Practical implementation of Depth Anything V2 as a LiDAR alternative in robotics navigation

Jakobus Murray Louw[1*], *Jaco* Verster[1], and *John* Dickens[1]

[1]Centre for Robotics and Future Production, CSIR, Pretoria, South Africa

Abstract. This paper presents a systematic evaluation of monocular depth estimation (MDE) using Depth Anything V2 as an alternative to LiDAR for real-time obstacle perception in mobile robot navigation. We investigate the performance of an MDE-based perception pipeline integrated into a Nav2 navigation stack on a differential-drive robot, comparing it to a LiDAR baseline across three structured indoor courses. Qualitative analyses reveal that MDE offers superior spatial resolution, successfully capturing small obstacles often missed by sparsely sampled LiDAR beams. However, challenges such as localisation inaccuracies at long range and edge-induced artefacts are observed, particularly when the camera is forward-facing. These issues are significantly mitigated when using a high camera view angle, which reduces artefact streaking and improves localisation accuracy. Quantitative results show that the MDE system achieves comparable navigation performance to LiDAR, with similar travel distances, path consistency, and replanning deviation, despite greater variability in obstacle localization. The robot successfully achieved collision-free operation in all trials, demonstrating that MDE is a viable modality for local planning in structured environments. This work highlights the potential of vision-based depth estimation to complement or replace LiDAR in resource-constrained robotics applications and identifies key directions for improving its reliability and deployment efficiency.

1 Introduction

Accurate depth perception is an important component of autonomous robotic navigation, enabling robots to detect obstacles and plan safe paths in unknown environments. Traditionally, LiDAR has been the preferred technology due to its high accuracy and reliability, but its cost remains a significant drawback. Recent advances in deep learning have significantly improved monocular depth estimation (MDE), making models like Depth Anything V2 state-of-the-art in metric depth estimation [1,2,3].

MDE presents a cost-effective alternative to estimating the three-dimensional geometry of a scene due to more affordable sensor costs. The drawback, however, is that image processing requires significant compute resources, and the reliability of depth measurements is more susceptible to environmental conditions.

* Corresponding author: mlouw2@csir.co.za
** Language editing to improve clarity and readability of this paper was assisted by ChatGPT.

In a previous paper the authors assessed Depth Anything V2 as a LiDAR alternative and evaluated the geometric performance of the model in various indoor environments [4]. While the results suggested that MDE may not meet the precision requirements of high-accuracy navigation tasks, the question remained whether its performance would be sufficient for deployment in environments where lower accuracy would be acceptable.

This paper continues this work by specifically evaluating the feasibility of using Depth Anything V2 for autonomous navigation on a differential drive robot performing structured navigation tasks. The robot is tasked with navigating predefined courses using only MDE-based perception, and its performance is compared to a baseline setup using LiDAR-based perception. The experimental results provide insights into MDE's potential as a LiDAR alternative, highlighting both its advantages and the challenges that must be addressed for reliable deployment in robotics.

2 Related work

MDE has emerged as a crucial perception component for robotics systems, offering a cost-effective alternative to traditional depth sensing technologies. Recent research demonstrates significant progress in implementing these models on various robotics platforms, addressing challenges related to real-time performance, metric accuracy, and environmental robustness [5]. While MDE offers many advantages, it faces challenges in environments with limited scale cues, self-similar appearances, and low texture. Recent advances in deep learning have addressed some of these limitations through end-to-end trained models that generate dense depth maps. These developments have substantially improved both the performance and accuracy of depth prediction systems. Depth Anything V2 achieves more robust and accurate depth estimation by effectively combining synthetic data with advanced model training techniques.

The impressive performance of the Depth Anything V2 model has led to its popularity in competitions such as the 2024 MDE Challenge [2] and the NTIRE 2024 Challenge [3]. The MDE Challenge focuses on evaluating MDE techniques on the SYNS-Patches dataset. The dataset provides a challenging variety of natural and urban scenes, including forests, residential settings, industrial estates, lecture theatres, offices, and more [6]. The dataset is accompanied by a dense LiDAR ground-truth for evaluation of various metrics. In 2024 the competition was dominated by solutions using Depth Anything V2 as a backbone architecture. The model does well when evaluated for metrics such as point cloud reconstruction, image-based depth accuracy and success over depth discontinuities. The MDE Challenge authors noted that "top-performing entries generally leverage the pre-trained Depth Anything model" indicating that the model has been widely adopted.

MDE continues to evolve as a crucial technology for robotics as future research directions focus on computational efficiency, enhancing accuracy across diverse environments, and developing more sophisticated sensor fusion techniques. The research demonstrates that MDE has moved beyond theoretical concepts to practical implementation but to the authors knowledge no research exists that uses a MDE model to enable a robot to navigate through an environment.

3 Methodology

This section outlines the experimental setup used to evaluate Depth Anything V2 for monocular depth-based obstacle detection during autonomous navigation. It describes the robot platform, camera configurations, model deployment, and point cloud processing

pipeline, followed by the design of the structured test environments used to compare the performance of MDE against a LiDAR baseline.

