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

A 3D Dynamic Scene Analysis Framework for Development of Intelligent Transportation Systems

Chien-Yi Wang[1] Athma Narayanan[1] Abhishek Patil[1] Wei Zhan[2] Yi-Ting Chen[1]

***Abstract*— Holistic driving scene understanding is a critical step toward intelligent transportation systems. It involves different levels of analysis, interpretation, reasoning and decision making. In this paper, we propose a 3D dynamic scene analysis framework as the first step toward driving scene understanding. Specifically, given a sequence of synchronized 2D and 3D sensory data, the framework systematically integrates different perception modules to obtain 3D position, orientation, velocity and category of traffic participants and the ego car in a reconstructed 3D semantically labeled traffic scene. We implement this framework and demonstrate the effectiveness in challenging urban driving scenarios. The proposed framework builds a foundation for higher level driving scene understanding problems such as intention and motion prediction of surrounding entities, ego motion planning, and decision making.**

I. INTRODUCTION

Driving scene understanding is a critical step toward intelligent transportation systems. It requires a system that can analyze, interpret and reason about traffic scenes and thus make a decision for safe navigation. We have witnessed tremendous advances in the development of intelligent transportation systems. However, it is still challenging for the existing technologies to navigate in all scenarios.

To accelerate the development of intelligent driving systems, open source platforms [1], [2] have been released. They provide architectures that include baseline algorithms for perception, localization, planning and decision making for developers to build their own solutions. These components have been studied extensively in academia and industries. In this paper, we focus on perception (scene analysis) for the development of intelligent transportation systems.

Scene analysis has been studied in the computer vision and robotics community for decades. A recent rapid evolution was triggered by publicly available large datasets, deep learning algorithms, and powerful computational resources in the field of object detection and tracking, scene labeling, 3D scene reconstruction, and simultaneous localization and mapping (SLAM). In this work, a unified framework that systematically integrates all the aforementioned fields is designed to analyze traffic scenes. Specifically, the framework generates the states (i.e., category, position, orientation and velocity) of traffic participants and the ego car. This allows us to simultaneously reconstruct and semantically label the entire traffic scene.

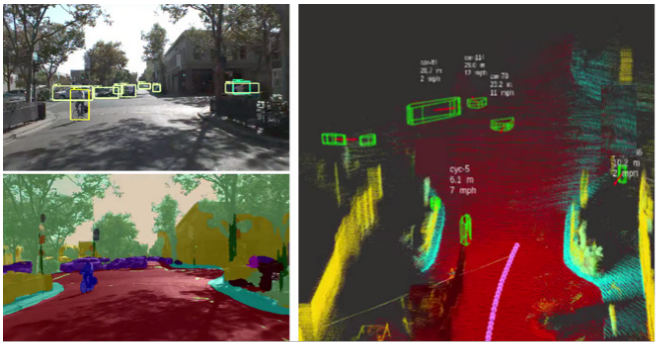


Fig. 1: We demonstrate a 3D dynamic scene analysis framework that systematically combines different perception modules. Specifically, given synchronized 2D and 3D sensory data from camera and LiDAR, the proposed framework generates 3D position, orientation, and velocity of traffic participants and ego car in a 3D reference coordinate system obtained from LiDAR SLAM. Moreover, we simultaneously reconstruct a 3D semantically labeled static map. The figure shows the effectiveness of our framework in a challenging urban driving scenario.

At one end of the spectrum, self-driving technologies by Google Waymo and others [1], [2] utilize High-Definition (HD) maps and perform real-time localization, obstacle avoidance, motion planning and decision making to navigate in those areas. While leading to impressive real-world deployment, this approach suffers from the fact that it requires a huge amount of manual efforts to build a high accurate map before testing. On the other end of the spectrum, research in traffic scene understanding using visual cues [3], [4], [5], [6], [7] have shown significant progress. These technologies have been used in the current advanced driver assist systems. However, vision only solutions still suffer from its inherent incapabilities in different illumination conditions and limited depth perception.

The proposed framework leans toward the spectrum without using sophisticated HD maps, and we leverage a multi-sensor fusion approach instead of using only monocular or stereo vision. Our goal is to provide a robust foundation to drive research in those advanced topics in driving scene understanding such as driver and traffic participant motion analysis, motion planning and decision making. We provide discussions on how we leverage the outputs of the proposed

[1]C.-Y. Wang, A. Narayanan, A. Patil, and Y.-T. Chen are with Honda Research Institute, 375 Ravendale Dr, Suite B, Mountain View, CA, 94043, USA. The first three authors contributed equally to this work {cwang,anarayanan,apatil,ychen}@honda-ri.com

[2] W. Zhan is with Department of Mechanical Engineering, University of California, Berkeley, Berkeley, CA, 94720, USA. wzhan@berkeley.edu

framework in Section V. A summary of the contributions are described as follows:

- A framework to analyze synchronized 2D and 3D sensory data to obtain category, position, orientation and velocity of traffic participants and the ego vehicle.
- A methodology to reconstruct and semantically label the surroundings of the ego vehicle.
- Implemented and tested in different challenging urban driving scenarios.

