LiDAR Fusion and Dynamic Object Masking for ORB-SLAM2

Yi-Ting Hsiao, Ting Wei Li, Ke Liu, Joshua Symonds, Sibo Wang (Team 27)

***Abstract*— When using SLAM (Simultaneous Localization and Mapping) to navigate an autonomous robot in an unknown environment, it is generally a prerequisite that the environment is static. This is problematic since mobile robots nearly always operate in dynamic environments. This is because when SLAM attempts to localize, it assumes the world is modeled accurately by its internal map which was generated with previous data. By using a 3D object detection algorithm, we can identify key objects in our environment that may be dynamic and create more accurate maps that minimize the impact of moving objects. By purging objects that we know to be mobile - such as humans and cars - from the data we use in SLAM to generate the map and localize, we will no longer be trying to localize using stale information stored in our map. Filtering mobile objects from the environment, can improve localization accuracy while running SLAM in dynamic environments and create a map that better reflects the mobile robot's environment. Further, using depth prediction sensor fusion techniques we can better represent the distances of objects in the environment, which allows for even better quality of information used by SLAM to construct its map and localize the mobile robot's position, especially at high speeds.**

I. INTRODUCTION

SLAM, or Simultaneous Localization and Mapping, is a critical process for mobile robots to navigate through unknown environments. Visual SLAM is an algorithm that accomplishes this with information from a camera. A typical visual SLAM model extracts features and tracks them over key frames, which are then used to optimize the camera pose over time and build a static map of the environment. One of the most well-known benchmark models is ORB-SLAM and its improved version, ORB-SLAM2 [1].

However, as a visual SLAM model, ORB-SLAM2 is limited to taking input in the form of monocular, stereo, or RGB-D data, which can constrain its performance in complex environments. To address this, we propose to enhance ORB-SLAM2 by integrating a depth-fusion model that can map LiDAR to pixel depth. Additionally, we propose to integrate a bidirectional 3D object detection model that can mask dynamic objects during the feature selection process, further improving ORB-SLAM2's performance.

II. RELATED WORKS

A. SLAM Models

ORB-SLAM2 is a visual SLAM system that builds upon the original ORB-SLAM model [1]. Specifically, ORB-SLAM2 has loop closure detection, robust feature extraction and matching algorithm based on ORB, and multi-threaded implementation allowing real-time performance. ORB-SLAM2 has become a popular benchmark for visual SLAM systems due to its robustness, efficiency, and accuracy. Thus, we have chosen ORB-SLAM2 for implementing modifications and comparing results.

Capturing dynamic objects is an essential aspect of improving SLAM performance, and current research is actively exploring this area. One of the most well-known benchmark models for this task is DynSLAM [2], which reconstructs the static background, moving objects, and potentially moving but currently stationary objects separately. To achieve this, DynSLAM uses instance-aware segmentation and sparse scene flow to classify objects, which improves camera pose and map reconstruction.

While DynSLAM is an effective and advanced method, for the sake of simplicity, our model focuses only on semantic segmentation of all moving and potentially-moving objects. The approach still enables us to improve the accuracy of the SLAM system by masking out dynamic objects and ensuring landmarks in ORB-SLAM2 are static. However, future work could explore the separation of potentially-moving objects, such as parked vehicles, and continuous tracking of moving objects to further enhance the performance of our model.

B. 3D Object Detection Models

Self-driving vehicles require an accurate understanding of their surrounding environment to operate reliably, and object detection is the fundamental function of the perception system[3]. However, most object detection models only focus on 2D monocular images [4] [5]. These models may have high performance, yet they fail to provide the depth information of these objects. In contrast, 3D object detection provides the third dimension to reveal more accurate location information.[3]

To achieve such information, some perception models [6] [7] adopted stereo cameras as the sensor. Yet these models are computationally expensive in depth estimation and perform poorly with textureless regions, during nighttime, or with limited FoV. Others utilize LiDAR point-cloud [8] [9], yet point-cloud data lacks texture information. Finally, there are also fusion-based models [10] [11] [12] that utilize sensor fusion techniques to map point-cloud and image data for better accuracy. However, cross-modal integration without losing information has always been a challenging task. Some fusion-based models incorporate multiple fusion stages to reduce information loss [13] [14], whereas others incorporate virtual or pseudo-point-based 3D object that seamlessly fuses RGB images and LiDAR data by depth completion [15] [16] [17].

