Obstacle Detection Pipeline using Monocular Depth Estimation in Mobile Robotics

Christian Schweighofer
University of Applied Sciences Upper Austria
Roseggerstraße 15, Wels, 4600, Austria
S2410564020@fhooe.at

Michael Zauner
University of Applied Sciences Upper Austria
Roseggerstraße 15, Wels, 4600, Austria
Michael.Zauner@fh-wels.at

Abstract

Autonomous mobile robots must navigate dynamic environments safely, yet high-end depth sensors are often expensive or impractical. Monocular cameras are widely available, but estimating metric depth and detecting obstacles in real time remain challenging. We address this by implementing a pipeline that combines monocular depth estimation with metric scale calibration, 3D back-projection, filtering, and clustering. Our marker-based calibration achieves a depth RMSE as low as 13 mm, while the proposed pipeline successfully detects all 8 obstacles in our evaluation. With OpenVINO optimizations, the model achieves an inference rate of up to 17 FPS, establishing a foundation for real-time processing. Overall, the pipeline demonstrates promising results for safe navigation using only monocular cameras on resource-constrained robots, evaluated in the context of the international robotic contest Eurobot.

1 Introduction

In autonomous mobile robotics, reliable obstacle detection in dynamic environments is essential for avoiding collisions. While high-end platforms often employ sensors such as LiDAR or stereo vision to perceive the environment in 3D, these solutions impose significant hardware costs and spatial constraints. This is particularly relevant in competitive settings such as Eurobot, where robots must navigate dynamic environments while adhering to strict size limits [1, 10, 12].

Eurobot is an international robotic contest that takes place in Europe, with teams participating from around the world. The goal is to build a robot that performs tasks based on the yearly changing set of rules against another robot. The one scoring more points wins the match. A situation that illustrates the need for an obstacle detection system is shown in Figure 1.

Most Eurobot platforms already use monocular cameras and embedded compute for task-specific computer vision, such as ArUco [7] marker localization. This paper proposes a method that leverages monocular vision for obstacle detection. By integrating a foundation model (Depth Anything V2 [4, 18, 19]) with polynomial regression for metric calibration, we transform 2D imagery into metric 3D point clouds. The resulting workflow enables the detection of obstacles (e.g., Eurobot game elements) and thereby supports safer navigation without requiring additional sensors.

This work demonstrates a lightweight, ArUco-based calibration strategy for converting foundation-model depth into metric 3D point clouds and shows the feasibility of real-time obstacle detection on resource-constrained mobile robots.

The remainder of this paper is structured as follows: Section 2 details the processing pipeline from image to clustered obstacles; Section 3 evaluates the system using game elements from the Eurobot 2024, 2025, and 2026 seasons; Section 4 summarizes the results and outlines future work.

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

2 Methods

To detect obstacles and assess whether they may lead to a collision, information such as object size and distance is required. These parameters are difficult to extract directly from a 2D image. The following steps describe how an RGB image is converted into a 3D point cloud of the environment, which is then filtered and clustered to obtain obstacle hypotheses.

2.1 Monocular depth estimation

Recovering depth information from a single RGB camera is challenging [15]. In recent years, considerable research has been directed toward monocular depth estimation foundation models [3, 8, 13, 17]. These models take an image as input and predict per-pixel relative (and sometimes metric) depth. Their outputs are often proportional to either depth $\hat{y}$ or disparity $\hat{d}$ (inverse depth). Given the predicted relative depth $\hat{y} = \frac{1}{\hat{d}}$, the metric depth Z can be recovered through a linear transformation:

$$Z = a_1 \hat{y} + a_0. \tag{1}$$

The unknown regression coefficients a_0, a_1 can be estimated via a calibration procedure. Specifically, an optimization algorithm can be used to find parameters that minimize the error between ground-truth and predicted depth over a set of pixels. Results from [2] indicate that a second-order polynomial regressor ($Z = a_2\hat{y}^2 + a_1\hat{y} + a_0$) can further reduce the metric-depth error. The regression coefficients a_0, a_1, a_2 are identified from a sequence of frames containing an ArUco marker moving along the camera optical axis, where the absolute depth is obtained by solving the perspective-n-point (PnP) problem [11]. The regressor is tuned to minimize the error between the marker center depth estimated via PnP and the predicted depth at the corresponding pixel. This approach avoids the need for additional sensors during calibration.

