engineering proceedings

MDPI

Proceeding Paper

Inverse Perspective Mapping Correction for Aiding Camera-Based Autonomous Driving Tasks †

Norbert Markó [1,2,*], Péter Kőrös [1] and Miklós Unger [1]

[1] Vehicle Industry Research Center, Széchenyi István University, H-9026 Győr, Hungary; koros.peter@ga.sze.hu (P.K.); miklos.unger@ga.sze.hu (M.U.)
[2] Machine Perception Research Laboratory, HUN-REN SZTAKI, H-1111 Budapest, Hungary
* Correspondence: marko.norbert@ga.sze.hu
† Presented at the Sustainable Mobility and Transportation Symposium 2024, Győr, Hungary, 14–16 October 2024.

Abstract: Inverse perspective mapping (IPM) is a crucial technique in camera-based autonomous driving, transforming the perspective view captured by the camera into a bird's-eye view. This can be beneficial for accurate environmental perception, path planning, obstacle detection, and navigation. IPM faces challenges such as distortion and inaccuracies due to varying road inclinations and intrinsic camera properties. Herein, we revealed inaccuracies inherent in our current IPM approach so proper correction techniques can be applied later. We aimed to explore correction possibilities to enhance the accuracy of IPM and examine other methods that could be used as a benchmark or even a replacement, such as stereo vision and deep learning-based monocular depth estimation methods. With this work, we aimed to provide an analysis and direction for working with IPM.

Keywords: inverse perspective mapping; deprojection; distance estimation; distance estimation correction

Citation: Markó, N.; Kőrös, P.; Unger, M. Inverse Perspective Mapping Correction for Aiding Camera-Based Autonomous Driving Tasks. *Eng. Proc.* **2024**, *79*, 67. https://doi.org/10.3390/engproc2024079067

Academic Editors: András Lajos Nagy, Boglárka Eisinger Balassa, László Lendvai and Szabolcs Kocsis-Szürke

Published: 7 November 2024

1. Introduction

Inverse perspective mapping (IPM) is a fundamental technique in camera-based autonomous driving systems, pivotal for converting the perspective view captured by the camera into a bird's-eye view and it is often combined with deprojection. This transformation is indispensable for accurate environmental perception, path planning, obstacle detection, and navigation. However, despite its critical role, IPM faces significant challenges, including distortion and inaccuracies arising from varying road inclinations and intrinsic camera properties [1]. These challenges can impede the overall reliability and effectiveness of autonomous driving systems utilizing this technique.

Our research focuses on the pinhole camera model-based IPM methodology and its inherent inaccuracy that must be addressed to optimize its application in real-world scenarios [2]. These inaccuracies often stem from the dynamic nature of road surfaces and the complex interplay of camera parameters. Consequently, there is a pressing need for correction techniques that can mitigate these errors and enhance the precision of IPM outputs. The IPM and deprojection algorithms are used in several works, but it is rarely discussed in and of itself, and to the best of our knowledge, there is no prior work on the analysis of deprojection errors for autonomous tasks [3–8].

In this study, we aimed to explore various correction possibilities to improve the accuracy of IPM. A significant gap in the current state of the art is the inherent inaccuracy of IPM algorithms, highlighting the need for extensive measurements and additional data to establish a more robust foundation for further research and improvements in these techniques. We also examined alternative approaches that could serve as benchmarks or even replacements for traditional IPM. These include stereo vision and deep learning-based monocular depth estimation methods. Stereo vision leverages the disparity between two camera views to infer depth, providing a more robust 3D understanding of the environment.

On the other hand, deep learning-based monocular depth estimation employs neural networks to predict depth from a single image, offering a promising solution that balances accuracy and computational efficiency.

By focusing on the enhancement of IPM through correction techniques and evaluating potential alternatives, this research aspires to serve as a roadmap for a more computationally efficient solution for distance estimation.

2. Materials and Methods

As we have mentioned before, inverse perspective mapping (IPM) transforms the perspective view captured by the camera into a bird's-eye view. To achieve accurate 3D understanding, IPM can be combined with deprojection techniques that convert 2D image points into their exact 3D coordinates. Deprojection lies at the heart of our work, serving as the core algorithm driving our approach. The basic idea is to leverage the inverse of the camera's intrinsic matrix (K) to transform 2D pixel coordinates into 3D space. To recover the depth information (the third dimension) for each pixel, we apply a few key assumptions (described later in this section). We break down the exact process of this depth recovery in the following paragraphs. Deprojection is the part where the challenging calculation happens; consequently, we need to focus on this part. The transformation to convert 3D coordinates to 2D homogeneous coordinates is mathematically described by the following equation:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \boldsymbol{K} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}, \tag{1}$$

where λ is the scaling factor, *K* is the intrinsic camera matrix describing the camera, (u v 1) ^{T} are homogeneous coordinates and (X_c Y_c Z_c) ^{T} are the 3D points in the camera coordinate frame. The process described by this equation is commonly known as projection, and this is our starting point [1,9]. We can take the inverse of the intrinsic camera matrix to calculate the 3D coordinates from the 2D points:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \lambda \boldsymbol{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{2}$$

