BOSTON UNIVERSITY

Spatially Encoded LiDAR Based Residual Learning for Obstacle Detection

Elizabeth Chung,[1,2] Abdelrahman Abdelgawad,[2] Roberto Tron[2]
Nashua High School South, 36 Riverside St, Nashua, NH 03062,[1] Boston University Robotics Laboratory, 750 Commonwealth Ave B29, Brookline MA 024462[2]

Introduction

Background:

Robots require reliable obstacle detection in order to safely navigate environments.

- Sensors: Traditional approaches to autonomous obstacle detection often employ LiDAR sensors[1] or vision-based systems[2]. LiDAR (light detection and ranging) utilizes laser light for depth perception of surrounding objects.
- Processing: Deep learning uses multi-layered artificial neural networks to extract meaningful features from input data to make predictions. This is essential to optimize the process of robotic obstacle detection, allowing robots to process and understand their surroundings. However, applications to high-level input data can be expensive and computationally heavy.
- Representation: Zili Wang et al. demonstrate how LiDAR data can be abstracted into higher-level representations in BoxMap[3].

Objectives:

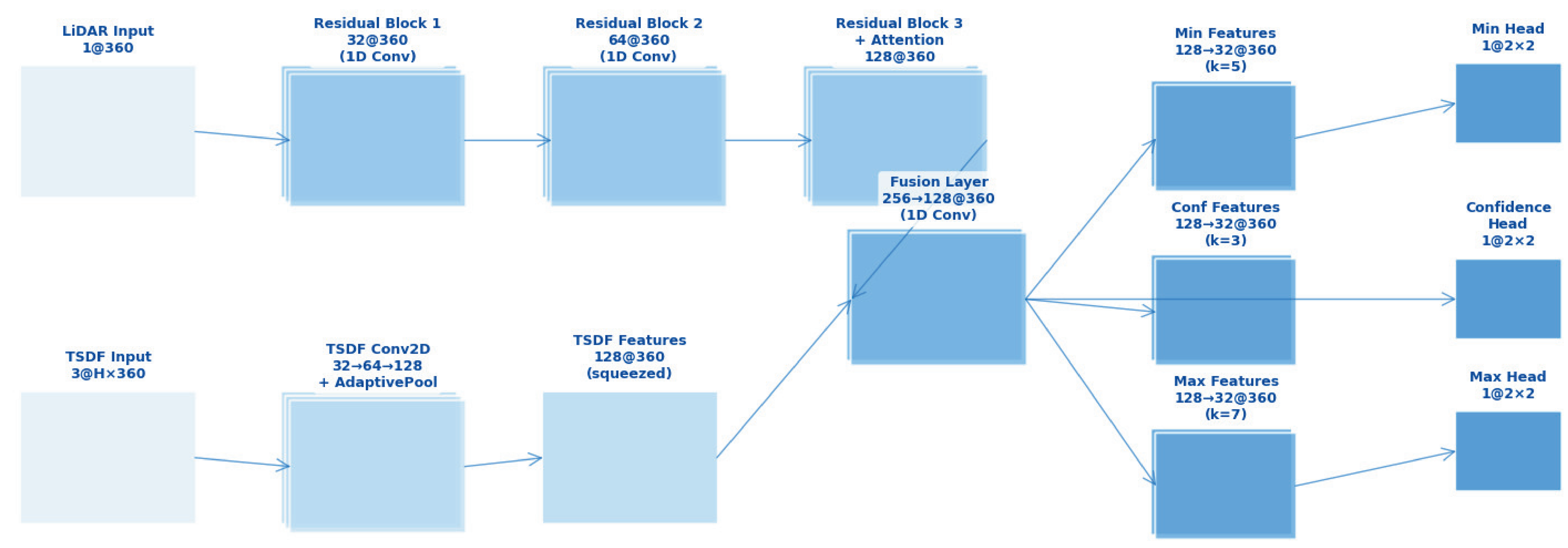
We investigate the problem of real-time obstacle detection in resource-constrained scenarios via sparse 1D LiDAR input.

