

Stereo-Camera Calibration Auto-Tuning for Simultaneous Localization and Mapping

Francisco Durão Leitão Enguita

Instituto Superior Técnico / University of Lisbon, Portugal

fenguita.jr@gmail.com, francisco.enguita@tecnico.ulisboa.pt

Abstract

For an autonomous vehicle to correctly navigate the environment, it must have a clear understanding of the surroundings and participating agents, such as other vehicles or pedestrians. It also has to be able to identify its pose and act upon it, if necessary. The main goal of this work is to implement the navigation segment, analyze the behavior of an autonomous vehicle when the cameras are not correctly calibrated and propose an online auto-calibration method to improve navigation. To do so, this work researches into state of the art (SOTA) sensor models, VSLAM algorithms as well as a thorough analysis on the mounting options of the stereo rig and the individual consequences of said parameters. To finalize, a structure from motion (SfM) auto-calibration routine is tested and the resulting intrinsics are given to the VSLAM algorithm to perform pose tracking in a real-time simulation environment.

I. Introduction

Autonomous vehicles are vehicles that are capable of navigating and operating without human intervention. These vehicles use a variety of sensors and technologies, such as cameras, LiDAR, RADAR, and GPS, to perceive their surroundings and make decisions on how to navigate it. This technology while still in its early days, has the potential to revolutionize the way we travel, as it could significantly reduce the need for human drivers and potentially make transportation safer, more efficient and convenient.

To make this ideal into fruition, extensive investment and work is being conducted by the likes of private operators such as Waymo or Cruise, and research institutions.

VIENA (Vehículo Inteligente Elétrico de Navegação Autónoma) is a project from IST university whose goal is to provide a pedagogical platform for students and researchers, to perform state-of-the-art research on electric systems and autonomous vehicles.

A. Related Work

Autonomous agents are considered perceptive, if they acquire information of the environment through sensors, to then extract relevant knowledge [24]. Simultaneous Localization and Mapping (SLAM) algorithms are commonly used to both map the environment and relevant actors

as well as inferring the agent’s pose. These imply vision systems and other sensors, such as Inertial Measurement Units (IMU), to accurately model the agent’s surroundings while maintaining adequate computational cost. Older trends, such as graph based SLAM [16] have solidified themselves while new deep learning capabilities based on Neural Radiance Fields (NeRF) such as iMAP [31] have increasingly become popular for SLAM tasks, due to their ability to not need known camera parameters [34].

However traditional visual SLAM (vSLAM) heavily relies on prior knowledge of camera intrinsics. Camera calibration leaped forward with planar pattern based calibration [37], allowing inexpensive cameras to be easily calibrated given a known object. This method was further refined by many other researchers, such as [7]. In parallel, a wave of Bundle Adjustment (BA) based calibration methods started emerging like [11] and [6], albeit with less force than pattern based ones due to robustness issues.

Before running these navigation algorithms on the real vehicle, it is desirable to test them in a safe and repeatable form [15] such as a simulator. Private operators usually build their own proprietary simulators or purchase a license for an established simulator such as CarSim. Open-source simulators such as LGSVL [27], that fully renders the environment from scratch or VISTA [2], that uses frames from RGB camera datasets to model the other sensors behaviour, have become increasingly popular in the research community due to their transparency and modularity. While CarSim has very complex vehicle dynamics simulation capabilities at the cost of simplistic sensor and graphical rendering, open-source simulators such as CARLA [8] typically target reinforcement learning applications, meaning sensors and image processing capabilities are detailed and accurate, while vehicle dynamics capabilities are basic.

B. Problem Formulation

The main objective proposed for this thesis is to analyze the behaviour of an Autonomous Ground Vehicle (AGV) when the vision system is poorly calibrated and propose a self-refinement method for the camera parameters that does not require special markers or surfaces. Pose tracking is deemed successful when the estimated trajectory of the vehicle closely resembles the absolute trajectory. However it is not clear how the miscalibration of specific parameters

affect pose tracking, specifically with added artifacts such as heavy rain or poor visibility. Using a state of the art simulator, we are able to replicate the characteristics of the stereo system as well as vehicle dynamics, while allowing for safe and repeatable testing. Another advantage of the simulator is that it allows for quick modification of the stereo camera position, without needing to intervene in the actual car. A structure from motion (SfM) stereo camera rig auto-calibration method is proposed as well as techniques and constraints for more accurate camera intrinsics estimation. This work complements on previous work in the VIENA project, as it intends to be the foundation for the navigation and real-time simulation segments. The following key objectives were tackled in this work.

- 1) Provide the VIENA system with an accurate simulation environment
- 2) Build the foundation of the VIENA navigation segment with a SLAM algorithm
- 3) Analyze the SLAM algorithm pose tracking performance when the stereo cameras are miscalibrated
- 4) Devise a method for camera auto-calibration and integrate into the system
- 5) Determine the best stereo rig mounting option that accounts for both tracking and auto-calibration robustness

II. Background and State of The Art

Advanced Driver Assistance Systems (ADAS) have seen a increase of applications in the automotive sector, as it aims to reduce accidents, improve energy efficiency and comfort for passengers. Virtual testing has also become an essential part of ADAS pipeline, since it would be challenging to tackle all the scenarios and environmental conditions of the real world. ADAS Vehicle tasks can be grouped into: sense, plan and act. In this chapter state of the art of sensor models will be covered, which form the basis of the sense group.

A. Navigation Sensors

Sensors are a key component of the ADAS pipeline, as they are responsible for perceiving the environment and agents. Our research will focus on cameras and inertial measurement units (IMU).

1) Camera Projection Model: The most common and simple model is the pin-hole camera [37]

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

where f_i are the focal lengths for the x-axis and y-axis, c_i the coordinates of the intersection of the sensor plane with the optical axis, $[x \ y \ z]$ is the 3d Cartesian space in camera coordinate system (since z points forward) and $[u,v]$ is the camera projection space.