The only problem is that the depth information is lost during the projection process. Specifically, when a 3D point is projected onto a 2D image plane, its depth (Z_c) is not directly observable from the image alone. This loss of information makes it challenging to accurately reconstruct the original 3D coordinates. To address this, additional constraints or information, such as multiple views or assumptions about the scene geometry, are often required to recover the depth and thus fully determine the 3D coordinates. We are building on the following assumptions. The road surface is flat (there are no elevations or depressions), we know the position and orientation of the camera relative to the road surface, that is, the transformation matrix [R t], and we choose the pinhole camera model as the mathematical model of the camera.

To calculate the deprojection of the 2D points, we take (2), but we use r(λ) instead of the 3D coordinates on the left-hand side because the depth Z_c of a point is lost during the projection process. By representing each pixel as a ray of light (a line basically), parameterized by λ, we essentially define a family of points along this ray (or line). Each point along the ray could correspond to a potential depth value. Using this approach allows us to model the ambiguity in depth and calculate the exact 3D coordinates once λ (the scaling factor needed to recover the true depth) is determined through further calculations.

The previously mentioned relationship is described with a line mathematically:

$$r(\lambda) = \lambda \boldsymbol{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, | \lambda \in R_{>0}, \tag{3}$$

and we are asking which scaling factor λ returns the correct 3D points. In this case, we are calculating the point intersecting the road surface (represented geometrically as a plane). The intersection of the line and the plane shows us the exact point on the line that represents the road surface; thus, the distance of that road surface point is recovered.

Following the meticulous implementation of the algorithm and precise calibration of the camera's position and orientation relative to the vehicle, we designed and conducted experiments to evaluate the accuracy of our proposed method. The goal of the experiments was to measure the inaccuracy of the deprojection algorithm. The camera we used was factory calculated and mounted on top of our SZEmission vehicle, and we set it to be perfectly level with zero roll, pitch or yaw with respect to the ground.

We measured the true distance from the ground directly under the camera to the selected distances marked with traffic cones and marker tapes. In the first experiment, the tapes were one meter apart and the traffic cones were five meters apart. In Figure 1, the starting tape closest to the vehicle was 3 m away, the first cone was 5 m away, and then the rest of the markings followed the rule defined above. This experiment only measured inaccuracies in the middle of the camera image with a lens focal length of 2.1 mm.

Figure 1. First experiment with lens focal length of 2.1 mm (1080 p resolution). Camera is factory calibrated and mounted to be perfectly level with the ground.

To measure the error near the image edges, we conducted another experiment, with another focal length of 4 mm. This change caused the cones in Figure 2 to appear closer while being at the same distance apart, starting from 5 m as in Figure 1. When the focal length is longer, the field of view is narrower, and the same object will appear closer. To make the second experiment comparable to the first one, we also measured the middle of the image here along with the edge.

Figure 2. Second experiment with lens focal length of 4 mm, including the edge of the image (2 K resolution). Camera is factory calibrated and mounted to be perfectly level with the ground.

3. Results

The quantitative results can be seen in Tables 1 and 2 for experiments one and two, respectively. If we compare experiment one with experiment two, the middle values are close to each other for the distance of 5 m, but somewhat different for 10 m. This can be explained with the different camera lenses and their different degree errors for a certain part of the image. The values also show that the measurements on the edges (right values) are farther from the middle values on the same image. This means that extra attention might be needed for the image edges even after rectification.

Table 1. Quantitative results of the first experiment (2.1 mm).

Distances	Middle Values
3 m	2.46
4 m	3.27
5 m	4.20
10 m	9.33
30 m	50.72

Table 2. Quantitative results of the second experiment (4 mm).

Distances	Middle Values	Right Values
5 m	4.04	3.96
10 m	8.06	7.69
15 m	12.47	11.45
20 m	16.91	15.29
25 m	21.86	19.10

The results are also plotted in Figures 3 and 4. These plots show that as we go forward in distance, the error is getting bigger, but it also stays relatively constant for the first 20 m.

Figure 3 shows us the results of several camera resolutions, including VGA, 720 p, 1080 p and 2 K resolutions. The experiments with the image edges only show the results with 2 K resolution.

