An evaluation of dynamic object tracking with 3D LIDAR

P. Morton, B. Douillard, J. Underwood
Australian Centre for Field Robotics (ACFR)
The University of Sydney, NSW Australia
{p.morton, b.douillard, j.underwood}@acfr.usyd.edu.au

Abstract

This paper evaluates experimentally approaches to tracking dynamic objects using 3D LIDAR data such as that obtained from a Velodyne HDL-64. Whilst many algorithms exist for tracking objects in 2D laser data, only recently have algorithms which model the 3D appearance of tracked objects been proposed. The benefits of these algorithms with respect to model free centroid based methods have not been fully examined, and therefore the key contribution of this paper is to analyse the behaviour of different implementations of these two broad approaches. Through a quantified comparison of simple tracking methods for 3D LIDAR data validated against manually labelled tracks it was found that centroid tracking outperformed 3D appearance model based techniques. In the experiments, which focused upon pedestrian tracking in particular, centroid tracking achieved up to 95% correct data associations compared to 60% for the appearance model based methods.

1 Introduction

The task of tracking, whereby the state of a moving object is estimated based on remote measurements [Bar-Shalom *et al.*, 2001], has long been studied; initially in the context of RADAR sensing, and more recently in video and laser data and combinations thereof.

Traditional tracking methods, initially developed for RADAR applications, operate on point observations and vary, therefore, mostly in their methods of data association and state approximation. These approaches are designed principally to allow for measurement inaccuracy, spurious observations caused by clutter or background noise and the potentially ambiguous relationship between observations and the physical objects which generate them [Bar-Shalom, 1978].

The development of laser ranging devices means that a different type of information is available to tracking systems and 2D LIDAR data in particular has been well studied, with algorithms developed for SLAM [Bailey and Nieto, 2008], object tracking [Arras *et al.*, 2007] and combinations of the two [Wang, 2004]. The dense 3D data obtained from modern LIDAR systems such as the Velodyne is fundamentally different to traditional sensors used in tracking such as RADAR primarily due the density of the information it provides, not only about a target and its shape but also about the entire scene (but mostly at relatively short ranges compared to the capabilities of some long range RADARs).

Early approaches to the problem of tracking in 3D LIDAR data, such as those used for the 2007 DARPA Urban Challenge, generally projected the 3D point cloud into a 2D representation [Darms *et al.*, 2008], [Montemerlo *et al.*, 2008]. For example, the Stanford entry re-sampled Velodyne data into 2D "virtual scans" in order to perform segmentation and tracking [Petrovskaya and Thrun, 2009]. Most of the teams relied on the assumption that all moving objects encountered during the challenge would be vehicles, and could therefore be represented by box models. Other tracking approaches also make use of 2D slices of the 3D Velodyne data as a way to cope with sloping ground [Navarro-Serment *et al.*, 2010].

Work by Spinello *et al.* (2010) proposes a learnt classifier which is able to detect pedestrians in 3D LIDAR data. By combining these detections into a multi-hypothesis tracking framework [Spinello *et al.*, 2011] the authors are able to robustly track the pedestrians within a scene, but do not consider other moving objects.

Recently, algorithms have been proposed which can segment the full 3D point cloud from a Velodyne sensor [Douillard *et al.*, 2011], [Moosmann *et al.*, 2009]. Given a good segmentation, tracking becomes a problem of motion estimation, registration, appearance modelling and data association. Approaches to motion estimation include independent Kalman Filters with nearest neighbour data association [Shackleton *et al.*,

2010] and Monte Carlo approximations to a full data-assignment solution such as Rao-Blackwellized particle filters (RBPFs) [Miller *et al.*, 2011], amongst others. Appearance modelling is a challenging step, since the non-rigid nature of objects such as pedestrians make template matching approaches difficult. Whilst many options for appearance modelling exist, ranging from centroids alone to features such as spin images [Johnson, 1997], [Shackleton *et al.*, 2010] or full 3D point cloud [Moosmann and Fraichard, 2010], the contribution of these models to overall tracking performance has not been fully analysed.

The principal contribution of this paper, therefore, is to examine how fundamental choices made in the design of a 3D LIDAR based tracking system affect its performance. Trackers of varying complexity are evaluated with respect to a ground truth comprising both labelled trajectories and localisation data from on-board sensors.

The layout of this paper is as follows. Firstly, some updates to the "mesh-voxel" segmentation method [Douillard *et al.*, 2011] are presented. This is followed by an overview of the different tracking methods tested in this paper. Section 4 describes the datasets gathered and experiments performed, and Section 5 presents and analyses the results of these experiments. A conclusion and statement of future work is given in Section 6.

2 Segmentation

Segmentation (clustering 3D points into groups which represent separate objects) is an important precursor to tracking as, intuitively, points which belong to the same object tend to move together and so can be tracked as a single object. Whilst there are methods that seek to solve the segmentation and tracking problem simultaneously [Leonard *et al.*, 2008], each of the methods tested herein relies on an initial segmentation process.

The algorithm presented in this section builds on prior work [Douillard *et al.*, 2011]. An updated formulation is proposed in Algorithm 1. The main steps of this segmentation process are as follows: (1) construction of a terrain mesh from a range image (as originally proposed in [Moosmann *et al.*, 2009]), line 1 in Algorithm 1; (2) extraction of the ground points based on the computation of a gradient field in the terrain mesh, lines 2 to 9 in Algorithm 1; (3) clustering of the non-ground points based on connectivity in a voxel grid, lines 10 and 11 in Algorithm 1.