The rest of this paper is organized as follows. In Section II, we review related studies in the area of driving scene analysis. In Section III, we provide details of the algorithms used in our approach. We show the performance of our framework under different driving scenarios in Section IV. Finally, we discuss the extension of the scene analysis framework to higher level driving scene understanding in SectionV.

II. RELATED WORK

A substantial amount of works [8], [9], [3], [5], [4], [10], [11] in the field of scene analysis have been studied for years. We review the most related works in the following.

Osep et. al. [7] proposed a framework that fuses inputs from camera and LiDAR to detect and track dynamic objects in 3D scenes. To generate coupled 2D-3D observations and tracks, a method based on Conditional Random Fields is used to fuse candidates from 2D and 3D spaces. The multi-object detection and tracking for the framework is performed in a world coordinate system that is obtained from stereo-based odometry estimation methods [12]. The proposed framework also leverage inputs from different sensors to detect and track objects. We fuse the two modalities motivated by a recent work [13]. We effectively combine the strength of 2D image for visual recognition [14] and 3D point cloud for its direct distance measurements. Additionally, we perform multi-object tracking under a global coordinate system constructed by LiDAR SLAM instead of stereo-based odometry estimation to improve the robustness.

Barsan et al. [15] and Geiger et al. [4] demonstrated unified frameworks which include visual cues such as vehicle tracklets, vanishing points, semantic scene labels, scene flow, occupancy grids and 3D reconstructions to determine 3D scene layout and the location and orientation of objects. These works are similar to the proposed framework. Some notable differences are summarized as follows. First, we leverage LiDAR SLAM instead of stereo-based SLAM techniques to ensure robustness of 3D scene reconstruction which is the first step toward developing intelligent transportation systems. Second, an on-line semantic mapping module for dynamic traffic scenes is introduced to provide scene priors for reasoning. Third, a framework that estimates traffic participant's state including position, size, velocity, orientation and category is proposed to enable research motion prediction and planning from a decision maker perspective with uncertainties.

The importance of scene analysis can be further gleaned from works such as DESIRE [16]. DESIRE predicts future locations of objects in multiple scenes. Specifically, they account for uncertainties in future predictions and reasoning based on past motion history, scene context as well as the interactions among the agents. A Recurrent Neural Network fusion module jointly captures past motion histories, the semantic scene context and interactions among multiple agents. Since such applications rely on scene context, parsing the scene is of utmost importance. The proposed framework serves as an initial step towards this goal and allows motion prediction.

III. ALGORITHMIC OVERVIEW

We process synchronized 2D and 3D sensory data to generate spatial and temporal attributes of a traffic scene. The framework is designed to obtain

(i) Category, position, orientation and velocity of traffic participants in a reference world coordinate
(ii) Position, orientation and velocity of ego car in a reference world coordinate
(iii) 3D semantically labeled map

The overall architecture is shown in Figure 2.

To obtain the attributes of traffic participants, we first perform 3D object detection. The detection results are used in LiDAR SLAM to construct a global reference coordinate system to localize traffic participants. The position and category of traffic participants are obtained from 2D/3D object detection and tracking. With the tracked positions of an object, we can estimate its absolute velocity.

While performing LiDAR SLAM, a sequence of point cloud frame is registered. Given the calibrated projection matrix, each 3D point is labeled by associating the corresponding semantic label in 2D image space. The labeled 3D point cloud is accumulated, voxelized and filtered to obtain a semantically labeled map.

The detailed description of each component is provided in the following.

A. Semantically Labeled Map

1) LiDAR SLAM: It is important to establish a global reference coordinate to ensure consistency and ease of adaptability of other modules in the framework. In this paper, the first frame of a given sequence is chosen as the reference frame with coordinate axes at the LiDAR sensor's origin.

In this paper, the state-of-the-art LiDAR SLAM algorithm [17] is used. In this method, the odometry and mapping algorithms run in parallel. Specifically, the odometry algorithm performs coarse feature points matching to estimate the velocity and correct the distortion in point cloud, while the mapping algorithm conducts fine feature points association to accurately estimate the pose of the ego-car. Two sets of feature points are extracted from the point cloud, i.e., the edge features and the surface features. These are extracted by calculating the smoothness of the local surface in the point cloud. In order to ensure robustness, we modified [17] with following changes: 1) increase the maximum number of iterations and 2) decrease the threshold for convergence, and 3) calculate odometry for every frame without skipping frames as done in the original paper.