3.1 Robot platform

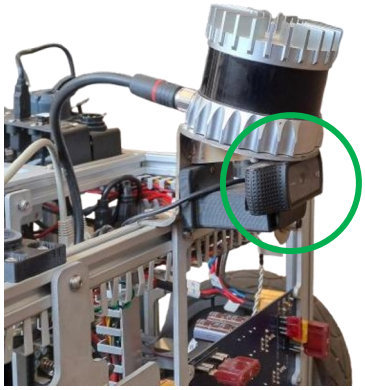
The evaluation of MDE for autonomous navigation was performed on the same Voyager mobile robotic platform [7] as the previous Depth Anything V2 geometric performance evaluation [4]. An additional elevated camera providing a third-person view was added to the robot in addition to its base Ouster OS0-32 LiDAR [8] and forward-facing webcam.

The navigation uses ROS2 [9] and Nav2 [10] with odometry based localisation for the path planning and navigation. For the limited size of the navigation courses, it was found that localisation based only on odometry was sufficiently accurate to be used for both the local path planning and the global navigation.

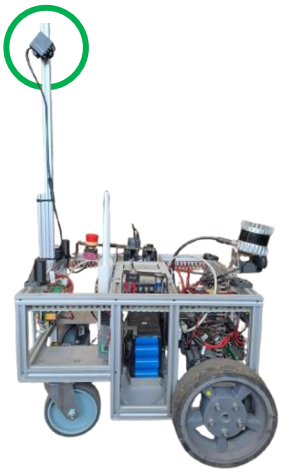
The cost-map for Nav2 was generated from the LiDAR or MDE point clouds using the Spatio-Temporal Voxel Layer (STVL) plugin [11]. The STVL plugin is a replacement for the *voxel_grid* representation of the environment with extended capabilities. Objects with a height greater than a threshold are considered obstacles and added to the cost-map as non-traversable with an inflation radius for the robot to safely clear the obstacle.

Based on previous work with the Voyager platform, the Nav2 setup was fine-tuned to navigate effectively using the on-board LiDAR as a baseline for comparison with the MDE perception.

Two USB cameras were used in this study, as shown in **Fig. 1**, including a C920 HD Pro Webcam [12] and an ELP 2-megapixel KL170IR camera with a fisheye lens [13]. The Logitech camera, a default component of the Voyager robot, was mounted front-facing with a 64° horizontal field of view and an optical axis parallel to the ground plane. Using the libraries available in OpenCV [14], camera calibration was performed using a chessboard pattern, and distortion was rectified using the plumb-bob camera model with radial and tangential distortion parameters.



Logitech camera with a forward-facing view.



ELP camera with a third-person view.

Fig. 1: Photos of the different cameras mounted on the Voyager robot.

The ELP camera, not part of the default Voyager configuration, was mounted on a rear column of the robot at a downward angle of 45° relative to the ground plane. It features a 170° horizontal field of view and provides a wide-angle, third-person perspective of the robot

and its immediate surroundings. This camera was calibrated using OpenCV [14], employing the fisheye camera model and corresponding distortion parameters to account for wide-angle lens characteristics.

3.2 Depth Anything V2 model

Depth Anything V2 advances the state-of-the-art in MDE [2]. The architecture of Depth Anything V2 focuses on enhancing the model's generalization capabilities by employing a teacher-student framework, utilising 595K synthetic images and 62M unlabelled images during training to perform well across diverse scenarios [15]. The indoor-tuned metric depth variant of Depth Anything V2 was used for this study. The largest (335.3M parameters) and most accurate model variant was chosen and used without changes or additional fine-tuning.

For this study, the depth estimation is processed in real-time on a computer with an Intel i9-11950H 2.60GHz CPU and a NVIDIA RTX A4000 Mobile GPU. Processing speeds of 12 Hz were achieved, which proved sufficient for real-time indoor navigation in static environments.

3.3 Point cloud processing

Point clouds from the inferred depth images were generated by projecting each pixel into 3D space using the intrinsic camera matrix, with the camera frame as the origin. Although Depth Anything V2 produces metric depth estimates, the resulting point clouds exhibit a consistent scale offset in the test environment. A constant scale factor is applied to the depth estimates to improve alignment between the MDE point clouds and the LiDAR-based obstacle positions. Applying this manually calibrated correction improves the spatial accuracy of perceived obstacles for local planning. The resulting point cloud is also down sampled to reduce downstream processing time.

A fixed transform between the camera and robot base frames is applied to express the point cloud in the robot's coordinate frame. In comparison, LiDAR data is natively in point cloud format and requires only a coordinate frame transformation before use in obstacle perception.

For both the camera and LiDAR point cloud data, the points are pre-processed to eliminate any misalignment between the measured ground plane and the local xy-plane of the robot. This ensures that the height-based obstacle detection works correctly. To estimate the ground plane, a RANSAC-based [16] model fitting is applied using the PCL [17] segmentation module. The algorithm is configured to detect the dominant plane roughly perpendicular to robot z-axis. Parameters such as maximum iterations, distance threshold, angular tolerance, and model probability are tuned to improve robustness. This detected plane is then used to compute the smallest transformation that aligns it with the robot's local xy-plane. We assume a planar floor, which holds true for most indoor environments and is valid for all experimental setups in this study.

