University of Tartu
Faculty of Science and Technology
Institute of Technology

Pavel Shumejko

# Robust Solution for Extrinsic Calibration of a 2D Laser-Rangefinder and a Monocular USB Camera

Master's Thesis (30 ECTS)
Robotics and Computer Engineering

Supervisor:
Assoc. Prof. Karl Kruusamäe

Tartu 2018

# Content

# List of Figures

# List of Tables

# Abbreviations

**LIDAR -** Light detection and ranging sensor

**UAV -** Unmanned aerial vehicle

**ROS -** Robot Operating System

**SVD -** Singular-value decomposition

**GCS** – Ground control station

**FOV** – Field of view

# Abstract

With today's technological development, almost every robotic system such as UAVs, mobile robots or autonomous ground vehicles are equipped with LIDARs and cameras. These sensors provide data which can be used to estimate the position and orientation of mobile robotic systems in the environment. The combining of the sensory data, commonly known as sensor fusion, contributes to the improvement of the environmental perception of the robots. In order to fuse this data, an extrinsic calibration of the sensors used is needed. In this thesis, a custom robust extrinsic calibration method of a 2D LIDAR and a USB camera is presented. The proposed approach uses a small calibration target with AR tags in the overlapping FOV of the sensors and projections of vectors onto a common plane to estimate the pose of the camera in reference to the LIDAR based on few constraints. The result of the calibration is a transformation matrix that can be later used by other applications. The calibration process is fast and simple to perform and is intended to be improved and used as a part of the greater UAV control system of the company Pro-Drone.

**CERCS:** T111 Imaging, image processing; T125 Automation, robotics, control engineering

**Keywords:** Camera, LIDAR, AprilTags, Extrinsic calibration, Intrinsic calibration, calibration target, transforms, correspondences

# Resümee

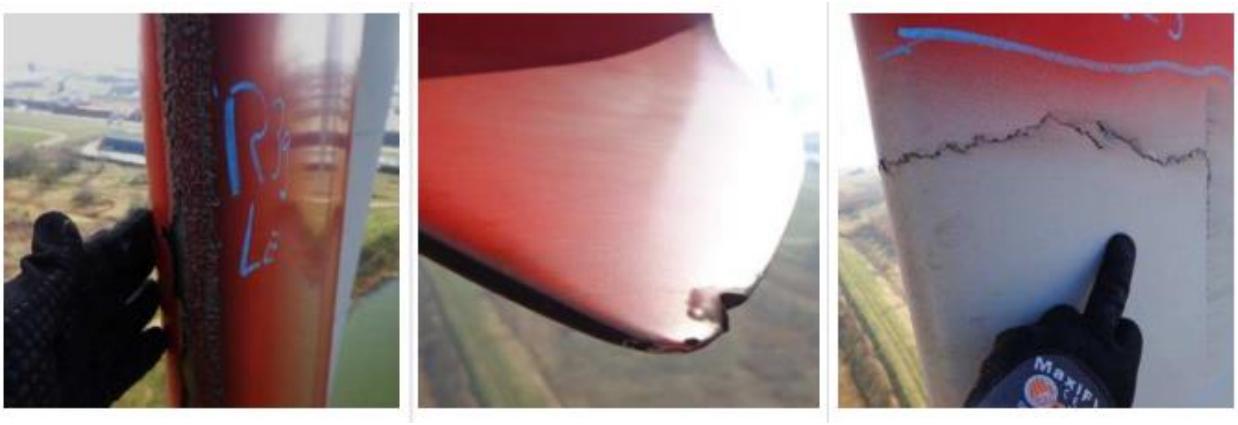
Tänapäeva tehnoloogia arengute juures on peaaegu igal robootilisel süsteemil, nagu näitaks UAV-d, mobiilisetel robotitel või autonoomsetel maismaasõidukitel lidarid ja kaamerad. Need sensorid võimaldavad robotsüsteemidel hinnata nende endi asukohta ja paiknemist keskkonnas. Sensoritest tulevate andmete kombineerimine parendab robootsüsteemide keskkonna tajumise võimekust. Selleks et andmeid kombineerida, on vaja sensoreid väliselt kalibreerida. Antud töös on välja toodud robustne 2D lidari ja USB kaamera välise kalibreerimise meetod. Antud lähenemine kasutab kalibreerimiseks väikest sihtmärki AR siltidega sensorite kattuvas nägemisväljas ja vektorite projektsioone sensorite ühisel tasandil, et hinnata kaamerate asukohta lidarite suhtes. Kalibratsiooni tulemuseks on teisendusmaatriks, mida saavad hiljem kasutada kolmandad rakendused. Selline kalibratisoon on kiire ja lihtne sooritada, ja see on mõeldud täiustamiseks ja kaustamiseks ühe osana suuremast UAV kontrollsüsteemist ettevõtte Pro-Drone droonidel.

**CERCS:** T111 Pilditehnika; T125 Automatiseerimine, robootika, control engineering;

**Märksõnad**: kaamera, LIDAR, AprilTags, välimine kalibreerimine, sisemine kalibreerimine, kalibreerimise sihtmärk, teisendused, vastavused

# 1 Introduction

Wind turbines are one of the fastest growing clean energy sources in the world. The number of wind turbine plants installed over the past two decades has increased the need for manufacturing, installation, maintenance and supporting services. As wind turbines are located outdoors and usually in remote locations, they are affected by different weather conditions. In this harsh environments, such as coastal or arid regions, blade damage due to erosion, lightning strike or cracks can appear (Figure 1).



*(a)* *(b)* *(c)*

**Figure 1.** *Blade damage due to erosion (a), lightning strike (b) and cracks (c) [1].*

In order to detect damage on the blades, regular inspections and maintenance is carried out. The current inspection technologies usually require human experts in blade damage inspection to be present on site and to be put in dangerous situations by using safety ropes or being lifted high by cranes [2]. Another way to inspect the blades is for the experts to use a ground camera but this has proven to give less quality data [2]. The drone solutions for blade inspection are gaining popularity recently and have proven to be less dangerous and time-consuming [3].

Pro-Drone is a startup company that develops aerial wind turbine blade inspection solutions. It was founded in 2015 by Andre Moura De La Cruz and it is based in Porto Salvo near Lisbon, Portugal [4]. Pro-Drone's mission is to revolutionize how wind turbine blades are inspected by

developing automated unmanned aerial vehicle (UAV) solutions for wind turbine inspection [4]. As the current wind turbine inspection methods are outdated, time consuming and/or quite expensive, the UAV solutions are becoming the current technological trend and promising to deliver a high quality, reliable and safe inspection at a minimal cost [3]. Pro-Drone's solution consists of a customized UAV control that allows semi-autonomous inspection of wind turbine blades and a payload containing all of the necessary equipment for the inspection. This solution is supposed to replace the existing ways of inspecting wind turbine blades.

Sensors such as LIDARs and cameras play a valuable role in the automation of different robotics systems. In recent years, the combination of LIDARs and cameras as well as GPS and IMU systems have become a common practice in various robotic systems such as unmanned aerial or ground vehicles [5][6]. These sensors improve the robot's perception and understanding of the surrounding environment. However, all these sensors have technological limitation which can be overcome by fusing the data from multiple types of sensors in order to improve the quality of information. For instance, a LIDAR provides high accuracy range measurements of very large field of view but the collected data is sparse and the sensor cannot be utilized in strong rain or snow. On the other hand, a camera usually has a limited field of view and is sensitive to environmental illumination changes. In addition, cameras also lack in quality of depth information but complement in dense and diverse data and are popular for object recognition. To effectively fuse the image and LIDAR data, a precise estimation, or in other words, an extrinsic calibration of the relative pose of both sensor frames is required. This sets the extrinsic calibration as a prerequisite for any sensor fusion method.

Pro-Drone is continuously trying to improve their solution for aerial wind turbine blade inspection in order to achieve safer, faster, and more reliable inspection and in the meantime retain the quality of the collected data. Their goal is to develop a fully autonomous inspection with the possibility of going for a full wind turbine inspection in one go instead of the current semi-autonomous method, which inspects every blade separately. Since the existing solution that Pro-Drone provides uses a LIDAR and camera, the automated blade inspection can be further improved with the implementation of sensor fusion applications. For instance, an intelligent way of combining the sensor data from the onboard LIDAR and camera can improve the blade and

hub detection thus increasing the awareness and environmental perception of the UAV. The camera can also be used for some image processing applications (detecting and segmenting different parts of the blade/turbine). By combining the LIDAR and camera data it is possible to improve the navigation and the positional awareness of the drone. As a first step in implementing any sophisticate sensor fusion method, a calibration of the sensors is required. This thesis presents my work at Pro-Drone focusing on development and implementation of a calibration tool for a 2D laser rangefinder and USB camera.

## 1.1 Aim and Contribution

The aim of the thesis is to present a custom solution for extrinsic calibration of camera and laser rangefinder.

In order to achieve this solution, the following steps were taken:

- Analysis of any existing solutions for the problem
- Intrinsic calibration of the camera
- Design and production of a calibration target
- Ground truth pose measurements of the camera in reference to the LIDAR
- Calibration algorithm and its software implementation based on the system requirements and limitations
- Testing and data acquisition
- Analysis and validation of the results

## 1.2 Thesis Outline

The thesis introduces the need for calibration, provides background information of the problem and presents different approaches by analyzing their strengths and weaknesses. Based on the system requirements and limitations, a custom solution for extrinsic calibration of a 2D LIDAR and a monocular camera is presented together with an analysis of the obtained data and results. The proposed calibration process is fast and simple to perform and does not require from the end user any previous knowledge of calibration procedures.

Chapter 2 contains a background of the current blade inspection technologies, Pro-Drone's inspection procedure and product together with an overview of existing calibration solutions, their significance, and the basics of camera-LIDAR calibration. The problem definition and the specific requirements and limitations are introduced in Chapter 3. The calibration process with the proposed technical setup and extrinsic calibration algorithm are presented in Chapter 4. The obtained results of several tests are shown and compared in Chapter 5. Finally, the current state and any future work of the developed calibration tool are discussed in Chapter 6.

# 2 Background

## 2.1 Inspection Technologies

Studies have shown that the state of wind turbine blades can account for up to 30% of the turbine power generation [7]. The wind turbine blades are exposed to different weather conditions and can be damaged due to erosion, lightning strikes, and cracks caused by other weather factors. The goal of a wind turbine blade inspection is to evaluate the structural health of the blades and identify any need for repairs. Therefore, wind turbine blade inspections are scheduled on a regular basis. However, the inspections increase the turbine's downtime which lowers its production of energy [1]. This sets the choice of an inspection method as a tradeoff in the turbine downtime and the cost, robustness and safety of the operation.

There are several methods (Figure 2) that are used today to inspect a wind turbine blade [2]:

- Using a **ground based camera** to take pictures of the blade.
- **Rope access** involves experts in blade damage inspection to be lowered from the turbine's hub to the end of the blade.
- **Lift cranes** can lift the experts close to the blade
- **UAVs** that are manually controlled to take pictures of the blade

The ground based camera solutions are quick and safe to perform but lack in quality and type of data that they can obtain since they are distanced far from the blades as the height of wind turbine towers is typically between 70 m to 120 m but in some extreme cases up to 220 m [8].

The rope inspection requires human experts to be present on site and to be put in potentially dangerous situations. This method can be quite time consuming and expensive. On the other hand, the presence of the experts usually allows minor damages to be repaired during the inspection.

More recent methods use drones mounted with high resolution photographic cameras to perform fast and accurate inspections. However, as the drone is usually controlled manually by an on-site operator, there is high-risk for collisions with the blades due to inconvenient viewing angles,

distance and strong winds. Today there are several companies in the market that try to automate the drone control process [4][9].

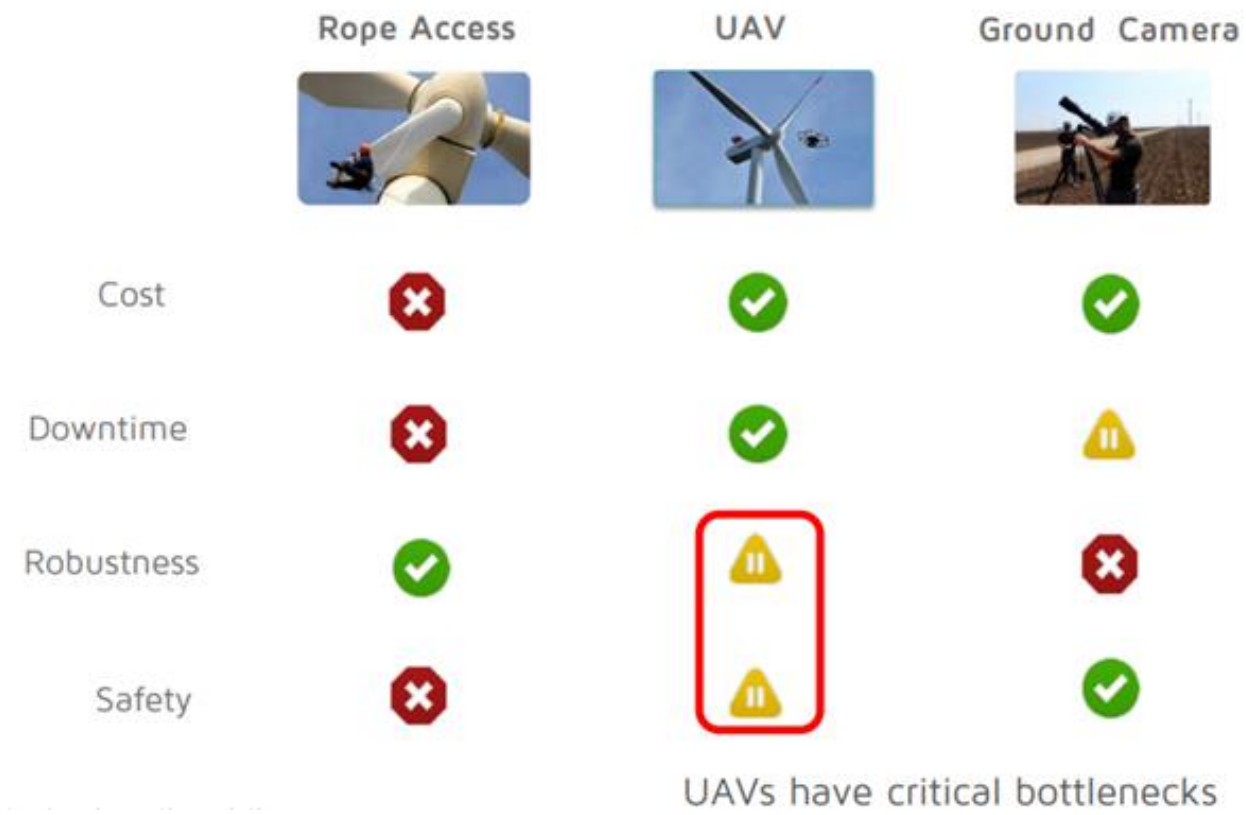



**Figure 2**. *Comparison of different inspection procedures [10]. The green, yellow and red markers indicate respectively whether a procedure improves, is neutral or has drawbacks in the specified inspection selection criteria factors.*

## 2.2 Pro-Drone's Inspection Procedure

Pro-Drone's inspection of a wind turbine blade is carried out in several steps:

1. Initially the operators of the wind turbine park rotate the blades until one of them is parallel to the turbine tower and is facing downwards.
2. Once the blade is in place, the pilot manually takes off and positions the drone in proximity of the suction side (right side) of the blade (Figure 3). The drone stays in this position until the software, based on the sensor data, can identify and lock onto the blade.

3. Once the blade is locked the drone goes into autonomous mode by going down to the tip of the blade and then going up the suction side until the end of the blade/turbine hub is detected.
4. When the hub on the suction side is reached the drone rotates to the leading edge (front side) of the blade and goes down until the blade's tip is reached.
5. When the blade's tip is detected a rotation to the pressure side (left side) of the blade is performed and the inspection continues until the hub is reached again.
6. At the pressure's side hub, the drone rotates to the trailing edge (back side) of the blade and goes down until it reaches the tip of the blade.

This is where the autonomous part of the inspection ends and the pilot takes manual control over the drone in order to land it safely. Then the wind park operators rotate the blades so that the next blade is in position for inspection and the whole step by step procedure described above is repeated.

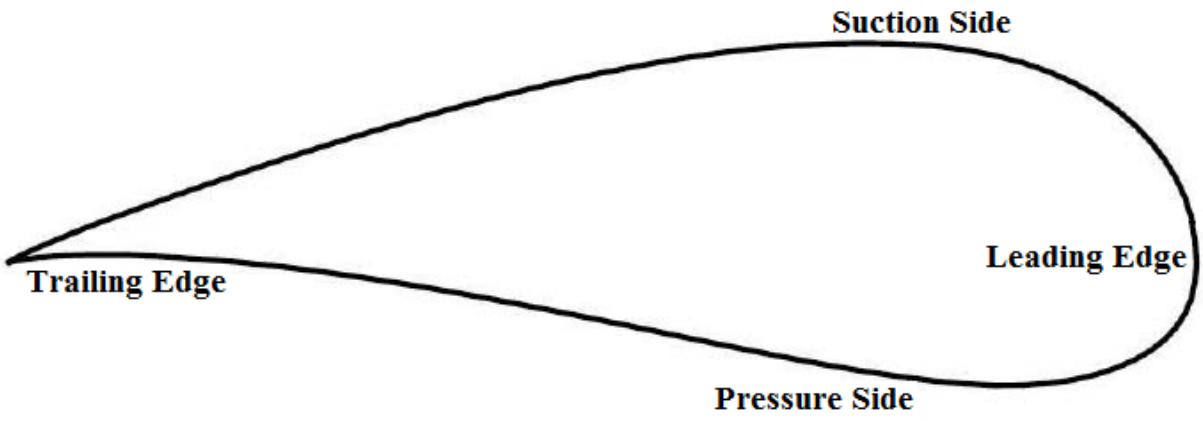



**Figure 3.** *Wind turbine blade sides.*

## 2.3 Pro-Drone's Commercial Product

Pro-Drone's integrated solution has two key technological components [4]:

1. **Proprietary control feature** that enables the UAV to have positional awareness in relation to the wind blade
2. **Custom payload** optimized for the specificities of inspecting a blade

Pro-Drone also provides a full cloud-based software solution for storing and processing inspection data to enable an intuitive analysis. The software platform also allows the storage of surveys and generation of reports for the completed inspections.

