2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Object Modeling from 3D Point Cloud Data for Self-Driving Vehicles

Shoaib Azam[1], Farzeen Munir[1], Aasim Rafique[2], YeongMin Ko[1], Ahmad Muqeem Sheri[1] and Moongu Jeon[1]

Abstract— **For autonomous vehicles to be deployed and used practically, many problems are still needed to be solved. One of them we are interested in is to make use of a cheap LIDAR for robust object modelling with 3D point cloud data. Self-driving vehicles require accurate information about the surrounding environments to decide the next course of actions. 3D point cloud data obtained from LIDAR give more accurate distance than the counterpart stereo images. As LIDAR generates low-resolution data, the object detection and modeling is prone to produce errors. In this work, we propose the use of multiple frames of LIDAR data in an urban environment to construct a comprehensive model of the object. We assume the use of LIDAR on a moving platform and the results are almost equal to the 3D CAD model representation of the object.**

I. INTRODUCTION

Self-driving car running on the road is going to be one of the major breakthroughs which intelligent vehicle community is anticipating in near future. All major stakeholders are striving hard to achieve the milestone and a routine demonstration of the model products is a proof of existence for their efforts. Although the self-driving vehicles brought fantasy to life, they are still not affordable for consumers. One of the reasons for increased cost is the use of expensive sensors. LIDAR is a laser sensor, which gives accurate distances of the surrounding objects from the vehicle carrying it. LIDAR along with cameras is one of the most commonly used sensors to conceive the entire surroundings of self-driving vehicles.

LIDAR generates 3D point cloud data with the accurate distances of the objects. The 3D point cloud data can be used to detect and classify the objects in the vicinity along with the 2D cameras, but with accurate distances of detected objects with LIDAR data. The lower resolution of the 3D point cloud data, the cheaper LIDAR price and the worse detection and classification results deteriorate.

In the recent years, use of LIDAR in the intelligent vehicles has gained a lot of popularity ranging from its applications to solving computer vision tasks to robotics. Many authors have used LIDAR along with cameras as a multiple-source modality to get more information. [1] has used LIDAR and camera in conjunction as an aiding source of information for pedestrian detection, where they have investigated the effect of multi-cue, multiple sources of information and multi-view classifier in the context of pedestrian detection both individually and altogether. Similarly in [2] the authors fused the color camera information with that of LIDAR for multiple-object detection and presented the segmentation scheme by using both 2D and 3D information. [3] unified the monocular camera and LIDAR for vehicle and pedestrian detection and tracking. In [3] Gaussian mixture and AdaBoost classifier have been used for object classification in laser and vision space respectively and a Bayesian-sum decision rule is introduced for combined classification results.

Localization of vehicles using LIDAR has been discussed in [4] by using vertical corner features. In addition, [4] also discussed the vehicle pose by accumulating the output obtained from the iterative closet point (ICP) and geometrical relation between the scans of 3D LIDAR. For perception of the environment, the two tasks that are important for any robotic system are simultaneous localization and mapping (SLAM) and detection and tracking of multiple objects. In [5] a new algorithm has been discussed that combines the two tasks and is applicable to 3D dense range data. Some authors have used map-based localization method for automated driving in an urban environment like in [6] where the authors have generated the 3D maps using LIDAR as a prior information.They localized the vehicle by using image data from monocular camera and matched them against different rotation of LIDAR data. They optimized the Mutual information.

Recognition of objects in the 3D point cloud data can be categorized in two major branches: one is extracting features from point cloud data and the other is transforming the data to voxelized point cloud data to form 3D CAD models. In [7], the authors have used a segment and holistic classifier as boosting framework for classification of objects in point cloud data. [8] has presented the lane recognition using LIDAR and they calculated the lane curvature, yaw angle and offset using Hough Transform and tracking of lane marks are being determined by Extended Kalman Filter. The other type of recognition that is extensively used for point cloud data is the generation of 3D CAD models [9], [10], [11], [12], [13]. 3D convolutional neural networks have also shown good results when applied to 3D data. [14] has applied 3D convolutional neural network to the voxel model of point cloud data to predict the class label and argued that the orientation played the vital role in 3D recognition. Similarly, in [15] they have introduced a 3D representation of an object called Voxel Pattern for estimating 3D properties of multiple objects in images and trained those voxel patterns for recognition.

