UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
THEORETISCHE INFORMATIK

Template for the LaTeX Class “uzl-thesis” for Bachelor’s and Master’s Theses Written at the University of Lübeck

Vorlage für die LaTeX-Klasse »uzl-thesis« zur Nutzung bei Bachelor- und Masterarbeiten an der Universität zu Lübeck

Bachelorarbeit

verfasst am
Institut für Theoretische Informatik

im Rahmen des Studiengangs
Informatik
der Universität zu Lübeck

vorgelegt von
Max Mustermann (alias Till Tantau)

ausgegeben und betreut von
Prof. Dr. Petra Wichtig-Wichtig

mit Unterstützung von
Harry Hilfreich

Lübeck, den 1. Januar 2021

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Max Mustermann (alias Till Tantau)

Zusammenfassung

Es ist nicht leicht, eine Abschlussarbeit so zu schreiben, dass sie nicht nur inhaltlich gut ist, sondern es auch eine Freude ist, sie zu lesen. Diese Freude ist aber wichtig: Wenn die Person, die die Arbeit benoten soll, wenig Gefallen am Lesen der Arbeit findet, so wird sie auch wenig Gefallen an einer guten Note finden. Glücklicherweise gibt es einige Kniffe, gut lesbare Arbeiten zu schreiben. Am wichtigsten ist zweifelsohne, dass die Arbeit in gutem Deutsch oder Englisch verfasst wurde mit klarem Satzbau und gutem Sprachrhythmus, dass keine Rechtschreib- oder Grammatikfehlern im Text auftauchen und dass die Argumente der Autorin oder des Autors klar, logisch, verständlich und gut veranschaulicht dargestellt werden. Daneben sind aber auch gut lesbare Schriftbilder und ein angenehmes Layout hilfreich. Die Nutzung dieser LaTeX-Vorlage hilft der Schreiberin oder dem Schreiber dabei zumindest bei Letzterem: Sie umfasst gute, sofort nutzbare Designs und sie kümmert sich um viele typographische Details.

Abstract

It is not easy to write a thesis that does not only advance science, but that is also a pleasure to read. While the scientific contribution of a thesis is undoubtedly of greater importance, the impact of *writing well* should not be underestimated: If the person who grades a thesis finds no pleasure in the reading, that person are also unlikely to find pleasure in giving outstanding grades. A well-written text uses good German or English phrasing with a clear and correct sentence structure and language rhythm, there are no spelling mistakes and the author's arguments are presented in a clear, logical and understandable manner using well-chosen examples and explanations. In addition, a nice-to-read font and a pleasing layout are also helpful. The LaTeX class presented in this document helps with the latter: It contains a number of ready-to-use designs and takes care of many small typographical chores.

Acknowledgements

This is the place where you can thank people and institutions, do not try to do this on the title page. The only exception is in case you wrote your thesis while working or staying at a company or abroad. Then you should use the Weitere␣Unterstützung key to provide a text (in German) that acknowledges the company or foreign institute. For instance, you could use texts like »Die Arbeit ist im Rahmen einer Tätigkeit bei der Firma Muster GmbH entstanden« or »Die Arbeit ist im Rahmen eines Forschungsaufenthalts beim Institut für Dieses und Jenes an der Universität Entenhausen entstanden«. Do not name and thank individual persons from the company or foreign institute on the title page, do that here.

Contents

1

Introduction

The increasing complexity of modern automotive systems has led to a growing demand for precise in-cabin monitoring technologies. Among these, steering wheel detection plays a pivotal role in monitoring driver behavior, enhancing safety, and facilitating autonomous control in advanced driver-assistance systems. Estimating the 3D location and orientation of the steering wheel is crucial for systems that rely on driver interaction and vehicle control. Yet, achieving accurate 3D detection of such an object within a car's interior presents unique challenges, including variations in rotation and position within confined, occlusion-prone environments. This is because the car's interior is a complex and constrained environment, with limited sensor coverage and frequent occlusions of key components like the steering wheel. Accurately determining the 3D position and orientation of the steering wheel is essential for enabling advanced driver monitoring and assistive technologies, but poses significant technical hurdles compared to object detection in open, outdoor environments.

1.1 Contributions of this thesis

This thesis addresses the problem of steering wheel detection using a 3D point cloud, which offers a rich representation of spatial information. Specifically, this research contributes in two significant ways: First, it introduces a novel dataset that represents the car interior, including 3D point clouds and annotated bounding boxes for the steering wheel. This dataset was created through a meticulous process involving the recording of point cloud data, feature extraction, and preprocessing to ensure data integrity and clarity. By accurately determining the steering wheel's 3D position and orientation, this dataset serves as a critical foundation for model training and validation.

Second, this thesis proposes a model-based approach for detecting the steering wheel's 3D position using a modified VoxelRCNN architecture. Existing 3D object detection models, such as those trained on KITTI dataset data, are typically optimized for outdoor environments with non-cubic point cloud dimensions and restricted rotational degrees of freedom (DOF). To overcome these limitations, the research modifies the VoxelRCNN architecture to account for a cubic point cloud structure and allows for rotation detection along the x-axis, which is crucial for accurately representing the orientation of the steer-

ing wheel.

Through these contributions, this research aims to advance the field of interior monitoring and lay groundwork for more robust in-cabin perception systems.

1.2 structure of this thesis

The remainder of this thesis is structured as follows. Chapter 2 reviews relevant literature on 3D object detection and the limitations of existing techniques in car interior applications. Chapter 3 details the dataset creation process, including data acquisition, preprocessing, and attempts at accurate position estimation, comparing two approaches to establish a reliable ground truth. Chapter 4 describes the proposed 3D object detection model and necessary modifications for steering wheel detection in cubic point clouds. Chapter 5 presents experimental results and analyzes the model's performance. Finally, Chapter 6 discusses the significance of the findings, potential applications, limitations, and directions for future research.

2

Background and related works

In this section, we explore the foundational concepts and advancements in 3D object detection techniques that are relevant to in-cabin monitoring and steering wheel detection. Given the critical role of perception in autonomous driving, achieving robust and reliable 3D detection of interior components like the steering wheel has become a key challenge in this field. This chapter presents an overview of various 3D object detection methods, focusing on point cloud-based approaches, multimodal fusion techniques, and image-based approaches. We also discuss the unique challenges posed by interior environments, including issues such as occlusions, irregular lighting conditions, and limited rotational freedom for 3D bounding boxes, which make existing detection methods difficult to apply directly to car interior scenarios. Finally, we provide an overview of publicly available datasets and the key performance metrics commonly used in 3D object detection for autonomous driving, assessing their relevance and applicability for interior monitoring applications.

2.1 3D Object Detection in Autonomous Driving

3D object detection in autonomous driving enables the vehicle to understand and perceive its surrounding environment by identifying, localizing, and characterizing nearby objects. In this context, 3D detection provides both precise spatial positioning and orientation information for objects, making it an essential component for crucial tasks such as path planning, collision avoidance, and motion prediction. However, current 3D object detection methods are primarily optimized for exterior, outdoor environments, where sensor setups, object dynamics, and available datasets differ significantly from the more constrained and complex interior scenarios found within a vehicle. Outdoor 3D detection has greatly benefited from the availability of large-scale datasets like KITTI, nuScenes, and Waymo, which provide extensive labeling for objects in open, relatively unobstructed spaces, captured under varied lighting and weather conditions. These datasets and approaches, however, do not adequately address the intricacies and unique challenges involved in detecting objects within a vehicle's constrained and complex interior space.