Furthermore, a point cloud filter is applied to remove noise before detecting actual obstacles. Artefacts, particularly in the LiDAR-generated point clouds, often appear as small, transient clusters of points. To suppress these, a voxel grid filter is applied, wherein all points within a voxel are discarded if their count falls below a predefined threshold. This effectively removes spurious small clusters while preserving the integrity of the true point cloud structure.

For the MDE-based method, a dilated edge mask is generated from the depth image to detect depth discontinuities. All points outside the dilated mask are retained for point cloud generation, while edge-adjacent pixels are discarded. This selective masking suppresses

artefacts such as the upward warping of the floor at image boundaries, which could otherwise lead to false obstacle detections in height-based filtering. Since genuine obstacles in the environment are typically defined by strong depth discontinuities, this method reliably preserves meaningful features while removing artefactual geometry.

3.4 Experimental setup

To evaluate camera-based obstacle detection for autonomous navigation, three structured obstacle courses, labelled Course A, B, and C, were designed, as illustrated in Fig. 2. The courses represent only structured indoor environments; evaluation in outdoor or dynamic settings is left for future work. In each course, the Voyager robot begins at the origin of the map frame and is tasked with navigating to the target xy-position at coordinates (5, 0) while avoiding obstacles. As a baseline, the robot uses point cloud data from an Ouster OS0-32 LiDAR for obstacle detection. The same procedure is then repeated using the rear-mounted monocular camera in combination with Depth Anything V2 for MDE. In both cases, the navigation stack parameters are kept constant to ensure comparability between methods. Each course is traversed five times with each perception modality.

Course A *Course B* *Course C*

Fig. 2: Photos of predefined courses used in navigation experiments.

Initial tests using the front-mounted webcam for MDE resulted in frequent navigation failures, primarily due to its limited field of view and poor coverage of the robot's surroundings. In contrast, the rear-mounted camera, featuring a fisheye lens and a downward viewing angle, provided significantly improved coverage of the surroundings and more stable navigation performance. As a result, the rear-mounted camera was selected for all MDE trials reported in the quantitative results.

For each trial, a rosbag [9] file is recorded to facilitate post-processing of the results. Sensor data and cost-maps are visualised in RViz [9] for qualitative comparison. Additionally, several performance metrics, including travel distance and replanning deviations, are extracted and compared, as detailed in the sections below.

4 Qualitative results

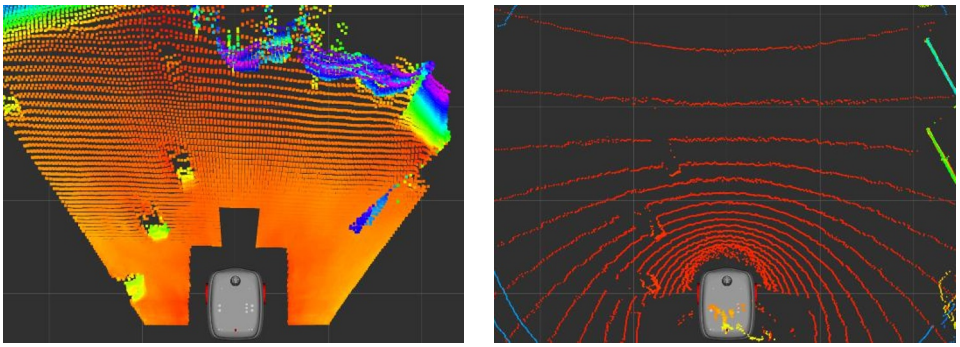
This section presents qualitative results from the experiments, focusing on the comparison MDE with LiDAR in terms of perception resolution, noise, artefacts, and obstacle localization. The analysis highlights the advantages and limitations of the MDE-based system through direct visual comparisons with LiDAR-derived outputs.

4.1 Perception resolution

Camera-based perception offers superior resolution compared to LiDAR, which can miss smaller obstacles at greater distances due to increased spacing between its discrete

measurement beams. In contrast, the high-resolution images captured by cameras provide a more detailed representation of the environment, enabling reliable detection of small obstacles that might otherwise be overlooked in LiDAR scans. This underscores the potential of camera-based perception to complement traditional LiDAR systems with higher spatial resolution.

Fig. 3 presents a side-by-side comparison of point clouds from MDE and LiDAR for the same scene in Course B. The MDE output provides a more detailed view of the environment and successfully captures small obstacles that are missed in the gaps between the widely spaced LiDAR beams at greater distances. Also note that nearby objects are perceived in comparable locations by both methods, whereas the MDE output erroneously includes distant objects that are absent in the LiDAR point cloud. This highlights a trade-off between spatial resolution and depth accuracy in the MDE and LiDAR approaches.