2) Stereo Camera, Depth Camera: Stereo systems are constituted by two cameras, camera L known as reference camera and camera R known as target camera. Assuming a rectified system and equal cameras, vertical coordinates will be equal across both cameras [10]. Coordinates of point P in Cartesian system with respect to L and R camera reference system are described in

$$\begin{aligned} P_r &= z_l \hat{q}_l \\ P_l &= z_r \hat{q}_r \end{aligned} \quad (2)$$

where \hat{q}_r and \hat{q}_l represent the normalized image coordinates $\hat{q} = [u' \ v' \ 1]^T$ (inverse camera intrinsics matrix multiplied by the respective image coordinate vector $q = [u \ v \ 1]^T$) The point $P_r = [x_r \ y_r \ z_r]^T$ will be represented in reference camera L coordinates via rigid transformation.

$$P_r = RP_l + t \quad (3)$$

By combination of [2] and [3] we end up with a system of equations [4].

$$\begin{aligned} z_r u'_r - z_l r_1^T \hat{q}_l &= t_1 \\ z_r v'_r - z_l r_2^T \hat{q}_l &= t_2 \\ z_r - z_l r_3^T \hat{q}_l &= t_3 \end{aligned} \quad (4)$$

By solving equation system for z_l we get the distance from point P to reference camera L [36].

$$z_l = \frac{t_1 - u'_r t_3}{u'_r r_3^T \hat{q}_l - r_1^T \hat{q}_l} \quad (5)$$

3) Inertial Measurement Unit, IMU: the most common technology used for semi-conductor accelerometers is MEMS (Micro Electro Mechanical System).

MEMS accelerometers are comprised of three main components: anchors, fixed electrode and moving electrode. The anchors restrain the moving electrode from shifting along an undesired axis. As an acceleration is applied, the moving electrode shifts opposite to the sum of forces being applied in that axis. This movement causes the capacitance between the moving and fixed electrode to increase or decrease. By measuring the differential capacitance between the two electrodes the magnitude and direction of the acceleration can be obtained.

MEMS gyroscope measure angular acceleration by means of the Coriolis acceleration. As the resonating mass moves and the body it is attached to rotates, the mass and frame perceives Coriolis acceleration and the displacement is perpendicular to the oscillating mass direction. As the rotation rate increases, the frame is displaced more, diminishing the distance between the moving and fixed frame plates. This causes a variation in capacitance between the two electrodes, where the magnitude and direction of acceleration can be obtained.

B. Camera Calibration

Incremental camera calibration is commonly used for robotics applications due to its widespread availability and accurate results. A brief explanation on the algorithm's key points will be done.

1) Measurement collection: A new measurement batch is collected D_{new} given the motion model h , observation model g and their subsequent state ξ_k and measurement ζ_k

$$\xi_k = h(x_{k-1}, \varrho, \mathcal{N}(0, W_k)) \quad (6)$$

$$\zeta_k = g(x_k, l_k, \psi, \mathcal{N}(0, N_k)) \quad (7)$$

where W_k and N_k denote covariances of white Gaussian noise model \mathcal{N} , ϱ is the control input, ψ are the intrinsic parameters of the camera and l_k is the landmark.

2) Optimization: Optimization is computed using the following expression with Gauss-Newton method with TQR updates

$$\hat{\mu}_{\psi, X, L} = \arg \max_{\psi, X, L} p(\psi, X, L | D_{info}, D_{new}) \quad (8)$$

where X is the array composed of the actor states up to timestep K , U is the array measuring control inputs and L is an array with the landmark positions.

3) Measurement selection (Mutual information):

This is done by quantifying the reduction of uncertainty of ψ given measurements D_{new} or in other words, how much information D_{new} conveys to current estimate $\psi | D_{info}$, where D_{info} is the array of selected measurements. If new measurements D_{new} reduce the uncertainty above a user-defined threshold λ , they should influence the estimated camera calibration parameters ψ .

C. Visual SLAM and Structure from Motion

The concept of SLAM (Simultaneous Localization and Mapping) revolves about the ability of the agent to estimate the map of the environment it resides, while at the same time, monitoring its pose. The process of using vision sensors to perform SLAM is effectively called VSLAM [32].

Every VSLAM algorithm is composed by 4 blocks: input search, where sensor measurements are collected and features are extracted, pose tracking, where current camera pose is determined, mapping, where landmarks global pose are stored and loop closing, where drift-free localization is ensured.

Structure from motion (SfM) is the process of estimating and reconstructing a 3D structure from 2D images. In the past decade there have been great improvements to the performance and accuracy of these reconstruction algorithms, especially in the incremental SfM and most recently deep SfM fields [17] [35]. The main advantage of photogrammetry SfM algorithms is their robustness to scene conditions and accuracy even while in movement. These algorithms are designed for accurate environment reconstructions, which require a large amount of images of the same scene causing a significant bottleneck in the large minimization function of the bundle adjustment (BA) step.

D. State-of-the-Art Simulation Environment

The end goal is to achieve a simulation pipeline that allows accurate representation of real-world dynamics and scenarios, fits tightly with the already built MATLAB/SIMULINK system and is transparent, allowing modifications to the vehicle dynamics or sensor models to be done if necessary.

CARLA (Car Learning to Act) [8] is an open-source simulator for testing and validating autonomous vehicles driving models. The primary focus of this simulator is to provide a training ground for reinforcement learning algorithms that tackles urban driving complexities, such as other vehicles, pedestrians, signals and environment effects.

a) CARLA camera model: the CARLA simulator uses the default Unreal Engine implementation, enabling effects such as bloom, vignette, lens flares, depth of field and grain jitter.

b) CARLA Depth camera model: it uses the default RGB camera model and codifies distance using the 3 color channels (from least to more significant bytes, R-G-B) to each pixel to the camera. The final result is an idealized ray-casted depth map, to which distortion effect can be added.

c) CARLA IMU model: gives orientation, angular velocity and linear acceleration of the actor. It has support for adding Gaussian noise for the accelerometer and gyroscope independently. This is commonly known as the simple IMU sensor model [9] and given by equations [9] and [10] for the gyroscope and accelerometer errors respectively.

$$\delta w = B_g + A_g w + \epsilon \quad (9)$$

where B_g is the vector comprised of the bias terms along the 3 axis, w is the vector with the angular velocity terms, ϵ is the Gaussian noise vector and A_g is the matrix comprised of the gyroscope scale factor errors S_{gi} and the gyroscope misalignment errors M_{gi} . Since there is no mention or possibility to modify the terms of the A_g matrix in the default gyroscope model in CARLA, A will be the identity matrix I .

$$\delta f = B_a + A_a f + \epsilon + D_a f^2 \quad (10)$$

Similarly with the gyroscope error model, B_a is the accelerometer bias vector, A_a is the matrix with the scale factor errors S_{ai} and misalignment errors M_{ai} of the accelerometer. The f vector represents the accelerometer specific force and the D_a matrix is the quadratic term acceleration error.