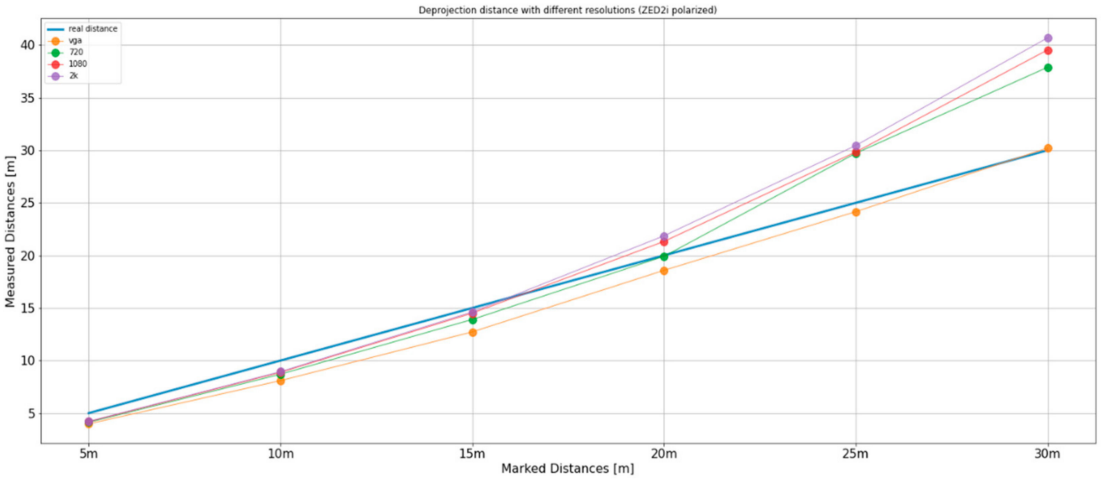

Figure 3. Plotted quantitative results of the first experiment (2.1 mm).

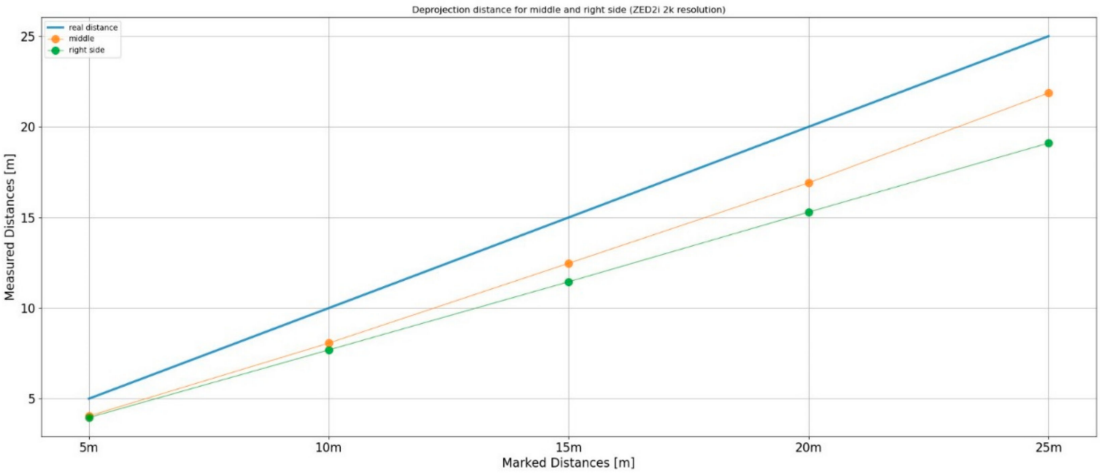

Figure 4. Plotted quantitative results of the second experiment (4 mm).

4. Discussion

In this work, we have shown that the deprojection algorithm has a slight intrinsic error, even when assuming a flat surface, but it can be a good distance estimation. Although the camera images are rectified to account for the lens distortions, there is still a level of error that we cannot fix with the adjustment of the intrinsic camera model. As we are getting further away on the image, a difference of one pixel means a much bigger difference than for pixels representing close ranges. This error needs improvement to make the algorithm more accurate and adaptable. External factors that can influence the algorithm's accuracy include variations in the scene geometry, such as inclines or surface irregularities, as well as camera-related issues like lens distortion and misalignment or calibration errors. Lighting conditions can also impact the accuracy, but only if the scene geometry is not assumed to be flat, as this would require proper assessment of the surface to account for depth variations. One way to further assess the accuracy of the algorithm would be to compare the results to some state-of-the-art alternatives, like in-built AI stereo camera algorithms or deep learning-based depth estimation algorithms. To perform these comparisons, we need to conduct measurements with cameras that have a stereo estimation enabled alongside the existing setup.