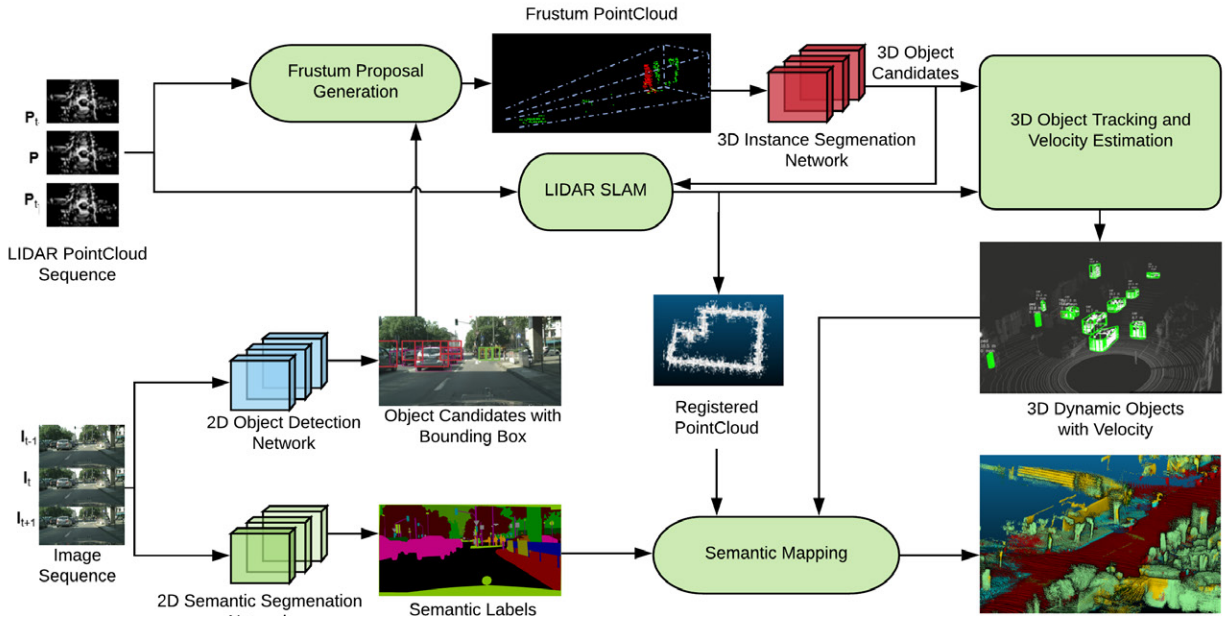


Fig. 2: Overview of the proposed 3D dynamic scene analysis framework.

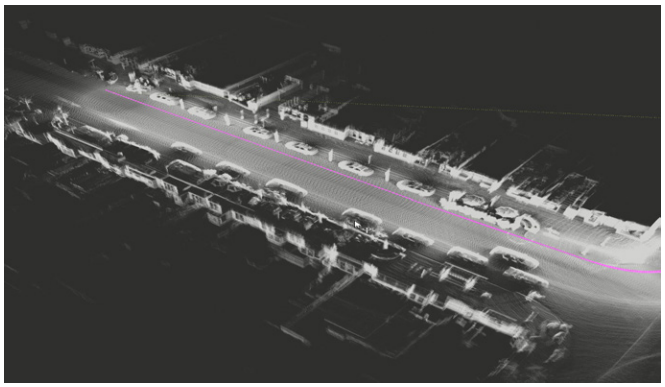


Fig. 3: An example of LiDAR SLAM output that consists of the ego car trajectory (in pink) and registered point cloud.

Fig. 4: An example of semantic segmentation output.

In addition, we remove 3D dynamic objects such as cars, pedestrians and cyclists from the output of 3D instance segmentation module (section III-B.1) prior to performing SLAM to further improve the odometry estimates and map registration. It is motivated by [18]. Note that in our current approach, 3D detections are obtained only in the front view (approximately -65 degrees to +65 degrees FOV). In order to achieve even better results, we plan to extend the FOV to 360 degrees around the ego-car by detecting and removing dynamic objects in the entire 3D scene by leveraging techniques similar to those highlighted in [19]. Figure 3 shows a SLAM output highlighting the point cloud registered nap and the odometry.