The CARLA simulator was ultimately chosen for its good flexibility, free access and tight ROS integration, which will be important for the communication with the VIENA Controller and Vehicle Model (built in MATLAB/SIMULINK).

III. Navigation System Testbed

For an agent to be autonomous and capable of navigating, it must first comprehend its own state as well

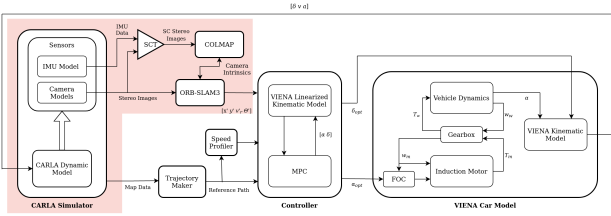


Fig. 1: Complete system architecture

as the state of other agents or objects in the environment. The navigation segment is commonly divided into two main sub-segments: perception and planning stages.

This work will focus on the perception stage, where the sensors and algorithms for the agent to understand its state and surroundings are held. Planning, as defined by Paden [23], is composed by 4 sequential layer: route planning, where a sequence of waypoints are extrapolated from road network data, the behavioral layer, where the type of motion is specified based on other perceived agents/objects, motion planning, where a reference path is created and the local feedback control where given the estimate of the vehicle state, steering and throttle commands are applied.

A. VIENA System Overview

A brief introduction into the components of the Veículo Inteligente Elettrico de Navegação Autônoma (VIENA) system will be done as well as explaining where this work contributes to the project.

1) Setup: project VIENA is based on a converted 1998 Fiat Seicento Elettra. The navigation segment will collect information from the environment through the ZED 2i Stereo Camera and be operated by a NVIDIA Jetson Xavier NX system on module (SOM). The ZED 2i camera uses the stereo working principle (no illuminator) with an AI model to do the matching and infer depth.

2) System Model: The complete VIENA model architecture is presented in Figure 1 and it expands on [25], with the CARLA Simulator, COLMAP, ORB-SLAM3 and SCT blocks, highlighted in red.

The CARLA Simulator is an Unreal Engine based graphical platform for autonomous vehicle reinforcement learning methodologies. It holds the Dynamic vehicle model whose parameters were filled with the real VIENA car parameters so that it matches as close as possible.

The SCT stands for Slow Corner Trigger and gets the input of IMU data and stereo images. When the vehicle turns enough and the velocity is sufficiently small, the stereo frames get sent to the COLMAP block.

Taking the camera intrinsics from the ORB-SLAM3 definition file, COLMAP will refine the camera parameters and return those to the ORB-SLAM3 algorithm, so that the resulting pose estimation and mapping is more accurate. Note that this will only be done offline, as the implementation of ORB-SLAM3 does not allow for the dynamic modification of the camera intrinsics, since these are tied to the initialization of the SLAM problem.

B. Communication Architecture

The main objective in the design of the communication architecture is to ensure modularity and easy transition between simulated data and real-world data. There are three main components in the architecture: Matlab, Robot Operating System (ROS) and CARLA. The main components are connected through two wrappers, MATLAB-ROS and CARLA ROS-bridge. Wrappers translate data, internal functions and commands into a ROS architecture.

Communication is handled through ROS [26], due to its open-source nature, allowing third party integration possible and widespread usage in the robotics research community.

The CARLA ROS-bridge is a wrapper that is provided by the CARLA Simulator. This could be anything from requesting the map waypoints that form the possible trajectories to inputting throttle and steering commands.

The MATLAB-ROS is a wrapper that was created for this work. The intent is the same as the CARLA ROS-bridge, but for connecting the MATLAB based VIENA control system implementation to the ROS network through the "matlab" node.

With this integration, if for example we wished to use the real sensors and actuators, we would only need to create a new ROS node capable of commanding the sensors. For the perception system this is already done since ZED provides ROS implementation for the ZED 2i camera.

C. Simulation Environment

The main advantage of using a simulated/virtual setup is that testing can be done throughout different sensor mountings without needing to integrate into the package, while also enabling easier behaviour evaluation of scenarios that could compromise the system and endanger the environment, such as mounting or calibration errors and adverse weather conditions.

IV. Navigation Setting-up and Running

In this chapter, the technical details of the aforementioned setup are explained, as well as the required steps to run the proposed testbed. As seen in Figure 2, we will first focus on the "traditional" stereo rig camera calibration (Kalibr), explain the functioning of Structure from Motion (SfM) based camera calibration (COLMAP), then describe the components of the implemented VSLAM algorithm (ORB-SLAM3) and to finalize, our method to extract stereo frames fulfilling certain criteria that yield a more successful self-calibration (SCT).

A. Camera Calibration

The typical offline incremental camera calibration approach is described as in section II-B. Kalibr's stereo camera [19] calibration necessary inputs and generated outputs are detailed in the following paragraphs.