- We aim to use LiDAR and spatial projections to detect obstacles within new environments via deep learning.
- As proof of concept, we utilize obstacle detection in a simple obstacle avoidance algorithm.

Methods

Neural Network Architecture:

- We propose a custom Residual Network (ResNet):



- The dual-branch architecture captures 2D spatial features and 1D LiDAR patterns, allowing the network to learn complementary representations.
- Our loss incorporates position loss via cross entropy, confidence loss via binary cross entropy, and diversity loss via intersection over union (IoU).

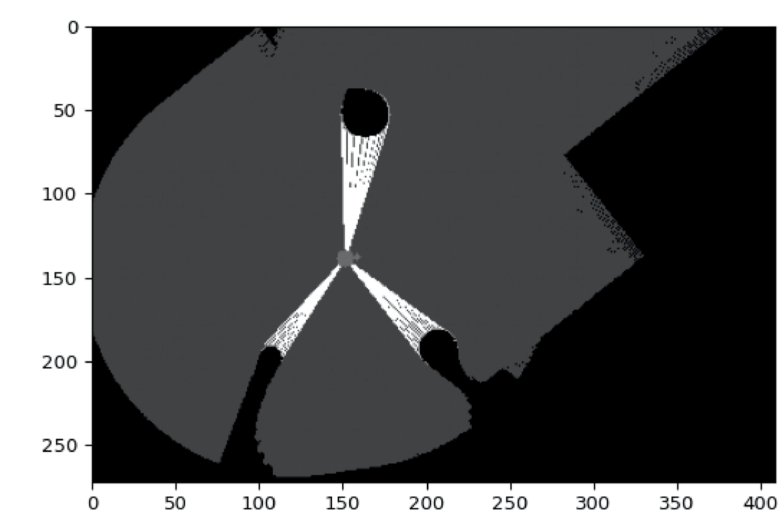


Fig. 1: Sample simulated LiDAR in a 3-obstacle environment

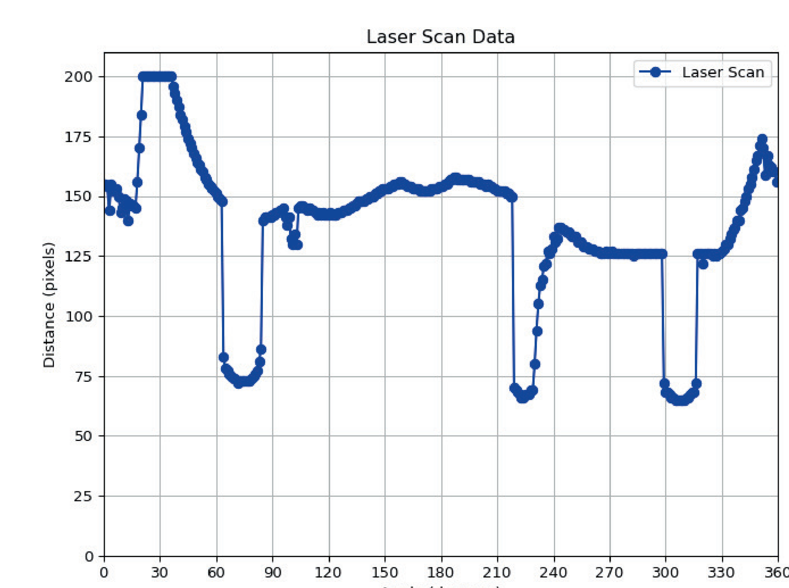


Fig. 2: Graph of LiDAR

Data Generation:

