LIDARBOX: A FAST AND ACCURATE METHOD FOR OBJECT PROPOSALS VIA LIDAR POINT CLOUDS FOR AUTONOMOUS VEHICLES

Haziq Razali and Nizar Ouarti

Image and Pervasive Access Lab (Sorbonne UPMC, CNRS, NUS, A*STAR)

ABSTRACT

The use of LIDAR technology for autonomous navigation has gained prominent status over the recent years due to its sensing capability. In this paper, we present an object proposal pipeline for LIDAR point clouds catered towards the domain of autonomous vehicles. Our method is built on the assumption that the distribution of point clouds (in the nearest neighbours sense) with respect to its depth can be modelled by a linear function with added gaussian noise. Combined with our occlusion handling algorithm, we show the efficacy of our object proposal pipeline on the KITTI dataset with extremely competitive results.

***Index Terms*—** Object Proposal, LIDAR, Point-Clouds, Autonomous Vehicles

1. INTRODUCTION

Light Detection and Ranging (LIDAR) sensors, once prohibitively expensive, have rapidly dropped in price. If this trend were to continue, it seems inevitable that LIDAR sensors will eventually become as ubiquitous as cameras in the autonomous domain. Although they have the same purpose as stereo cameras - to serve as an active warning system - LIDAR sensors have several advantages over their stereo-vision-based counterparts. (i) Since these sensors make use of emitted light for range sensing, it works independently of the ambient light; night or day, it pretty much sees the same. (ii) Time-of-flight technology is more reliable than the traditional stereo-matching and reconstruction paradigm of stereo-imaging based systems whose output is dependent on the algorithm in use.

Despite these attractive features, the use of LIDAR sensors are still underexploited by the vision community [1, 2, 3, 4, 5, 6, 7] mainly due to the absence of color information and sparsity compared to stereo-imaging based systems. In this study however, we show that this sparsity can be mitigated by making an assumption that the distribution of LIDAR point clouds in outdoor scenes - in the nearest neighbor sense - fits a linear function with added gaussian noise. We focus our attention to the task of object proposals and ask whether a system based only on LIDAR can compete with conventional camera systems in the context of outdoor objects detection.

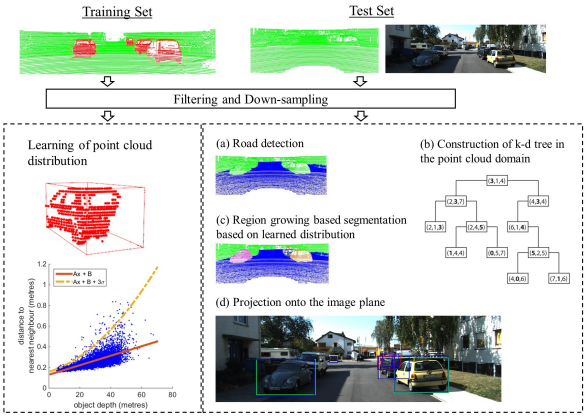

Fig. 1: The workflow of our object proposal algorithm.

In this paper, we propose a computationally efficient approach to generate object proposals at real-time by exploiting the distribution of point clouds as well as prior knowledge specific to the domain of autonomous driving. Our method plays in the same league as current state-of-the-art methods but is at least an order of magnitude faster. We believe our method is the first to take into account the relationship between depth and sparsity.

2. RELATED WORKS

The last few years have seen the vision community shifting from the exhaustive and computationally expensive sliding window paradigm to the faster region proposals framework. The goal of generating proposals is to create a relatively small set of candidate bounding boxes among which at least a few accurately cover the objects of interest. While there exists a variety of works on 2D [8, 9, 10, 11, 12] and 3D [13, 14, 15, 16] object proposals, we will omit a detailed discussion and would instead encourage readers to refer to [17] for a comprehensive review on object proposals. Here, we briefly review works that will be used as a baseline for comparison.