2.2 2D-to-3D back projection with pinhole model and intrinsics

The standard pinhole model allows the conversion from 3D-space into the 2D-image plane via

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}. \tag{2}$$

This assumes that the intrinsic camera matrix $\mathbf{K}$ is known and that an undistorted image is used [20]. Given metric depth $Z_c = s$ for each pixel along the Z-axis, the formula can be rearranged to

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = Z_c \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{3}$$

This transformation projects the 2D depth map into a 3D point cloud. Initially, the points are expressed in the camera coordinate frame; other frames can be obtained by applying an extrinsic transformation.

2.3 Point cloud filtering

Once the environment is represented as a 3D point cloud, points that do not correspond to obstacles are filtered out. We remove points beyond a maximum distance by thresholding the Euclidean distance to the origin. In our experimental setup, most remaining points correspond to the planar floor; thus, we remove floor points by applying a height threshold parallel to the floor plane [14]. While RANSAC-based plane fitting or Hough-transform-based methods offer robust plane estimation [6, 9], a fixed-height pass-through filter was chosen for computational efficiency.

2.4 Object Clustering

The final step is to cluster the remaining points into obstacle candidates. For this purpose, we apply DBSCAN, a density-based clustering algorithm [5, 16]. Once clusters are obtained, properties such

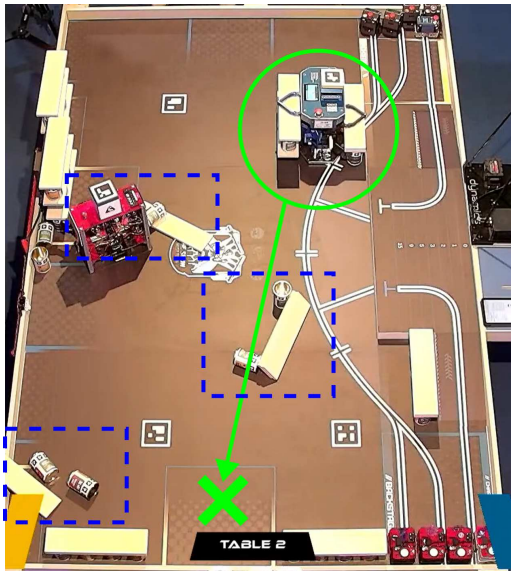

Figure 1: Match situation illustrating a potential obstacle encounter in Eurobot season 2025.

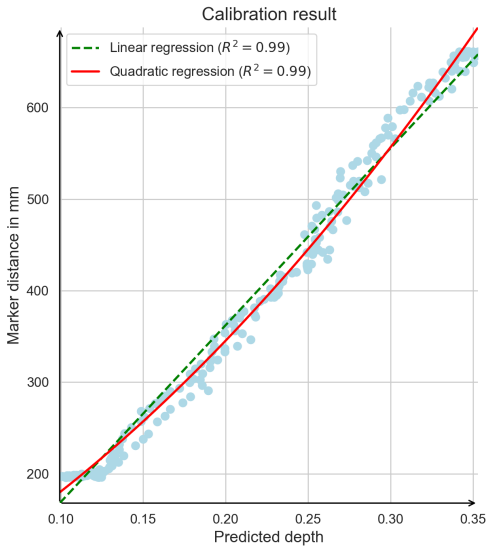

Figure 2: Calibration results of first and second order regression.

as obstacle size and distance can be estimated, enabling downstream decisions for path planning and collision avoidance.

3 Results

This section evaluates the results of the proposed obstacle detection workflow. The situations in Figure 3a, 3f, and 3k are used as test scenarios. The tests were conducted on a Surface Pro 7 with an Intel Core i5-1035G4 CPU and 8GB of RAM. The Depth Anything V2 - base model with an input image size of $(256, 256)$ pixels was used for inference.

