01 Jan 2023

Aerial LiDAR-based 3D Object Detection And Tracking For Traffic Monitoring

Baya Cherif

Hakim Ghazzai

Ahmad Alsharoa
Missouri University of Science and Technology, aalsharoa@mst.edu

Hichem Besbes

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/5087

Recommended Citation

B. Cherif et al., "Aerial LiDAR-based 3D Object Detection And Tracking For Traffic Monitoring," *Proceedings - IEEE International Symposium on Circuits and Systems*, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.1109/ISCAS46773.2023.10181371

Aerial LiDAR-based 3D Object Detection and Tracking for Traffic Monitoring

Baya Cherif[1,2], Hakim Ghazzai[1], Ahmad Alsharoa[2], Hichem Besbes[3], and Yehia Massoud[1]
[1]Innovation Technologies Laboratories, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[2]Missouri University of Science and Technology, Rolla, Missouri, USA
[3]Higher School of Communications of Tunis, University of Carthage, Tunis, Tunisia

2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10181371

***Abstract*—The proliferation of Light Detection and Ranging (LiDAR) technology in the automotive industry has quickly promoted its use in many emerging areas in smart cities and internet-of-things. Compared to other sensors, like cameras and radars, LiDAR provides up to 64 scanning channels, vertical and horizontal field of view, high precision, high detection range, and great performance under poor weather conditions. In this paper, we propose a novel aerial traffic monitoring solution based on Light Detection and Ranging (LiDAR) technology. By equipping unmanned aerial vehicles (UAVs) with a LiDAR sensor, we generate 3D point cloud data that can be used for object detection and tracking. Due to the unavailability of LiDAR data from the sky, we propose to use a 3D simulator. Then, we implement PointVoxel-RCNN (PV-RCNN) to perform road user detection (e.g., vehicles and pedestrians). Subsequently, we implement an Unscented Kalman filter, which takes a 3D detected object as input and uses its information to predict the state of the 3D box before the next LiDAR scan gets loaded. Finally, we update the measurement by using the new observation of the point cloud and correct the previous prediction's belief. The simulation results illustrate the performance gain (around 8%) achieved by our solution compared to other 3D point cloud solutions.**

***Index Terms*—Traffic monitoring, deep learning, detection, tracking, UAV, LiDAR**

I. Introduction

The global population is expected to reach around 8.5 billion in 2030, which will significantly elevate urbanization challenges [1]. With the increase in traffic volume, a lack of adequate traffic management would result in significant economic and environmental losses, e.g., energy consumption, greenhouse gas emissions, and time delays. To cope with these challenges, smart traffic management are needed , such as smart parking integration [2], advanced safety and pollution analytics [3], electronic road pricing and toll collection [4], and video traffic detection systems. Although the commonly used systems are based on Closed-Circuit Television (CCTV), traffic management systems are still suffering from limited control over the traffic network via standard viewpoints.

To overcome the limitations of gathering traffic data via CCTV, video collection utilizing Unmanned Aerial Vehicles (UAVs) has lately been proposed in [5] due to their ability to monitor a huge spectrum of roadways. In fact, UAVs can monitor a broad range of roadways by shifting altitude and location. Moreover, UAVs can operate on-demand by traveling to a specific area to observe unpredictable situations such as road traffic accidents. On the other hand, due to its rapid, precise, and accurate data collection, Light Detection and Ranging (LiDAR) technology has shown great potential in a variety of applications such as terrestrial ecology, agriculture [6], astronomy [7], hydrology, and atmospheric science. However, LiDAR is becoming increasingly popular in transportation and navigation. For instance, it has demonstrated exceptional performance in monitoring traffic congestion and guiding vehicles to park safely. LiDAR is also used for autonomous vehicle navigation, collision avoidance [4], and autonomous cruise control [8]. LiDAR's high spatial resolution and mapping accuracy make it an ideal solution for planning transport and road networks [9]. Furthermore, it provides the high-grade reliability needed while preserving anonymity among road users. Indeed, the point cloud data generated from LiDARs is a convenient container for storing, processing, and visualizing 3D raw scanner measurements.