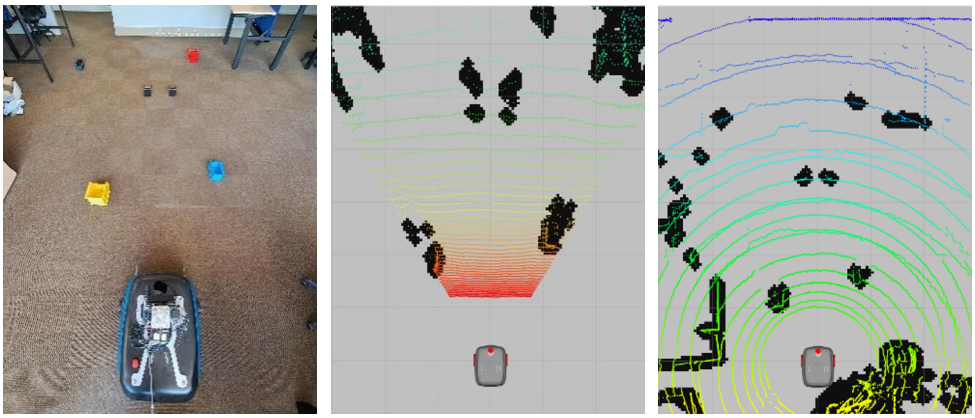
MDE *LiDAR*

Fig. 3: Point cloud visualisations of different perception methods in the same scene.

4.2 Noise and artefacts in MDE

Depth estimation models such as Depth Anything V2 often produce noise at object boundaries due to the smoothing of sharp depth discontinuities. These artefacts are subtle in the 2D depth image but become pronounced in 3D point clouds as elongated structures radiating along the camera's principal axis. This distorts scene geometry and results in false obstacle detections.

Fig. 4 presents an overhead photo of a course alongside the corresponding cost-maps generated with MDE (using the front-facing camera) and LiDAR, respectively. The MDE cost-map shows elongated obstacles due to edge artefacts, whereas the LiDAR output provides a more accurate representation. An edge filter can be applied to mitigate this issue by removing high-gradient pixels in the depth image before point cloud generation. However, this filtering reduces the spatial extent of detected features, potentially omitting small obstacles.



Overhead photo of Voyager facing obstacles

MDE point cloud and cost-map showing elongated obstacles

LiDAR point cloud and cost-map

Fig. 4: Overhead view of the test course alongside cost-maps generated using LiDAR and MDE with a front-facing camera.

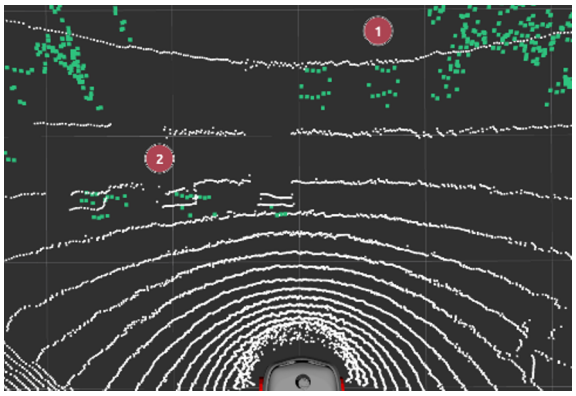
In this study, edge artefacts were more severe with the front-facing camera than with the rear-mounted camera. The increased noise is attributed to the forward-facing view angle, which aligns with the principal axis and thus exacerbates radial streaking from object boundaries. In contrast, the rear-mounted downward view angle results in more lateral depth transitions across the image plane, significantly reducing the impact of these artefacts.

4.3 Obstacle localization

Since Depth Anything V2 estimates depth from individual frames without incorporating temporal information, the resulting point clouds show variability in the perceived positions of obstacles across consecutive frames. This variability is most pronounced along the camera's principal axis, where depth is inferred rather than directly observed. In contrast, the image plane axes (x and y) show less variability, as pixel positions are directly measured and projected using the camera intrinsic matrix and therefore not subject to the same estimation uncertainty.

As a result, the obstacle locations perceived by the downward-facing camera exhibit lower variability compared to those from the front-facing camera. This is because, in the downward-facing configuration, most depth transitions occur across the image plane rather than along the principal axis, reducing sensitivity to depth estimation noise.

Fig. 5 shows an example scene from Course A, showing the LiDAR point cloud in white and the pre-processed MDE point cloud in green. In the region labelled as 2, the obstacle positions estimated by MDE closely align with those from the LiDAR. However, in region 1, the obstacles appear at approximately 2.2 m in the MDE point cloud, while their true distance from the robot is 3 m. This underestimation of depth at longer ranges was consistently observed in all tested courses. Obstacles closer to the robot were estimated more accurately and with less positional variability than those further away. To mitigate planning errors due to this variability, only obstacles within a 2 m radius are considered by the navigation stack during local path planning.



