1-22-2024 2:00 PM

A Target-Based and A Targetless Extrinsic Calibration Methods for Thermal Camera and 3D LiDAR

Farhad Dalirani, *Western University*

Supervisor: El-Sakka, Mahmoud R., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

This thesis introduces two novel methods for the extrinsic calibration of a thermal camera and a 3D LiDAR sensor, which are crucial for seamless data integration. The first method employs a distinctive calibration target, leveraging lines and plane equations correspondence in both modalities for a single pose, and incorporating more poses by matching the target's edges. It achieves reliable results, even with just one pose yielding 10.82% translation and 0.51-degree rotation errors. This outperforms alternative methods, which require eight pairs for similar results. The second method eliminates the need for a dedicated target. Instead, by collecting data during the sensor setup movement in environment and using a novel evolutionary algorithm optimizes a loss that measures alignment of humans in both modalities. This approach results in a 4.43% loss improvement compared to extrinsic parameters obtained by target-based methods. These methods save calibration time, reduce costs, and make sensor integration more accessible.

Keywords: Extrinsic Calibration, Cross Calibration, LiDAR, Thermal Camera, Sensor Fusion

Summary for Lay Audience

This thesis introduces two calibration methods for seamlessly integrating a thermal camera and a 3D LiDAR dataset, focusing on aligning their coordinate systems via a rotation matrix and a translation vector.

The first method utilizes a distinctive calibration target visible in both sensors. For a single pair of thermal image-point cloud data, the algorithm establishes correspondences between the target's lines and plane equations in both modalities, determining extrinsic parameters. Further enhancement involves incorporating more pairs by matching the target's edges in both modalities. This method demonstrates reliability even with just one pair and exhibits notable performance with sparse LiDARs. In testing, it achieves 10.82% translation and 0.009 radian rotation errors with a single pose, surpassing methods requiring 8 data pairs. Beyond accuracy, this approach offers practical advantages. It notably reduces time expenditure by adopting a single-pose calibration strategy, which is particularly beneficial in scenarios like automobile sensor setups, where challenges in target positioning and thermal stability are prominent.

The second method introduces an extrinsic calibration approach that eliminates the need for a dedicated calibration target. Instead, it leverages data collected during sensor setup movements in environments with human presence, such as streets or farm fields. The algorithm optimizes extrinsic parameters based on a designed loss function measuring the alignment of humans in both modalities. To minimize this loss, a novel evolutionary algorithm is employed. This method exhibits a 4.43% improvement in loss compared to target-based calibration parameters in one dataset. Its efficacy extends to challenging real-world environments and stands out for not requiring an initial solution. Beyond accuracy improvements, this method presents a range of practical benefits, including cost reduction in creating thermal camera-visible calibration targets, time-saving in diverse pose collection, mitigating the tedium of repetitive calibration in scenarios with sensor drift, and enhanced accessibility by eliminating the need for a specialized target during calibration.

Co-Authorship Statement

Chapter 2 of this thesis is derived from the following paper: Dalirani, Farhad, Farzan Heidari, Taufiq Rahman, Daniel Singh Cheema, and Michael A. Bauer. "Automatic Extrinsic Calibration of Thermal Camera and LiDAR for Vehicle Sensor Setups." In 2023 IEEE Intelligent Vehicles Symposium (IV), pp. 1-7. IEEE, 2023. Farzan Heidari recommended considering PnP instead of homography and helped in constructing the target and data collection. Dr. Taufiq Rahman assisted in constructing the target, assisting in creating more aesthetically pleasing figures with TikZ, and editing certain sections of the initial draft. Dr. Mike Bauer conducted revisions and edited the paper to enhance its submission quality, and he also presented the paper at the conference. Daniel Singh Cheema provided significant assistance in constructing the designed calibration target and facilitating the data collection process. I developed a novel calibration target specifically designed for achieving extrinsic calibration between a LiDAR sensor and a thermal camera. I created algorithms and methodologies to accurately determine the lines and plane equations representing the calibration target in the image, camera, and LiDAR coordinate systems. I formulated the calibration problem by defining the key parameters and variables involved in the extrinsic calibration process. Additionally, I designed cost functions for the optimization of extrinsic parameters. To evaluate and analyze the performance of the calibration method under various noise conditions, I developed a simulator, enabling a comprehensive investigation of its effectiveness. Furthermore, I conducted a thorough literature review and authored the paper on the topic.

Chapter 3 is a reformatted version of the following paper: Dalirani, Farhad, and Mahmoud R. El-Sakka. 2024. "Extrinsic Calibration of Thermal Camera and 3D LiDAR Sensor via Human Matching in Both Modalities during Sensor Setup Movement" Sensors 24, no. 2: 669. I handled conceptualization, methodology, experiments, and original draft preparation. Dr. Mahmoud R. El-Sakka took charge of paper flow, validation, writing, review, editing, and supervision.

Acknowlegements

I would like to express my gratitude and acknowledge the invaluable contributions of my supervisor, Professor Dr. Mahmoud El-sakka, and my former supervisor, Professor Dr. Mike Bauer, whose guidance and support made this work possible. Special thanks are also extended to my family and friends, especially my wife, Maryam, for their unwavering and continuous support throughout this endeavor.

I would like to extend my appreciation to Farzan Heidari, Dr. Taufiq Rahman, and Daniel Singh Cheema. Without their help, Chapter 2 would not have been possible, and during my collaboration with them, I learned a lot of valuable lessons.

Part of this thesis has been supported by the National Research Council of Canadas Artificial Intelligence for Logistics Program and the Natural Sciences and Engineering Research Council of Canada.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviation	Definition
LiDAR	Light Detection and Ranging
SLAM	Simultaneous Localization and Mapping
IMU	Inertial Measurement Unit

Symbol	Definition
$\mathbf{K}$	3×3 camera intrinsic matrix
$\mathbf{R}$	3×3 rotation matrix
$\mathbf{t}$	3×1 translation vector
$\mathbf{p}^L$	A 3D point in LiDAR coordinate system
$\mathbf{p}^C$	A 3D point in thermal camera coordinate system
$\mathbf{p}^I$	A 2D point in thermal image coordinate system
$\mathbf{n}$	3D normal vector of a plane
$\mathbf{d}$	3D direction vector of a line in 3D space
$\mathbf{P}$	A LiDAR point cloud
$\mathbf{I}$	A thermal camera image

Chapter 1

Introduction

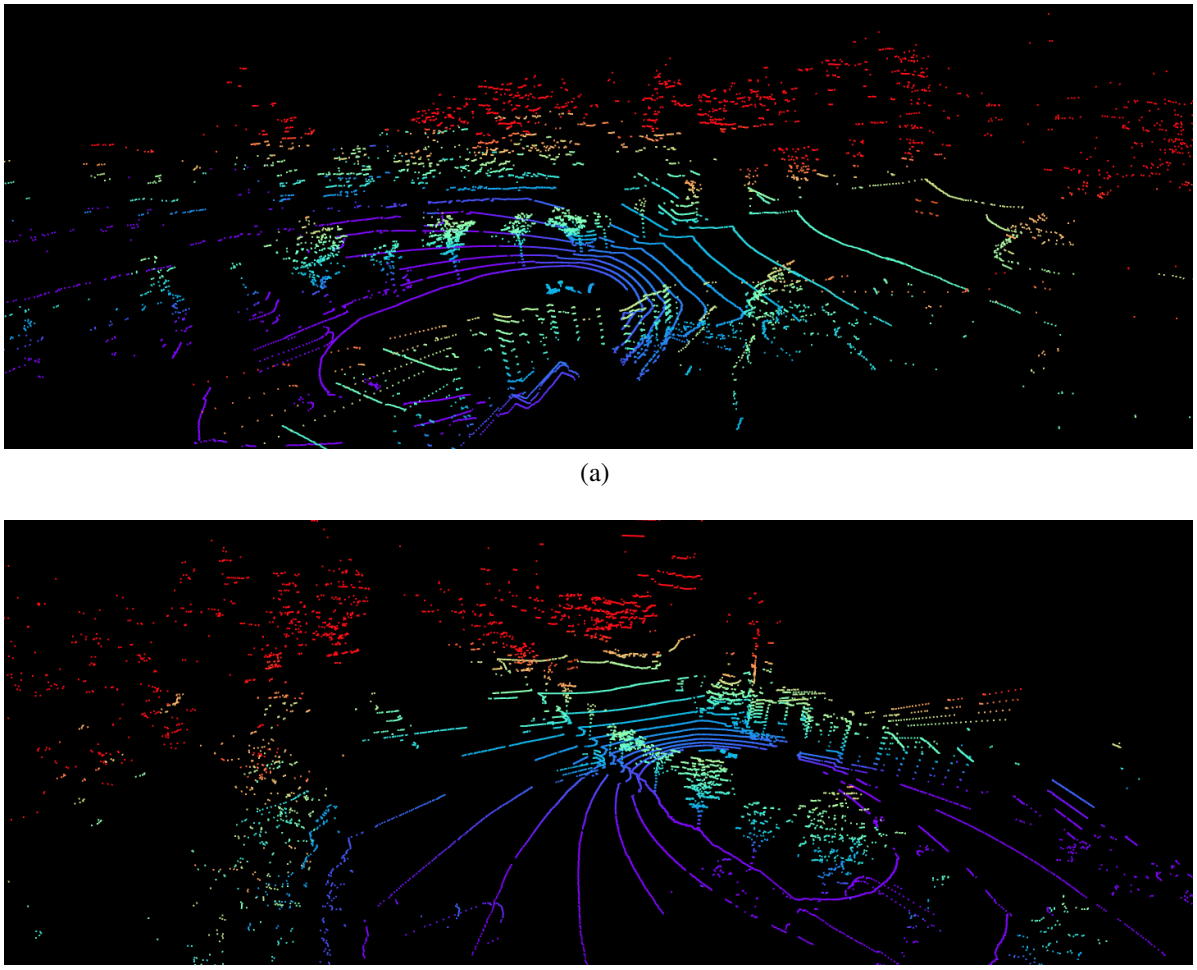
In this chapter, we begin by introducing 3D LiDAR (Light Detection and Ranging) and thermal camera sensors. Following that, we define the extrinsic calibration task. Subsequently, we explore various applications that arise from combining these two sensors. Afterward, we provide a concise summary of our proposed methods. Finally, we will outline the contributions of various individuals and provide an overview of the structure of the remaining sections of this thesis.

1.1 LiDAR Sensor

A 3D LiDAR sensor utilizes an array of laser beams to generate a 3D point cloud of the environment in which the sensor operates. The time between the emission of a laser ray and the observation of its reflection is used to calculate the distance of the hit point from the sensor.

In the context of LiDAR sensor, a scan refers to the process of systematically directing laser beams toward the surrounding environment and measuring the reflected signals. During a LiDAR scan, the sensor emits laser pulses in different directions, and these pulses interact with objects in the scene. The result of a scan is a set of spatial data points, often referred to as a point cloud. Each point in the cloud represents a specific location in the scanned environment and is defined by its coordinates (x, y, z) in the LiDAR sensor's coordinate system. Also, some LiDAR sensors, in addition to capturing spatial locations of points, output the strength of the returned laser pulses. This information indicates the reflectivity of the hit surface. Fig. 1.1 illustrates an example of a point cloud from two different points of view captured by a 3D LiDAR, showing a street scene.

3D LiDAR sensors provide an accurate 3D point cloud, and for this reason, they have found a lot of applications in various fields such as surveillance, agriculture, forestry, robotics, advanced driving assistance systems, and the self-driving car industry. For example, Wisultschew et al. [2] used a 3D LiDAR to detect and track objects in a railway level crossing. Proudman et al. [3] proposed a method to reconstruct forests and calculate features of each tree, such as diameter at breast height (DBH). Liu et al. [4] proposed a method to find fruit trees and some of their features, such as the center of mass, to minimize the amount of used pesticides. Li et al. [5] designed a method to perform 3D object detection in autonomous driving scenarios with a 3D LiDAR sensor. In the study of Vizzo et al. [6], an algorithm was proposed that gener-



(a)

(b)

Figure 1.1: Two screenshots captured from different perspectives showcasing a point cloud generated by the presence of a 3D LiDAR sensor on a street. The point cloud is derived from the MS^2 dataset [1]. The colors of the points are based on the height and distance of each point from the LiDAR sensor, enhancing visualization.

(a) (b)

Figure 1.2: Comparison of the same scene captured with a thermal camera (a) and an RGB camera (b) from the FLIR dataset [12]. Thermal image is shown as a grayscale image.

ates an accurate 3D map of the environment using 3D LiDAR scans during movement of their robot.

1.2 Thermal Camera

Unlike visible light cameras that capture the visible spectrum of light, thermal cameras form an image by capturing heat information from the scene. This enables thermal cameras to capture information that is invisible to other sensors like normal cameras. Additionally, they can operate effectively in adverse conditions that normal cameras cannot, such as smoke, darkness, fog, dust, rain, snow, etc. This makes them an ideal sensor choice for many tasks in various industries. Fig. 1.2 illustrates a thermal image alongside its comparison with an RGB image captured from the same scene. For instance, in the study by Leira et al. [7], a thermal camera was employed to detect and track objects in the sea. Kristo et al. [8] investigated object detection using a thermal camera in adverse weather conditions, including the identification of humans engaged in illegal activities around protected areas, such as illegal immigration at borders. Miethig et al. [9] utilized a thermal camera for object detection in autonomous driving scenarios under challenging lighting and weather conditions. Steen et al. [10] utilized a thermal camera to prevent accidental harm or mortality of wild animals during farming operations, such as mowing. In the work of Ibarguren et al. [11], a thermal camera was used for early inspection and leakage detection in thermal power plants.

1.3 Problem Definition

In this thesis, we address the extrinsic calibration task between a thermal camera and a 3D LiDAR sensor, which is crucial for fusing data from both sensors. The extrinsic calibration

of these two sensors entails determining the 3×3 orthogonal rotation matrix $\mathbf{R}$ and the 3-dimensional translation vector $\mathbf{t}$ that characterize the spatial relationship from the LiDAR sensor coordinate system to the thermal camera coordinate system. Fig. 1.3 depicts this concept.

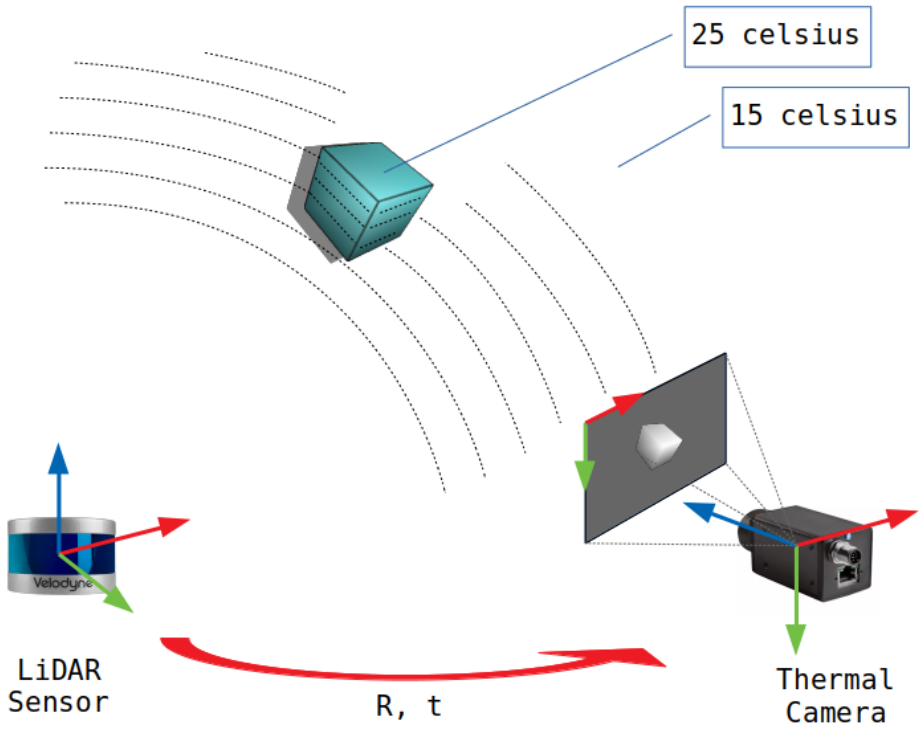


Figure 1.3: A simplified and symbolic representation of an object with a temperature of 25 degrees Celsius in an environment with a temperature of 15 degrees Celsius, placed in front of a LiDAR sensor and a thermal camera. Additionally, coordinate systems for the thermal camera, thermal image and the LiDAR sensor are depicted. $\mathbf{R}$ and $\mathbf{t}$ define the spatial relationship between the coordinate systems of the LiDAR sensor and the thermal camera.

By employing a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, the point $\mathbf{p}^L$ in the LiDAR coordinate system can be represented in the camera coordinate system as $\mathbf{p}^C$ through Eq. 1.1. Additionally, utilizing the 3×3 intrinsic camera matrix $\mathbf{K}$ allows us to derive the image coordinate $\mathbf{p}^I$ using Eq. 1.2. $\mathbf{p}^I$ is in homogeneous format.

$$\mathbf{p}^C = \mathbf{R}\mathbf{p}^L + \mathbf{t} \tag{1.1}$$

$$\mathbf{p}^I = \mathbf{K}(\mathbf{R}\mathbf{p}^L + \mathbf{t}) \tag{1.2}$$

1.4 Application of Fusing Data from Thermal Camera and LiDAR

LiDAR sensors solely provide the 3D location of points in the LiDAR coordinate system. Therefore, data from other sensors can be used as complementary information to enhance performance in various tasks. For example, Shan et al. [13] proposed a method that performs accurate simultaneous localization and mapping by fusing 3D LiDAR data with data from an IMU (Inertial Measurement Unit). Asvadi et al. [14] integrated data from a 3D LiDAR sensor and a color camera to improve vehicle detection compared to using each modality alone. In the work by Hwang et al. [15], a method was proposed that leverages complementary data from a color camera and a 3D LiDAR for highly efficient detection and tracking in intelligent vehicles. Hajri et al. [16] proposed a method that capitalizes on the fact that LiDAR sensors are highly accurate in determining obstacle positions but less precise in estimating their velocities, while radars exhibit greater precision in capturing obstacle velocities but are less accurate in pinpointing their positions. The fused approach aims to enhance obstacle detection by leveraging the advantages of both sensors.

Due to the unique information captured by thermal cameras, as well as the diverse adverse conditions under which they can operate, coupled with their wide-ranging applications in various industries, there has been a recent surge of interest in integrating them with 3D LiDARs. For example, in the study conducted by Fritsche et al. [17], a robot was equipped with various sensors, including a thermal camera and LiDAR, to detect hotspot hazards. Narvaez et al. [18] employed a thermal camera and LiDAR technology to generate a thermal 3D reconstruction of fruit trees for the purpose of monitoring and characterizing it. Choi et al. [19] integrated data from a thermal camera and a LiDAR sensor to enhance detection capabilities in autonomous vehicles during low visibility conditions. Kragh et al. [20] equipped a tractor with a variety of sensors, such as LiDAR and a thermal camera, to identify obstacles, including humans. This implementation aimed to enhance safety measures during farming operations. Choi et al. [21] created a diverse dataset that incorporates LiDAR sensor and thermal camera information. This dataset serves as a valuable resource for investigating tasks such as drivable region detection, object detection, localization, and more. The scope of the dataset covers both assisted and autonomous driving scenarios, encompassing daytime and nighttime conditions.

1.5 Proposed Methods

We introduce two novel calibration methods. The first method uses a unique calibration target visible in both LiDAR and thermal camera modalities. For a single pair of thermal image-point cloud data, the algorithm detects the target and establishes correspondences between their lines and plane equations to determine extrinsic parameters. When using multiple pairs, the algorithm matches the targets edges in both modalities. This method produces reliable results, even with just one pair, and works well with sparse LiDARs. In our tests, it reached a translation error of 10.82% and a rotation error of 0.51 degree with a single pose, while another method needed 8 data pairs to match these errors. This work offers several advantages. Firstly, using a single pose instead of collecting multiple poses saves time. Secondly, calibration targets can become cool or exhibit heat leaks shortly after, rendering them ineffective for calibration.