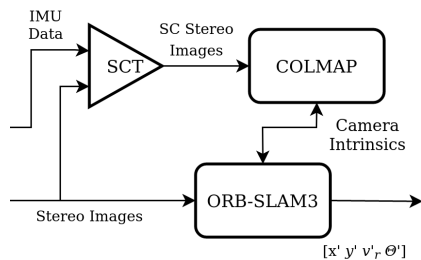


Fig. 2: Navigation components

t_1, t_2 are the observations (images) from camera 1 and 2, which are extracted from the rosbag by specifying the publishing topics. Kalibr is a multi-camera calibration system, meaning we could calibrate N cameras simultaneously, by inputting N image topics. T_g is the side length of an April tag that is provided by Kalibr. Since the size of the tags depends on the scale the target is printed on, the user must correctly measure these. M_{t1}, M_{t2} specifies the type of camera model (pin-hole, omnidirectional, double sphere or extended unified), D_{t1}, D_{t2} specifies the type of distortion model (radial-tangential, equidistant, fov or none) and f is the observation frequency. The resulting outputs are the camera intrinsics $\psi_1, \psi_2 \in \mathbb{R}^{3 \times 3}$, the distortion values $D_1, D_2 \in \mathbb{R}^{1 \times 4}$ and the relative pose between camera 1 and camera 2, $[R, t] \in \mathbb{R}^{3 \times 4}$.

This method will serve as a baseline for the camera calibration experiments and comparisons that will be done in section V. However, for our use case, we desire a calibration method that can be done online and does not require special surfaces or targets. The next section will explain the building blocks of our auto-calibration routine.

B. Calibration based on Structure from Motion

There are two main stages in the incremental SfM pipeline, search and reconstruction [30].

1) Search: The search stage goal is to find feature pairs between input images I , so that an overlap geometry can be found.

a) Feature Extraction and Matching: In this sub stage, identifiable points of interest are extracted from the frames and then matched. Common feature extractors used are SIFT [18], ORB [28] and SURF [3]. Due to the fragility of the initialization sub stage, robustness is preferred over performance [14]. There are several matching methods that are significant improvements over the basic strategy of exhaustive matching (where every feature of frame 1 N_1 is compared against every feature of frame 2 N_2), such as the Bounded Alignment and the Branch-and-Bound algorithm [20].

b) Geometric verification: A transformation between frame 1 and 2 is said to be geometrically verified, if there are sufficient matches that share the same projection transformation. Mismatches are prevalent, so common robust outlier removal techniques such as RANSAC [9] are needed.

2) Reconstruction:

a) Initialization: Initialization is a critical step, as algorithm might not converge if given a bad image pair [29]. In photogrammetry robustness is often preferred over performance, so denser parts of the scene graph are chosen for the initial image pair. This results in a more expensive bundle adjustment operation.

b) Global refinement: Also known as Bundle Adjustment [33], consists in the minimization of the total geometric error by refining point parameters $\{P_j \in \mathbb{R}^3\}$, camera parameters $\{\psi_i\}$ and image poses $\{(R_i, t_i) \in SE(3)\}$ in rotation matrix form [17].

$$E_{BA} = \sum_j \sum_{(i,u)} \|\Pi(R_i P_j + t_i, \psi_i) - p_u\|_2 \quad (11)$$

C. Visual SLAM

ORB-SLAM3 [4] builds upon the existing strong foundation of ORB-SLAM-VI [21] and ORB-SLAM2 [22] with two main contributions: an improved visual-inertial system based on maximum a posteriori (MAP) estimation and a multi-map system capable of recalling if tracking is lost. A brief introduction into the ORB-SLAM3 stereo SLAM will be done.

1) Feature extracting, stereo keypoints and camera pose optimization: Stereo SLAM starts with the extraction of ORB features in both left and right frames assuming we are working with a rectified stereo pair and epipolar geometry. Camera pose gets optimized using a bundle adjustment method that minimizes the reprojection error as seen below.

$$\{R, t\} = \arg \min_{R,t} \sum_{i \in X} p \|\Pi_s(RX^i + t) - x_s^i\|_\Sigma^2 \quad (12)$$

where R is the camera rotation, t is the camera position, X^i are the keypoints position, Σ is the scale covariance of the keypoint, p is the Huber loss function [13] and Π_s is the rectified stereo projection.

2) Map recalling system: Two main improvements are done in the form of the map recalling system over ORBSLAM2: a new place recognition algorithm running the same DBow2 database [12] now with geometric verification for better precision and the definition of a local window in the covisibility graph for further relative pose refinement.

D. Conditions-Based Self-Calibration, Slow Corner Trigger

Agapito's work [1] demonstrated that camera intrinsic parameters can be obtained from pure rotations. This indicates that auto-calibration of cameras on cars benefit in case of significant rotations. On an opposite way, excessive rotational speed usually implies motion blur, therefore lesser quality imaging for calibration cases.

In this section we propose a detector of non-zero rotation, taken at slow speeds. The Slow Corner Trigger (SCT) takes the IMU data from the vehicle, measuring linear velocity and lateral velocity.

The SCT involves the next steps:

- Linear velocity is compared with threshold γ and is used to determine if the vehicle is moving slow enough. Negative velocities are not considered for these experiments.
- Angular velocity is compared with threshold α and is used to determine if the vehicle is turning hard enough.
- If both conditions are fulfilled, then the SCT saves the respective frames into a new image folder, whose directory will then be fed into the auto-calibration method.

V. Experiments and Results

In this chapter several experiments will be conducted. The first goal is understanding the physical stereo rig setup and achieving with the extracted information, an accurate virtual camera model that can be used the simulation environment.

The simulation environment also provides another key advantage, being able to quickly iterate through different rig mounting positions without complicated implementation in the real car and comprehend how they would affect our camera parameter refinement method. An exhaustive analysis will be done as well as adding noise at a later stage, to ensure robustness. To finalize, our camera parameter refinement method will be engaged while tracking with calibration errors parameters.

A. Default Setup, Stereo-Camera Parameters

The main goal for this experiment is to extract the characteristics of the real cameras and use those to model the virtual cameras. In case of the VIENA project, the ZED 2i camera provides factory calibrated settings and we will use these as ground truth. Despite having factory calibrated settings we have decided to make our own assessments, since the majority of consumer grade cameras do not have this feature. For these scenarios we have tested 2 techniques: Kalibr and COLMAP to extract the camera intrinsics.