Different strategies have been investigated in the realm of 2D object proposals. In the Selective Search (SS) algorithm [8], pixels analyzed in different colour domains (HSV, Lab, Grayscale, etc.) are iteratively assigned into separate entities

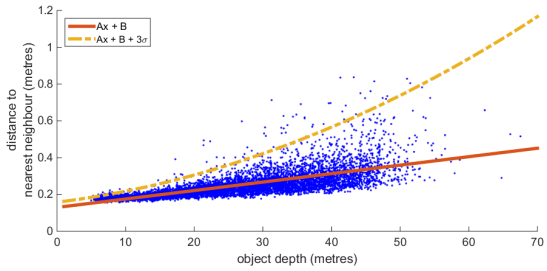

Fig. 2: Point cloud distribution downsampled at $c = 0.2$.

Fig. 3: Generating additional bounding boxes. **Top:** Given the initial blue box that corresponds to the white vehicle behind the black SUV, the decision on whether to generate additional boxes to the left or right depends on the signed difference between the 2 yellow rectangles. **Bottom:** The resulting bounding box with the ground truth in red. Note that the depth image has been manually altered for ease of visualization as LIDAR point clouds are sparse.

based on a set of similarity measures between neighbours. The Multiscale Combinatorial Grouping (MCG) framework in [9] partitions each hierarchy of an image pyramid into distinct regions via the normalized cuts algorithm [18] before producing a ranked list of proposals via combinatorial grouping. Edge Boxes (EB) [10] rapidly score millions of windows based on contour information inside and on the boundary of each window. Their observation lies on the fact that the number of edges wholly enclosed by a bounding box is indicative of the likelihood of the box containing an object.

Approaches that exploit depth information include the MCG-Depth (MCG-D) [13] that extends the MCG framework of [9] by incorporating an additional set of depth features. Most closely related to our work is that of [14] where proposals are generated by minimizing an energy function that encodes information such as object size prior, distance above ground plane as well as several depth informed features. The main difference in our approach is the use of a model for the clustering of LIDAR point clouds, leading to a faster implementation.

3. 3D OBJECT PROPOSALS

In this section, we delineate the workflow of our algorithm (Figure 1). Starting from the 3D point clouds, these point clouds are fed through a sequence of modules until the very end where proposals are projected back onto the image plane. We begin with the task of pre-processing.

3.1. Pre-processing

Field-of-view filtering: In the KITTI object detection benchmark, images are captured from the front camera and range scans perceive a 360° view of the environment. Since the ground-truth is only provided for objects that lie within the camera's field-of-view, we only retain point clouds that are inside this region. This region of interest can be easily determined by computing the back projected rays that intersect the horizontal limits of the image plane.

Down-sampling of point clouds: Since we formed an efficient model to mitigate the sparsity encountered in LIDAR point clouds, we are able to down-sample the point clouds while maintaining high accuracy. This down-sampling was done by placing cubes of dimension c throughout the 3D space and taking a spatial average of the points within each cube, leading to an increase of our algorithm's efficiency. Note that the down-sampling of information in 3D is not equivalent to that in 2D where information throughout the image gets destroyed. In the point cloud domain, the dimension c of the cube thus dictate the extent at which sampling is done. More specifically, since the distances between point clouds increase proportionally with depth, points that are far enough from the sensor would be so sparsely distributed to the extent that each cube would only contain a single point cloud. Moreover, since our intention is to learn the distribution of point clouds with respect to its depth, we posit that a proper setting of c will in no way, degrade the performance of our system. In other words, our hypothesis is that the bounding box formed with these down-sampled point clouds will be sufficiently adequate in terms of the intersection over union metric. In our implementation, we simply increase c while learning the resulting distribution until a drop in recall scores is reported. On a side-note, objects too far out in the scene that are represented by only a single point cloud prior to down-sampling are not taken into account.

