Fall 2023 - Digit - Computer Vision Report

Abirath Raju
LiDAR Lab
Georgia Institute of Technology
Atlanta, Georgia

***Abstract*—It is crucial for a bipedal robot to be able to perceive humans in a social environment and learn to navigate around them to reach a goal. This semester, the Digit Computer Vision team concentrated on various avenues to equip Digit with the necessary perception capabilities to track the x,y coordinates of 8 pedestrians in the vicinity and pass these coordinates to the MPC controller which will help the robot navigate around humans and avoid collisions. Different computer vision frameworks such as VoxelNet, SFA3D, Complex YOLO v4[1] and 2D Monocular Depth estimation were tried this semester and progress has been made, which can be built upon in the future semesters in addition to trying out different avenues. A MoCap system is also being developed for this purpose to validate the working of the MPC with the idea to circumvent the need of a perception system over the Winter of 2023.**

Keywords—Computer Vision, Velodyne LIDAR, Deep Learning, VoxelNet, SFA3D.

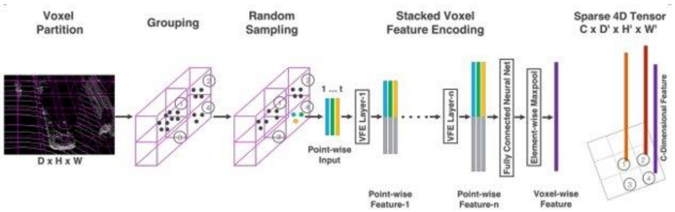
I. Introduction

Robotics applications are expanding at an increasing rate, and current developments in this area suggest that this tendency will only continue. To accomplish their tasks, legged robots must increasingly navigate a variety of environments. We are focusing on enabling Digit, the bipedal robot developed by Agility Robotics, to perceive other humans in the environment and navigate like a human through a crowd. Digit has a variety of sensors on board, which includes a RGB camera, 4 RGBd cameras and a 360 degree Velodyne LIDAR mounted on top. For this project, we are exploring techniques that use the LiDAR scan information from the Velodyne sensor present on Digit. We have 4 RGBd cameras present on the robot, but none of them have a view of what's in front of Digit and thus making them unusable for our purpose of tracking human beings in the vicinity. Hence, we explored using the LIDAR scan information to perform detection of humans using deep learning algorithms and tried outputting the coordinates of detected humans in the vicinity. We are also working on building a Motion capture system using the Vicon system and attaching markers on humans to try and track during the Winter of 2023 which will help in validating the functioning of the developed MPC controller for Digit. The go-to dataset for human lidar scan data is the KITTI dataset, created by Karlsruhe Institute of Technology and Toyota Technological Institute. We used this data throughout the project to train our computer vision models.

II. Approaches Pursued

A. *VoxelNet*

We developed a working Voxel Feature Encoding network[2] from scratch which was capable of outputting bounding boxes on lidar data with an average precision of 14%. This is lower than expected, but is considerably okay given that the state of the art implementation is only able to detect humans using only the lidar data with an accuracy of ~50%. The working of the VoxelNet network is as follows:



The figure above illustrates the various layers comprised in the network. This approach employs n-pointwise fully connected VFE layers to create a feature space, where we can aggregate information from the point features and encode the shape of the surface contained within the voxel (non-empty voxels aren't considered). As the output feature combines both point-wise features and locally aggregated features, stacking VFE layers encodes point interactions within a voxel and enables the final feature representation to learn descriptive shape information. Post this process, convolutional layers are applied to the Voxel features to output a more comprehensive feature map, which is fed to a Region Proposal Network which further down samples and up samples the encoded features to create a probability map and a regression map.