Their proprietary control feature provides the position of the UAV relative to the blade. The information provided by other onboard sensors allows the UAV to always keep a safe distance and follow a stable and robust path that covers the entire blade [4]. This makes the blade inspection a semi-autonomous and robust procedure as it no longer depends on pilot expertise and ensures that the UAV does not collide with the blade. This offers reliability, repeatability and operational safety when an inspection is performed. At this point, continuous manual control is only needed for takeoff and landing, or in the case of an emergency override. Furthermore, Pro-Drone is continuously improving the control system with the aim of reaching a fully automated blade inspection.

The current product consists of a custom payload (Figure 4) optimized for wind turbine blade inspection. The payload is mounted on a drone and it consists of a gimbal with the following devices:

- Onboard computer (ODROID-XU4)
- 2D laser range finder (Hokuyo UTM-30LX)
- Monocular USB camera (Logitech c920 HD Pro Webcam)
- Photographic camera (Sony alpha 7R)

The ODROID handles the control and navigation of the drone and processes the sensor data. The 2D laser range finder (hereinafter referred to as LIDAR) is used for detecting the objects for

inspection. The USB camera is used to stream a real time video feedback to the ground control station (GCS) and can be used for image processing. The photographic camera is constantly taking high quality pictures of the blade which are later analyzed in the post processing pipeline to assess any structural damage.

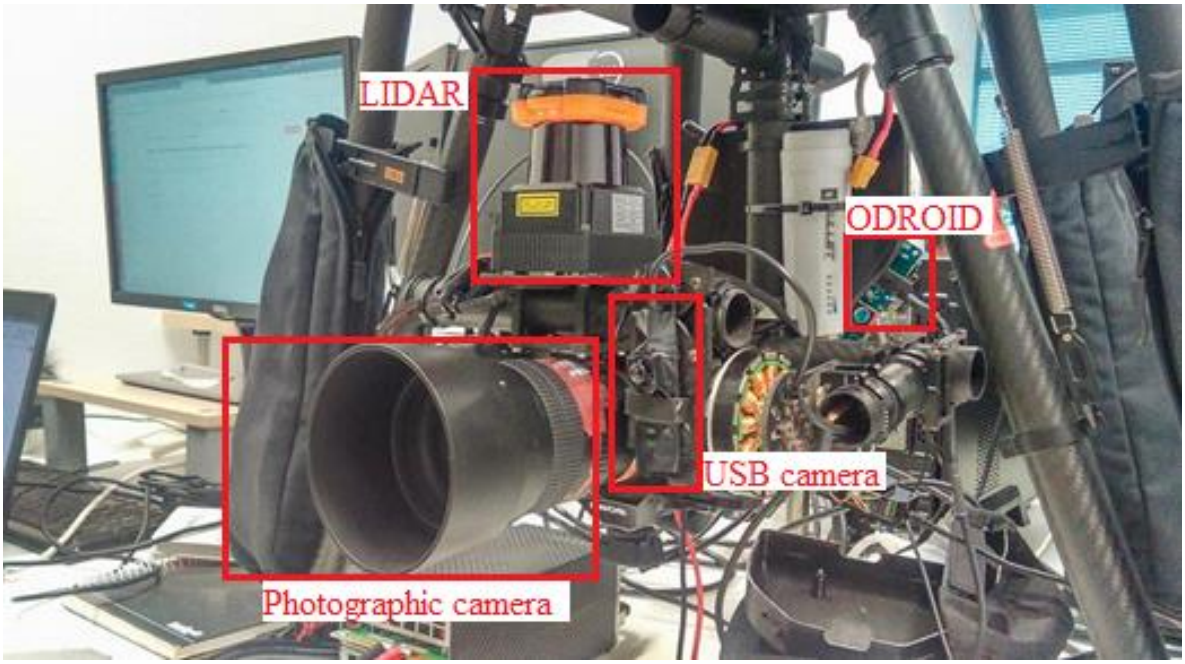



**Figure 4.** *Pro-Drone custom payload during the development of the calibration tool*.

## 2.4 Calibration

Calibration in measurement technology and metrology is the comparison of measurement values from a device under test with those of a calibration standard or another measurement device of known accuracy [11]. The outcome of the comparison can result in:

- no significant error being noted on the device under test,
- a significant error being noted but no adjustment made, or
- an adjustment made to correct the error to an acceptable level.

In order to obtain accurate sensor measurements so that we can combine data from a camera and a LIDAR, the extrinsic and intrinsic parameters must be estimated. This means that the camera-LIDAR calibration itself can be divided into [12]:

- Intrinsic camera calibration - affine transformation of a 3D point from camera coordinates to image coordinates. The result of an intrinsic calibration is the camera matrix containing the intrinsic parameters.
- Extrinsic calibration - rigid transformation between the LIDAR and the camera. The result of the extrinsic calibration are the translation and rotation matrices which denote the coordinate system transformations from LIDAR coordinates to camera coordinates.

According to MathWorks [13], the camera calibration, also known as camera resectioning, estimates the parameters of the lens and the image sensor of a camera. These parameters can then be used to correct the effects of lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene. This can be useful in applications such as machine vision to detect and measure the size of objects, robotics, navigation systems, and 3D scene reconstruction.

The full calibration (Figure 5) process estimates the intrinsic and extrinsic parameters [12]. The intrinsic calibration gives the intrinsic parameters (intrinsics) of the camera such as the focal length, principal point, skew, scale and distortion parameters [13]. The output of the intrinsic calibration is usually the camera and distortion matrices that contain these parameters. The camera matrix can also be used for projective transformation of 3D world scene points onto the 2D image plane.

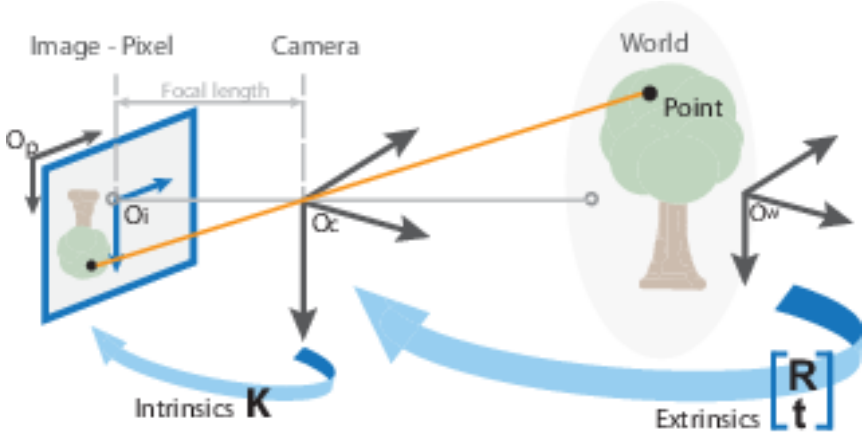



**Figure 5.** *Extrinsic and intrinsic parameters [13].*

The extrinsic calibration estimates the six degrees of freedom transformation between the sensors [12][13]. The extrinsic parameters (extrinsics) represent the location of the sensors in the 3D scene. In our case, the extrinsic calibration (Figure 6) is the rigid body transformation between the LIDAR and the camera.

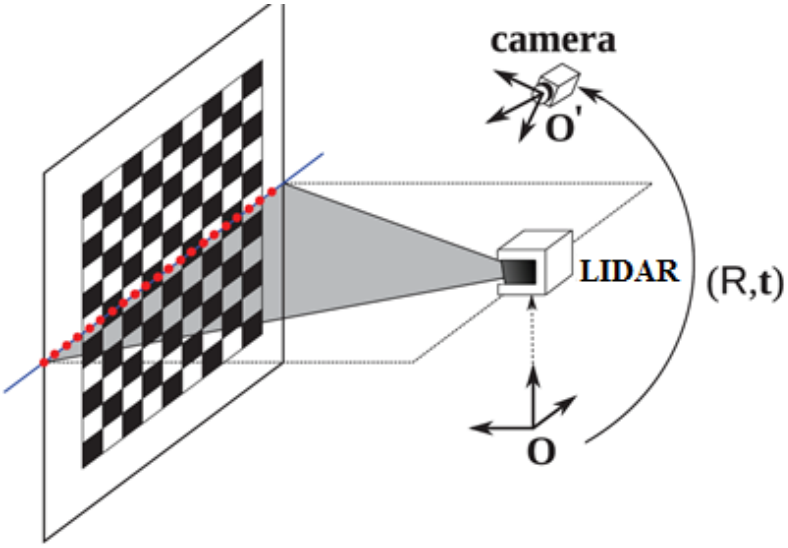



**Figure 6.** *Typical calibration system setup containing a LIDAR, camera and a calibration object [14].*

Extrinsic calibration between a LIDAR and a camera can be roughly classified into two categories: direct and indirect correspondence searches [12]. The direct method requires hand-labeling of correspondences between the LIDAR points and camera pixels with the help of special equipment such as IR cameras since the LIDAR operates outside of the visual spectrum [15]. Alternatively, we can indirectly estimate the correspondences by finding and matching the same features separately in each modality.

The extrinsic calibration approaches can be also divided into target and targetless approaches whether they use a calibration target (object) [12]. Targetless approaches have difficulty in matching a low density LIDAR (such as 2D LIDAR) data to a relatively dense image data [12].

## 2.5 Related Work

Today, there are multiple different approaches to camera-LIDAR calibration (e.g., [16], [17]]. All of these methods tend to require many different steps in order to achieve reliable and accurate results. They differ based on application areas as most of them are tailored for the system requirements and the specific sensor setup. There are both 3D [18] and 2D LIDARs [19] available on the market. In addition to this, different types of cameras, such as monocular IP camera, the popular Kinect camera [20] or a stereo camera for range imaging [21] are often used in the sensor setups.

S. Sim, J. Sock, and K. Kwak have proposed a robust extrinsic calibration algorithm that can be implemented easily and has small calibration error [12]. In this case extrinsic calibration parameters are estimated by minimizing the distance between corresponding features projected onto the image plane. These correspondences are the edge and centerline features on a v-shaped calibration target (Figure 7). The calibration target design consists of two planar boards arranged in a v-shape so that the centerline, left and right boundaries are marked with a black tape to enable detection of the edges in the image. The proposed calibration algorithm consists of three steps: data acquisition, feature extraction and optimization. One drawback of the proposed solution is that for the feature extraction manual selection of the line features and border points is needed.

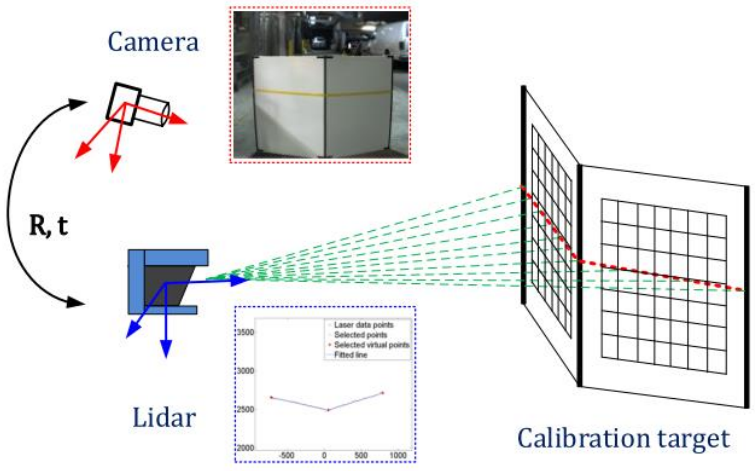



**Figure 7.** *The relative pose between the line scanning LiDAR and the camera is estimated using features extracted from images of a v-shaped target [12].*

The approach proposed by Q. Zhang and R. Pless uses a planar checkerboard pattern as a calibration target [22]. The calibration process is based on observing the planar checkerboard pattern and solving for constraints between the views of a planar checkerboard pattern from the camera and laser range finder [22]. A point-to-plane constrain between the laser points and the pattern plane is used to obtain a linear solution which is employed as initial value in a final nonlinear optimization process. The extrinsic calibration parameters are estimated by solving a nonlinear optimization problem that minimizes the position error of the LIDAR data with respect to the calibration board in 3D. The downside of this approach is that the initial value may not be a valid pose, it sometimes leads to a local minimum and is unreliable at long distances because of the need to detect the corners of the checkerboard calibration target. It also assumes that the camera is already calibrated and that there is no significant lens distortion but it later introduces latter refinement of the camera intrinsic parameters. Zhang and Pless argue that theirs was the first published extrinsic calibration tool for a camera and 2D LIDAR. The source code is written in MATLAB and is available to be downloaded and tested [23].

Z. Chen, L. Zhuo, K. Sun, and C. Zhang propose a simple method for calibration of a camera and a laser range finder that uses a cube shaped object (Figure 8) as the calibration target [16]. The cube has to be simultaneously visualized by both the LIDAR and the camera and requires three changes of the pose of the cube to gather data. The extrinsic calibration here is based on a constraint that in the image coordinate system, the projection of laser point onto the image plane is from the three cube edges. It uses a point-to-line constraint, for which a linear equation for the unknown variable is solved. The downside of this method is the need of a good corner detection and line fitting algorithm to extract the corners of the cube and the lines that correspond to the edges. Similar approach [24] introduced line-to-plane and point-to-plane constraints and uses a stereo camera. This method instead of the traditional calibration target relies on observations of orthogonal trihedrons (scene corners) and is argued to have better performance in terms of the precision of the estimated relative pose than the methods that use V-shaped patterns.

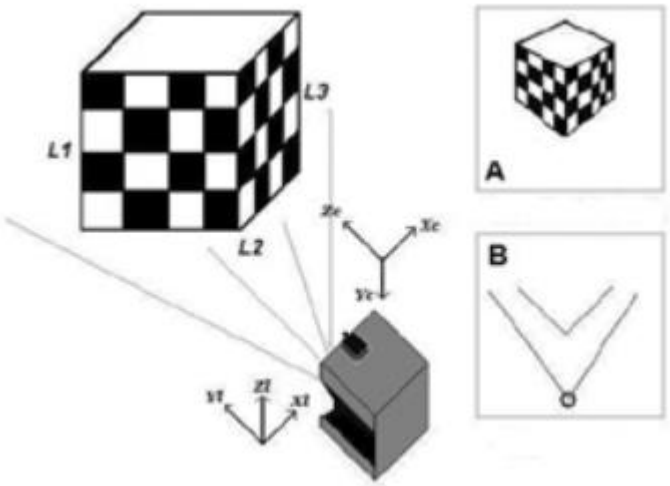



**Figure 8.** *The configuration for calibration between a camera and a laser ranger finder. The calibration cube are both imaged with the camera (Block A) and the laser range finder (Block B) [16].*

In some cases, SVD (Singular-Value Decomposition) can be used to find the optimal rigid transformation that aligns two sets of corresponding points [25]. The downside of this method is that it requires corresponding datasets of identical dimensions. Otherwise in the case of different dimensionality a complexity in the correspondence search can arise. An approach using SVD and 3D-3D correspondence is presented by Jong-Eun Ha [26]. Here for extrinsic calibration of a camera and laser range finder, a checkerboard calibration structure of a plane with a triangular hole is used as a calibration object. The corresponding 3D coordinates are obtained by finding the absolute location of the laser scan data on the calibration object and creating 3D control points. Once these points are created the corresponding 3D coordinates of control points with respect to the camera coordinate system are obtained through camera calibration. To find the absolute location, lines of varying gradient and intercept are generated on the calibration structure and a matching score between the generated line and real scan data is computed. The line which has the minimum value from the scoring function among the whole search space is chosen and 3D control points are generated from a fitted line of the real laser scan data. In the end, SVD is used on the 3D-3D correspondence data to compute the extrinsic parameters between the camera and laser range finder.

The more recent approaches for extrinsic calibration between cameras and LIDARs are focusing on automatic and targetless methods [17]. The drawback of these approaches is the need of high density 3D points. In addition, they work in specific environments for finding correspondences such as urban areas and give poor results when used in natural environments [12].

# 3 Problem Definition

## 3.1 Problem Statement

In order to fuse data from a LIDAR and a camera the relative transformation between these sensors needs to be known. In the case of Pro-Drone, these sensors are a 2D LIDAR and an USB monocular camera that are mounted on a gimbal as part of their custom payload for blade inspection (Figure 4). As the sensors are mounted in fixed positions, this means that the pose of the camera relative to the LIDAR can be determined with careful measurements. Based on these measurements a transformation matrix can be formed and used in the system to transform data from one sensor coordinate frame to another. However, in few cases it has been noticed that due to vibrations during transportation and operation of the drone or an interference from a human factor (for example when the sensors are cleaned from dust) minimal displacements of the sensors (especially the camera), from their original state may occur. Even though these offsets in the pose are usually small enough not to disturb the normal operation of the drone during an inspection, they will significantly influence the output of any application that fuses data from the sensors. An example of an visual output introducing an projection offset of around 2 cm due to an displacement (rotation) of around 1° in one of the axes when the calibration object is placed at a distance of one 1 m is shown in Figure 9.