1) Real camera settings: To extract the characteristics of the ZED 2i camera, Fig. 3, the April tags were attached to a wall with good lighting conditions. A rosbag was then recorded while exciting the IMU along its individual axis and targeting the April tags throughout the camera's field of view. There is a noticeable magnitude difference in the RMSE between the left and right camera intrinsics, however the result from both methods are consistent. There are a couple of theories on what might have caused this disparity. The first would be improper lighting or feature disadvantage between cameras. These options were ruled out since the left camera receives better lighting than the right camera and the rig is always pointing to the target. Rerunning the calibration on another dataset yielded similar results. The two prevailing theories are that either our physical calibration setup is not good enough or ZED does not individually calibrate the cameras.

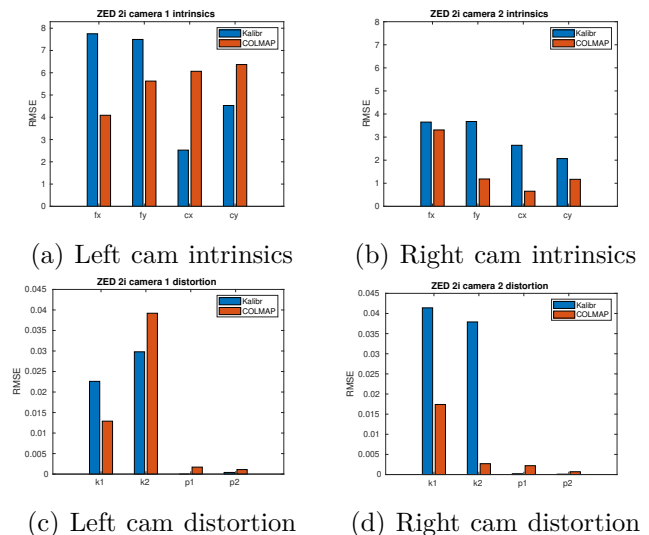


Fig. 3: RMSE error between factory and calibration obtained ZED 2i camera parameters

As seen in from the results in TABLE I, there is a stark difference in required computational time between the Kalibr and the COLMAP calibration. Kalibr is at a clear advantage since it is able to infer the absolute orientation of the April tags. On the other hand, COLMAP sees the April tags as generic features, from which to match and generate map points.

TABLE I: Time to estimate ZED camera intrinsics

	Kalibr	COLMAP
Time to complete (min)	13.21	773.45

2) Virtual stereo-camera calibration: While the goal is to simulate the real camera in a virtual environment, the CARLA simulator has some limitations. For example, you cannot independently specify horizontal and vertical field of view, hence the difference in values between. Similarly to the previous experiment, a rosbag was collected to perform calibration.

As seen from Fig. 4 a) and c), both Kalibr and COLMAP calibration methods can accurately extract the intrinsics of camera 1. Both methods suffer from the lack of camera amplitude in the yaw axis due to the camera being mounted to a terrestrial vehicle.

The analysis of the second camera is analogous to the first camera, as seen in Fig. 4 b) and d), with small differences in magnitude but maintaining consistency across results.

As seen in TABLE I, there is also a significant difference between the computing time of Kalibr and COLMAP. As explained before Kalibr leverages the April tags to infer relative orientation while COLMAP uses them as generic features from which to perform the bundle adjustment. This results in a more complicated bundle adjustment problem, which greatly impacts computing time.

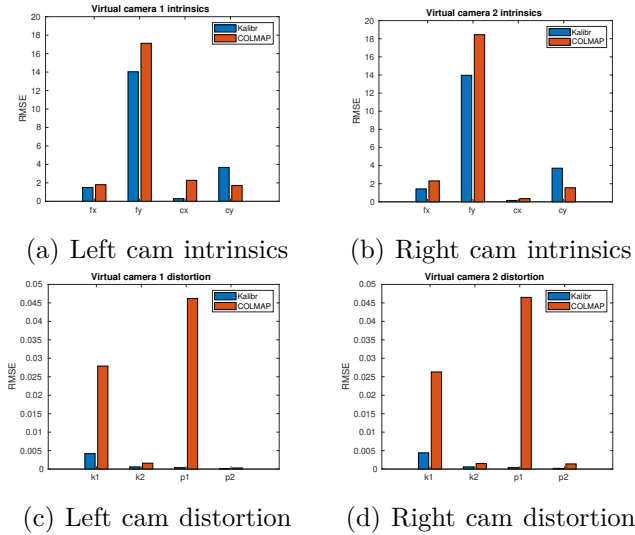


Fig. 4: RMSE error between ideal and calibration obtained virtual camera parameters

TABLE II: Time to estimate virtual camera intrinsics


	Kalibr	COLMAP
Time to complete (min)	9.34	468.72

Conclusions on experiment: The previous experiments show that it is generally preferred to use a traditional calibration method such as Kalibr over a SfM calibration method such as COLMAP, due to its robustness, better distortion results and computing time. It is also shown that having the data collection done while exciting all of the axis (as done for the real camera) equally distributes the errors over the x and y camera components.

B. Stereo Rig Positioning

The goal when positioning the stereo cameras, was to provide representative views of a real system while matching some characteristics that could cause problems.

Through experimentation it was found the COLMAP's stereo and mono bundle adjustment did not converge when the vehicle was moving primarily along the z-axis (forward and backwards).

Camera self-calibration is able to be completed with pure rotation along the yaw axis , but due to the vehicle movement model this is not geometrically possible. However, next experiments show that targeting low speed corners for frame collection have satisfactory results in both computing performance and calibration accuracy.

1) Performance: performance conclusions are able to be inferred from the results of TABLE III. The first is that as expected, with a larger amount of frames the processing time is higher.

TABLE III: COLMAP camera parameters estimation time (in minutes)

	Radiator Mounting	Windshield Mounting	Roof Mounting
1 corner, 90 frames	17.54	18.42	20.93
1 corner, 182 frames	45.87	52.48	56.32
2 corners, 182 frames	50.23	58.85	67.34

The different mounting options also have different performance values. Processing time is directly correlated with the number of matches found in a frame set, meaning mounting options that benefit from added vertical perspective, such as the roof view, will have a higher number of matches and by proxy, higher computational effort.

