EPSRC Leadership Fellowship, EPSRC Grant EP/I005021/1. The authors wish to thank Jasper Snoek for making the Spearmint Bayesian optimisation package publicly available. We would also like to thank our reviewers for many insightful comments and helpful advices.

# REFERENCES


[Arras et al., 2008] Arras, K., Grzonka, S., Luber, M., and Burgard, W. (2008). Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1710–1715.

[Arras et al., 2007] Arras, K., Mozos, O., and Burgard, W. (2007). Using Boosted Features for the Detection of People in 2D Range Data. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3402–3407.

[Asano et al., 1988] Asano, T., Bhattacharya, B., Keil, M., and Yao, F. (1988). Clustering algorithms based on minimum and maximum spanning trees. In *Proceedings of the fourth annual symposium on Computational geometry*, SCG '88, pages 252–257, New York, NY, USA. ACM.

[Bar-Shalom et al., 2002] Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R. (2002). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA.

[Besl and McKay, 1992] Besl, P. and McKay, N. D. (1992). A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256.

[Biswas et al., 2002] Biswas, R., Limketkai, B., Sanner, S., and Thrun, S. (2002). Towards object mapping in non-stationary environments with mobile robots. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 1014–1019 vol.1.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338.

[Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vision*, 59:167–181.

[Gavrila and Munder, 2007] Gavrila, D. M. and Munder, S. (2007). Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. *Int. J. Comput. Vision*, 73(1):41–59.

[Granström, 2012] Granström, K. (2012). *Extended target tracking using PHD filters*. PhD thesis, Linköping University, Automatic Control, The Institute of Technology.

[Hahnel et al., 2003] Hahnel, D., Triebel, R., Burgard, W., and Thrun, S. (2003). Map building with mobile robots in dynamic environments. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1557–1563 vol.2.

[Leonard et al., 2008] Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., Koch, O., Kuwata, Y., Moore, D., Olson, E., Peters, S., Teo, J., Truax, R., Walter, M., Barrett, D., Epstein, A., Maheloni, K., Moyer, K., Jones, T., Buckley, R., Antone, M., Galejs, R., Krishnamurthy, S., and Williams, J. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774.

[Mertz et al., 2013] Mertz, C., Navarro-Serment, L. E., MacLachlan, R., Rybski, P., Steinfeld, A., Suppe, A., Urmson, C., Vandapel, N., Hebert, M., Thorpe, C., Duggins, D., and Gowdy, J. (2013). Moving object detection with laser scanners. *Journal of Field Robotics*, 30(1):17–43.

[Miyasaka et al., 2009] Miyasaka, T., Ohama, Y., and Ninomiya, Y. (2009). Ego-motion estimation and moving object tracking using multi-layer LIDAR. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 151–156.

[Montesano et al., 2005] Montesano, L., Minguez, J., and Montano, L. (2005). Modeling the Static and the Dynamic Parts of the Environment to Improve Sensor-based Navigation. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 4556–4562.

[Neira and Tardos, 2001] Neira, J. and Tardos, J. (2001). Data association in stochastic mapping using the joint compatibility test. *Robotics and Automation, IEEE Transactions on*, 17(6):890 –897.

[Schulz et al., 2001] Schulz, D., Burgard, W., Fox, D., and Cremers, A. (2001). Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1665–1670 vol.2.

[Snoek et al., 2012] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. In *Neural Information Processing Systems*.

[Tipaldi and Ramos, 2009] Tipaldi, G. and Ramos, F. (2009). Motion clustering and estimation with conditional random fields. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 872–877.

[Topp and Christensen, 2005] Topp, E. and Christensen, H. (2005). Tracking for following and passing persons. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2321–2327.

[van de Ven et al., 2010] van de Ven, J., Ramos, F., and Tipaldi, G. (2010). An integrated probabilistic model for scan-matching, moving object detection and motion estimation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 887–894.

[Vu and Aycard, 2009] Vu, T.-D. and Aycard, O. (2009). Laser-based detection and tracking moving objects using data-driven Markov chain Monte Carlo. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3800–3806.

[Vu et al., 2007] Vu, T.-D., Aycard, O., and Appenrodt, N. (2007). Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 190–195.

[Wang et al., 2003] Wang, C.-C., Thorpe, C., and Thrun, S. (2003). Online Simultaneous Localization And Mapping with Detection And Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

[Wang et al., 2012] Wang, D., Posner, I., and Newman, P. (2012). What Could Move? Finding Cars, Pedestrians and Bicyclists in 3D Laser Data. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*.

[Wang et al., 2013] Wang, D., Posner, I., and Newman, P. (2013). A New Approach to Model-Free Tracking with 2D Lidar. In *Proceedings of the International Symposium on Robotics Research (ISRR)*, Singapore.

[Williams, 2001] Williams, S. B. (2001). *Efficient Solutions to Autonomous Mapping and Navigation Problems*. PhD thesis, Australian Centre for Field Robotics, The University of Sydney.

[Wolf and Sukhatme, 2005] Wolf, D. F. and Sukhatme, G. S. (2005). Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. *Auton. Robots*, 19(1):53–65.

[Yang and Wang, 2011] Yang, S.-W. and Wang, C.-C. (2011). Simultaneous egomotion estimation, segmentation, and moving object detection. *Journal of Field Robotics*, 28(4):565–588.

[Zahn, 1971] Zahn, C. (1971). Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *Computers, IEEE Transactions on*, C-20(1):68 – 86.

[Zhao and Thorpe, 1998] Zhao, L. and Thorpe, C. (1998). Qualitative and Quantitative Car Tracking from a Range Image Sequence. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '98, pages 496–, Washington, DC, USA. IEEE Computer Society.