In this work, we propose a method of completing the sparse 3D point cloud data generated from a cheap LIDAR. We rightly assume that the LIDAR is mounted on a self-

[1] School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, South Korea
mgjeon@gist.ac.kr
[2]Department of Computer Science, Quaid-e-Azam University, Pakistan

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

driving vehicle and successive frames of LIDAR data are available for computation. The overall structure of our work is shown in Fig.1.

II. Methodology

LIDAR has been extensively used in autonomous driving for obstacle detection and avoidance. LIDAR outputs a 3D point cloud of the environment, which shows the current location of each object in the environment. It is vital to recognize each object accurately so that the next course of action of the autonomous car can be decided. Even though the LIDAR point cloud is considered as 3D data, the depth information of object could not be determined from single frame of point cloud. For object detection, researchers have used the method of template based pre-processing of 3D point cloud data for CAD model data. This is computationally expensive and consumes more memory. This paper presents an exquisite method for object detection by using successive frames of LIDAR data.

A. Proposed Method

Velodyne LIDAR gives 3D point cloud data of surroundings in real time. As can be seen in Fig.2(c), the point cloud is sparse and points are insufficient for estimating the shape of the object. The 3D representation of an object using the single frame of point cloud data from LIDAR is unobtainable. The LIDAR data are pre-processed for 3D reconstruction of object. Here we proposed to use successive frames of point cloud data to reduce the sparsity of data and to completely reconstruct the 3D object.

1) Ground Plane Extraction: The extraction of ground plane plays a decisive role in deciding a reference point for various geometric estimates. In this step, we removed ground plane by fitting a planar model to the point cloud data. Normally RANdom SAmple Consensus (RANSAC) algorithm has been used for this, but we have used an M-estimator SAmple Consensus (MSAC) algorithm [16] to find the plane which is a generalized version of RANSAC.

2) Registration of Successive Frames: Since data are collected while driving, each frame of the point cloud is shifted from the previous frame. The successive frame needs to be registered for object reconstruction. The registration of the point cloud is performed by finding the translation and rotation matrices that map one set of point cloud to another. The Modified version of ICP algorithm is used for registration of successive frames[17]. The technique used the robust M-estimation to reduce the effect of the outliers and it incorporates the iterative re-weighted least square. The expression 1 shows the minimization function, where R represents the rotation matrix and t represents the translation matrix. In objective function (2) X defines the set of reference point cloud and P_i specifies the point cloud that is registered.

$$\min_{R,t} f(R,t), \tag{1}$$

where,

$$f(R,t) = \sum_{i=1}^{N} Q(d(RP_i + t, X)), \tag{2}$$

$$d(p,X) = min_{y \in X} \left\| p - y \right\|_2, \tag{3}$$

Q is a robust criterion function, which provides convergence. The successive frames are registered using the modified ICP and then mapped to the reference frame.

3) 3D Model Generation: As our main objective is to use the concept of successive frames from LIDAR scans, we first extract the ground plane, and remove noise from the LIDAR data as a pre-processing step. Fig.2(a) and (c) show the original scan of LIDAR data along with extracted ground plane respectively.

After extracting the ground plane, we reconstruct the 3D model using the concept of successive frames to LIDAR data. In order to do so we have used the ICP algorithm and applied it to LIDAR data and then merge those using box grid filter. The box grid filter computes the axis-aligned bounding box of overlapped region between two point cloud data. The bounding box is divided in to grid boxes and the size of these grid boxes is being specified empirically. The points which are in the region of these grid boxes are merged together by averaging their locations, normals and colors, besides other points which lie outside the overlapped region are intact.

Fig.3 shows the results obtained after applying ICP and box grid filter to our LIDAR data. After this the desired object is being cropped from the point cloud data and the segmented point cloud data is de-noised and up-sampled for 3D CAD model generation.

III. Experimentation and Results

In this section, we evaluate the proposed methodology on our dataset, collected from Velodyne VLP-16 in Cheomdan, Gwangju, South Korea, and then compare our results qualitatively and quantitatively with Sydney Urban dataset [18].