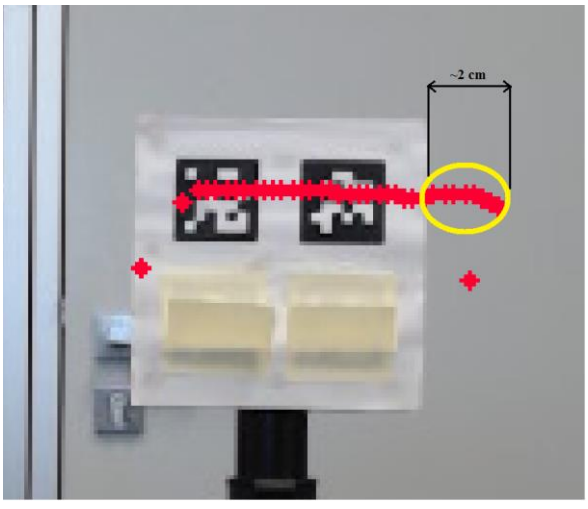



**Figure 9.** *Visual offset (yellow) of the projected points (red) due to an error in the estimation of one of the angles.*

Every time these offsets are registered, the physical measurement of the sensor's pose will have to be repeated which demands special qualification and equipment for measuring and it can be quite time consuming and tedious. There is also the fact that the exact orientation of the camera relative to the LIDAR is difficult to measure (requires precise milimetric measurements and measuring equipment). As Pro-Drone's product is used by licensees and users that do not have a direct access to the code or are often not qualified to measure and adjust the sensor offsets this would mean that the product would need to be sent back for additional maintenance every time when these offsets occur, potentially resulting in added complexity and dissatisfaction on the customer's side. This is the reason why there is a need for a fast, simple, and accurate extrinsic calibration tool that can be used when the offsets are detected or when a scheduled maintenance is carried out.

The problem of developing an extrinsic calibration tool for these two sensors can be put as identifying the rigid transformation between the sensor frames. In order to test and visually validate the output from the calibration tool the initial task was to overlay the image from the USB camera with the LIDAR points. In other words, the LIDAR provides a dataset of 2D points that needs to be accurately projected in the camera image. This option can be easily used for visual debugging during the development of the calibration tool and in the later stages kept as an additional option in the GCS for visualization of the LIDAR points.

## 3.2 Requirements

As the calibration tool is intended to be implemented as a feature in Pro-Drone's UAV control system, its development is tailored for the custom setup. The sensors to be extrinsically calibrated are the monocular USB camera and the 2D LIDAR. These sensors are already a part of Pro-Drone's custom payload for blade inspection and are the best fit for the company when it comes to cost, technical compatibility and quality of data.

The usual distance of the drone to the blade in Pro-Drone's inspections is 7-12 m. In order to get satisfactory results from the calibration, it was agreed that the maximum allowed error after validation is 0.2 m at the distance of 12 m. This corresponds to approximately 1° uncertainty when determining the orientation of camera relative to LIDAR.

The calibration tool needs to be implemented as a ROS package in Pro-Drone uav_control workspace and written in the C++ programming language. ROS is an open-source collection of software frameworks for robot software development and provides libraries and tools to help software developers create robot applications [27]. Pro-Drone uses the Jade distribution of ROS [27] as a framework for software development and Ubuntu 14.04 Linux distribution [27] as an operating system since ROS Jade is compatible and supported by Ubuntu 14.04.

In order for the calibration to be quick the runtime should be less than 2 minutes. In addition the calibration procedure needs to be simple and easy to perform so that it can be done by licensees in the future. This means that a software tool with a proper setup and configuration guide needs to be implemented. The calibration should be as flexible as possible and should not depend on a specific distance and orientation of the calibration target. This would make the calibration process easy and fast to perform. Also, the calibration target should be the shape and size that allows easy shipment with the rest of the payload to the customers.

## 3.3 Limitations

This combination of the used sensors introduces its own limitations. For instance, the monocular camera does not give depth information for the captured image. The data provided by the LIDAR is a 2D laser scan which contains only the 2D coordinates of the LIDAR points which sets the need of establishing 2D-3D correspondences.

The computational performance of the computer is also a factor. The use of the ODROID-XU4 as an onboard computer limits the use of computationally expensive actions such as image processing or machine learning. Therefore, during the development the USB camera was connected directly to a separate laptop computer and the calibration node was run on it. The LIDAR was connected to the onboard computer and the LIDAR data was transmitted via wireless network to the laptop. This was done so that during the development the code does not have to be re-uploaded and re-compiled in the ODROID as this has proven to be quite time consuming. In addition, because the camera was connected directly to the laptop, the delay and loss of the quality of camera data if it was streamed via wireless network was avoided.

# 4 Proposed Solution

The proposed method for extrinsic calibration of a LIDAR and a USB camera relies on using a small calibration object with AR tags in the overlapping field of view of the sensors. A rough illustration of the system setup is shown in Figure 10. The system consists of three main components:

- Calibration object consisting of two AR tags
- Hokuyo UTM-30LX 2D LIDAR
- Camera (Logitech c920 HD Pro Webcam)

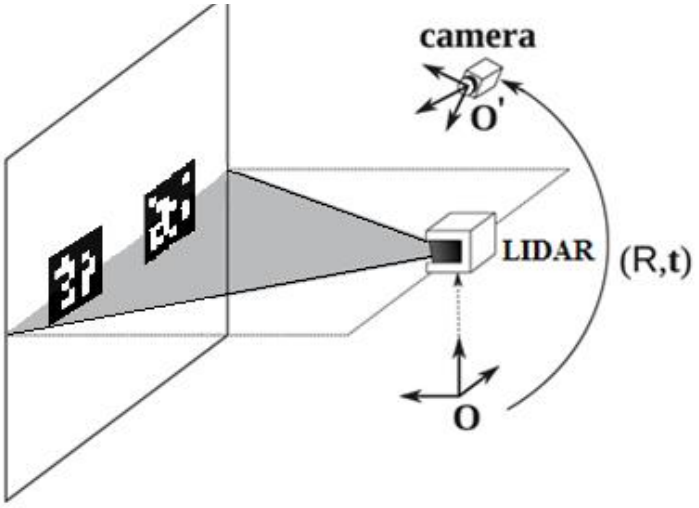



**Figure 10.** *System setup of a LIDAR, camera and a calibration target with AR tags.*

The estimation of the angles of orientation is done by using 2D projections of vectors onto a corresponding planes. The result of the calibration is a transformation matrix which is saved in a separate *.csv* file and can be later used by other applications that need to transform data from one coordinate system to another. The proposed solution is based on constraints that the LIDAR scan always falls on a specific place on the calibration object, the initial orientation of the camera in reference to the LIDAR is known and that only small (less than 10°) angular displacements are expected.

Since the displacement in the position of the camera is less likely to occur or if it does it is small enough not to be considered, the measured position of the camera in reference to the LIDAR is used as a temporary static translation and the focus of the calibration is shifted towards finding a way to calculate the orientation. Once the orientation is known, the frames can be aligned and the translation vector can be calculated by subtracting the coordinates of a known correspondence point.

# 4.1 Technical Solution

## 4.1.1 Coordinate Systems

The LIDAR data is expressed with x-y coordinates in the LIDAR 2D frame (x-forward, y-left) but can be expanded to a 3D frame (x-forward, y-left, z-up) by considering the z coordinates to be equal to zero. There are two camera frames of interest (Figure 11). The standard coordinate system (camera frame) of the pinhole camera is expressed as "z-forward, x-right, y-down". If the intrinsic parameters of the camera are known, the 3D points in camera frame can be projected onto the 2D image frame. The other camera frame is the camera object frame (x-forward, y-left, z-up). This is the frame in which the AprilTags software (covered in Section 4.1.3) retrieves the pose of the printed tags. The proposed solution for extrinsic calibration uses the camera object frame for estimation of the extrinsic parameters while the camera frame is used only for projection of 3D points onto the 2D image.

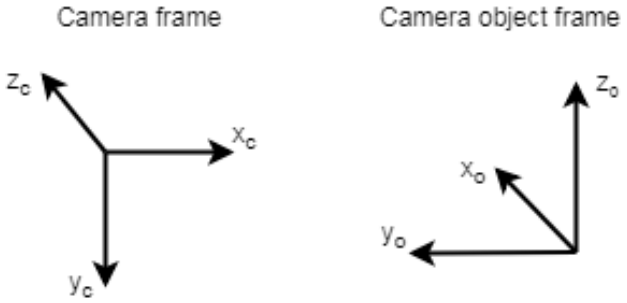



**Figure 11.** *Camera related frames.*

A rough illustration of the orientation of relevant camera coordinate frames in reference to the LIDAR frame in the sensor setup are depicted in Figure 12.

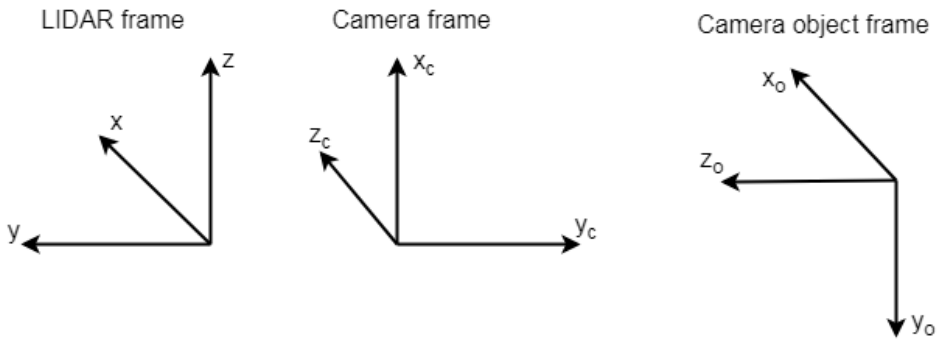



**Figure 12.** *Orientation of the camera and camera object frames in reference to the LIDAR.*

Figure 13 shows the transformation pipeline from left (LIDAR frame) to right (image frame) and is used to transform and project the LIDAR points onto the image. The mathematical relationship between the coordinates and coordinate systems is covered in Section 4.2.2.

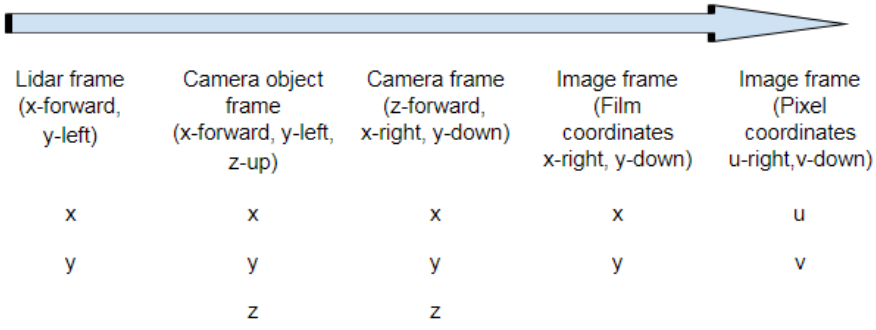
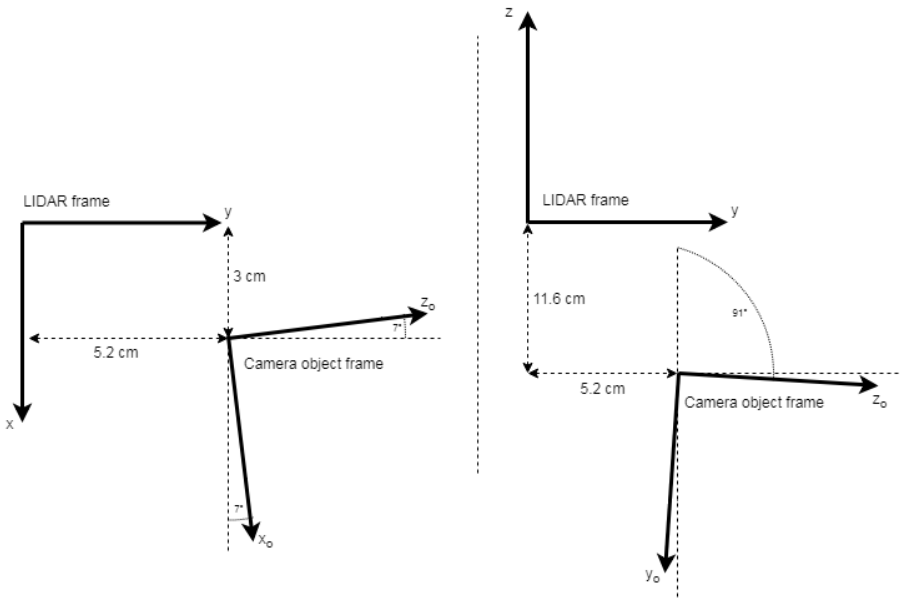



**Figure 13.** *Coordinate system transformation pipeline.*

### 4.1.2 System Setup

The quantified translation and rotation values between the LIDAR and camera object frame are given in Figure 14. These values were carefully measured using rulers and spirit levels and were later used for ground truth validation. The right-handed coordinate system convention is used to determine the rotations.




*(a)* *(b)*

**Figure 14.** *Top view (a) and front view (b) of the LIDAR and camera object coordinate frames*.

The calibration target is initially positioned around 1m (+/- 3cm) in front of the LIDAR (Figure 15). The orientation of the target is free as long as its front side is in the FOV of both sensors.

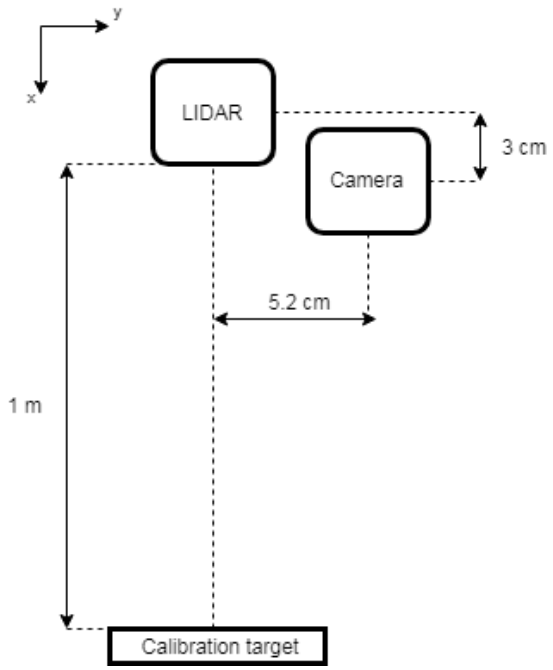



**Figure 15.** *Positioning of LIDAR, camera and calibration target in the system*.

### 4.1.3 Calibration Target

Even though automatic and targetless approaches are the main trend at the moment, the solution presented in this thesis uses a calibration target and is the best fit for the company when it comes to the current setup, hardware limitations and the available low density 2D LIDAR data.

The correspondence search can be defined as searching for corresponding geometrical elements such as points or lines in both LIDAR and camera modality. A calibration target with a specific shape can ease the process of identifying the such correspondences. The calibration presented in this thesis uses a flat square-shaped board with printed AR tags on it as a calibration target. The AR tags together with a specific image processing software are used to obtain 3D coordinates from objects in the scene since the monocular camera does not provide any depth information.

The board is 14×14 cm and it is attached on a camera tripod as shown in Figure 16. The tripod keeps the object fixed and allows it to be positioned with a millimeter precision in the FOV of the sensors at different distances and heights. The shape and size of the object and the printed tags was chosen to be easily detected by the sensors. There are 2 square tags printed on the calibration object (Figure 16). Each tag has dimensions 4×4cm and are at 2 cm distance from each other and the left, right and top edge of the calibration target. It was assumed that 2 tags are enough to form geometrical elements that can be used in the LIDAR-camera correspondence search.

Due to the office work space limitation, during the development and testing of the calibration tool, the distance between the sensors and the calibration object was always around 1 m since it was assumed that the error at greater distances can be easily calculated using the basic trigonometry.

**Figure 16.** *Calibration target with two tags.*

AprilTags (Figure 17) is a visual fiducial system, used in augmented reality, robotics, and camera calibration [28]. The AR tags are 2D markers that and can be easily detected by a camera. The AprilTags [29] library allows detection and full 6 DOF localization of the AR tags with their respective IDs from a single image frame [30]. AprilTags mainly depends on OpenCV [31] and Eigen3 [32] software libraries. There is also an ROS wrapper package for the AprilTags library (apriltags_ros [33]) that enables an implementation of the calibration tool as a ROS package in the future.

The chosen tags with tag ID's "12" and "21" are from the 36h11 [28] tag family and are shown in Figure 17.

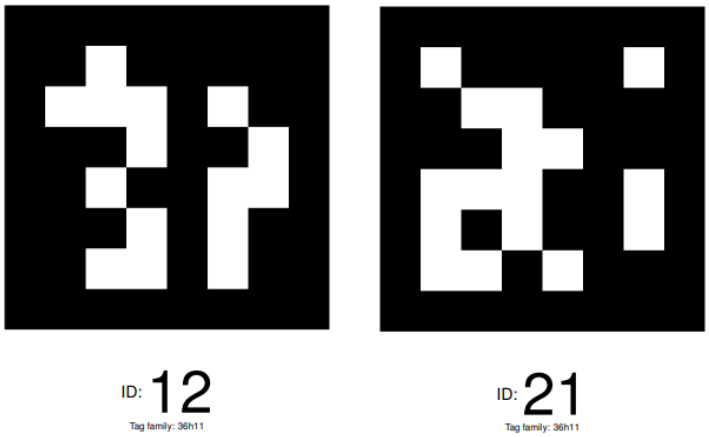



**Figure 17.** *AprilTags used in our setup and their ID's [34].*

AprilTags retrieves the 3D pose of the tags in the camera object frame and these two 3D poses are used as the camera correspondences in the camera-LIDAR correspondence search. To use the AprilTags library the intrinsic parameters of the camera and the physical size of the tags need to be *a priori* known.