In the updated version proposed below, the segmentation process implements iterative ground extraction and propagation in each raster line of the range image by proceeding from the centre outwards. Mesh building (line 1), gradient computations and transition detection at each node (line 3 and 5) are implemented as in [Douillard *et al.*, 2011]. The main differences with respect to the previous version of the algorithm are the following.

Input : *scan*
Output : *segments*, *ground*
Parameters: *res*, *mingraddist*, *maxgrad*, *maxdh*

```
1  mesh ← BuildMesh(scan);
2  for line ∈ scan do
3      ComputeGradients(mesh, line, mingraddist);
4      PropagateGround(mesh, line, maxgrad, res);
5      DetectTransition(mesh, line, maxdh);
6      if no ground node found then
7          break;
8      end
9  end
10 voxels ← FillVoxelGrid(scan − ground,res);
11 segments ← SegmentVoxelGrid(voxels);
```

Algorithm 1: The Mesh Voxel Segmentation Algorithm

Nodes in a given scan line (a scan line corresponding to one of the "rings" traced by individual lasers of the Velodyne) can only be labelled as ground if the previous scan line contains at least one ground node (as detailed in [Douillard *et al.*, 2011]). The process of growing the ground partition (lines 1-9) being now conducted iteratively from the image's centre outwards, it can be stopped as soon as no more ground nodes are identified in the current scan line (lines 6-7). This optimisation allows to avoid attempting the identification of ground nodes when the connectivity to ground partition is no longer direct.

A second modification of the original algorithm is in the way the ground is propagated in each scan line (line 4). The initialisation of the ground in the inner most scan line follows the original implementation. The propagation of the label GROUND is then based on the following criteria: (1) the current node n has to have a gradient value below $maxgrad$, (2) n has to have at least one direct neighbour ne already labelled GROUND, whether ne is in the previous or the current scan line, (3) the difference between the range of n and ne must be positive (to some tolerance set to res in our implementation); i.e., ne must be closer to the sensor. The latter criteria was not used in the original implementation. However, it is necessary since it avoids the label GROUND to be propagated from the background of the scene along artificially flat links, caused by the horizontal geometry of the sparse scan pattern on distant objects. Such links can be observed in Fig. 1(a). Connecting forward objects to the background, these links often correspond to small slopes and as a consequence to flat gradients. Such flat links allow the GROUND label to be propagated from the background onto foreground objects, which materialises as ground points in the middle of object segments

as illustrated in Fig. 1(c). The sets of ground points extracted with and without criterion (3) are illustrated in Fig. 1(b) and 1(c), respectively. As can be seen in these figures, criterion (3) allows a more accurate extraction of the ground. One may argue that these longer links causing ground extraction artefacts could be discarded based on their length. This would require defining an additional threshold and would not allow to propagate the GROUND label when the data becomes sparser (typically, further away from the sensor).

An advantage of the Mesh-Voxel Segmentation algorithm is the explicit differentiation between ground and non-ground points. This has a number of distinct advantages. In the context of a mobile platform, this ground/non-ground separation is required for path planning and obstacle avoidance. In the context of tracking, ground points can be eliminated from further processing, helping to greatly reduce the computational load on the system. For example, projecting Velodyne returns into a global coordinate frame in real time (at 1,300,000 points/second) is a computationally expensive task. However, as Table 1 shows, from 40% up to 70% of all Velodyne returns may be classified as ground in typical outdoor scenes. Making use of this fact could help significantly in pushing these kinds of tracking algorithms towards real-time performance.

3 Tracking

Tracking in 3D LIDAR data can be separated into four principal components: *motion estimation*, *registration*, *appearance modelling* and *data association*. The combined choice of these four components forms a 'configuration space' for the tracking method. In this section, the individual component implementations are described.

The common input to all the methods tested in this paper is the 3D data points, which are augmented with additional columns of information. After segmentation, the data is filtered through a 870×64 pixel range image (as described in [Moosmann *et al.*, 2009]). This resamples the data to an equal resolution in both elevation and azimuth, and allows for the computation of surface normals and a range-based local descriptor [Moosmann and Fraichard, 2010]. The surface normals and local descriptors are only required by some of the tracking methods. Note also that points which were labelled 'ground' as a result of the segmentation are discarded and that filtered object points are projected into a global coordinate frame based on the current navigation solution. This process yields a set of points, with each point represented by a 13 dimensional vector;

$$t, x, y, z, range, descriptor, normal, segment_id \quad (1)$$

where t is the point's time stamp, x, y, z represent its position in the global coordinate frame, *descriptor* is a range-based local descriptor of the form (f_1, f_2, f_3, f_4), *normal* is the surface normal to the point (n_x, n_y, n_z), and *segment_id* is a unique integer id output by the mesh-voxel segmentation algorithm.