2) Semantic Segmentation: To label 3D scenes, we adopt a projection based approach that transfers semantic segmentation on 2D images to 3D registered point cloud. For 2D semantic segmentation, the goal is to classify every pixel into one class in a predefined category. Following the state-of-the-art approaches semantic segmentation, we apply Resnet-101 [20] as the backbone for feature extraction and initialize weights from a pretrained ImageNet model [21]. Additionally, we modified Resnet-101 [20] to adopt the Feature Pyramid Network (FPN) structure as in [22], i.e., by adding top-down pathway and skip-connections, to enrich features at higher resolution feature maps. Each level of the pyramid will make a prediction independently. The final result is fused by averaging the predictions from different levels. Compared to a dilated convolution structure that is used in DeepLab [23], the FPN-based semantic segmentation network can generate high resolution outputs while reducing

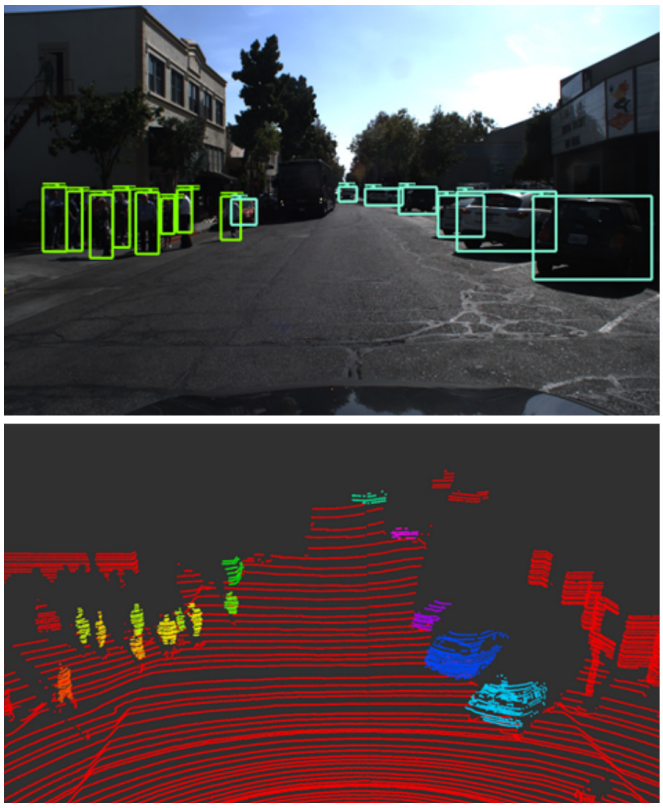


Fig. 5: An example of 3D Instance segmentation output with instance clusters of pedestrians and vehicles and the corresponding 2D object proposals results are shown.

the memory usage and computation.

We train the network using mini-batch SGD with the following training parameters, i.e., minibatch size is 4, patch size is 768×768 (randomly cropped from the original image), initial learning rate is 5×10^{-4}, and a polynomial learning rate with power 0.9. We evaluate our single model on Cityscapes dataset [24]. The model achieves a mean IOU of 75.42% on the validation set.

We use the following 17 classes in our dataset, i.e., *Static: Road, Sidewalk, Lane markings, Traffic signals, Traffic signs, Poles, Billboards, Vegetation, Building structures, Sky* and *Dynamic: Car, Truck, Bus, Two-wheeler, Railed Vehicle, Person, Rider*. We train the model using the aforementioned protocol with 200K iterations on 1048 finely annotated images on a single Nvidia Titan X GPU. An example output is shown in Figure 4.

3) Semantic Mapping: Semantic Mapping is the process of a) classifying each 3D point in a point cloud into one of the predefined 17 classes as we discussed in the Section III-A.2 b) Accumulating sequence of labeled point cloud into a 3D semantically labeled map. We describe the process as follows.

We project each 3D point to the 2D image plane with the projection matrix obtained from Camera-LiDAR calibration [25], [26]. Hence, we can find the correspondence of a 3D point (x, y, z) and the associated 2D pixel (u, v). Given the pixel (u, v), we obtain the associated semantic label $s_{u,v}$ and form a colored point cloud $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{s})$. The colored point cloud is voxelized to generate a voxelized colored point cloud $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}, \hat{\mathbf{s}})$, where a 3D point $(\hat{x}, \hat{y}, \hat{z}) \in (\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ is the centroid within a voxel with a size 0.2 m^3 and the corresponding semantic label $\hat{s} \in \hat{\mathbf{s}}$ is the majority vote of all semantic labels.

However, a semantic label $\hat{s}$ within a voxel may not be correct at first place due sensor noise as well as inaccuracies in 2D semantic segmentation. To alleviate this issue, we account for temporal consistency of the label of a voxel. Specifically, we accumulate the observations over time and update the label if the probability of another class is higher than the original class.