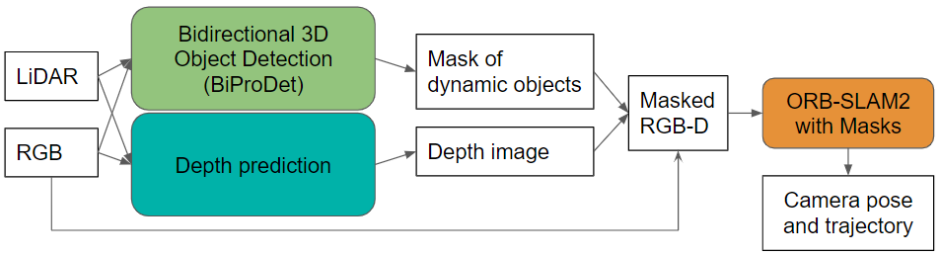


Fig. 1: System architecture overview

C. Depth Fusion

Depth Fusion is necessary in many cases such as autonomous driving, robot navigation. Modern techniques would generate RGB images and LiDAR point clouds for mapping and localization, and depth map is helping to improve prediction accuracy in tasks like object detection, 3D point clouds reconstruction. There are several ways to implement depth map generation. Firstly, it is common to predict depth using stereo images. For example, the stereoBM opencv model and a MATLAB algorithm were developed to construct depth map using two static images[18]. Their method includes finding the disparity between two matching points in images. Then, the depth value is generated, which is inversely proportional to the difference in distance of the corresponding points.

Another approach to generate depth map is combining images with LiDAR point clouds. While RGB image can perceive the texture and color, the LiDAR sensor data contains the depth information and is not impacted by poor lighting conditions. The article focusing on the fusion of LiDAR and Monocular Vision provides a suitable algorithm for depth interpolation and data fusion[19]. This algorithm can filter the sparse point clouds to dense point clouds, and project points from 3D to a 2D depth map. In our methodology, with pre-processed 3D LiDAR point clouds and RGB as our input, we utilize this algorithm to generate RGB-D map for mapping and localization.

III. Methodology

We first introduce the overall proposed architecture, shown in Figure 1. With monocular and LiDAR input, we first use a bidirectional 3D object detection model, BiProDet, to produce pixel-level labels for dynamic objects. Meanwhile, we also predict the depth of each pixel based on point cloud projection. Then, we combine the outputs to form masked RGB-D data. We then modify the ORB-SLAM2 code base such that pixels that are masked as dynamic objects are not considered in key-point selection. Finally, we analyze the produced camera pose and trajectory.

A. Object Detection with BiProDet

Inspired by Bidirectional Propagation For Cross-Modal 3D Object Detection [13], or BiProDet, our perception algorithm involves an image pipeline and a point cloud pipeline that learn feature representations from RGB images and LiDAR point clouds respectively. As shown in Figure 2, point-to-pixel and pixel-to-point propagation methods were adopted to allow features to flow between the point cloud branch and the image branch in a multi-stage fashion throughout the feature-learning process. This bidirectional feature propagation approach makes use of the potential of the image branch to enhance the expressive power of the point cloud backbone network. Additionally, normalized local coordinate (NLC) map estimation is employed to promote the learning of rich semantic and spatial representations. These tasks are necessary to complement the sparse spatial representation extracted from point clouds, especially for distant or highly occluded cases. An example of BiProDet identifying 3D objects in its environment can be seen in Figure 3, where 4 cars are identified and surrounded in a green bounding box. Eventually, based on the semantic segmentation output from the image backbone, the dynamic objects will be masked-out and the remaining data will be passed down to ORB-SLAM2.

