What *Could* Move? Finding Cars, Pedestrians and Bicyclists in 3D Laser Data

Dominic Zeng Wang and Ingmar Posner and Paul Newman

***Abstract*— This paper tackles the problem of segmenting things that *could* move from 3D laser scans of urban scenes. In particular, we wish to detect instances of classes of interest in autonomous driving applications - cars, pedestrians and bicyclists - amongst significant background clutter. Our aim is to provide the layout of an end-to-end pipeline which, when fed by a raw stream of 3D data, produces distinct groups of points which can be fed to downstream classifiers for categorisation. We postulate that, for the specific classes considered in this work, solving a binary classification task (i.e. separating the data into foreground and background first) outperforms approaches that tackle the multi-class problem directly. This is confirmed using custom and third-party datasets gathered of urban street scenes. While our system is agnostic to the specific clustering algorithm deployed we explore the use of a Euclidean Minimum Spanning Tree for an end-to-end segmentation pipeline and devise a RANSAC-based edge selection criterion.**

I. Introduction

In this paper, we present an end-to-end system that detects instances of cars, pedestrians and bicyclists from a raw stream of 3D laser data for autonomous driving applications.

Autonomous driving has become a prominent application domain for robotics research. This is witnessed by a cornucopia of publications in this area [1], [2], [3]. The success of the DARPA Grand- [4] and Urban Challenges [5] as well as Google's endeavour to promote autonomous driving for data gathering purposes [6] has heightened expectations that autonomous cars will be able to operate in environments of realistic complexity. Our community's aspiration to create self-driving cars has further served to highlight the importance of - and to focus efforts within - environment understanding.

Much research effort is being expended on the detection and classification of objects pertinent to navigating a realistic road environment, both using vision and laser data. Of particular interest are potentially dynamic objects - that is objects which *could* move - since their presence and potential change of state will influence the planning of actions and trajectories. The work presented here also falls within this category. In particular, we restrict ourselves to the detection of *cars*, *pedestrians* and *bicyclists* in a stream of 3D laser data obtained from sensors commonly deployed on autonomous vehicles based on shape information in a per-frame basis. A shape based approach is taken because a *potentially* moving object may not be *actually* moving.

In focusing on *detection* our work immediately differentiates itself from a significant body of work targeted at the *classification* of these objects. In fact, the latter often assume (explicitly or otherwise) that a suitable segmentation of the 3D point cloud into *complete entities* of interest is already available [7] or is straight-forward to obtain [8]. However, obtaining such a segmentation is widely acknowledged to be a hard problem [7], [8], [9] since the number of objects in the scene is commonly not known and only a small proportion of the data contain relevant class information. This motivates the work presented here.

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

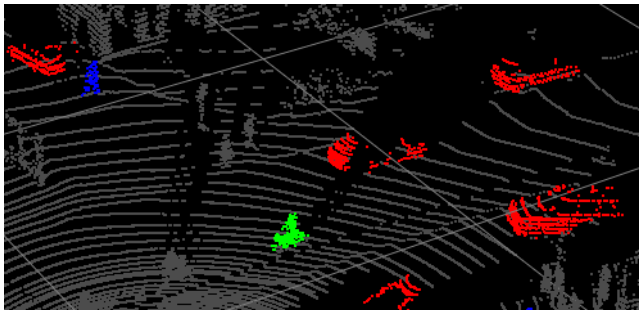

Fig. 1. Sample output from one of the proposed foreground/background schemes (the *F/B binary* scheme, see section IV-D). The detected cars, pedestrians and bicyclists are coloured in red, blue and green respectively, with background showing in grey. This figure is best viewed in colour.

The objective of this work is to group salient subsets of the raw data stream into contiguous and *complete* entities corresponding to objects of interest without prior knowledge of the number and location of the objects present. Since we have knowledge of the object classes of interest (and the set of them is comparatively small) we employ a supervised approach. We investigate the application of graph based techniques to the problem, and establish that, for the specific classes considered in this work, solving a binary classification task (i.e. separating the data into foreground and background first) outperforms approaches that tackle the multi-class problem directly. We consider this to be the primary contribution of this paper. Further, in order to provide the layout for an end-to-end pipeline, we demonstrate the use of a particular graph-based clustering algorithm as a back-end to our segmentation approach (see Fig. 1 for a typical example of the output of the end-to-end system).