2) Conclusions on experiments: While the experiments present a one-shot approach (going from initial estimate to a final estimate), the parallel thread where COLMAP is ran should be thought off as an incremental method. As seen from some of the views with 5% and 10% errors, 182 frames is sometimes not enough for the method to fully converge. Effectively if the first iteration of COLMAP allows the camera to increase accuracy from 10% to 5% error, a second iteration with the initial estimate as the 5% error inferred parameters can be ran, further improving the result. Conclusions can be had on the auto-calibration performance of each individual mounting option and what might be causing some of the views to fail.

a) Radiator mounting analysis: independent of sequential and exhaustive matching, results are mediocre. While the focal length f_x is able to converge, f_y can quickly diverge. The vehicle when turning, makes the camera move along the x-axis, however the y-axis is only excited when heavy breaking/acceleration occurs, which causes the camera to only slightly move due to the suspension setup. Effectively the camera amplitude is not enough for a correct bundle adjustment problem to be optimized properly. An additional factor is that a large portion of the lower half captured frame is filled up by the uniform grey color of the asphalt, which difficults the SIFT feature extraction process. This issue is causing c_x to sometimes diverge, as it effectively lacks sufficient information from the lower half of the frame.

b) Windshield mounting analysis: components f_y and c_x might struggle to converge as rapidly as wanted, but they do not seem to diverge. A disadvantage from this view is that there is a significant blind spot in front of the radiator. Depending on the needs, this issue alone can invalidate the view as a potential mounting option.

c) Roof mounting analysis: most often than not, auto-calibration does not converge, sometimes diverging. The assumption is that throughout the corners, the frame has a large portion of the lower half with points that are difficult to differentiate from each other, since they are mostly found in the asphalt cracks. Since the roof view has the best vertical visibility, these points will not get identified as outliers, which would justify the poor bundle adjustment.

C. Calibration-Error Effect on Navigation

An underlying theory is that for terrestrial vehicles the calibration components in the horizontal axis (f_x, c_x) will have a greater impact on tracking than the vertical calibration parameters (f_y, c_y), since there is limited movement in the y axis.

1) Calibration Error on One Camera: The goal of this experiment is to analyze the effect that the calibration error of each parameter can have in tracking performance. Specifically when there are discrepancies between the parameters of both cameras. This can happen when one of the cameras in the stereo rig converges faster or slower than the other, creating these discrepancies.

a) Calibration Error on the Focal Length: The system loses scale and there is a striking difference between the minus and plus components of the error. This happens because at the start of the trajectory, the left side has fewer distinguishable objects from which to track from (sky-box and lack of buildings) and increasing focal length essentially "flattens" the image, causing the tracking errors seen. On the other hand, errors in f_y have little effect on the robustness of the system. The logical reasoning is that flattening the image in the y component is fairly irrelevant since the points extracted suffer small translations in subsequent frames as the camera does not move or rotate from the vehicle's movement.

b) Calibration Error on the Center Point: Adding error to the horizontal center point c_x causes a drastic reduction in the number of point matches, since the left camera is calibrated and the high translation in the x axis causes most points to not match and some false positives. Most attempts at running the experiments fail and the results are not satisfactory. Regarding the vertical center point c_y , it was surprising to see that even a small error in c_y can cause such a significant effect on the vehicle tracking pose. The reasoning is that while the matched points do not translate large amounts in the y axis, which explains why feature matching is acceptable, a small position variation has large consequences on the camera's perceived distance.

2) Calibration Error on Both Cameras: The scenario to characterize is when the calibration procedure fails to fully converge due to lack of calibration images. The result is that both cameras are similarly miscalibrated from their ideal parameters. The procedure is that the intrinsics of both the left and right camera get added the same error, to be then analyzed component by component.

a) Calibration Error on the Focal Length: Results are satisfactory despite being on the x axis. As cameras are miscalibrated by the same amount, most of the reprojection points match, causing correct tracking but with wrong scaling. Regarding f_y , the results are great, even with 20% error it has small to no effect on ORB-SLAM3's ability to track vehicle pose correctly.

b) Calibration Error on the Center Point: Results are satisfactory for c_x , since having calibration errors on both cameras still allows for feature matching between stereo pairs to be effective, causing issues in scale due to the

flattening/unflattening of the points causing perspective distance errors.

3) Conclusions on experiments: As seen by the previous experiments, camera pair parameter mismatches for an autonomous car, can have severe implications in the tracking pose especially on the horizontal components f_x and c_x . Even small errors can cause the points reprojections to fail, greatly diminishing matches and by extent compromising the algorithm's ability to understand its surroundings. Having both cameras with similar calibration errors allows the algorithm to retain its robustness, but has accuracy issues with large values. However, the scale problems only happen with errors to the horizontal parameters f_x and c_x , as errors in vertical parameters f_y and c_y only partially affect the tracking performance compared to the ideal camera parameter tracking.

D. Effect of Visual Artifacts on Navigation

Leveraging the CARLA simulator capabilities a scenario of heavy rain and no fog was played out. Note that the CARLA simulator is not able to simulate water droplets on lenses or car surfaces (like the windshield). While difficult to see, Fig. 5 d) presents a scenario where there are worse lighting conditions (which affects feature extraction), heavy rain (which affects feature matching) and puddles (which affects outlier removal due to matches from the reflections).

1) Radiator mounting: there is a 46.15% drop in map points (MPs) between the good and bad weather scenarios. Most of the points that were no longer detected, were either located on the road and got obscured by the puddles or were points found in the buildings but lacked differentiability and contrast when illumination got worse. The ideal camera parameters meant that there was a successful reprojection when distinguishable points were matched, which explains the good tracking result. Comparing with the remaining views, the radiator mounting lacks vertical perspective, which accounts for the lowest amount of MPs.

2) Windshield mounting: there is a 33.85% drop in map points (MPs) between the good and bad weather scenarios. Windshield view benefits from the added vertical position, allowing for more building features to be extracted, justifying the larger amount of MPs. While having better vertical perspective than the radiator view, it remains worse than the roof view, with the added artifact of the car's hood covering part of the road from which features could have been extracted.