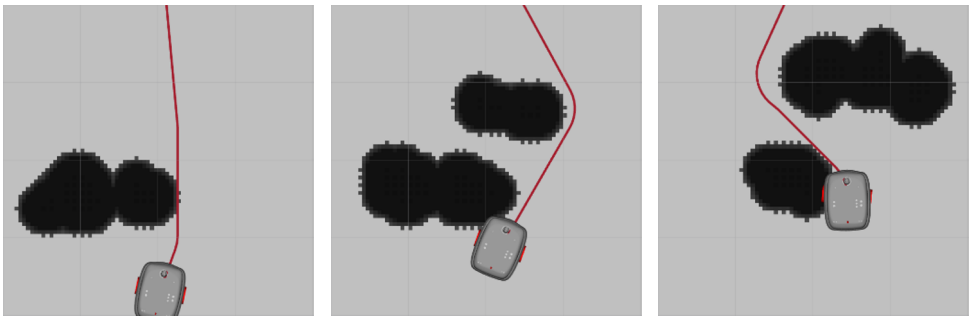
LiDAR point cloud (white), pre-processed MDE point cloud (green), and region labels (red)

Photo of the scene

Fig. 5: Point cloud visualisation and photo of a scene in Course A.

Note that obstacle perception is employed solely for local path planning in this study and is not used for mapping unknown space. Using obstacle perception for local planning confines any variability in obstacle localization to the immediate vicinity of the robot, reducing its potential impact on navigation decisions. Therefore, the courses considered are all designed as reactive environments that do not require exploration to find a valid path. In these environments, the robot navigates based on obstacles currently within its sensor field of view, without integrating persistent global maps.

Fig. 6 shows three sequential scenes of the robot navigating through Course B and illustrates the path replanning as new obstacles come into view. In each image, black pixels represent occupied cells in the cost map, while the red line indicates the current path generated by the planner. As the robot advances, the cost map is updated with newly perceived obstacles, resulting in corresponding updates to the planned path. The figure illustrates how the planner continuously adapts to the local environment as more obstacles enter the 2 m range.



Scene 1 *Scene 2* *Scene 3*

Fig. 6: Sequential scenes from Course B showing path replanning in response to new obstacles in the perception range.

5 Quantitative results

This section presents quantitative results comparing MDE and LiDAR-based navigation. Metrics include the actual travel paths, total distance travelled, planned path consistency, and

waypoint deviations during replanning. The analysis highlights how each perception method affect planning behaviour and overall navigation performance.

5.1 Travel paths

To visually compare the paths taken using MDE and LiDAR-based perception, the trajectories from each trial are plotted on a common axis for each course, as shown in Fig. 7. Each subplot corresponds to one of the three predefined obstacle courses. In all trials, the robot successfully completed the course without collisions. Minor variations in trajectory between trials are expected and can be attributed to sensor noise and asynchronous timing within the multi-threaded navigation stack.

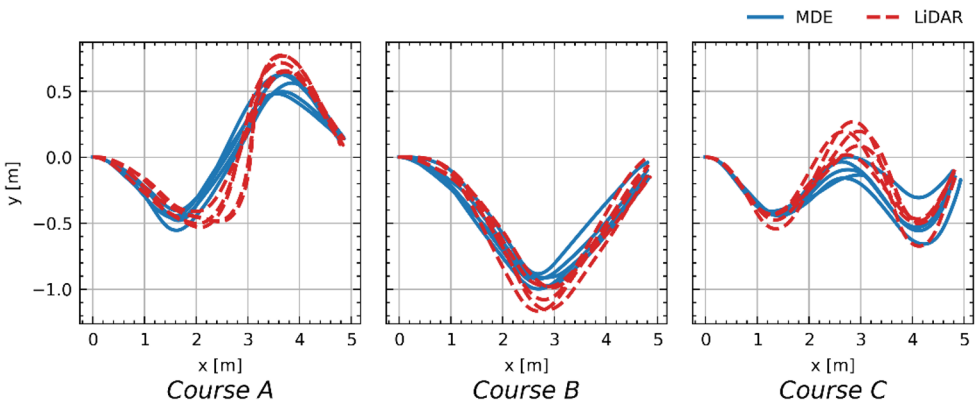


Fig. 7: Actual trajectories followed in each trail with MDE- and LiDAR-based navigation.

Interestingly, the robot consistently executes wider turns when navigating with LiDAR compared to MDE. This may be due to conservative obstacle inflation from noisy LiDAR returns or an underestimation of obstacle size in the MDE-derived point clouds, both of which can be mitigated through additional filtering techniques. Nonetheless, both methods exhibit tightly clustered paths, indicating good reproducibility and stable navigation performance across trials.

5.2 Box plot of total distance travelled

Fig. 8 presents box plots summarising the travel distances for each trial across the three courses using MDE and LiDAR-based navigation. MDE consistently results in shorter travel distances, likely due to tighter turns compared to the more conservative paths taken with LiDAR, as discussed in the previous section. In Courses A and B, the MDE trials exhibit a smaller spread in travel distances than LiDAR, indicating better consistency with MDE. In contrast, Course C shows a tighter spread for the LiDAR-based trials. However, the number of repetitions per course is limited, and the results should therefore be interpreted with caution.

