A New Approach to Model-Free Tracking with 2D Lidar

Dominic Zeng Wang, Ingmar Posner and Paul Newman

Abstract This paper presents a unified and model-free framework for the detection and tracking of dynamic objects with 2D laser range finders in an autonomous driving scenario. A novel state formulation is proposed that captures joint estimates of the sensor pose, a local static background and dynamic states of moving objects. In addition, we contribute a new hierarchical data association algorithm to associate raw laser measurements to observable states, and within which, a new variant of the Joint Compatibility Branch and Bound (JCBB) algorithm is introduced for problems with large numbers of measurements. The system is calibrated systematically on 7.5K labeled object examples and evaluated on 6K test cases, and is shown to greatly outperform an existing industry standard targeted at the same problem domain.

1 Introduction

In this paper, we describe a unified framework for the detection and tracking of moving objects from a 2D laser range finder for autonomous driving applications. The aim of this work is to formulate a light-weight standalone system that takes a minimal number of sensory inputs to produce reliable motion estimates for *only* objects that are dynamic at the time of observation. Moreover and central to this work, we place no requirement on the shape or parametric form of the tracked objects.

Recent years have seen a succession of triumphs of autonomous navigation systems on the road. Both the successes of the DARPA Grand [11] and Urban [14] Challenges and recent demonstrations across the world heighten our community's belief that self-driving vehicles are truly within our reach. Safe navigation of such systems is a challenging but yet arguably the most critical task that requires sen-

Mobile Robotics Group
University of Oxford
e-mail: {dominic, ingmar, pnewman}@robots.ox.ac.uk

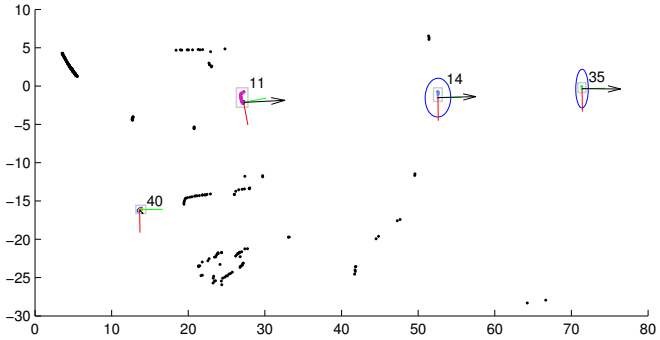


Fig. 1 A typical system output. Detections are highlighted with bounding boxes. Frame axes attached to each detection show objects' local reference frames, and coloured points indicate estimated locations of object boundary points (cf. Section 4.1). Uncertainty ellipses are shown for each object's estimated position, and numbers next to each detection denote unique tracking ID's. Three manoeuvring cars and one walking pedestrian are detected, of which two cars' motions are being predicted in the absence of direct observation. Note the scene clutter and it is far from easy to say what is a car and what is not. Note also that the vehicle itself is moving from frame to frame.

sible interactions with complex dynamic environments. To be able to perceive the dynamic aspects of the environment and predict future movements of the manoeuvring objects is thus essential to every successful autonomous vehicle on the road.

It has been observed by many authors [8, 16] 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 jointly the sensor pose, a local static background that maps the static structure around the sensor and the dynamic states of the tracked moving objects, but with an emphasis on the task of dynamic object detection.

Fig. 1 shows a typical output of the proposed system. As can be noted, detection and tracking of dynamic objects is particularly challenging in the urban driving scenario due to the presence of a significant amount of background clutter, further complicated by the fact that the sensor itself is also attached to a moving vehicle. Despite of the difficulties, our system is able to successfully identify and track moving objects *without* any restriction to their type.

In what follows, Section 2 reviews existing approaches to detection and tracking of dynamic objects. Section 3 states our main contributions in the paper. Section 4 presents the details of our proposed system. We then quantitatively evaluate the performance of the proposed system with real-world data, and show that it outperforms an industry standard solution that was designed for the same problem domain of object tracking from a moving sensor in Section 5. Finally we conclude the paper in Section 6.

2 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 [2].