Using just one pose can address this issue. These benefits are especially significant in scenarios like sensor setups in automobiles, where adjusting the target's position and height is more challenging.

To the best of our knowledge, the exploration of the extrinsic calibration between a thermal camera and a 3D LiDAR for a vehicle sensor setup, accomplishing satisfactory results with just one pose, and without the need for an initial estimate of the extrinsic parameters, has not been undertaken previously.

In the second method, we introduce an extrinsic calibration approach that eliminates the need for a dedicated calibration target. Instead, we use data collected during sensor setup movements in environments with human presence, like streets or farm fields. The algorithm optimizes the extrinsic parameters based on a designed loss that measures the alignment of humans in both modalities. The method then minimizes the loss using a novel evolutionary algorithm. Our method achieved a 4.43% improvement in the loss compared to the extrinsic parameters obtained through a target-based calibration method in one of the datasets we used. This method excels in challenging real-world environments and does not require an initial solution. It offers several advantages. First, creating a calibration target visible in a thermal camera can be expensive. Second, collecting various poses is time-consuming. Third, repetitive calibration in scenarios where regular sensor drift and setup changes occur can be highly time-consuming and frustrating. Finally, Eliminating the need for a special target during calibration makes this combination of sensors more accessible to a broader range of users.

To the best of our knowledge, the extrinsic calibration of a thermal camera and a 3D LiDAR without the need for a dedicated calibration target visible in the thermal camera modality, by matching humans in both modalities, has not been addressed previously.

1.6 Contributions

Chapter 2 of this thesis is derived from the following paper: Dalirani, Farhad, Farzan Heidari, Taufiq Rahman, Daniel Singh Cheema, and Michael A. Bauer. "Automatic Extrinsic Calibration of Thermal Camera and LiDAR for Vehicle Sensor Setups." In 2023 IEEE Intelligent Vehicles Symposium (IV), pp. 1-7. IEEE, 2023. Farzan Heidari recommended considering PnP instead of homography and helped in constructing the target and data collection. Dr. Taufiq Rahman assisted in constructing the target, assisting in creating more aesthetically pleasing figures with TikZ, and editing certain sections of the initial draft. Dr. Mike Bauer conducted revisions and edited the paper to enhance its submission quality, and he also presented the paper at the conference. Daniel Singh Cheema provided significant assistance in constructing the designed calibration target and facilitating the data collection process. I developed a novel calibration target specifically designed for achieving extrinsic calibration between a LiDAR sensor and a thermal camera. I created algorithms and methodologies to accurately determine the lines and plane equations representing the calibration target in the image, camera, and LiDAR coordinate systems. I formulated the calibration problem by defining the key parameters and variables involved in the extrinsic calibration process. Additionally, I designed cost functions for the optimization of extrinsic parameters. To evaluate and analyze the performance of the calibration method under various noise conditions, I developed a simulator, enabling a comprehensive investigation of its effectiveness. Furthermore, I conducted a thorough litera-

ture review and authored the paper on the topic.

Chapter 3 is a reformatted version of the following paper: Dalirani, Farhad, and Mahmoud R. El-Sakka. 2024. "Extrinsic Calibration of Thermal Camera and 3D LiDAR Sensor via Human Matching in Both Modalities during Sensor Setup Movement" Sensors 24, no. 2: 669. I handled conceptualization, methodology, experiments, and original draft preparation. Dr. Mahmoud R. El-Sakka took charge of paper flow, validation, writing, review, editing, and supervision.

1.7 Thesis Structure

The thesis follows an integrated article structure with two articles, each featuring an abstract outlining research objectives. The introduction provides context, the literature review examines existing scholarship, the proposed method details the methodology, and the experiment section presents findings. The conclusion discusses key insights and suggests avenues for future research. Additionally, the bibliography section refers to other papers.

References

[1] U. Shin, J. Park, and I. S. Kweon, "Deep depth estimation from thermal image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1043–1053, 2023.

[2] C. Wisultschew, G. Mujica, J. M. Lanza-Gutierrez, and J. Portilla, "3d-lidar based object detection and tracking on the edge of iot for railway level crossing," *IEEE Access*, vol. 9, pp. 35718–35729, 2021.

[3] A. Proudman, M. Ramezani, S. T. Digumarti, N. Chebrolu, and M. Fallon, "Towards real-time forest inventory using handheld lidar," *Robotics and Autonomous Systems*, vol. 157, p. 104240, 2022.

[4] L. Liu, Y. Liu, X. He, and W. Liu, "Precision variable-rate spraying robot by using single 3d lidar in orchards," *Agronomy*, vol. 12, no. 10, p. 2509, 2022.

[5] Z. Li, F. Wang, and N. Wang, "Lidar r-cnn: An efficient and universal 3d object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7546–7555, 2021.

[6] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1029–1036, 2023.

[7] F. S. Leira, H. H. Helgesen, T. A. Johansen, and T. I. Fossen, "Object detection, recognition, and tracking from uavs using a thermal camera," *Journal of Field Robotics*, vol. 38, no. 2, pp. 242–267, 2021.

[8] M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using yolo," *IEEE access*, vol. 8, pp. 125459–125476, 2020.

[9] B. Miethig, A. Liu, S. Habibi, and M. v. Mohrenschildt, "Leveraging thermal imaging for autonomous driving," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–5, IEEE, 2019.

[10] K. A. Steen, A. Villa-Henriksen, O. R. Therkildsen, and O. Green, "Automatic detection of animals in mowing operations using thermal cameras," *Sensors*, vol. 12, no. 6, pp. 7587–7597, 2012.

[11] A. Ibarguren, J. Molina, L. Susperregi, and I. Maurtua, "Thermal tracking in mobile robots for leak inspection activities," *Sensors*, vol. 13, no. 10, pp. 13560–13574, 2013.

[12] F. Teledyne, "Free teledyne flir thermal dataset for algorithm training," *Teledyne FLIR*, 2018.

[13] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, IEEE, 2020.

[14] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3d-lidar and color camera data," *Pattern Recognition Letters*, vol. 115, pp. 20–29, 2018.

[15] S. Hwang, N. Kim, Y. Choi, S. Lee, and I. S. Kweon, "Fast multiple objects detection and tracking fusing color camera and 3d lidar for intelligent vehicles," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 234–239, IEEE, 2016.

[16] H. Hajri and M.-C. Rahal, "Real time lidar and radar high-level fusion for obstacle detection and tracking with evaluation on a ground truth," *arXiv preprint arXiv:1807.11264*, 2018.

[17] P. Fritsche, B. Zeise, P. Hemme, and B. Wagner, "Fusion of radar, lidar and thermal information for hazard detection in low visibility environments," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp. 96–101, IEEE, 2017.

[18] F. J. Y. Narváez, J. S. del Pedregal, P. A. Prieto, M. Torres-Torriti, and F. A. A. Cheein, "Lidar and thermal images fusion for ground-based 3d characterisation of fruit trees," *Biosystems Engineering*, vol. 151, pp. 479–494, 2016.

[19] J. D. Choi and M. Y. Kim, "A sensor fusion system with thermal infrared camera and lidar for autonomous vehicles and deep learning based object detection," *ICT Express*, vol. 9, no. 2, pp. 222–227, 2023.

[20] M. F. Kragh, P. Christiansen, M. S. Laursen, M. Larsen, K. A. Steen, O. Green, H. Karstoft, and R. N. Jørgensen, "Fieldsafe: dataset for obstacle detection in agriculture," *Sensors*, vol. 17, no. 11, p. 2579, 2017.

[21] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

Chapter 2

Automatic Extrinsic Calibration of Thermal Camera and LiDAR for Vehicle Sensor Setups

This Chapter is a reformatted version of the following article:

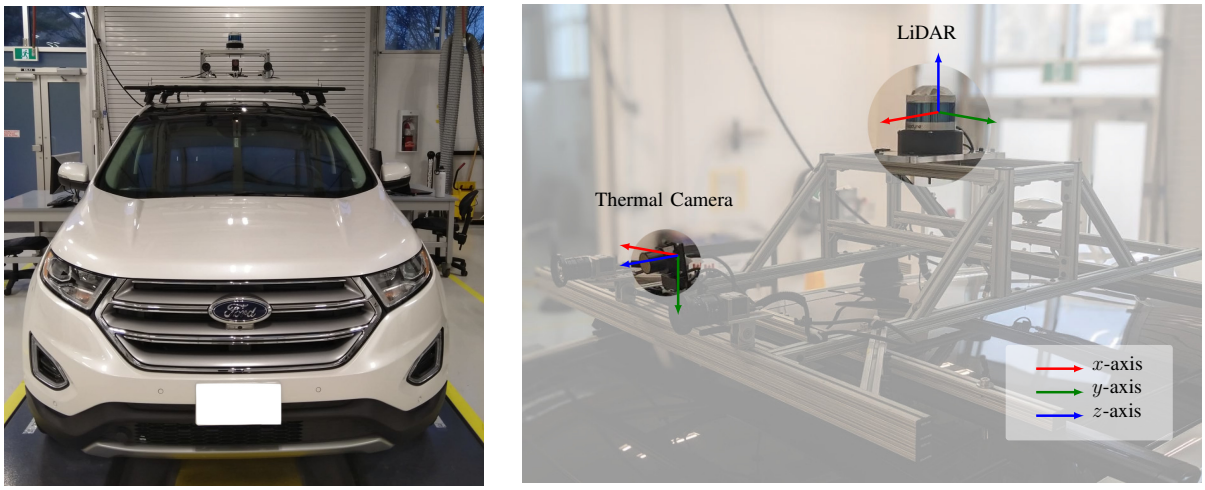
Dalirani, Farhad, Farzan Heidari, Taufiq Rahman, Daniel Singh Cheema, and Michael A. Bauer. "Automatic Extrinsic Calibration of Thermal Camera and LiDAR for Vehicle Sensor Setups." In 2023 IEEE Intelligent Vehicles Symposium (IV), pp. 1-7. IEEE, 2023.

2.1 Abstract

LiDAR is one of the most used sensors in many areas like robotics, self-driving cars, and advanced driving assistance systems due to providing an accurate point cloud of the surroundings. However, to cope with challenges in perceiving the environment around a vehicle, LiDAR data is often combined with data from other sensors. Thermal cameras can provide complementary information that can be beneficial, especially for detecting pedestrians and seeing at nighttime and in fog, dust, etc. In this paper, we propose an algorithm for the extrinsic calibration of a thermal camera and a LiDAR sensor in a vehicle. First, one or more thermal image-point cloud pairs of our designed calibration target are collected. Then line and plane equations of the target's edges and plane in both data modalities are found. Finally, the algorithm uses lines and plane correspondences to cross-calibrate the sensors. The proposed method obtains good results with one or more poses. We also show that it works well with sparse LiDAR data. Several experiments are presented to illustrate the effectiveness of the method.

2.2 Introduction

LiDAR (light detection and ranging) has become a useful tool in computer vision to solve various problems, such as 3D object detection [1], 3D segmentation [2], SLAM [3], etc. In the past decade, the usage of LiDAR has greatly increased, particularly in autonomous vehicles and vehicles with advanced driving assistance systems. LiDAR sensors provide sparse depth



(a) (b)

Figure 2.1: a) Our instrumented vehicle. b) A close view of LiDAR and the thermal camera with their coordinate systems.

point clouds, and they usually work in near infra-red ($0.7\mu - 1\mu$) spectrum range, which has little or minor intersection with the spectrum range in which color and thermal cameras operate. The advantages of using LiDAR can be enhanced by combining it with data from other kinds of cameras [4].

Thermal imaging is one of the sensing modalities that has recently gained much attention. Thermal cameras have applications in medicine, the military, surveillance, detecting and tracking of humans, etc. [5]. Thermal cameras can also work effectively at night and in fog, rain, smoke, snow, and dust and so can provide a good complement to other sensing devices. This also makes them a good sensor choice in advanced driver-assistance systems and autonomous driving [6].

Consequently, fusing data from LiDAR and thermal cameras can provide many benefits. For example, Choi et al. [7] created a multi-modal dataset that included LiDAR and thermal data to study different tasks such as object detection, driveable region detection, and localization. Tsoulias et al. [8] used thermal and LiDAR data to detect disorders on the surface of fruit, and Yue et al. [9] used thermal and LiDAR data with other data to create 3D point cloud maps of unstructured environments in day and night.

To use the data from a LiDAR and a thermal camera simultaneously, the two sensors need to be cross-calibrated. Cross-calibration is the process whereby two extrinsic parameters $\boldsymbol{R}$, the rotation matrix, and $\boldsymbol{t}$, the translation vector, between two sensors are found. With $\boldsymbol{R}$ and $\boldsymbol{t}$, points in the LiDAR coordinate system can be expressed in terms of the camera coordinate system. In this paper, we propose a thermal camera and LiDAR cross-calibration algorithm for vehicle sensor setups. Fig. 2.1 shows our vehicle setup. The contributions of this work are as follows:

- The method uses both plane and line correspondence between the plane and edges of the calibration target in the thermal image and LiDAR point cloud.

- The calibration algorithm obtains accurate results with one pose, and there is no need to collect several pairs of thermal images and LiDAR point clouds. However, more than one pose can be used to obtain slightly better results.
- The process is automatic, and there is no need for manual interactions, such as manually selecting points or providing good initial $\boldsymbol{R}$ and $\boldsymbol{t}$.
- The method works on data from sparse LiDARs, such as a 16-beam LiDAR.
- We show performance results of the method on both real and simulated data.

The rest of the paper is organized as follows. In Section 2.3, we introduce and review related work. Section 2.4 describes the cross-calibration algorithm. Section 2.5 presents experiments and results. Finally, in Section 2.6, we conclude our paper and explain directions for future research.

2.3 Related Works

Some previous research has addressed the extrinsic calibration of a LiDAR and a thermal camera. Borrmann et al. [10] created a circular calibration target using light bulbs as the target for a thermal camera. For the calibration, they collected several pairs of images and point clouds. Then, for each pair, they specified the location of light bulbs in the thermal images and point clouds. To determine the position of the light bulbs in the LiDAR coordinate system, they found the calibration target in the point cloud. By utilizing the known geometry of their calibration target, they calculated the location of the light bulbs in the LiDAR coordinate system. Afterward, for each pair, they projected the position of light bulbs in the point cloud to the thermal image. Finally, to find extrinsic parameters, they solved an optimization problem that minimizes the distance between the position of light bulbs in the thermal image and the projected position of light bulbs in the point cloud.

Krishnan, et al. [11] created a calibration target using black and white melamine squares which they glued together to create a checkerboard pattern. To use their target, they put it in the sun for one hour. To find the extrinsic calibration parameters between the thermal camera and Lidar, a user manually selects four corners of the calibration target in the thermal image. Then, they used a region-growing segmentation algorithm to segment the calibration target in the LiDAR point cloud. Their method also requires an initial rotation and translation vector. They map the edges of the calibration target in the point cloud to the thermal image. They utilized an optimization function to minimize the distance between mapped points and their closest thermal camera edge points. Their method needs several poses and a good initial rotation and translation vector to converge.

In a slightly different approach, Azam et al. [12] used a thermal camera that outputs both visual and thermal images which also supplies extrinsic parameters between visual and thermal images. They used well-known RGB camera-LiDAR calibration methods to cross-calibrate the visual camera and LiDAR. Then, they used this information along with extrinsic calibration parameters between the visual and thermal cameras to obtain the rotation and translation vectors from the LiDAR to the thermal camera.

Krishnan and Srikanth [13] created a calibration target by carving out a circle with a known radius on white cardboard. To detect the circle in the thermal camera, they put a wet black cloth in the target's background. Wetness allows the thermal camera to see the circle. The user manually clicks on a pixel inside the circle in the thermal image; this is used as an initial seed for a region-growing algorithm to segment the hole inside the image. Similarly, the user selects a point on the cardboard which provides an initial seed for finding the pattern in the point cloud. Afterward, they found the edges of the circle in both sensors. They collected several image and point cloud pairs by placing the target in different locations and orientations. For each pair, they projected the edges of the circle in the point cloud to the thermal image. Finally, they solved an optimization problem that tries to align the circle edges in the thermal camera and the projected edges.

Zhang et al. [14] breaks the calibration between the thermal camera and LiDAR into two steps. First, they cross-calibrated an RGB camera with LiDAR. Second, they found extrinsic parameters between the RGB camera and the thermal camera.

Some approaches do not make use of a target. Fu et al. [15] proposed a targetless algorithm that can calibrate a thermal camera and a LiDAR sensor based on aligning the edge features in scenes. Mharolkar et al. [16] proposed a targetless calibration method that used deep learning methods for feature extraction and feature matching. They trained a convolutional neural network that gets thermal images and mapped point clouds on images and which outputs extrinsic calibration parameters.

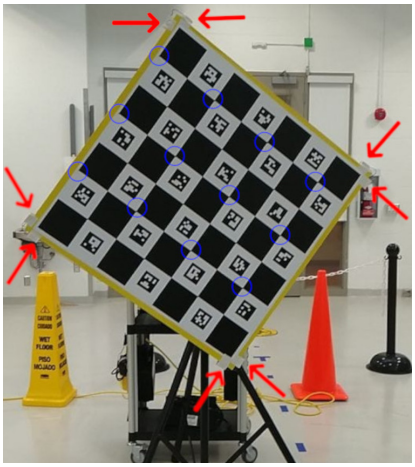
These previous approaches require one or more of the following: user's manual interaction, need for more than one pose, good initial rotation matrix and translation vector. Our proposed method does not require users to select points, nor is there a need for suitable initial extrinsic parameters. Furthermore, it achieves accurate results with just one pose.

2.4 Calibration Method

Our proposed method is based on plane and line correspondence of the calibration target and its edges in the thermal image and the LiDAR point cloud. It achieves good accuracy with one thermal image and a point cloud, and there is no need for multiple poses. However, it can use more poses by trying to align the edges of the calibration target in both data modalities. The algorithm outputs a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$. In the following, the steps of the algorithm are explained. Lower-case letters show scalar variables, bold lower-case letters are used for vectors, and bold upper-case letters are used for matrices. In superscripts, C, L, P, and E correspond to the camera coordinate, LiDAR coordinate, plane, and edge.