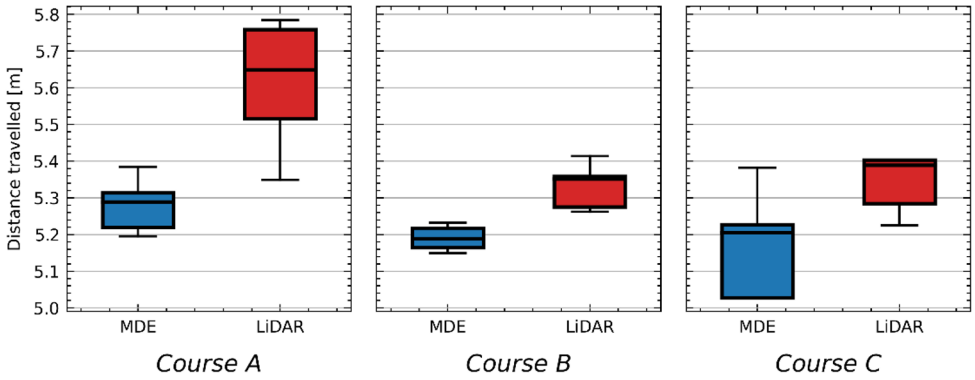


Fig. 8: Box plots showing travel distance distributions across all trials and courses.

5.3 Comparison of actual and planned paths

Fig. 9 presents the actual path alongside all planned paths from the first three trials of Course A. The columns of the subplot grid represent different trials, and the rows correspond to different perception methods. The x and y axes represent the global xy-frame of the obstacle course, with the robot starting at the origin. Refer to Fig. 2 for a photo of this course.

As shown in the figure, a new local plan is generated at sequential steps as the robot advances toward the goal, based on the obstacles detected in its immediate surroundings. Since the cost-map depends on the current perception range, the planned path evolves accordingly.

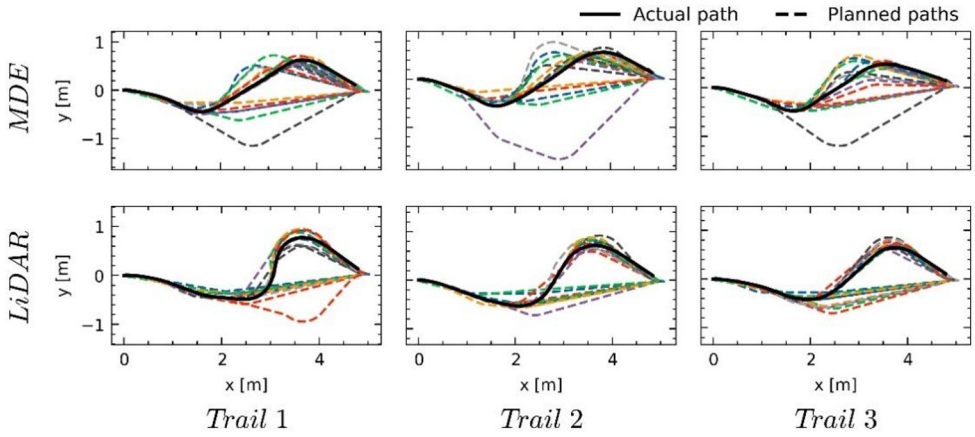


Fig. 9: Actual path and all planned paths for the first three trails of Course A.

The LiDAR-based method produces relatively consistent plans, with only two main path clusters visible. In contrast, the MDE method results in a broader spread of planned paths, with greater deviation between successive plans. This is likely due to the higher accuracy and stability of LiDAR in representing obstacle locations accurately when observed from different distances, leading to fewer changes in the cost-map. With MDE, depth estimates are more variable, which causes the perceived obstacle positions to shift more frequently and results in major course corrections. Nevertheless, the robot successfully avoids collisions in all trails with both perception methods.

Fig. 10 presents a similar plot for the first three trails of Course B. The diagonal row of obstacles in this course causes both methods to repeatedly replan in the first half, as new obstacles continuously enter the detection range. In the second half of the course, the path toward the goal remains unobstructed, and the planned paths form a more consistent cluster. As in Course A, the LiDAR-based plans exhibit a tighter cluster in the second half, reflecting lower perception variability, though the difference between the two methods is minimal. Despite sensor noise, both methods demonstrate strong reproducibility in path planning.

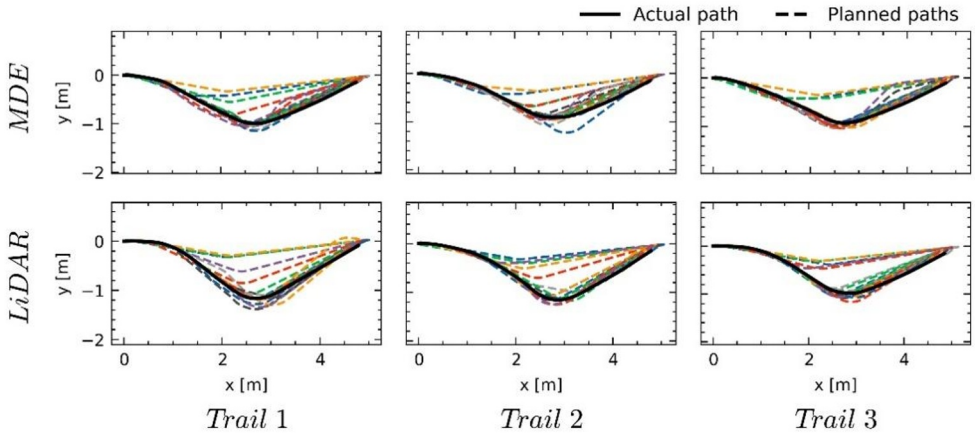


Fig. 10: Actual path and all planned paths for the first three trails of Course B.