Challenges in 3D Detection for Interior Monitoring

Detecting objects in the car interior, specifically the steering wheel, involves overcoming several challenges:

- Limited Field of View (FOV): Unlike outdoor detection, where sensors cover a 360° view around the vehicle, interior sensors are constrained to the car's internal structure, resulting in limited and often obstructed views of certain objects.
- Occlusions and Sparse Data: Objects like the steering wheel may be partially or fully occluded by the driver's hands or other interior elements, and point cloud data from LiDAR or other depth sensors can be sparse due to occlusion and reflective surfaces.
- Variability in Rotational Degrees of Freedom (DOF): While outdoor models assume limited rotational DOF (often around the z-axis for vehicles), steering wheel detection requires accurate modeling of rotation around the x-axis to capturithe essential rotational characteristics of the steering wheel.
- Adaptability to Point Cloud Dimensions: Standard point clouds from datasets like KITTI have unbalanced dimensions, favoring a broader lateral view. In contrast, interior point clouds are typically cubic, and models must adapt to these differences for effective detection.

2.2 3D Object Detection Methods

3D object detection techniques are generally categorized based on the input modality (image-based, point cloud-based, or multimodal fusion) and architecture type (two-stage, single-stage, or hybrid). Table 2.1 shows a summary of the key approaches and their characteristics.

Image-Based Methods

Image-based 3D detection methods rely on RGB images or stereo images to infer depth and spatial information. These methods are cost-effective and widely applicable, but they lack direct depth information, which makes it challenging to accurately localize objects in 3D space. Image-based approaches typically involve:

- Result-Lifting: Detecting objects in 2D and projecting their bounding boxes to 3D space based on geometric constraints or templates, as in methods like Stereo R-CNN [6].
- Feature-Lifting: Estimating depth maps from images and creating pseudo-LiDAR [16] point clouds, which can then be processed similarly to actual LiDAR data. Approaches like Pseudo-LiDAR++ [19] and DSGN [2] have shown promising results, but they still struggle with accuracy over long distances and under variable lighting, which is especially problematic in the dimly lit car interiors.

Table 2.1: Categorization of 3D object detection approaches by input modality and architecture type

Category	Method	
Image-Based Methods	Result-Lifting	Stereo R-CNN [6]
	Feature-Lifting	Pseudo-LiDAR [16]
		Pseudo-LiDAR++ [19]
		DSGN [2]
Point Cloud-Based Methods	Voxel-Based	VoxelNet [**voxnet**]
		VoxelRCNN [3]
		SE-SSD [17]
		BtcDet [18]
	Point-Based	PointRCNN [11]
		Point-GNN [12]
		PointNet++ [8]
	Hybrid	PV-RCNN [10]
Multimodal Fusion-Based	Early Fusion	AVOD [5]
		MV3D [14]
	Deep Fusion	PointPainting [15]
	Sequential	Frustum PointNets [9]

Point Cloud-Based Methods

Point cloud-based methods are better suited for direct 3D localization, using data from LiDAR or other depth sensors to create dense, spatial representations. These methods can be further divided into voxel-based, point-based, and hybrid approaches:

- Voxel-Based Methods: VoxelNet [**voxnet**] and its successors, such as VoxelRCNN [3], partition the point cloud into voxel grids and apply convolutional operations, capturing spatial structure but at a high computational cost. Another effective voxel-based approach is SE-SSD (Self-Ensembling Single-Stage Detector) [17] that improves on voxel-based approaches by using a student-teacher framework. The student model learns from a teacher model that is refined with additional structure-aware information, improving robustness and accuracy in challenging conditions such as occlusions and sparse point clouds. Another voxel-based approach, BtcDet [18], focuses on long-range and sparse data detection. BtcDet enhances detection performance in challenging scenes by using shape-adaptive learning to manage low-quality data, which is particularly useful in adverse weather or complex scenes.
- Point-Based Methods: Approaches like PointRCNN [11] and Point-GNN [12] use permutation-invariant operations on raw point clouds, maintaining fine-grained details but often requiring high computational resources and memory. For example, PointNet++ [8] , a popular point-based method, hierarchically extracts features from individual points, preserving fine-grained geometric details but demanding significant computational power.

– Hybrid Methods: Combining voxel and point approaches, methods like PV-RCNN [10] aim to balance computational efficiency with spatial accuracy by processing both coarse-grained voxel features and fine-grained point details. This hybrid approach has proven advantageous for complex environments but has yet to be fully adapted for indoor environments with unique constraints, like a car interior.

Multimodal Fusion-Based Methods

Multimodal fusion combines RGB images and point cloud data to enhance detection accuracy by leveraging complementary data. Early, deep, and late fusion strategies vary based on when the modalities are combined in the model. PointPainting [15], for example, demonstrates deep fusion by blending intermediate features from both modalities, while methods like AVOD [5] and MV3D [14] rely on early fusion to incorporate image information directly into the point cloud processing pipeline. Additionally, Frustum PointNets [9], for instance, fuses LiDAR and image features within 3D frustums projected from 2D object detection results, effectively reducing the search space for 3D object detection. However, aligning features between images and point clouds in indoor environments poses significant challenges, particularly due to spatial and view misalignments.

2.3 Datasets for 3D Object Detection in Autonomous Driving

The field has developed multiple benchmark datasets, each with unique attributes for testing 3D detection models:

– KITTI: One of the earliest and most influential datasets, KITTI [4] includes over 200,000 labeled objects captured in outdoor urban settings. However, its limited class diversity and lack of interior scenes make it less suitable for in-cabin monitoring.
– nuScenes [1] and Waymo [13]: These datasets provide richer scenes, diverse weather and lighting conditions, and multimodal data, including LiDAR, radar, and cameras. Although they cover more object classes than KITTI, they still focus on outdoor scenarios.

The absence of publicly available in-cabin datasets emphasizes the need for specialized datasets to meet the requirements of car interior monitoring tasks.

2.4 Performance Metrics for 3D Object Detection

3D object detection performance is commonly evaluated through metrics such as Average Precision (AP) and mean Average Precision (mAP). Additional metrics include:

– Intersection over Union (IoU): IoU evaluates the overlap between predicted and ground truth bounding boxes. For autonomous driving, models are often tested at varying IoU thresholds depending on object size and importance.

– NuScenes Detection Score (NDS) and Average Precision with Heading (APH): Used in the nuScenes and Waymo datasets, these metrics account for orientation and other attributes specific to 3D space, offering a comprehensive measure of performance for complex scenes.

2.5 Summary

In summary, while 3D object detection has seen substantial advancements in outdoor environments, these methods face limitations when applied to in-cabin monitoring. The unique challenges of interior environments require adapted models, tailored datasets, and precise metrics that can account for both spatial constraints and occlusions. In the following chapters, we detail our approach to address these gaps by creating a dedicated dataset and adapting a 3D object detection model, optimized for steering wheel detection and localization within a confined interior space.