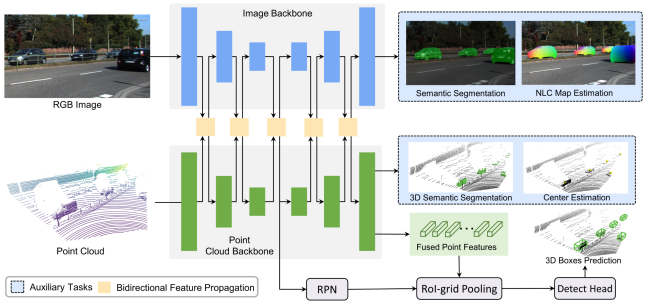


Fig. 2: BiProDet System Architecture

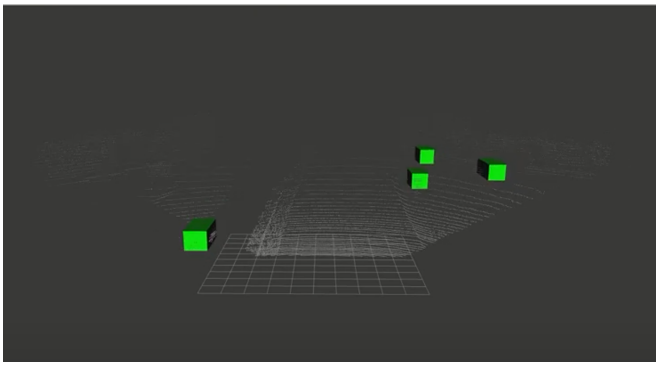


Fig. 3: Bounding Boxes of Dynamic Objects in BiProDet

B. Masking Data with Semantic segmentation

In order for ORB-SLAM2 to be able to properly understand our filtered data, we must also identify which pixels in the RGB image correspond to dynamic objects. This information is a direct byproduct of BiProDet and we are able to extract which pixels represent data we want to filer out from the data being given to ORB-SLAM2. This can be seen in Figures 4 and 5 where we filter out the pixels with dynamic objects from the image, shown by coloring them green.

Fig. 4: Input image with mobile objects

Fig. 5: Output image with mobile objects removed

C. Data Fusion for RGB-D map

In most cases, the ORB-SLAM2 model is able to take monocular data and perform well[20]. Since this model can pre-process the input and generates the salient keypoint locations' features, it's monocular vision system is independent of stereo or RGB-D sensor. However, taking monocular-only inputs will have a delayed initialization, which is not suitable for situation of more challenging outdoor scenarios with fast moving camera. Therefore, it's necessary to obtain a dataset with depth information, so that it can generate point clouds directly from each frame, which is critical for not losing tracking in a high speed camera movement. In this regard, we will generate depth map frame by frame using LiDAR point clouds and RGB image provided by the KITTI odometry dataset, and then compare the newly generated camera trajectory to the trajectory generated by only monocular vision.

In the first step, we need to filter the sparse LiDAR points from the point cloud clouds. The LiDAR point clouds is in the format $V_i = (X, Y, Z, r)$, where X, Y, Z are the 3D coordinates and r is the reflectance value. The coordinates with positive reflectance value is selected and being projected into the 2D camera frame $p_i = (u, v, z)$ by equation 1. Inside this function, the i^{th} 3D point V_i is pre-multiplied by $Tr_{velo\_to\_cam}$ and P_j, where P_j indicates the 3x4 projection matrices after rectification for j^{th} camera. Then, the projected point p_i will be normalized by dividing the last element of the vector z. After projecting the 3D point clouds data into the camera frame, we need to implement depth interpolation to form the depth map, where we form a KDTree using the point clouds, and assign the depth value of a pixel using the k nearest neighbors' weighted sum.

$$p_i = \frac{1}{z} \cdot P_j \cdot Tr_{velo\_to\_cam} \cdot V_i \tag{1}$$

An example of our generated depth maps can be seen in Figure 6 and Figure 7. Figure 6 was the input monocular image. In the depth image, objects appear closer to the camera as the colors shift towards purple, while objects that are farther away are represented by colors that shift towards yellow-green. Experimentally, lowering the hyper-parameter k has yielded a better result of depth mapping, shown in Figure 7.

Fig. 6: Input image for depth mapping

Fig. 7: Visualization of result of depth mapping

D. Processing ORB-SLAM2 with masked dataset

Here we briefly introduce the high-level idea of how ORB-SLAM2 works on RGB-D datasets. ORB-SLAM2 will detect and track features in the RGB images using the Oriented FAST and Rotated BRIEF (ORB) algorithm. After that, given the depth map we generated from our depth interpolation algorithm, ORB-SLAM2 can estimate the depth of the detected ORB features and project those points into a 3D map. By comparing the current frame with the previous frame and the map, ORB-SLAM2 can estimate the camera's pose. The additional depth information help improve the accuracy of the pose estimation. Another critical stage of ORB-SLAM2 is loop closure detection, where ORB-SLAM2 detects when the camera revisits a previously seen location and improves the map and pose estimation.