After a survey of related works in the next section, we introduce the graph-based clustering algorithm used in this work in Section III. A number of schemes for the extraction of foreground data from a stream of raw 3D laser points are detailed in Section IV. We evaluate these schemes in Section V and conclude in Section VI.

II. Related Works

Existing works on object detection and recognition in 3D laser data can be coarsely divided into three categories.

The first commonly assumes that point clouds representing *entire* objects have already been segmented out of the data and, therefore, focus mainly on classification. Examples include Teichman et al. [7], [10], who classify complete tracks of segmented objects into one of the classes *car*, *pedestrian*, *bicyclist* or *background*. Lai et al. [8] combine advantages of both shape and appearance with a Kinect-style sensor to classify indoor objects using sparse distance metric learning with a Group-Lasso regulariser. In this case the segmentation task is facilitated by the controlled environment the objects are placed in. Endres et al. [9] on the other hand take an unsupervised approach to discover object categories in the presented segments using Latent Dirichlet Allocation (LDA).

The segmentation of desirable objects from amongst an often large amount of background clutter in 3D laser data is a pivotal precursor to such systems. Existing works include that by Douillard et al. [11], where the existence of a ground plane is assumed and object segments are derived in an unsupervised fashion from non-ground data only. Klasing et al. [12] perform clustering based on Euclidean distance between individual laser points, implicitly assuming that objects are not connected by scene clutter.

The second class of methods label a scene directly into regions belonging to object classes (with possibly a *background* class), but do not distinguish separate object instances. Anguelov et al. [13], for example, take a supervised approach based on a Markov Random Field (MRF) using local features computed at individual data points to produce globally consistent labels. Triebel et al. [14] employ an approach based on Conditional Random Fields (CRFs) constructed in both feature space and Euclidean space to obtain a scene segmentation into object categories that often correspond to repeated patterns.

A third class of approaches focus on a *targeted* segmentation of the data. Here the class of interest is known and a segmentation scheme is devised which accommodates this specifically. An example is work by Spinello et al. [15], which concentrates expressly on the detection of pedestrians.

The approach we present here populates the space between the second and third categories above. While we also have prior knowledge of the classes of interest our work aims to cater for a *range* of categories (i.e. *cars*, *pedestrians* and *bicyclists*), thus sacrificing the benefit of a relatively narrow segmentation problem. To achieve this we leverage the same pre-segmentation algorithm and the same descriptors as are used in [14]. However, in contrast to [14], our supervised approach produces object clusters which correspond only to object categories of interest. In addition, the output of our system distinguishes between object instances rather than partitioning the scene into regions generally belonging to object classes.

Finally, we mention a related body of work catering for the detection of *instantaneously* dynamic objects, i.e. objects that are moving at the time of detection (see, for example, Katz et al. [16] or Yang and Wang [17]). In contrast to these works the problem addressed here includes the detection and classification of entities that *could* move but may not be moving at the time data are recorded.

III. Graph-Based Clustering

Oftentimes an unknown number of objects of interest exist in a single scene. A successful categorisation of these objects requires the ability to distinguish between separate object instances within the data stream, even under clutter-free conditions (e.g. after removal of the background, see Section IV). In this section we formulate this problem as a clustering task.

Unsupervised data clustering has been an active area of research for decades and many methods exist which circumvent the lack of prior information, such as the number of clusters present. Variational Bayesian methods [18], for example, provide an attractive mechanism but are often plagued with convergence issues. Jenssen et al. [19] use an information theoretic measure for model selection to determine the optimal number of clusters from amongst various possibilities.

Another popular approach is graph-based clustering using a Euclidean Minimum Spanning Tree (EMST) constructed from the data [20]. EMST-based techniques made their appearance in the literature as early as the 1970's [21] and are often used when cluster boundaries are expected to be irregular. Given a finite point set $\mathcal{P} \subset \mathbb{R}^d$ EMST-based algorithms first compute the Minimum Spanning Tree over the complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{i : p_i \in \mathcal{P}\}$ and $\mathcal{E} = \{\{i, j\} : p_i \in \mathcal{P}, p_j \in \mathcal{P}, i \neq j\}$, with edge weights given by the pairwise Euclidean distances. Edge statistics gathered throughout the tree are used to determine where to break the linkage. For example, it was shown in [22] that, by removing the $K - 1$ longest edges in the EMST, a clustering is obtained which maximises the minimum inter-cluster distance in the space of all possible disjoint partitions of the point set into K groups. When K is unknown, as in our case, heuristics are used to determine which edges to remove.