In the autonomous driving application domain, however, further complications arise when moving targets are usually buried deep within significant background clutter and they exhibit more complex motions than single point targets. 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. Most existing practical dynamic tracking systems, for example, systems deployed in the Urban Grand Challenge [6, 7], function by first segmenting measurements from multiple laser range finders, and then extracting geometric features from the segments, which are used to compile a list of object hypotheses. Then, the dynamic objects are 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 along side detecting and tracking of moving objects. Examples include Toyota's tracking system [8] and Wang's system [16] 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.Yang and Wang [18] 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. [12, 15] 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.

We also mention another body of work that is targeted at detection and tracking of particular object classes of interest. For example, Arras et al. [1] and Topp and Christensen [13] focus on the detection and tracking of people from a laser range finder by first detecting legs from a segmented laser scan and group them into person tracks. Our proposed system is different in that moving objects of any class and shape may be modelled.

3 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.

In addition, we propose a hierarchical data association algorithm to assign raw laser measurements to potential state updates, and present a greedy variant of the Joint Compatibility Branch and Bound (JCBB) algorithm [9] that is suitable for associating a large number of measurements.

4 Model-Free Tracking of Moving Objects

The system we propose is run within a recursive Bayesian framework (implemented as an Extended Kalman Filter). In this section, we describe in detail the system formulation in terms of state representation, prediction and measurement models as well as how data association is handled within the same framework.

4.1 An Unusual State Representation

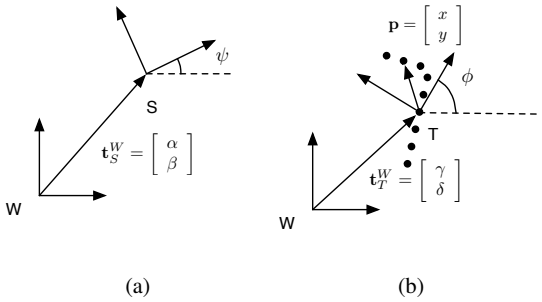
The motions of dynamic objects can be arbitrary and independent of each other. The sensor, however, does not observe their motions directly but ranges and bearings of points on the surface of the objects. Thus once conditioned on the measurements, motions between different objects become correlated, 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 from static objects. The state therefore consists of three parts: the sensor pose, the dynamic objects, and the static map.

Sensor Pose Representation and Related The sensor pose $\mathbf{x}_S = [\alpha, \beta, \psi]^T$ is represented by a 2D transform from the sensor's frame of reference to a stationary world frame of reference as depicted in Fig. 2(a), which is updated by vehicle odometry measurements at the prediction stage, and by laser measurements as part of the update stage as will be described in Section 4.2.

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 $\mathbf{x}_C = [\delta\alpha, \delta\beta, \delta\psi]^T$. This is the 2D transform that transforms points from the sensor frame into the vehicle frame.

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, thus its motion state is represented by the 6-vector $\mathbf{x}_T^i = [\gamma_i, \delta_i, \phi_i, \dot{\gamma}_i, \dot{\delta}_i, \dot{\phi}_i]^T$ as shown in



(a) (b)

Fig. 2 Illustrations of frame conventions and variable definitions. (a) Transform from the sensor frame to the world frame. (b) Transform from a track frame to the world frame, with boundary points represented locally to the track frame.

Fig. 2(b) (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 as illustrated in Fig. 2(b). 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. 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 4.3.2.

We make the assumption that dynamic objects observed in the 2D scanning plane of the sensor behave as rigid bodies. This assumption, though 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 are unaltered at forward-prediction.

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.

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.

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, $\mathbf{x}_b$ the collection of boundary points on the static background, $\mathbf{x}_p$ the collection of all boundary points of all dynamic objects arranged sequentially, and finally, $\mathbf{x}_C$, the extrinsic calibration parameters of the sensor.

4.2 Top Level Algorithm Description

In this section, we give a top level description of the algorithm. Each time a new measurement arrives, the mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$ of the joint state are updated differently according to the type of the measurement (odometry or laser).

Odometry Measurement Processing 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.