A similar plot is presented in Fig. 11 for the first three trails of Course C. This course exhibits the highest variability in path planning, as the robot's path toward the goal is frequently obstructed by new obstacles. The MDE method shows the greatest deviations in replanning, particularly in Trail 2. The increased complexity of Course C, with a denser placement of obstacles, amplifies the variability in the MDE method, making the deviations more pronounced as the robot encounters new obstacles. Despite this variability, the MDE method successfully avoids collisions by consistently detecting obstacles in the robot's immediate vicinity. Although distant objects are localized with less accuracy, they are estimated more precisely as the robot approaches, allowing for successful obstacle avoidance.

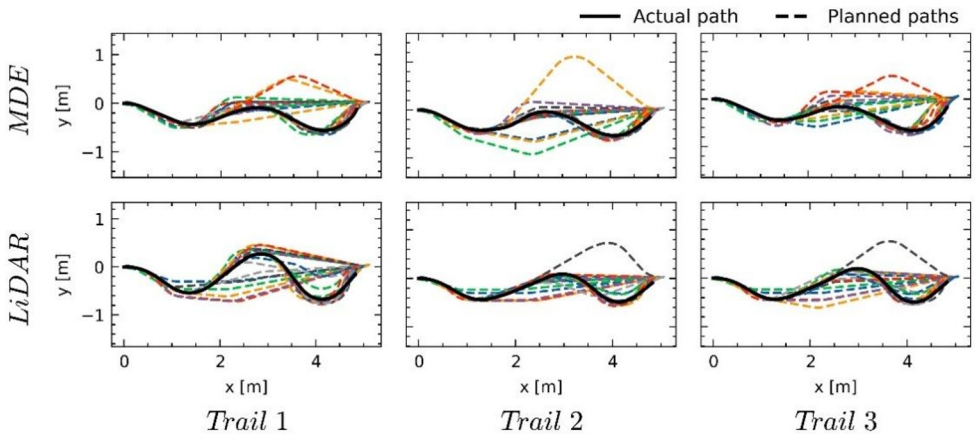


Fig. 11: Actual path and all planned paths for the first three trails of Course C.

5.4 Replanning events

The mean replanned waypoint deviation is calculated for each trail to quantify the spatial discrepancy between successively planned paths during navigation. It serves as a measure of how significantly the planner adjusts the robot's trajectory in response to new obstacles coming into view and variability caused by environmental factors. For each pair of consecutively planned paths, the initial alignment point is determined by locating the closest point in the previous path to the starting point of the new one. The paths are truncated to equal length, and the mean Euclidean distance between corresponding waypoints is computed. This metric captures the average positional change for waypoints across re-planning events, offering insight into the consistency and stability of the planning strategy under different sensing modalities.

Fig. 12 plots the mean replanned waypoint deviation for each replanning event index for all trails across the three courses. All courses were completed in less than 30 planning events, which run at 1 Hz. Since the planner runs at a consistent frequency, the replanning event index is approximately proportional to time. As each trail follows a similar path, corresponding indices generally align with similar positions along the course.

Notably, the deviation peaks occur at different indices for MDE- and LiDAR-based navigation, with MDE peaks consistently appearing earlier. This suggests that major course corrections are triggered sooner when using MDE, likely due to distant obstacles being inaccurately perceived as closer and within the local planning radius. Despite this temporal shift, the magnitudes of the deviation peaks are comparable across both perception methods, indicating similar levels of overall path planning stability.

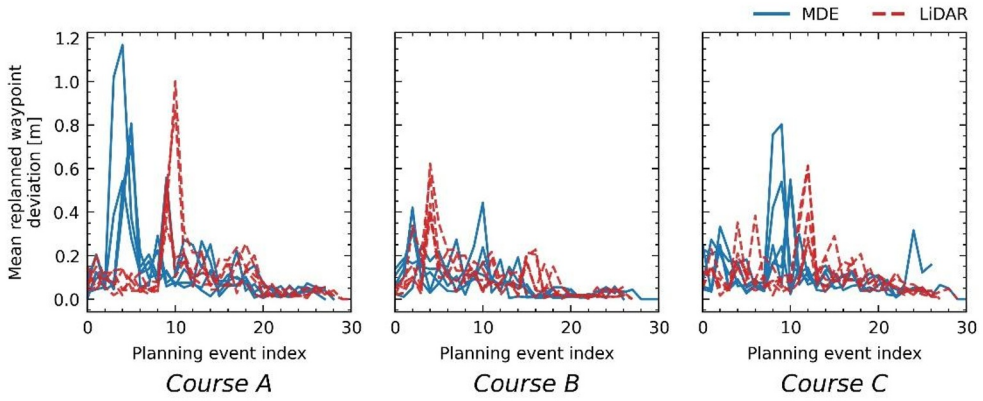


Fig. 12: Mean replanned waypoint deviation plotted against planning event index across all trails and courses