For example, Zahn [21] defines inconsistency measures using local statistics of the edge weights in the MST and removes edges which violate any one of a set of consistency criteria. Grygorash et al. [23] propose a sequence of edge removal operations such that the standard deviation of the edge weights is minimised. The optimal number of clusters is found when a (local) minimum has been reached.

Our approach also leverages an EMST. In particular, we observe that, as a result of the formation process of the spanning tree, edges connecting points of the same object instance tend to be of similar lengths (up to sensor noise) corresponding to the sample width of the sensor. Edges linking individual object instances, on the other hand, tend to be comparatively long. We exploit this observation by using the RANSAC paradigm [24] to estimate outliers amongst the edge weights. The spanning tree is broken wherever

an outlier is found. To illustrate, consider Fig. 2, where clustering is performed on a synthetic scene containing a car, two pedestrians and a bicyclist (all examples from a real dataset). In subsequent sections we refer to this clustering algorithm as the EMST-RANSAC algorithm. It can be used to segment a point cloud into multiple entities without prior knowledge of the number of objects contained in the scene. However, since the algorithm clusters data based on Euclidean distance, unwanted points belonging to background clutter (i.e. anything other than the object classes of interest) must be removed before the algorithm can be applied. In the following section we formulate this removal of background clutter as a supervised classification task.

IV. Foreground Extraction

Consider a set of points $\mathcal{P} \subset \mathbb{R}^3$ generated by a 3D laser scanner. In order to apply the EMST-RANSAC algorithm described in the previous section we require to split $\mathcal{P}$ into the set of *foreground* data, $\mathcal{P}_f \subseteq \mathcal{P}$ - i.e. those belonging to object classes of interest - and its complement, the set of *background* data $\mathcal{P}_b = \mathcal{P} - \mathcal{P}_f$.

We employ a bottom-up approach, starting by preprocessing the point cloud to obtain an *over-segmentation* in the form of a set of point-cloud patches. While we do not require the segmentation to be perfect, it is necessary for each segment to span no more than a single class of interest. Features are then extracted for each patch. This representation is used for classification of each patch as to its membership of $\mathcal{P}_f$.

A. Preprocessing

In common with other works we perform an off-the-shelf pre-segmentation step based on point normal estimates in the point cloud to obtain a set of super-voxels as atomic inputs for our entity segmentation approach (see Section IV-B). To obtain reliable normal estimates for input to the pre-segmentation algorithm, we follow a popular approach for normal computation, which finds the local set of nearest neighbours within a search radius r for each datum $\mathbf{p}_i$ and then, assuming local planarity, performs PCA on it. The eigenvector corresponding to the least eigenvalue is taken to be the estimated normal direction, thus essentially performing a least-square plane fit to the neighbourhood.

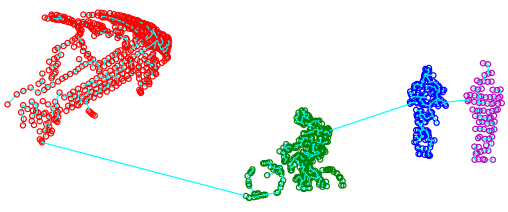

Fig. 2. The output of the EMST-RANSAC clustering algorithm when applied to a synthetic scene containing fours objects of interest: a car, two pedestrians and a bicyclist (all examples from real data). Different colours denote different clusters produced. Cyan line segments show edges in the EMST. This figure is best viewed in colour.

This approach has been shown empirically to perform best in terms of the trade-off between robustness and computation overhead [25].

The edge set needed for the super-voxel segmentation is given by N nearest neighbour linkage. That is

$$\mathcal{E} = \{\{i, j\} : \mathbf{p}_i \in \mathcal{P},\, \mathbf{p}_j \in \mathcal{N}_i^N\}\ , \tag{1}$$

where $\mathcal{N}_i^N$ denotes the set of N nearest neighbours of the point $\mathbf{p}_i$, *excluding* the point itself.