Laser Measurement Processing When a new laser scan frame arrives, the current state is first tested against out-of-date dynamic tracks and boundary points on the static background that have fallen out of the sensor's field of view. These are removed from the joint state. It also determines parts of the static background that have changed due to a static object transitioning into a dynamic state, hence must be removed to allow a new dynamic track to be initiated. Next, the motion part of all dynamic tracks is forward-predicted according to an appropriate motion model as will be described in Section 4.3.1, and followed by data association and measurement updates (Section 4.3.2). 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 object. This merging procedure is described in Section 4.3.3.

4.3 Detailed Algorithm Description

4.3.1 Dynamic Object Motion Prediction

At the prediction step after conducting a laser measurement, all dynamic tracks are predicted forward according to a generic motion model before being updated with the measurements. To capture a wide range of dynamic objects, it is desirable to use a general motion model. In this work, all dynamic tracks are assumed to follow the constant velocity model [2, Chapter 6].

4.3.2 Hierarchical Data Association

Not all state variables in the joint state (Section 4.1) 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 that

1. It is an observation on a static object.
   a. It is an observation of an existing boundary point.
   b. It is an observation of a new boundary point.
2. It is an observation on an existing dynamic object.
   a. It is an observation of an existing boundary point on the object.
   b. It is an observation of a new boundary point on the object.
3. It is 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.

Coarse Level Data Association The measurements in a given laser scan are first divided into a set of clusters $\mathscr{C} = \{C_1, C_2, \ldots, C_{|\mathscr{C}|}\}$. The clusters are then assigned to the static background and dynamic objects recursively with the ICP [3] algorithm as follows. First, boundary points on the static background are aligned to the set of measurements Z with ICP, and clusters in $\mathscr{C}$ 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 $\mathscr{C}$ and a similar procedure follows recursively for each dynamic track. The clusters that remain in $\mathscr{C}$ 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 as will be detailed in Section 4.3.3.

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) [9] 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 4.1, 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).

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:

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

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

Joint Compatibility Under the assumption of independent observations, the joint observation model of a complete association σ is given by

$$\mathbf{h}_\sigma(\mathbf{x}) = [\mathbf{h}^T_{\sigma(1)}(\mathbf{x}), \mathbf{h}^T_{\sigma(2)}(\mathbf{x}), \ldots]^T \,, \tag{3}$$

with the innovation covariance

$$\mathbf{S}_\sigma = \begin{bmatrix} \mathbf{H}_{\sigma(1)}\mathbf{P}\mathbf{H}_{\sigma(1)}{}^T + \mathbf{R} & \mathbf{H}_{\sigma(1)}\mathbf{P}\mathbf{H}_{\sigma(2)}{}^T & \cdots \\ \mathbf{H}_{\sigma(2)}\mathbf{P}\mathbf{H}_{\sigma(1)}{}^T & \mathbf{H}_{\sigma(2)}\mathbf{P}H_{\sigma(2)}{}^T + \mathbf{R} & \\ \vdots & & \ddots \end{bmatrix}, \tag{4}$$

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 σ is jointly compatible if

$$(\hat{\mathbf{z}}_\sigma - \mathbf{h}_\sigma(\hat{\mathbf{x}}))^T \mathbf{S}_\sigma^{-1} (\hat{\mathbf{z}}_\sigma - \mathbf{h}_\sigma(\hat{\mathbf{x}})) \leq \chi^2_{N_a d,\alpha} \,. \tag{5}$$

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

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 σ^* that gives the lowest joint Normalised Innovation Squared (jNIS, defined to be the expression to the left of the inequality in Equation 5).

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.

We introduce the JCBB-Refine algorithm, which instead of aiming to find the optimum assignment σ^*, 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 σ_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 σ_0. The initial association σ_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 σ_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 σ_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. Explicit forms of our observation models for different types of boundary points are stated in an appendix.