3.2. Learning of point-cloud distribution

From the filtered and down-sampled point clouds, we extract all points enclosed by ground-truth objects and compute the nearest neighbour of each point. Each object is thus associated with two sets of information: the 3D position of the point clouds and the distances to their nearest neighbour. For each object, we then simply take the mean depth and nearest neighbour distance to represent the statistics of point clouds at that

depth. Given this pair of information for every object, we fit an equation of the form:

$$F(x) = A + Bx + 3\sigma(x) \quad (1)$$

where x denotes the depth, $F(x)$ the distance to the nearest neighbour, $A + Bx$ the best fit line through the mean and $\sigma(x)$ the standard deviation, extrapolated with a polynomial function of order 2. Our choice behind such a model is not arbitrary. Rather, it is based on the observation that the distribution of point clouds - in the nearest neighbour sense - tend to deviate away from the centerline the further away they are from the sensor. The relationship between the depth of a pair of point clouds and its euclidean distance thus no longer becomes linear at greater depths. Our hypothesis is that the curve $F(x)$, when utilized at test time will not only provide us with an automatic threshold for the region growing technique, but also return us high quality proposals.

3.3. Object Proposals

Estimation of Road Surface: At test time, given the filtered and down-sampled point clouds, our system begins by estimating the road plane. Since we have prior information of the road, i.e. the plane that represents the road should have a normal vector close to [0 1 0], we run RANSAC under this constraint, allowing deviations only up to 5 degrees. We then remove all inliers supporting the ground plane to prevent it from being clustered over the next steps. Through our experiments, we noted that it is more desirable to set a relatively large threshold for RANSAC at 0.2m before fine-tuning it via least squares estimation as it ensures the removal of all 'noisy point clouds'.

Construction of k-d tree: Since our method of generating proposal is built around the class of region growing techniques, we have chosen to utilize the k-d tree due its advantage in having a very efficient nearest-neighbor search algorithm with an average time complexity of $O(log\, n)$ where n is the number of points.

Region growing based on learned distribution: From this k-d tree and the learned distribution, we generate proposals via clustering point clouds by means of a region growing approach. Our approach slightly differs from traditional region growing methods in that we incorporate depth into this decision making i.e. we allow a point cloud x to grow to its nearest neighbour x' if their distance does not exceed the a certain threshold that is:

$$\|x - x'\| < F(\|x\|) \quad (2)$$

where F is the function we have chosen to fit. To further improve efficiency implementation wise, we have discretized the thresholds of $F(\|x\|)$ into something akin to the staircase function with steps at intervals of 10 meters.

Generation of additional boxes: At this stage, the bounding boxes retrieved - especially for cars - may not be representative of the actual object due to occlusion. To counter this, we have engineered an algorithm that generates additional proposals based on surrounding depth information. Given an integral image of the depth map, obtained by projecting the down-sampled point clouds onto the image plane, we project the 3D bounding box proposals onto the image and take the minimum 2D bounding rectangle then compute what we call, the difference in depth, i.e. with respect to figure 3, we sum all points in each yellow rectangle and get their signed difference. This difference essentially reveals the posiiton of the occluder with respect to the occludee. As there is no information regarding the orientation and class of the occludee, we simply place additional boxes whose dimensions resemble a car, oriented at 90 and 45 degrees to maximize recall rate. Obvious extensions could include the use of 2D features and/or 3D geometry for class-based pose estimation at the cost of running time [20]. Our focus in this article is to maximize speed while maintaining competitive recall rates. With the use of an integral image, the decision on whether to place additional proposals to the left or right can be easily computed in O(1) time. On a final note, our hypothesis is that this is one way whereby humans resolve occlusions.