We modified the source code of ORB-SLAM2 to accommodate the 2D semantic mask. Given the output mask from BiProDet, we mainly modify the files: `Frame.cc` and `ORBextractor.cc` to take an additional sequence

of binary mask images as input and exclude the key points within the area of objects. Specifically, for each frame, we filtered the key points generated by the function: `DistributeOctTree` by the binary values obtained from the mask corresponding to each pixel in this frame.

IV. Results

Our new SLAM architecture[1] was applied to the KITTI dataset [21], which we believe is an ideal use case for our model, as it can effectively reflect the dynamic surroundings encountered in self-driving cars. The dataset consists of sequential sets of LiDAR geometric data and color images from stereo camera, which serve as the input data, and a ground truth position that we can use to evaluate our performance. Specifically, we have chosen sequence 07, which has a loop closure and many dynamic objects (pedestrians, cyclists, cars, etc.).

1) Data Masking Accuracy: Before integrating to overall pipeline, we have first evaluated the semantic segmentation model. Accurate semantic segmentation is crucial since failure to identify a moving object accurately can cause us to try to localize with respect to unreliable landmarks and identifying static objects as dynamic removes possible landmark candidates and can decrease ORB-SLAM2's performance. To evaluate our masking accuracy, we used the intersection over union of our classifications against a ground truth classification. Further, we visualized our classification accuracy for each image, which can be seen in Fig. 8, where green represents a correct classification, blue represents a missed classification, and red represents an erroneous classification. Our total IoU was 0.73 which we were content with. For simple images such as one with only a few nearby cars, our IoU was generally about 0.90, but we struggled to identify objects too far or near as well as trucks and vans which significantly decrease our performance.

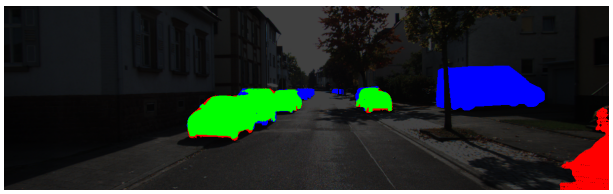


Fig. 8: Masking accuracy

2) Localization Performance (Masking Only): We have evaluated the localization performance of our model based on the 07 sequence from the KITTI visual odometry dataset [21]. To first evaluate the performance of the masking mechanism, we have compared monocular with mask against the baseline of monocular only. Figure 9 compares the camera trajectories we generated to the ground truth trajectory (in black dotted line), where green is based on only monocular input, while blue is based on monocular and dynamic object mask. As shown, adding mask to the input data has helped ORB-SLAM2 to localize, especially when turning at intersections, where there are more moving objects in the scene. The quantitative analysis is based on the widely adopted absolute pose error (APE) metric [1]. The masking has reduced the root mean squared APE from 2.54 to 2.32. Table I lists more statistics on the APE analysis.

[1] `https://gitlab.eecs.umich.edu/jsymonds/semantic_slam/`

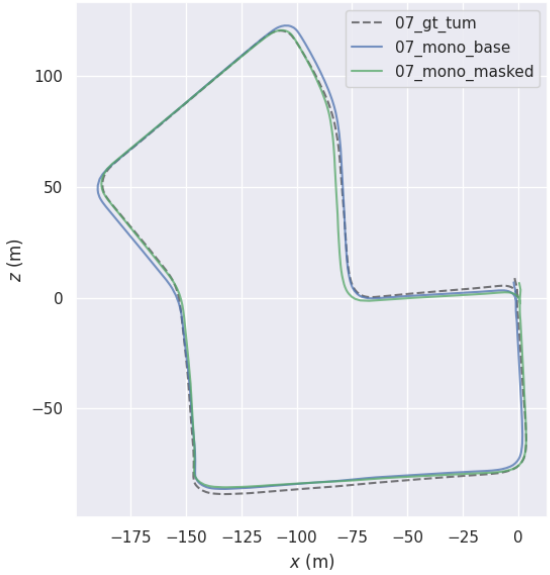


Fig. 9: Comparisons of camera trajectory in KITTI 07

	Mono.	Stereo	Mono. + Mask	Mono. + DP	Mono. + DP + Mask
RMSE	2.54	0.45	**2.32**	**0.456**	**0.43**
mean	2.28	0.41	**2.02**	**0.42**	**0.41**
min	0.29	0.08	**0.13**	**0.04**	**0.09**
max	5.01	0.81	**4.08**	**0.80**	**0.68**

TABLE I: Statistics of absolute pose error (APE)

Figures 10 shows the detailed comparison of the camera pose along the trajectory path. As seen, the mask has improved the position accuracy, especially along the y-direction.