There are a collection of multidimensional points that represent physical surfaces. In particular, 3D object detection has emerged as an active research topic in the field of computer vision. However, due to the sparse and complex nature of 3D point cloud data, this type of data remains a challenging task. In [10], the authors proposed a solution to capture traffic video using UAVs integrated with onboard cameras, where the data is processed in the cloud. Their aerial prototype can collect and transmit real-time videos that include a vehicle detection stage based on the Haar cascade model [11] and a frame-by-frame tracking stage. Although this method exhibits excellent performance, it does not overcome the lack of information in 2D images by using conventional cameras. In [12], the authors introduced a robust multiple object detection and tracking (MODT) algorithm for a non-stationary base, using multiple 3D LiDARs for perception. The solution can be applied in real-time on a vehicle-embedded computer. However, such an approach uses multiple 3D ground LiDARs, which can significantly increase the cost and the complexity of its deployment in further applications.

In this paper, we propose the employment of a traffic monitoring system using LiDAR-equipped UAVs. The goal is to create a high-performance 3D object detection and tracking solution geared toward traffic monitoring. The framework uses raw 3D LiDAR data as input to perform multi-target object detection while keeping track of the identified objects in a robust and real-time manner. Firstly, we investigate the main challenges of using LiDAR-equipped UAVs to capture real-world traffic. However, due to the unavailability of real-world LiDAR data from the sky, we simulate various traffic scenarios using a 3D simulator to generate our own dataset. Afterward, for car and pedestrian detection, we use PointVoxel-

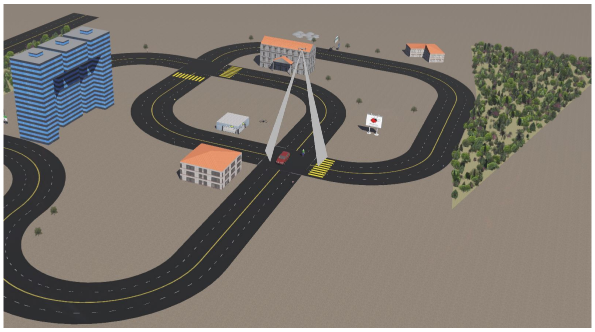

Fig. 1: A screenshot illustrating the developed environment on Webot simulator.

RCNN(PV-RCNN) and Point-RCNN as 3D detection models. Our proposed tracking algorithm employs a Kalman filter to predict the trajectory of the detected object, estimate the motion model, and tackle the data-matching problem. The evaluation results show that the proposed solution achieves promising results in both the detection and tracking stages.

II. Data Acquisition

Deep learning algorithms depend heavily on the amount of training data on the models. This data can be available as open-source data or collected in real life. However, the existing dataset may not meet the requirements of the investigated problem. Furthermore, collecting the data may be a time-consuming and labor-intensive task. For this work, we propose to use Webots simulator for data collection. Webots is an open-source 3D robot simulator that offers highly detailed simulations with realistic capabilities [13]. It allows the creation of different traffic scenarios and scenes containing moving ground objects, as shown in Fig. 1. We generate point cloud data using a LiDAR laser sensor attached to a static UAV operating at a flight altitude of 50 m above the ground level.

The diversity and size of the dataset used to train the deep learning algorithm can significantly affect the learning. In this work, we simulated different scenarios while mixing between pedestrians and vehicles, as well as single-target and multi-target situations. Further, we consider different shapes and sizes of the objects to maximize the diversity of the dataset. A sample illustration of a point cloud capture generated from Webots are given at the top of Fig. 2.

III. Proposed Architecture

The main goal of this work is to implement a 3D detection and tracking solution that is able to automatically detect and track pedestrians and vehicle objects given 3D raw point cloud capture as input.

The proposed system consists of three main modules, as illustrated in Fig. 2:

1. Labeling stage: it consists of annotating the raw point cloud with the semi-automatic labeling tool labelCloud.

2. Detection stage: we pass the annotated 3D data to PV-RCNN model.

3. Tracking stage: the detected object is given to the Kalman filter to predict its kinematic state, like position and velocity.