B. Patch Segmentation

To obtain the initial patch segmentation, we follow the approach proposed by Triebel et al. [14] who adapted the popular segmentation algorithm introduced by Felzenszwalb and Huttenlocher [26] to operate on normal estimates for points in $\mathcal{P}$. The algorithm operates on an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with the edge weights representing a dissimilarity measure between adjacent points. Starting with each vertex $i \in \mathcal{V}$ as a single segment, the algorithm traverses the edges in order of non-decreasing weight, merging the adjacent segments when there is no evidence of a boundary. Consider the set of vertices $\mathcal{V} = \{i : \mathbf{p}_i \in \mathcal{P}\}$ and the set of edges $\mathcal{E}$ as in (1). The dissimilarity measure is defined by

$$w(i, j) = 1 - |\mathbf{n}_i \cdot \mathbf{n}_j|\ , \tag{2}$$

where $\mathbf{n}_i$ denotes the normal estimated at point $\mathbf{p}_i$. Intersections between smooth surfaces will thus give rise to segmentation boundaries.

C. Feature Extraction

For each patch a fixed-dimensional feature vector is constructed by concatenating five sets of common invariant descriptors. The descriptors consist of 50-dimensional spin images [27] computed at the centroid and about the vertical Z axis, 32-dimensional shape distributions [28] using pairwise Euclidean distances as shape function, 32-dimensional shape distributions using magnitude of dot products between normals at point pairs as shape function, three-dimensional shape factors [29], and the three dimensions of the bounding box along PCA directions. These give rise to a 120-dimensional feature vector.

D. Patch Classification

For foreground-background separation any of a number of classification frameworks can be employed. The foreground class simply constitutes the union of the *car*, *pedestrian* and *bicyclist* classes. We propose two schemes of merging these three foreground classes in the patch classification stage to produce a clean foreground-background segmentation of the scene so that the EMST-RANSAC algorithm is applicable.

F/B binary: in this scheme the three foreground classes are pooled into a single class and a binary classifier is trained to separate them.

F/B N-class: here, N one-versus-all binary classifiers are employed for the *car*, *pedestrian*, *bicyclist* and *background* classes, respectively. After classification is performed the outputs for the three foreground classes are merged into a single set.

Fig. 3. The Bowler Wildcat research mobile platform, equipped (on top) with a Velodyne HDL-64E S2 sensor.

In the next section we show the effectiveness of these two schemes in terms of both patch classification and overall performance at the object detection level. The two schemes are also benchmarked against a third, **N-class**, where the individual foreground classes are treated separately up to, and including, the object detection level.

V. Experimental Results

We evaluate our segmentation approach using both a publicly available dataset as well as data gathered using our own autonomous vehicle. In particular, we make use of the Stanford Track Collection (STC) dataset released to the public with [7]. The STC contains a significant number of labelled objects of interest (cars, pedestrians and bicyclists) and has the added advantage of being gathered using the same sensor as deployed on our car. However, the dataset was originally produced for the task of track classification and therefore contains only instances of trackable objects. Scene clutter is especially underrepresented (see Table I). For this work we therefore augment the STC with data gathered using our Bowler Wildcat research platform (Fig. 3) equipped with a Velodyne HDL-64E SE2 laser range finder.

A. Patch Classification

The performance of the three patch classification schemes introduced in Section IV-D was evaluated using the data detailed in Table I. Our approach is agnostic to the patch segmentation scheme employed as long as it produces an *over*-segmentation of the data with respect to the classes of interest. The parameters for the patch segmentation algorithm used here (see Section IV-B) were determined empirically based on a qualitative evaluation of performance on a small number of scenes. For classifier training and evaluation, 70% of the data were selected at random to form the training set. The remainder were used as a hold-out set for classifier evaluation. For classification we employ Support Vector Machine (SVM) classifiers with the non-linear Radial Basis Function (RBF) kernel. Parameters are trained using five-fold cross-validation. For scheme *F/B binary* a single binary SVM classifier is trained for the *foreground* and *background* classes. For schemes *F/B* N*-class* and N*-class*, four individual SVM classifiers are trained in a one-versus-all configuration for each of the *car*, *pedestrian*, *bicyclist* and *background* classes. Final class decisions are made greedily such that the winner takes all. Fig. 4 shows precision-recall curves generated for the three schemes using the held-out data. It is evident from Fig. 4 that the binary *foreground/background* separation in this particular case presents an easier task for the classifiers than separating the data into the individual classes *car*, *pedestrian*, *bicyclist* and *background*. When combining the output of the classifiers for the three individual foreground classes into a single class for the *F/B* N*-class* scheme the performance is almost identical to that of the binary *foreground/background* classifier of the *F/B binary* scheme. Note also that, by definition, the performance of the background-versus-all classifier of the scheme N*-class* is identical to that of the binary *foreground/background* classifier evaluated for the *background* class. This indicates that the separate classification of the *car*, *pedestrian*, *bicyclist* classes introduces significant confusion amongst only these classes which is remedied by collating them into a single *foreground* class. Further evidence of this can be found in the confusion matrices for the N*-class* scheme depicted in Fig. 5. These imply that the biggest confusion between *foreground* and *background* is caused by *background* data being mistakenly classified as *car*. On the other hand, significant confusion exists amongst the individual *foreground* classes. These results indicate that, for the task of separating *car*, *pedestrian* and *bicyclist* from *background* in 3D laser data, the predominantly shape-based features employed here are not sufficient. This lends further support to the intuitive notion that over-segmented patches do not carry enough shape information to be classified correctly. Formulating the task as a binary classification problem, on the other hand,