B. Detection and Tracking of Traffic Participants

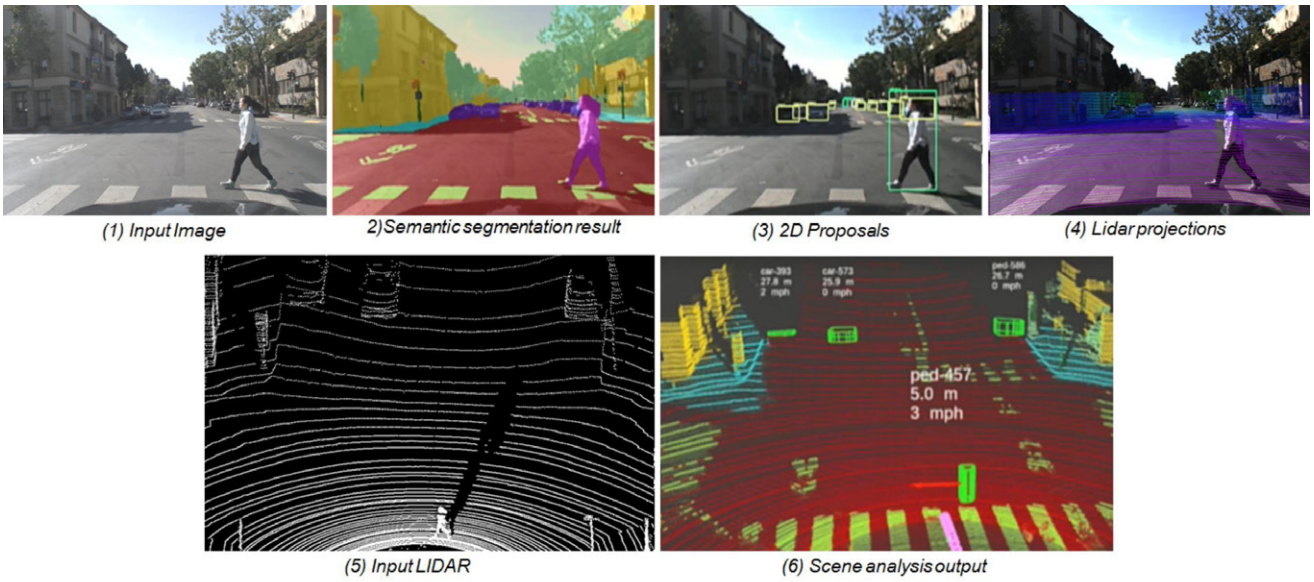
1) Frustum Proposal Generation: To track traffic participants, we generate 3D object candidates for each frame and associate them temporally. A 3D object candidate is represented as a cluster of 3D points. We obtained 3D object candidates by the Frustum-Based PointNet [13] approach, which is discussed in the following.

2D Object Detection. We leverage a state-of-the-art instance segmentation algorithm, Mask R-CNN [14], for 2D object detection. Please refer to [14] for the detailed introduction of Mask R-CNN. We train Mask R-CNN on Cityscapes dataset [24] using mini-batch SGD with the following parameters. A patch size is set to 1024×1024 (randomly cropped from the original image, padded with zeros to maintain aspect ratio) and the minibatch size is set to be 2 in a single GPU. The model is trained for 24k iterations, starting from a learning rate of 0.01 and reducing it to 0.001 at 18k iterations. We evaluate our model on Cityscapes dataset and achieve a mean AP of 32.0% on the validation set (similar to [14]).

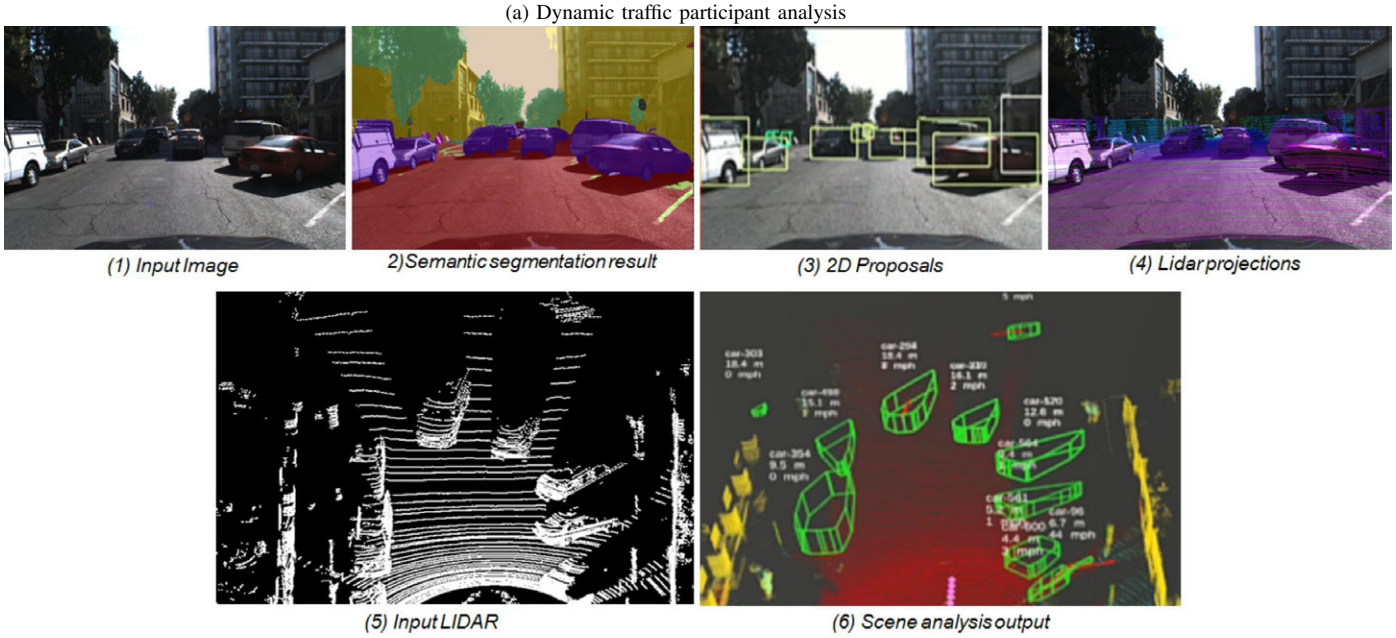
We use the following three classes, *Car*, *Person*, *Cyclist* for our dataset. We train Mask R-CNN using the same training protocol on 3000 images with instance segmentation annotation, where we have around 20k *Car*, 17k *Person*, 3k *Cyclist* instances. The smallest box encapsulates all the pixels of a mask is defined as the corresponding bounding box The final output is further refined by using non-maximum suppression with a threshold of 0.5.