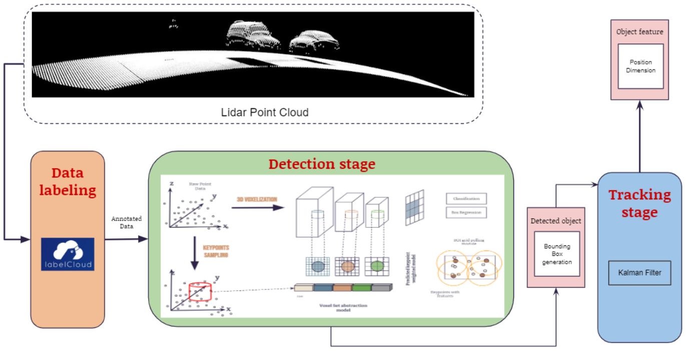

Fig. 2: Overview of the proposed architecture for detecting and tracking road users.

a) Data Labeling: The LiDAR sensor specifications used in this work provide 180-degree spatial information and yield a minimum of $20,000$ 3D points per measurement. This representation is our raw input to the object detection and tracking model. However, before proceeding with these steps, we first annotate every collected point cloud by drawing a cuboid, which is a 3D bounding box located around every object in the point cloud scene. We use labelCloud which is an annotating tool built for versatile use and aims at supporting all common point cloud file formats and label formats for storing 3D bounding boxes. Using labelCloud, the point cloud captures are labeled by cuboid annotation. Each object is assigned to a class and its dimensions (height, width, and length) together with the coordinates of its centroid. Afterward, we implement an automatic algorithm that transforms the labels into TXT files as follows:

$$\Big\langle classname \Big\rangle \Big\langle w \Big\rangle \Big\langle l \Big\rangle \Big\langle h \Big\rangle \Big\langle R \Big\rangle \Big\langle x \Big\rangle \Big\langle y \Big\rangle \Big\langle z \Big\rangle$$

where Class name is the class to which the object belongs, mainly cars or pedestrians, w, l, and h are the height, width, and length of the object, respectively, x, y, and z are the 3D object location in LiDAR coordinates, respectively, and R is the rotation of the cuboid around the Z-axis in LiDAR coordinates.

b) PV-RCNN for Object Detection: PV-RCNN is an efficient and accurate 3D deep learning model designed to improve 3D object detection from raw point cloud. Contrary to the previous detection models which are either Voxel-based or Point-based, PV-RCNN combines the pros of Point-based and Voxel-based models [14].

The PV-RCNN model main steps employed with the proposed LiDAR-equipped UAV are given as:

- **3D sparse convolution backbone:** a sequence of sparse convolutions is used to downsample the point cloud by 8 ×. The layers' intermediate outputs are then saved for use in the Voxel Set Abstraction layer.
- **Bird's eye view converter:** the features are stacked on top of each other to form a 3D array of H×W×Fetaures.
- **Sectorized keypoint sampling:** only points under a certain radius of these proposals are deemed significant, and the remaining are discarded. Following that, the point cloud is split into sectors, and the pertinent keypoints are acquired using farthest point sampling (FPS) in each sector.

- **Voxel set abstraction and keypoint weighting:** after extracting the keypoints from preceding layers, we retrieve the known features from the point cloud using vector pool aggregation. With the features of each keypoint, a multi-layer perceptron is used to evaluate and multiply the relative importance associated with each keypoint.
- **Vectorpool aggregation:** the point cloud is subdivided into N neighborhoods, each with its center point. After that, each neighborhood is split into a number of high-density voxels.
- **ROI-GRID pooling:** the 3D proposal's keypoint features are grouped to RoI-grid points with several receptive fields.
- **Box refinement and confidence prediction:** finally, two sibling sub-networks are used for confidence prediction and proposal refinement, and then, the refinement branch predicts the size and location residuals in relation to the previous 3D proposal box.

passed to the tracker.

A. *Unscented Kalman Filter for Object Tracking*

Kalman Filter (KF) was introduced by R. E. Kalman in 1960, to deal with the issue of retrieving the output signals from noisy measurement variables [15]. The measurement variables are used as input signals based on the statistical properties of the system noise and measurement noise, and the unknown variables are the filter's output. However, the Kalman filter can only be applied if the state and measurement models are both linear.