Implementation details:

The positive and negative anchors are parameterized as vectors containing x,y,z,l,w,h and ϴ (where ϴ is the yaw orientation). The ground truths are also parameterized similarly and we take the residuals of each variable like so:

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

$$\Delta x = \frac{x_c^g - x_c^a}{d^a}, \Delta y = \frac{y_c^g - y_c^a}{d^a}, \Delta z = \frac{z_c^g - z_c^a}{h^a},$$

$$\Delta l = \log(\frac{l^g}{l^a}), \Delta w = \log(\frac{w^g}{w^a}), \Delta h = \log(\frac{h^g}{h^a}),$$

$$\Delta\theta = \theta^g - \theta^a \qquad d^a = \sqrt{(l^a)^2 + (w^a)^2}$$

We implemented the loss function using the following equation:

$$L = \alpha \frac{1}{N_{\text{pos}}} \sum_i L_{\text{cls}}(p_i^{\text{pos}}, 1) + \beta \frac{1}{N_{\text{neg}}} \sum_j L_{\text{cls}}(p_j^{\text{neg}}, 0) + \frac{1}{N_{\text{pos}}} \sum_i L_{\text{reg}}(\mathbf{u}_i, \mathbf{u}_i^*)$$

Where p_i^{pos} and p_j^{neg} represent the softmax output for positive anchor a_i^{pos} and negative anchor a_j^{neg} respectively, while $\mathbf{u}_i \in \mathbb{R}^7$ and $\mathbf{u}_i^* \in \mathbb{R}^7$ are the regression outputs and the ground truths for the positive anchor a_i^{pos}. The first two terms are the normalized classification losses, where the Lcls stands for binary cross entropy loss and α, β are positive constant hyper parameters balancing the relative importance. Lreg is the regression loss.

We also implemented a custom average precision metric to evaluate the model based on the confusion matrix (TP, TN, FP, FN) and IoU (Intersection over union).

B. SFA3D

SFA3D (Super Fast and Accurate 3D object detection)[3] is an object detection algorithm that we tried implementing in parallel with VoxelNet. This algorithm takes in the raw lidar feed without any preprocessing as input , creates a Bird's eye view of the environment using the lidar scans and outputs bounding boxes on the BEV map. The algorithm produces seven outputs which are heatmaps of the different classes present in the scene, the distance of the identified classes from the center of the ego agent, the angles in which the predicted objects are facing, the dimensions of the detected objects and the z-coordinates of the identified objects. The main constituents of SFA3D which make it unique are the keypoint feature pyramid network, the multiple loss functions it uses and the learning rate scheduling technique employed.

Given an input image, the FPN creates multiple feature maps of different sizes. By post-processing these output feature maps we can perform object detection. Skip connections are employed to make sure that relevant information is not lost in the process of downsampling and upsampling the input. Because of this reason, FPN's are scale-invariant and are able to detect objects which are of different sizes.

Loss functions employed:

Focal loss:

$$-(1 - Pt)^r . logPt$$

This loss encourages the network to learn the correct classes. This helps the network especially when there is a class imbalance in the dataset. Pt is the confidence value of the network. The scaling factor in the expression helps guide the network to give more attention to the classes in which it is less confident about and vice-versa.

L1 loss:

$$|yt1 - yt|$$

It takes in the ground truth value and the predicted value and computes the difference.

Balanced L1 loss:

This loss function is employed to deal with discrepancies in the weights of the neural network in the case of outliers and inliers.

The BEV map which is fed as input is created by using the U-net architecture.

C. 2-D Monocular depth estimation[4]

Taking a step back, we tried to get the coordinates of the humans in the vicinity just by using the 2D image feed captured using the front-facing RGB camera on Digit. We used the YOLO detection algorithm to get the bounding boxes on humans in the frame and then used Mediapipe which is a tool for pose estimation to get the distance of the human from the agent, which in this case is Digit. The algorithm recognizes shoulders of a human in the image (feature) and then calculates the distance if any are detected. This approach produces an error of about 2 inches per prediction which can be a considerable amount depending upon the application.

III. DATASET PREPARATION

The LIDAR scan information coming from Digit is accessed currently using a ROS topic, and it outputs matrices which contain X, Y, Z, R, T, I information. We have to record this information using the Agility perception code and then save the data in the form of binary files which can be used by the deep learning models to output predictions.