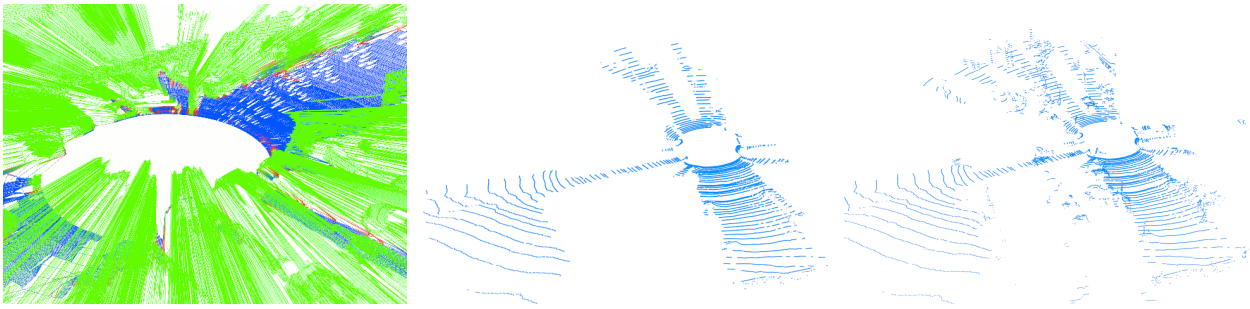
Input : *tracks*, *scan*
Output: *tracks*
for *track* $\in$ *tracks* **do**
    track $\leftarrow$ `Predict(`*track, timestamp*`)`;
    segment_id $\leftarrow$ `Associate(`*track, scan*`)`;
    motion $\leftarrow$ `Register(`*track, segment_id*`)`;
    track $\leftarrow$ `Update(`*track, motion, segment_id*`)`;
end
for *segment* $\in$ *scan* **do**
    if *segment not associated with a track* **then**
        new_track $\leftarrow$ `InitTrack(`*segment*`)`;
        tracks $\leftarrow$ `Append(`*tracks, new_track*`)`;
    end
end

Algorithm 2: Tracking framework

A simple tracking framework is described in Algorithm 2, where the above set of vectors is represented as *scan*. The algorithm uses a motion model to predict the likely location of future observations then associates these observations with the current set of tracks. The motion of the objects can be computed by registering the observations with the tracks, and this information is used to update the track's state. Any observations that are not associated with an existing track form new tracks. Variations on both the implementation and the ordering of these steps results in different tracking methods. For example, data associations may be computed for all tracks simultaneously to form a globally optimal solution, rather than the independent assignment solutions shown in the algorithm. Data association might also occur before or after registration depending on the particular tracking method. Several variations are tested in this paper, as described in the remainder of this section and summarised in Table 3.

3.1 Centroids

The simplest representation of a tracked object within a 3D point cloud is its *centroid* (the mean of the segmented object's points). This representation means that any traditional tracking method, such as those developed for RADAR data [Bar-Shalom, 1978] can be used. *Motion estimation* is used to predict the future location of a target, but in the most simple cases is not required – the 'predicted' location is simply where it was last observed. Another method is to employ a Kalman Filter for each tracked object, in which case the likely location of future observations are predicted according to a constant-velocity motion model. *Data association* is, for centroid



(a) Mesh (b) With Range Difference Checking (c) Without Range Difference Checking

Figure 1: (a) Illustration of the mesh built from the Velodyne range image (line 1 in Algorithm 1). Blue links have been identified as ground, green links as non-ground (once Algorithm 1 has reached line 9). The longer green links connecting the foreground objects to the background of the scene are the ones which may cause the label GROUND to propagate from the background onto objects (see text). (b), (c) Ground points obtained with and without checking the difference in range between neighour nodes during the propagation of the label GROUND. As can be seen, implementing this check allows a more accurate ground extraction (b) and avoids mis-labelling points effectively belonging to objects above the ground (c).

Table 1: Velodyne scan statistics

Dataset	**Ground**	**Non-Ground**	**Total**	**# scans**	**% ground**
Lawn area on University Campus	90644	40689	131334	1308	69.0
Urban streets	61669	64656	126325	4359	48.8
Sydney CBD	54361	69482	123843	2111	43.4
Tree-lined semi-urban street	69918	55644	125562	2747	55.7

Mean number of ground and non-ground points in a Velodyne scan as determined by mesh-voxel segmentation. The number of scans used from each dataset is also shown.

methods, a linear assignment problem on Euclidean distances between the centroids; a commonly-used solution to which is the Hungarian algorithm [Kuhn, 1955]. Many more robust data association approaches exist in the literature (e.g. [Kirubarajan and Bar-Shalom, 2004]) – but the simple methods described here were tested to provide a benchmark for later algorithms. *Registration*, for the *centroids* method, takes place after data association and uses the centroid of the new observation to adjust the tracked position or update the Kalman Filter.

3.2 Point Clouds

An alternative approach to object representation is to use all points belonging to a segmented object as the observation, and use this to form a track's appearance model. This was tried as part of a complete framework by Moosmann and Fraichard (2010), but in this paper the technique is broken down into parts so the contribution of each step to the tracking performance can be determined.

Appearance modelling can exploit the richer data provided by a point cloud object representation to enhance tracking performance. The simplest approach when tracking with point clouds is to create a track with a static appearance model – those points corresponding to the first observation. Following the framework mentioned above [Moosmann and Fraichard, 2010], registration is a multi-step process which yields the object's 6 degree of freedom (DOF) displacement and uses the local feature (from Equation (1)) to compute an initial estimate followed by registration refinement using the Point-Plane ICP (Iterative Closest Points) algorithm [Chen and Medioni, 1991]. Note that registration does not require segmentation of the current point cloud, instead each track is aligned individually via ICP to the complete current scan. The filter update, therefore, also requires no explicit data association.