Recursive Updates in Triangular Form It is shown [9] 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 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, so that $\mathbf{S}_k^{-1} = \mathbf{G}_k \mathbf{G}_k^T$, $\mathbf{G}_k = \mathbf{U}_k^{-1}$ (note $\mathbf{G}_k$ is also upper triangular). And the next iteration selects a new boundary point to be assigned to a measurement such that

$$\mathbf{S}_{k+1} = \begin{bmatrix} \mathbf{S}_k & \mathbf{W}_k^T \\ \mathbf{W}_k & \mathbf{N}_k \end{bmatrix} . \tag{6}$$

Then it can be shown that

$$\mathbf{U}_{k+1} = \begin{bmatrix} \mathbf{U}_k & \mathbf{R}_k^T \\ \mathbf{0} & \mathbf{F}_k \end{bmatrix} , \tag{7}$$

where $\mathbf{R}_k = \mathbf{W}_k \mathbf{G}_k$, $\mathbf{F}_k = \mathrm{chol}(\mathbf{N}_k - \mathbf{R}_k \mathbf{R}_k^T)$, and

$$\mathbf{G}_{k+1} = \begin{bmatrix} \mathbf{G}_k & -\mathbf{G}_k \mathbf{R}_k^T \mathbf{M}_k \\ \mathbf{0} & \mathbf{M}_k \end{bmatrix} , \tag{8}$$

where $\mathbf{M}_k = \mathbf{F}_k^{-1}$. In addition, we keep track of a vector $\boldsymbol{\xi}_k = \mathbf{G}_k^T \boldsymbol{\nu}_k$. Its update equation is given by $\boldsymbol{\xi}_{k+1} = [\boldsymbol{\xi}_k^T, \boldsymbol{\mu}_k^T]^T$ where $\boldsymbol{\mu}_k = \mathbf{M}_k^T (\tilde{\boldsymbol{\nu}}_k - \mathbf{R}_k \boldsymbol{\xi}_k)$. Here, $\tilde{\boldsymbol{\nu}}_k$ is the innovation vector of the newly assigned measurement. And the jNIS can be updated simply as $\mathrm{jNIS}_{k+1} = \mathrm{jNIS}_k + \boldsymbol{\mu}_k^T \boldsymbol{\mu}_k$. Given $\mathbf{U}_k$, $\mathbf{G}_k$ and $\boldsymbol{\xi}_k$ at any stage of the computation, the innovation covariance $\mathbf{S}_k$, its inverse and the innovation vector $\boldsymbol{\nu}_k$ can be easily retrieved if needed.

We note this form has the same computational complexity as the recursion introduced in [9], but is more numerically stable.

4.3.3 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 [17], 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, a fictitious noiseless measurement of values all zero on the absolute velocity (including the angular velocity) of the tentative track is considered. If this measurement passes the validation gate, the tentative track is considered to be a static object, therefore should be merged into the static background. The fictitious measurement is hence used to update the joint state $\mathbf{x}$ as if it was an actual sensor measurement. After the update, all available information from the track will have been transferred to the rest of the joint states, and it can 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.

The same merging procedure is also conducted at the end of each processing cycle for testing each existing track against merging with the static background and other existing tracks as mentioned in Section 4.2.

5 System Evaluation

In this section, we quantitatively evaluate the proposed system, and compare its performance against an industrial standard solution for benchmarking. We note there exists a large body of work on similar application domains (For example, [8, 7, 16]), 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.

5.1 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 mobile 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.

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

Table 1 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 frame.

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

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. Table 1 lists the details of the two datasets respectively. Ground truth detections of dynamic objects are obtained from both datasets by manual labelling.

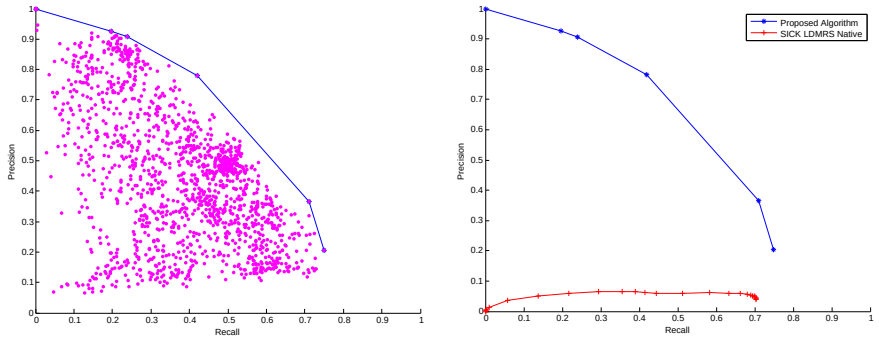
5.2 Evaluation Metric and System Training

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-labeled ground truth object boxes using the overlapping criterion as is commonly used in the computer vision community [4]. 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 [5] 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. 3(a) for an example).