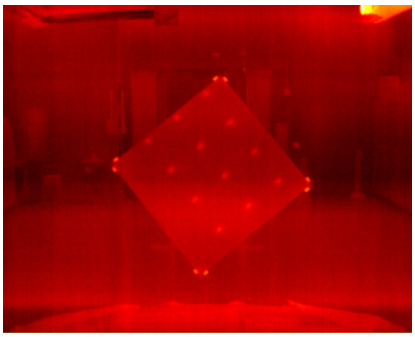
2.4.1 Calibration Target

Fig. 2.2 shows our thermal calibration target. We used a checkerboard pattern and used resistors to generate heat. The checkerboard was glued onto a thin wooden surface. By using the checkerboard pattern, we created a grid of equally spaced resistors and glued them to the back of the target. The resistors' heat patterns are different and have no specific shapes, such as circles. To detect the checkerboard's corners accurately, we drilled a small hole in the checkerboard's corners, which allowed a corner on the checkerboard to have a slightly higher

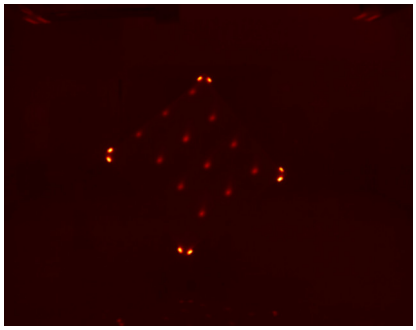


(a)

(b)



(c)



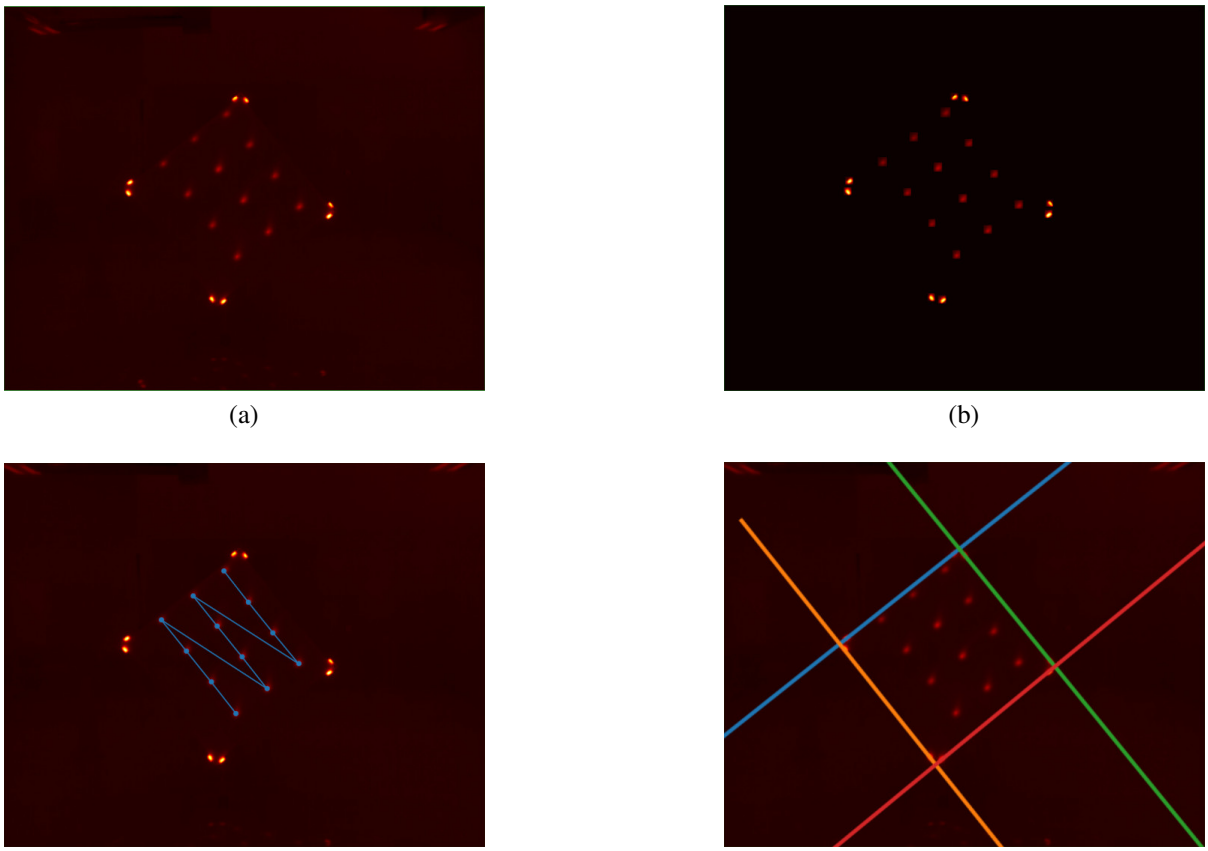
(d)

Figure 2.2: a) Front of the thermal calibration target. Red arrows show the resistors' position on the edges of the target; blue circles show the checkerboard corners where resistors were attached to the back of the target. b) Back of the target showing placement of resistors. c) Image of the target before reaching the desired temperature. d) Image of the target after reaching a temperature above room temperature.

temperature than its surroundings. To detect the edges of the target, we attached two resistors to the ends of each edge. As shown in Fig. 2.2(d), after the resistors reached a temperature above the room temperature, few other things can be seen in the thermal image. For our algorithm, the calibration target should be placed in the scene like a diamond shape, as depicted in Fig. 2.2.

2.4.2 Plane and Lines Equations of Target in Thermal Camera

To find the equations of the calibration target's plane and four edges in the thermal camera coordinate system, we first find the resistors in the thermal image. To do this we used the SimpleBlobDetector algorithm [17] to detect resistors as blobs in the thermal image. To remove blobs that are not resistors, We kept the top r_{num} closest blobs to the mass center of all blobs, where r_{num} is the number of resistors. This step is shown in Fig. 2.3(b). Each blob has a 2D location and diameter in the image. To find the sub-pixel location of checkerboard corners, we



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

Figure 2.3: a) Active calibration target in the scene. b) Thermal image after detecting blobs that contain resistors and discarding the rest of the image. c) Detected corners on the checkerboard. d) Detected lines on the edges of the calibration target.

found the pixel with the maximum value for each blob. Then we used the weighted average in its 3×3 neighborhood to find the sub-pixel location of the corner. We denote the i^{th} checkerboard corner point in the thermal image with $\boldsymbol{p}_i^{I,P}$. These points are depicted in Fig. 2.3(c) as blue dots. We selected the top three pixels with the highest values for each blob belonging to resistors located on the edges of the target. Unlike resistors situated behind small drilled holes, those on the edges are entirely visible to the camera. Selecting multiple high-value pixels on these resistors allows us to obtain points that are sufficiently close to the edges.

The six selected pixels of two resistors on each edge are used with a RANSAC algorithm [18] to find the line equation of the calibration target's edges in the thermal image. $\boldsymbol{l}_i^{I,E}$ and $\boldsymbol{p}_{ij}^{I,E}$ are line equation and 2D position of the j^{th} selected pixel of edge number i in the thermal image. $i \in \{1, 2, 3, 4\}$ since the target has four edges, and $j \in \{1, 2, ..., 6\}$. These lines are depicted in Fig. 2.3(d).

To find the rotation and translation of the thermal camera with respect to the calibration target, we used Perspective-n-Point with RANSAC [18, 19]. PnP with RANSAC was used because positioning the target with drastic skew can sometimes create anomalies in the corners' positions.

The output of PnP with RANSAC is used on $\boldsymbol{p}_i^{I,P}$ for all i to calculate the position of checkerboard corners in the thermal camera coordinates. These points are depicted by $\boldsymbol{p}_i^{C,P}$, where $\boldsymbol{p}_i^{C,P}$ is i^{th} checkerboard corner point in the camera coordinates. These points are used to find the calibration target's plane equation in the camera coordinates in the form of $n_1^C x + n_2^C y + n_3^C z + d^C = 0$, which we denote as $[\boldsymbol{n}^C; d^C]$. The plane equation is divided by a constant, so the length of the plane's Euclidean norm becomes equal to one. Therefore, $\|\boldsymbol{n}^C\|_2 = 1$.

We used the 3×3 camera matrix $\boldsymbol{K}$ that contains intrinsic parameters of the thermal camera and $\boldsymbol{p}_{ij}^{I,E}$ for all i and j to find points of calibration target's edges in the thermal camera coordinate system. We used $\boldsymbol{p}_{ij}^{C,E}$ to denote them, where j is j^{th} point on edge number i. To find $\boldsymbol{p}_{ij}^{C,E}$, $\boldsymbol{K}^{-1}[\boldsymbol{p}_{ij}^{I,E}; 1]$ is used to calculate the ray that passes from the center of camera coordinate through the point $\boldsymbol{p}_{ij}^{I,E}$ in the thermal image and $\boldsymbol{p}_{ij}^{C,E}$ equals the intersection of the ray with plane $[\boldsymbol{n}^C; d^C]$. We used $\boldsymbol{p}_{ij}^{C,E}$ with RANSAC to find the line equation of edges of the calibration target in the thermal camera coordinate system. The line equation for the i^{th} edge in the camera coordinates is denoted $[\boldsymbol{d}_i^{C,E}; \hat{\boldsymbol{p}}_i^{C,E}]$, where $\boldsymbol{d}_i^{C,E}$ is a 3-dimensional vector that shows the direction of the line, $\hat{\boldsymbol{p}}_i^{C,E}$ is a point on the line and $\|\boldsymbol{d}_i^{C,E}\|_2 = 1$.

2.4.3 Target Equations in LiDAR Coordinate System

To find the plane and line equation of the calibration target inside the LiDAR point cloud, we first find the calibration target. With distance thresholding, an area in front of the LiDAR in the vehicle setup is selected, and points on the floor and ceiling are removed. Then, a Sample Consensus (SAC) segmentation method [20] is used to find the calibration target.

Like Zhou et al. [21], we denoised LiDAR points and found points on the edges of the target. First, by using the RANSAC algorithm [18], we found the plane equation of the calibration target inside the point cloud. To reduce the noise of points, points are projected to the plane. Fig. 2.4(b) shows an example of this step. After that, we sorted points according to their value on the Z axis. By observing a sharp changes in the z value of points, each beam of LiDAR is found, as shown in Fig. 2.4(c). For points on each beam, we used the RANSAC algorithm [18] to find the line that passes through the beam's points. By projecting points on the line, points are denoised further. As depicted in Fig. 2.4(d), the endpoints of each beam are found by sorting the points with their y value. Endpoints with minimum and maximum values of y and z are used to determine endpoints on the left and right edges. The directions of all consecutive points on the left and right edges are calculated. Based on sharp changes in the directions of two consecutive vectors, edges on the left and right are divided into top and bottom edges. Fig. 2.4(e) illustrates points on each edge of the calibration target. Finally, as shown in Fig. 2.4(f), the line equation of each edge is calculated by using the RANSAC algorithm [18] on the edge's endpoints.

Similar to the previous section, $\boldsymbol{p}_{ij}^{L,E}$ is the j^{th} point on the i^{th} edge, and $\boldsymbol{p}_i^{L,P}$ is the i^{th} point on the plane of the calibration target in the point cloud. Also, $\bar{\boldsymbol{p}}^{L,P}$ and $\bar{\boldsymbol{p}}_i^{L,E}$ are centroids of points on the plane and i^{th} edge in the point cloud. We show the plane of the calibration target in the point cloud with $[\boldsymbol{n}^L; d^L]$, which $\boldsymbol{n}^L$ is the norm of the plane, and $\|\boldsymbol{n}^L\|_2$ equals to 1. $[\boldsymbol{d}_i^{L,E}; \hat{\boldsymbol{p}}_i^{L,E}]$ is the line equation of i^{th} edge of calibration target in the LiDAR point cloud. $\boldsymbol{d}_i^{L,E}$ is direction of the line, $\hat{\boldsymbol{p}}_i^{L,E}$ is a point that belongs to the line and $\|\boldsymbol{d}_i^{L,E}\|_2 = 1$.

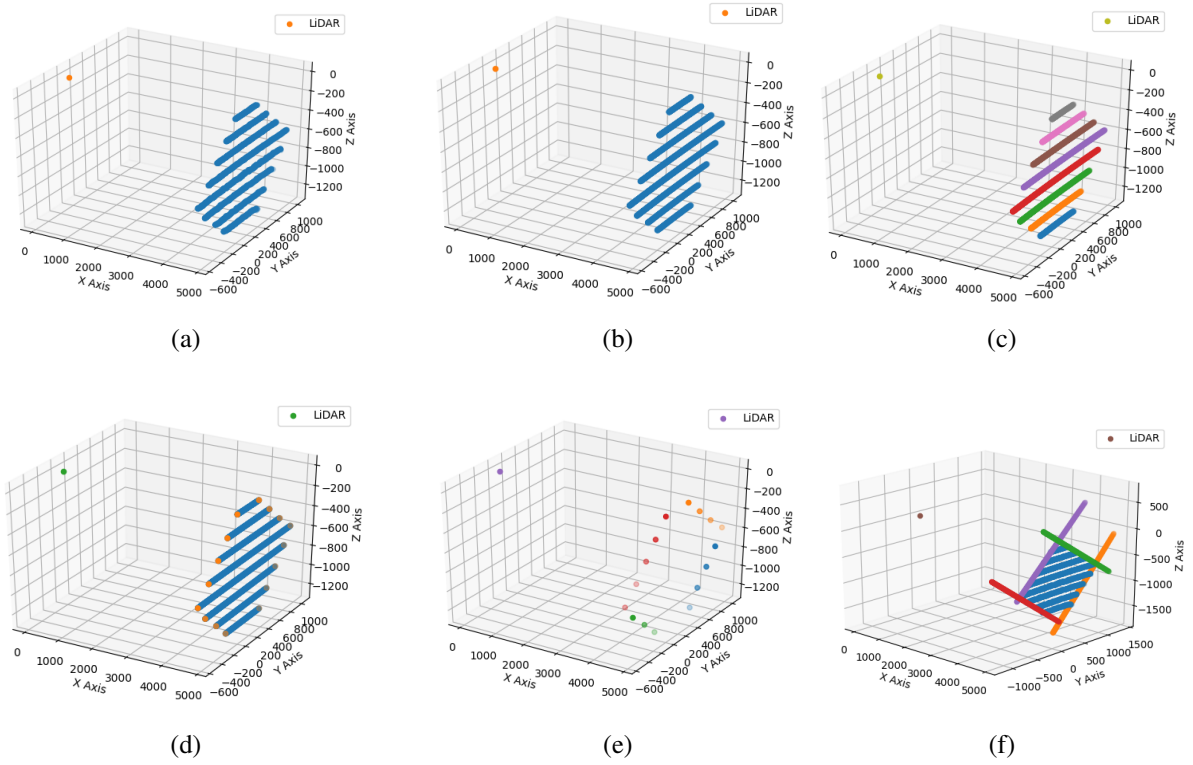


Figure 2.4: This shows the steps of finding line and plane equations in the LiDAR point cloud of the target. a) LiDAR points on the calibration target. b) Calibration target's points after projecting them on the target's plane. c) Finding different beams. d) Projecting points of each beam to the line that passes it and finding points at two ends of each beam. e) Determining points on each edge of the target. f) Lines that pass each edge of the target.

2.4.4 Optimization Problem to Find R and t

After calculating line and plane equations of the calibration target in the thermal camera and LiDAR coordinate systems, line and plane correspondences can be utilized to calculate rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$. To find $\boldsymbol{R}$ and $\boldsymbol{t}$, we used the formulation based on Zhou et al. [21].

$\tilde{\boldsymbol{R}}$ is an estimation of the rotation matrix and can be found by minimizing (2.1).

$$\tilde{\boldsymbol{R}} = \underset{\boldsymbol{R}}{\operatorname{argmin}}\Big(\sum_{i=1}^{4} \|\boldsymbol{R}\boldsymbol{d}_i^{L,E} - \boldsymbol{d}_i^{C,E}\|_2^2\Big) + \|\boldsymbol{R}\boldsymbol{n}^L - \boldsymbol{n}^C\|_2^2. \tag{2.1}$$

Equation (2.1) is independent of $\boldsymbol{t}$ and estimates the rotation matrix by minimizing two terms. The first term is the distance between the direction vector of edges in the camera coordinate and the converted direction of edges from the LiDAR coordinate to the camera coordinate. The second term does the same to norm vectors in camera and LiDAR coordinate systems. According to Arun et al. [22], it has the following closed-form solution. Define $\boldsymbol{M}_L = [\boldsymbol{n}^L, \boldsymbol{d}_1^L, \boldsymbol{d}_2^L, \boldsymbol{d}_3^L, \boldsymbol{d}_4^L]$ and $\boldsymbol{M}_C = [\boldsymbol{n}^C, \boldsymbol{d}_1^C, \boldsymbol{d}_2^C, \boldsymbol{d}_3^C, \boldsymbol{d}_4^C]$, estimation of the rotation matrix can

be obtained by $\tilde{\boldsymbol{R}} = \boldsymbol{V}\boldsymbol{U}^T$, which $\boldsymbol{U}$ and $\boldsymbol{V}$ are result of SVD decomposition [23] of matrix $\boldsymbol{M}_L\boldsymbol{M}_C^T$ in form of $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$. For simplification, we assumed $\boldsymbol{A}_i = \boldsymbol{I} - \boldsymbol{d}_i^{C,E}(\boldsymbol{d}_i^{C,E})^T$, where $\boldsymbol{I}$ is the identity matrix. This results in the linear system in (2.2).

$$\begin{aligned} \boldsymbol{n}^C \cdot \boldsymbol{t} &= -\boldsymbol{n}^C \cdot \tilde{\boldsymbol{R}}\bar{\boldsymbol{p}}^{L,P} - d^C, \\ \boldsymbol{A}_i\boldsymbol{t} &= -\boldsymbol{A}_i(\tilde{\boldsymbol{R}}\bar{\boldsymbol{p}}^{L,E} - \hat{\boldsymbol{p}}_i^{C,E}) \text{ for } i = 1, \cdots, 4. \end{aligned} \tag{2.2}$$

The linear system is in the form of $\boldsymbol{H}\boldsymbol{t} = \boldsymbol{v}$, in which $\boldsymbol{H}$ is the matrix on the left side of the linear equation and $\boldsymbol{v}$ is the vector on the right side of the equation. An estimate for the translation vector can be obtained according to (2.3):

$$\tilde{\boldsymbol{t}} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{v}. \tag{2.3}$$

After obtaining initial estimates $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{t}}$, the rotation matrix and translation vector, they can be jointly optimized by the cost function in (2.4). $N^{L,P}$ and $N_i^{L,E}$ are the number of points in the point cloud of the plane and i^{th} edge. Since we want to keep $\boldsymbol{R}$ as an orthonormal matrix, instead of optimizing $\boldsymbol{R}$, we optimize its corresponding Rodrigues' rotation vector, which can be obtained by OpenCV's *Rodrigues* function [17]. In the equation, $R(\boldsymbol{r})$ is a function that calculates the rotation matrix from the rotation vector $\boldsymbol{r}$.

$$\begin{aligned} (\boldsymbol{r}, \boldsymbol{t}) = \underset{\boldsymbol{r},\boldsymbol{t}}{\operatorname{argmin}} \frac{1}{N^{L,P}} \sum_{m=1}^{N^{L,P}} \|\boldsymbol{n}^C \cdot (R(\boldsymbol{r})\boldsymbol{p}_m^{L,P} + \boldsymbol{t}) + d^C\|^2 + \\ \sum_{i=1}^{4} \frac{1}{N_i^{L,E}} \sum_{k=1}^{N_i^{L,E}} \|\boldsymbol{A}_i(R(\boldsymbol{r})\boldsymbol{p}_{ik}^{L,E} - \hat{\boldsymbol{p}}_i^{C,E} + \boldsymbol{t})\|^2. \end{aligned} \tag{2.4}$$

Although one pose is enough for the algorithm, more poses can be used. For other poses, the obtained $\boldsymbol{R}$ and $\boldsymbol{t}$ are used to further refine the rotation matrix and translation vector using (2.5). When the calibration target is placed in front of the thermal camera at a considerable angle, detecting corners on the checkerboard can lead to inaccuracy in determining their position or a complete failure in detecting them. Therefore, in case of more than one pose, we defined the optimization problem in (2.5), which is not based on the location of corners on the checkerboard. It only uses the line equation of the calibration target in thermal images and the projected points of the edges of the target in the LiDAR point cloud into the thermal images. It tries to align the edges of the calibration target in the thermal image to the projected points.

$$\begin{aligned} (\boldsymbol{r}, \boldsymbol{t}) = \underset{\boldsymbol{r},\boldsymbol{t}}{\operatorname{argmin}} \sum_{j=1}^{N_{pose}} \sum_{i=1}^{4} \sum_{m=1}^{N_i^{L,E}} \|\boldsymbol{\phi}_i^{I,E}(\boldsymbol{p}_{im}^{L,E}) - \\ \boldsymbol{K}(R(\boldsymbol{r})\boldsymbol{p}_{im}^{L,E} + \boldsymbol{t})\|. \end{aligned} \tag{2.5}$$

N_{pose} is the number of new poses, and $\boldsymbol{\phi}_i^{I,E}(\boldsymbol{p}_{im}^{L,E})$ is a function that returns the closest point on the line number i in the thermal image to the projection of point $\boldsymbol{p}_{im}^{L,E}$ from LiDAR coordinate system to image coordinate. Both (2.4) and (2.5) are solved with LevenbergMarquardt optimizer implemented in *SciPy* [24].