### 4.1.4 Sensors

#### *LIDAR*

The 2D LIDAR used in the sensor setup is the Hokuyo UTM-30LX (Figure 18).

**Figure 18.** *Hokuyo UTM-30LX Scanning Laser Rangefinder [35].*

Based on the datasheet [36], the Hokuyo UTM-30LX Scanning Laser Rangefinder is a small, accurate, high-speed scanning device for robotic applications. This model uses USB 2.0 interface for communication and can obtain measurement data in a wider field of view up to a distance of 30 meters with millimeter resolution in a 270º arc. Due to the internal filtering and protective housing, this device can be used outdoors and is less susceptible to ambient light. Its low power consumption, allows the scanner to be used on battery-operated platforms. These benefits made this LIDAR a suitable choice for Pro-Drone since its angle coverage is enough for horizontal object detection and the system is always used outdoors and runs on the drone batteries. The sensor specifications are shown in Table 1.

| Model No. | UTM-30LX |
|---|---|
| Power source | 12VDC±10%(Current consumption: Max:1A, Normal: 0.7A) |
| Power consumption | Less than 8W |
| Light source | Laser Semiconductor λ = 870nm , Laser Class 1 |
| Detection Range | Guaranteed Range: 0.1 to 30m<br>Max Range: 60m |
| Scan angle | 270° |
| Accuracy | 0.1 to 10m:±30mm, 10 to 30m:±50mm*1 |
| Angular Resolution | 0.25° (360°/1,440 steps) |
| Scan Time | 25msec/scan |
| Interface | USB2.0(Full Speed) |
| Weight | Approx. 370g(with cable attachment) |
| External Dimension (W×D×H) | 60mm×60mm×85mm |

**Table 1.** *Hokuyo UTM-30LX Specifications.*

## *Camera*

The monocular USB camera used in the sensor setup is the Logitech c920 HD Pro Webcam (Figure 19).


**Figure 19.** *Logitech c920 HD Pro Webcam [37].*

The webcam allows full HD 1080p recording [38]. It uses the USB 2.0 protocol and supports H.264 encoding which allow the video to be streamed over a wireless network. H.264 is a lossy format and takes a lot of the CPU usage for decompressing which can be avoided with hardware decoding. The camera specifications are given in Table 2.

| Model No. | HD PRO WEBCAM C920 |
|---|---|
| Interface | USB 2.0 |
| Diagonal Field of View | 78° |
| Focal Length | 3.67 mm |
| Video Capture (16:9 W) | 360p, 480p, 720p, 1080p |
| Frame Rate (max) | 1080p@30fps |
| Weight | 162g |
| External Dimension (W×D×H) | 94mm×24mm×29mm |

**Table 2.** *Logitech HD Pro Webcam c920 Specifications*.

## 4.2 Calibration Method

The intrinsic camera calibration is a prerequisite for the extrinsic calibration process and provides the intrinsic parameters of the camera. These parameters are used in both the extrinsic calibration and the projection of LIDAR points onto the image. The extrinsic calibration algorithm is divided in three parts:

- Calibration target detection
- Data acquisition and filtering
- Transformation matrix calculation

The result is a transformation matrix that can be used for projection of LIDAR points onto the live video feed or by any other sensor fusion applications. The overall flow of the calibration pipeline is presented in Figure 20.

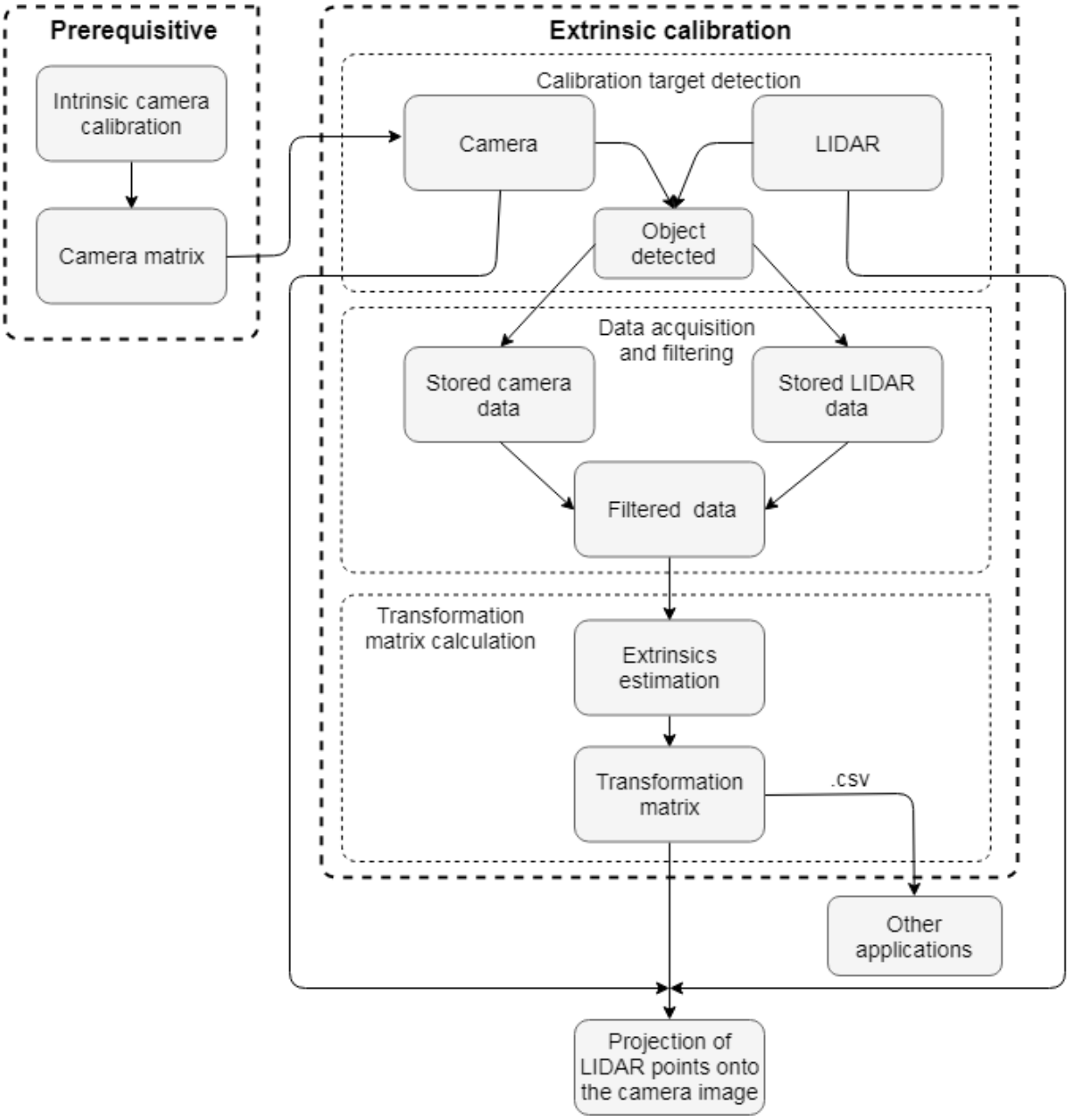



**Figure 20.** *Diagram showing the calibration pipeline.*

### 4.2.1 Intrinsic Camera Calibration

The intrinsic calibration is used to estimate the linear intrinsic parameters of the camera such as focal length (distance between the lens and the image sensor), principal point (the intersection of the optical axis with the image plane), skew and scale parameters.

The intrinsic calibration was done using the camera_calibration package [39] from ROS. The output of a successful camera calibration is the camera matrix that can be used to correctly localize the AR tags and project points from camera frame (3D) to image frame (2D). The camera calibration also outputs the non-linear intrinsic parameters in a form of a distortion matrix but since it was assumed that there is no significant lens distortion they were not considered. Once the intrinsic parameters of the camera are obtained, it can be assumed that they will never change.

The camera matrix has the following form (Eq 1):

$$camera\ matrix = s\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_x & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where:

$f_x,\ f_y$ are the focal length values in pixels,

$c_x,\ c_y$ optical center values in pixels,

$s$ is the scaling factor.

To calibrate the USB camera, a 8x6 checkerboard with 4,75 cm squares was used as intrinsic calibration target as shown in Figure 21.

As a result of intrinsic camera calibration, the following (Eq 2) camera matrix was obtained:

$$camera\ matrix = \begin{bmatrix} 497.077502 & 0.000000 & 328.680889 \\ 0.000000 & 495.736716 & 158.582483 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix} \quad (2)$$


**Figure 21.** *Monocular camera calibration using ROS camera_calibration package*.

### 4.2.2 Mathematical Model

The transformation of coordinates from LIDAR to camera object frame is done using the transformation matrix $T$ in Eq 3.

$$\begin{bmatrix} X_O \\ Y_O \\ Z_O \end{bmatrix} = T \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} \tag{3}$$

where:

$X_O,\ Y_O,\ Z_O$ are coordinates of a point in camera object frame,

$X_L,\ Y_L,\ Z_L$ are coordinates of a point in LIDAR frame,

$T$ is the transformation matrix.

The transformation matrix $T = [R|T_r]$ in 3D space is obtained from the rotation matrix $R$ and translation vector $T_r$:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = R_x R_y R_z \tag{4}$$

$$T_r = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{5}$$

where:

$t_x,\ t_y,\ t_z$ are the translation values,

$R_x,\ R_y,\ R_z$ are the rotation matrices around x, y, and z, respectively.

Using the right-hand rule, the rotation matrices $R_x$, $R_y$ and $R_z$ rotate vectors (3D points) by an angle θ about the x, y, or z axis respectively:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{6}$$

$$R_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{7}$$

$$R_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

AprilTags retrieves the coordinates of the tags in camera object frame. In order to be projected onto the image they need to be transformed from camera object to camera frame:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = R_{xz} \begin{bmatrix} X_O \\ Y_O \\ Z_O \end{bmatrix} \tag{9}$$

where:

$X_C$, $Y_C$, $Z_C$ are the camera coordinates,

$R_{xz}$ denotes the rotation around z and x ($R_x R_z$)axis for $\theta = 90°$.

Once we have the camera coordinates (Eq. 9) we can easily map them into 2D image coordinates by using the camera matrix (Eq. 2) obtained from the camera calibration:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_x & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \tag{10}$$

Where:

*u, v* are the image coordinates in the 2D image plane.

### 4.2.3 Calibration Target Detection

The LIDAR provides points from a 270° semicircular field laser scan. The LIDAR points that are of interest are those that fall onto the calibration target. Therefore, the rest of the points need to be filtered out. A top-down visualization of the laser scan and the LIDAR origin obtained form the visualization tool RViz can be seen in Figure 22. Here the size (radius) of the points was set to 5 cm for visualization purposes and the points from the calibration target can be seen along the x (red) axis of the LIDAR at a distance of 1 m from it.

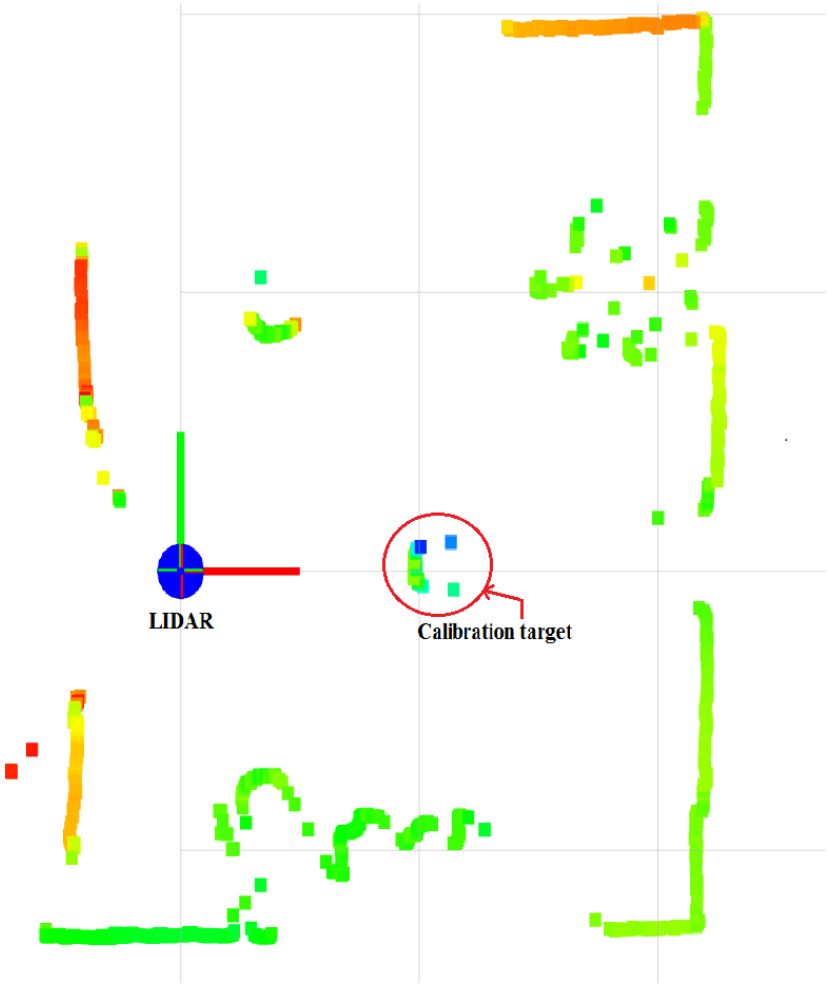



**Figure 22.** *Top view of the laser scan visualized in RViz with a grid size of 1 m.*

The calibration target detection by the LIDAR is done in several steps. First the LIDAR points are grouped into bodies (clusters) using ProDrone's LIDAR splitter library. Then the correct body is chosen based on the following checks:

- Distance of the body from the LIDAR (0.8 to 2 m)
- If the body is in the expected FOV (+/- 30° of the LIDAR x axis) of the sensors
- Size (length) of the body (14 cm +/- 1 cm tolerance)
- Flatness of the body (a check if the detected body is flat so it can be confirmed that it matches the flat surface of the calibration target)

The distance and size of the body, the angle defining the expected FOV and the tolerances are defined as body filtering parameters in the system and can be changed based on the calibration setup.

It is important to note that the "jumping" edge points of the body that can be seen in Figure 22 (two jumping points can be seen in the calibration target cluster) are filtered out to ease the target detection and remove potential outliers. The appearance of these jumping points on the edges of objects is common with LIDARs. This is due to the fact that when the laser beam falls on the edge only a part of the beam is reflected back causing a change in its intensity. In order to validate the reliability of LIDAR readings the object was positioned at different known distances in front of the LIDAR. It was determined that after removing a few outliers, the measurement uncertainty of LIDAR remained within 1 cm, which was considered sufficient for the task at hand.

Once the correct body has been segmented out a check if the two tags can be seen by the camera is done. The tag detection is done using aprilTags software. The aprilTags function that retrieves the pose of the tags uses the intrinsic parameters (focal length and principal point) from the camera calibration and the size of the tag (defined as a parameter).

If these conditions are met, the calibration object is in the FOV of the both sensors and the data acquisition starts.

### 4.2.4 Data Acquisition and Filtering

The data acquisition involves storing the poses of two LIDAR points in LIDAR frame and the poses of the two tags in camera object frame. The LIDAR points that are stored and used as LIDAR correspondences are the first and the last point of the filtered body after the jumping points on the edges have been removed. These two particular points were chosen because they are at the maximum distance from each other when compared to any other pair of points on the filtered object. This acts as primitive line fitting which minimizes the horizontal angle deviation of the line formed by a pair of points. The number of sequences (laser scans and image frames) recorded is set by a parameter with a default value of 100 laser scans and 50 image frames. The recorded data was visualized in order to analyze the coordinate distribution of the correspondence

points and detect outliers so that a suitable filter can be applied. Based on these visualizations (see Appendix) it was assumed that a simple median or a mean filter would be enough. Both filters were tried out but since no binary threshold was applied to remove outliers that are not considered as jumping points, the median filter was used in the end to remove the outliers caused of potential jumps. The output of the filter are the pairs of filtered LIDAR and tag poses.

For analysis and future optimization, the recorded LIDAR and tag coordinates together with the image frames were saved in a calibration folder on the laptop computer. This makes the data acquisition the most time consuming part of the calibration process taking 99% of the calibration time. The time is proportional with the amount of data recorded. For example, for 100 laser scans and 50 image frames, the calibration process takes roughly 15-17 seconds.

### 4.2.5 Extrinsic calibration algorithm

In order to find correspondences, the calibration object has to be adjusted in a way that we can assume with a certain confidence that the laser scan falls on a specific place on the calibration target such that it goes through the center of the printed tags. In this case, we can say that the centers of the tags and the filtered points of the laser scan lie on the same line and this line can be considered as a correspondence between the sensors (Figure 23). The assumption that the selected points lie on a common line was confirmed with careful measurement using rulers and spirit levels in combination with the laser scan output in RViz.