3

SWD dataset

The unique constraints of in-cabin monitoring and the specific requirements for steering wheel detection necessitated the development of a custom dataset. Existing datasets, such as KITTI and nuScenes, predominantly focus on external vehicle perception, capturing objects like cars, pedestrians, and road signs. These datasets are tailored for outdoor environments and often lack the spatial resolution, balance, and context required for confined in-cabin spaces. Furthermore, the irregular rotational degrees of freedom (DOF) in interior setups, along with occlusions and limited sensor viewpoints, further complicate the application of outdoor 3D detection models to in-cabin scenarios.

To address these limitations, a dataset specifically designed for steering wheel detection was developed, incorporating high-quality 3D point clouds and ground truth bounding boxes to capture the full 3DOF of the steering wheel's position and

The diversity in subjects, poses, and conditions in this dataset provides a comprehensive foundation for training and evaluating 3D object detection models specifically for steering wheel estimation. The following sections outline the methodology used for recording, data extraction, and ground truth generation, detailing the steps taken to produce high-fidelity 3D representations suitable for accurate steering wheel detection and orientation modeling.

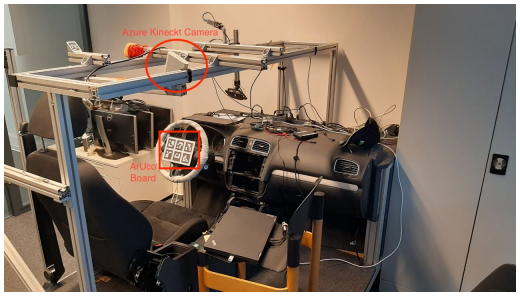
3.1 Recording Procedure

To accurately capture the steering wheel's position across multiple configurations, I set up a recording protocol using a board with ArUco markers[1] centered on the steering wheel as shown in fig. 3.1a. ArUco board helps with estimating position and orientation of the steering wheel in 3D space. A positional reference card, similar to fig. 3.1b, was placed adjacent to the steering wheel, displaying 16 distinct steering wheel orientations for standardized positioning. This approach allowed me to systematically record the steering wheel in various orientations, providing a comprehensive dataset for training and evaluating 3D object detection models focused on steering wheel estimation.

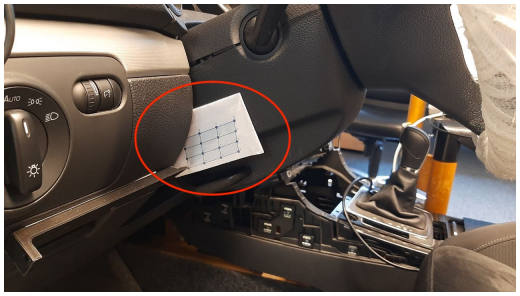
The video recordings were conducted using an Azure Kinect camera embedded in a car demo model as demonstrated in fig. 3.1a. This camera was positioned on the ceiling,

[1] https://docs.opencv.org/3.4/db/da9/tutorial_aruco_board_detection.html

slightly over the driver's shoulder, and angled to capture the entire steering wheel.



(a) ArUco board placed at the center of the steering wheel and the Azure Kinect camera over the driver's shoulder on the aluminium frames.



(b) A reference card was positioned next to the steering wheel, illustrating 16 unique steering wheel orientations to ensure standardized alignment.

Figure 3.1: The demo model setup used for recording streams.

The dataset was composed of recordings from 10 unique participants, with each participant engaging with the steering wheel in 16 distinct orientations. To introduce variability, the participants wore various accessories during the recordings. This resulted in a comprehensive dataset encompassing 72,747 training frames, 42,712 validation frames, and 29,099 testing frames. The steering wheel itself was covered with a white fabric to reduce reflections and enhance detection accuracy.

The dataset was recorded in two phases for each subject. First, a ground truth stream was captured with no driver present and the marker board fully visible as shown in fig. 3.2a. This short footage was used to extract accurate ground truth data on the position and orientation of the steering wheel. Subsequently, a second stream was recorded with the driver interacting with the steering wheel, simulating a driving scenario. During these main recordings, the marker board was covered with a black surface to prevent unintended biases in the model training as illustrated in fig. 3.2b. The driver performed typical driving actions, such as steering with one or both hands, operating the dashboard, and other natural behaviors.

For data processing, ground truths were extracted from the initial ground truth streams, while the main streams with the driver were used to extract point clouds for further experimentation.

3.2 Data Extraction

In the initial data extraction phase, depth and amplitude information was derived from the video streams to create high-quality point clouds. The raw point clouds were then refined through a series of preprocessing steps to enhance their quality and suitability for accurate steering wheel position detection:

– **Noise Reduction**: Filtering out extraneous points to achieve a clear and uncluttered point cloud.

(a) Ground truth data on the steering wheel's position and orientation were obtained from a stream recorded without a driver, ensuring full visibility of the marker board.

(b) A second stream was recorded with the driver interacting with the steering wheel, simulating a driving scenario, while the marker board was covered to prevent biases in model training.

Figure 3.2: Two stages of recording streams.

- **Distortion Correction**: Applied using OpenCV's distortion matrices to enhance the clarity and accuracy of both depth and amplitude images.
- **Edge Removal**: Points on the edges of the point cloud were omitted due to their ambiguous positioning, which could reduce the overall accuracy of the model.
- **Cropping**: The point cloud was cropped to a 50cm x 50cm x 50cm area around the steering wheel, ensuring that the data was focused and relevant for model training and reducing the dataset size.

These comprehensive preprocessing steps helped to create precise, refined point clouds that were well-suited for accurately detecting the position of the steering wheel.

3.3 Ground Truth

To generate ground truth data for the steering wheel's position and orientation, I explored two approaches due to the challenges presented by point sparsity and data distortion. The first approach, which involved annotating 2D points on images of the steering wheel and then mapping them to 3D space, failed to accurately represent the true position and orientation of the steering wheel. This was because the sparse and noisy distribution of the sampled points resulted in an imprecise oriented bounding box that did not capture the steering wheel's actual location and orientation with sufficient precision.

In contrast, the second approach using ArUco markers [7] and plane fitting proved to be a much more effective method. By first accurately estimating the 3D location of

the ArUco markers and then fitting a plane to the markers' board, I was able to precisely determine the steering wheel's location and orientation with three degrees of freedom. This method overcame the limitations of the initial approach and provided high-quality ground truth data that could be used for training and evaluating 3D object detection models.

Each approach will be discussed in more detail in the following sections.

In this dataset, the ground truth for each frame includes an oriented bounding box that represents the precise 3D position and orientation of the steering wheel. Bounding boxes are encoded as $[x, y, z, l, w, h, rx, ry, rz]$, where (x, y, z) represents the center of the bounding box, (l, w, h) represents the dimensions of the box, (rx, ry, rz) represent the bounding box angle with the x, y, and z axes, respectively. Figure 3.3 illustrates the ground truth data and oriented bounding boxes for sample frames from the dataset, providing a comprehensive visual representation of the steering wheel's 3D position and orientation.