A. Velodyne LIDAR Puck

The Velodyne LIDAR Puck is the latest product of the Velodyne, which is aimed to reckon with the demands of autonomous driving and ADAS system. VLP-16 sensor is cost effective and the smallest in size. 16 lasers and detectors pairs are assembled into a compact unit, which rotates at a frequency of 15 Hz to obtain a 360 view of the surroundings. It gives a range of 100 meters and vertical field of view of $30°$ ($+15°$ to $-15°$). $300,000$ points/sec are generated which specify the X, Y and Z coordinates of objects in the surroundings in Cartesian coordinate. Fig.2(a) shows the sample data obtained from Velodyne LIDAR PUCK. The Velodyne is mounted on a car and data is collected while driving in Cheomdan Area, Gwangju, South Korea. Each frame of point cloud gives the current state of surrounding environment.

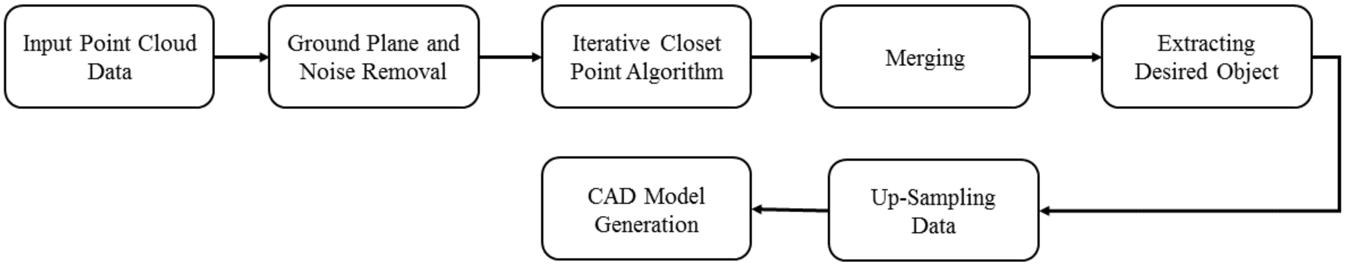

Fig. 1. *Overall structure of CAD Model generation using Successive frame of LIDAR data.*

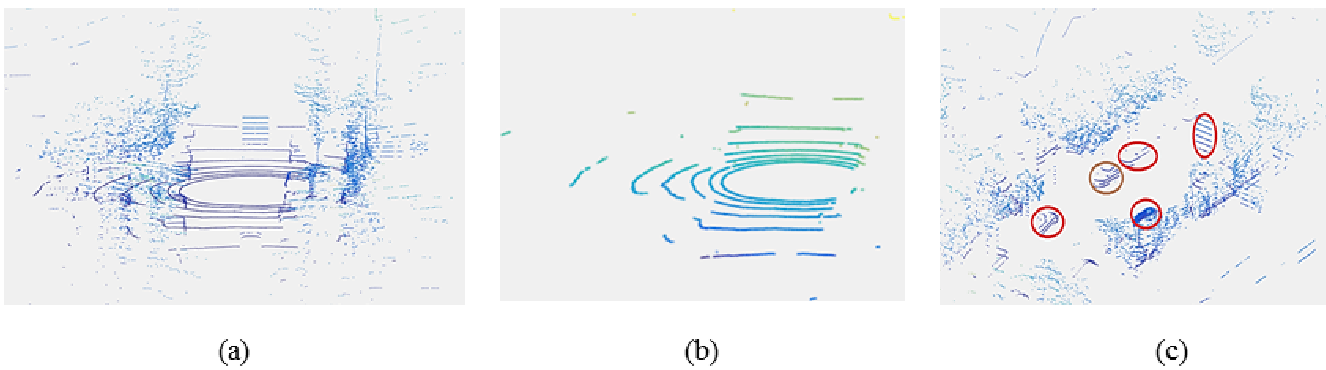

Fig. 2. *(a) Original LIDAR point cloud data. (b) Ground plane extracted from the original point cloud data using MSAC algorithm. (c) Removed ground plane point cloud data showing the objects above ground plane as encircled in the figure.*

B. Qualitative Analysis

To compare our results qualitatively with the Sydney urban dataset[18] we have generated the CAD models of our data and as well as Sydney dataset. Fig.4 depicts the CAD models of ours and Sydney dataset. As it is clearly seen from the results that our CAD models are complete and more close to original CAD model as compared to Sydney dataset, which are incomplete in nature because of the fact that they are using only single frame data and the depth information about the certain object, is missing from their data.