We are also currently experimenting with some recently released depth estimation algorithms (Figure 5), although they need calibration to return the absolute distances from it based on the relative distance results [10]. The primary sources of errors in depth estimation algorithms include inaccuracies in the intrinsic camera parameters (such as focal length and principal point) and errors in the camera's positioning and orientation relative to the road surface. Additionally, external factors like changes in scene geometry, lighting conditions, or surface irregularities can further affect the accuracy of depth recovery. In the future we would like to correct these problems by taking more measurements and fitting adaptive functions to account for the error, thus making the deprojection algorithm more accurate. Another plan is to compare this adjustment with depth estimation algorithms and the results of built-in stereo camera images.

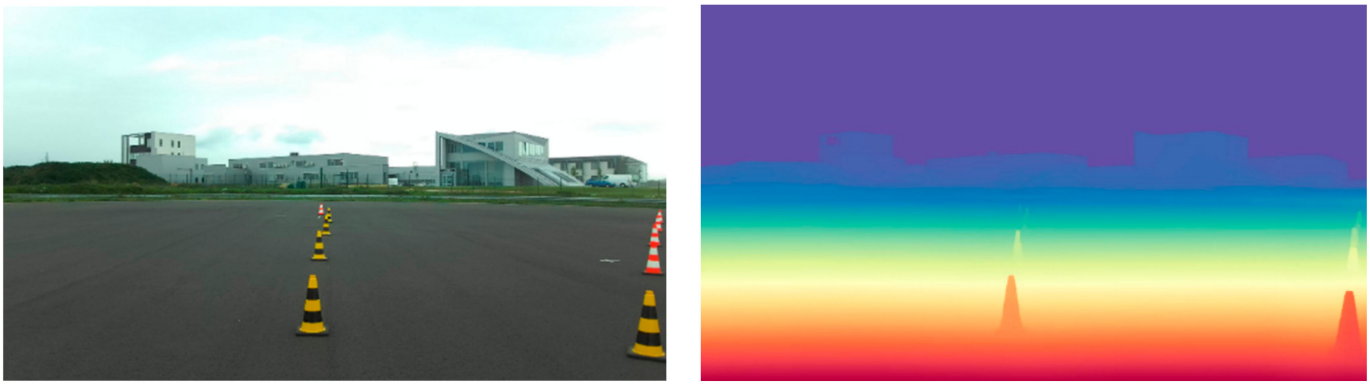

Figure 5. The Depth Anything V2 algorithm, showing the relative depth of the image used for our measurements. The **left** image shows the raw camera frame, the **right** image is the prediction result.

Author Contributions: Data curation, P.K. and M.U.; investigation, N.M.; writing—original draft preparation, N.M.; Writing—review and editing, N.M.; visualization, N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available online at: https://jkk-research.github.io/dataset/ (accessed on 14 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Theers, M.; Singh, M. Algorithms for Automated Driving. Available online: https://thomasfermi.github.io/Algorithms-for-Automated-Driving/Introduction/intro.html (accessed on 11 October 2024).
2. Szeliski, R. *Computer Vision: Algorithms and Applications*, 2nd ed.; Springer Nature: Cham, Switzerland, 2022; pp. 77–92, ISBN 978-3-030-34371-2.
3. Salman, Y.D.; Ku-Mahamud, K.R.; Kamioka, E. Distance measurement for self-driving cars using stereo camera. In Proceedings of the International Conference on Computing and Informatics, Kuala Lumpur, Malaysia, 25–27 April 2017.
4. Mu, X.; Ye, H.; Zhu, D.; Chen, T.; Qin, T. Inverse perspective mapping-based neural occupancy grid map for visual parking. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023.
5. Dhall, A.; Dai, D.; Van Gool, L. Real-time 3D Traffic Cone Detection for Autonomous Driving. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019.
6. Klappstein, J.; Stein, F.; Franke, U. Applying Kalman Filtering to Road Homography Estimation. In Proceedings of the Workshop on Planning, Perception and Navigation for Intelligent Vehicles in conjunction with IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy, 10–14 April 2007.
7. Man, Y.; Weng, X.; Li, X.; Kitani, K.M. GroundNet: Monocular Ground Plane Normal Estimation with Geometric Consistency. In Proceedings of the ACM Multimedia Conference (MM), Nice, France, 21–25 October 2019; pp. 2170–2178.

8. Zhang, J.; Wei, S.; Zhang, Q.; Chen, T.; Yang, C. Towards Accurate Ground Plane Normal Estimation from Ego-Motion. *Ital. Natl. Conf. Sens.* **2022**, *22*, 9375. [CrossRef] [PubMed]
9. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2003; pp. 325–343, ISBN 978-0-521-54051-3.
10. Yang, L.; Kang, B.; Huang, Z.; Zhao, Z.; Xu, X.; Feng, J.; Zhao, H. Depth Anything V2. *arXiv* **2024**, arXiv:2406.09414.