3) Localization Performance (With Depth Prediction): We then evaluated the performance of depth prediction and our overall model. As shown in the Figure 11, the trajectory of with DP (depth prediction) is much closer aligned to the ground truth than the monocular-only baseline. Quantitatively, the RMSE APE is improved to 0.456, which largely outperforms the monocular baseline, and performs comparatively to the stereo baseline, as shown in Table I. Figures 12 provides a more detailed error evaluation for both DP and our final model, DP + mask.

To further evaluate the performance, we select the stereo input as another benchmark, since it also incorporate depth information. From Table I, with stereo images as input, ORB-SLAM2 has about 0.45 RMSE APE. Comparatively, our final model of monocular with depth prediction and masking has 0.43 RMSE APE, which shows a slight improvement.

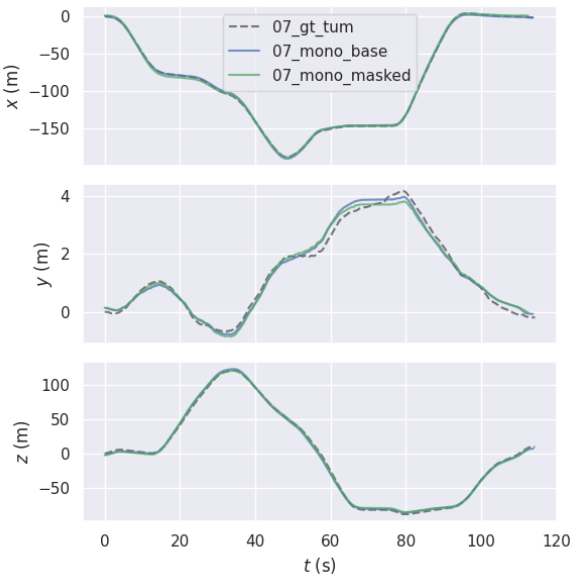


Fig. 10: Comparisons of pose positions along the path

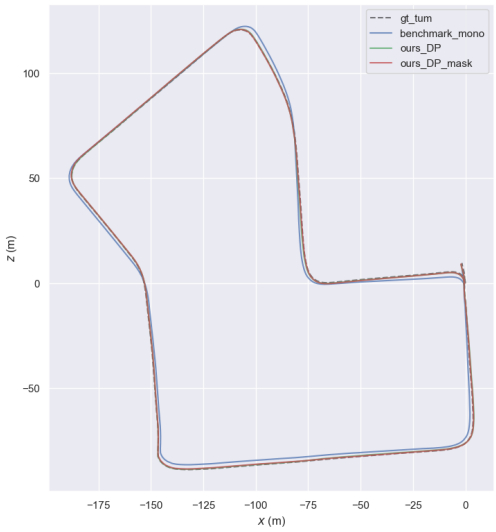
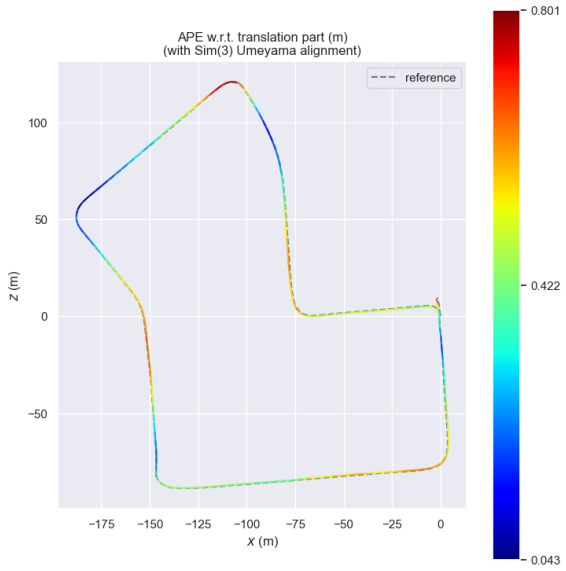
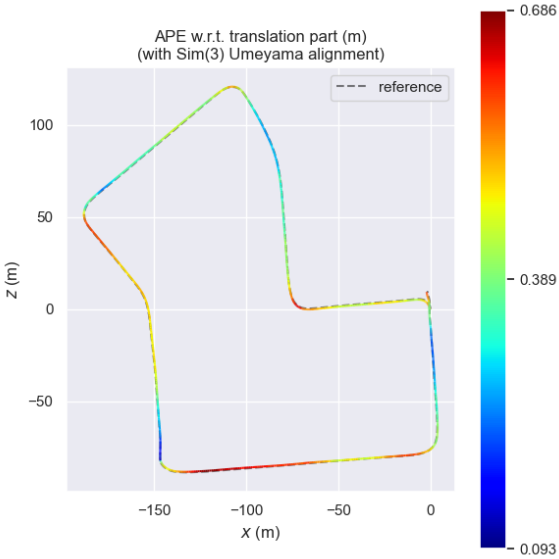