Given a 2D bounding box obtained from Mask R-CNN, we form a *frustum point cloud* [13] by collecting projected 3D points (given a known camera project matrix) that fall into the region of a 2D bounding box.

3D Instance Segmentation. Given a *frustum point cloud* as the search space of a 3D object, it is easier to segment a foreground object from background clutters as they are usually well separated in 3D than in 2D. However, it is still hard to design simple methodology to cluster the object as the shape will be very different under different scenarios (object category, distance, and angle). Here we follow the same approach as in [13] to realize 3D instance segmentation by classifying each point in the frustum with a PointNet-based network [27]. The network takes point cloud in the



(a) Dynamic traffic participant analysis



(b) Motion planning and driver behavior

Fig. 6: Two sample results of the proposed framework tested in different driving situations. As shown in Figure 6 (a), the framework is able to provide the position, orientation and velocity of the pedestrian. Through the analyzed results, we can infer the pedestrian is crossing the intersection. Figure (b) is an example of how the pipeline outputs can be used for motion planning in a complicated driving scenario by avoiding static and dynamic vehicles.

frustum and predicts a probability score for each point that indicates how likely the point belongs to the foreground. A threshold 0.5 is chosen in our implementation to generate the final 3D object candidate. The network is trained with the same protocol as in [13] on KITTI [28] dataset for 20 epochs, which takes 5 hours. An example result of 3D instance segmentation is shown in Figure 5.

2) Tracking and Velocity Estimation: To track 3D objects, we leverage a similar framework as outlined in [29] with the following two modifications. First, we use a class-specific approach to achieve better tracking performance since 3D appearance of these classes varies substantially and hence require unique parameters. Second, we improve the motion estimation approach in [29] by integrating the velocity estimation component from [30], which leverages 3D shapes, color and motion cues to obtain robust velocity estimation.

The overall tracking algorithm is described as follows. Given a set of 3D object candidates in a global reference

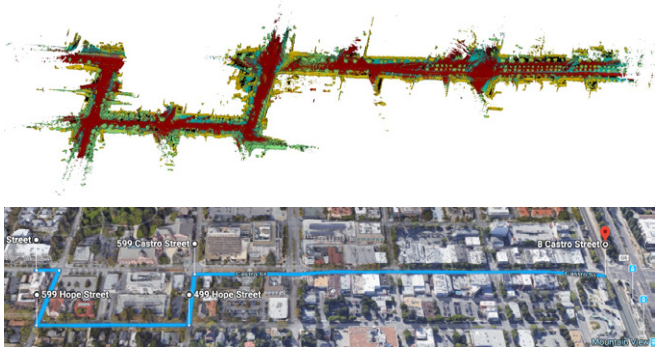


Fig. 7: Example of semantic map of Castro street, Mountain View, California created using the pipeline. The semantic map has been constructed after removing dynamic objects from the scene. Note that the bottom image is captured from Google Map.

frame, if no prior tracks are identified, we first initialize each 3D candidate as new a track (Kalman Filter initialization step). If prior tracks exist, we calculate the corresponding convex hull and centroid of each 3D candidate and determine the associated track in the prior tracks by using a simple yet effective distance criteria (matching distance threshold). If we cannot find a nearest track for a certain number of frames (life span), the 3D candidate is set to be a new track. If a nearest track is found, we update the tracker and assign the 3D candidate with the corresponding track ID (Kalman Filter update step). Finally, for all remaining tracks that are not associated with 3D object candidates, we predict a possible location of the 3D object candidate using the prior velocity estimate (Kalman Filter predict step). Moreover, a matching distance threshold parameter is used to determine when the tracks should be matched. The above parameters are used in a class-specific manner such that each class of object - car, pedestrian and cyclist - can be tracked robustly.