Whilst the propagation of existing tracks requires no data association or segmentation of the scene, segmentation is still performed to enable the detection and instantiation of new tracks. During ICP, point-to-point associations between the point clouds being registered (the track's appearance model and the current scan) are formed. For each point in the appearance model, the *segment_id* of the associated scan point is computed. By taking a majority vote over these values, a single segment is determined and this segment is associated with the track. After all tracks have been processed, any remaining segments form new tracks. This method differs slightly from that used in [Moosmann and Fraichard, 2010], where track correspondence was computed from pixel overlap in the range image.

The weakness of using only the first observation of an object as the track's appearance model is that over time a tracked object may rotate relative to the sensor, exposing previously unseen faces. As the view changes, the similarity between the appearance model and the observed points may become insufficient for the ICP process. One approach to address this issue is to replace the track's appearance model with the new observation at each iteration. This ensures that the difference in appearance between the track and the actual object will always be small, and therefore a good ICP match should be possible.

Directly replacing the track's appearance model with each new observation has the disadvantage that an incorrectly associated observation will be used for subsequent registration steps, likely compounding the error and causing tracking to fail. A potentially more robust approach is to accumulate observations, but some form of re-sampling must be used to bound the number of points in the accumulated model.

The approach employed in [Moosmann and Fraichard, 2010] is to combine the existing appearance model with a new observation and filter the set of points based on their measurement uncertainty. For this work, the mean error in points returned by the LIDAR sensor was found (by the calibration method described in [Underwood *et al.*, 2010]) to be around 10cm and so a voxel grid with this resolution was used to filter points in the *constant density* experiment.

Combining points in this way without any form of batch association or re-adjustment will result in a blurring of the appearance model over time, as registration errors during each ICP alignment will accumulate. To address this, methods which gradually replace points in the appearance model with new observations were also tested. In one experiment, for example, a randomly sampled 50% of the points in the existing appearance model were combined with 50% of the points in the new observation to form the new appearance model of the track. This has the effect that the likelihood of a point remaining in the appearance model decays exponentially over time. In another experiment these values were set to 90% and 10% respectively, so that existing points were more likely to remain.

The aforementioned appearance modelling techniques can be used in tracks with or without Kalman Filter estimation (as described for centroids in Section 3.1).

4 Experiments

To evaluate the performance of different tracking methods, a range of experiments of increasing complexity were designed. These experiments were executed using the ACFR Centre for Intelligent Mobile Systems (CIMS) mobile sensor platforms "Shrimp" and "Mantis": skid-steer vehicles based on the Segway RMP-400 platform, which carry tactical-grade DGPS/INS navigation systems, on-board computing and comprehensive sensor payloads (Fig. 2). For these experiments, only the Velodyne LIDAR sensor mounted on "Shrimp" was used to perform tracking.

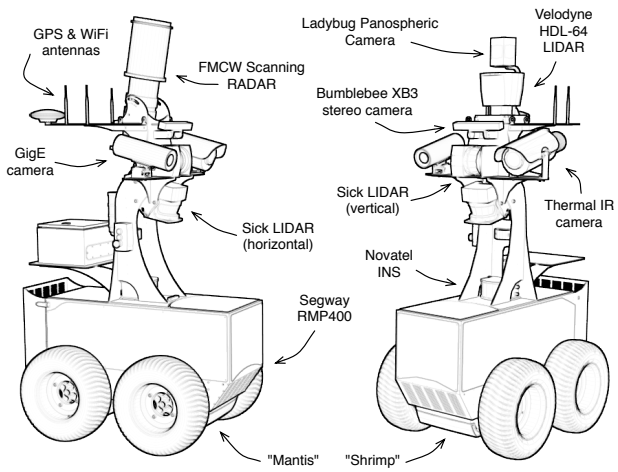


Figure 2: Mobile sensor platforms "Mantis" and "Shrimp"

The navigation systems of both robots were fed with DGPS corrections, yielding very accurate position and orientation estimates (the error in position was on average 2.5cm). The navigation information for "Shrimp" (the sensor vehicle) was used to map sensor readings into a global coordinate frame, and the navigation data from "Mantis" (the *target vehicle*) was used to provide a ground-truth position and velocity for its 'tracked' trajectory as seen by "Shrimp".

The datasets were acquired on a lawn area at the University of Sydney, measuring approximately 15m by 30m, with both robots moving under remote operator control as multiple people moved in and through the scene. An overhead view of the area and an example path followed by the sensor vehicle is shown in Fig. 3. Datasets of varying complexity were logged and used to evaluate the tracking methods off-line. Table 2 describes the datasets used for evaluation.

Nine different tracking methods were implemented and compared. Their configuration according to the *motion estimation*, *registration*, *appearance modelling* and *data association* options (described in Section 3) is shown in Table 3. For the centroids experiments, the constant velocity Kalman Filter (KF) modelled the location and velocity of the tracked objects with state vector, $\vec{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$, whilst for the ICP based tracking methods, the full 6 DOF state vector ($\vec{x} = (x, y, z, roll, pitch, yaw, \dot{x}, \dot{y}, \dot{z}, \dot{roll}, \dot{pitch}, \dot{yaw})$ was used.