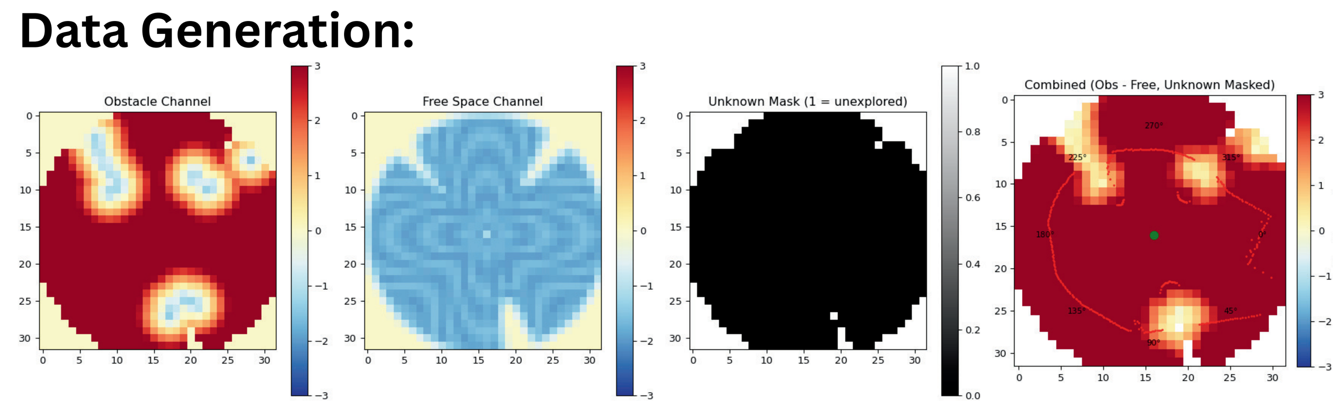


Fig. 3: Tri-channel TSDF to display spatial obstacle features extracted from LiDAR

- Our simulation[4] uses keyboard exploration of maps with obstacles. We simulate LiDAR by projecting rays from the robot over a span of 360°. We generate a binary label array indicating obstacle presence using trigonometry (Fig. 1, 2).
- We implement a Truncated Signed Distance Function (TSDF) to create a spatial representation of the LiDAR (Fig. 3).

Pathing: We simulate exploration by generating random actions for the robot (left, right, or forward). As the robot travels, we filter out actions that would result in collisions by checking if the robot's heading would fall in the direction of an obstacle based on the ResNet prediction and verifying the distance to the obstacle with the LiDAR scan.

Results

Preliminary Specialized Models:

- We trained three models exclusively on a single map to detect 1, 2, or 3 obstacles (Fig. 4-6).
- The neural networks successfully learned the positions of obstacles on the map (Fig. 7-9) and demonstrated high accuracy.

Accuracy:

# of Obstacles	Accuracy (%)
1	95.54
2	91.62
3	88.44

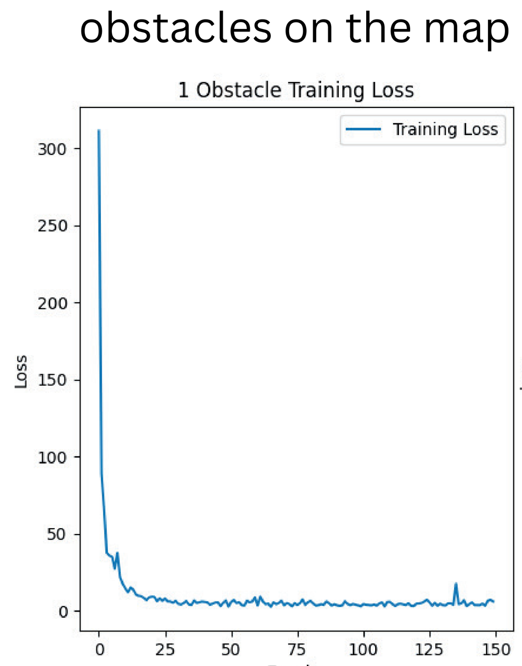


Fig. 4: Loss of 1-obstacle network

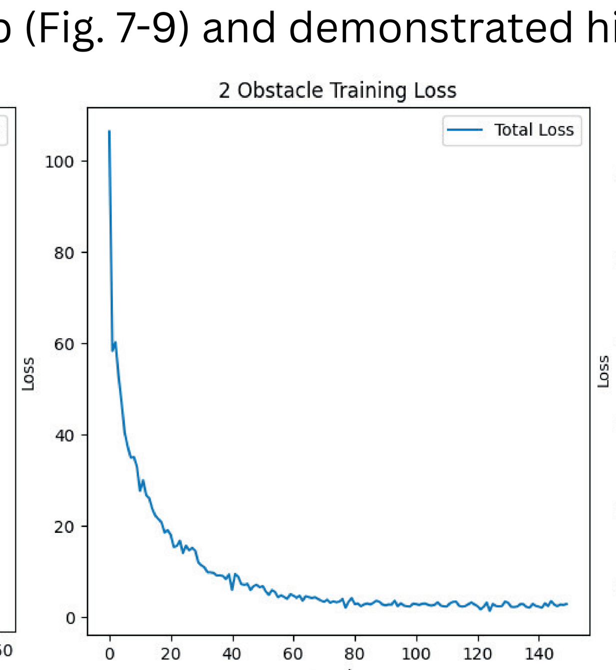


Fig. 5: Loss of 2-obstacle network

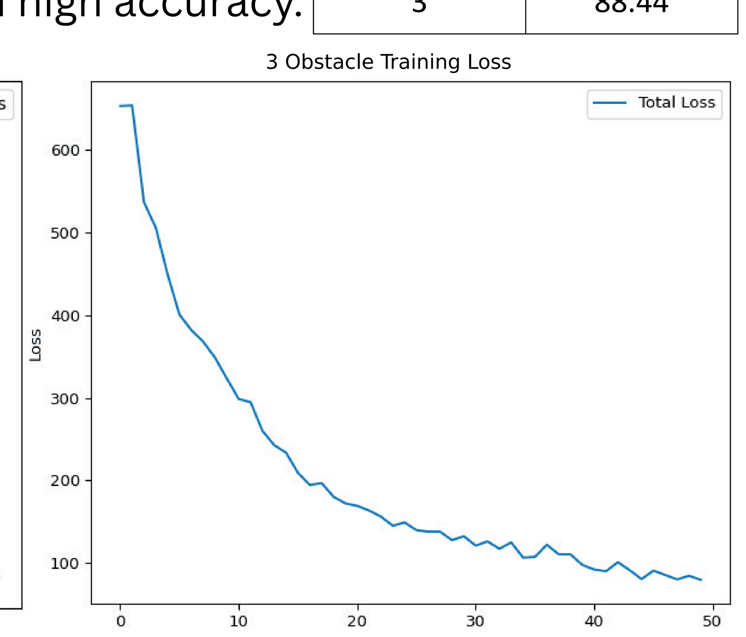


Fig. 6: Loss of 3-obstacle network

Fig. 7: 1-ob prediction

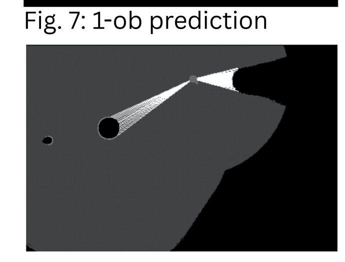


Fig. 8: 2-obs predictions

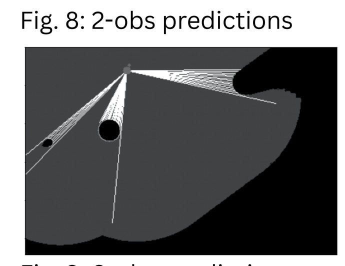


Fig. 9: 3-obs predictions

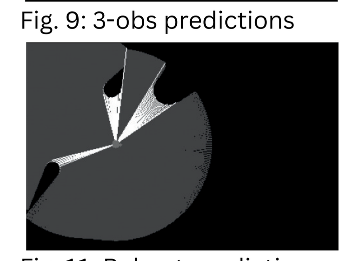


Fig. 11: Robust predictions

Robust Multi-Map Model:

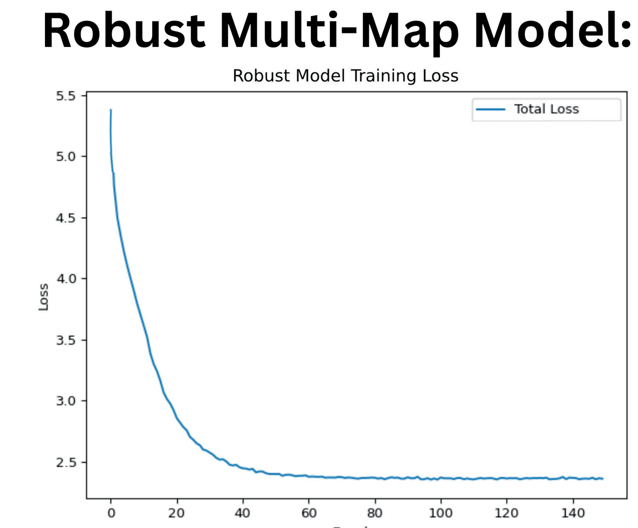


Fig. 10: Loss of the robust network

- We trained a fourth model on 50 different maps to learn from a variety of obstacle placements (Fig. 10).
- This model is thus able to identify obstacles on unseen maps with 78.67% accuracy (Fig. 11).

To Navigation:

- To gain an understanding of collision probability, we created a random exploration algorithm without obstacle detection.
- In 500 steps of 7 actions each (3500 total actions), 84 collisions occurred, representing a 2.4% chance of collision.
- We then implemented our pathing algorithm with obstacle detection (Fig. 12).

Fig. 12: Video of the path planning algorithm with obstacle detection

Conclusions

Obstacle Detection via ResNet:

- Successful: Both the specialized and robust models were able to detect obstacles with high accuracy. However, there is a clear tradeoff between accuracy and scalability.
- The robust model was able to generalize to unexplored maps, meaning it learned the spatial and distance-based features to identify obstacles regardless of their size or position.

Pathing:

- The pathing algorithm was successful in obstacle avoidance, as no collisions occurred. However, because it was based on filtering out randomly generated actions, it was not very efficient at exploring the whole map. Even after 700 actions, some parts of the map were still unexplored.

Future Directions:

- Implement a gating module to generate a binary mask that excludes invalid obstacle ranges.
- Utilize real-world 2D or 3D LiDAR (particularly LiDAR point cloud data) to understand both horizontal and vertical features of obstacles for enhanced detection.
- Integrate neural network with BoxMap to understand room features like doors.

References

(1) Raut, G.; Patole, A. End-to-End 3D Object Detection Using LiDAR Point Cloud. arXiv, December 24, 2023. https://doi.org/10.48550/arXiv.2312.15377.
(2) Blum, P.; Crowley, P.; Lykotrafitis, G. Vision-Based Navigation and Obstacle Avoidance via Deep Reinforcement Learning. arXiv, November 9, 2022. https://doi.org/10.48550/arXiv.2211.05243.
(3) Xu, Z.; Shen, H.; Han, X.; Jin, H.; Ye, K.; Shimada, K. LV-DOT: LiDAR-Visual Dynamic Obstacle Detection and Tracking for Autonomous Robot Navigation. arXiv, February 28, 2025. https://doi.org/10.48550/arXiv.2502.20607.
(4) Li, T.; Ho, D.; Li, C.; Zhu, D.; Wang, C.; Meng, M. Q.-H. HouseExpo: A Large-Scale 2D Indoor Layout Dataset for Learning-Based Algorithms on Mobile Robots; GitHub, 2019. https://github.com/zhuxd/HausExpo (accessed 2025-08-04).}

Acknowledgements

Firstly, I would like to thank Professor Tron and Abdelrahman Abdelgawad for their mentorship throughout the research process. I am also grateful for the BU Robotics Laboratory for providing me with its cutting-edge resources and RISE for this excellent research opportunity. Lastly, I would like to thank my family for their endless support.