Table 2.1: Translation and Rotation Errors of the proposed algorithm for different levels of uniform noise for one pose.

	Random uniform noise, (displacement in point cloud, displacement in resistors)									
	(3, 0.1)	(3, 0.2)	(3, 0.3)	(3, 0.4)	(3, 0.5)	(3, 0.6)	(3, 0.7)	(3, 0.8)	(3, 0.9)	(3, 1.0)
Average T Error (%)	6.8812	7.1679	8.4155	10.8244	12.2642	14.7538	17.2930	17.3794	19.0472	22.0898
Average R Error (Radian)	0.006245	0.006402	0.007673	0.009240	0.010497	0.012267	0.014337	0.014320	0.015735	0.018170
Median T Error (%)	6.8931	6.2905	8.3368	10.9995	11.4105	14.5892	15.6268	16.5950	18.0372	20.2854
Median R Error (Radian)	0.006025	0.005731	0.007504	0.009029	0.009777	0.011893	0.013194	0.013605	0.014763	0.016671

2.5 Experiments

In our setup, a thermal camera and a LiDAR sensor were installed on the vehicle's roof. We used a FLIR Boson thermal camera with a known camera matrix that provides images with 640×512 resolution. Also, we used a 16-beam Velodyne VLP-16 LiDAR. The calibration target has 114.2×115.0 cm dimensions, and it has 12 resistors on its back and two resistors on each edge. To heat up resistors we connected them to a power source of 33 V and 2.75 A.

Since it is not possible to obtain ground truth for real thermal images and LiDAR point cloud pairs, we created a simulator to evaluate the proposed method. The simulator gets the properties of a LiDAR sensor, the camera matrix of a thermal camera, the calibration target's details, the location, and direction of calibration targets and thermal camera with respect to the LiDAR sensor, and the amount of noise in the position of resistors in thermal image and points in the point cloud. These details are used to generate synthetic image point cloud pairs. For creating noise, we used uniform noise.

To calculate the performance of the proposed method with one pose, we designed ten scenarios. In all of them, we used a uniform noise that leads to a maximum displacement of points in point clouds up to 3 cm, and we considered ten values in pixel $[0.1, 0.2, \cdots, 1.0]$ for the uniform noise displacement of points in thermal images. We selected 3 cm and the ten value according to our LiDAR manual and analysis of thermal images. For each scenario, 100 simulations were done and the average and median error for the translation vector and rotation matrix were obtained. In simulations, for generating a pose, the target was positioned 4 meters to 7 meters from the LiDAR, in a diamond shape like Fig. 2.4(f). Then, it was rotated by a random value in the range $[-15, 15]$ degree around x axis and $[-20, 20]$ around other axes. Translation error is calculated according to $\frac{\|\boldsymbol{t}_{est}-\boldsymbol{t}_{gt}\|_2 \times 100}{\|\boldsymbol{t}_{gt}\|_2}$, which respectively est and gt mean output of the algorithm and the ground truth value. To calculate the error of obtained rotation matrix $\boldsymbol{R}_{est}$ with respect to ground truth rotation matrix $\boldsymbol{R}_{gt}$, we used the method in [25]. First $\boldsymbol{R}_{est}\boldsymbol{R}_{gt}^{-1}$ is calculated which gives a new 3×3 matrix. Then OpenCV's *Rodrigues* function [17] is used on the new matrix to calculate its rotation axis and rotation angle. The rotation error is equal to the rotation angle. The translation and rotation errors of the proposed method for one pose for different scenarios are presented in Table 2.1. As shown in the table, by increasing the level of noise, both errors in translation and rotation increase. The rotation error's growth is slow, while the effect of adding noise is more tangible on the translation error.

To show the effectiveness of our proposed method with one pose, we compared it with Krishnan, et al. [11], which uses multiple poses to obtain the translation vector and rotation matrix between a thermal camera and a LiDAR. The method in Krishnan, et al. [11] obtains promising results and we can use it with our target to obtain results. Again, we used the

Table 2.2: Translation and Rotation Errors of the proposed algorithm and Krishnan, et al for the different number of poses.

		Number of Poses							
		1	2	3	4	5	6	7	8
Krishnan, et al.	Average T Error (%)	29.43754	19.3683	17.3752	16.2909	14.3328	12.5100	11.7089	10.3241
	Average R Error (Radian)	0.360947	0.032370	0.019313	0.015057	0.009150	0.008521	0.007754	0.007494
	Median T Error (%)	29.4458	19.3301	17.0115	16.1566	13.5137	12.2103	11.4965	10.2248
	Median R Error (Radian)	0.362786	0.033491	0.018794	0.015642	0.009497	0.008080	0.007638	0.007294
Proposed Method	Average T Error (%)	10.8244	10.7423	10.4303	10.3739	10.1134	9.9614	9.8443	9.7565
	Average R Error (Radian)	0.009240	0.010263	0.009212	0.008736	0.008916	0.008123	0.007784	0.007548
	Median T Error (%)	10.9995	10.7336	10.4208	10.2049	10.0452	9.9190	9.6904	9.7449
	Median R Error (Radian)	0.009029	0.009435	0.008713	0.008495	0.008387	0.007627	0.007643	0.007523

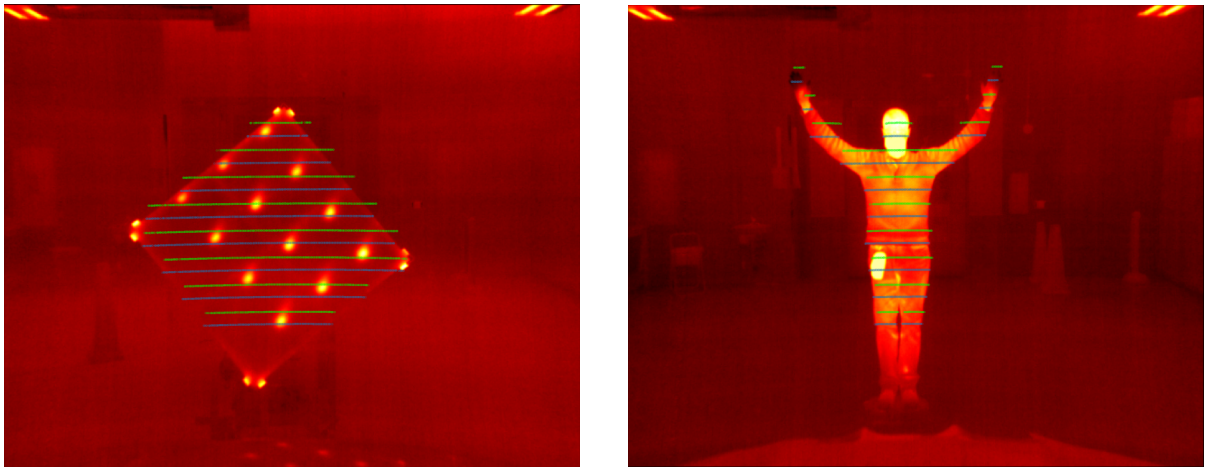
simulator and we examined eight scenarios from one to eight poses. For each scenario, we repeated the experiment 100 times. In all simulations, we used uniform noise with a maximum 3 cm displacement for points in point clouds and 0.4 pixel displacement for resistors in thermal images. The results are presented in Table 2.2. As shown in the table, Krishnan, et al. [11] needed eight poses to reach the translation error of the proposed method with one pose. Also, by using five poses, it reached the rotation error of the proposed algorithm with one pose. Moreover, by using equation 2.5, the proposed method can use more than one pose. As shown in Table 2.2, using more poses led to lower translation and rotation errors. For example, the proposed method with eight poses obtains 1.05% less error in translation and 0.001692 radian less error in rotation with respect to one pose.

In Fig. 2.5, we compared the results of the proposed method with the results of Krishnan, et al. [11] on two thermal image-point cloud pairs. Blue dots are the result of the proposed model obtained with one pose and the green dots are the outcome of Krishnan, et al. calculated with three poses. As can be seen in the figure, the proposed method obtained better outcomes. In Fig. 2.5(a), the calibration target's green projected points to the thermal image have tangible distances to the edges of the calibration target. Similarly, in Fig. 2.5(b), the endpoints of green lines are clearly off the edges of the person.

Fig. 2.6 contains a person in front of the thermal camera and LiDAR sensor with six different distances and directions with the respect to the sensors. some parts of the six images are magnified for better visualization. As can be seen, the edges of the person's points in the point clouds are highly aligned with the edges of the person in the thermal images.

2.6 Conclusion and Future Work

In this work, we presented an algorithm for the extrinsic calibration of a thermal camera and a LiDAR sensor. It is automatic and there is no need for human interaction. Also, it works with one pose and sparse LiDARs. We introduced our design for an active calibration target that is visible in thermal camera images. We explained how to find the line and plane equations of the calibration target's edges and plane in the thermal image and camera coordinates. We then described the process of detecting and denoising the calibration target in the point cloud and extracting the line and plane equations. After that, we described optimization problems that can be used to obtain the rotation matrix and translation vector with one and more poses. Finally, we conducted different experiments to show the performance of the proposed method. For



(a) (b)

Figure 2.5: Projected LiDAR points of the calibration target and a person on their corresponding thermal images. The blue dots show the result of our calibration target with one pose and the green dots belong to the results of Krishnan, et al. [11] with three poses.

future work, we want to expand the optimization problems in order to also do online calibration by aligning the edges of the environment in the thermal images and LiDAR point clouds to address the problem of slight changes in the position and direction of sensors caused by car movement.

2.7 Acknowledgement

This research has been supported by the National Research Council of Canadas Artificial Intelligence for Logistics Program and the Natural Sciences and Engineering Research Council of Canada. Also, the first two authors contributed equally to this work.

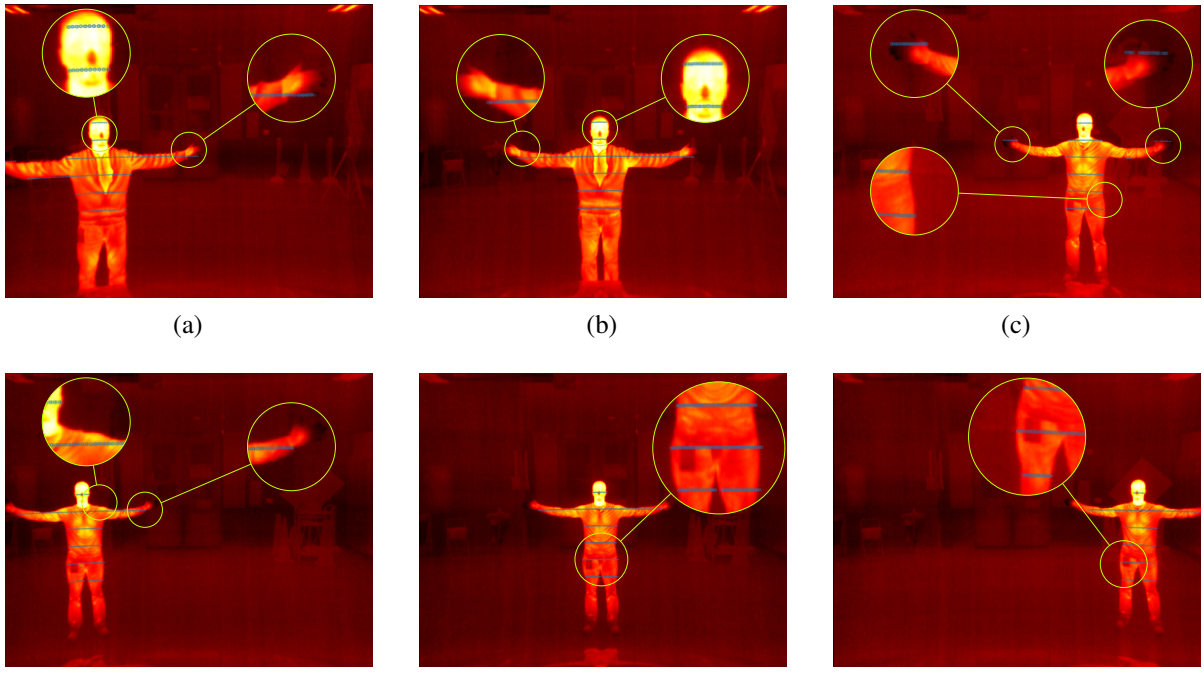
References

[1] D. T. Le, H. Shi, H. Rezatofighi, and J. Cai, "Accurate and real-time 3d pedestrian detection using an efficient attentive pillar network," *IEEE Robotics and Automation Letters*, 2022.

[2] Y. He, H. Yu, X. Liu, Z. Yang, W. Sun, Y. Wang, Q. Fu, Y. Zou, and A. Mian, "Deep learning based 3d segmentation: A survey," *arXiv preprint arXiv:2103.05423*, 2021.

[3] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *arXiv preprint arXiv:2207.00254*, 2022.

[4] K. Bayoudh, R. Knani, F. Hamdaoui, and A. Mtibaa, "A survey on deep multimodal



(a) (b) (c)

(d) (e) (f)

Figure 2.6: Projected LiDAR points of a person with various distances and directions to the sensors onto the corresponding thermal images. The projections are calculated by the output of the proposed algorithm with one pose. For better visualization, some parts of images are magnified.

learning for computer vision: advances, trends, applications, and datasets," *The Visual Computer*, vol. 38, no. 8, pp. 2939–2970, 2022.

[5] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine vision and applications*, vol. 25, no. 1, pp. 245–262, 2014.

[6] B. Miethig, A. Liu, S. Habibi, and M. v. Mohrenschildt, "Leveraging thermal imaging for autonomous driving," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–5, IEEE, 2019.

[7] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[8] N. Tsoulias, S. Jörissen, and A. Nüchter, "An approach for monitoring temperature on fruit surface by means of thermal point cloud," *MethodsX*, vol. 9, p. 101712, 2022.

[9] Y. Yue, C. Yang, J. Zhang, M. Wen, Z. Wu, H. Zhang, and D. Wang, "Day and night collaborative dynamic mapping in unstructured environment based on multimodal sensors," in *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 2981–2987, IEEE, 2020.

[10] D. Borrmann, *Multi-modal 3D mapping-Combining 3D point clouds with thermal and color information*. Universität Würzburg, 2018.

[11] A. K. Krishnan, B. Stinnett, and S. Saripalli, "Cross-calibration of rgb and thermal cameras with a lidar," in *IROS 2015 Workshop on Alternative Sensing for Robot Perception*, 2015.

[12] S. Azam, F. Munir, A. M. Sheri, Y. Ko, I. Hussain, and M. Jeon, "Data fusion of lidar and thermal camera for autonomous driving," in *Applied Industrial Optics: Spectroscopy, Imaging and Metrology*, pp. T2A–5, Optical Society of America, 2019.

[13] A. K. Krishnan and S. Saripalli, "Cross-calibration of rgb and thermal cameras with a lidar for rgb-depth-thermal mapping," *Unmanned Systems*, vol. 5, no. 02, pp. 59–78, 2017.

[14] J. Zhang, P. Siritanawan, Y. Yue, C. Yang, M. Wen, and D. Wang, "A two-step method for extrinsic calibration between a sparse 3d lidar and a thermal camera," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1039–1044, IEEE, 2018.

[15] T. Fu, H. Yu, Y. Hu, and S. Scherer, "Targetless extrinsic calibration of stereo cameras, thermal cameras, and laser sensors in the wild," *arXiv preprint arXiv:2109.13414*, 2021.

[16] S. Mharolkar, J. Zhang, G. Peng, Y. Liu, and D. Wang, "Rgbdtcalibnet: End-to-end online extrinsic calibration between a 3d lidar, an rgb camera and a thermal camera," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3577–3582, IEEE, 2022.

[17] G. Bradski, "The opencv library.," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[19] L. Quan and Z. Lan, "Linear n-point camera pose determination," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 8, pp. 774–780, 1999.

[20] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*, pp. 1–4, IEEE, 2011.

[21] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5562–5569, IEEE, 2018.

[22] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[23] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on automatic control*, vol. 25, no. 2, pp. 164–176, 1980.

[24] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[25] Z. Kukelova, J. Heller, and A. Fitzgibbon, "Efficient intersection of three quadrics and applications in computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1799–1808, 2016.

Chapter 3

Extrinsic Calibration of Thermal Camera and 3D LiDAR Sensor via Human Matching in Both Modalities during Sensor Setup Movement

This Chapter is a reformatted version of the following article:

Dalirani, Farhad, and Mahmoud R. El-Sakka. 2024. "Extrinsic Calibration of Thermal Camera and 3D LiDAR Sensor via Human Matching in Both Modalities during Sensor Setup Movement" Sensors 24, no. 2: 669. https://doi.org/10.3390/s24020669

3.1 Abstract

LiDAR sensors, pivotal in various fields like agriculture and robotics for tasks such as 3D object detection and map creation, are increasingly coupled with thermal cameras to harness heat information. This combination proves particularly effective in adverse conditions like darkness and rain. Ensuring seamless fusion between the sensors necessitates precise extrinsic calibration. Our innovative calibration method leverages human presence during sensor setup movements, eliminating the reliance on dedicated calibration targets. It optimizes extrinsic parameters by employing a novel evolutionary algorithm on a specifically designed loss function that measures human alignment across modalities. Our approach showcases a notable 4.43% improvement in the loss over extrinsic parameters obtained from target-based calibration in the FieldSAFE dataset. This advancement reduces costs related to target creation, saves time in diverse pose collection, mitigates repetitive calibration efforts amid sensor drift or setting changes, and broadens accessibility by obviating the need for specific targets. The adaptability of our method in various environments, like urban streets or expansive farm fields, stems from leveraging the ubiquitous presence of humans. Our method presents an efficient, cost-effective, and readily applicable means of extrinsic calibration, enhancing sensor fusion capabilities in the critical fields reliant on precise and robust data acquisition.

3.2 Introduction

The challenges encountered in the realm of computer vision often present a high degree of complexity. To address these complexities effectively, it is common to employ a range of sensors that work collaboratively to augment the information gathered from the scene and the objects within it. The integration of diverse sensors frequently leads to solutions that not only enhance accuracy but also bolster robustness [1]. 3D LiDAR (Light Detection and Ranging) sensors and thermal cameras, valued for their accurate point clouds and heat information receptivity, are gaining attention for use in data fusion. Extrinsically calibrating these sensors, each with its own coordinate system, is essential for their accurate data integration.

3D LiDAR sensors have emerged as one of the most popular sensors in fields such as agriculture, autonomous vehicles, and robotics. Some of their applications include odometry and SLAM (Simultaneous Localization and Mapping) [2] in robotics, semantic scene understanding [3], and 3D object detection [4] in self-driving cars, forest attribute estimation [5], and precision farming [6].

A LiDAR sensor produces a 3D point cloud where each point is precisely defined by its x, y, and z LIDAR coordinates. Furthermore, this point cloud includes data regarding the strength of the reflected laser pulse at each point. Consequently, a LiDAR sensor does not offer supplementary information for individual points, such as color. However, when we integrate LiDAR data with additional data from other sensors, it becomes feasible to improve performance across a range of tasks. For instance, in the study by Xu et al. [7], LiDAR data was combined with data from an RGB camera to enhance 3D object detection.

Thermal cameras have gained attention as alternative sensors to fuse with LiDAR data due to their ability to create high-quality images based on temperature differences in objects and their surroundings, even in adverse conditions like darkness, snow, dust, smoke, fog, and rain [8]. Because thermal cameras can capture spectra that other sensors like visual light cameras cannot, they have numerous applications in agriculture, security, healthcare, the food industry, aerospace, and the defense industry, among others [9, 10].