C. Quantitative Analysis

For the quantitative analysis, we compare our results in the context of recognition with the Sydney dataset. Recognition of 3D cad models have been classified by their representation. Normally there are two categories; i) Rasterized (regular) and ii) Geometric(irregular) data representation of 3D cad models. We have selected both representations for our data and use state-of-the-art algorithms in both domains for recognition.

To rasterized domain, there are two approaches: multi-view images and volumetric. We have focused on the volumetric approach and have used VoxNet[19], which is the state-of-the-art algorithm and uses 3D convolutional neural network. The architecture of VoxNet is C(32, 5, 2)–C(32, 3, 1)–P(2)–FC(128)–FC(K), where K is the number of classes, C is for convolutional layers, P is for pooling layers and FC stands for fully connected layer.

We have trained the VoxNet on the ModelNet dataset [20] on those objects respective to the urban environment like cars and persons. We also keep in mind that the objects in both Sydney dataset and our dataset are similar and for training, we make sure that the ModelNet dataset includes the same objects. We evaluated our results by training the VoxNet on ModelNet using two strategies. One is to train the VoxNet by using all the data rotation available in the ModelNet dataset that is equal to twelve views.The results using this strategy are shown in Fig.5(a). As it can be seen from our results, the CAD models generated from our proposed method are correctly recognized by the VoxNet while on the other side the CAD models of Sydney dataset are not classified correctly. The other strategy that we use is to train the VoxNet using only single rotation and then compare the recognition results of our data and Sydney data. Fig.5(b) shows the results using this approach and it is evident from the results that our CAD models are recognized as correctly. The advantage of this approach is that it is more computationally efficient besides using all the data rotations. The quantitative results are being shown in Table I and Table II for recognition of our cad model and Sydney cad model using VoxNet respectively.

Similarly, we have also used PointNet [21] from geometric(irregular) data representation domain, for the recognition which uses the point cloud data itself without transforming it to some other representation. The architecture of PointNet is being shown in Fig6. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is