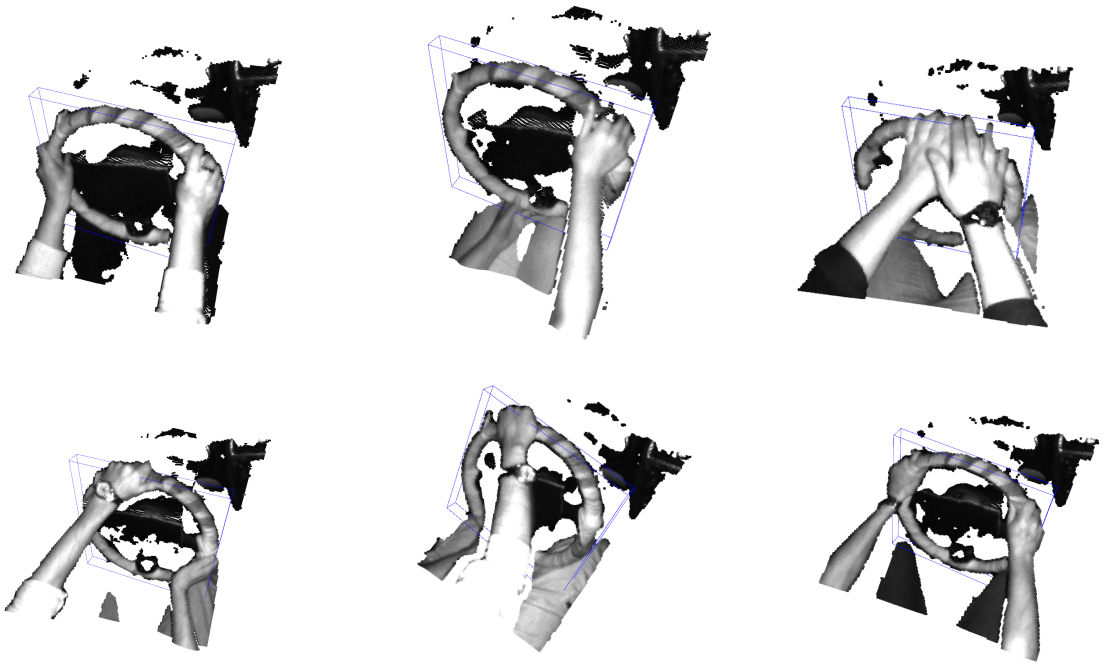


Figure 3.3: oriented bounding boxes for sample frames from the dataset

Given the ground truths, we can also represent the steering wheel as a torus shape, as depicted in fig. 3.4. The torus shape can provide a more accurate geometric representation of the steering wheel, capturing its circular cross-section and three-dimensional nature more effectively than a bounding box. By modeling the steering wheel as a torus, we can better account for its curvature and overall shape, which is essential for accurate 3D pose estimation and object detection tasks.

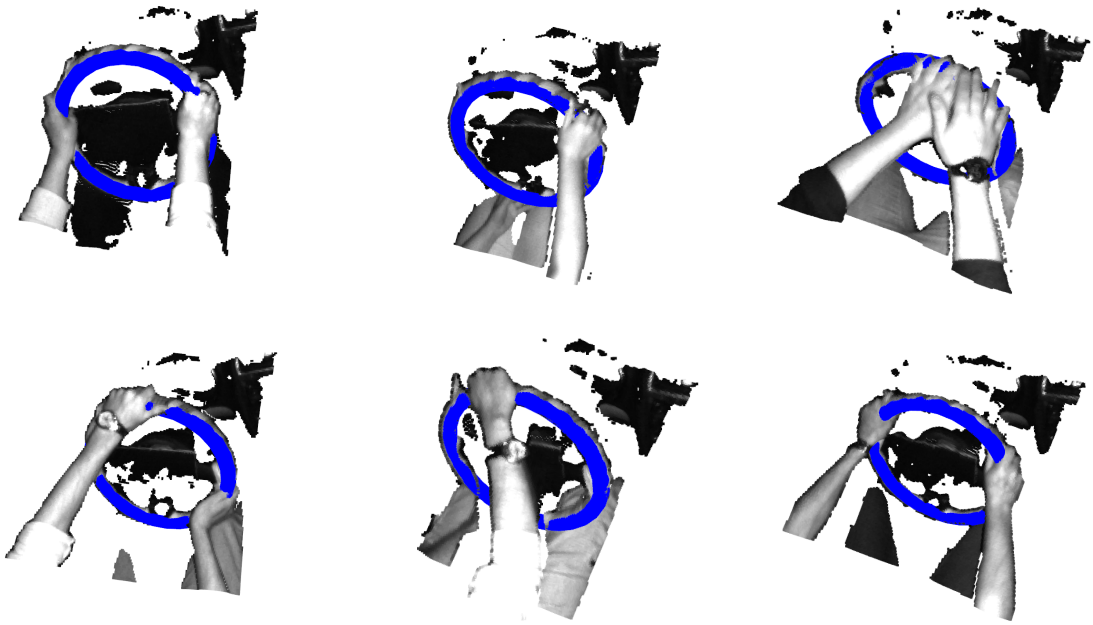


Figure 3.4: Torus representation of the steering wheel for sample frames from the dataset

4

Grand Truth Generation: Fitting a Circle (initial approach)

4.1 Introduction

The objective of this chapter is to present the initial approach developed for generating ground truth data to estimate the 3D location and orientation of a steering wheel using 2D and 3D geometric techniques. This approach sought to capture the steering wheel's geometric structure by fitting a 2D ellipse to annotated points and extending this representation into 3D space through additional processing steps.

Given the complexity of manually annotating the steering wheel across numerous images and the inherent sparsity and noise in the data, we aimed to automate and enhance the annotation process by fitting a mathematical model to a few number of spars annotations. This model would allow us to generate a more comprehensive representation of the steering wheel geometry.

The approach was divided into several phases:

1. Annotating the steering wheel with 2D points.
2. Fitting an ellipse to the annotated 2D points.
3. Sampling additional points from the ellipse and mapping them into 3D space.
4. Fitting a sphere to the 3D points to refine the geometry.
5. Deriving a circular cross-section from the sphere to better represent the steering wheel.

Although this initial approach showed some promise, it was constrained by the sparse and noisy nature of the data, resulting in a lack of precision in the final 3D representation of the steering wheel. Furthermore, there was no systematic method to effectively evaluate the ground truth data accuracy generated by this approach.

This chapter details the methodology, results, and the reasons for transitioning to a more reliable approach.

4.2 methodology

Annotating the Steering Wheel with 2D Points

The process began with manual annotation of the steering wheel in 2D images. Annotating a dense and accurate set of points across all images was infeasible due to the time-consuming nature of manual annotation and the large number of frames. Hence, a sparse set of 2D points was initially marked to provide a basic outline of the steering wheel as depicted in **??**.

Fitting an Ellipse to 2D Points

In order to create a denser representation, we decided to fit an ellipse to the existing 2D annotation points. This provided a more complete 2D shape that captured the true geometric structure of the steering wheel as illustrated in fig. 4.1a.

We fitted an ellipse to these points using the general equation of an ellipse:

$$c_0X^2 + c_1XY + c_2Y^2 + c_3X + c_4Y = b$$

To find the coefficients $[c_0, c_1, c_2, c_3, c_4]$ that best fit the observed points (X, Y), we applied linear regression. Defining $A = [X^2, XY, Y^2, X, Y]$ and $c = [c_0, c_1, c_2, c_3, c_4]$, we calculated the coefficients by solving for c that satisfies the equation $Ac = b$, where b represents the constant in the ellipse equation.