3) Roof mounting: there is a 31.06% drop in map points (MPs) between the good and bad weather scenarios. Comparing with the previous views, roof mounting is the most robust and the one that achieves the largest number of map points. This occurs due to it having the largest vertical perspective, allowing far away features on the ground to be extracted accurately while also being more robust to the environmental effects in this scenario. The added vertical perspective also helps with outlier removal since point reprojection is more accurate.

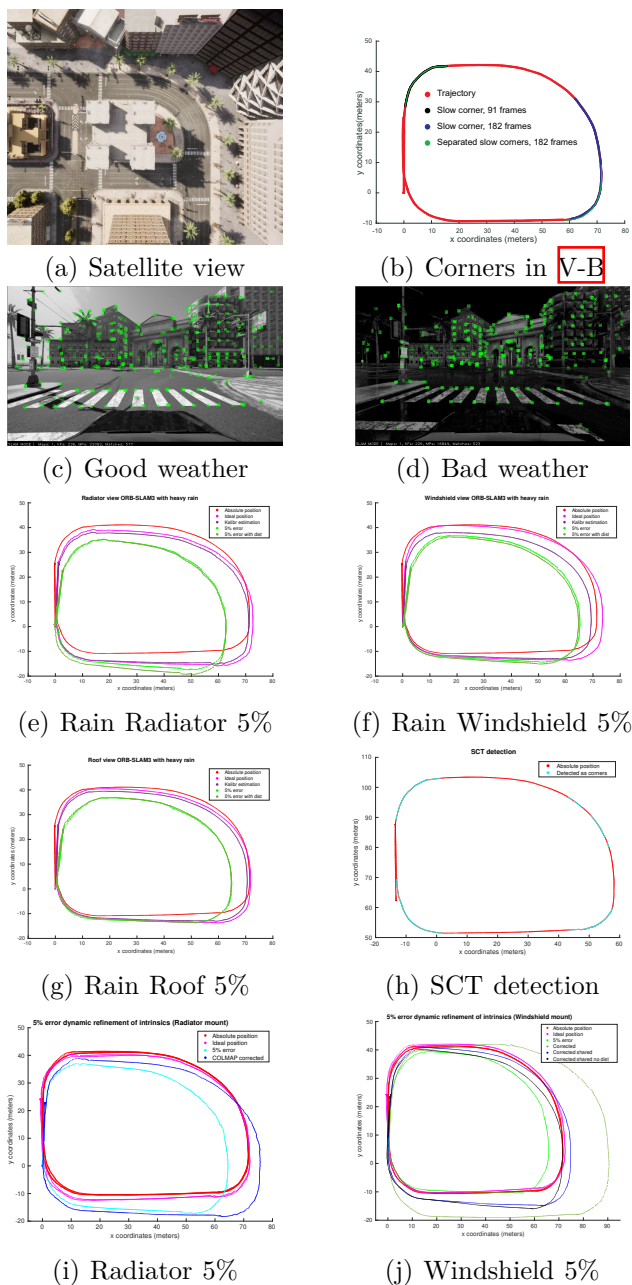


Fig. 5: Sample of experiments conducted

E. Calibration Auto-tuning

The goal of this experiment is to simulate the scenario where ORB-SLAM3 is ran for two of the views (radiator and windshield) since the roof mounting does not converge. Initially these views are miscalibrated by 1%, 5% and 10% and a full lap is completed. Then for the next lap, the camera parameters resulting from the COLMAP auto-calibration are inserted into ORB-SLAM3. This experiment will be done offline, since the SLAM problem needs to be reinitialized to allow the change of camera parameters. Throughout the next Figures, there will be two common plots: the absolute position and the ideal position. The absolute position is taken from the odometry ROS topic of the vehicle and as the name entails, it

represents the absolute position of the vehicle in the CARLA simulator global coordinate system. The ideal position is the ORB-SLAM3 pose estimate with the ideal intrinsics of the virtual camera.

1) Slow Corner Trigger: Despite some outliers on the corners, the SCT is able to accurately detect the corners of interest only from IMU data. A threshold of $\pm 0.25 \text{ rad/s}$ was set to detect when the actor is cornering and a streak counter was implemented, to account for when the car is cornering but the angular velocity does not surpass the threshold. The streak counter is incremented when the actor is outside the threshold at that time instant. When the actor is inside the threshold, meaning theoretically it is not cornering, the streak counter is decremented. If it reaches 0, then the vehicle is not cornering. A cap of 10 was set to the streak counter, to prevent long corners from incrementing the counter too much and identifying the straights as corners.

2) Conclusion on experiments: While the roof view would be superior for SLAM due to it's higher visibility, if auto-calibration is desired, the remaining option is the radiator view. The windshield view must be discarded due to the glass windshield acting as a lens, making it unreliable when small distortion values and discrepancies between the left and right camera intrinsics are had.

VI. Conclusion and Future Work

The work described in this thesis provides the foundation for the localization and mapping segment of the VIENA project. It is intended at a future date, to use the VIENA-ROS wrapper that was created for the purposes of this work, to connect the controller and vehicle model with ORB-SLAM3 in the real vehicle and achieve a fully functioning navigation segment. In the meantime, multiple simulator options were researched, ending with the full integration of the CARLA simulator into the pipeline with the creation of the VIENA-CARLA wrapper, allowing the controller to issue throttle and steering commands to the virtual vehicle.

It was also proposed to analyze the effect of a poorly calibrated stereo rig on the VSLAM pose estimate tracking performance. Cameras can be poorly calibrated due to an incomplete calibration process or from environmental effects such as heat lightly warping the lenses. In this work it was shown that some parameters have more influence on accurate tracking than others and that reducing discrepancies between cameras is critical for maintaining robustness.

An auto-calibration procedure was suggested and tested for each of the mounting options, giving a potential option to circumvent a poorly calibrated stereo rig without needing special targets. Suggestions to achieve a more robust auto-calibration from motion were given, such as using images from slow corners to approximate the pure rotation scenario.