The basis of the proposed workflow lies in an accurate conversion from 2D-image points into a 3D-spatial representation. The ground truth distance was obtained via PnP-estimation of an ArUco marker. The results of our calibration approach, which utilizes ground truth distance to find regressor parameters for conversion from relative to absolute depth, can be seen in Figure 2.

Consistent with the findings of [2], both the first-order and second-order polynomial regressors approximate the ground truth well. The first-order approximation achieves an RMSE of 18 mm, while the second-order polynomial achieves an RMSE of 13 mm. In the following experiments, we use the second-order polynomial.

After converting each pixel from 2D to 3D using Eq. 3, the resulting point clouds for the test scenarios are shown in Figures 3c, 3h, and 3m.

After removing points beyond 1000 mm and points below a height threshold of 30 mm above the floor, the filtered point clouds in Figures 3d, 3i, and 3n remain. DBSCAN ($\epsilon = 0.008$, min. points $= 20$) then segments the scene into discrete clusters and successfully detects all 8 obstacles in our evaluation (Figures 3e, 3j, and 3o).

While the above implementation shows promising results, its real-time performance is currently insufficient. Using the calibration frames, we measured inference time and calibration error on different models and optimizations, as shown in Table 1. Non-inference steps—including point cloud generation, filtering, and clustering—currently account for approximately 0.2 s per frame in the unoptimized Python implementation. These steps can be further accelerated using standard techniques such as voxelization, more efficient algorithms, and compiled code. Overall, these results indicate that the proposed pipeline is feasible for real-time obstacle detection on resource-constrained mobile robots.

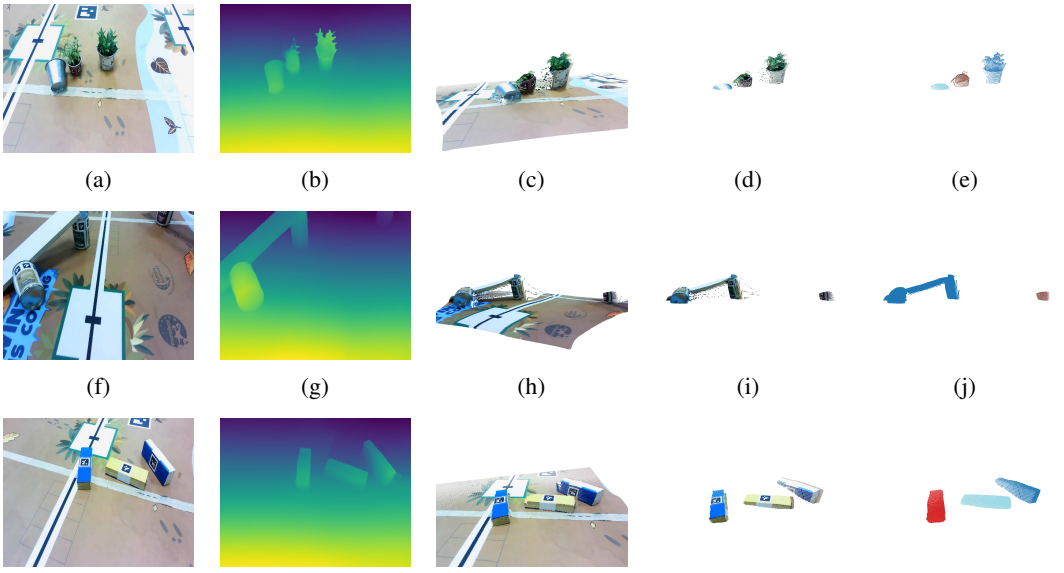

Figure 3: Sample obstacle-detection results for three seasons of Eurobot (2024–2026). Rows correspond to the different seasons, while columns illustrate the processing pipeline from left to right: undistorted RGB input, estimated depth map, 3D point-cloud projection, filtered point cloud, and final clustered obstacles.