Sampling Points from the Ellipse and Mapping to 3D

Once the ellipse was fitted, 1000 points were sampled from the ellipse and its neighboring regions to create a denser representation as shown in fig. 4.1b. These points were mapped from 2D to 3D using depth information from the camera and the intrinsic parameters.

However, due to the sparse and noisy nature of the data, the sampled points could not accurately represent the true geometry of the steering wheel, resulting in an inaccurate approximation of the 3D position and orientation. As depicted in fig. 4.1c, the bounding box fitted to the mapped 3d points was unable to accurately represent the position of the steering wheel.

Fitting a Sphere to the 3D Points

To further refine the 3D representation, we attempted to fit a sphere to the sampled points on the steering wheel as demonstrated in fig. 4.1d. Given the sample points (x, y, z), we used the following general equation for a sphere:

$$(x - h)^2 + (y - k)^2 + (z - m)^2 = r^2$$

Expanding this equation, we obtain:

$$x^2 + h^2 - 2xh + y^2 + k^2 - 2yk + z^2 + m^2 - 2mz = r^2$$

Rearranging terms yields:

$$2xh + 2yk + 2zm + (r^2 - h^2 - k^2 - m^2) = x^2 + y^2 + z^2$$

Letting $A = [2x, 2y, 2z, 1]$, $c = [h, k, m, r^2 - h^2 - k^2 - m^2]$, and $b = x^2 + y^2 + z^2$, we can fit a sphere to the points by solving for c in the equation:

$$Ac = b$$

This solution provides us with the center (h, k, m) and radius r of the sphere.

Refining with a Circular Cross-Section from the Sphere

To better capture the essential structure of the steering wheel, we took a circular cross-section of the fitted sphere in the XY plane by fixing $z = m$. This simplification was based on the observation that the steering wheel's geometry is primarily defined in the XY plane, with minimal variation along the z-axis. Principal Component Analysis (PCA) of the steering wheel points revealed that the distribution along the z-axis is very limited, indicating that this axis does not contain significant information about the overall shape or structure of the steering wheel. By constraining the representation to the XY plane, we avoided unnecessary complexity from minor variations along the z-axis, which could detract from the accuracy of the wheel's positioning.

This circular cross-section effectively represents the X and Y correlations in the steering wheel geometry, simplifying the model while preserving the essential spatial characteristics of the wheel.

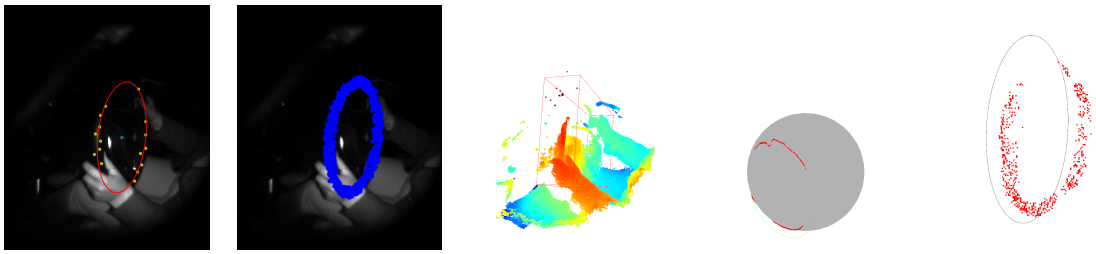
Aligning the Circle with a Rotation Matrix

Finally, we aligned this circle to the steering wheel's position by applying a rotation matrix followed by a translation to center of the sphere. The rotation matrix was derived from the oriented bounding box of the steering wheel 3D points illustrated in fig. 4.1c. Open3D OrientedBoundingBox method [2] was employed to compute the orientation and rotation matrix, which allowed us to align the circle with the steering wheel's orientation in 3D space. Figure fig. 4.1e shows the final circular representation of the steering wheel's 3D position.

4.3 Conclusion

Despite these steps, this approach was ultimately limited by the sparse and noisy data, as well as the lack of an accurate evaluation method for the generated ground truths. This made it difficult to ensure the precision of the ground truths in accurately capturing the true location and orientation of the steering wheel. These limitations prompted us to pursue a second approach using ArUco markers and plane fitting for more reliable results.

[2] available at http://www.open3d.org/docs/release/python_api/open3d.geometry.OrientedBoundingBox.html



(a) (b) (c) (d) (e)

Figure 4.1: (a) Steering wheel annotated with 2d points and an ellipse fitted to the 2D points. (b) 1,000 points were sampled from the ellipse and its small neighborhood.(c) The sampled points were mapped to 3D space, with the bounding box illustrating these points in 3D. (d) A sphere was fitted to the sampled points. (e) The 3D model was refined by deriving a circular cross-section from the sphere, representing the steering wheel's 3D position as a circle in 3D space.

5

Ground Truth Generaiton: ArUco Markers and Plane Fitting (Refined Approach)

5.1 Introduction

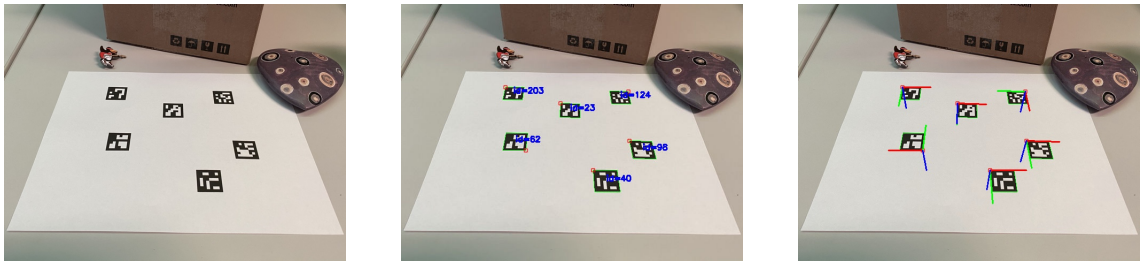
To overcome the limitations of the initial ellipse fitting approach, a second method was developed using ArUco markers and plane fitting. This method aimed to achieve precise 3D estimation of the steering wheel's location and orientation. The use of ArUco board, and a plane fitting technique allowed for a more robust and accurate approach to capturing the 3D geometry of the steering wheel. By leveraging the known properties and spatial relationships of the ArUco markers, this refined method was able to provide a reliable and high-quality ground truth for the steering wheel's position and orientation, which could then be used for training and evaluating 3D object detection models.

ArUco markers [7] are widely used in computer vision for precise localization and pose estimation tasks. An ArUco marker as depicted in fig. 5.1a, is a synthetic square marker featuring a thick black border and an inner binary matrix that encodes its identifier (ID). The black border enables quick detection in images, while the binary matrix allows for identification and supports error detection and correction. The marker size dictates the dimensions of the internal matrix; for example, a 4x4 marker contains 16 bits. can be easily detected and identified in images. They offer several advantages, including robustness to variations in lighting, simple detection with minimal computational overhead, and compatibility with popular libraries like OpenCV, which provides dedicated functions for detecting ArUco markers and estimating their pose in both 2D and 3D. Figures fig. 5.1b and fig. 5.1c demonstrate OpenCV's capabilities in detecting ArUco markers and estimating their 3D position, respectively.