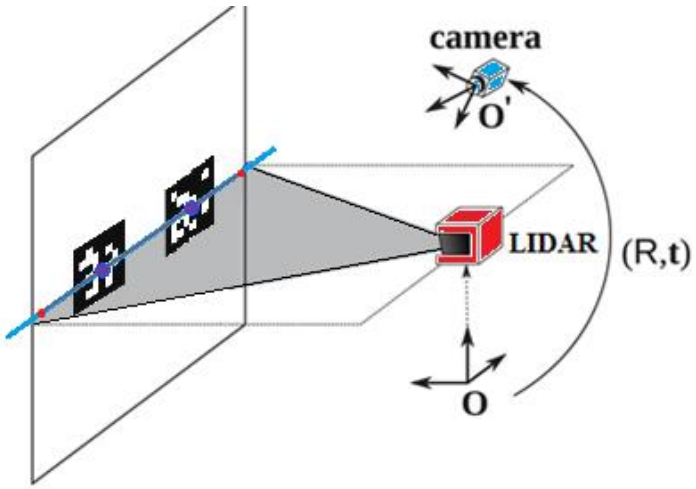



**Figure 23.** *Correspondence line formed tag centers (blue) and LIDAR points (red).*

The estimation of the orientation angles of the camera in reference to the LIDAR is achieved by projecting 3D vectors to a common 2D plane for each axis. These vectors are formed from the filtered LIDAR and tag points respectively as shown in Figure 24. The vectors orientation in reference to the LIDAR is in the following order:

- The vector formed from the LIDAR coordinates is pointing from the right (Point 1) to the left edge (Point 2) of the calibration target
- The vector formed from the tag coordinates is pointing from the center of the right (Tag 1) to the center of the left tag (Tag 2)

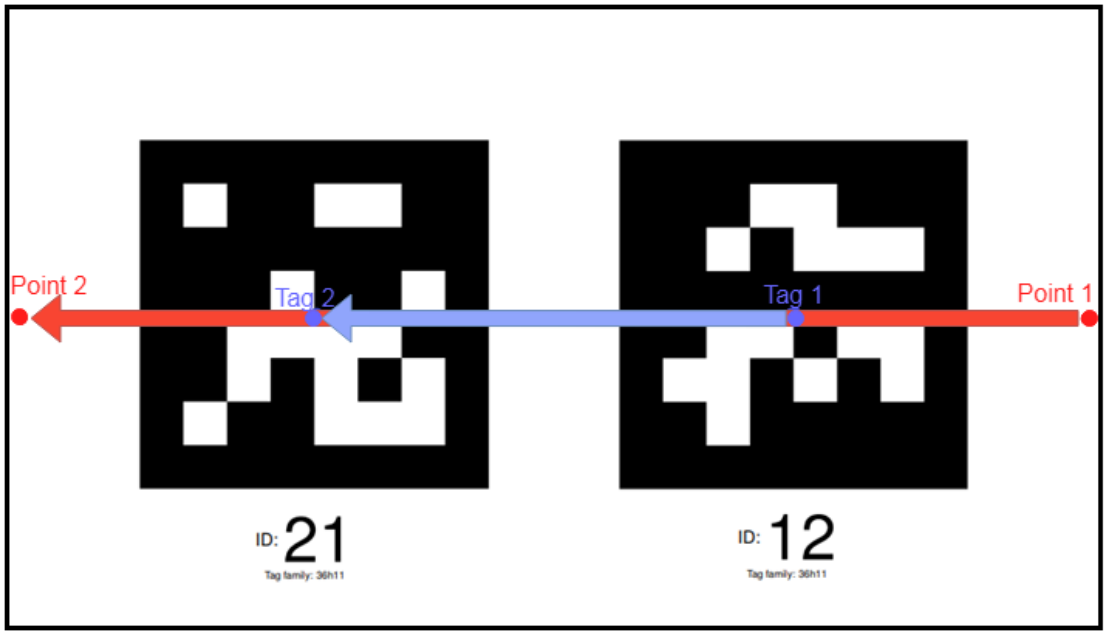



**Figure 24.** *LIDAR (red) and tag (blue) vectors formed from the filtered points.*

Based on the previously stated constraints that the overall orientation of the camera in reference to the LIDAR is known (Figure 25) and that only small angular displacements are expected common (corresponding) pair of planes between the LIDAR and camera object frame are defined:

- XY LIDAR plane corresponds to $X_oZ_o$ camera plane
- YZ LIDAR plane corresponds to $Y_oZ_o$camera plane
- XZ LIDAR plane corresponds to $X_oY_o$ camera plane

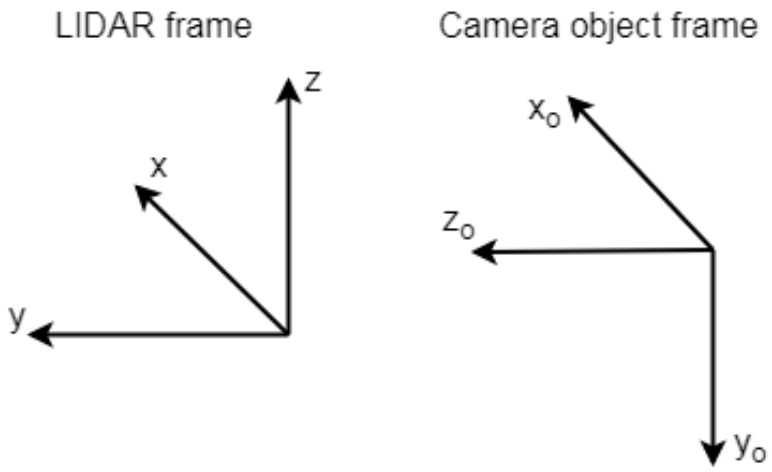



**Figure 25.** *Overall orientation of the camera in reference of the LIDAR frame*.

Once the corresponding planes are defined, 2D projections of the 3D vectors are done for every pair of corresponding planes and the angles of rotation are obtained by calculating the angle difference of the projected vectors.

Let's consider that A and B represent the initial (Point 1) and terminal (Point 2) points of the projected LIDAR vector while C and D represent the initial (Tag 1) and terminal (Tag 2) points of the projected camera vector (Figure 24). If the coordinates of the initial and terminal points are known then the coordinates of a vector can be calculated using Eq. 11:

$$\begin{aligned} \overline{AB} &= \{B_x - A_x; B_y - A_y\} = \{AB_x; AB_y\} \\ \overline{CD} &= \{D_x - C_x; D_y - C_y\} = \{CD_x; CD_y\} \end{aligned} \tag{11}$$

In the case of a plane the dot product of two vectors can be calculated using Eq. 12:

$$\overline{AB} \cdot \overline{CD} = AB_x \cdot CD_x + AB_y \cdot CD_y \tag{12}$$

The vector length (magnitude) is calculated using Eq. 13:

$$\begin{aligned} |\overline{AB}| &= \sqrt{{AB_x}^2 + {AB_y}^2} \\ |\overline{CD}| &= \sqrt{{CD_x}^2 + {CD_y}^2} \end{aligned} \tag{13}$$

Finally, the angle between the vectors is obtained from Eq. 14:

$$\cos\alpha = \frac{\overline{AB} \cdot \overline{CD}}{|\overline{AB}||\overline{CD}|} \tag{14}$$

The assumption that there is no difference in the orientation of both LIDAR and camera frames (in case of perfect alignment) implies that there also should not be any offset (except small and negligible one due to imperfect sensor readings) in the angles of the projected 2D vectors. This means that if the both frames have the same orientation then the vectors would also have the same orientation which implies that they are parallel and the angle difference should be a value near to zero. However, the coordinate frames do not have the same orientation as it was stated in Sections 4.1.1. and 4.1.2 so the camera angles of rotation in reference to the LIDAR are estimated using the abovementioned projections.

For example, the angle of rotation of the camera around the z axis (yaw) is the calculated angle difference between the 2D projections of the LIDAR vector in the XY LIDAR plane and the tag vector in the $X_oZ_o$ camera plane. An illustration of the projected vectors, their unit vectors and the angle between them is shown in Figure 26. The projections of the vectors formed from the LIDAR points and the centers of the tags and their unit vectors are displayed in red and blue respectively.

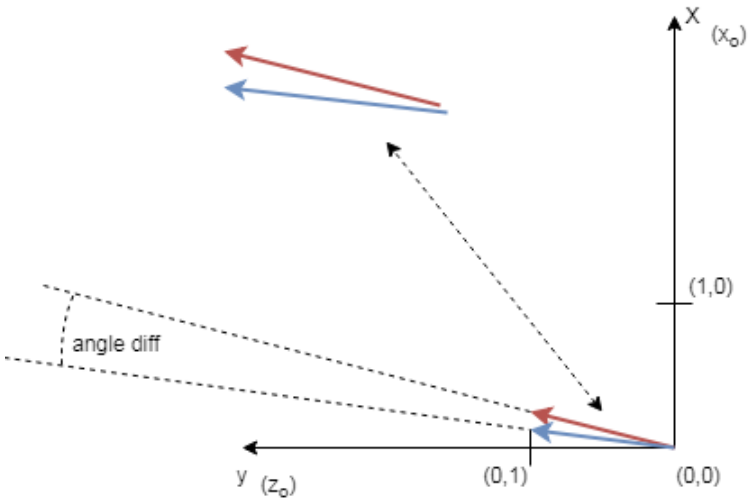



**Figure 26.** *Projection of LIDAR (red) and tag (blue) vectors, their respective unit vectors and the angle difference in the corresponding XY* $(X_oZ_o)$ *plane.*

Similar calculation is conducted for the angle of rotation of the camera around the x-axis (roll). Here the vectors are projected in the YZ plane of the LIDAR frame. The corresponding camera object plane is the $Y_oZ_o$ plane in this case. The vector projection, their unit vectors and the expected angle difference is shown in Figure 27.

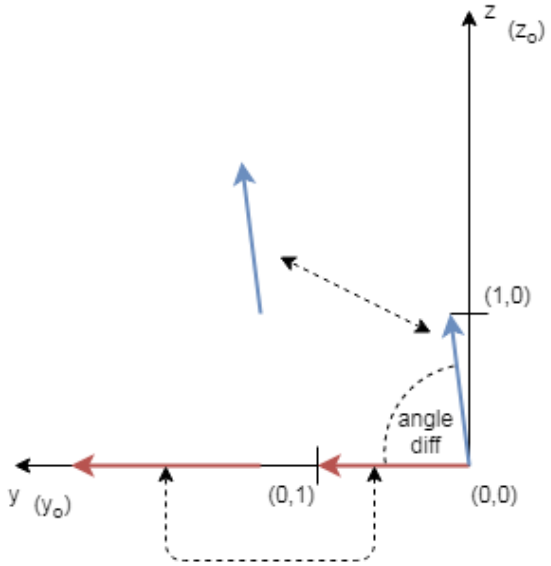



**Figure 27.** *Projection of LIDAR (red) and tag formed (blue) vectors, their respective unit vectors and the angle difference in the corresponding YZ* $(Y_oZ_o)$ *plane.*

The same method is used to calculate the angle of rotation around the y-axis (pitch). Here the tag points are projected onto the XZ plane. However, this angle is the hardest one to calculate using the abovementioned projections since the difference between the x and y coordinates of the recorded tag points is significantly lower (usually less than 2 mm for y) than in the previous cases. The angle calculation is sensitive to even minor fluctuations in the recorded values. The error of the calculated angle is even greater when the calibration object is positioned parallel to the LIDAR's y-axis. This is due to the fact that the difference between the x coordinates of the tags is smaller than when the object is positioned at an angle. Since in the current sensor setup the measured pitch is close to zero (0.2°), a temporary constraint that there is no rotation around the y axis was introduced and this angle was not considered in the calculation of the transformation matrix.

Once the rotation matrix has been formed the camera and LIDAR frames are aligned and the translation vector can be calculated by subtracting the appropriate coordinates of a new specific correspondence point. This point has to be seen in both the LIDAR and camera frame simultaneously. From the current setup it was assumed that the midpoint (Figure 28) between the filtered tag points corresponds with the midpoint of the filtered LIDAR points and can be used as a correspondence point for the translation calculation.

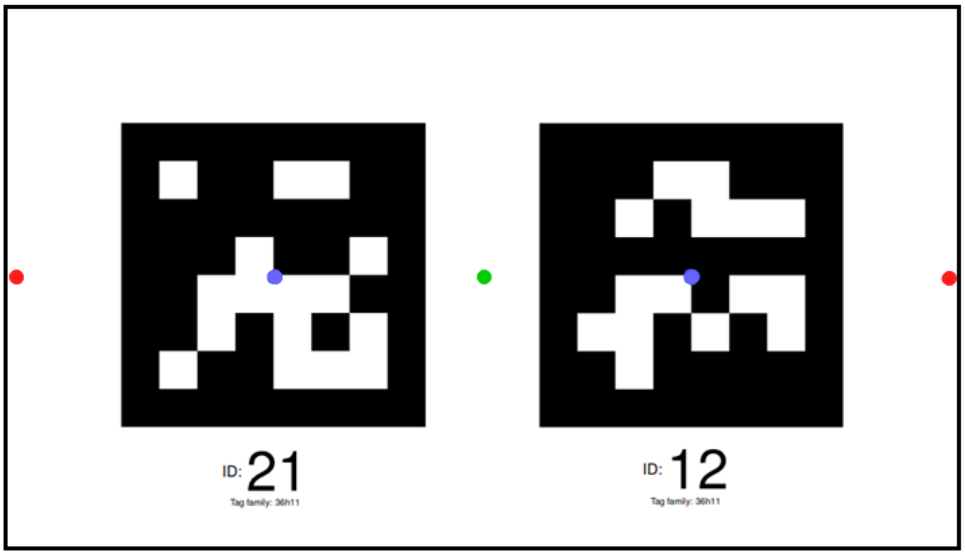



**Figure 28**. *Midpoint (green) between tag (blue) and LIDAR points (red) used as correspondence point for translation calculation.*

# 5 Experimental Results and Analysis

Several experiments were carried out to test the performance of the proposed extrinsic calibration algorithm. In order to determine if/how the position of the calibration target and the amount of recorded data influence the estimation of the orientation angles four test cases were conducted. In addition to them an experiment using a bigger version of the calibration target is also presented. The validation of the results together with a comparative analysis of all test cases is given in the end.

## 5.1 Parallel positioning

### 5.1.1 Test case 1

The calibration target was initially positioned 1 m along the x axis and parallel to the y axis of the LIDAR as shown in Figure 29. A total of 300 LIDAR scans and 50 camera frames were recorded. More information and visualizations of the coordinate data distribution and filtered coordinate values can be found in Appendix 1.

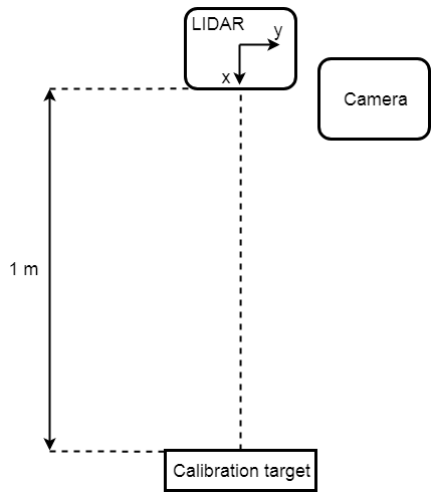



**Figure 29.** *Top-down view of the system setup for Test case 1 of parallel positioning.*

In this case the following angles of rotation were computed using Eq. 14:

Roll: -92.911°

Pitch: -15.255°

Yaw: 7.934°

The visual output showing the projection of LIDAR points on the image is shown in Figure 30. The angle around the y axis (pitch) was not considered in the projection of the LIDAR points.

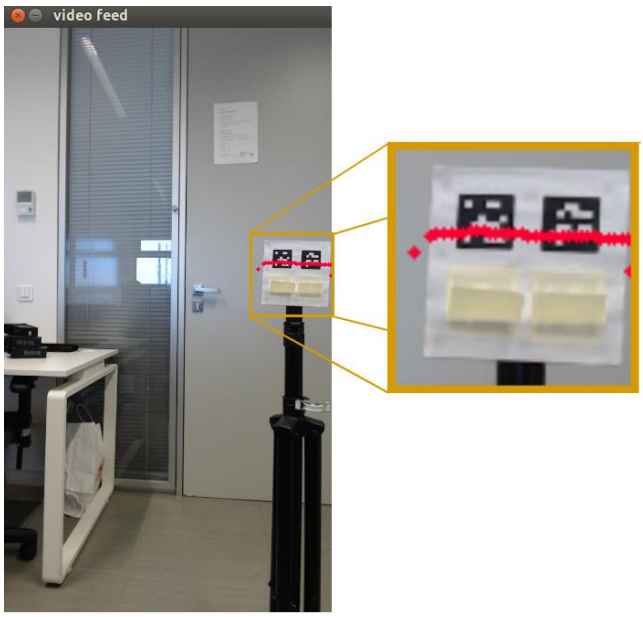



**Figure 30**. *Projection of LDIAR points in Test case 1 of parallel positioning*.

### 5.1.2 Test case 2

In another case of parallel positioning, the calibration object was only moved along the y-axis (~ 14 cm) in reference of the LIDAR (Figure 31). A total of 100 LIDAR scans and 50 camera frames were recorded. More information and visualizations of the coordinate data distribution and filtered coordinate values can be found in Appendix 2.