Formally, the 2D feasible region parameterised by the set of all possible parameters $\mathscr{P}$ is given by $\mathscr{F} = \{(R(\mathbf{p}),P(\mathbf{p})) : \mathbf{p} \in \mathscr{P}\}$, the frontier parameter set of $\mathscr{F}$ is given by $F = \{\mathbf{q} \in \mathscr{P} : \forall \mathbf{p} \in \mathscr{P},\ R(\mathbf{p}) \leq R(\mathbf{q})$ 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. [10] 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. 3(a) shows the obtained 1803 sample parameter settings with the algorithm, and the the blue curve shows the extracted frontier.



(a) (b)

Fig. 3 (a) Scatter plot of the obtained 1803 sample parameter settings using [10] with the estimated frontier overlaid. (b) Precision-Recall tuning curves for the proposed system and the SICK LDMRS native tracking system.

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.

Fig. 3(b) presents the Precision-Recall curves for the proposed, and LDMRS's native tracking systems. The curve of the LDMRS's native system is generated by varying the speed threshold as described above. As can be seen, the proposed system outperforms the LDMRS's native system by a significant margin. This is somewhat expected, since the LDMRS's native system tracks 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.

5.3 Test Case Performance

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. 3 for each system. Specifically we choose the parameter setting that gives $R = 0.53$ and $P = 0.57$ for the proposed system and the speed threshold that achieves $R = 0.69$ and $P = 0.05$ for the LDMRS's native system. All experiments that follow report metrics evaluated on the *test* dataset using these chosen operating points.

Fig. 4(a) and (b) show performance metrics for the two systems on the test dataset

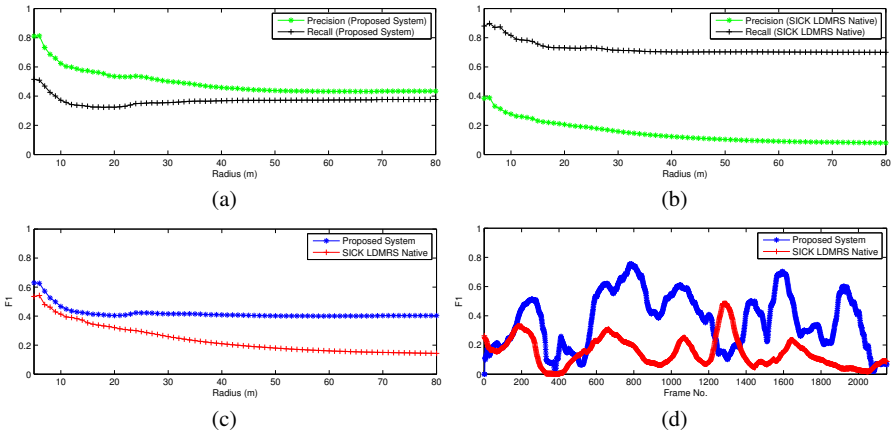


Fig. 4 (a) Precision and Recall versus operating radius for the proposed system. (b) Precision and Recall versus operating radius for the SICK LDMRS's native system. (c) F_1-measure versus operating radius for both systems. (d) F_1-measure over past 100 frames versus frame number for both systems.

as the detection range is varied. Both show a decreasing trend on both Precision and Recall as the detection radius increases. Fig. 4(c) places the systems under common axes for comparison. From the figure, although the close-range performances are similar (with the proposed system slightly outperforming), the difference is significant from 20m onwards.

Fig. 4(d) compares the instantaneous performance at each frame of the two 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.

Our current prototype implementation of the proposed system in MATLAB runs in real-time at 2Hz on a MacBook Pro equipped with a duo-core 2.4GHz Intel i5 CPU and 4GB of RAM.

6 Conclusions

We presented a unified framework for jointly estimating the sensor pose, a local static background and dynamic states of moving objects, focused mainly on accu-

rate moving object detection. Observations in our formulation are raw sensor measurements and object states are inferred as hidden variables under a rigid body constraint.