This chapter presents the structured methodology, the evaluation of different techniques for location and orientation estimation, and the results of this second approach in accurately capturing the steering wheel's 3D geometry.

This refined approach aimed to address the challenges encountered in the initial method, particularly the issues of data sparsity, noise, and inaccurate geometric representation. By leveraging the advantages of ArUco markers, this method sought to tackle the difficulties in precisely estimating the 3D position and orientation of the steering wheel.

The proposed technique involved positioning a board with multiple ArUco markers at



(a) Single ArUco Markers

(b) These are the detected markers (in green). Note that some markers are rotated. The small red square indicates the marker's top left corner

(c) The marker coordinate system that is placed in the center or in the top left corner of the marker with the Z axis pointing out. Axis-color correspondences are X: red, Y: green, Z: blue.

Figure 5.1: ArUco merkers' detection and pose estimation by OpenCV.

the center of the steering wheel like in fig. 3.2a and employing plane fitting to enhance the accuracy of orientation estimation. This approach was extensively evaluated to validate its reliability in capturing the 3D geometry of the steering wheel.

The following sections detail the assessment of ArUco markers for location and orientation estimation, as well as the generation of the steering wheel's oriented bounding box using the validated methods.

5.2 Methodoloy

To address the limitations identified in the initial approach, a structured methodology was developed using ArUco markers and plane fitting. The ArUco markers were placed on a board mounted at the center of the steering wheel as depicted in fig. 3.2a. This setup allowed us to capture both the 2D and 3D spatial data required for accurately estimating the steering wheel's position and orientation.

The methodology involved two primary components for estimating the location and orientation of the steering wheel. each component is described in detail in the following subsections:

location estimation

As depicted in fig. 3.2a, the ArUco board is positioned at the center of the steering wheel. Therfore, the detected centroid of the ArUco board directly corresponds to the 3D spatial coordinates of the steering wheel's center. Given the known arrangement of markers on the ArUco board, the location of the board's center, and consequently the steering wheel's center, can be readily inferred from the detected positions of the individual markers.

To achieve precise location estimation of the ArUco markers, we considered two primary approaches:

1. Direct 3D Estimation Using OpenCV: OpenCV's *estimatePoseSingleMarkers* function provides a direct way to estimate the 3D position of markers relative to the camera. This function relies on the known physical size of the ArUco markers and the intrinsic camera parameters to calculate the translation vector, which represents the 3D position of the markers as [dx, dy, dz] where dx dy, and dz are the marker's position along the x, y, and z axes, respectively. While this method offers a straightforward approach, its accuracy can be sensitive to marker orientation, distance from the camera, and the angle of view.

2. 2D Estimation with Mapping to 3D: This alternative method starts by detecting the 2D position of the markers in the image plane using OpenCV's *detectMarkers* function. Figure 5.2 shows merkers' location detected by OpenCV on the 2D image. After detecting the 2D positions, these points are mapped to 3D space using depth information derived from the point cloud data and the camera's intrinsic parameters. This approach separates the detection process into two stages, potentially improving accuracy by isolating the detection of 2D marker positions from their 3D mapping.

Figure 5.2: OpenCV detected the ArUco markers' location in the 2d image.

Given the known structure and location of the markers on the board, we employed a method to calculate the board's center based on the spatial relationship of specific marker pairs. This technique ensured that the center could be reliably calculated even if some markers were undetected.

To calculate the center of the board, we utilized three pairs of markers that are symmetrically positioned on opposite sides of the board as depicted in fig. 5.3. These marker pairs are: {(0, 5), (1, 4), (2, 3)}

For each detected pair, the midpoint between the two markers was computed by averaging their translation vectors *tvec*, which represent the 3D coordinates of each marker. This midpoint provides an estimate of the board's center based on that particular pair.

For each detected pair (i, j), we compute the midpoint as:

$$\text{midpoint}_{ij} = \frac{tvec_i + tvec_j}{2}$$

where $tvec_i$ and $tvec_j$ are the translation vectors (3D coordinates) of markers i and j. Once the midpoints for all detected pairs are obtained, the overall center of the board is calculated as the mean of these midpoints:

$$\text{center}_{\text{board}} = \frac{\sum (i,j) \in \text{detected pairs} \text{midpoint}_{ij}}{N}$$

where N is the number of detected pairs. The resulting averaged translation vector represents the estimated 3D center of the board.

By considering the relative positions and midpoints of detected marker pairs, this method can robustly determine the center point of the board, providing an accurate 3D location for the steering wheel despite potential occlusions or missing markers.

Evaluating ArUco Markers for Orientation Estimation

Accurately determining the orientation of the marker board was critical for estimating the steering wheel's orientation. As the ArUco board is accurately placed tangent to the steering wheel, the orientation of the board corresponds to the orientation of the steering wheel in 3D space.

Two methods were used exploited to estimate the orientation of the ArUco board:

Direct Orientation Estimation Using OpenCV: OpenCV's *estimatePoseSingleMarkers* function also calculates the orientation of markers in the form of rotation vectors. These vectors indicate the rotational pose of each marker relative to the camera's coordinate system. Assuming the we know the orientation of the markers on the board, we can remove the outlier markers and average the remaining rotation vectors to estimate the overall orientation of the board. However, this method has limitations, as it relies on the accuracy of OpenCV's built-in pose estimation.

As illustrated in fig. 5.4, OpenCV's estimations of marker positions on a 3×4 board are highly sensitive to several factors. These include the camera angle, the distance of the markers from the camera, occlusion of the markers, and other environmental factors. Consequently, when markers are missing or occluded, the reliability and accuracy of the orientation estimation can decrease significantly. This sensitivity to various spatial and environmental conditions highlights the limitations of relying solely on OpenCV's built-in pose estimation functions for robust 3D orientation determination.

Plane Fitting Method: The plane fitting method provided a more robust and reliable means of estimating the orientation, as it doesn't depend solely on the accuracy of OpenCV's pose estimation for individual markers. This approach overcomes the limitations of the direct orientation estimation using OpenCV, which can be sensitive to factors like marker size, distance, and viewing angle. Hence, the plane fitting method offers a more stable and accurate estimate of the steering wheel's orientation under a variety of conditions.

This method involves the following steps:

1. *Finding the Center of the ArUco Board:* To initiate plane fitting, it is essential to first locate the center of the ArUco board. This is done using the detected marker locations.

Given that the markers are arranged in a known, fixed structure on the board, the center can be reliably estimated even if one or more markers are not detected. By analyzing the relative positions of the detected markers, the center of the board can be computed with high accuracy.