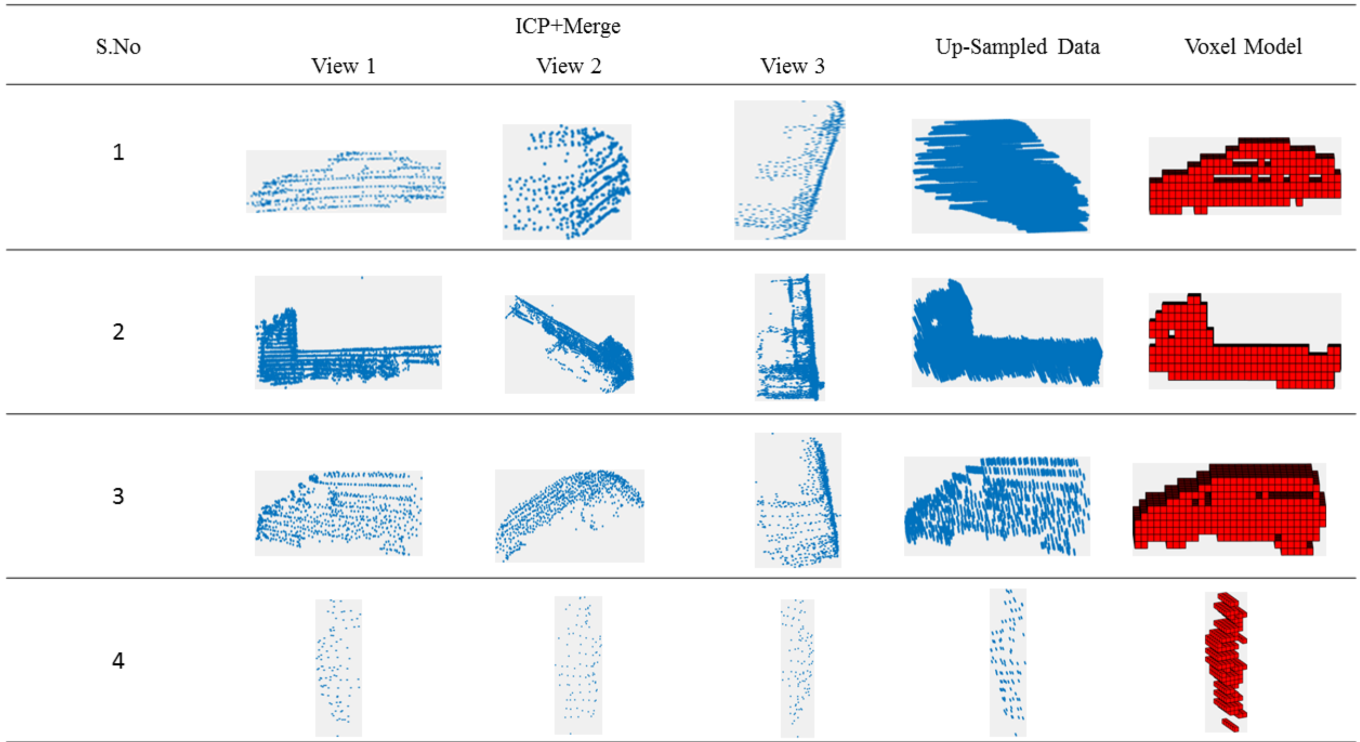

Fig. 3. *The results of our proposed method containing ICP and Merging. ICP+Merge containing different views of point cloud data. Up-sampled results show the up-sampled point cloud data obtained after successive frames and voxel model contains the 3D CAD models obtained by proposed algorithm*