Table 2: Datasets

ID	Duration (s)	# scans	Moving targets	Notes
A	137	1307	1 robot	Both sensor and target robot were driven according to no pre-planned trajectory
B	133	1261	1 robot, 3 people	People were asked not to walk behind trees or other objects, and to stay 2m from each other
C	149	1411	1 robot, 3 people	People were allowed to walk behind trees and obstacles, but remain in vicinity of robot
D	165	1558	1 robot, 3 people	People walked in and out of the scene (for a total of 6 different tracks)

Table 3: Tracking methods

ID	Registration	Estimation	Modelling	Association
1	centroid	none	centroid only	Hungarian algorithm, Euclidean distance
2	centroid	constant velocity KF	centroid only	Hungarian algorithm, Euclidean distance
3	ICP	none	1st observation	ICP track-scan links
4	ICP	none	most recent observation	ICP track-scan links
5	ICP	constant velocity KF	1st observation	ICP track-scan links
6	ICP	constant velocity KF	most recent observation	ICP track-scan links
7	ICP	constant velocity KF	50% current, 50% observation	ICP track-scan links
8	ICP	constant velocity KF	90% current, 10% observation	ICP track-scan links
9	ICP	constant velocity KF	constant density	ICP track-scan links

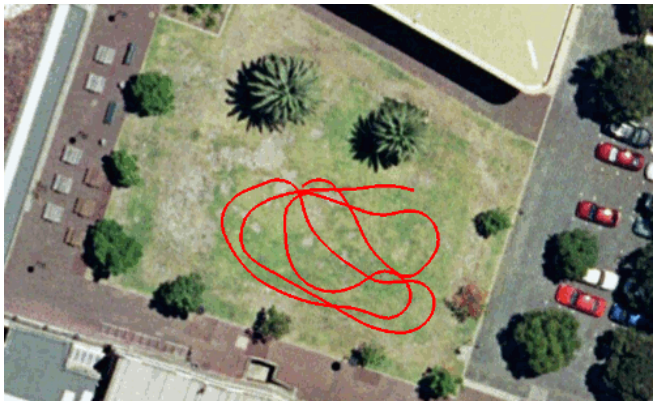


Figure 3: Lawn area where experiments were carried out, showing a trajectory computed from the sensor vehicle's nagivation system.

5 Results

To provide a benchmark for the comparison of the tracking methods, the tracks of the dynamic objects (the target vehicle and pedestrians) were hand labelled. This was achieved by taking the output of segmentation prior to any tracking, and annotating the segments with their correspondence to the dynamic objects. i.e. for each dynamic object, the sequence of *segment_ids* that comprise its trajectory are noted as a single list per object. This is equivalent to performing the tracker data association step manually.

The track segments produced by the tracker can now be compared to the manually labelled segments. In the case that an object was tracked correctly over the duration of the dataset, a perfect match is obtained. In the case that the tracker fragments the target trajectory, it is still possible to associate each fragment with the single physical target, thus the extent of the failure can be quantified.

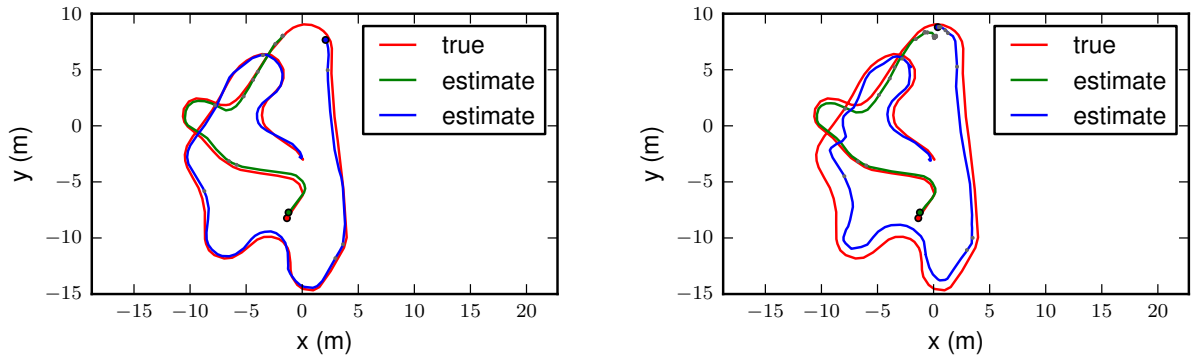
An intuitive first quantification of tracking performance is to evaluate the fragmentation of the target trajectories produced by the tracker. An ideal tracker will instantiate only one track for each moving object it observes. Any additional tracks created for the one object indicate that tracking failed, and was reinitialised. A case of track fragmentation is illustrated in Fig. 4(a). Over the duration of the dataset the tracker instantiated two separate tracks for the target vehicle. The fragmentation of the target vehicle's trajectory according to each tracking method is shown in Table 4. An important result demonstrated in this table is that information from past observations is required for robust appearance based tracking. Whilst the the majority of methods perform similarly well, it is methods 4 and 6 (which rely on the most recent observation only) that frequently lost track of the target vehicle.