TABLE I

Details of the Evaluation Set Composition for Patch Classification in Units of Patches Obtained from the Initial Pre-Segementation. Columns denote the source of the data.

	STC	Wildcat	Total
Car	13998	580	14578
Pedestrian	4619	0	4619
Bicyclist	4601	2	4603
Background	10000	30500	40500
Total	33218	31082	64300

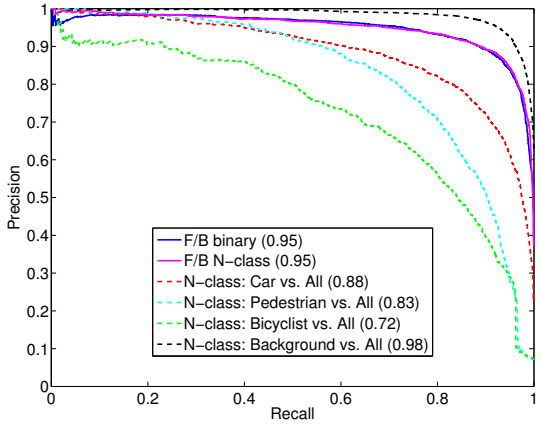

Fig. 4. Precision-recall curves for schemes *F/B binary*, *F/B* N*-class* and N*-class*. For the latter, individual curves are shown for each one-versus-all classifier. Numbers in brackets represent the area under the curve (AUC).

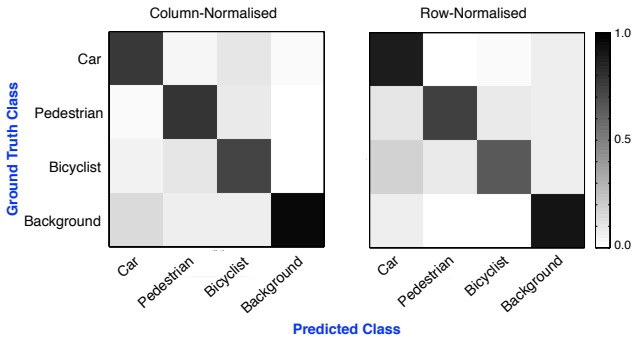

Fig. 5. Confusion matrices for the N-*class* scheme normalised to precision (left) and recall (right) along the diagonal.

remedies this problem as the *foreground* and *background* classes appear much more amenable to separation when characterised by shape features.

B. Overall System Evaluation

In this section we evaluate the performance of the overall system starting with a raw data stream as input and performing preprocessing, patch classification and EMST-RANSAC clustering. However, in order to demonstrate the efficacy of foreground/background-based schemes against the N-*class* scheme, the clusters obtained have to be classified into one of the three foreground object classes. For this purpose we trained a N-class SVM classifier using the same set of features listed in section IV-C, now computed over *entire entity clusters* (as opposed to patches representing object *parts*) returned by the *F/B binary* scheme. For the *F/B* N-*class* scheme, since it uses an N-class classifier in an intermediate stage, we retain the prediction scores from patch classification and determine the class of an entity cluster by a majority vote amongst the constituent laser points. The votes are weighted by prediction confidence. For the N-*class* scheme, detections are obtained by running the EMST-RANSAC algorithm over the three foreground regions independently and entity cluster classes are self-evident. However, we stress that classification is not the focus of this paper. More sophisticated classification approaches exist (see, for example, [7]) .