Fig. 11: Comparisons of camera trajectory in KITTI 07 with DP input and DP+Mask input



(a) Camera trajectory colored by APE of monocular + depth prediction



(b) Camera trajectory colored by APE of monocular + depth prediction + masking

Fig. 12: The predicted camera trajectory with depth prediction and masking

V. Conclusion

In this study, we presented a modification to the ORB-SLAM2 model to improve its performance in dynamic environments. By integrating a depth prediction sensor-fusion model and a 3D object detection algorithm, our model can better represent the distances of objects in the environment and identify key objects that may be dynamic and remove them from the data used in SLAM. Our results demonstrated that our modified model outperforms traditional ORB-SLAM2 in terms of localization accuracy in dynamic environments.

One of the main contributions of our work is the integration of a depth-fusion model, which allowed us to capture richer information about the environment. By fusing data from camera and LiDAR sensors, we were able to obtain a more accurate representation of the distances of objects in the environment. This was particularly useful in dynamic environments, where the observer may be moving and their distance from keypoints in the environment may change rapidly. Our results showed that the use of a depth-fusion model led to generally improved localization accuracy, particularly in scenarios where traditional ORB-SLAM2 struggled like navigating an intersection with many other moving vehicles.

Another important contribution of our work is the integration of a 3D object detection algorithm, which allowed us to identify key objects in the environment that may be dynamic and remove them from the input data. This was particularly useful in scenarios where there were many dynamic objects in the scene, such as in urban environments with heavy traffic. By explicitly removing these objects from the input data, we were able to improve the accuracy of the SLAM system and reduce the number of errors and ensure that no keypoints used for localizing were assigned to unreliable landmarks. This improved allowed for us to consistently outperform ORB-SLAM2's benchmark both regarding monocular and depth-informed localization.

Our results showed that the proposed modification is particularly effective in scenarios where the autonomous vehicle is maneuvering through an intersection which is a particularly crucial part of autonomous driving, where accurate localization is essential to ensure the safety of passengers and other road users. There are many other applications where the work presented here may be useful such as localization for drones - which tend to move very quickly and abruptly change direction, and collaborative autonomous systems - where the observer will frequently be interacting with several mobile agents in the environment.

In conclusion, our study demonstrated that the integration of a depth-fusion model and a 3D object detection algorithm can significantly improve the performance of ORB-SLAM2 in dynamic environments. The proposed modification has the potential to enhance the performance of visual SLAM systems, making them more reliable for autonomous robots to navigate through complex environments. Future work could focus on further improving the accuracy of the sensor-fusion depth prediction model and the object detection algorithm, and testing the modified model in various other real-world scenarios.

VI. Future Work

There are several directions that this project can be expanded to further improve its performance. Here we will outline a few ideas that we feel would be good extensions of our work.