2. *Sampling Points Around the Center:* Once the center of the ArUco board is determined, a set of points is sampled from the board around this central point within a specified radius. This sampling process ensures that the points used for plane fitting are well-distributed across the board, enhancing the reliability of the orientation estimation. Figure 5.5 shows the sampled points from the board in red.
3. *Fitting a Plane to the Sampled Points:* The sampled points are then used to fit a plane. The fitting process involves computing the best-fit plane that minimizes the distance between the plane and the sampled points. This plane effectively represents the overall orientation of the ArUco board.
4. *Using the Normal Vector for Orientation:* The normal vector of the fitted plane is then calculated. This normal vector serves as a robust indicator of the board's orientation in 3D space. As fig. 5.5 shows, the light blue line indicated the norm vector of the board. Since the board is mounted on the steering wheel tangent to the steering wheel's surface, the orientation of this normal vector directly corresponds to the orientation of the steering wheel.

One of the significant advantages of this method is its robustness to marker detection failures. Even if some markers are not detected due to occlusions, lighting conditions, or other factors, the known arrangement of the markers allows for accurate estimation of the board's center and subsequent orientation. This makes the plane fitting method highly resilient to errors in individual marker detections, ensuring reliable orientation estimation under a variety of conditions.

Generating the Steering Wheel's Oriented Bounding Box

5.3 Results

Location Estimation Results

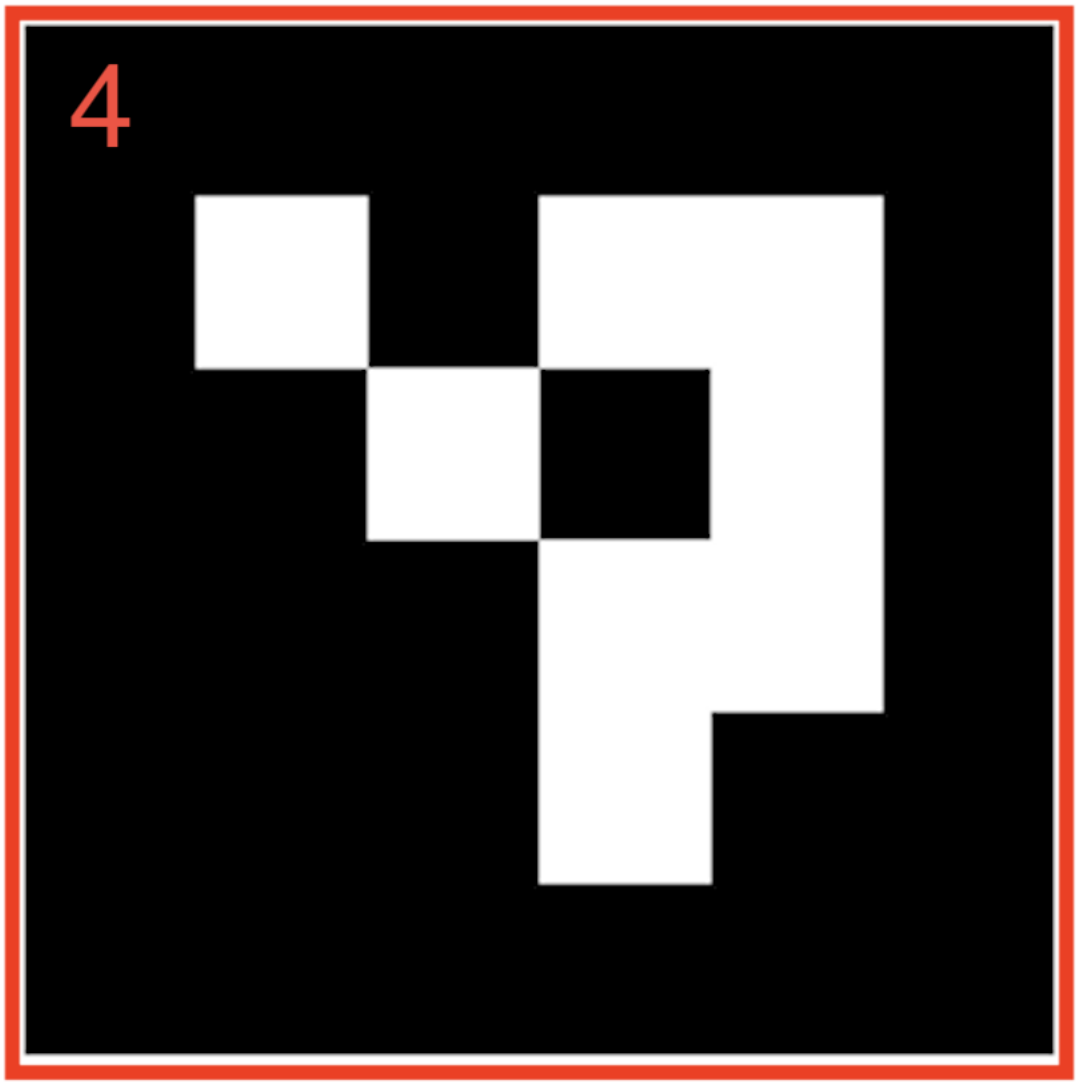
Inter-Marker Distance Evaluation Distance and Angle Variation