For the purposes of this evaluation we collate data points returned within a full 360° rotation into a single scene (frame) for processing. The EMST-RANSAC algorithm involves only a single parameter: the inlier support width w to evaluate hypotheses [24]. This parameter was trained for on 200 frames extracted from the STC dataset that are disjoint from those used in producing the training data for patch classification. For the *F/B binary* and *F/B* N-*class* schemes, a single value for w was determined since the EMST-RANSAC algorithm is applied only once on patches belonging to the *foreground* class. For the N-*class* scheme, EMST-RANSAC is applied to each of the object classes, resulting in a three-element vector $\mathbf{w}$. We trained the three support widths independently for the N-*class* scheme.

To evaluate the performance of the system, we hand-labelled 100 randomly chosen frames from a busy urban scene taken at a local town centre. These data are entirely independent from those used during any of the training phases. A set of qualitative results on a sample frame for the *F/B* N-*class* scheme is shown in Fig. 6. A similar result for the *F/B binary* scheme is shown in Fig. 1. For quantitative analysis we adopt evaluation metrics derived from those used in a popular object detection challenge in the vision community, the PASCAL Visual Object Classes Challenge [30]. In particular, a detection is marked as correct if it overlaps with a ground-truth annotation more than 50%. The measure of overlap is computed as

$$a_o = \frac{|\mathcal{C}_\mathrm{p} \cap \mathcal{C}_\mathrm{gt}|}{|\mathcal{C}_\mathrm{p} \cup \mathcal{C}_\mathrm{gt}|}, \tag{3}$$

where $\mathcal{C}_\mathrm{p}$ and $\mathcal{C}_\mathrm{gt}$ denote sets of points belonging to the predicted and ground-truth object clusters respectively. Each detection is assigned to at most one ground-truth object, and multiple detections are treated as false positives. Table II thus lists evaluation results for each of the foreground object classes obtained on the 100 evaluation frames containing, in total, 818 cars, 899 pedestrians and 39 bicyclists. The F_1-measures indicate that the schemes based on a foreground-background formulation of the problem outperform the N-*class* scheme where the classes are treated differently from the patch level. However, the latter still does surprisingly well considering the findings in the patch classification evaluation. This observation can be explained by the two extra degrees of freedom found in the EMST-RANSAC algorithm for this scheme. For example, the system has the freedom to learn that instances of people tend to be closer together than instances of cars. The comparatively poor result on the *bicyclist* class (especially the precision) can be attributed to the low number of class instances present in the test data, thus causing a notable effect of false positives from the patch classification. Performance differences between the *F/B binary* and *F/B* N-*class* schemes are due to the difference in entity classification schemes.

C. Timing

A prototype of our system has been implemented in C++ and Matlab and was deployed on a vanilla MacBook Pro equipped with a dual core Intel i5 processor (2.4GHz) with 4GB of RAM. For point normal estimation our implementation takes advantage of the facilities provided in the Point Cloud Library (PCL) [31]. SVM training and prediction were carried out using LibSVM [32]. For efficient EMST computation, we implemented the fast EMST algorithm proposed by March et al. [33]. Based on measurements from our 100 evaluation frames (containing of the order of 100,000 points per frame), the per-frame run-time is currently dominated by the EMST-RANSAC clustering step (3.3s) and the normal computation (1.7s). We are currently investigating various options to achieve real-time performance with our next implementation, which will be deployed on our research platform.