Table 1: Inference time and calibration error (RMSE in mm) for selected depth models at 256×256 px, except OpenVINO (OV) at 252×252 px.

Model	Time [ms]	RMSE 1st order [mm]	RMSE 2nd order [mm]
DepthAnythingV2 Base	840	18	13
DepthAnythingV2 Small	325	36	35
DPT Hybrid [13]	1019	29	24
DepthAnythingV2 Small (OV)	57	25	24

4 Conclusion

In conclusion, we show that monocular depth estimation models can be used for obstacle detection in the Eurobot setting. Using an ArUco-based calibration procedure, we map relative predictions to a metric 3D point cloud representation and successfully detect representative game elements from recent seasons. However, metric accuracy of obstacle dimensions and distances beyond the calibration setup remains unverified, as the inferred scale was not validated on an independent dataset. While the current implementation is neither fully optimized nor evaluated on the target platform, the results indicate that the proposed pipeline is a viable alternative when dedicated depth sensors are impractical.

Future work should validate and improve the metric scale conversion on independent test data and under changes in camera pose, scene composition, and illumination. In addition, further engineering efforts could benchmark and optimize the pipeline on embedded platforms.

Acknowledgments and Disclosure of Funding

The authors used AI tools from OpenAI (ChatGPT) and Google (Gemini) to assist with writing and coding. All methodological choices and experimental results were independently designed and verified by the authors. The authors gratefully acknowledge the financial support of the University of Applied Sciences Upper Austria for this project.

References

[1] Seongmin Ahn, Yunjin Kyung, Seunguk Choi, Dongyoung Choi, and Dongil Choi. Monocular vision-based obstacle height estimation for mobile robot. *Applied Sciences*, 15(23), 2025. doi: 10.3390/app152312711.

[2] Soofiyan Atar, Yuheng Zhi, Florian Richter, and Michael Yip. Kinedepth: Utilizing robot kinematics for online metric depth estimation. 2025. doi: 10.48550/arXiv.2409.19490.

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth. 2023. doi: 10.48550/arXiv.2302.12288.

[4] Xingshuai Dong, Matthew A. Garratt, Sreenatha G. Anavatti, and Hussein A. Abbass. Towards real-time monocular depth estimation for robotics: A survey. 2021. doi: 10.48550/arXiv.2111.08600.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[6] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In Martin A. Fischler and Oscar Firschein, editors, *Readings in Computer Vision*, pages 726–740. Morgan Kaufmann, San Francisco (CA), 1987.

[7] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[8] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging Into Self-Supervised Monocular Depth Estimation. 2019. doi: 10.1109/ICCV.2019.00393.

[9] Rostislav Hulik, Michal Spanel, Pavel Smrz, and Zdenek Materna. Continuous plane detection in point-cloud data based on 3D Hough Transform. *Journal of Visual Communication and Image Representation*, 25(1):86–97, 2014. doi: 10.1016/j.jvcir.2013.04.001.

[10] Kornél Katona, Husam A. Neamah, and Péter Korondi. Obstacle avoidance and path planning methods for autonomous navigation of mobile robot. *Sensors*, 24(11), 2024. doi: 10.3390/s24113573.

[11] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81, 2009. doi: 10.1007/s11263-008-0152-6.

[12] Yu Liu, Shuting Wang, Yuanlong Xie, Tifan Xiong, and Mingyuan Wu. A review of sensing technologies for indoor autonomous mobile robots. *Sensors*, 24(4), 2024. doi: 10.3390/s24041222.

[13] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. 2021. doi: 10.48550/arXiv.2103.13413.

[14] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, 2011. doi: 10.1109/ICRA.2011.5980567.

[15] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

[16] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.*, 42(3), 2017. doi: 10.1145/3068335.

[17] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. 2024. doi: 10.1109/CVPR52733.2024.00987.

[18] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2, 2024. arXiv preprint arXiv:2406.09414.

[19] Jiuling Zhang. Survey on monocular metric depth estimation. 2025. doi: 10.48550/arXiv.2501.11841.

[20] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000. doi: 10.1109/34.888718.