However, this condition is extremely hard to meet in real system applications, including our problem. The suggested system will differ significantly from the current system in some ways. Furthermore, the noise environment in real-world applications is more complicated. Many improved filtering techniques have been proposed to overcome these limitations to extend the implementation of KF like the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). UKF was introduced [16] to realize tracking by employing a non-linear model. UKF opts to use an approximation based on so-called sigma points from a Gaussian distribution. Pedestrians and vehicles in congested areas are expected to use multiple motion models rather than a linear constant velocity, as they would in a less congested area. Second, tracking multiple objects in an environment at the same time is a difficult task. Therefore, we propose a tracking solution based on UKF to address these two issues.

In a nutshell, assuming that the prior probability distribution of the state variable X is follow a Gaussian distribution (ν,σ^2) UKF uses a non-linear transformation of the system along with posterior probability distribution to predict the mean value and the covariance matrix. The movement of cars and pedestrians is considered a combination of movements in the x-axis, y-axis, and z-axis. We need to use an appropriate equation to represent the dynamic behavior of each movement. The state vector is represented with a 13-dimensional vector T=(x,y,z,θ,l,w,h,v_x,v_y,v_z,a_x,a_y,a_z) describing the current state of an object's trajectory where v_x, v_y, and v_z represent the velocity while a_x, a_y, and a_z represent the acceleration in 3D space. A nonlinear system is thus used to define the moving model for the i-th object, as follows:

$$x_i(n) = x_i(n-1) + v_{xi}(n-1)dt + \frac{1}{2}a_{xi}(n-1)dt^2,$$

$$y_i(n) = y_i(n-1) + v_{yi}(n-1)dt + \frac{1}{2}a_{yi}(n-1)dt^2.$$

The process of tracking objects with UKF, can be resumed in three main steps:

Step 1: For each detected object, we take the bounding box with dimension, center position, and yaw. We define the state vector T, the transition matrix F, and the measurement matrix H.

Step 2: Use the UKF to predict each object's position and velocity.

Step 3: Update the state vector with measurement values of each object's location.

IV. Results & Discussions

This section presents selected results to show the performance of our proposed method.

a) Data Augmentation (DA): it is essential to improve learning performance and avoid overfitting. Therefore, we employ three 3D DA techniques, including: (i) *Random global scaling:* every point cloud is multiplied by a scale factor between [0.97,1.3] to obtain a new point. (ii) *Global translation:* we translate all the points in the point cloud along X, Y, and Z values by offset values. The offset values are three values drawn at random from a normal distribution, with the zero mean and different variance values from a set of 0.2, 0.3, 0,4, and 0.5. (iii) *Global rotation around Z axis:* we rotate every point in the point cloud along the Z axis as defined in the following:

$$P' = R \times P$$

where P is a point cloud defined with (x, y, z), $\times$ is the matrix multiplication operator and R is the rotation matrix characterized by rotation angle β randomly picked from $-\pi$/4 to π/4.

b) Training Phase: Point-RCNN is an object detection framework 3D detector that offers accurate and precise 3D detection performance by working directly on 3D point clouds [17]. It uses Pointnet++ as a backbone for point-wise feature learning [18]. It is composed of two main sub-stages: bottom-up 3D proposal generation stage and canonical 3D Box refinement. In our work, we used Point-RCNN as a baseline model to compare it to our proposed solution. The dataset contains over 4000 frames. Each frame represents a minimum of one car and one pedestrian. Our model is trained with 80% of the data, and the remaining 20% of the data is used for validation. The PointRCNN's two-stage sub-networks are trained independently. In contrast to the stage-2 sub-network, which is trained for 30 epochs with batch size 256, the stage-1 sub-network is trained for 100 epochs with batch size 16. The learning rate is 0.002 for both stages. For PV-RCNN model, the training details were set as: the feature dimensions of the four-level 3D voxel CNN are 16, 32, 64, 64, respectively. The two neighboring radii of each grid point in the RoI-grid