4. EXPERIMENTS

We evaluate our approach on the KITTI object detection benchmark [21] which contains 7481 training and 7518 test images with three object classes: *car*, *pedestrian* and *cyclist*, each split across 3 difficulty levels at easy, moderate and hard based on the severity of occlusion and truncation. Since the test ground truth labels are not available, we split the training set into train and validation sets with each containing half the images. To obtain an equivalent 2D bounding box for the original KITTI criterion in the image space, we projected the 3D bounding box proposals onto the image via the provided transformation matrix and take the minimum 2D bounding rectangle as the 2D bounding box. For each ground-truth object, we find the proposal with the best overlap and compute the Intersection over Union. We also compute the recall rate of our system, thresholded at varying levels of IOU. Additionally, since efficiency is a crucial factor in proposal based methods, we also record the running time of our algorithm for comparison with other works. For the remainder of this section, we will only be comparing our results with state-of-the-art proposal methods with publicly available source code or with published results and running time on the KITTI dataset.

Comparison to State-of-the-art: Figure 4 compares our approach to several baselines with a plot of the recall rate

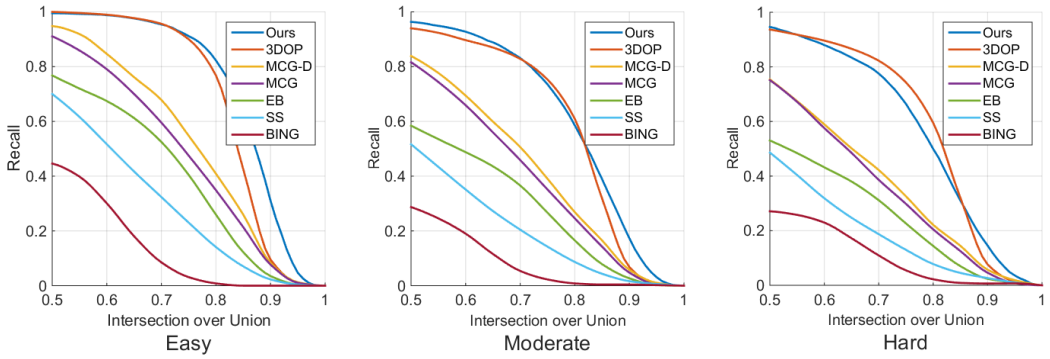

Fig. 4: Recall vs. IOU for class *car* with an average of 500 proposals.

Method	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
	Car			Pedestrian			Cyclist		
MCG-D [13]	67.84	50.61	42.20	61.77	53.19	46.40	67.84	50.61	42.20
MCG [9]	59.54	45.86	38.61	50.20	44.53	40.45	59.54	45.86	38.61
SS [8]	32.36	20.45	18.72	24.38	22.06	20.53	32.36	20.45	18.72
EB [10]	52.48	36.57	31.15	29.57	22.06	22.58	52.48	36.57	31.15
3DOP [15]	95.39	82.85	**82.25**	89.48	83.29	**74.94**	95.11	77.63	**77.38**
BING [19]	8.50	5.37	10.93	24.64	27.21	32.23	8.50	5.37	10.93
Ours	**96.54**	**83.15**	77.54	**96.46**	**87.77**	73.13	**95.63**	**91.44**	73.96

Table 1: Recall of different proposal methods with an average of 500 proposals. The IOU criteria for car, pedestrian and cyclist are at 0.7, 0.5 and 0.5 respectively. In bold are the best results.

Methods	**Time in seconds**
MCG-D [13]	160
MCG [9]	100
SS [8]	15
EB [10]	1.5
3DOP [15]	1.2
BING [19]	**0.01**
Ours	0.12

Table 2: Running time of proposal methods

as a function of the Intersection over Union (IOU) for class *car*. Our performance is comparable to the authors of 3DOP who worked with a dense stereo map. The recall rate of our system experiences a noticeable drop in the hard category. This would most probably be due to the lack of point clouds for objects that are too far out in the scene.

Recall scores based on KITTI IOU criteria: Following the standard setup of the KITTI evaluation criteria, we also report recall scores for all three classes *car* with an IOU of at least 0.7 for *car* and 0.5 for both *pedestrian* and *cyclist* respectively on table 1.

Running Time: Table 2 shows the running time of different proposal methods. On a single thread, our system takes an average of 120ms to run the whole algorithm. With respect to recall scores, we are in the same league as current state-of-the-art methods while being at least an order of magnitude faster.