Combining data from 3D LiDAR sensors and thermal cameras can yield the benefits of both sensors simultaneously. By leveraging both 3D spatial information and heat signatures, a more comprehensive and accurate representation of the environment is achieved. This integration enhances overall situational awareness, robustness, and accuracy across many tasks, especially when compared with the use of either technology in isolation. For example, in any application involving the heat data of a scene and its objects, it can be augmented with LiDAR data to obtain the 3D location of various elements within the scene. For instance, when measuring the attributes of fruits on a tree or detecting pedestrians in the streets, leveraging the 3D location can provide accurate positioning information to allow the robotic arm to harvest the fruit or enable the control component in an autonomous vehicle pipeline to take necessary actions to avoid colliding with pedestrians. The following are some of the existing applications of combining these two sensors for various purposes. Kragh et al. [11] instrumented a tractor with multi-modal sensors, including LiDAR and a thermal camera, to detect static and moving obstacles, including humans, to increase safety during operations in the field. Choi et al. [12] developed a multi-modal dataset including LiDAR and thermal camera data for studying various tasks, including drivable region detection, object detection, localization, and more, in the context of assisted and autonomous driving, both during the day and at night. Shin et al. [13]

used LiDAR and thermal cameras to investigate depth estimation in challenging lighting and weather conditions for autonomous vehicles. In their research, Yin et al. [14] built a ground robot instrumented with various sensors, including a thermal camera and LiDAR. They argued that visual SLAM with an RGB camera is ineffective in low visibility situations such as darkness and smoke, and using a thermal camera can address some of these challenges. Tsoulias et al. [15] used a thermal camera and LiDAR to create a 3D thermal point cloud to detect disorders caused by solar radiation on fruit surfaces. Yue et al. [16] incorporated a thermal camera alongside LiDAR to enhance the robots' ability to create a map of the environment, both during the day and at night.

A thermal camera and LiDAR have their own coordinate systems. To use data from both modalities, these two sensors should be extrinsically calibrated. Here, extrinsic calibration is the task of finding the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ to express the coordinate of a point in the LiDAR's coordinate system in the camera's coordinate system. $\mathbf{R}$ is an orthogonal 3×3 matrix that describes rotation in 3D space, and $\mathbf{t}$ is a 3D vector that represents a shift in 3D space. After obtaining the extrinsic parameters, the point $\mathbf{p}^C$ in the thermal camera system corresponding to the LiDAR point $\mathbf{p}^L$ in the LiDAR coordinate system can be obtained according to $\mathbf{p}^C = \mathbf{R}\mathbf{p}^L + \mathbf{t}$.

In the extrinsic calibration of visible light cameras and LiDAR, various types of targets, including checkerboard targets [17], are typically employed. Nonetheless, these targets are not visible to a thermal camera. To adapt them for the extrinsic calibration of a thermal camera and LiDAR, these targets can be modified by crafting them from various heat-conductive materials and then either pre-cooling or heating them before use [18], or by incorporating heat-generating electrical elements such as light bulbs [15]. Using these adopted targets comes with some drawbacks. Creating them is both challenging and expensive. Using them in situations where the sensor setup frequently changes or sensor drift occurs can be cumbersome. Additionally, over time, heating leaks can occur from the heat-generating elements, or their temperature can become similar to the surrounding environment, rendering them ineffective for use, and getting them operational again can take some time.

The mentioned difficulties encountered while working with calibration targets motivated our proposed method. We propose a novel method for the extrinsic calibration of a thermal camera and a LiDAR without using a dedicated calibration target based on matching segmented people in both modalities during the movement of the sensor setup in environments such as farm fields or streets that contain humans. The extrinsic parameters are obtained by optimizing a designed loss function that measures the alignment of human masks in both modalities. This is achieved using a novel optimization algorithm based on evolutionary algorithms. We present two versions of our algorithm. The first version disregards input noise, while the second version seeks to mitigate the effects of noisy inputs. This innovative approach minimizes the expenses associated with the creation of calibration targets for thermal cameras and eliminates the often labor-intensive and time-consuming process of collecting diverse poses for calibration targets, particularly in the context of autonomous vehicles where positioning a large target at various angles and heights can be challenging. It also addresses the issue of the repetitive calibration efforts required when sensor drift or setting changes occur, making the process more efficient. Additionally, it enhances the accessibility of 3D LiDAR and thermal camera fusion by eliminating the necessity for specific targets.

The remainder of the paper is structured as follows: In Section 3.3, we provide an overview

and examination of prior research. Section 3.4 outlines the cross-calibration algorithm. Section 3.5 showcases our experiments and their outcomes on the FieldSAFE [11] and MS^2 [13] datasets. Lastly, Section 3.6 serves as the conclusion of our paper and outlines potential avenues for future research.

3.3 Related Work

Some studies have explored the calibration of thermal cameras and LiDAR systems using various target-based approaches. These methods typically involve utilizing the known specifications of the calibration targets and minimizing a cost function to establish the extrinsic parameters that align these specifications across both sensor modalities. Krishnan et al. [18] used a checkerboard target made of laser-cut black and white melamine with different heat conductivity. They placed it in front of the sun for approximately one hour to enable the detection of checkerboard corners by a thermal camera. A user manually selected the four outer corners of the target inside the thermal image, and to detect the calibration target within the point cloud, they used a region-growing algorithm. They determined the rotation matrix and translation vector by attempting to minimize the distance between the points on the edges of the target in the LiDAR point cloud and their nearest points on the edges of the target in the thermal image. Their algorithm requires a good initial rotation, translation, and several poses. Krishnan et al. [19] developed a cross-calibration method that involved the creation of a target by cutting a circular hole in white cardboard with a precisely known radius. They utilized a damp black cloth as the background, which improved the circle's visibility in the thermal camera. The process started by manually selecting a pixel in the circle for a region-growing algorithm to segment it in the image. Likewise, the user picked a point on the cardboard to locate the target in the point cloud. They captured multiple poses for cross-calibration. In each pair, they projected the circle's edges from the point cloud onto the thermal image. Finally, they solved an optimization problem of aligning the thermal camera's circle edges with the projected edges, ensuring precise calibration. Borrmann et al. [20] devised a calibration target visible in thermal cameras by creating a dot pattern on a board using light bulbs. In the calibration process, they collected multiple pairs of images and their corresponding point clouds. For each of these pairs, they precisely determined the locations of the light bulbs in in both modalities. To establish the positions of the light bulbs within the LiDAR coordinate system, they located the calibration target within the point cloud data. Leveraging the well-defined geometry of their calibration target, they computed the positions of the light bulbs in the LiDAR coordinate system. Subsequently, for each image-point cloud pair, they mapped the positions of the light bulbs from the point cloud to the thermal image. Finally, to determine the extrinsic parameters, they solved an optimization problem aimed at minimizing the disparity between the light bulb positions in the thermal image and their projected positions in the point cloud. In the proposed method of Dalirani et al. [21], an active checkerboard target with embedded resistors for generating heat was used, and extrinsic parameters between both the thermal and LiDAR sensors were obtained from the correspondence of lines and plane equations of the calibration target in the image and point cloud pair. Zhang et al. [22] created four equally spaced circles on an electric blanket. They identified these circles in both modalities and optimized the extrinsic parameters by minimizing the 2D re-projection error.

In many studies, when using a thermal camera and LiDAR data, instead of directly performing extrinsic calibration between the thermal camera and LiDAR, each of them is extrinsically calibrated with another sensor, such as an RGB camera, for example. Then, the two sets of obtained extrinsic calibration parameters are used to determine $\mathbf{R}$ and $\mathbf{t}$ between the thermal camera and LiDAR. Azam et al. [23] employed a thermal camera capable of providing both visual and thermal images, along with extrinsic parameters linking these two types of images. They applied an established RGB camera-LiDAR calibration technique to achieve extrinsic calibration between the visual camera and LiDAR. Subsequently, they utilized this knowledge, in conjunction with extrinsic calibration parameters connecting the visual and thermal cameras, to derive the transformation between the thermal camera and the LiDAR. Similarly, Zhang et al. [24] divided the calibration process for the thermal camera and LiDAR into two sequential steps. In the FieldSAFE dataset [11], a similar method [25] was employed to determine the rotation and translation between sensors. They calculated the extrinsic parameters between the LiDAR and the stereo vision system using the iterative closest point algorithm [26]. To calibrate the stereo vision system and the thermal camera, they constructed a checkerboard with both copper and non-copper materials and attached 60 resistors to generate heat. Subsequently, through post-processing, they were able to employ a regular cross-calibration tool for two visual light cameras to extrinsically calibrate the RGB and thermal cameras. Finally, by comparing the two solutions, the parameters between the thermal and LiDAR sensors could be obtained. In the MS^2 dataset [13, 27], for their instrumented car, they established extrinsic calibration parameters between all sensors, including the thermal cameras and LiDAR, in conjunction with the NIR camera. The rotation and translation between other sensors can be obtained by using these extrinsic parameters with the NIR camera. To calibrate the NIR and thermal cameras, they used a 2×2 AprilTag board with metallic tape attached to it.

In another approach, targetless extrinsic calibration methods do not use a target but instead employ feature alignment in both modalities. Fu et al. [28] introduced a targetless extrinsic calibration method that calibrates a stereo visual camera system, a thermal camera, and a LiDAR sensor. In their method, first, the transformation between LiDAR and the stereo system is estimated. Then, the thermal camera is calibrated with the left camera in the stereo system by simultaneously using data from LiDAR and the left stereo camera. By establishing transformations between the thermal camera and the stereo system, as well as between LiDAR and the stereo system, the transformation between LiDAR and the thermal camera can be calculated. Their method optimizes extrinsic parameters by maximizing the alignment of edges in the three modalities. To derive edges from the LiDAR point cloud, they employed the horizontal depth difference and utilized the Canny edge detector [29] to detect edges in the thermal camera and the left stereo camera. Their method requires sufficient edge features in the modalities and a rough initial guess for optimization. Mharolkar et al. [30] proposed a targetless cross-calibration method for visual and thermal cameras with LiDAR sensors by utilizing a deep neural network. Instead of employing hand-crafted features, they utilized multi-level features from their network and used these extracted feature maps to regress extrinsic parameters. To train the network for calibrating the visual camera and LiDAR on the KITTI360 dataset [31], they utilized 44,595 image-point cloud pairs. For training the network for calibrating the thermal camera, they employed pre-trained weights for the visual camera and LiDAR and trained the model on their thermal camera and LiDAR dataset, consisting of 8075 thermal images and LiDAR pairs. Additionally, for a new set of sensors, the network should be re-trained.

Our proposed method does not require a target and optimizes extrinsic parameters during the movement of the sensor setup in an environment with human presence by aligning segmented people in both modalities. Importantly, it does not rely on the presence of rich edge features, making it applicable even in environments like farm fields, which often lack distinct edges. Moreover, it does not demand a precise initial solution, enhancing its versatility and ease of use. To the best of our knowledge, in the literature on the extrinsic calibration of thermal cameras and 3D LiDAR sensors, our proposed method represents a novel approach distinct from any existing methodologies. To date, no other method for these sensors has demonstrated the same innovative techniques employed in our study.

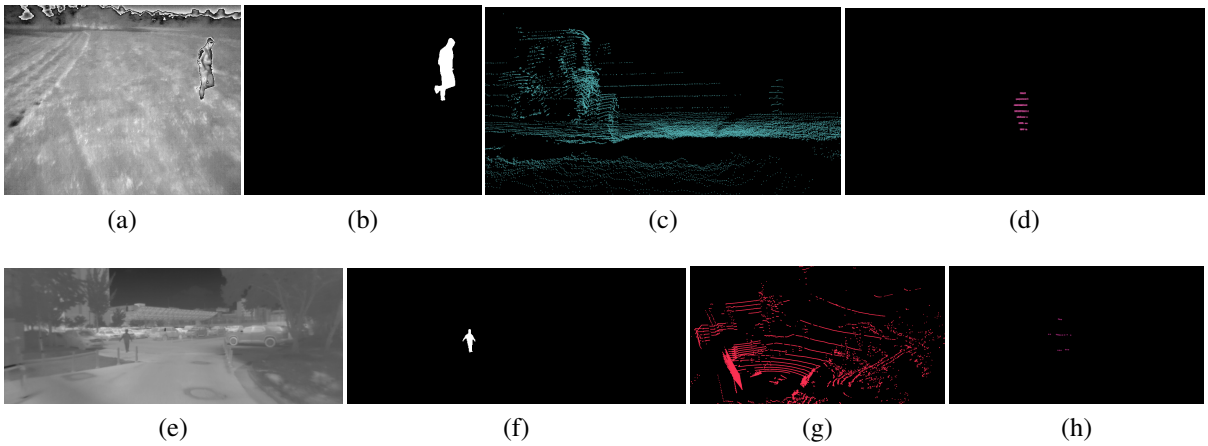
3.4 Methodology

In this paper, we propose an extrinsic calibration method for determining rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ between a thermal camera and a 3D LiDAR sensor without the need for a target. Our method relies on matching segmented humans in both modalities during the movement of the sensor setup. In the following, we will explain the steps of the proposed method, including data collection, formulating the problem, designing a cost function, and the method for optimizing the extrinsic parameters by minimizing the cost function.

3.4.1 Data Collection

While the thermal camera and LiDAR sensor setup is in motion on a moving vehicle, such as a tractor, robot, or car, in various environments like streets and farm fields, the dataset D is created by capturing several frames at different time points, denoted as t_1, t_2, ..., $t_{N_{pose}}$, for both modalities. N_{pose} denotes the number of captured frames. At each time t_i, both the LiDAR and thermal camera capture the scene simultaneously, producing the captured image and point cloud, which we denote as I_{t_i} and P_{t_i}, respectively. Given that our method relies on matching humans in both modalities, it is essential that each image and point cloud pair in the dataset contains human subjects, and the number of humans should be equal, which may vary from one or more individuals. As the number of humans increases, the likelihood of overlapping also rises, introducing more errors in segmenting humans in both data modalities. Therefore, only frames containing between one and a small number, denoted as H_{max}, of humans are retained.

In the beginning, the dataset D is empty. During the movement of the sensor setup in the environment at the moment t_i, a thermal image and a point cloud are captured. Then, an off-the-shelf person segmentation model and a human detector are applied to the captured image and point cloud, respectively. If the number of humans found in both modalities is equal and is greater than zero, the image and point cloud pair are kept; otherwise, it is discarded. In the provided pair, $I^h_{t_i}$ is generated by assigning a value of one to pixels within the human masks and zero to pixels outside the masks in the thermal image. Similarly, $P^h_{t_i}$ is produced by retaining the points in the point cloud that correspond to humans and removing all other points. Subsequently, the $I^h_{t_i}$ and $P^h_{t_i}$ pair is included in the dataset D. This process continues until the dataset D reaches a specific size, denoted as N_{pose}. Two examples from the FieldSAFE [11] and MS^2 [13] datasets are shown in Figure 3.1.



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

(e) (f) (g) (h)

Figure 3.1: Images (**a**–**d**) are sourced from the FieldSAFE dataset [11], whereas images (**e**–**h**) are obtained from the MS^2 dataset [13]. In each row, the images from left to right show a thermal image (I_{t_i}), the segmentation mask for human(s) in the thermal image ($I^h_{t_i}$), a shot from its corresponding point cloud (P_{t_i}), and a shot from the corresponding point cloud with only human(s) points ($P^h_{t_i}$).

In collected data pairs, one important consideration is that humans should be positioned at various locations and sizes within the thermal image. Otherwise, the obtained extrinsic parameters will exhibit bias toward specific areas, causing them to deviate from the actual parameters. Furthermore, since the positions of humans in both thermal images and point clouds do not change significantly in consecutive frames, when a thermal image and point cloud are added to the dataset at time t_i, the next three frames will not be considered for inclusion in the dataset.

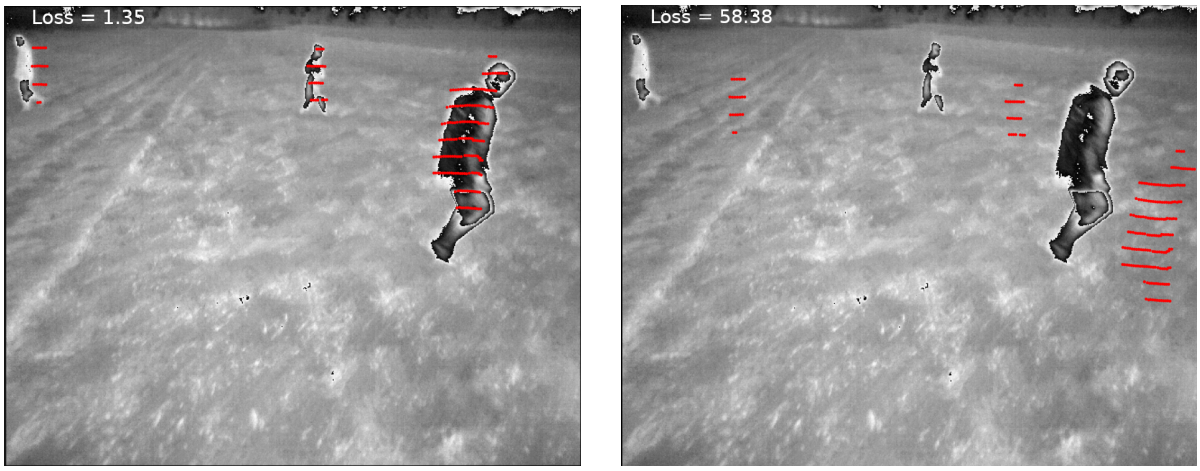
3.4.2 Cost Function

To optimize the extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$ between a thermal camera and a 3D LiDAR sensor, based on human matching in both modalities, a cost function is required to measure the alignment of humans in both modalities for all thermal image and point cloud pairs ($I^h_{t_i}$, $P^h_{t_i}$) in the dataset D, with respect to a set of extrinsic parameters.

When provided with a candidate rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ for image and point cloud pairs ($I^h_{t_i}$, $P^h_{t_i}$), the loss is calculated according to Equation (3.1).

$$Loss(I^h_{t_i}, P^h_{t_i}; \mathbf{R}, \mathbf{t}) = \frac{1}{|P^h_{t_i}|} \sum_{\mathbf{p}^L \in P^h_{t_i}} \psi(\mathbf{K}(\mathbf{R}\mathbf{p}^L + \mathbf{t}); I^h_{t_i}) \tag{3.1}$$

In Equation (3.1), $\mathbf{p}^L$ iterates points in the point cloud $P^h_{t_i}$, $\mathbf{K}$ is the 3×3 intrinsic camera matrix, and $|\cdot|$ denotes the number of points in a point cloud. In this equation, $\mathbf{R}\mathbf{p}^L + \mathbf{t}$ maps the point $\mathbf{p}^L$ from the LiDAR coordinate system to camera coordinate ($\mathbf{p}^C$), and multiplying it by $\mathbf{K}$ maps the point to camera image coordinate ($\mathbf{p}^I$). $\mathbf{p}^I$ is inhomogeneous representation and should be converted to inhomogeneous. $\psi(\mathbf{p}^I; I^h_{t_i})$ is a function that outputs a penalty score



(a) (b)

Figure 3.2: Images (**a**) and (**b**) show the projection of a point cloud onto a thermal image for a sample pair from the FieldSAFE dataset [11] with two different sets of **R** and **t**. Equation (3.1) loss value for the extrinsic parameters used in image (**a**) is 1.35, while the loss value for the extrinsic parameters used in image (**b**) is 58.38.

based on distance of the projected point $\mathbf{p}^I$ from LiDAR coordinate system to image coordinate to the nearest human pixel in $I^h_{t_i}$. The function ψ is defined according to Equation (3.2).

$$\psi(\mathbf{p}^I; I^h_{t_i}) = \begin{cases} \|\mathbf{p}^I - \mathbf{p}_{near}\|_1 & \text{if } \mathbf{p}^C \text{ is in front of the camera image} \\ c_1 \times max(h(I^h_{t_i}), w(I^h_{t_i})) & \text{if } \mathbf{p}^C \text{ is behind the camera image} \end{cases} \tag{3.2}$$

In Equation (3.2), $\|\cdot\|_1$ represents the Manhattan distance, and $h(\cdot)$ and $w(\cdot)$ provide the height and width of $I^h_{t_i}$. Additionally, $\mathbf{p}_{near}$ represents the nearest human pixel in $I^h_{t_i}$ to $\mathbf{p}^I$. ψ is a piecewise function. If a projected point from the LiDAR coordinate system to the camera coordinate system is in front of the camera, the function calculates the distance of the point projection in the thermal image coordinate system to the nearest human pixel. If the projected point from the LiDAR coordinate system to the camera coordinate system is behind the camera, it indicates that the projection is highly invalid. In such cases, we impose a significant penalty by assigning a large value. We determined this penalty to be the maximum value between the image height and width, multiplied by the constant c_1. Selecting a low value, such as one for c_1, means that we do not differentiate enough between a mapping that projects a LiDAR point in front of the camera, outside the image, and not too far from the edges of the image, and a mapping that projects the LiDAR point to the back of the camera. A larger value of c_1, such as five, makes cases like this more distinguishable. In Figure 3.2, the loss for two sets of extrinsic parameters for one pair of thermal images and point clouds from the FieldSAFE dataset [11] is shown. The loss for Figure 3.2a is 1.35, which is much smaller than the 58.38 loss for Figure 3.2b. In the case of Figure 3.2b, greater deviations in the extrinsic parameters from the true values caused LiDAR-projected points to be further from humans in the image, resulting in a larger loss.