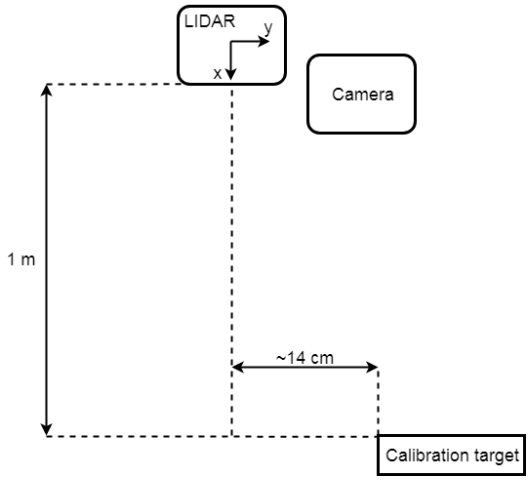



**Figure 31.** *Top-down view of the system setup for Test case 2 of parallel positioning.*

In this case the following angles of rotation were computed:

Roll: -91.941°

Pitch: 161.56°

Yaw: 7.834°

The visual output showing the projection of LIDAR points onto the image is shown in Figure 32. The angle around the y axis (pitch) was not considered in the projection of the LIDAR points.

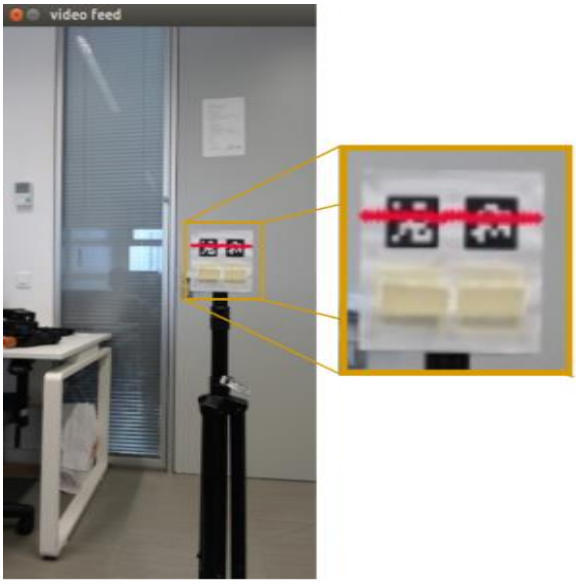



**Figure 32**. *Projection of LDIAR points in Test case 2 of parallel positioning.*

## 5.2 Angled positioning

### 5.2.1 Test case 1

In this test case the calibration target was positioned 1 m along the x-axis of the LIDAR at an angle of around 55° (Figure 33). A total of 1000 LIDAR scans and 500 camera frames were recorded. More information and visualizations of the coordinate data distribution and filtered coordinate values can be found in Appendix 3.

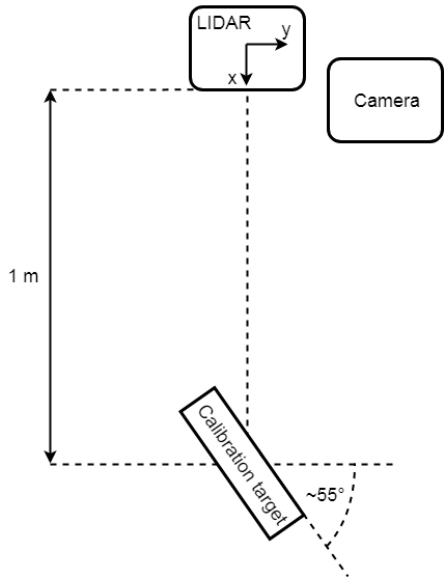



**Figure 33.** *Top-down view of the system setup for Test case 1 of angled positioning.*

In this case the following angles of rotation were computed:

Roll: -93.945°

Pitch: -2.121°

Yaw: 6.608°

The visual output showing the projection of LIDAR points onto the image is shown in Figure 34. The angle around the y axis (pitch) was not considered in the projection of the LIDAR points.

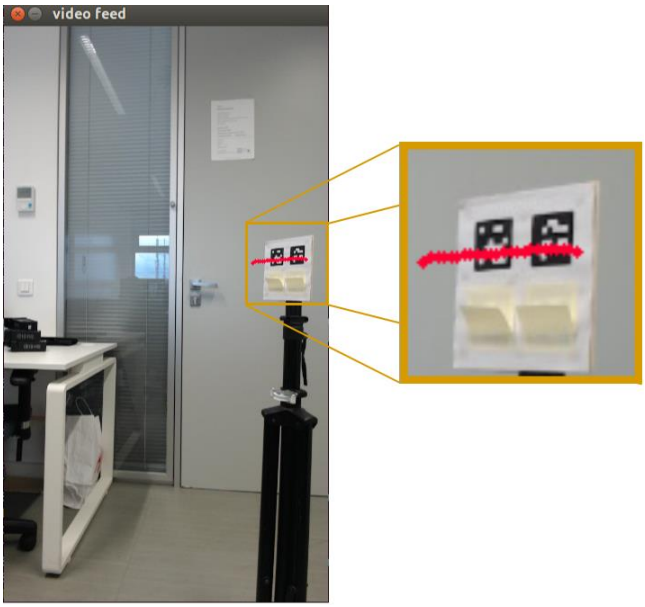



**Figure 34**. *Projection of LDIAR points in Test case 1 of angled positioning.*

### 5.2.2 Test case 2

In another test case the calibration target was positioned again around 1 m along the x axis of the LIDAR but at a smaller angle of around 10° in the opposite direction then the case presented in Section 5.2.1 (Figure 35). A total of 1000 LIDAR scans and 500 camera frames were recorded. More information and visualizations of the coordinate data distribution and filtered coordinate values can be found in Appendix 4.

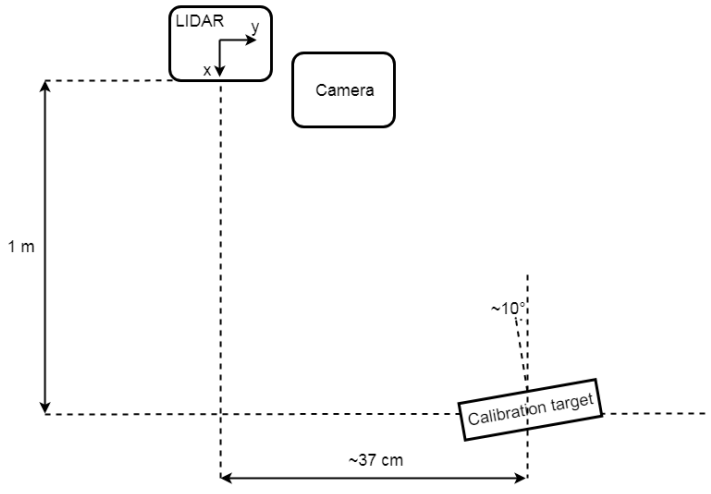



**Figure 35.** *Top-down view of the system setup for Test case 2 of angled positioning.*

In this case the following angles of rotation were computed:

Roll: -91.941°

Pitch: 18.435°

Yaw: 7.729°

The visual output showing the projection of LIDAR points onto the image is shown in Figure 36. The angle around the y axis (pitch) was not considered in the projection of the LIDAR points.


**Figure 36**. *Projection of LDIAR points in Test case 2 of angled positioning*.

## 5.3 Bigger calibration object

In one instance a bigger calibration object (Figure 37) was used to test the calibration algorithm. The object was designed to have a similar shape to the previously used target with a change in the object width and tag size. The width was increased to 65 cm and tags with a size of 16x16 cm were used. The object was positioned around 1.3 m along the x axis of the LIDAR at an angle of around 30° (Figure 38). A total of 100 LIDAR scans and 50 camera frames were recorded.

In this case the following angles of rotation were computed:

Roll: -91.916°

Pitch: 3.126°

Yaw: 6.925°

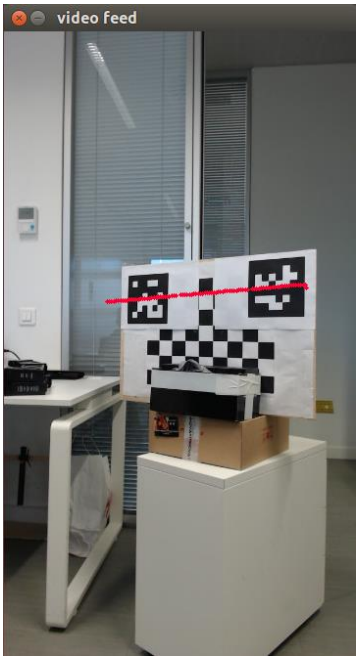



**Figure 37**. Bigger calibration object

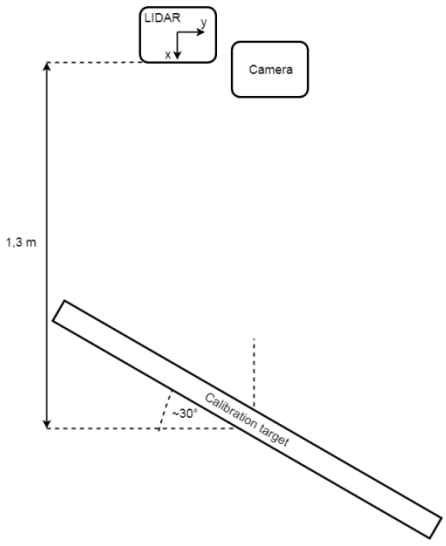



**Figure 38.** *Top-down view of the system setup with a bigger calibration target*

## 5.4 Validation

The result of the calibration is obtained in the form of a transformation matrix and can be used to project the LIDAR points on the live camera feed (Figure 39).

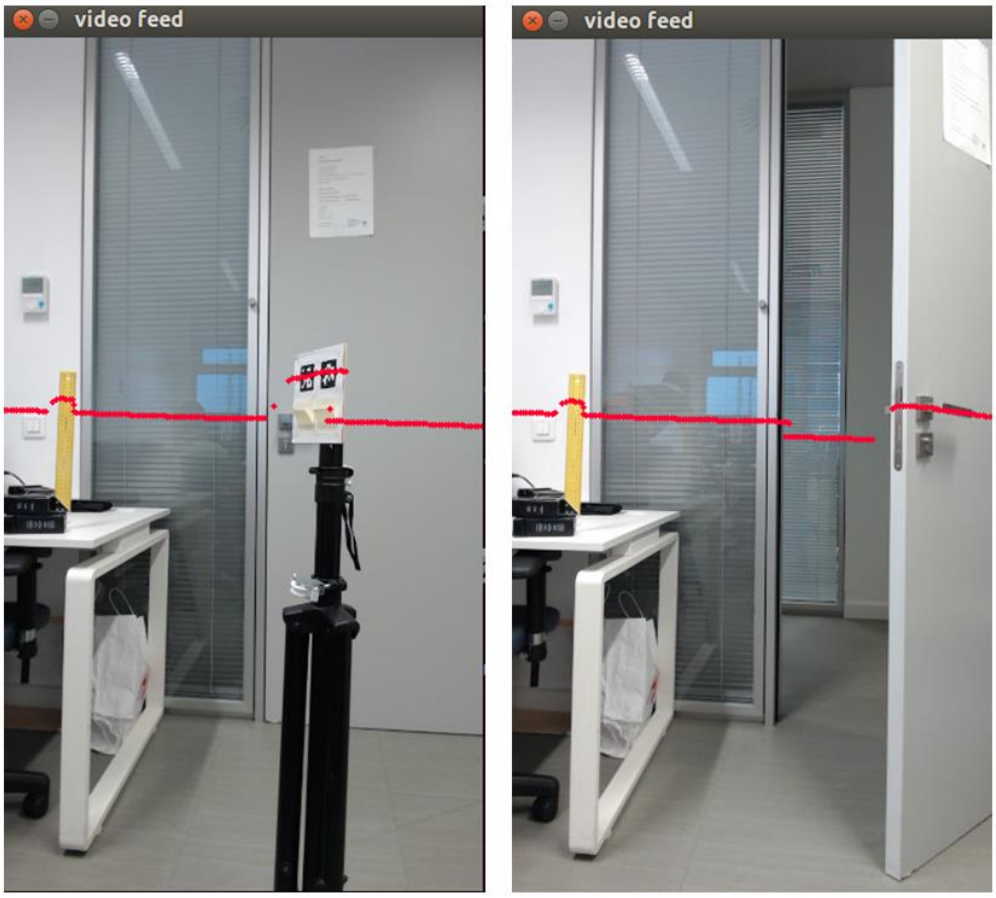



**Figure 39.** *Visual validation of the projected laser scan on the camera video feed using the computed transformation matrix*

In addition to the visual validation, the accuracy of the result for each test case was validated with the following ground truth measurements of the orientation angles:

Roll: -91°

Pitch: -0.2°

Yaw: 7°

The rotation errors of the estimated angles are given in Table 3.

| | Parallel positioning | | Angular positioning | | Big calibration target |
|---|---|---|---|---|---|
| | Test case 1 | Test case 2 | Test case 1 | Test case 2 | |
| Roll [deg] | -1,911 | -0,941 | -2,945 | -0,941 | -0,916 |
| Pitch [deg] | -15,055 | 161,760 | -1,921 | 18,635 | 3,326 |
| Yaw [deg] | 0,934 | 0,834 | -0,392 | 0,729 | -0,075 |

**Table 3.** *Rotation errors in degrees for different test cases*

## 5.5 Comparison

The comparison shows that that the size and orientation of the calibration object in reference to the sensors influences the accuracy of the calibration. From Table 3 we can see that in all test cases the error of the estimated roll is less than 3° while the error in the estimated yaw is less than 1°. Both test cases of parallel positioning and test case 2 of angular positioning where the angle of the calibration target is around 10° give greater errors (-15.055°, 161.760° and 18.635° respectively) in the estimation of the pitch. This is due to the sensitivity of the calculation when the coordinate difference is significantly lower for a parallel positioning of the calibration target. The test case where a big calibration object is used gives the most accurate estimation for the roll and yaw angles while test case 1 of angular positioning gives the most accurate estimation of the pitch when compared with the other cases. From this it can be assumed that optimal results can be obtained if the big calibration target is used and positioned at an angle of 55° as in test case 2. The angled positioning improves the estimation of the pitch because in this case the difference in the x coordinates when the object is placed at an angle is increased. The bigger calibration object also increases the coordinate distance between the tags/LIDAR points which makes the calculation less sensitive to fluctuations in the data.

Large amount of recorded data has a small influence on the results of the calibration when a mean or a median filtered is used. This is due to the micrometric difference in the filtered coordinates when different amount of data is considered. However, small amount of data is subjected to noise caused by short illumination changes in the environment or jumps in the laser scan. Therefore a minimum amount of scans/frames to be recorded needs to be set. For example, in test case 1 of

angular positioning (Section 5.2.1) 1000 laser scans and 500 camera frames were used in the median filter. A comparison showing how the absolute error of the rotation angles changes for different amounts of recorded data (scans/frames) is shown in Figure 40.

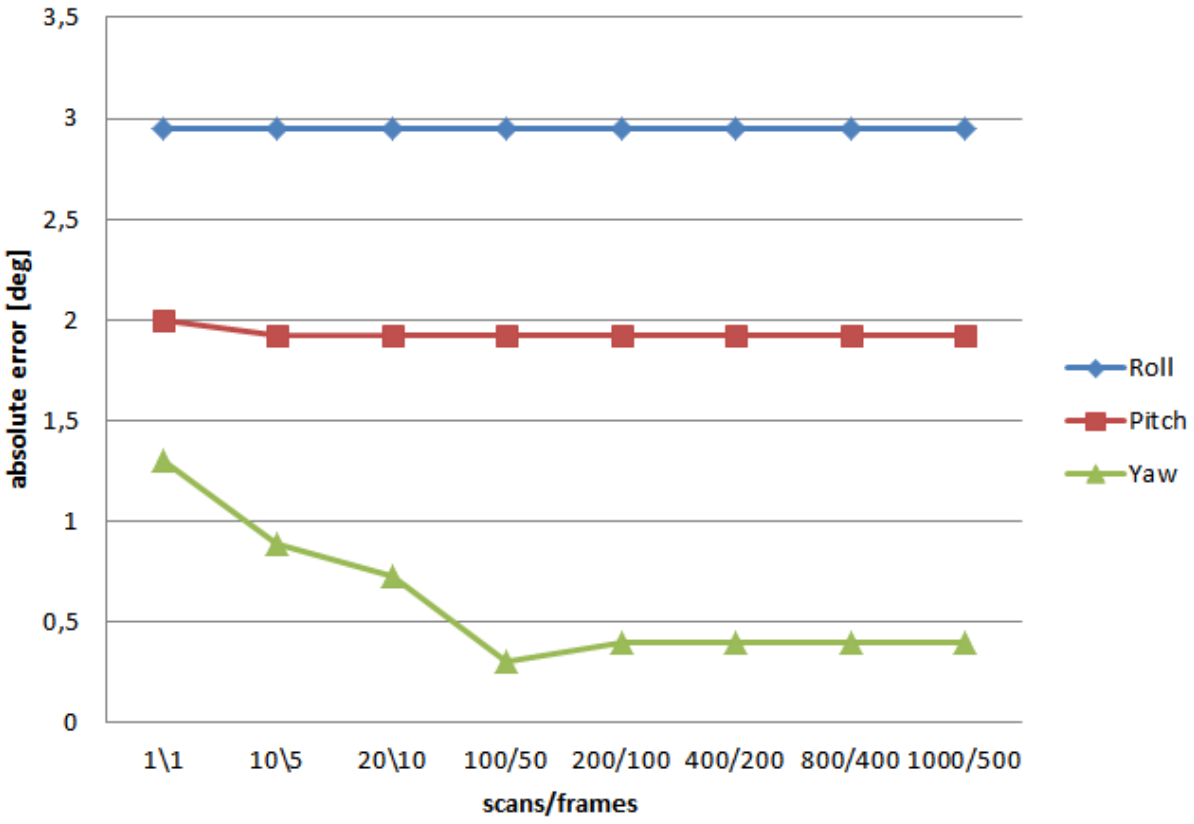



**Figure 40.** *Absolute errors for the rotation angles for different amounts of recorded data for Test case 1 of angular positioning*

From Figure 40 it can be concluded that the minimum amount of recorded data should be in the interval of 100/50 to 200/100 scans/frames. After this interval, there is no improvement in the estimated results.