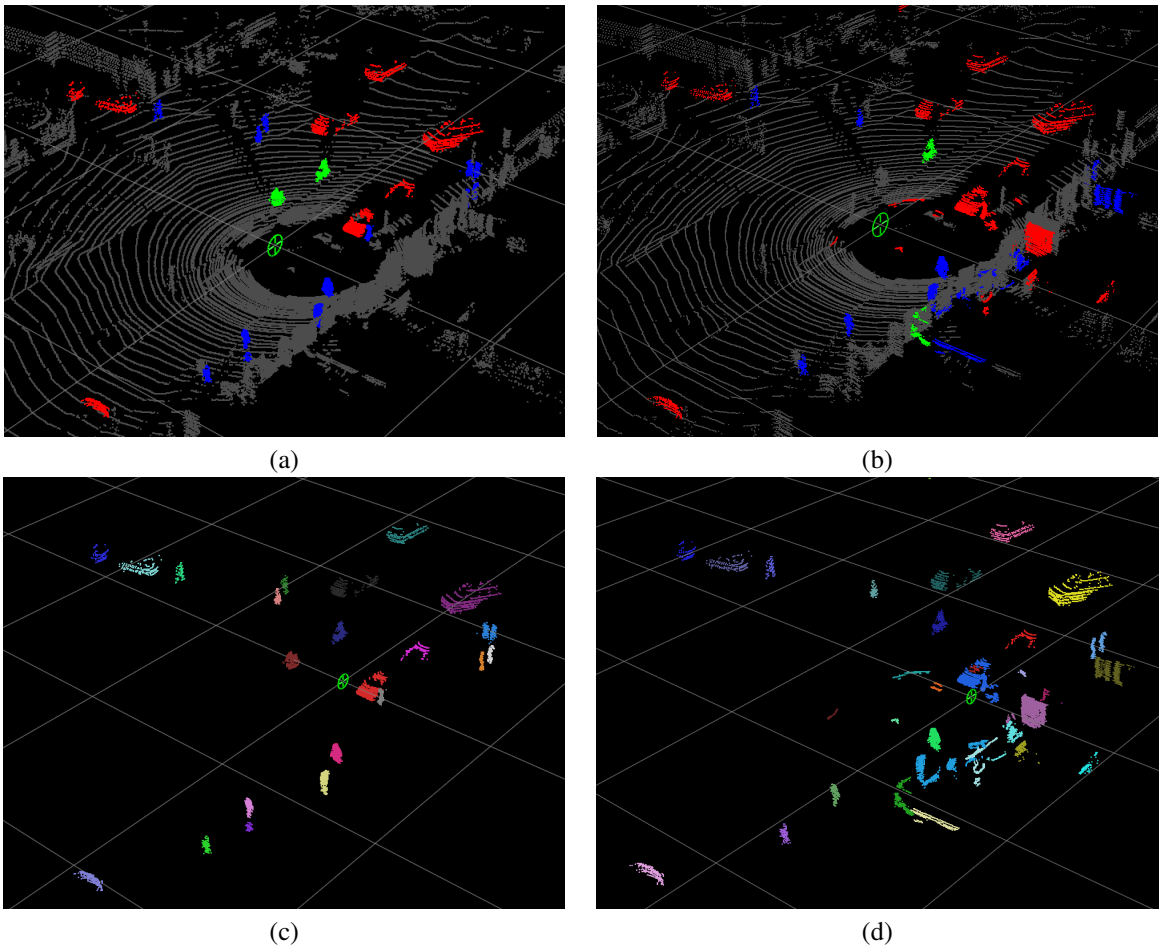

Fig. 6. Sample scene frame showing results of the *F/B N-class* scheme. (a) Ground truth scene labels. (b) Objects detected by *F/B N-class*. (c) Ground truth objects in the scene. (d) Object clusters produced by *F/B N-class*. In both (a) and (b), regions coloured red, blue, green and grey correspond to points belonging to the *car*, *pedestrian*, *bicyclist* and *background* classes, respectively. In both (c) and (d), different colours denote different object instances, with colours chosen at random. This figure is best viewed in colour.

TABLE II

SYSTEM EVALUATION RESULTS BY CLASS. P, R AND F_1 STAND FOR PRECISION, RECALL AND THE F_1-MEASURE, RESPECTIVELY. NUMBERS IN BRACKETS DENOTE THE GROUND-TRUTH NUMBER OF OBJECTS OF A GIVEN CLASS IN THE EVALUATION DATA.

		Car (818)			Pedestrian (899)			Bicyclist (39)		
		P	R	F_1	P	R	F_1	P	R	F_1
N-class		0.1856	**0.5122**	0.2725	0.4415	**0.5751**	0.4995	0.0294	0.4359	0.0551
F/B	binary	**0.2795**	0.4401	**0.3419**	0.4696	0.3782	0.4190	0.0256	0.4103	0.0483
	N-class	0.2102	0.5037	0.2966	**0.5877**	0.4360	**0.5006**	**0.0989**	**0.4615**	**0.1629**