The total loss for a candidate **R** and **t** on dataset D is the average of losses on all image and

point cloud pairs in the dataset, as defined in Equation (3.3).

$$Loss(D; \mathbf{R}, \mathbf{t}) = \frac{1}{|D|} \sum_{(\mathbf{p}_I^L, I_{t_i}^h) \in D} Loss(I_{t_i}^h, P_{t_i}^h; \mathbf{R}, \mathbf{t}) \tag{3.3}$$

3.4.3 Optimization Method

In the proposed method, the estimate of the extrinsic parameters, **R** and **t**, that describes the relationship between a thermal camera and a LiDAR sensor, involves the minimization of Equation (3.3). To achieve this, we introduced an optimization approach rooted in evolutionary algorithms for the purpose of parameter calculation between these two sensors. Since errors, such as false positives, false negatives, under-segmentation, and over-segmentation, can occur in the detection and segmentation of humans in both modalities, the proposed algorithm incorporates a mechanism to reduce the effect of outliers. First, we will explain the algorithm that does not consider outlier rejection, Algorithm 1. Afterward, we will provide a comprehensive explanation of the Algorithm 2.

We decided to create the optimization algorithm based on evolutionary algorithms for the following reasons. First, in the case of non-differentiable or noisy objective functions, evolutionary optimization can obtain good solutions. Second, evolutionary optimization is much less likely to be affected by local minima, and it eliminates the need for an initial solution in our calibration method. Third, evolutionary algorithms often exhibit greater robustness in the face of noisy and uncertain observations.

Algorithm 1 presents the proposed algorithm, omitting any outlier rejection. The algorithm creates a population of random individuals and gradually evolves the population in each generation to optimize **R** and **t**. Each individual of the population is an instance of Individual structure. As demonstrated in lines 1–6 Algorithm 1, the Individual structure consists of four fields. The first field, denoted as t, represents the translation vector from a LiDAR sensor to a thermal camera. The second field, labeled as r, corresponds to Rodrigues' rotation vector from the LiDAR to the thermal camera. Instead of directly optimizing the rotation matrix **R** with its 9 elements and managing its orthogonality, we optimize rotation vector **r** with only 3 parameters and subsequently convert it to rotation matrix **R** using OpenCV's Rodrigues function [32]. The third field comprises the resulting loss on the dataset based on the individual's r and t, which is calculated according to Equation (3.3). The fourth field for an individual represents the probability of selection for crossover and mutation, a concept we will elaborate on further.

This algorithm operates on a dataset denoted as D, which has been generated in accordance with Section 3.4.1. It takes parameters like N_{pop}, signifying the number of individuals in the population, and $interval_{rot}$ and $interval_{tran}$, representing the rotation and translation intervals used for generating random initial individuals in the population. Furthermore, we have pct_{elite}, a parameter that determines the percentage of the best-performing individuals with the lowest loss to be retained in the next generation. Additionally, $pct_{crossover}$ is another parameter that specifies the percentage of the population selected for crossover.

In line 7, the initial population is generated using the 'initialPopulation' function. To enhance diversity, the size of the population that it generates is set to be c_2 times larger than N_{pop}. However, after the first iteration, the population size is reduced to N_{pop}, as shown in lines 10–12. If the number of individuals in the population is low, setting c_2 to a value like five

Algorithm 1 Proposed algorithm without outlier handling

Require: D, N_{pop}, $iter_{max}$, $interval_{rot}$, $interval_{tran}$, pct_{elite}, $pct_{crossover}$

Struct Individual {
    vector3D t;
    vector3D r;
    float $loss$;
    float $prob$;
}
$population$ = initialPopulation(size=$c_2 \times N_{pop}$, $interval_{rot}$, $interval_{tran}$)
for $iter_i$ = 1 to $iter_{max}$ **do**
    $nextPopulation$ = {}
    if $iter_i > 1$ **then**
        $population$ = top N_{pop} lowest loss individuals in $population$
    end if
    for *individual* in $population$ **do**
        $individual.loss$ = Loss(D; $Rodrigues(individual.r)$, $individual.t$) (Eq. 3.3)
    end for
    for *individual* in $population$ **do**
        $individual.prob$ = selectionProbability($population$, *individual*)
    end for
    Add the top ($pct_{elite} \times |population|$) lowest loss individuals to $nextPopulation$
    Randomly select ($pct_{crossover} \times |population|$) pairs with replacement from $population$ based on the probability of each individual.
    Apply the 'crossOver()' operation to each selected pair and add the resulting new individuals to the $nextPopulation$.
    Randomly select ($|population| - |nextPopulation|$) individuals with replacement from the $population$ based on the probability of each individual.
    Apply the 'mutation()' operation to each selected individual and add the resulting new individuals to the $nextPopulation$.
    $population \leftarrow nextPopulation$
end for
return **R** and **t** of *individual* in $population$ with smallest $individual.loss$

can increase diversity. However, when the population is large, it can be set to one to prevent unnecessary computation. To create a random individual within the population, 'initialPopulation' initializes an instance of the Individual structure. The function randomly samples all three elements of vectors t and r from the intervals $interval_{tran}$ and $interval_r$, respectively. In all our experiments, we assumed no prior information about the LiDAR and thermal camera position and orientation relative to each other. We selected a wide interval of $[-3.5, 3.5]$ radians and $[-1, 1]$ meters; however, a user can choose smaller intervals if they wish to incorporate prior knowledge about the positions and orientations of sensors. Next, the produced individual becomes part of the population under the condition that, for a pair of $I^h_{t_i}$ and $P^h_{t_i}$ in dataset D, a minimum of 50% of the points in $P^h_{t_i}$ project within the thermal image. This projection is achieved through the utilization of a randomly generated rotation vector r and translation vector t associated with the individual. In case this criterion is not met, the individual is discarded, and a new one is generated in its place.

Between lines 8 and 25, the next generation is formed through a process that combines elitism, crossover, and mutation techniques. In lines 13–15, the loss on dataset D for each individual is computed as per Equation (3.3). In lines 16–18, *individual.prob* is calculated for each *individual* in the population using the 'selectionProbability' function as defined in Equations (3.4) and (3.5). The first one computes a fitness score based on individual loss relative to the population, and the second one calculates the selection probability for an individual, taking their fitness score and the sum of fitness scores for the entire population into account.

$$individual.score = 1 - \frac{individual.loss}{\sum_{ind \in population} ind.loss} \tag{3.4}$$

$$individual.prob = \frac{individual.score}{\sum_{ind \in population} ind.score} \tag{3.5}$$

In line 19, the top pct_{elite} percent of individuals with the lowest loss in the population are directly copied to the next generation. This elitism ensures that the best solutions found so far are not lost and continue to contribute to the population's overall quality over the next generations.

Between lines 20–23, individuals for crossover and mutation are selected, and the functions 'crossOver' and 'mutation' are applied. 'crossOver' creates a new individual from a pair of individuals according to Equations (3.6) and (3.7). In these two equations, *individualOne* and *individualTwo* are two members of the population, and *individualOne* has a lower loss than the other one. Also, α is a random number between 0.5 and 1. The function 'mutation' affects an individual by applying noise to its rotation and translation vectors, creating a new individual. The 'mutation' operation adds random uniform noise within the range of $[-\sigma_{rot}, \sigma_{rot}]$ to each element of the rotation vector and independently adds noise within the range of $[-\sigma_{trans}, \sigma_{trans}]$ to each element of the translation vector.

$$newIndividual.r = \alpha \cdot individualOne.r + (1 - \alpha) \cdot individualTwo.r \tag{3.6}$$

$$newIndividual.t = \alpha \cdot individualOne.t + (1 - \alpha) \cdot individualTwo.t \tag{3.7}$$

Algorithm 2 contains the complete proposed algorithm, which attempts to mitigate the effects of outlier data pairs. The general idea of this algorithm is to handle outliers in a dataset

(D) by iteratively fitting a model to a small subset of the data, identifying and removing outliers based on a loss threshold, and then re-fitting the model to the inliers. The algorithm is designed to robustly estimate rotation ($\mathbf{R}$) and translation ($\mathbf{t}$) parameters for a given dataset.

Algorithm 2 Proposed algorithm with outlier handling

Require: D, N_{pop}, $iter_{max}$, $interval_{rot}$, $interval_{tran}$, pct_{elite}, $pct_{crossover}$, min_{sample}, $iter_{outlier}$, $ratio_{solution}$, $threshold_{sample}$

Create an array, $isInlier$, with a size of $|D|$ and initialize each element with $True$
for $iter_i = 1$ to $iter_{outlier}$ **do**
  Create D_{train} by randomly sampling min_{sample} data pairs from D
  Create D_{val} using the remaining data pairs from D
  Obtain $\mathbf{R}$ and $\mathbf{t}$ by using Algorithm 1
  $listLosses$ = loss of $\mathbf{R}$ and $\mathbf{t}$ for each data pairs in D_{val} using Eq. 3.1
  $ratio_{inliers} = \frac{\sum_{a \in listLosses} I(a <= threshold_{sample})}{|ratio_{inliers}|}$
  if $ratio_{inliers} >= ratio_{solution}$ **then**
    for $pair_i$ in D_{val} **do**
      if $listLosses[pair_i] > threshold_{sample}$ **then**
        $isInlier[pair_i] \leftarrow$ False
      end if
    end for
  end if
end for
Create D_{inlier} by selecting elements in D where the corresponding element in $isInlier[pair_i]$ is True
Obtain $\mathbf{R}$ and $\mathbf{t}$ by applying Algorithm 1 to D_{inlier}
return $\mathbf{R}$ and $\mathbf{t}$

Algorithm 2 requires all the inputs of Algorithm 1, with the addition of some extra inputs. min_{sample} represents the size of a random subset of D that is chosen to find extrinsic parameters. $iter_{outlier}$ denotes the number of fitting attempts to detect outliers. $threshold_{sample}$ determines whether a sample should be considered an outlier or not. If the calculated loss for a sample pair, as per Equation (3.1), is greater than $threshold_{sample}$, it is considered an outlier. A solution of a fitting attempt on the selected subset of D is deemed correct if the ratio of samples with a loss smaller than or equal to the value of $threshold_{sample}$ is greater than or equal to $ratio_{solution}$. Furthermore, $I(\cdot)$ represents the indicator function. It outputs the value of one when the condition is met and zero otherwise.

The proposed algorithms aim to determine a rigid body transform between the coordinate systems of a thermal camera and a LiDAR sensor by estimating the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$. It is essential for both sensors to operate with the same scale for accurate results. If the two sensors are not on the same scale, and assuming the factory configurations of sensors are available (which is almost always the case for these two types of sensors), this information can be used to preprocess the data and convert them to the same scale. In Equation (3.1), $\mathbf{K}(\mathbf{R}\mathbf{p}^L + \mathbf{t})$ is utilized to map a LiDAR point in the image coordinate system in a homogeneous format. Subsequently, the homogeneous point is converted to an inhomogeneous coordinate in the thermal image. When using data with different scales, as the cost function minimizes

the distance in the thermal image, it can yield a solution that effectively maps LiDAR points to their corresponding thermal image pixels, even when dealing with data of varying scales. However, the obtained translation vector may not accurately represent the real distance between the sensors, as it will be scaled by the difference in scale between the two sensors.

To efficiently calculate the function in Equation (3.1), for each I^h_{ti} in a collected dataset, an array with a height of h and a width of w can be created, where each element represents the distance from that pixel to the nearest pixel belonging to a human. For a dataset of size $|D|$, the computational complexity of this operation is $O(|D|.w.h)$. In Equation (3.2), for a given $I^h_{t_i}$ and $P^h_{t_i}$ pair, for the number of points in the point cloud ($|P^h_{t_i}|$), several fixed matrix multiplications and summations take place. Therefore, for one pair, the computational complexity will be $O(|P^h_{t_i}|)$. According to Equation (3.3), its computation complexity is $O(|D|.|P^h_{t_i}|)$. Therefore, since Algorithm 1 performs $iter_{max}$ iterations, and each iteration calculates the loss of individuals on a scale of N_{pop}, the total computational complexity will be $O(|D| \cdot w \cdot h) + O(|D| \cdot |P^h_{max}| \cdot N_{pop} \cdot iter_{max})$, where $|P^h_{max}|$ is the number of points in the point cloud with the most points. The computational complexity of Algorithm 2 remains the same, with the additional step of calculating extrinsic parameters using a subsampled dataset of size min_{sample} for $iter_{outlier}$ times.

3.5 Experiments

To evaluate our method, we used the FieldSAFE dataset [11] and the first sequence of the MS^2 dataset [13]. The selection of this sequence was random, as it is assumed to be representative of the dataset, given that the sensor setup is identical across all sequences. The FieldSAFE dataset [11] contains data from a tractor equipped with various sensors, including a thermal camera and a LiDAR sensor, captured during a grass-mowing scenario in Denmark. The MS^2 dataset comprises data collected by an instrumented car with different sensor types, such as a thermal camera and LiDAR sensor, in various environments, including city, residential, road, campus, and suburban areas. The thermal camera in the FieldSAFE dataset is a FLIR A65 with a maximum frame rate of 30 frames per second (FPS) and a resolution of 640 $\times$ 512 pixels. It obtained LiDAR data from the Velodyne HDL-32E, which is a 32-beam LiDAR sensor with a 10 FPS data rate and 2 cm accuracy. The thermal camera in the MS^2 dataset is the same as in FieldSAFE, and the LiDAR is a Velodyne VLP-16, which has sixteen LiDAR beams, a maximum frame rate of 20 FPS, and 3 cm accuracy. In the MS^2 dataset, the provided thermal images have a resolution of 640 by 256 pixels. Moreover, in both datasets, the positions and orientations of the sensors with respect to each other are highly different. Our proposed algorithm produces accurate results on both setups, including sparse 16-beam and dense 32-beam LiDARs, demonstrating its effectiveness. Also, in both datasets, the intrinsic camera matrices ($\mathbf{K}$) of thermal cameras are available.

We created two datasets from FieldSAFE and MS^2 following the guidelines in Section 3.4.1. Additionally, we generated two other datasets for evaluation purposes by manually selecting and annotating the data. For human segmentation in thermal camera images, we utilized Faster R-CNN [33] trained on a FLIR thermal dataset [34] and subsequently fed the bounding boxes into the Segment Anything Model (SAM) [35]. To extract humans from the LiDAR point cloud, we employed MMDetection3D [36]. The dataset created from FieldSAFE consists of

Table 3.1: Hyper-parameters for Algorithms 1 and 2.

Hyper-parameter	Value
N_{pop}	500
$iter_{max}$	400
$interval_{rot}$	[-3.5, 3.5] rad
$interval_{trans}$	[-1000, 1000] mm
pct_{elite}	15%
$pct_{crossover}$	40%
c_1	5
c_2	5
min_{sample}	20 if number of train sample $\geq$ 40 else 15
$iter_{outlier}$	2
$ratio_{solution}$	0.7
$threshold_{sample}$	FieldSAFE: 2.0, MS^2: 1.5
σ_{rot}	0.02 rad
σ_{trans}	20 mm

63 training examples and 20 test samples, while the dataset extracted from MS^2 comprises 55 training examples and 19 test samples. For simplicity, we denote them as D_{FS}^{train}, D_{FS}^{test}, D_{MS}^{train}, and D_{MS}^{test}. Since there are often only one to three persons in the sequences used from both the FieldSAFE and MS^2 datasets, we selected $H_{\max}$ to be equal to three. In D_{FS}^{train}, the mean spatial location of all humans in thermal images is (305.82, 103.49), with standard deviations of 155.9 and 43.3 along the x and y axes, respectively. Additionally, the average number of persons per image is 1.16. For D_{MS}^{train}, the corresponding values are (330.2, 140.2) for the mean spatial location, with standard deviations of 166.3 and 11.95 along the x and y axes, respectively. The average number of persons per image is 1.03. In the following, we compare the loss values obtained via Equation (3.3) on both the training and test datasets for our proposed methods in Algorithms 1 and 2 across different settings. Since the used data were collected in the past, we compare the proposed method with the extrinsic parameters provided by FieldSAFE and MS^2 using target-based calibration methods. For simplicity, we refer to them as $FS_{[R,t]}$ and $MS_{[R,t]}$.

In all our experiments, we used the hyper-parameters in Table 3.1 by default, unless another configuration was specified. We determined the hyper-parameters for the proposed algorithms through a process of testing various candidates and relying on intuition.

To compare Algorithms 1 and 2 with each other as well as with $FS_{[R,t]}$ and $MS_{[R,t]}$, in Table 3.2, we reported the Equation (3.3) loss values obtained by their corresponding $\mathbf{R}$ and $\mathbf{t}$ on the test datasets D_{FS}^{test} and D_{MS}^{test}. As can be seen in the table, Algorithm 2, which uses outlier handling, obtains better results than Algorithm 1. Additionally, Algorithm 2 outperforms $FS_{[R,t]}$ and $MS_{[R,t]}$, which are obtained using calibration methods based on the target.

The values reported in Table 3.2 and subsequent experiments are derived by utilizing Equation (3.3) loss function on one of the two test datasets. Following the estimation of extrinsic parameters using either Algorithm 1 or 2, we can reasonably expect that all LiDAR points corresponding to individuals, once transformed with the resulting rotation matrix and translation vector, will be situated in front of the camera within the camera coordinate system. Conse-

Table 3.2: Comparison of Equation (3.3) loss for different methods on D_{FS}^{test} and D_{MS}^{test} datasets.

		Dataset	
		D_{FS}^{test}	D_{MS}^{test}
Method	Algorithm 1	0.953	0.352
	Algorithm 2	**0.798**	**0.34**
	$FS_{[R,t]}$	0.835	-
	$MS_{[R,t]}$	-	2.731

quently, within the piecewise function described by Equation (3.2), only the first segment will be applicable. Therefore, the loss value obtained from Equation (3.2) for a pair in a test dataset demonstrates the average Manhattan distance of each human LiDAR point projected onto the thermal image, measured to the nearest human pixel. For example, this value for the thermal camera-point cloud pair and assumed extrinsic parameters in Figure 3.2a is 1.58. Consequently, the overall loss value obtained from Equation (3.3) for all pairs in the test dataset represents the average of all values obtained by Equation (3.2) for each pair in the dataset. In other words, the loss value calculated by Equation (3.3) provides a summary measure that represents the central tendency of the average Manhattan distance of each human LiDAR point projected onto the thermal image, measured to the nearest human pixel across pairs in a test dataset.