# 6 Future Work

As the calibration tool is still work in progress at Pro-Drone, the current state can be improved in few instances. For example, an improvement of the calibration object that would ease and speed up the process of correspondence positioning can be done. Figure 41 shows such concept design that has extended handles on the both sides. The calibration object detection explained in Section 4.2.3 can be adjusted to detect this body only when the laser scan goes through these handles. This would mean that if the body is detected then it can be assumed that the line passes through the tag centers as well. The size of the object and tags can also be increased as it was shown in Section 5.3 and discussed in Section 5.5.

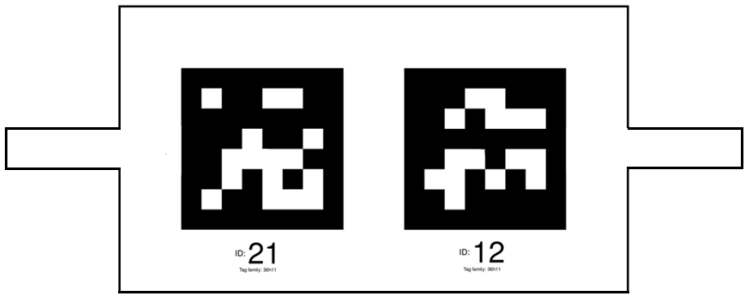



**Figure 41**. Concept design of a calibration object with handles

Since the calibration performance depends on the accuracy of the recorded data the filtering must be improved. At the moment only a basic median filter is applied. A comparison showed that a mean filter produces a similar result. In addition, RANSAC (random sample consensus) outlier detection method and other line fitting algorithms can be implemented to get a better estimate of recorded data.

The implementation of the translation calculation is still to be done although it is assumed to be a simple calculation of the difference between two aligned frames.

Once the rotation and translation matrices are obtained, the projection of the camera 3D points onto the image can be improved by implementing the OpenCV **projectPoints** [31] function in the

projection pipeline. The function computes projections of 3D object points to the image plane given intrinsic and extrinsic camera parameters.

Another extrinsic calibration algorithm that can be easily implemented by making changes in the correspondence search is the Perspective-n-Point (PnP). OpenCV provides this algorithm in the **solvePnP** [31] function that solves the problem of estimating the pose of a calibrated camera given a set of 3D points in a model coordinate system and their corresponding 2D projections in the image.

Finally an implementation of a graphic user interface (GUI) is needed to allow the users to interact with the calibration tool. The GUI should also have visual indicators that will notify the user when the calibration target is in position and correspondences have been found so that he can start the data acquisition and visualize the results of the calibration.

# 7 Conclusion

The increased usage of LIDARs and cameras in robotic systems and applications that seek to fuse the sensor data introduces the need for an extrinsic calibration of the sensors. In this thesis, a customized robust solution for an extrinsic calibration of a LIDAR and a monocular camera is presented. The algorithm for computing the extrinsic parameters requires a calibration target and uses basic geometry to calculate the pose of the camera in respect to the LIDAR. The work acts as a proof of concept that the extrinsics can be roughly estimated in a custom calibration setup assuming that the sensor data is accurate. The software of the calibration tool is written in the C++ programming language and implemented as a ROS package. The presented work is still a project under development at the company Pro-Drone and is intended to be improved and used as a part of their UAV control system. At its current state, the calibration process is fast and simple to perform and does not require any previous knowledge of calibration procedures.

# 8 Acknowledgement

I would like to thank Pro-Drone for giving me the internship opportunity and for providing me with support and necessary hardware for the calibration. It was a great professional experience and pleasure to work in an international and flexible startup environment. I would like to express my sincere gratitude to my supervisors from Pro-Drone, Ilia Sheremet (Head of Robotics) and Tiago Ferreira (Head of Engineering) for their support during my internship. I would also like to thank my mentor, Assoc. Prof. Karl Kruusamäe his assistance and advice throughout the preparation of the thesis. Furthermore, I would like to credit University of Tartu as well for the courses that prepared me for the professional practice and Erasmus+ programme for the traineeship grant.

# References


[1] R. Decoret, “Why inspect wind turbine blades?”, *3E Technical Field Note*, pp. 1–2, 2014.

[2] Bladena, Vattenfall, EON, Statkraft and KIRT x THOMSEN, “INSTRUCTION Blade Inspection”*, EUDP Project*, rev. 1.0, 2018.

[3] P. Dvorak, “Self-flying drones and wind turbine blades: Inspections in about 15 minutes” [Online]. Available: https://www.windpowerengineering.com/tools-and-equipment/self-flying-drones-wind-turbine-blades-inspections-15-minutes/.

[4] “Pro-Drone” [Online]. Available: http://www.pro-drone.eu/.

[5] J. Li, X. He, and J. Li, “2D LiDAR and Camera Fusion in 3D Modeling of Indoor Environment”, pp. 379–383, 2015.

[6] C. Premebida, O. Ludwig, and U. Nunes, “LIDAR and Vision-Based Pedestrian Detection System”*, Journal of Field Robotics*, vol. 26, no. IV, pp. 696–711, 2009.

[7] A. Sareen, C. A. Sapre, and M. S. Selig, “Effects of leading edge erosion on wind turbine blade performance”, *Wind ENERGY*, pp. 1531–1542, 2014.

[8] MHI Vestas Offshore Wind, “V164-8.0 MW breaks world record for wind energy production”*,* 2014.

[9] “SkySpecs” [Online]. Available: https://skyspecs.com/.

[10] A. Moura, “ProDrone promise of UAV inspections.”

[11] “Calibration” [Online]. Available: https://en.wikipedia.org/wiki/Calibration.

[12] S. Sim, J. Sock, and K. Kwak, “Indirect Correspondence-Based Robust Extrinsic Calibration of LiDAR and Camera”, *Sensors*, 2016.

[13] “MathWorks” [Online]. Available: https://se.mathworks.com/help/vision/ug/camera-calibration.html.

[14] F. Vasconcelos, J. P. Barreto and U. Nunes, “A Minimal Solution for the Extrinsic Calibration of a Camera and a Laser-Rangefinder”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. *34,* no. 11, pp. 2097 - 2107, 2012.

[15] V. Niola, C. Rossi, S. Savino, and S. Strano, “A method for the calibration of a 3-D laser scanner”, *Robotics and Computer-Integrated Manufacturing,* vol. 27, no. 2, pp. 479–484, 2011.

[16] Z. Chen, L. Zhuo, K. Sun, and C. Zhang, “Extrinsic calibration of a camera and a laser range finder using point to line constraint”, *Procedia Engineering*, vol. 29, pp. 4348–4352, 2012.

[17] G. Pandey, J. R. Mcbride, S. Savarese and R. M. Eustice, “Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information”, *Journal of Field Robotics*, vol. 1, no. 1, pp. 1–27, 2014.

[18] “Velodyne” [Online]. Available: http://velodynelidar.com/index.html.

[19] “Hokuyo 2d” [Online]. Available: https://www.hokuyo-aut.jp/search/single.php?serial=169.

[20] “Kinect Robotics” [Online]. Available: https://spectrum.ieee.org/automaton/robotics/diy/top-10-robotic-kinect-hacks.

[21] J. Knight, “Robot Navigation by Active Stereo Vision”, *OUEL Technical Report No OUEL 2220/00*, 2000.

[22] Q. Zhang and R. Pless, “Extrinsic Calibration of a Camera and Laser Range Finder ( improves camera calibration )”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems,* pp. 38–43, 2004.

[23] R. Pless, “Code for calibration of cameras to planar laser range finders” [Online]. Available: https://research.engineering.wustl.edu/~pless/code.html.

[24] J. Briales and J. Gonzalez-Jimenez, “A minimal solution for the calibration of a 2D laser-rangefinder and a camera based on scene corners”*, IEEE International Conference on Intelligent Robots and Systems*, no. 3, pp. 1891–1896, 2015.

[25] O. Sorkine-Hornung and M. Rabinovich, “Least-Squares Rigid Motion Using SVD”, *Department of Computer Science, ETH Zurich*, 2017.

[26] Jong-Eun Ha, “Extrinsic Calibration of a Camera and Laser Range Finder Using a New Calibration Structure of a Plane with a Triangular Hole”, *International Journal of Control Automation and Systems,* 2012.

[27] “ROS” [Online]. Available: http://wiki.ros.org/.

[28] “apriltags” [Online]. Available: https://april.eecs.umich.edu/software/apriltag.html.

[29] “apriltags library” [Online]. Available: http://people.csail.mit.edu/kaess/apriltags/.

[30] E. Olson, “AprilTag : A robust and flexible visual fiducial system”, *IEEE International Conference on Robotics and Automation*, 2011.

[31] “OpenCV” [Online]. Available: https://opencv.org/.

[32] “Eigen” [Online]. Available: http://eigen.tuxfamily.org/index.php?title=Main_Page.

[33] “apriltags ros” [Online]. Available: http://wiki.ros.org/apriltags_ros.

[34] “April.Tag.Tag36h11”, [Online]. Available: https://robot2016.mit.edu/sites/default/files/documents/project_apriltag36h11.pdf.

[35] “Hokuyo” [Online]. Available: https://www.hokuyo-aut.jp/search/single.php?serial=169.

[36] Hokuyo, “Scanning Laser Range Finder”, *UTM-30LX/LN Specification datasheet,* 2008.

[37] “Logitech HD Pro Webcam” [Online]. Available: https://www.logitech.com/en-us/product/hd-pro-webcam-c920.

[38] Logitech, “Logitech ® hd pro webcam c920”, 2011.

[39] “camera_calibration monocular” [Online]. Available: http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration.

# Appendix

## 1 Parallel positioning (Test Case 1)

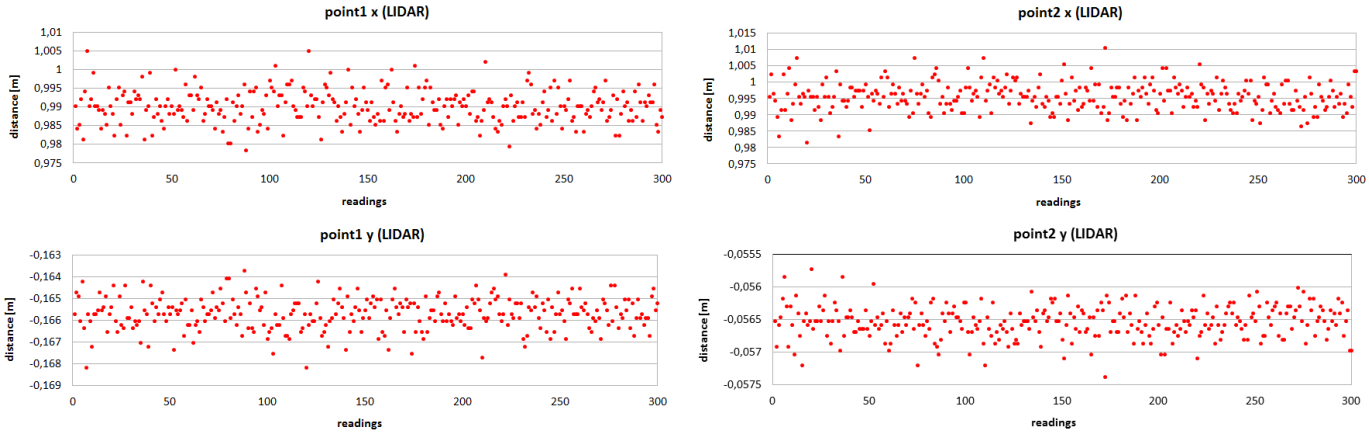



**Figure 42.** *Distribution of recorded LIDAR coordinates for Test case 1 of parallel positioning*

| LIDAR | Point 1 | | Point 2 | |
|---|---|---|---|---|
| | x | y | x | y |
| Min | 0,978 | -0,168 | 0,981 | -0,057 |
| Max | 1,005 | -0,164 | 1,010 | -0,056 |
| Mean | 0,990 | -0,166 | 0,996 | -0,057 |
| Median | 0,990 | -0,166 | 0,995 | -0,057 |

**Table 4.** *Min, max, mean and median of recorded LIDAR coordinates for Test case 1 of parallel positioning*

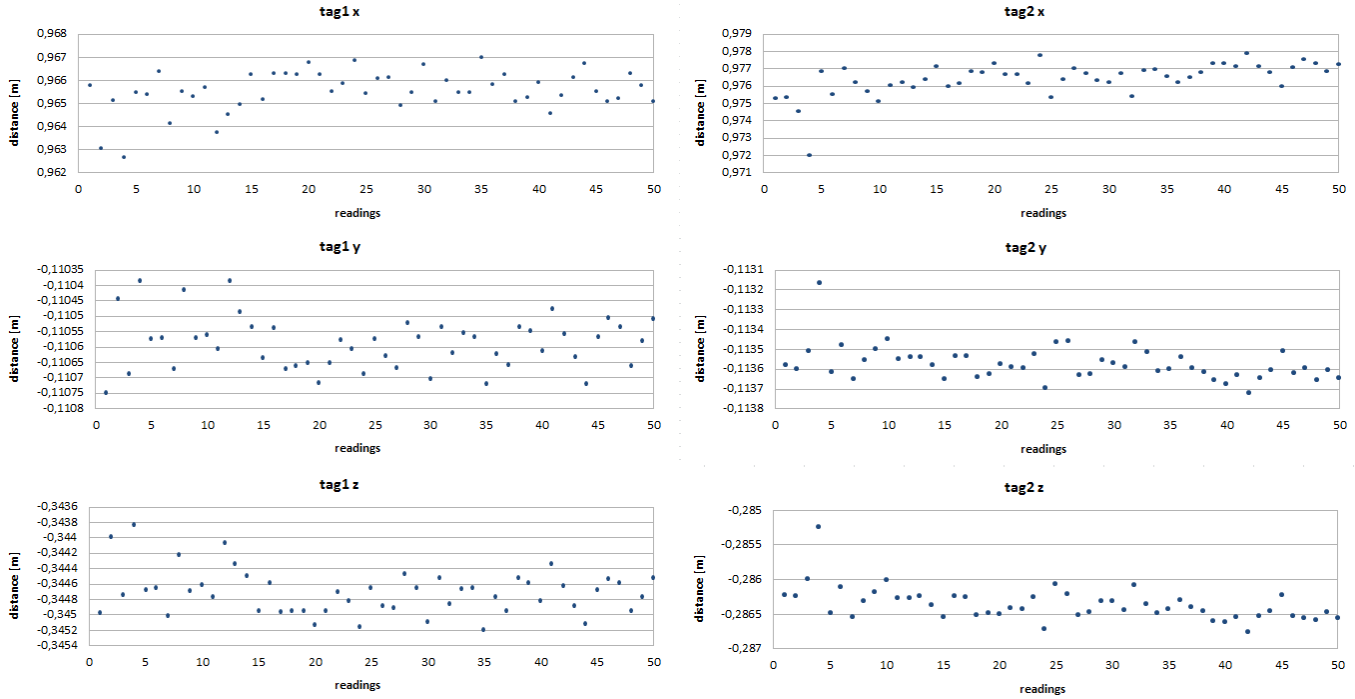



**Figure 43.** *Distribution of recorded tag coordinates for Test case 1 of parallel positioning*

| Camera | Tag 1 | | | Tag 2 | | |
|---|---|---|---|---|---|---|
| | x | y | z | x | y | z |
| Min | 0,963 | -0,111 | -0,345 | 0,972 | -0,114 | -0,287 |
| Max | 0,967 | -0,110 | -0,344 | 0,978 | -0,113 | -0,285 |
| Mean | 0,966 | -0,111 | -0,345 | 0,976 | -0,114 | -0,286 |
| Median | 0,966 | -0,111 | -0,345 | 0,977 | -0,114 | -0,286 |

**Table 5.** *Min, max, mean and median of recorded tag coordinates for Test case 1 of parallel positioning*

# 2 Parallel positioning (Test Case 2)

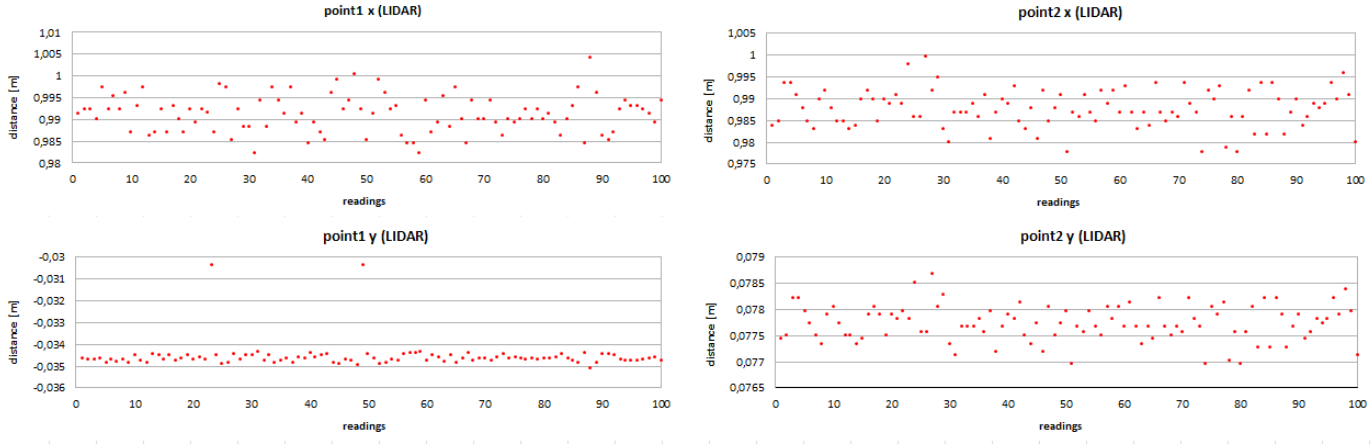



**Figure 44.** *Distribution of recorded LIDAR coordinates for Test case 2 of parallel positioning*

| LIDAR | Point 1 | | Point 2 | |
|---|---|---|---|---|
| | x | y | x | y |
| Min | 0,982 | -0,035 | 0,978 | 0,077 |
| Max | 1,004 | -0,030 | 1,000 | 0,079 |
| Mean | 0,991 | -0,035 | 0,988 | 0,078 |
| Median | 0,991 | -0,035 | 0,987 | 0,078 |