Table 4: Fragmentation of vehicle tracks

	1	2	3	4	5	6	7	8	9
A	1	1	1	1	1	2	1	1	1
B	1	1	1	2	1	2	2	1	1
C	1	1	1	5	1	5	1	1	1
D	2	2	2	7	1	7	2	2	1

Number of track fragments which correspond to the target vehicle (ideal value is 1)

Additionally, the navigation solution of the target ve-



(a) Estimated target vehicle trajectory (Experiment D-2) (b) Estimated target vehicle trajectory (Experiment D-7)

Figure 4: (a) shows the trajectory estimate for the best performing tracker according to RMS position error for dataset D (experiment D-2). (b) shows the same plot for experiment D-7. Both tracking methods lost track of the object at some point in the experiment and restarted tracking later.

hicle was used to provide the 'true' path shown in Fig. 4. This means that the root mean squared (RMS) error between the true and estimated trajectories can be calculated and is shown in Table 5. In the case that multiple tracks correspond to the true object path, the RMS position error is calculated on the combination of all of the tracks. Due to the Velodyne sensor only being able to observe one side of an object, there is a systematic bias in both the Centroid and ICP tracking methods. Fig. 4 shows the results of running two of the trackers on dataset D. Visually, it is evident that tracker 2 estimates the position of the target vehicle better than tracker 7. This is reflected by the respectively low and high values in cells D-2 and D-7 in the table of RMS position errors (Table 5), indicating that despite the bias problem, RMS position error is a useful relative metric.

In terms of this metric, the centroid methods performed particularly well and it is the ICP methods that incorporate observations into their appearance model which performed poorly. As each observation is incorporated into the model, any ICP error causes misalignment of the new points. Over time these errors accumulate, causing the track to drift with respect to its coordinate frame. Method 6, which relies completely on the new observation performs worst whilst method 8, which more slowly incorporates new observations, has a slower drift and therefore better tracking performance. Method 9, the constant density approach, performed similarly well to 8.

Another metric that can be computed is the RMS error in the tracked speed of the object, again by comparing it to the ground truth provided by the navigation system. The results of this are shown Table 6. The high RMS error for methods 1 and 3 reflect the fact that they do not use a Kalman Filter and so the velocity estimates

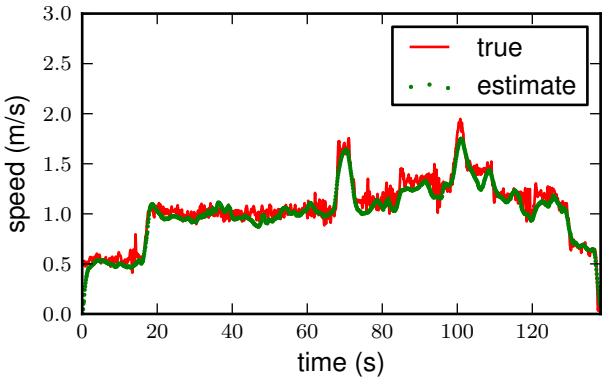


Figure 5: Speed estimate from tracker for experiment A-2

are not smoothed. The plot of tracked versus true speeds for experiment A-2 is shown in Fig. 5 and demonstrates that tracking of centroids alone can provide very good velocity estimates.

The performance of the tracking methods was also tested with respect to the other dynamic objects in the scene. However, the intuitive method developed earlier (comparing the number of track fragments to the number of moving objects) does not necessarily apply in this case. Table 7 demonstrates this problem; the trackers with integrating appearance models (methods 7, 8 and 9) seem to perform better than centroids (1 and 2) since they exhibit less fragmentation. In dataset C, method 7 appears to perform well as it generates only 5 tracks for the 3 moving objects, as opposed to method 2 which instantiates 15 separate tracks for the same data. The 'good' performance of experiment C-7 is, however, due to the fact that the tracker made data association errors and switched from tracking a pedestrian to tracking a tree. This meant that the track was very long and

Table 5: RMS position error (m)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
A	0.98	**0.95**	0.96	3.23	0.96	2.96	1.93	1.11	1.13
B	0.52	**0.51**	0.66	1.18	0.69	1.53	1.41	0.63	0.84
C	0.41	**0.39**	0.47	0.8	0.5	0.74	0.99	0.5	0.48
D	0.37	**0.36**	0.44	1.45	0.53	1.12	0.97	0.59	0.56

RMS position error of tracks corresponding to the target vehicle, with respect to its navigation solution

Table 6: RMS speed error (m/s)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
A	12.46	**0.09**	4.5	141.8	0.11	0.32	0.24	0.09	0.1
B	30.67	**0.07**	300.9	97.97	0.09	0.18	0.18	0.08	0.12
C	11.95	0.11	14.84	8.1	**0.09**	0.17	0.14	**0.09**	**0.09**
D	7.83	**0.09**	0.89	7.98	0.12	0.22	0.15	**0.09**	0.1

RMS speed error of tracks corresponding to the target vehicle, with respect to its navigation solution

stable, but ultimately wrong (as shown in Fig. 6(b)). Whilst for tracks of the target vehicle the RMS position error can be used to ensure that the correct object is being tracked, in the absence of a navigation solution track fragmentation is less informative and an alternate method must be used.

Properly validating tracking performance on arbitrary dynamic objects in the absence of a ground truth position solution relies, therefore, on having hand labelled data associations. Counting the number of times a tracker makes the same data association decision as the hand labelled data, and dividing this by the total number of hand labelled data associations gives a useful metric. The result of this computation is shown in Table 8. The low percentage score in cell C-7 of the table correctly reflects the poor tracking performance of method 7 on this dataset. The higher percentage score in cell C-2 also reflects the better tracking performance of method 2, which can be visually verified in Fig. 6(a).