Figure 3.3 presents some performance metrics for Algorithm 2 optimized on D_{FS}^{train}. Figure 3.3 includes four plots, each displaying different aspects of the optimization process in each generation. All the loss values for the figure are computed using Equation (3.3). We just reported the plots for D_{FS}^{train} as the representative of both the D_{FS}^{train} and D_{MS}^{train} datasets. Figure 3.3a shows the training loss value of the individual with the lowest training loss. Because of elitism, mutation, and crossover, the training loss value for the individual with the lowest training loss always remains non-increasing across generations. Figure 3.3b,c illustrate the log-average training loss of all individuals and the standard deviation of the loss among all individuals in the population. As individuals with lower training loss have a higher probability of being selected for crossover and mutation, increasing the number of generations results in a decrease in the log-average and standard deviation of train loss. However, due to randomness in mutation and crossover, these values eventually converge to a certain point and fluctuate around it. Finally, Figure 3.3d demonstrates the test loss of the individual with the lowest training loss. As depicted in the figure, both the training and test losses exhibit an initial exponential decrease, followed by a gradual convergence to a small value.

To assess the influence of the training dataset size on Algorithms 1 and 2, we performed the sub-sampling of D_{FS}^{train} and D_{MS}^{train}, resulting in new training datasets ranging in size from 5 to the full dataset size, with a step size of 5. Since Algorithm 2 requires a minimum of 15 samples to determine a set of extrinsic parameters and subsequently test other samples for inlier status, we opted not to execute Algorithm 2 for configurations with 15 samples or fewer. As shown in Table 3.3 and its equivalent bar charts in Figure 3.4, increasing the number of data pairs for the training set from a small number decreases the test loss values significantly. Also, Algorithm 1 exhibits fluctuation in test loss values as the number of thermal images and point cloud pairs in the training set increases. In contrast, Algorithm 2 experiences fewer fluctuations. Additionally, in almost all cases, Algorithm 2 demonstrates superior performance

Table 3.3: The effect of varying the size of the training dataset on the test loss values of Algorithms 1 and 2. The reported loss values calculated by Equation (3.3) on D_{FS}^{test} and D_{MS}^{test}.

	No. of used pairs	5	10	15	20	25	30	35	40	45	50	55	60	63
D_{FS}^{test}	Algorithm 1	2.474	1.858	1.179	1.236	0.905	0.9	0.884	0.878	0.959	0.999	1.047	0.959	0.953
	Algorithm 2	-	-	-	0.957	0.902	0.919	0.804	0.791	0.785	0.808	0.869	0.799	0.798
D_{MS}^{test}	Algorithm 1	0.709	0.47	0.412	0.408	0.425	0.424	0.345	0.414	0.408	0.363	0.352	-	-
	Algorithm 2	-	-	-	0.414	0.405	0.383	0.343	0.405	0.325	0.357	0.34	-	-

compared with Algorithm 1 with the same training dataset size. In the case of 30 pairs in the dataset D_{FS}^{train} and 20 pairs in the dataset D_{MS}^{train}, Algorithm 2 obtained a slightly worse result, which could be attributed to randomness, especially in the selection of a subsampled set from the dataset to assess the inlier or outlier status of non-subsampled data pairs. As the results in Table 3.3 suggest, not having a sufficient number of samples prevents us from executing the algorithms or obtaining good results.

To assess the robustness of Algorithms 1 and 2 under more extreme conditions, we generated D_{FS-SW4}^{train} by swapping the thermal mask ($I_{l_i}^h$) for two random samples with another two random samples in D_{FS}^{train}. It caused four pairs of thermal images and LiDAR point clouds to lack matching masks in both modalities. Similarly, we created D_{MS-SW4}^{train} using the same method. Furthermore, to investigate under different levels of mismatch, we generated comparable datasets by interchanging 4, 6, and 8 pairs, resulting in 8, 12, and 16 mismatched samples, respectively. As shown in Table 3.4 and its equivalent bar charts in Figure 3.5, Algorithm 2 achieved significantly better test loss and demonstrated greater robustness. In this experiment, $threshold_{sample}$ and $iter_{outlier}$ were set to three and five for new datasets derived from D_{FS}^{train}, and the variable $ratio_{solution}$ was set to 0.3 for $D_{MS-SW12}^{train}$ and $D_{MS-SW16}^{train}$. By increasing the number of mismatched pairs, the performance of both algorithms dropped; however, this effect was more significant for Algorithm 1. As the results suggest, it is critical to have good object detection in both modalities; otherwise, large amounts of false positives and false negatives from object detectors can degrade the quality of extrinsic parameters. Another interpretation could be that the presence of many people in a thermal image-point cloud pair may result in more mistakes in segmenting humans in both modalities due to a higher chance of overlapping. Therefore, selecting a large value for $H_{\max}$ may consequently lead to poorer results.

As mentioned earlier, it is important to collect a dataset with thermal images depicting humans in different locations and sizes. In order to assess the robustness of Algorithms 1 and 2 when dealing with highly unbalanced human locations in a collected dataset, we generated D_{FS-NL}^{train} from D_{FS}^{train} by removing samples where the human masks are located in the left one-third section of the image. D_{FS-NL}^{train} comprises 27 samples. Similarly, we created D_{MS-NR}^{train} by removing samples where the human masks are located in the right one-third of the image. D_{MS-NR}^{train} consists of 36 samples. We generated these two imbalanced pose datasets from various imbalanced datasets that can be created to serve as a representative sample of this issue. As Table 3.5 shows, the mentioned unbalanced condition decreases performance when compared with the performance on a balanced dataset of a similar size in Table 3.3. However, Algorithm 2 is less affected by this in comparison with Algorithm 1. Therefore, it is important to have humans in diverse locations in the thermal camera's field of view; otherwise, the pose imbalance can negatively affect the extrinsic calibration.

Table 3.4: Comparing Algorithms 1 and 2's test loss under harsher conditions by introducing artificial mismatches between masks in both modalities. The provided values correspond to the loss values computed using Equation (3.3) on D_{FS}^{test} and D_{MS}^{test}.

	Algorithm 1	Algorithm 2
D_{FS}^{train}	0.953	0.798
D_{FS-SW4}^{train}	1.001	0.826
D_{FS-SW8}^{train}	1.015	0.868
$D_{FS-SW12}^{train}$	1.159	1.100
$D_{FS-SW16}^{train}$	1.558	1.356
D_{MS}^{train}	0.352	0.340
D_{MS-SW4}^{train}	0.415	0.342
D_{MS-SW8}^{train}	0.480	0.343
$D_{MS-SW12}^{train}$	1.305	0.500
$D_{MS-SW16}^{train}$	1.577	0.832

Table 3.5: Comparing Algorithms 1 and 2's test loss values calculated using Equation (3.3) on D_{FS}^{test} and D_{MS}^{test} under unbalanced human locations in a collected dataset.

	Algorithm 1	Algorithm 2
D_{FS-NL}^{train}	1.482	1.415
D_{MS-NR}^{train}	0.463	0.356

Table 3.6: The effect of removing different components from Algorithms 1 and 2 on the loss of Equation (3.3) on the dataset D_{FS}^{test}.

Removed part	None	Elitism	Mutation	Crossover	No init. const.
Algorithm 1	0.953	3756.001	1.596	8.082	0.972
Algorithm 2	0.798	3391.615	1.595	23.626	0.928

To assess the importance of each component in Algorithms 1 and 2, we systematically removed one component at a time and reported the results by calculating the Equation (3.3) loss using the test dataset D_{FS}^{test}, as shown in Table 3.6. The table reveals that removing elitism results in divergence and has the most significant impact. Subsequently, both the crossover and mutation exhibit notable importance, albeit to varying degrees. Removing the condition that projects 50% of the point cloud into the thermal image during the creation of the initial population has the least impact on test loss.

To observe the impact of the changes in certain hyper-parameters of Algorithms 1 and 2 and explain our intuition for selecting default values of hyper-parameters, we modified one parameter at a time while keeping all other parameters constant, as specified in Table 3.1. The corresponding results are presented in Table 3.7. In most cases, selecting values near the default showed no significant degradation in the performance of both algorithms. To demonstrate a more pronounced effect, we opted for more extreme values in comparison with the defaults. However, even in this scenario, in many cases, the results were not substantially different from the results of the default hyper-parameters.

As depicted in Table 3.7, a small population size (N_{pop}) results in poorer outcomes than the default value due to insufficient diversity. Conversely, a large population size slows down convergence and adds unnecessary computational overhead, approaching results similar to the default value. A low value of pct_{elite} implies that many of the found good solutions do not directly transition to the next generation, diminishing their contribution to the overall population quality. Conversely, a large value of pct_{elite} restricts the introduction of new individuals. In both cases, the results are inferior compared with the default value. A smaller value of $pct_{crossover}$ implies that fewer individuals in the next generation are produced by crossover, and more individuals are created by mutation. In the proposed algorithms, crossover covers a large area in the optimization space, and, as shown, a small value of $pct_{crossover}$ resulted in significantly poorer performance compared with the default value. In these algorithms, the mutation operation allows for the discovery of better solutions in the proximity of an existing solution. On the contrary, a large value of $pct_{crossover}$ means less mutation, leading to lower performance compared with the default values. Finding a balance between the crossover and mutation is crucial for achieving good results. σ_{rot} and σ_{trans} represent the noise levels for the mutation operator, determining how much change in a found solution is applied to generate a new individual. A very small amount does not alter parameters in the optimization space enough to produce a meaningful change in the outcome, while a large amount results in an individual that is very different from the original solution and does not retain its attributes. As shown, in both cases, the results are worse than the default values.

A low value of $threshold_{sample}$ imposes a stringent criterion for considering a sample in the dataset as an inlier, potentially causing issues by incorrectly classifying many good pairs in the

Table 3.7: The effect of changing some of the default hyper-parameters on Algorithms 1 and 2 on the loss of Equation (3.3) on the dataset D_{FS}^{test}.

Hyper-parameter	All	N_{pop}		pct_{elite}		$pct_{crossover}$		σ_{rot}		σ_{trans}	
Value	Default	100	800	2%	60%	10%	90%	0.005	0.2	5	200
Algorithm 1	0.953	1.588	0.962	2.209	0.97	1.011	1.043	1.021	1.023	1.509	1.049
Hyper-parameter	All	$threshold_{sample}$		min_{sample}		$iter_{outlier}$		$ratio_{solution}$		-	
Value	Default	0.5	6.0	10	30	1	5	0.1	0.9	-	
Algorithm 2	0.798	0.952	0.845	0.957	0.84	0.95	0.8	0.798	0.95	-	

data as outliers and rejecting them from the calculation of extrinsic parameters. Conversely, a high-value results in the ineffective detection of outlier samples in data. In both cases, the performance is weaker compared with the default value. As depicted in Table 3.3, augmenting the pairs for optimizing extrinsic parameters generally leads to improved performance. A small value of $threshold_{sample}$ results in the identification of suboptimal extrinsic parameters, leading to poor outlier detection performance. Conversely, when the value of $threshold_{sample}$ is large, there is a higher likelihood of including a significant amount of outliers. The algorithm may face challenges in identifying a robust model amidst the abundance of irrelevant data. As indicated in Table 3.7, in both scenarios, the performance is diminished compared with the default value. We selected the default value for min_{sample}, as represented in Table 3.1, based on the performance of Algorithm 1 in Table 3.3. As shown in Table 3.7, a low value for min_{sample} can result in obtaining a poor initial estimate for extrinsic calibration parameters, thereby impacting the performance of determining inliers. Additionally, a large value can lead to the exclusion of a significant number of samples from the determination of whether they are outliers or not, resulting in poorer results. As can be interpreted from Table 3.7, a small value for $iter_{outlier}$ can cause many samples not to be examined for being outliers, resulting in a decrease in performance. On the other hand, a large value does not contribute to finding more outliers, and the performance remains similar to a balanced $iter_{outlier}$ while only increasing computation. As indicated by the values in Table 3.7, a low value of $ratio_{solution}$ does not alter the performance in the specific experiment of D_{FS}^{test}. However, a high value of $ratio_{solution}$ led to poor performance, as the proportion of inliers in each iteration of Algorithm 2 was smaller than the $ratio_{solution}$, and, consequently, the detected outliers were rejected.

In Figure 3.6, the dots represent projected points in the LiDAR point cloud onto a thermal image using a set of $\mathbf{R}$ and $\mathbf{t}$. This figure presents a qualitative comparison of Algorithm 2 (blue dots) with $FS_{[R,t]}$ and $MS_{[R,t]}$ (red dots) on two frames from D_{FS}^{test} and D_{MS}^{test}. As can be observed, both the red and blue dots are closely aligned, demonstrating that our proposed algorithm and $FS_{[R,t]}$ and $MS_{[R,t]}$ are in close agreement. However, as depicted in the zoomed-in patches in Figure 3.6b,d, the blue projected points that correspond to humans in the point cloud are more closely aligned with the humans in the thermal images. Additionally, in Figure 3.6d, the blue points are more centered on the streetlight.

3.6 Conclusion and Future Work

In this paper, we have highlighted the advantages of combining data from thermal cameras and LiDAR sensors and emphasized the importance of accurately determining the rotation matrix $\mathbf{R}$

and the translation vector **t** to effectively utilize data from both the thermal camera and LiDAR. Also, we mentioned certain challenges associated with using specific targets visible in thermal cameras, especially when dealing with regular sensor drift or changing settings. To address these challenges, we have introduced an extrinsic calibration algorithm. This algorithm aligns a thermal camera and a LiDAR without the need for a dedicated target. This calibration is achieved by matching segmented human subjects in both modalities using pairs of thermal images and LiDAR point clouds that were collected during the sensor setup's movement. Firstly, we introduced the procedure for constructing a dataset comprising pairs, where each pair consists of thermal camera data and its corresponding point cloud. Secondly, we presented a novel loss function that quantifies the alignment between the LiDAR and thermal camera coordinate systems given the rotation matrix **R** and translation vector **t**. Thirdly, we introduced two evolutionary algorithms, one of which does not explicitly address the issue of outliers, while the other mitigates the impact of outliers. Also, our proposed algorithm obviates the need for an initial estimate of **R** and **t**. Finally, we conducted a series of comprehensive experiments to assess the efficiency of the proposed algorithms under various settings and to compare the performance of them with the provided extrinsic parameters in the FieldSAFE dataset [11] and the MS^2 dataset [13]. This comparison offers a quantitative and qualitative assessment of our method's performance, providing valuable insights into its effectiveness and robustness. In one instance, our method exhibits a noteworthy 4.43% improvement in the designed loss compared with extrinsic parameters derived from target-based calibration in the FieldSAFE dataset. In another instance, distorting a dataset by randomly swapping thermal cameras of four pairs in the data with another four pairs to create a new dataset with eight mismatches between thermal images and point clouds only resulted in an 8.7% increase in the loss, showcasing its robustness.

For future work, we plan to explore several directions based on the different experiments presented. Firstly, we aim to achieve better results with fewer pairs in the dataset. Secondly, as demonstrated, the dataset collected from thermal cameras indicates that humans are often not in varying positions, and distances from the camera can negatively impact the quality of the extrinsic calibration. We will investigate methods, such as weighting different pairs, to address this issue. Thirdly, we will explore multi-objective optimization to incorporate more complex information about masked humans in both modalities in order to obtain better results.

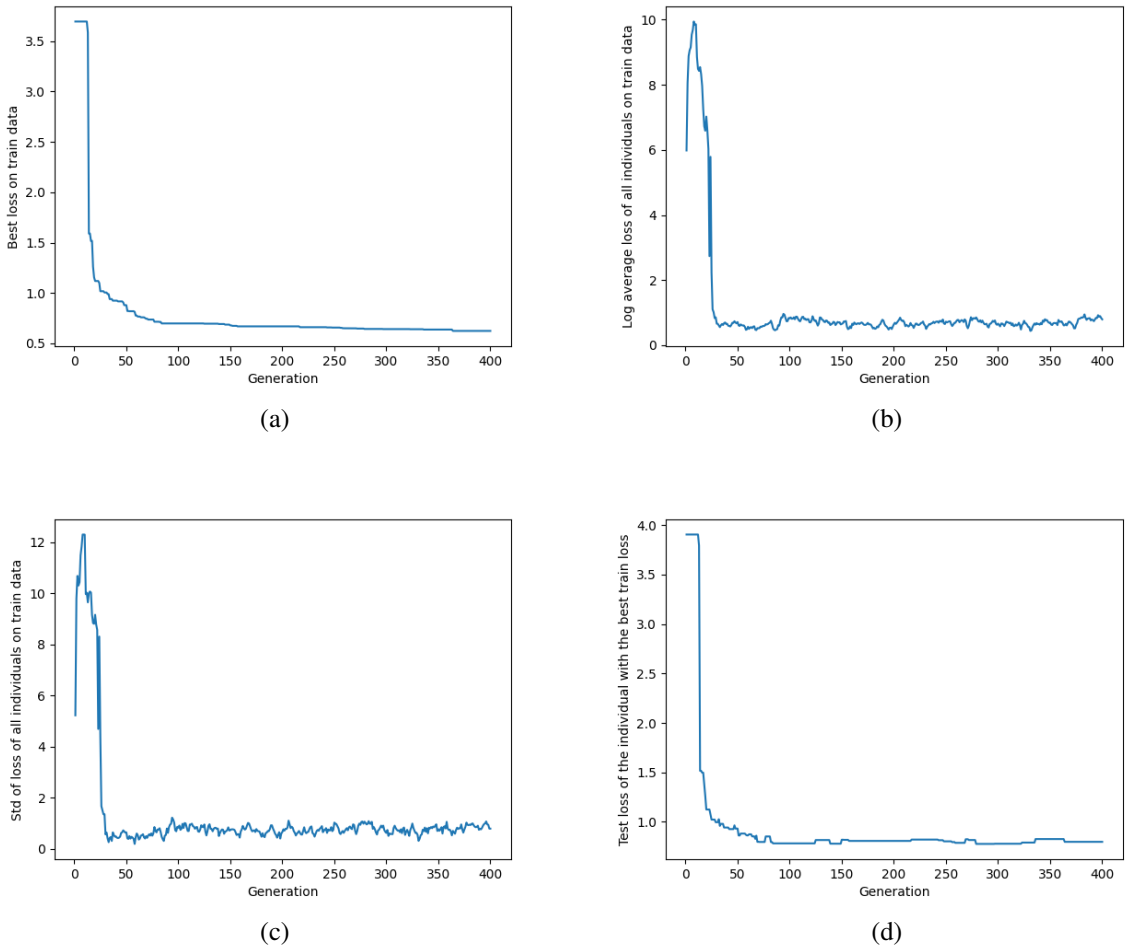


Figure 3.3: Plots for Algorithm 2 optimized on D_{FS}^{train} depicting (**a**) the train loss of the individual with the lowest train loss in each generation, (**b**) the log-average train loss of all individuals in the population in each generation, (**c**) the standard deviation of the loss among all individuals in the population for each generation, and (**d**) the test loss of the individual with the lowest train loss in each generation.

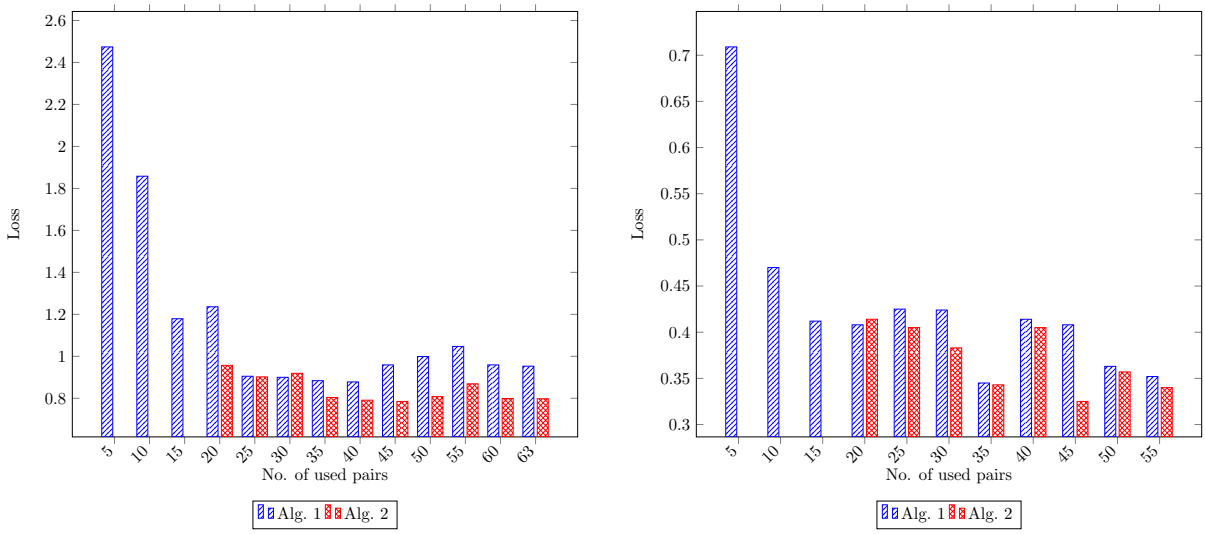


Figure 3.4: (**a**,**b**) are bar charts for datasets derived by subsampling from D_{FS}^{train} and D_{MS}^{train}, respectively, as created from Table 3.3. They display the test loss values of Algorithms 1 and 2 calculated by Equation (3.3) on D_{FS}^{test} and D_{MS}^{test}.