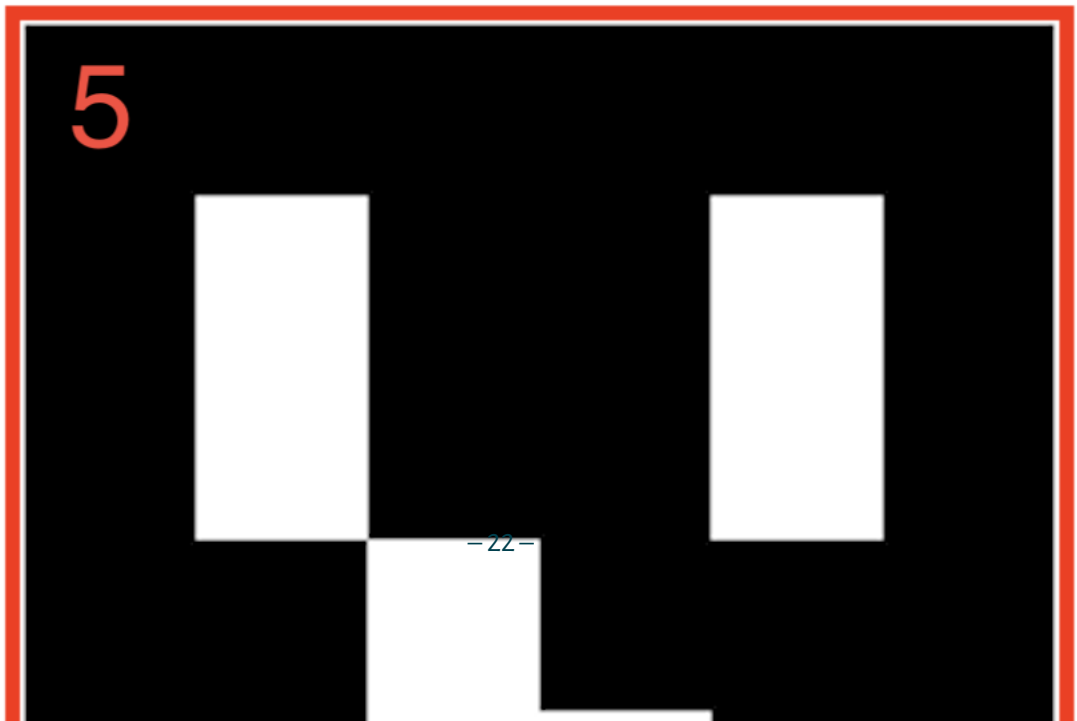
Orientation Estimation Results

Systematic Error in Orientation Estimation

Generating the Steering Wheel's Oriented Bounding Box

5.4 Conclusion


4


5

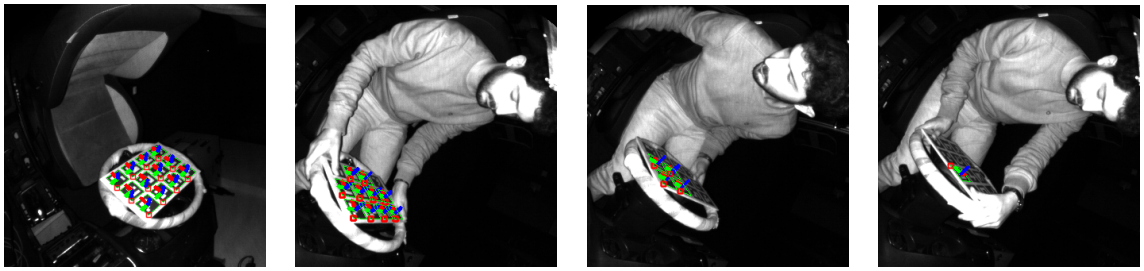


Figure 5.4: The markers' coordinate system estimated by OpenCV's *estimatePoseSingleMarkers* function on a 3x4 board shows that this method is highly sensitive to the markers' distance, camera angle and other environmental factors. Axis-color correspondences are X: red, Y: green, Z: blue, with Z axis pointing out

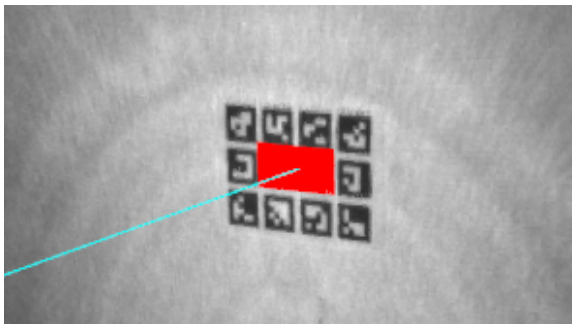


Figure 5.5: Illustration of the plane fitting method, showing the sampled points around the center of the ArUco board in red and the normal vector of the fitted plane in light blue, which represents the orientation of the steering wheel.

6

Model and Methodology

7

discussion

8

conclusion

Bibliography

[1] Caesar, H., Bankiti, V., Lang, A., Vora, S., Liong, V. E., Xu, Q., Anush, K., Yu, P., Baldan, G., and Beijbom, O. *nuScenes: A multimodal dataset for autonomous driving*. Jan. 2019. DOI: 10.48550/arxiv.1903.11027. URL: https://arxiv.org/abs/1903.11027.

[2] Chen, Y., Liu, S., Shen, X., and Jia, J. DSGN: Deep Stereo Geometry Network for 3D Object Detection. In: June 2020. DOI: 10.1109/cvpr42600.2020.01255. URL: https://doi.org/10.1109/cvpr42600.2020.01255.

[3] Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., and Li, H. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In: *Association for the Advancement of Artificial Intelligence* 35(2):1201–1209, May 2021. DOI: 10.1609/aaai.v35i2.16207. URL: https://doi.org/10.1609/aaai.v35i2.16207.

[4] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets Robotics: The KITTI Dataset. In: *International Journal of Robotics Research (IJRR)*, 2013.

[5] Ku, J. S., Mozifian, M., Lee, J., Harakeh, A., and Waslander, S. L. *Joint 3D Proposal Generation and Object Detection from View Aggregation*. Oct. 2018. DOI: 10.1109/iros.2018.8594049. URL: https://doi.org/10.1109/iros.2018.8594049.

[6] Li, P., Chen, X., and Shen, S. *Stereo R-CNN Based 3D Object Detection for Autonomous Driving*. June 2019. DOI: 10.1109/cvpr.2019.00783. URL: https://doi.org/10.1109/cvpr.2019.00783.

[7] OpenCV *ArUco Marker Detection*. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html. Accessed: 2024-11-04. 2024.

[8] Qi, C. R., Li, Y., Su, H., and Guibas, L. *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. Jan. 2017. DOI: 10.48550/arxiv.1706.02413. URL: https://arxiv.org/abs/1706.02413.

[9] Qi, C. R., Liu, W., Wu, C., Su, H., and Guibas, L. J. Frustum PointNets for 3D Object Detection from RGB-D Data. In: *CoRR* abs/1711.08488, 2017. arXiv: 1711.08488. URL: http://arxiv.org/abs/1711.08488.

[10] Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, W., and Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In: June 2020. DOI: 10.1109/cvpr42600.2020.01054. URL: https://doi.org/10.1109/cvpr42600.2020.01054.

[11] Shi, S., Wang, W., and Li, H. *PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud*. Jan. 2018. DOI: 10.48550/arxiv.1812.04244. URL: https://arxiv.org/abs/1812.04244.

[12] Shi, W., Ragunathan, and Rajkumar *Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud*. Jan. 2020. DOI: 10.48550/arxiv.2003.01251. URL: https://arxiv.org/abs/2003.01251.

[13] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J. C. Y., Zhou, Y., Chai, Y., Caine, B., et al. *Scalability in Perception for Autonomous Driving: Waymo Open Dataset*. June 2020. DOI: 10.1109/cvpr42600.2020.00252. URL: https://doi.org/10.1109/cvpr42600.2020.00252.

[14] undefined, C. X. M. H. W. J. L. B. X. T. *Multi-View 3D Object Detection Network for Autonomous Driving*. Nov. 2016. URL: https://arxiv.org/abs/1611.07759v2.

[15] Vora, S., Lang, A., Helou, B., and Beijbom, O. PointPainting: Sequential Fusion for 3D Object Detection. In: June 2020. DOI: 10.1109/cvpr42600.2020.00466. URL: https://doi.org/10.1109/cvpr42600.2020.00466.

[16] Wang, Y., Chao, W., Garg, D., Hariharan, B., Campbell, M., and Weinberger, K. Q. *Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving*. June 2019. DOI: 10.1109/cvpr.2019.00864. URL: https://doi.org/10.1109/cvpr.2019.00864.

[17] Wu, Z., Tang, W., Jiang, L., and Fu, C. SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud. In: June 2021. DOI: 10.1109/cvpr46437.2021.01426. URL: https://doi.org/10.1109/cvpr46437.2021.01426.

[18] Xu, Q., Zhong, Y., and Neumann, U. Behind the Curtain: Learning Occluded Shapes for 3D Object Detection. In: *Association for the Advancement of Artificial Intelligence* 36(3):2893–2901, June 2022. DOI: 10.1609/aaai.v36i3.20194. URL: https://doi.org/10.1609/aaai.v36i3.20194.

[19] You, Y., Wang, Y., Chao, W., Garg, D., Pleiss, G., Hariharan, B., Campbell, M., and Weinberger, K. Q. *Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving*. Jan. 2019. DOI: 10.48550/arxiv.1906.06310. URL: https://arxiv.org/abs/1906.06310.