Component	**Time in ms**	**Time in %**
Pre-processing	42	33
RANSAC	2	1
Proposals	82	64

Table 3: Breakdown in the running time our system with a single core machine. Time in ms is averaged across the validation set.

Component runtime analysis: We also break down how much time each component of our approach consumes at test time in table 3. Recall that the speeds are obtained with a single core i7 machine. Since each step is easily parallelizable, we can expect to see a significant increase in performance when harnessing the power of GPU.

5. CONCLUSION

We have presented a novel region proposal pipeline for outdoor objects detection via the use of LIDAR point clouds. Our technique provides an automatic solution for object clustering of point clouds in large environments by learning the distribution of point clouds. We believe that we are the first to incorporate such a relationship into region growing methods. As of now, we are investigating the use of color information to enhance the robusticity of our algorithm and are also working on a proof of the model.

6. REFERENCES

[1] A. Lopez A. Gonzalez, D. Vazquez and J. Amores, "Onboard object detection: Multicue, multimodal, and multiview random forest of local experts," in *IEEE Transactions on Cybernetics*, 2016.

[2] J. Xu D. Vazquez J. Amores A. Gonzalez, G. Villalonga and A. Lopez, "Multiview random forest of local experts combining rgb and lidar data for pedestrian detection," in *IEEE Intelligent Vehicles Symposium*, 2015.

[3] D. Zeng Wang C. Hay Tong M. Engelcke, D. Rao and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *ArXiv e-prints*, 2016.

[4] T. Zhang B. Li and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," in *Robotics: Science and Systems*, 2015.

[5] V. Steinhage J. Behley and A. Cremers, "Laser-based segment classification using a mixture of bag-of-words," in *IEEE International Conference on Intelligent Robots and Systems*, 2013.

[6] J. Batista C. Premebida, J. Carreira and U. Nunes, "Pedestrian detection combining rgb and dense lidar data," in *IEEE International Conference on Intelligent Robots and Systems*, 2014.

[7] D. Wang and I. Posner, "Voting for voting in online point cloud object detection," in *Robotics: Science and Systems*, 2015.

[8] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," in *International Journal for Computer Vision*, 2015.

[9] P. Arbelaez, J. Pont-Tuset, J.T. Barron, F. Marques, and J.Malik, "Multiscale combinatorial grouping," in *Computer Vision and Pattern Recognition*, 2014.

[10] C. L. Zitnick and P. Dollar, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*, 2014.

[11] S. Fidler T. Lee and S. Dickinson, "A learning framework for generating region proposals with mid-level cues," in *International Conference on Computer Vision*, 2015.

[12] I. Endres and D. Hoiem, "Category independent object proposals," in *European Conference on Computer Vision*, 2010.

[13] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *European Conference on Computer Vision*, 2014.

[14] A. Karpathy, S. Miller, and L. Fei Fei, "Object discovery in 3d scenes via shape analysis," in *International Conference on Robotics and Automation*, 2013.

[15] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun, "3d object proposals for accurate object class detection," in *Neural Information Processing Systems*, 2015.

[16] S. Fidler D. Lin and R. Urtasun, "Holistic scene understanding for 3d object detection with rgbd cameras," in *International Conference on Computer Vision*, 2013.

[17] Jan Hosang, Rodrigo Benenson, and Bernt Schiele, "How good are detection proposals, really?," in *Proceedings of the British Machine Vision Conference*, 2014.

[18] Shi Jianbo and Malik Jitendra, "Normalized cuts andimage segmentation," in *IEEE Transactions Pattern Analysis Machine Intelligence*, 2015.

[19] M. Cheng, Z. Zhang, M. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Computer Vision and Pattern Recognition*, 2014.

[20] Justin Lev, Joo-Hwee Lim, and Nizar Ouarti, "Principal curvature of point cloud for 3d shape recognition," in *International Conference on Image Processing*, 2017.

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition*, 2012.