Table 8 shows that for the datasets tested, centroid tracking outperformed appearance based methods. In the case where the appearance model is built up over time, this can be explained by the fact that incorrect data associations will cause the wrong data to be accumulated into the appearance model. This will then add evidence for the wrong data association in future scans, and may cause the tracker to end up tracking the wrong object; as was demonstrated in Fig. 6(b). For the static appearance model used in method 3, the poor performance is explained by the method's inability to adapt to the changing appearance as a tracked object changes orientation.

The set of experiments demonstrates there are few advantages to appearance based methods over tracking using centroids alone in the context of tracking as evaluated in this paper. However, in some contexts an appearance model may be important not just for tracking, but also for higher level application-specific situational awareness. For example, knowing the size, shape and orientation of a tracked vehicle is important for autonomous driving, which motivates the use of class-specific models in these situations.

Whilst the ability of ICP methods to accumulate an appearance model of an object under track may, over short periods of time, aid data association and object classification, ICP registration has a fundamental problem. Without any ability to 'close the loop' on observations, registration errors from ICP will always accumulate; leading to an appearance model which over time tends towards noise. Even when using a method that discards older data (such as 7), the combination of ICP error and KF estimation error can cause the appearance model to drift with respect to the tracked coordinate frame. Fig. 7 demonstrates the significance of this drift. Small errors in the 6-DOF ICP solution and KF estimation accumulate over the 672 intervening scans (approximately 1 minute of data) and cause the appearance model to move significantly from the tracked centroid. Imposing additional constraints on the object motion, for example that it must remain on the ground and rotate only about its Z-Axis, reduces the orientation drift but cannot eliminate the drift in position. To accumulate a consistent appearance model from 3D data, therefore, other techniques need to be explored.

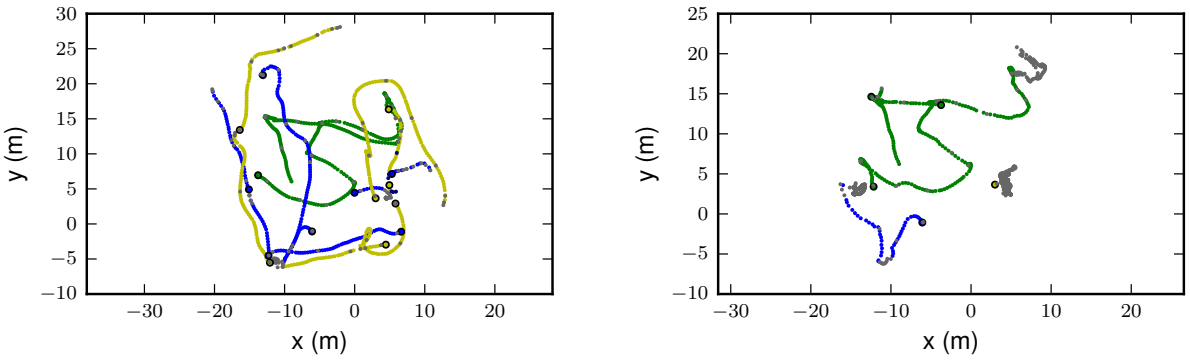
6 Conclusion

This paper analysed a set of methods for tracking multiple targets from a moving vehicle in an unstructured environment using 3D LIDAR data. Each of the methods were tested on four hand labelled datasets, containing multiple dynamic objects including one with an accurate ground truth navigation solution. The results showed that ICP-based tracking methods did not have any significant advantages over (and can in fact be out-

Table 7: Fragmentation of people tracks

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**True**
B	4	4	**3**	11	**3**	15	6	5	**3**	**3**
C	16	15	7	15	12	19	5	6	**3**	**3**
D	10	10	7	10	8	10	7	**6**	7	**6**

Number of fragments which correspond to people.
(final column is number of labelled trajectories)



(a) Tracked trajectories (C-2)

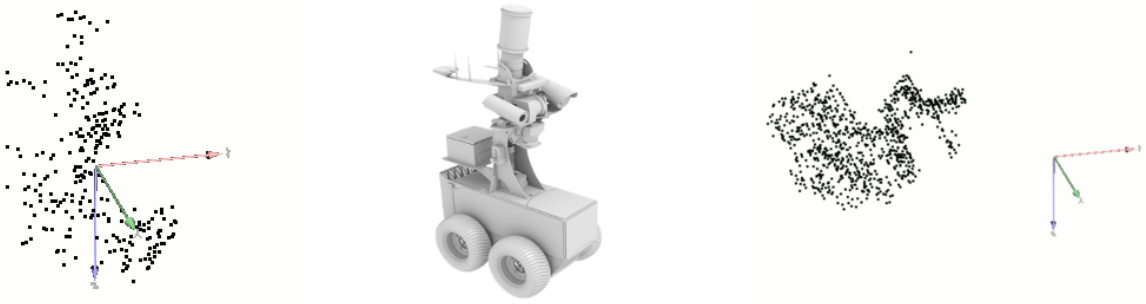
(b) Tracked trajectories (C-7)

Figure 6: People trajectories in dataset C. Track fragments corresponding to the same labelled object were given the same colour (there were three pedestrians in the scene). Black circles indicate where a track fragment begins, and grey dots indicate data associations which are marked invalid with respect to the ground truth. (a) shows the result of experiment C-2, the best performing experiment according to Table 8. (b) shows experiment C-7, the worst performing experiment. The yellow trace is missing in (b), as the tracker confuses the person with a tree and continues to track the tree for the remainder of the experiment.