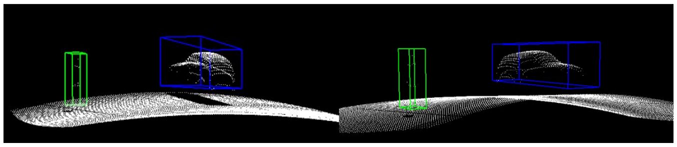

Fig. 3: Snapshots of the 3D point cloud detection.

pooling operation are (7m, 7m). The voxel size is (0.03 m, 0.03 m, 0.05 m) in each axis, the batch size is 48, the optimizer is ADAM optimizer and the learning rate is 0.01 for 80 epochs. Training each model takes 2-4 hours to converge with Pytorch and 2 NVIDIA RTX A6000 GPUs.

c) Quantitative Results: the classification, position, and dimension data of the cuboid box are obtained from the detection results. The detecting performance of the predictive model is evaluated by the overlap volumes between the cuboid boxes and ground truth boxes based on the evaluation dataset. As mentioned in Table 1, the detection results are evaluated by using the Average Precision metric with a threshold 0.7 for cars and 0.55 for pedestrians. The mean average precision (mAP) is calculated using 11 recall positions.

TABLE I: Object detection performance and DA impact.

	AP (Cars)	AP (Pedestrians)	mAP
Point-RCNN w/o DA	82.56	79.37	80.96
Point-RCNN with DA	86.41	82.98	84.67
PV-RCNN w/o DA	89.32	86.19	87.75
PV-RCNN with DA	92.51	90.33	91.42

We compare the networks that were trained on the samples after augmentation with those that were trained on the raw samples to validate the impact of DA. After applying our DA algorithm, the mAP of the Point-RCNN has been improved by about 4.6%, and the mAP of PV-RCNN has been increased by about 4.18%. This implies that the augmented modes have a greater chance of making an accurate prediction. This is due to the model now having a wider variety of object representations and positions from which to learn features. The achieved Average Precision by Point-RCNN and PV-RCNN in testing were 77.54 and 91.42 respectively. In Fig. 3, we provide an example of the detected road users using LiDAR from an aerial view.

d) Tracking Results: to inspect the tracking performance, we chose 2 samples from the dataset, which were selected to compare our UKF tracker with EKF tracker.

In Fig. 4, we can notice that EKF and UKF trackers followed the real values perfectly when the pedestrian was on a linear trajectory. When the pedestrian shifted to a non-linear trajectory, the EKF had a bias and continued to follow a linear line, while our proposed UKF tracker had a more accurate estimation.

In Fig. 5, we simulate a car with a 10 mph speed. After $x = 75$ m, we kept fluctuating the speed by -5mph and +5 mph. As can be seen, the EKF values had some bias since the velocity is varying continuously. However, the UKF was not heavily affected by velocity change. This shows the capability of the UKF tracker to follow different target trajectories moving with non-constant speed.

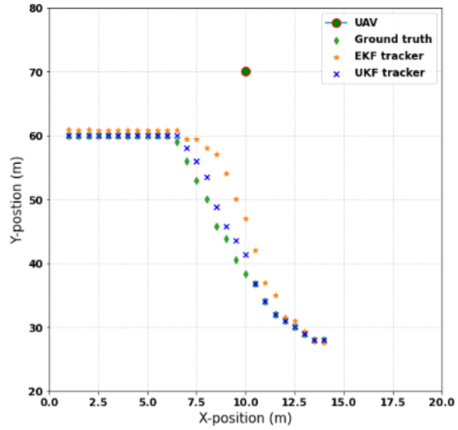

Fig. 4: Tracking spatial evolution of a pedestrian with a fixed speed.

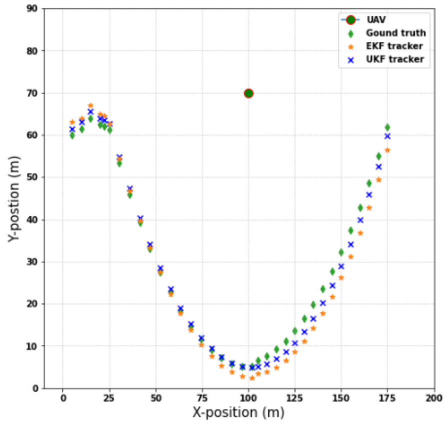

Fig. 5: Tracking spatial evolution of a car with fluctuated speed.