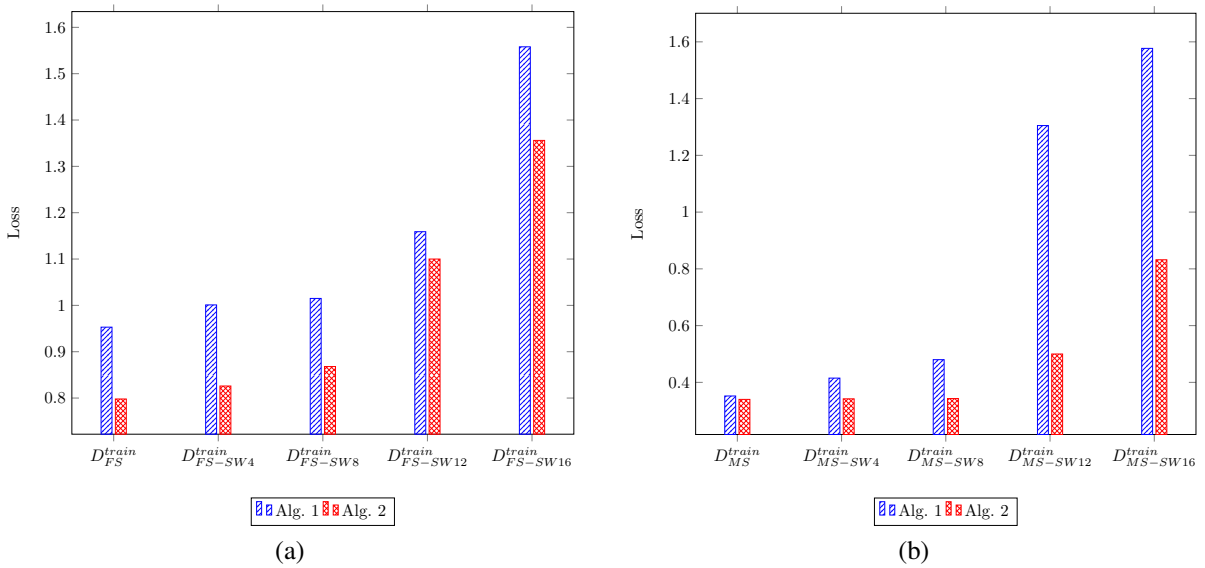


Figure 3.5: (**a**,**b**) are bar charts, respectively, for datasets derived from D_{FS}^{train} and D_{MS}^{train} by swapping thermal masks. Bar charts (**a**,**b**) are created from Table 3.4. The provided values correspond to the losses computed using Equation (3.3) on D_{FS}^{test} and D_{MS}^{test}.

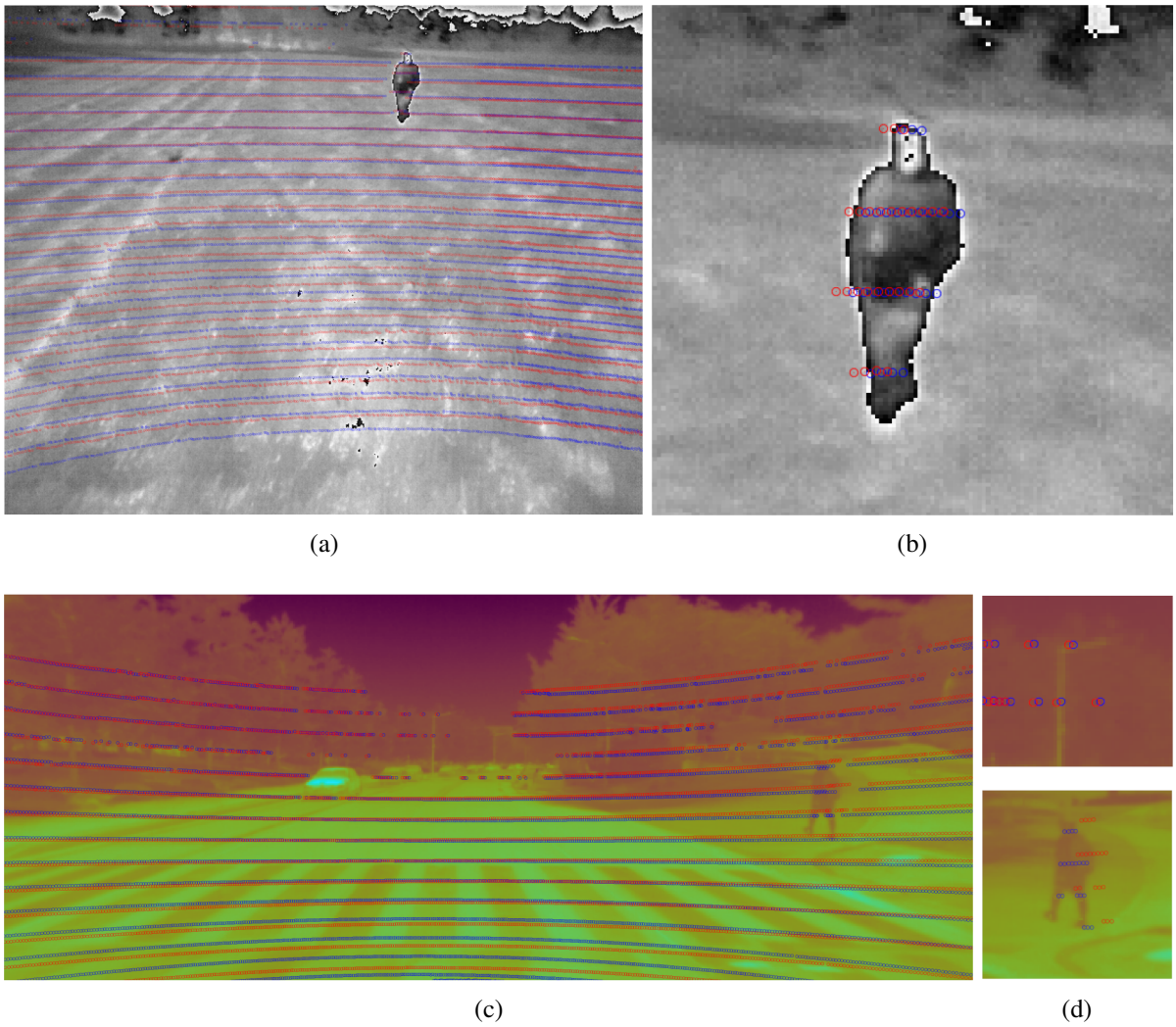


Figure 3.6: Images (**a**,**c**) respectively show a comparison of Algorithm 2 (blue dots) with $FS_{[R,t]}$ and $MS_{[R,t]}$ (red dots) on two samples from FieldSAFE [11] and MS^2 [13] datasets. The dots represent projected points from the LiDAR point cloud onto the thermal image. Additionally, the images (**b**,**d**) are zoomed-in patches taken from the frames on (**a**) and (**c**), respectively. To enhance visual interpretation, the image in (**c**) and its zoomed-in patches in (**d**) were pseudo-colored from the original grayscale image.

References

[1] Kocić, J.; Jovičić, N.; Drndarević, V. Sensors and sensor fusion in autonomous vehicles. In Proceedings of the 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 20–21 November 2018; pp. 420–425.

[2] Vizzo, I.; Guadagnino, T.; Mersch, B.; Wiesmann, L.; Behley, J.; Stachniss, C. Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1029–1036.

[3] Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9297–9307.

[4] Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 12697–12705.

[5] Guo, Q.; Su, Y.; Hu, T.; Guan, H.; Jin, S.; Zhang, J.; Zhao, X.; Xu, K.; Wei, D.; Kelly, M.; et al. Lidar boosts 3D ecological observations and modelings: A review and perspective. *IEEE Geosci. Remote Sens. Mag.* **2020**, *9*, 232–257.

[6] Debnath, S.; Paul, M.; Debnath, T. Applications of LiDAR in Agriculture and Future Research Directions. *J. Imaging* **2023**, *9*, 57.

[7] Xu, X.; Dong, S.; Xu, T.; Ding, L.; Wang, J.; Jiang, P.; Song, L.; Li, J. FusionR-CNN: LiDAR-Camera Fusion for Two-Stage 3D Object Detection. *Remote Sens.* **2023**, *15*, 1839.

[8] Miethig, B.; Liu, A.; Habibi, S.; Mohrenschildt, M.V. Leveraging thermal imaging for autonomous driving. In Proceedings of the 2019 IEEE Transportation Electrification Conference and Expo (ITEC), Detroit, MI, USA, 19–21 June 2019; pp. 1–5.

[9] Vadivambal, R.; Jayas, D.S. Applications of thermal imaging in agriculture and food industryA review. *Food Bioprocess Technol.* **2011**, *4*, 186–199.

[10] Gade, R.; Moeslund, T.B. Thermal cameras and applications: A survey. *Mach. Vis. Appl.* **2014**, *25*, 245–262.

[11] Kragh, M.F.; Christiansen, P.; Laursen, M.S.; Larsen, M.; Steen, K.A.; Green, O.; Karstoft, H.; Jørgensen, R.N. Fieldsafe: Dataset for obstacle detection in agriculture. *Sensors* **2017**, *17*, 2579.

[12] Choi, Y.; Kim, N.; Hwang, S.; Park, K.; Yoon, J.S.; An, K.; Kweon, I.S. KAIST multi-spectral day/night data set for autonomous and assisted driving. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 934–948.

[13] Shin, U.; Park, J.; Kweon, I.S. Deep Depth Estimation From Thermal Image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 1043–1053.

[14] Yin, J.; Li, A.; Li, T.; Yu, W.; Zou, D. M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots. *IEEE Robot. Autom. Lett.* **2021**, *7*, 2266–2273.

[15] Tsoulias, N.; Jörissen, S.; Nüchter, A. An approach for monitoring temperature on fruit surface by means of thermal point cloud. *MethodsX* **2022**, *9*, 101712.

[16] Yue, Y.; Yang, C.; Zhang, J.; Wen, M.; Wu, Z.; Zhang, H.; Wang, D. Day and night collaborative dynamic mapping in unstructured environment based on multimodal sensors. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 2981–2987.

[17] Geiger, A.; Moosmann, F.; Car, Ö.; Schuster, B. Automatic camera and range sensor calibration using a single shot. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 3936–3943.

[18] Krishnan, A.K.; Stinnett, B.; Saripalli, S. Cross-calibration of rgb and thermal cameras with a lidar. In Proceedings of the IROS 2015 Workshop on Alternative Sensing for Robot Perception, Hamburg, Germany, 28 September–2 October 2015.

[19] Krishnan, A.K.; Saripalli, S. Cross-calibration of rgb and thermal cameras with a lidar for rgb-depth-thermal mapping. *Unmanned Syst.* **2017**, *5*, 59–78.

[20] Borrmann, D. *Multi-Modal 3D Mapping-Combining 3D Point Clouds with Thermal and Color Information*; Universität Würzburg: Würzburg, Germany, 2018.

[21] Dalirani, F.; Heidari, F.; Rahman, T.; Cheema, D.S.; Bauer, M.A. Automatic Extrinsic Calibration of Thermal Camera and LiDAR for Vehicle Sensor Setups. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 4–7 June 2023; pp. 1–7.

[22] Zhang, J.; Liu, Y.; Wen, M.; Yue, Y.; Zhang, H.; Wang, D. $L^2V^2T^2$Calib: Automatic and Unified Extrinsic Calibration Toolbox for Different 3D LiDAR, Visual Camera and Thermal Camera. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 4–7 June 2023; pp. 1–7.

[23] Azam, S.; Munir, F.; Sheri, A.M.; Ko, Y.; Hussain, I.; Jeon, M. Data fusion of lidar and thermal camera for autonomous driving. In *Applied Industrial Optics: Spectroscopy, Imaging and Metrology*; Optica Publishing Group: Washington, DC, USA, 2019; pp. T2A-5.

[24] Zhang, J.; Siritanawan, P.; Yue, Y.; Yang, C.; Wen, M.; Wang, D. A two-step method for extrinsic calibration between a sparse 3d lidar and a thermal camera. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; pp. 1039–1044.

[25] Christiansen, P.; Kragh, M.; Steen, K.; Karstoft, H.; Jørgensen, R.N. Platform for evaluating sensors and human detection in autonomous mowing operations. *Precis. Agric.* **2017**, *18*, 350–365.

[26] Zhang, Z. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* **1994**, *13*, 119–152.

[27] Shin, U.; Park, J.; Kweon, I.S. Supplementary Material: Deep Depth Estimation from Thermal Image. Available online: https://openaccess.thecvf.com/content/CVPR2023/supplemental/Shin_Deep_Depth_Estimation_CVPR_2023_supplemental.pdf (accessed on 10 December 2023).

[28] Fu, T.; Yu, H.; Yang, W.; Hu, Y.; Scherer, S. Targetless Extrinsic Calibration of Stereo Cameras, Thermal Cameras, and Laser Sensors in the Wild. *arXiv* **2021**, arXiv:2109.13414.

[29] Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698.

[30] Mharolkar, S.; Zhang, J.; Peng, G.; Liu, Y.; Wang, D. RGBDTCalibNet: End-to-end Online Extrinsic Calibration between a 3D LiDAR, an RGB Camera and a Thermal Camera. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 3577–3582.

[31] Xie, J.; Kiefel, M.; Sun, M.T.; Geiger, A. Semantic instance annotation of street scenes by 3d to 2d label transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3688–3697.

[32] Bradski, G. The openCV library. *Dr. Dobb'S J. Softw. Tools Prof. Program.* **2000**, *25*, 120–123.

[33] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.

[34] Teledyne, F. *Free Teledyne FLIR Thermal Dataset for Algorithm Training*; Teledyne FLIR: Wilsonville, OR, USA, 2018.

[35] Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. *arXiv* **2023**, arXiv:2304.02643.

[36] Contributors, M. OpenMMLab's Next-Generation Platform for General 3D Object Detection. 2020. Available online: https://github.com/open-mmlab/mmdetection3d (accessed on 10 December 2023).

Chapter 4

Conclusion

In this thesis, two methods are proposed for the extrinsic calibration of thermal cameras and 3D LiDAR sensors. These methods aim to estimate the rotation matrix and translation vector between the coordinate systems of both sensors. Finding these extrinsic parameters, which describe the spatial relation between the two coordinate systems, is crucial for fusing data from both sensors.

In Chapter 2, we introduced a target-based extrinsic calibration method for thermal cameras and 3D LiDAR sensors, leveraging the correspondence between line and plane equations of the calibration target in both modalities. The calibration algorithm achieves satisfactory results with a single pose, although employing additional poses can yield slightly improved accuracy, albeit optionally. It achieves reliable results, even with just one pose, yielding a translation error of 10.82% and a rotation error of 0.51 degrees. This outperforms an alternative method that requires eight pairs for similar results. This approach alleviates the burdensome and labor-intensive process of collecting diverse poses for extrinsic calibration of thermal cameras and LiDAR sensors. The procedure is fully automated, eliminating the need for manual interactions such as point selection or providing initial (R, t) estimates. Moreover, the method is compatible with data from sparse LiDAR sensors, such as a 16-beam LiDAR.

In Chapter 3, we introduced a targetless extrinsic calibration method designed for calibrating a thermal camera with a 3D LiDAR sensor while the sensor setup is in motion on a vehicle (such as a tractor or car) in environments where humans are present, such as farm fields or streets. This calibration process involves minimizing a specifically designed loss function, which assesses the alignment of segmented humans in both thermal images and LiDAR point clouds, given a rotation matrix and translation vector pair. The optimization algorithms proposed for determining extrinsic parameters are rooted in evolutionary algorithms. Our approach demonstrates a significant 4.43% enhancement in loss compared to extrinsic parameters acquired from target-based calibration in the FieldSAFE dataset. Our method eliminates expenses tied to creating thermal camera visible calibration targets and avoids related difficulties, like cooling or heat leaks from heat-generating parts, which render them useless for calibration. Additionally, it streamlines the labor-intensive process of collecting diverse poses for calibration targets, particularly challenging in contexts like autonomous vehicles. It also tackles the problem of repetitive calibration efforts due to sensor drift or setting changes, making the process more efficient. Moreover, it enhances the accessibility of thermal cameras and 3D LiDAR fusion by eliminating the need for specific targets.

Users can choose one of the methods according to their needs. If the sensor setup is used in an environment where humans are present, such as farm fields or streets, and during data collection for calibration, there is not a lot of overlap between people in both modalities, making segmenting and detecting them difficult and prone to error, the targetless method can be used. However, if it is important to establish extrinsic parameters between the thermal camera and LiDAR sensor before the movement of the sensor setup, or if there is no guarantee of human presence under the necessary conditions for the targetless method—such as having humans at sufficient and varied distances and directions with respect the sensor setup—during the period when sensors should be extrinsically calibrated, then the target-based method can be employed.

Curriculum Vitae

Name:	Farhad Dalirani
Post-Secondary Education and Degrees:	Western University, 2021-2024 M.Sc. Computer Science Amirkabir University of Technology M.Sc. AI and Robotics, 2017 - 2020 Bu-Ali Sina University B.Sc. Computer Engineering, 2012 - 2017
Honours and Awards:	Western Graduate Research Scholarship, 2021-2023
Related Work Experience:	TA. and RA., Western University, 2021-2023 Internship, NRC-CNRC, 2023 Internship, Deeplite Inc., 2022

Publications:

- Dalirani, Farhad, and Mahmoud R. El-Sakka. 2024. "Extrinsic Calibration of Thermal Camera and 3D LiDAR Sensor via Human Matching in Both Modalities during Sensor Setup Movement" Sensors 24, no. 2: 669. https://doi.org/10.3390/s24020669

- Farzan Heidari, Dalirani, Farhad, Taufiq Rahman, Daniel Singh Cheema, and Michael A. Bauer. "A Solution for Cross-Calibration of Gaze Tracker System and Stereoscopic Scene System." In 26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023). IEEE, 2023.

- Farzan Heidari, Dalirani, Farhad, Taufiq Rahman, Daniel Singh Cheema, and Michael A. Bauer. "Multi-Depth Cross-Calibration of Gaze Tracker and LiDAR Systems." In 26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023). IEEE, 2023.

- Dalirani, Farhad, Farzan Heidari, Taufiq Rahman, Daniel Singh Cheema, and Michael A. Bauer. "Automatic Extrinsic Calibration of Thermal Camera and LiDAR for Vehicle Sensor Setups." In 2023 IEEE Intelligent Vehicles Symposium (IV), pp. 1-7. IEEE, 2023.

- Qorbani, Mohammad Ali, Farhad Dalirani, Mohammad Rahmati, and Mohammad Reza Hafezi. "A deep convolutional neural network based on U-Net to predict annual luminance maps." Journal of Building Performance Simulation 15, no. 1 (2022): 62-80.