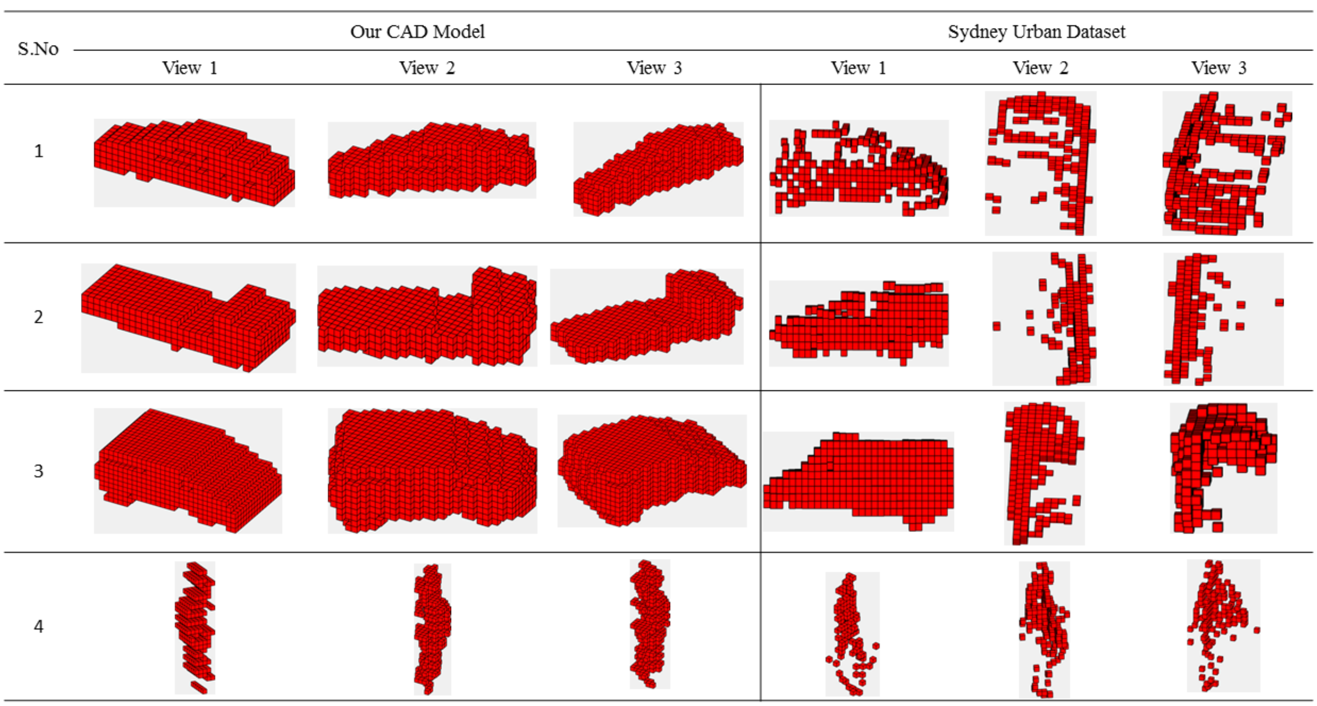

Fig. 4. *The qualitative comparison between our CAD models and Sydney Urban data. It includes the three views of our CAD model and Sydney Urban dataset CAD model of the same object. From the CAD model of Sydney Urban data, it can be seen that they are incomplete as compared to our CAD model.*