References

- [1] Agapito, L., Hayman, E. and Reid, I. (2001) Self-calibration of rotating and zooming cameras. *International Journal of Computer Vision*, 45:107–127.
- [2] Amini, A., Wang, T., Gilitschenski, I., Schwarting, W., Liu, Z., Han, S., Karaman, S. and Rus, D. (2021) VISTA 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. *International Conference on Robotics and Automation (ICRA)*, abs/2111.12083, pages 2419–2426.
- [3] Bay, H., Tuytelaars, T. and Van Gool, L. (2006) Surf: Speeded up robust features. *Computer Vision – ECCV 2006, Berlin, Heidelberg, Springer Berlin Heidelberg*, pages 404–417.
- [4] Campos, C., Elvira, R., Rodriguez, J. J. G., Montiel, J. M. M. and Tardos, Juan D. (2021) ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890.
- [5] Chen, K., Shen, F., Zhou, J. and Wu, X. (2020) Simulation platform for sins/gps integrated navigation system of hypersonic vehicles based on flight mechanics. *Sensors*, 20(18),5418.
- [6] Dang, T., Hoffmann, C. and Stiller, C. (2009) Continuous stereo selfcalibration by camera parameter tracking. *IEEE Transactions on Image Processing*, 18(7):1536–1550.
- [7] Datta, A., Kim, J. and Kanade, T. (2009) Accurate camera calibration using iterative refinement of control points. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1201–1208.
- [8] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A. and Koltun, V. (2017) CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, PMLR 78:1–16.
- [9] Fischler, M. A. and Bolles, R. C. (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 26(6):381–395.
- [10] Forsyth, D. A. and Ponce, J. (2002) *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference
- [11] Furukawa, Y. and Ponce, J. (2009) Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84:257–268.
- [12] Galvez-Lopez, D. and Tardos, J. (2012) Bags of binary words for fast place recognition in image sequences., *IEEE Transactions on Robotics*, 28:1188–1197.
- [13] Huber, P. J. (1964) Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101.
- [14] Ebrahim Karami, Siva Prasad, and Mohamed S. Shehata. Image matching using sift, surf, BRIEF and ORB: performance comparison for distorted images. CoRR, abs/1710.02726, 2017.
- [15] Kaur, P., Taghavi, S., Tian, Z. and Shi, W. (2021) A survey on simulators for testing self-driving cars. *arXiv preprint, arXiv:2101.05337*.
- [16] Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K. and Burgard, W. (2011) G2o: A general framework for graph optimization. *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613.
- [17] Lindenberger, P., Sarlin, P., Larsson, V. and Pollefeys, M. (2021) Pixel-perfect structure-from-motion with featuremetric refinement. *2021 IEEE/CVF International Conference on Computer Vision (ICCV) IEEE - Institute of Electrical and Electronics Engineers Inc.*, pages 5967–5977.
- [18] Lowe, D. (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.
- [19] Maye, J., Furgale, P. and Siegwart, R. (2013) Self-supervised calibration for robotic systems. *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 473–480.
- [20] Mount, D. M., Netanyahu, N. S. and Le Moigne, J. (1999) Efficient algorithms for robust feature matching. *Pattern Recognition*, 32(1):17–38.
- [21] Mur-Artal, R. and Tardós, J. D. (2017) Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803.
- [22] Mur-Artal, R. and Tardós, J. D. (2017) Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.
- [23] Paden, B., Cap, M., Yong, S. Z., Yershov, D. and Frazzoli, E. (2016) A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55.
- [24] Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., Rus, D., Ang, M. H. (2017) Perception, Planning, Control, and Coordination for Autonomous Vehicles. *Machines*, 5(1).6.
- [25] Portelinha, F. A. R. (2022) Steering and speed control for autonomous electric vehicles. In *MSc in Electrical and Computer Engineering, Instituto Superior Técnico*.
- [26] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A. (2009) ROS: an open-source robot operating system. *ICRA workshop on open source software*, 3, pages 1–6.
- [27] Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Mozeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., Agafonov, E., Kim, T. H., Sterner, E., Ushiroda, K., Reyes, M., Zelenkovsky, D. and Kim, S. (2020) LGSVL simulator: A high fidelity simulator for autonomous driving. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, IEEE Press, pages 1–6, abs/2005.03778.
- [28] Rublee, E., Rabaud, V., Konolige, K. and Bradski, G. (2011) Orb: An efficient alternative to sift or surf. *2011 International Conference on Computer Vision*, pages 2564–2571.
- [29] Schönberger, J. L. and Frahm, J. (2016) Structure-from-motion revisited. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113.
- [30] Shah, R., Deshpande, A. and Narayanan, P. J. (2014) Multistage sfm: Revisiting incremental structure from motion. *2014 2nd International Conference on 3D Vision*, volume 1, pages 417–424.
- [31] Sucar, E., Liu, S., Ortiz, J. and Davison, A. J. (2021) iMAP: Implicit mapping and positioning in real-time. *IEEE/CVF International Conference on Computer Vision*, pages 6229–6238.
- [32] Tourani, A., Bavle, H., Sanchez-Lopez, J. L., and Voos, H. (2022) Visual slam: What are the current trends and what to expect? *Sensors*, 22(23),9297.
- [33] Triggs, B., McLauchlan, P. F., Hartley, R. I. and Fitzgibbon, A. W. (1999) Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms 1999*, pages 298–372.
- [34] Wang, Z., Wu, S., Xie, W., Chen, M. and Prisacariu, V. A. (2021) Nerf: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.
- [35] Wei, X., Zhang, Y., Li, Z., Fu, Y. and Xue, X. (2019) DeepSfM: Structure from motion via deep bundle adjustment. *Computer Vision – ECCV 2020 16th European Conference, Glasgow, UK*, pages 230–247.
- [36] Zanuttigh, P., Marin, G., Dal Mutto, C., Dominio, F., Minto, L. and Cortelazzo, G. M. (2016) *Operating Principles of Structured Light Depth Cameras*, Springer International Publishing: Heidelberg, Germany, pages 43–79.
- [37] Zhang, Z. (2000) A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.