Table 8: Correct associations (%)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
B	**98.7**	**98.7**	97.7	97.8	96.0	96.7	96.4	96.1	96.3
C	81.2	**86.4**	59.6	44.3	60.8	44.4	30.1	51.1	42.3
D	87.6	**95.3**	38.5	34.1	43.6	47.9	36.2	36.2	24.0

Percentage of correct data associations made by the tracker,
validated against the hand labelled data.



(a) Appearance model after 1 scan

(b) CAD model of “Mantis”

(c) Appearance model after 673 scans

Figure 7: Drift in the appearance model over time using method 7. (a) the appearance model after 1 scan. (b) CAD model of the vehicle from a similar perspective as in (a), to aid in interpretation. (c) shows how the appearance model has shifted with respect to the tracked coordinate frame.

performed by) simply tracking the centroids of each observation.

Additionally, updates to the "mesh-voxel" segmentation method were presented, and the updated method was used to provide the segmented point clouds which were the common input to the tracking methods. The high percentage of laser returns classified as ground across a variety of different scenes motivates explicit ground subtraction as a way to reduce the computational cost of tracking algorithms.

Topics left for future work include using different methods to try to reduce the drift in appearance model due to the accumulation of ICP and estimation errors. Incorporating multi-hypothesis methods to cope with data association ambiguity would likely improve performance, as would the use of tracking frameworks which handle track splitting and merging explicitly and can compensate for segmentation errors (such as the methods suggested in [Shackleton *et al.*, 2010]).

7 Acknowledgements

This research was supported in part by the Centre for Intelligent Mobile Systems (CIMS), funded by BAE Systems as part of an ongoing partnership with the University of Sydney and by the New South Wales State Government.

References

[Arras *et al.*, 2007] K.O. Arras, O.M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3402–3407, 2007.

[Bailey and Nieto, 2008] Tim Bailey and Juan Nieto. Scanslam: Recursive mapping and localisation with arbitrary-shaped landmarks. In *Robotics: Science and Systems Conference (RSS)*, 2008.

[Bar-Shalom *et al.*, 2001] Y. Bar-Shalom, X.R. Li, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. Wiley-Interscience, 2001.

[Bar-Shalom, 1978] Y. Bar-Shalom. Tracking methods in a multitarget environment. *Automatic Control, IEEE Transactions on*, 23(4):618 – 626, August 1978.

[Chen and Medioni, 1991] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2724 –2729 vol.3, April 1991.

[Darms *et al.*, 2008] M. Darms, P. Rybski, and C. Urmson. Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1197 –1202, 4-6 2008.

[Douillard *et al.*, 2011] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3d lidar point clouds. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2011.

[Johnson, 1997] Andrew Eddie Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Carnegie Mellon University, 1997.

[Kirubarajan and Bar-Shalom, 2004] T. Kirubarajan and Y. Bar-Shalom. Probabilistic data association techniques for target tracking in clutter. *Proceedings of the IEEE*, 92(3):536 – 557, March 2004.

[Kuhn, 1955] H.W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[Leonard *et al.*, 2008] John Leonard, Jonathan How, Seth Teller, Mitch Berger, et al. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727 – 774, October 2008.

[Miller *et al.*, 2011] I. Miller, M. Campbell, and D. Huttenlocher. Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes. *Robotics, IEEE Transactions on*, 27(1):29 –46, February 2011.

[Montemerlo *et al.*, 2008] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, et al. Junior: The Stanford entry in the urban challenge. *Journal of Field Robotics*, 2008.

[Moosmann and Fraichard, 2010] Frank Moosmann and Thierry Fraichard. Motion estimation from range images in dynamic outdoor scenes. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 142 –147, 3-7 2010.

[Moosmann *et al.*, 2009] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion. pages 215 –220, 2009.

[Navarro-Serment *et al.*, 2010] Luis E. Navarro-Serment, Christoph Mertz, and Martial Hebert. Pedestrian detection and tracking using three-dimensional ladar data. *The International Journal of Robotics Research*, 29:1516–1528, 2010.

[Petrovskaya and Thrun, 2009] Anna Petrovskaya and Sebastian Thrun. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26:123–139, 2009.

[Shackleton *et al.*, 2010] John Shackleton, Brian VanVoorst, and Joel Hesch. Tracking people with a 360-degree lidar. *Advanced Video and Signal Based Surveillance, IEEE Conference on*, 0:420–426, 2010.

[Spinello *et al.*, 2010] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart. A layered approach to people detection in 3d range data. In *Proc. of The AAAI Conference on Artificial Intelligence: Physically Grounded AI Track (AAAI)*, 2010.

[Spinello *et al.*, 2011] L. Spinello, M. Luber, and K. O. Arras. Tracking people in 3d using a bottom-up top-down people detector. In *Proc. of The International Conference in Robotics and Automation (ICRA)*, 2011.

[Underwood *et al.*, 2010] J.P. Underwood, A. Hill, T. Peynot, and S.J. Scheding. Error modeling and calibration of exteroceptive sensors for accurate mapping applications. *Journal of Field Robotics*, 27(1):2–20, 2010.

[Wang, 2004] Chieh-Chih Wang. *Simultaneous Localisation, Mapping and Moving Object Tracking*. PhD thesis, Carnegie Mellon University, 2004.