6 Conclusion

This study demonstrates that Depth Anything V2 can function as a real-time alternative to LiDAR for local obstacle perception in structured navigation tasks on a differential-drive robot. Across three obstacle courses, the MDE-based system consistently achieved collision-free navigation at 12 Hz, with comparable travel distances and replanning deviations to the LiDAR baseline. Qualitatively, the high-resolution point clouds captured small obstacles missed by widely spaced LiDAR beams and provided comparable obstacle localization within a 2 m local planning radius.

Current limitations of the MDE approach include systematic depth under-estimation at longer ranges, and edge-artifact streaking around depth discontinuities. Both effects reduce significantly with a higher camera view-angle. Furthermore, the model's inference demands necessitate a GPU for real-time performance, increasing the hardware complexity and power requirements of the robot. Despite these limitations, the system demonstrated reliable navigation performance in structured indoor environments, indicating that MDE is a viable perception modality for mobile robots in low accuracy applications when LiDAR is unavailable.

Future work may explore CPU-based inference to reduce computational requirements and improve accessibility on lower-cost platforms, reinforcing the potential of MDE as a cost-effective alternative to LiDAR. Additionally, integrating ground plane detection could enable reliable depth scaling, enhancing metric accuracy for improved obstacle localisation.

References

1. L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, H. Zhao, *Depth anything V2*, arXiv preprint arXiv:2406.09414v2, (2024).
2. J. Spencer, F. Tosi, M. Poggi, R. S. Arora, C. Russell, S. Hadfield, R. Bowden, G. Zhou, Z. Li, Q. Rao, Y. Bao, The third monocular depth estimation challenge, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 1-14 (2024).
3. P. Ramirez, F. Tosi., L. Di Stefano, R. Timofte, A. Costanzino, M. Poggi, J. Hwang, NTIRE 2024 challenge on HR depth from images of specular and transparent surfaces. Proc. IEEE Conf. Comput. Vis. Pattern Recognit., (pp. 6499-6512) (2024).
4. J.M. Louw, J. Verster, J. Dickens, Assessing Depth Anything V2 monocular depth estimation as a LiDAR alternative in robotics, in Proc. 2025 RAPDASA-RobMech-PRASA-AMI Conf., (under review).
5. U. Rajapaksha, F. Sohel, H. Laga, D. Diepeveen, M. Bennamoun, Deep learning-based depth estimation methods from monocular image and videos: A comprehensive survey, ACM Comput. Surv. 56, 1-51, (2024).
6. J. Spencer, C. Russell, S. Hadfield, R. Bowden. Deconstructing self-supervised monocular reconstruction: The design decisions that matter. arXiv preprint arXiv:2208.01489. (2022)
7. K. Purdon, J. Dickens, W. de Ronde, K. Ramruthan, G. Crafford, Voyager, a ground mobile robotic platform for research development, in Proc. 2023 RAPDASA-RobMech-PRASA-AMI Conf., (2023).
8. Ouster, OS0: Ultra-Wide View High-Resolution Imaging LiDAR Datasheet, REV: 12/2024, (2024), Available: https://data.ouster.io/downloads/datasheets/datasheet-rev7-v3p1-os0.pdf [Accessed: 29 January 2025].
9. S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, Robot Operating System 2: Design, architecture, and uses in the wild, Sci. Robot. 7 (2022).
10. S. Macenski, F. Martin, R. White and J. G. Clavero, "The Marathon 2: A Navigation System," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, 2020.
11. S. Macenski S, Tsai D, Feinberg M. Spatio-temporal voxel layer: A view on robot perception for the dynamic world. International Journal of Advanced Robotic Systems. 2020;17(2). doi:10.1177/1729881420910530
12. Logitech, C920 HD Pro Webcam, https://www.logitech.com/en-za/products/webcams/c920-pro-hd-webcam.960-001055.html [Accessed: 7 April 2025] (n.d.)

13. ELP, ELP 2Megapixel 1920×1080 IR Webcam, https://www.svpro.cc/product/svpro-2megapixel-19201080-ir-webcam-1-2-7-cmos-ov2710-mjpeg-30fps-60fps-120fps-ir-cut-filter-day-and-night-usb-camera/ [Accessed: 7 April 2025] (n.d.)
14. G. Bradski, The OpenCV Library, Dr. Dobb's J. Softw. Tools (2000).
15. L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, H. Zhao, *Depth anything: Unleashing the power of large-scale unlabeled data*, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, pp. 10371-10381, (2024)
16. M.A. Fischler, R.C. Bolles, Commun. of the ACM **24(6)**, 381-395, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, (1981)
17. R.B. Rusu, S. Cousins, Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 3D is here: Point Cloud Library (PCL), (2011)