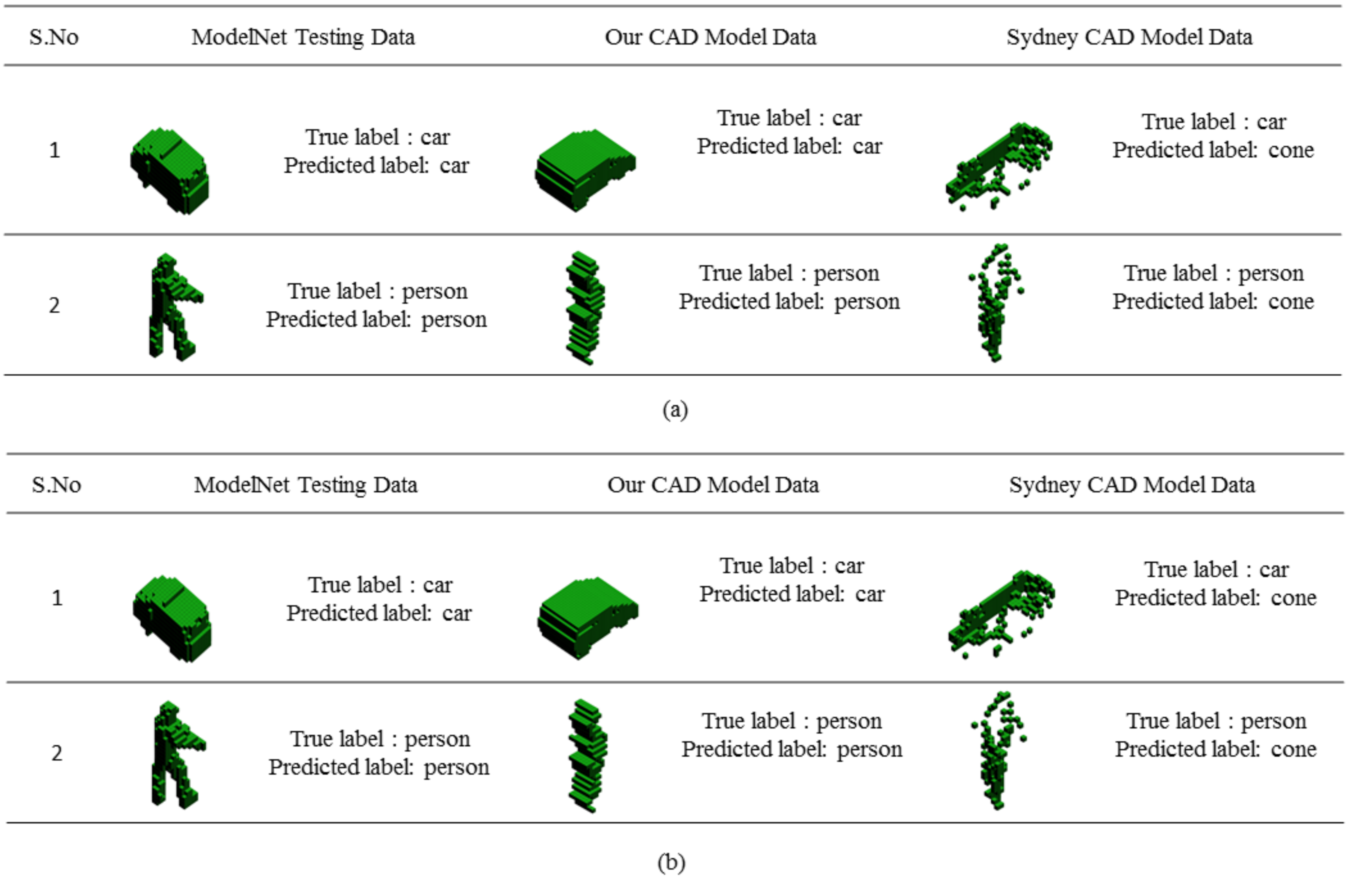

Fig. 5. *Quantitative comparison between our CAD models and Sydney Urban data using VoxNet for recognition. (a) Recognition results when VoxNet is trained using the all the data rotations of ModelNet dataset. (b) Recognition results when VoxNet is trained only on single rotation of ModelNet dataset.*

classification score for m classes. For training the PointNet we have combined both ModelNet data and our point cloud data which is being uniformly sampled and normalized to unit sphere. Data augmentation is being done during training by rotating along up axis and jitter the position of point cloud points by Gaussian noise with zero mean and 0.02 standard deviation. We compare our results with those of Sydney dataset and found that our recognition results are more accurate as compared to those of Sydney Dataset. The quantitative results are shown in Table III and Table IV respectively.

TABLE I
QUANTITATIVE RESULTS OF OUR CAD MODEL USING VOXNET

	Bus	Car	Person	Truck	F1-score
Bus	8	1	0	1	0.73
Car	1	7	0	2	0.74
Person	2	0	8	0	0.89
Truck	1	1	0	8	0.76

IV. CONCLUSION

In this work, we have utilized the concept of successive frames for gathering information from LIDAR. For this, we have adopted the ICP algorithm and merging technique to reconstruct the 3D CAD model of point cloud data. We have generated our own dataset using the VLP-16 for this and then qualitatively compared our CAD models with Sydney Urban dataset. Therefore, qualitatively our CAD models are complete and more compact in nature as compared to those of Sydney Urban dataset. We have also done the comparison quantitatively using the VoxNet with the Sydney Urban dataset in term of data rotation augmentation and found that our CAD model is computationally efficient and

TABLE II
QUANTITATIVE RESULTS OF SYDNEY CAD MODEL USING VOXNET

	Bus	Car	Person	Truck	F1-score
Bus	6	1	0	3	0.5
Car	4	5	0	1	0.56
Person	2	0	7	1	0.82
Truck	2	2	0	6	0.57

TABLE III
QUANTITATIVE RESULTS OF OUR CAD MODEL USING POINTNET

	Bus	Car	Person	Truck	F1-score
Bus	9	1	0	0	0.86
Car	1	8	0	1	0.76
Person	1	0	9	0	0.95
Truck	0	2	0	8	0.84

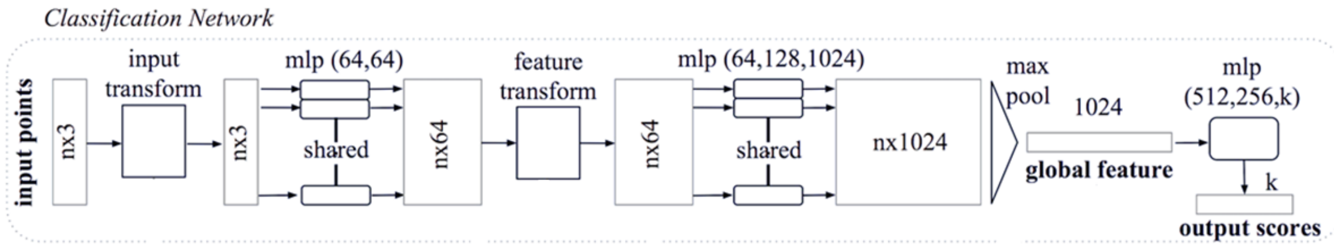

Fig. 6. *Classification network of PointNet architecture [21].*

TABLE IV
QUANTITATIVE RESULTS OF SYDNEY CAD MODEL USING POINTNET

	Bus	Car	Person	Truck	F1-score
Bus	7	0	0	3	0.7
Car	2	6	0	2	0.67
Person	0	0	8	2	0.89
Truck	1	2	0	7	0.58

are better recognized with few data rotations as compared to Sydney Urban dataset. In addition to that we have also performed comparison by using point cloud as it is without transforming it to some other representation by using the PointNet architecture for recognition and found that the point cloud formed by using the successive frame technique to our data has better recognition results as compared to Sydney dataset.