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 [9] algorithm was suggested to tackle with large numbers of measurements, and a solution to numerical stability issues under such scenarios was also presented.

The proposed system was tuned systematically, and demonstrated to outperform an existing industry standard.

Acknowledgements 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.

Appendix

In this appendix, we state the exact forms of the observation models applied to boundary points on the static background and dynamic objects respectively. All variables involved in what follows are defined in Section 4.1, and the function $\mathbf{u}$ maps a pair of 2D cartesian coordinates into polar coordinates.

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:

$$\mathbf{h}_j(\mathbf{x}) = \mathbf{u}(\mathbf{g}(\mathbf{x}_S, \mathbf{b}_j)) \ , \ \mathbf{g}(\mathbf{x}_S, \mathbf{b}_j) = \mathbf{R}^T(\psi) \left(\begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right) . \tag{9}$$

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* frame, and is given by:

$$\mathbf{h}_j(\mathbf{x}) = \mathbf{u}(\mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i)) \ , \ \mathbf{g}(\mathbf{x}_S, \mathbf{x}_T^i, \mathbf{p}_j^i) = \mathbf{R}^T(\psi) \left(\mathbf{R}(\phi_i) \begin{bmatrix} x_j^i \\ y_j^i \end{bmatrix} + \begin{bmatrix} \gamma_i \\ \delta_i \end{bmatrix} - \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right) . \tag{10}$$

References

1. Arras, K., Grzonka, S., Luber, M., Burgard, W.: 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, pp. 1710–1715 (2008)

2. Bar-Shalom, Y., Kirubarajan, T., Li, X.R.: Estimation with Applications to Tracking and Navigation. John Wiley & Sons, Inc., New York, NY, USA (2002)
3. Besl, P., McKay, N.D.: A method for registration of 3-D shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on **14**(2), 239–256 (1992)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision **88**(2), 303–338 (2010)
5. Gavrila, D.M., Munder, S.: Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. Int. J. Comput. Vision **73**(1), 41–59 (2007)
6. 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., Williams, J.: A perception-driven autonomous urban vehicle. Journal of Field Robotics **25**(10), 727–774 (2008)
7. Mertz, C., Navarro-Serment, L.E., MacLachlan, R., Rybski, P., Steinfeld, A., Suppe, A., Urmson, C., Vandapel, N., Hebert, M., Thorpe, C., Duggins, D., Gowdy, J.: Moving object detection with laser scanners. Journal of Field Robotics **30**(1), 17–43 (2013)
8. Miyasaka, T., Ohama, Y., Ninomiya, Y.: Ego-motion estimation and moving object tracking using multi-layer LIDAR. In: Intelligent Vehicles Symposium, 2009 IEEE, pp. 151–156 (2009)
9. Neira, J., Tardos, J.: Data association in stochastic mapping using the joint compatibility test. Robotics and Automation, IEEE Transactions on **17**(6), 890 –897 (2001)
10. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian Optimization of Machine Learning Algorithms. In: Neural Information Processing Systems (2012)
11. Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., Mahoney, P.: Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics **23**(9), 661–692 (2006)
12. Tipaldi, G., Ramos, F.: Motion clustering and estimation with conditional random fields. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pp. 872–877 (2009)
13. Topp, E., Christensen, H.: Tracking for following and passing persons. In: Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, pp. 2321–2327 (2005)
14. Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M.N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T.M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.W., Singh, S., Snider, J., Stentz, A., Whittaker, W.R., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., Ferguson, D.: Autonomous driving in urban environments: Boss and the Urban Challenge. Journal of Field Robotics **25**(8), 425–466 (2008)
15. van de Ven, J., Ramos, F., Tipaldi, G.: An integrated probabilistic model for scan-matching, moving object detection and motion estimation. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 887–894 (2010)
16. Wang, C.C., Thorpe, C., Thrun, S.: 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 (2003)
17. Williams, S.B.: Efficient Solutions to Autonomous Mapping and Navigation Problems. Ph.D. thesis, Australian Centre for Field Robotics, The University of Sydney (2001)
18. Yang, S.W., Wang, C.C.: Simultaneous egomotion estimation, segmentation, and moving object detection. Journal of Field Robotics **28**(4), 565–588 (2011)