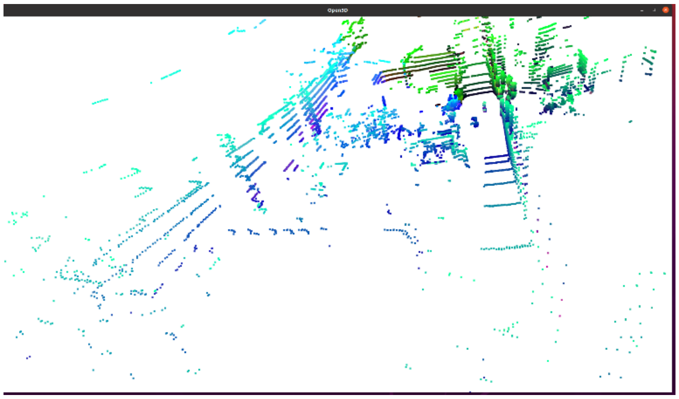
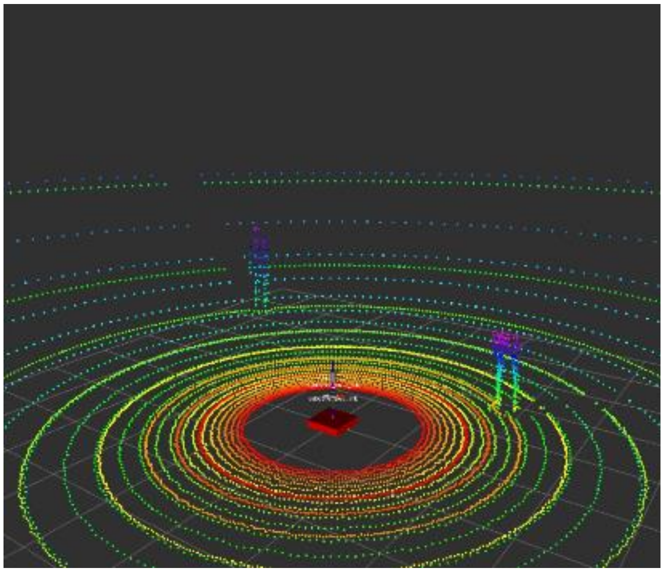
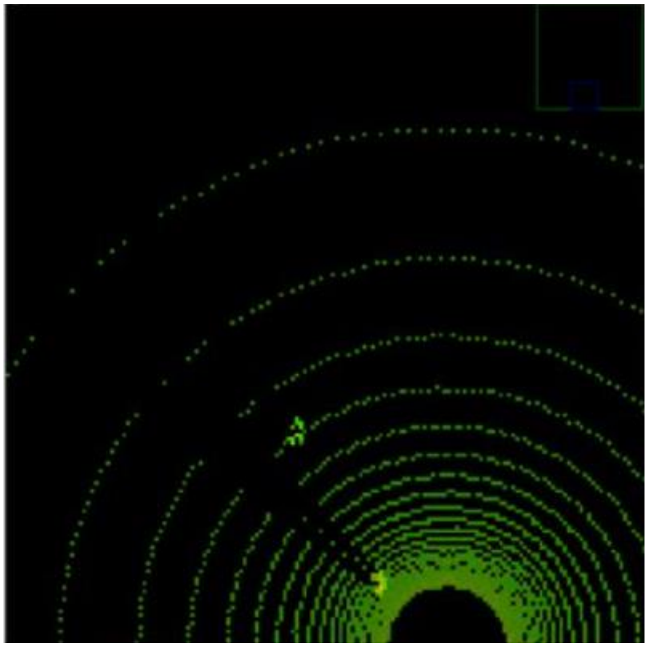


Figure depicting LIDAR scan information of Velodyne sensor on Digit visualized using Open 3D[5].

As seen from the lidar scan image above, the lidar scan data appears to be sparse and doesn't contain enough information for the model to identify humans. To figure out the source of this issue, we also tried simulating an environment in Gazebo with a Velodyne Lidar and seeing if the model is able to predict humans from the simulated data. But the model wasn't able to output proper bounding boxes in both the cases.