IV. DATA COLLECTION PLATFORM AND EXPERIMENTAL RESULTS

The data was collected using an instrumented vehicle equipped with the following sensors:

(i) 3 x Point Grey Grasshopper 3 video camera, resolution: 1920 1200 pixels, frame rate: 30Hz, field of view (FOV): 80 degrees x 1 (center) and 90 degrees x 2 (left and right).

(ii) 1 x Velodyne HDL-64E S2 3D LiDAR sensor, spin rate: 10 Hz, number of laser channel: 64, range: 100 m, horizontal FOV: 360 degrees, vertical FOV: 26.9 degrees.

All sensors on the vehicle were logged using a PC running Ubuntu Linux 14.04 with two eight-core Intel i5-6600K 3.5 GHz Quad-Core processors, 16 GB DDR3 memory, and a RAID 0 array of four 2TB SSDs, for a total capacity of 8 TB. The sensor data are synchronized and timestamped using ROS and a customized hardware and software design for multimodal data analysis.

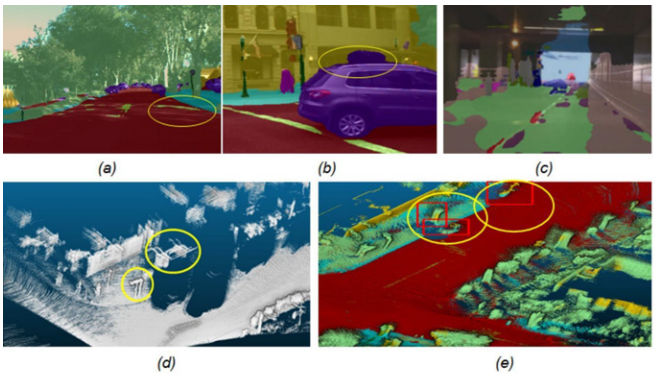


Fig. 8: Examples of challenges. **(a)** Shadows causes misclassification in semantic segmentation. **(b)** Imperfect segmentation quality causes projection errors. **(c)** Domain shift causes supervised learning algorithms fail in unseen environment. **(d)** A duplication of objects is presented due to SLAM inefficiency. **(e)** Defects in semantic mapping causes by a combination of LiDAR SLAM and projection issues.

Each module in the pipeline is built into a ROS node for easier communication and synchronized process. The input and output of each module are sent through ROS messages, and the final output message of dynamic object tracking and semantically labeled map are visualized and validated using RViz. We validate the proposed framework using data collected from the aforementioned platform in San Francisco and Bay Area, California. The experimental results are shown in Figure 6 (a) and (b). An example of a semantically labeled map of Castro street, Mountain View, CA is demonstrated in Figure 7.

While the framework successfully demonstrates its capability in challenging urban driving scenarios. However, several limitations are found and described as follows. First, the misclassification caused due to semantic segmentation and object detection can impair the quality of the framework as shown in Figure 8 (a), (b) and (c). Ability to adapt to new scenarios and locations involve better generalization of a learning algorithm, i.e., avoid over fitting to the training sequences. Second, in a real driving scene, the presence of occlusions affects tracking and velocity estimation. It is crucial to develop a long term tracking system with occlusion handling. Third, the point cloud registration via LiDAR SLAM is not perfect due to lack of features and presence of dynamic objects. This creates artifacts in the mapping process as shown in Figure 8 (d) and (e).

V. FUTURE WORKS

Intention and Motion Prediction of Surrounding Entities. Intention and motion prediction of road users is a crucial step to achieve safe and high-quality decision-making and motion planning for intelligent transportation systems. Recent works on probabilistic intention and motion prediction of vehicles used bird's-eye view motion dataset, NGSIM [31], to train and evaluate learning models, in which

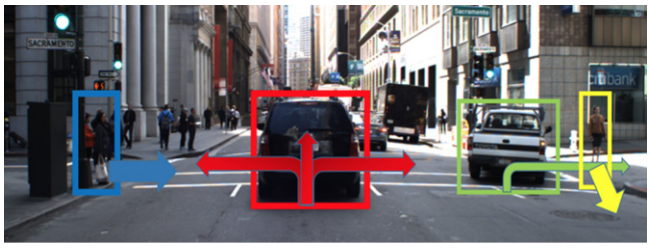


Fig. 9: Traffic participants intention and motion prediction.

the surroundings of the predicted vehicle were fully observable. However, such observability is not always available from a decision maker perspective as shown in Figure 9. The situation in front of the vehicle in the red box cannot be observed due to the occlusion, which may significantly impact the behavior of the vehicle. Including such uncertainties is very crucial to completely understand surrounding entities. In the absence of complete observability, we require large scale datasets to learn how to tackle such problems.