1) One extension of our work would be attempting to implement our work in this paper to other scopes where it may be helpful such as localization for controlling drones or in collaborative autonomous systems, where the ability to localize at high speeds and in dynamic environments is uniquely important.
2) A very logical extension of our work here is that instead of fully masking out dynamic objects in our environment, we track their speeds and estimate where they may be in the future. This would allow for us to still use the information from all objects in our background without making the risky assumption that they're static. Further, this would allow for us to still use the information from objects that could be dynamic but don't happen to be moving - such as a parked car - and use that information to help localize.
3) Another idea for future work would be to identify objects in the environment that are static as well and use them to construct a semantic map of the environment, then use that information to localize better. An example of this within autonomous driving would be identifying trees and lampposts along the side of the road and tracking them as landmarks. Encoding the map with rich information about the specific objects in the environment may improve localization accuracy and build a more meaningful map.
4) We also see potential improvement to our depth prediction model that could be made by using machine learning to predict the depth of objects in the environment instead of using direct computation.

REFERENCES

[1] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[2] I. A. Bârsan, P. Liu, M. Pollefeys, and A. Geiger, "Robust dense mapping for large-scale dynamic environments," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[3] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[4] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[5] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[6] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[7] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 922–928.

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[10] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[11] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.

[12] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[13] Y. Zhang, Q. Zhang, J. Hou, Y. Yuan, and G. Xing, "Bidirectional propagation for cross-modal 3d object detection," *arXiv preprint arXiv:2301.09077*, 2023.

[14] X. Li, T. Ma, Y. Hou, B. Shi, Y. Yang, Y. Liu, X. Wu, Q. Chen, Y. Li, Y. Qiao *et al.*, "Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion," *arXiv preprint arXiv:2303.03595*, 2023.

[15] X. Wu, L. Peng, H. Yang, L. Xie, C. Huang, C. Deng, H. Liu, and D. Cai, "Sparse fuse dense: Towards high quality 3d detection with depth completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5418–5427.

[16] H. Zhu, J. Deng, Y. Zhang, J. Ji, Q. Mao, H. Li, and Y. Zhang, "Vpfnet: Improving 3d object detection with virtual point based lidar and stereo data fusion," *IEEE Transactions on Multimedia*, 2022.

[17] H. Wu, C. Wen, S. Shi, X. Li, and C. Wang, "Virtual sparse convolution for multimodal 3d object detection," *arXiv preprint arXiv:2303.02314*, 2023.

[18] A. Aslam and M. S. Ansari, "Depth-map generation using pixel matching in stereoscopic pair of images," 2019.

[19] L. Lou, Y. Li, Q. Zhang, and H. Wei, "Slam and 3d semantic reconstruction based on the fusion of lidar and monocular vision," *Sensors*, vol. 23, no. 3, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/3/1502

[20] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, oct 2017. [Online]. Available: https://doi.org/10.1109%2Ftro.2017.2705103

[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[22] A. Thyagharajan, B. Ummenhofer, P. Laddha, O. J. Omer, and S. Subramoney, "Segment-fusion: Hierarchical context fusion for robust 3d semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1236–1245.

[23] X. Wu, L. Peng, H. Yang, L. Xie, C. Huang, C. Deng, H. Liu, and D. Cai, "Sparse fuse dense: Towards high quality 3d detection with depth completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 5418–5427.

[24] H. Zhu, J. Deng, Y. Zhang, J. Ji, Q. Mao, H. Li, and Y. Zhang, "Vpfnet: Improving 3d object detection with virtual point based lidar and stereo data fusion," *IEEE Transactions on Multimedia*, pp. 1–14, 2022.

[25] H. Yang, Z. Liu, X. Wu, W. Wang, W. Qian, X. He, and D. Cai, "Graph r-cnn: Towards accurate 3d object detection with semantic-decorated local graph," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 662–679.

[26] "Papers with code - semantickitti benchmark (3d semantic segmentation)." [Online]. Available: https://paperswithcode.com/sota/3d-semantic-segmentation-on-semantickitti

[27] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1689–1696, 2019.

[28] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2dpass: 2d priors assisted semantic segmentation on lidar point clouds," *ArXiv*, vol. abs/2207.04397, 2022.

[29] R. Cheng, R. Razani, E. M. Taghavi, E. Li, and B. Liu, "(af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 542–12 551, 2021.

[30] H. Yang, Z. Liu, X. Wu, W. Wang, W. Qian, X. He, and D. Cai, "Graph r-cnn: Towards accurate 3d object detection with semantic-decorated local graph," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*. Springer, 2022, pp. 662–679.