Gazebo simulation environment of 2 people

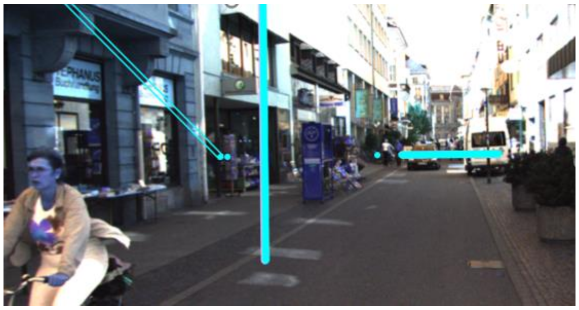


BEV map of the simulated environment generated by SFA3D

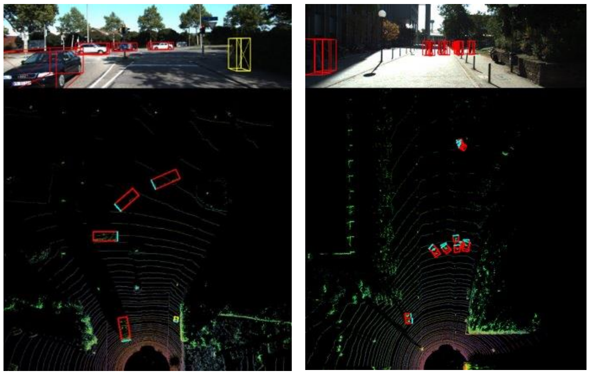
This led us to believe that there is something wrong with the way we are passing LIDAR data to the models and we set to explore any discrepancies between the KITTI dataset that we used for training and the .bin files collected from Digit used for testing. We made some progress in identifying some header differences in the binary files, but we believe there is more to explore. This is one avenue that we wish to explore further in the upcoming semester.

IV. RESULTS

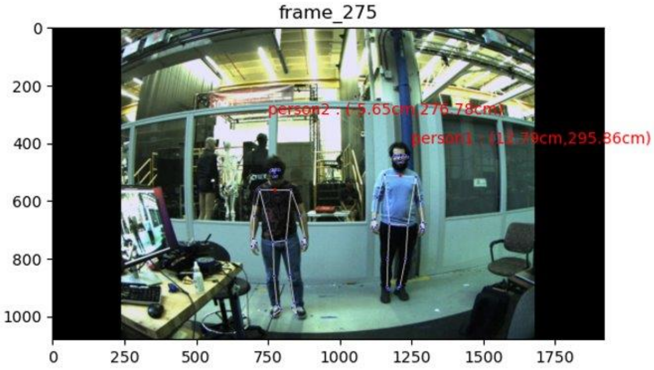
We have training results of both VoxelNet and SFA3D, which are two of the most promising models that we implemented suited for this task and below are the results we obtained.



Prediction outputted using VoxelNet (~14% precision score)



Output using SFA3D which outputs bounding boxes accurately during inferencing.



Monocular 2D results for depth estimation

As seen, SFA3D predicts bounding boxes on unseen KITTI dataset better than VoxelNet, and it has to be tuned/modified appropriately in order to accommodate for the differences in the .bin files generated by Digit. Monocular 2D depth estimation provided satisfactory results but has to be refined further to produce dependable and accurate results.

V. AVENUES TO EXPLORE

For the upcoming semester, we will be focusing on getting x,y coordinates of humans in the environment using a motion capture system by attaching markers on humans in the environment and validating the performance of the MPC controller.

We will be working on identifying the discrepancies between the KITTI dataset and the .bin files generated by Digit and trying to make them compatible with the deep learning models developed.

We are also looking to explore sensor fusion methods where we aim to fuse the RGB sensor with the LIDAR data (to get a sense of depth) to output bounding boxes and get the coordinates of the pedestrians.

We will also be looking to refine and fine tune the VoxelNet and the SFA3D models developed during the Fall'2023 semester to improve accuracy of predicted labels.

We are also striving to implement our algorithm online on Digit to detect humans in real-time and navigate around them.

VI. CONCLUSION

This is a pretty hard problem to solve and we have made some progress by implementing models which achieved good results by outputting bounding boxes on unseen data. Work has to be done to translate the results on data obtained from the LIDAR on Digit, which looks promising given the progress we have made this semester.

REFERENCES

[1] Martin Simon et. al: Complex-YOLO: Real-time 3D Object Detection on Point Clouds, 2018.

[2] Yin Zhou et. al: VoxelNet:End-to-end Learning for Point Cloud Based 3-D Object Detection.

[3] Nguyen Mau Dung: Super-Fast-Accurate-3D-Object-Detection-PyTorch, 2020.

[4] Xingshuai Dong et. al: Towards Real-Time Monocular Depth Estimation for Robotics: A Survey.

[5] Qian Yi-Zhou et. al: Open3D, A modern library for 3D Data Processing, arXiv:1801.09847, 2018.