KITTI dataset [32] is one such dataset which can be used for studying motion prediction from a decision maker perspective. However, the interactions between road participants are relatively rare in the dataset, which makes it not suitable for such purpose. Therefore, a large dataset with amounts of interaction among road participants need to be constructed. However, it is known to be challenging to collect and annotate a large scale dataset. The proposed framework can be utilized to provide automatic annotation with human in the loop to construct a large dataset to study intention and motion prediction.

Decision Making and Planning with Imperfect Perception. Enormous works were focused on decision-making and motion planning with perfect perception. When deployed in real driving scenes, such decision and planning modules may lead to fatal consequences since perfect perception can never be achieved. Uncertainties in the perception module should be retained, and we need to design algorithms for decision-making and planning with imperfect perception accordingly. A recent work by Jha et al [33] addressed perception uncertainty. The proposed framework can provide exemplar uncertainties from perception modules to facilitate the design of decision and planning modules with imperfect perception.

REFERENCES

[1] N. University, "Autoware," 2014, [Online; accessed 23-January-2018]. [Online]. Available: https://www.autoware.ai/

[2] Baidu, "Apollo," 2017, [Online; accessed 23-January-2018]. [Online]. Available: http://apollo.auto/

[3] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele, "Monocular Visual Scene Understanding: Understanding Multi-object Traffic Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, 2013.

[4] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3D Traffic Scene Understanding from Movable Platforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.

[5] S. Song and M. Chandraker, "Joint SFM and Detection Cues for Monocular 3D Localization in Road Scenes," in *CVPR*, 2015.

[6] S. Wang, S. Fidler, and R. Urtasun, "Holistic 3D Scene Understanding from a Single Geo-tagged Image," in *CVPR*, 2015.

[7] A. Osep, W. Mehner, M. Mathias, and B. Leibe, "Combined image- and world-space tracking in traffic scenes," in *ICRA*, 2017.

[8] W. Choi and S. Savarese, "Multi-target Tracking in World Coordinate with Single, Minimally Calibrated Camera," in *ECCV*, 2010.

[9] J. Yao, S. Fidler, and R. Urtasun, "Holistic Scene Understanding," in *ICCV*, 2013.

[10] Y. Xiang and D. Fox, "DA-RNN: Semantic Mapping with Data Associated Recurrent Neural Networks," in *RSS*, 2017.

[11] C. Hne, C. Zach, A. Cohen, and M. Pollefeys, "Dense Semantic 3D Reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1730–1743, 2017.

[12] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D Reconstruction in Real-time," in *IV*, 2011.

[13] W. Qi, Charlesand Liu, C. Wu, H. Su, and L. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," *arXiv preprint arXiv:1711.08488*, 2017.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-cnn," in *ICCV*, 2017.

[15] I. A. Barsan, P. Liu, M. Pollefeys, and A. Geiger, "Robust Dense Mapping for Large-Scale Dynamic Environments," *ICRA*, 2018.

[16] N. Lee, W. Choi, P. Vernaza, C. Choy, P. Torr, and M. Chandraker, "DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents," in *CVPR*, 2017.

[17] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *RSS*, 2014.

[18] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *IJRR*, vol. 26, no. 9, 2007.

[19] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *arXiv preprint arXiv:1711.06396*, 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *CVPR*, 2017.

[23] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *arXiv preprint arXiv:1606.00915*, 2016.

[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *CVPR*, 2016.

[25] Q. Zhang and R. Pless, "Extrinsic Calibration of a Camera and Laser Range Finder (Improves Camera Calibration)," in *IROS*, 2004.

[26] F. Vasconcelos, J. P. Barreto, and U. Nunes, "A Minimal Solution for the Extrinsic Calibration of a Camera and a Laser-range Finder," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2097–2107, 2012.

[27] C. Qi, H. Su, K. Mo, and L. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *CVPR*, 2017.

[28] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *CVPR*, 2012.

[29] P. Morton, B. Douillard, and J. Underwood, "An Evaluation of Dynamic Object Tracking with 3D Lidar," in *ACRA*, 2011.

[30] D. Held, J. Levinson, S. Thrun, and S. Savarese, "Combining 3D Shape, Color, and Motion for Robust Anytime Tracking," in *RSS*, 2014.

[31] V. Alexiadis, J. Colyar, J. Halkias, R. Hranac, and G. McHale, "The Next Generation Simulation Program," *Institute of Transportation Engineers. ITE Journal*, vol. 74, no. 18, pp. 22–26, 2004.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[33] S. Jha, V. Raman, D. Sadigh, and S. Seshia, "Safe Autonomy Under Perception Uncertainty Using Chance-Constrained Temporal Logic," *Journal of Automated Reasoning*, vol. 60, no. 1, pp. 43–62, 2018.