**Table 6.** *Min, max, mean and median of recorded LIDAR coordinates for Test case 2 of parallel positioning*

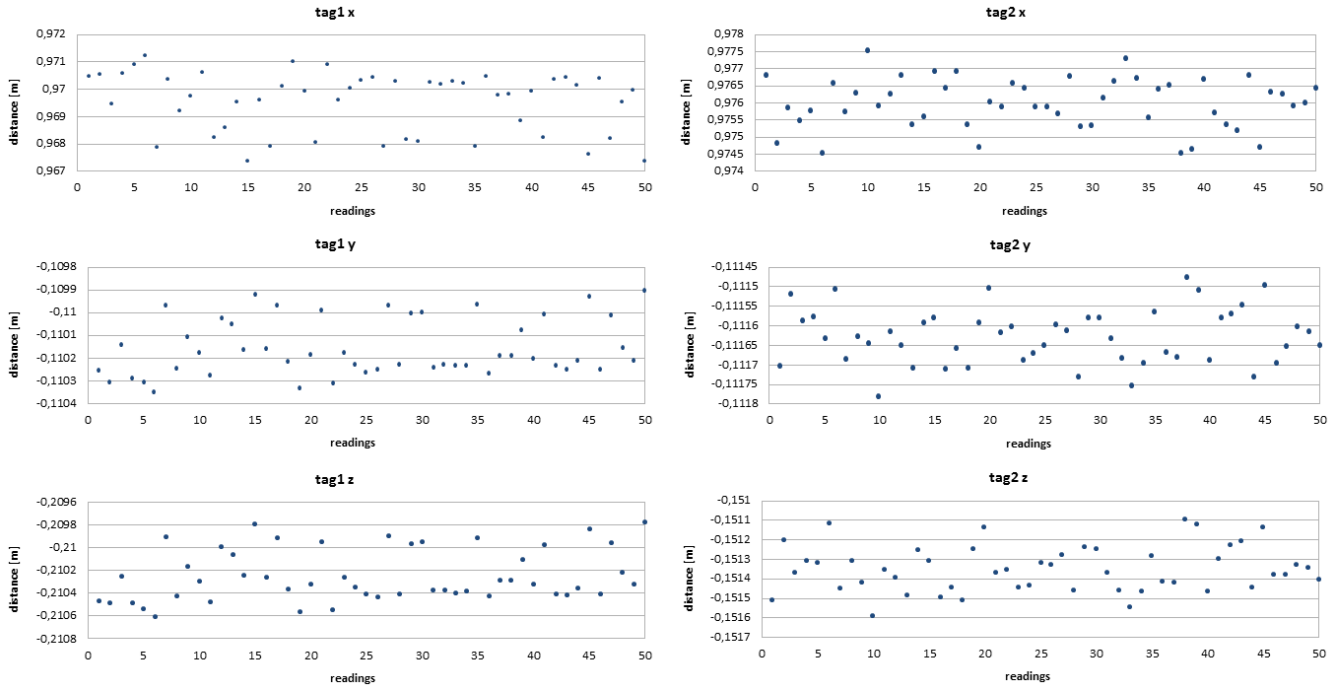



**Figure 45.** *Distribution of recorded tag coordinates for Test case 2 of parallel positioning*

| Camera | Tag 1 | | | Tag 2 | | |
|---|---|---|---|---|---|---|
| | x | y | z | x | y | z |
| Min | 0,967 | -0,110 | -0,211 | 0,975 | -0,112 | -0,152 |
| Max | 0,971 | -0,110 | -0,210 | 0,978 | -0,111 | -0,151 |
| Mean | 0,970 | -0,110 | -0,210 | 0,976 | -0,112 | -0,151 |
| Median | 0,970 | -0,110 | -0,210 | 0,976 | -0,112 | -0,151 |

**Table 7.** *Min, max, mean and median of recorded tag coordinates for Test case 2 of parallel positioning*

# 3 Angled positioning (Test Case 1)

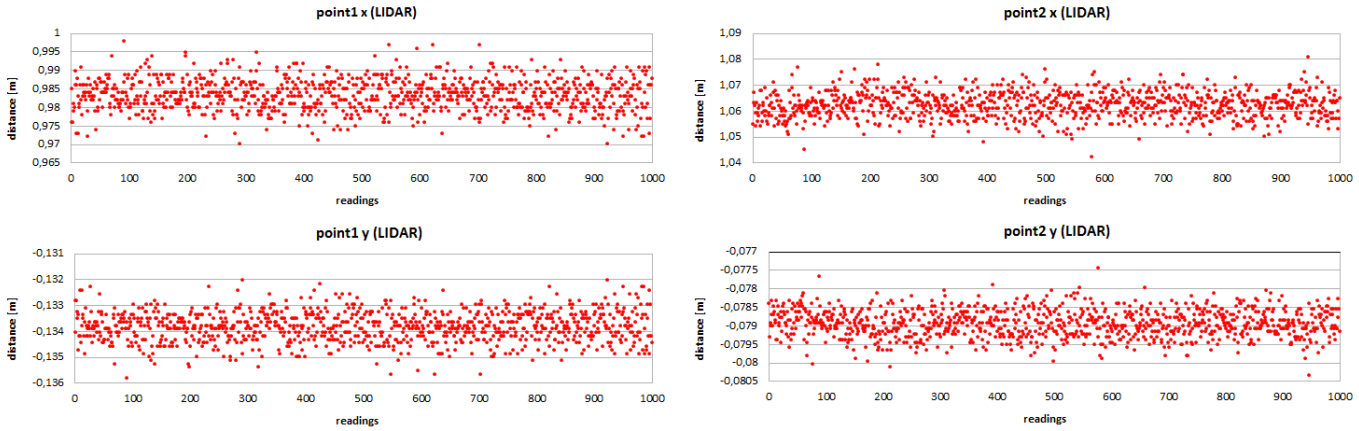



**Figure 46.** *Distribution of recorded LIDAR coordinates for Test case 1 of angled positioning*

| LIDAR | Point 1 | | Point 2 | |
|---|---|---|---|---|
| | x | y | x | y |
| Min | 0,970 | -0,136 | 1,042 | -0,080 |
| Max | 0,998 | -0,132 | 1,081 | -0,077 |
| Mean | 0,983 | -0,134 | 1,062 | -0,079 |
| Median | 0,983 | -0,134 | 1,062 | -0,079 |

**Table 8.** *Min, max, mean and median of recorded LIDAR coordinates for Test case 1 of angled positioning*

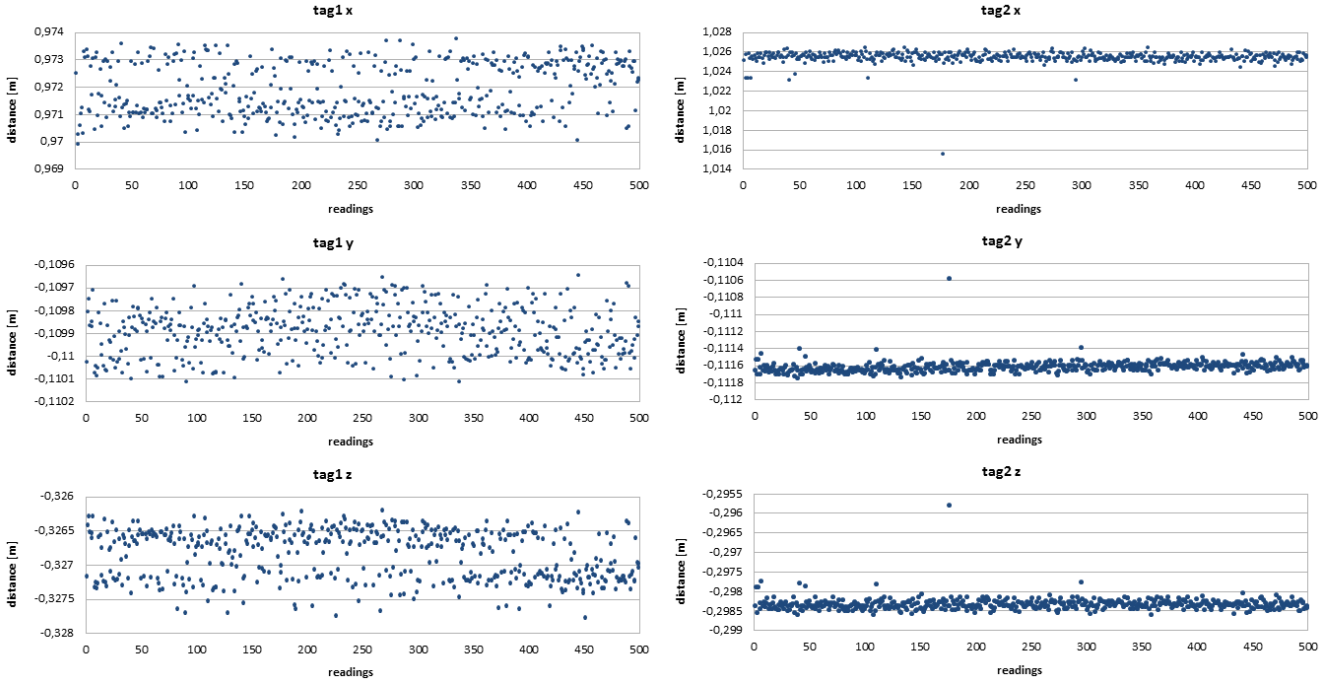



**Figure 47.** *Distribution of recorded tag coordinates for Test case 1 of angled positioning*

| Camera | Tag 1 | | | Tag 2 | | |
|---|---|---|---|---|---|---|
| | x | y | z | x | y | z |
| Min | 0,970 | -0,110 | -0,328 | 1,016 | -0,112 | -0,299 |
| Max | 0,974 | -0,110 | -0,326 | 1,026 | -0,111 | -0,296 |
| Mean | 0,972 | -0,110 | -0,327 | 1,025 | -0,112 | -0,298 |
| Median | 0,972 | -0,110 | -0,327 | 1,026 | -0,112 | -0,298 |

**Table 9.** *Min, max, mean and median of recorded tag coordinates for Test case 1 of angled positioning*

# 4 Angled positioning (Test Case 2)

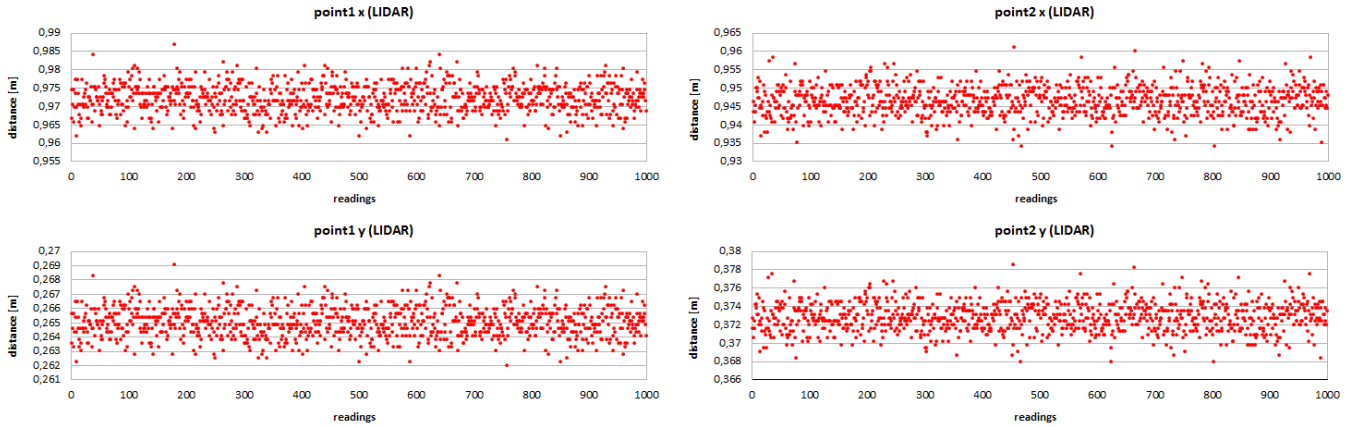



**Figure 48.** *Distribution of recorded LIDAR coordinates for Test case 2 of angled positioning*

| LIDAR | Point 1 | | Point 2 | |
|---|---|---|---|---|
| | x | y | x | y |
| Min | 0,961 | 0,262 | 0,934 | 0,368 |
| Max | 0,987 | 0,269 | 0,961 | 0,379 |
| Mean | 0,972 | 0,265 | 0,946 | 0,373 |
| Median | 0,972 | 0,265 | 0,946 | 0,373 |

**Table 10.** *Min, max, mean and median of recorded LIDAR coordinates for Test case 2 of angled positioning*

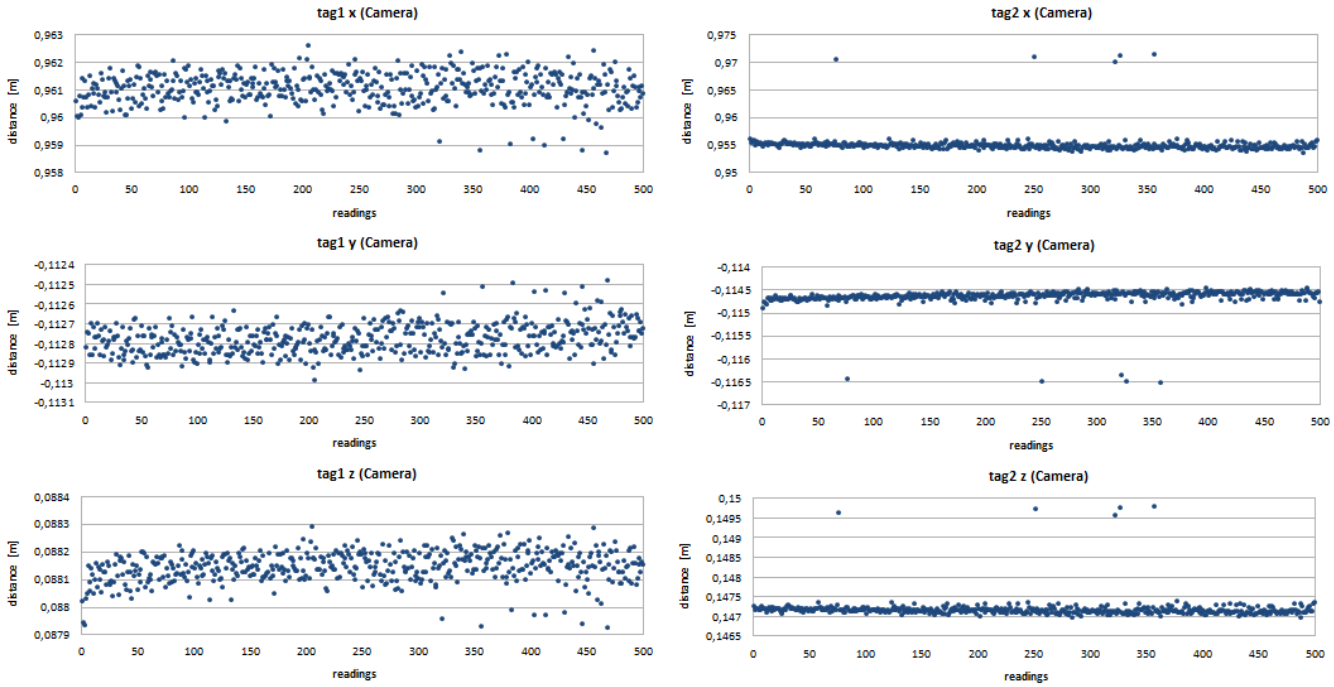



**Figure 49.** *Distribution of recorded tag coordinates for Test case 2 of angled positioning*

| Camera | Tag 1 | | | Tag 2 | | |
|---|---|---|---|---|---|---|
| | x | y | z | x | y | z |
| Min | 0,959 | -0,113 | 0,088 | 0,954 | -0,117 | 0,147 |
| Max | 0,963 | -0,112 | 0,088 | 0,971 | -0,114 | 0,150 |
| Mean | 0,961 | -0,113 | 0,088 | 0,955 | -0,115 | 0,147 |
| Median | 0,961 | -0,113 | 0,088 | 0,955 | -0,115 | 0,147 |

**Table 11.** *Min, max, mean and median of recorded tag coordinates for Test case 2 of angled positioning*

# Non-exclusive licence to reproduce thesis

I, Pavel Shumejko,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for the purpose of preservation in the DSpace digital archives until the expiry of the term of copyright,

Robust Solution for Extrinsic Calibration of a 2D Laser-Rangefinder and a Monocular USB Camera

supervised by Assoc. Prof. Karl Kruusamäe.

Publication of the thesis is not allowed.

2. I am aware of the fact that the author retains the right specified in p. 1.

3. This is to certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Pavel Shumejko

Tartu, 24/12/2018