The future work includes shape estimation from the point cloud data to generate CAD model as an application to scene completion in both indoor and urban environments and also for autonomous driving.

ACKNOWLEDGMENT

This work was partly supported by the ICT R&D program of MSIP/IITP. (2014-0-00077, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis) and GIST Autonomous Vehicle project.

REFERENCES

[1] A. Gonz, G. Villalonga, J. Xu, V. David, J. Amores, and M. L. Antonio, "Multiview Random Forest of Local Experts Combining RGB and LIDAR data for Pedestrian Detection," no. Iv, 2015.

[2] S. Hwang, N. Kim, Y. Choi, S. Lee, and I. S. Kweon, "Fast Multiple Objects Detection and Tracking Fusing Color Camera and 3D LIDAR for Intelligent Vehicles," pp. 234–239, 2016.

[3] C. Premebida, U. Nunes, and P. Peixoto, "A LIDAR and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking," pp. 1044–1049, 2007.

[4] J. H. Im, S. H. Im, and G. I. Jee, "Vertical corner feature based precise vehicle localization using 3D LIDAR in Urban area," *Sensors (Switzerland)*, vol. 16, no. 8, 2016.

[5] F. Moosmann and C. Stiller, "Joint Self-Localization and Tracking of Generic Objects in 3D Range Data," pp. 1146–1152, 2013.

[6] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 176–183, 2014.

[7] A. Teichman, J. Levinson, and S. Thrun, "Towards 3D Object Recognition via Classification of Arbitrary Object Tracks," pp. 4034–4041, 2011.

[8] U. Weiss, P. Biber, S. Laible, K. Bohlmann, and A. Zell, "Plant species classification using a 3D LIDAR sensor and machine learning," in *Proceedings - 9th International Conference on Machine Learning and Applications, ICMLA 2010*, pp. 339–345, 2010.

[9] T. Ogawa and K. Takagi, "Lane Recognition Using On-vehicle LIDAR," *Intelligent Vehicles Symposium*, pp. 540–545, 2006.

[10] F. Bosché, "Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction," *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 107–118, 2010.

[11] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "CAD-model recognition and 6DOF pose estimation using 3D cues," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 585–592, 2011.

[12] J. J. Lim, A. Khosla, and A. Torralba, "FPM: Fine pose parts-based model with 3D CAD models," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8694 LNCS, no. PART 6, pp. 478–493, 2014.

[13] M. Stark, M. Goesele, and B. Schiele, "Back to the Future: Learning Shape Models from 3D CAD Data.," *Bmvc*, pp. 1–11, 2010.

[14] N. Sedaghat, M. Zolfaghari, and T. Brox, "Orientation-boosted Voxel Nets for 3D Object Recognition," *arXiv:1604.03351 [cs.CV]*, pp. 1–22, 2016.

[15] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D Voxel Patterns for object category recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 1903–1911, 2015.

[16] Torr, Philip HS, and Andrew Zisserman "MLESAC: A new robust estimator with application to estimating image geometry,"*Computer Vision and Image Understanding*, 78.1 (2000): 138-156.

[17] Bergstrm, Per, and Ove Edlund,"Robust registration of point sets using iteratively reweighted least squares,"*Computational optimization and applications*, 58.3 (2014): 543-561.

[18] De Deuge, Mark, et al,"Unsupervised feature learning for classification of outdoor 3d scans.,"*Australasian Conference on Robitics and Automation*, Vol. 2. 2013.

[19] Maturana, Daniel, and Sebastian Scherer,"Voxnet: A 3d convolutional neural network for real-time object recognition,"*Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE*, 2015.

[20] Wu, Zhirong, et al,"3d shapenets: A deep representation for volumetric shapes,"*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*,2015.

[21] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." arXiv preprint arXiv:1612.00593 (2016).