In the case of having multiple objects, we identify every detected object by giving it a specific index. Subsequently, we assign detections to tracks using the James Munkres's variant of the Hungarian assignment algorithm. We then decide which tracks are missing and which detections should start with new ones. We compute the indices of assigned and unassigned tracks, as well as the indices of unassigned detection. To do this, we calculate the cost matrix, which is a t by d matrix. T equals the number of tracks in this matrix, and d represents the number of detections.

V. Conclusion

In this paper, we proposed an aerial LiDAR-based solution to detect and track moving objects, primarily pedestrians and cars. A comparison between two detection models operating on the 3D LiDAR point cloud, namely Point-RCNN, and PV-RCNN, was performed and has shown the superiority of the PV-RCNN method. The object detection stage is then followed by a tracking stage based on the Unscented Kalman filter, which estimates the target's state vector and enables effective tracking of the detected mobile road users.

References

[1] D. o. E. United Nations and P. D. Social Affairs, "Population 2030: Demographic challenges and opportunities for sustainable development planning," *(ST/ESA/SER.A/389)*, 2015.

[2] G. Revathi and V. R. S. Dhulipala, "Smart parking systems and sensors: A survey," *International Conference on Computing, Communication and Applications, Dindigul, India*, pp. 1–5, Apr. 2012.

[3] S. Li and F. Robusté, "From urban congestion pricing to tradable mobility credits: A review," *Transportation Research Procedia*, vol. 58, pp. 670–677, July 2021.

[4] S. Ramasamy, R. Sabatini, A. Gardi, and J. Liu, "LiDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid," *Aerospace Science and Technology*, vol. 55, pp. 344–358, Aug. 2016.

[5] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. T. Matson, "Real-time and accurate drone detection in a video with a static background," *Sensors*, vol. 20, no. 14, pp. 1–19, July 2020.

[6] N. Borowiec and U. Marmol, "Using LiDAR system as a data source for agricultural land boundaries," *Remote Sensing*, vol. 14, pp. 1–17, Feb. 2022.

[7] A. Comerón, C. Muñoz-Porcar, F. Rocadenbosch, A. Rodríguez-Gómez, and M. Sicard, "Current research in LiDAR technology used for the remote sensing of atmospheric aerosols," *Sensors*, vol. 17, pp. 1–16, July 2017.

[8] M. A. Bharmal and M. H. Rashid, "Designing an autonomous cruise control system using an A3 LiDAR," in *proc. of the 2nd International Conference on Innovation in Engineering and Technology (ICIET), Dhaka, Bangladesh*, pp. 1–6, Dec. 2019.

[9] J. Zhao and S. You, "Road network extraction from airborne LiDAR data using scene context," in *proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA*, pp. 9–16, June 2012.

[10] H. Niu, N. Gonzalez-Prelcic, and R. W. Heath, "A UAV-based traffic monitoring system," in *proc. of the 87th IEEE Vehicular Technology Conference (VTC Spring), Porto, Portugal*, pp. 1–5, June 2018.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA*, Dec. 2001.

[12] M. Sualeh and G.-W. Kim, "Dynamic multi-LiDAR based multiple object detection and tracking," *Sensors*, vol. 19, no. 6, pp. 1–20, Mar. 2019.

[13] O. Michel, "Cyberbotics ltd. webots™: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 1–5, Mar. 2004.

[14] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: point-voxel feature set abstraction for 3D object detection," *Arxiv*, Sept. 2021.

[15] R. E. Kalman, "A new approach to linear filtering and prediction problems," *The American Society of Mechanical Engineers - Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, Mar. 1960.

[16] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *SPIE*, vol. 3068, pp. 182–193, July 1997.

[17] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3D object proposal generation and detection from point cloud," *Arxiv*, May. 2019.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *proc. of the 31st Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA.*, Oct. 2017.