VI. CONCLUSION AND FURTHER WORK

This paper presents an approach to segmenting objects of interest from a raw data stream as commonly obtained from a 3D laser range finder. In particular, we consider the domain of autonomous driving and focus on the supervised extraction of potentially dynamic objects such as cars, pedestrians and bicyclists. The output of the system are clusters of points representing *entire* objects, which is often assumed to be available by work on object classification in 3D point clouds. We show that, for the specific classes considered, solving a binary classification task (i.e. separating the data into foreground and background first) outperforms approaches that tackle the multi-class problem directly. This is primarily the case because *parts* of objects, as commonly obtained by a pre-segmentation step, do not contain enough shape information to be robustly categorised as belonging to the classes considered here. While our pipeline is agnostic to the graph-based clustering algorithm used we explore the use the EMST algorithm, and extend it by a RANSAC-based outlier rejection step to automatically determine the number of clusters present in a scene. In doing so, we explicitly exploit the sampling characteristics of the laser range finder. While the results on patch classification presented here are particular to the popular *car*, *pedestrian*, *bicyclist* and *background* classes, the EMST-RANSAC approach is agnostic to the choice of classes and, therefore, more generally applicable.

Our prototype pipeline produces promising results and

harbours potential for real-time performance. However the current frame-based detection framework would clearly benefit from introducing tracking information for the detected objects. The dynamic nature of our classes of interest also suggests that system performance could benefit from the inclusion of motion cues. This will be the focus of future research.

VII. Acknowledgements

This work was supported by the Clarendon Fund. Paul Newman was supported by an EPSRC Leadership Fellowship, EPSRC Grant EP/I005021/1. The authors would like to thank Rudolph Triebel for many helpful discussions.

References

[1] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 4889 –4895.

[2] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[3] A. Huang and S. Teller, "Probabilistic Lane Estimation using Basis Curves," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[4] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[5] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. R. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[6] N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 5421 –5426.

[7] A. Teichman, J. Levinson, and S. Thrun, "Towards 3D object recognition via classification of arbitrary object tracks," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 4034 –4041.

[8] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining RGB and depth information," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 4007 –4013.

[9] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard, "Unsupervised discovery of object classes from range data using latent Dirichlet allocation," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[10] A. Teichman and S. Thrun, "Tracking-Based Semi-Supervised Learning," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[11] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D LIDAR point clouds," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 2798 –2805.

[12] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3D laser data," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, may 2008, pp. 4043 –4048.

[13] D. Anguelov, B. Taskarf, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of Markov random fields for segmentation of 3D scan data," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, june 2005, pp. 169 – 176 vol. 2.

[14] R. Triebel, J. Shin, and R. Siegwart, "Segmentation and Unsupervised Part-based Discovery of Repetitive Objects," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[15] 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.

[16] R. Katz, J. Nieto, and E. Nebot, "Unsupervised classification of dynamic obstacles in urban environments," *Journal of Field Robotics*, vol. 27, no. 4, pp. 450–472, 2010.

[17] S.-W. Yang and C.-C. Wang, "Simultaneous egomotion estimation, segmentation, and moving object detection," *Journal of Field Robotics*, vol. 28, no. 4, pp. 565–588, 2011.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[19] R. Jenssen, I. Hild, K.E., D. Erdogmus, J. Principe, and T. Eltoft, "Clustering using Renyi's entropy," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 1, july 2003, pp. 523 – 528 vol.1.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[21] C. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *Computers, IEEE Transactions on*, vol. C-20, no. 1, pp. 68 – 86, jan. 1971.

[22] T. Asano, B. Bhattacharya, M. Keil, and F. Yao, "Clustering algorithms based on minimum and maximum spanning trees," in *Proceedings of the fourth annual symposium on Computational geometry*, ser. SCG '88. New York, NY, USA: ACM, 1988, pp. 252–257.

[23] O. Grygorash, Y. Zhou, and Z. Jorgensen, "Minimum Spanning Tree Based Clustering Algorithms," in *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference on*, nov. 2006, pp. 73 –81.

[24] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981.

[25] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, "Comparison of surface normal estimation methods for range sensing applications," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 3206 –3211.

[26] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *Int. J. Comput. Vision*, vol. 59, pp. 167–181, September 2004.

[27] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 5, pp. 433 –449, may 1999.

[28] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, pp. 807–832, October 2002.

[29] C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz, "Geometrical Diffusion Measures for MRI from Tensor Basis Analysis," in *ISMRM '97*, Vancouver Canada, April 1997, p. 1742.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[31] R. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 1 –4.

[32] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, May 2011.

[33] W. B. March, P. Ram, and A. G. Gray, "Fast euclidean minimum spanning tree: algorithm, analysis, and applications," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 603–612.