An Automated Single-Shot LiDAR and Camera Extrinsic Calibration Method Using Image Processing

Pasindu Ranasinghe
School of Minerals and Energy Resources Engineering
University of New South Wales
Sydney, Australia
pasindu.ranasinghe@unsw.edu.au

Dibyayan Patra
School of Minerals and Energy Resources Engineering
University of New South Wales
Sydney, Australia
d.patra@unsw.edu.au

Bikram Banerjee
School of Surveying and Built Environment
University of Southern Queensland
Toowoomba, Australia
bikram.banerjee@unisq.edu.au

Simit Raval
School of Minerals and Energy Resources Engineering
University of New South Wales
Sydney, Australia
simit@unsw.edu.au

***Abstract*— LiDAR and cameras have become primary sensors for environmental perception in the recent past, providing complementary depth and visual information. Accurate extrinsic calibration between these sensors is essential for effective sensor fusion. This paper presents a fully automated, robust, and generalisable method to estimate the LiDAR-camera transformation using a simple and efficient pipeline. Unlike traditional methods, requiring specific scenes or multiple captures, our approach uses only a single point cloud and corresponding image. The LiDAR point cloud is projected into a 2D intensity image, where a custom algorithm aligns the coordinate frames of both sensors. The proposed method achieves near-pixel reprojection accuracy (1.06 pixels) with rapid execution time (<10 seconds) in a single pass. Additionally, developed with minimal dependencies, it is lightweight, easy to deploy, and customisable for various applications.**

Keywords—LiDAR-camera calibration, single-shot, image processing, sensor fusion

I. Introduction

LiDAR and camera systems are widely used to perceive and interpret the surrounding environment. LiDAR provides depth information, while cameras capture detailed visual data, making them complementary tools. Integrating these data sources involves accurately aligning the coordinate frames of both sensors, a process known as LiDAR and camera extrinsic calibration.

Despite over a decade of research on LiDAR and camera calibration techniques, new challenges continue to arise as various types of LiDAR systems are introduced to the market, each with unique characteristics. Most existing calibration methods rely on edge and plane features to detect calibration points, which is effective for traditional spinning LiDARs but often fails with solid-state LiDARs that use non-repetitive scanning patterns. Additionally, these methods require multiple captures for reliable results. The need to arrange scenes to satisfy specific algorithm requirements adds further complexity. Methods that rely on unique, non-repetitive environmental features often struggle in standard indoor settings. Learning-based methods, introduced recently, face limitations due to their need for extensive labelled datasets, which restricts their generalisability.

To address these limitations, this paper introduces a fully automated, robust, single-shot, and generalisable LiDAR and camera calibration toolbox.

- Fully Automated: The pipeline captures images and point clouds and calculates the transformation matrix without external inputs. It achieves high reliability without any false detections of key point pairs in LiDAR and camera inputs.
- Single-Shot Calibration: A single point cloud and corresponding image are sufficient for calibration, although additional data pairs can further enhance accuracy.
- Easily Deployable: The package requires only OpenCV, NumPy, and basic ROS dependencies. It can also operate offline without relying on ROS, provided the data is captured and stored in standard point cloud and image formats.
- Generalisable: The method is compatible with imaging LiDARs and solid-state LiDARs. Low-channel mechanical LiDARs may require additional preprocessing.

II. Related Work

Traditional calibration methods use rectangular boards with clear edges and corners as calibration targets [1]. Early methods required manual selection of corners and edges, which were later automated using feature-detection algorithms based on geometric constraints [2]. These features, extracted from images and point clouds, were then matched to compute the transformation matrix between the LiDAR and the camera. However, due to the sparse resolution of LiDAR data, limitations arise, as edges often fail to intersect with LiDAR scan lines, resulting in incorrect detection of target corners. To address these limitations, researchers have used objects with non-horizontal edges, such as spherical targets [3]. Their contours are detectable from multiple viewing angles. While accurate, these spherical targets required customised calibration objects and multiple orientations, making the setup complex and time-consuming [4]. Another calibration technique uses geometric features in surroundings, such as lines, vanishing points, and corners [5, 6]. These methods remove the need for artificial targets but rely heavily on scene regularity, making them less effective in unstructured outdoor settings [4].

The use of semantic features has become a recent trend in calibration. It is achieved using semantic segmentation models to identify specific objects or regions in both LiDAR point clouds and camera images. Instead of relying purely on geometric constraints, these methods align data by recognising shared elements in the scene [7, 8]. However, the accuracy of these methods depends on the performance of the segmentation models. Recently, researchers have used motion estimation to predict sensor alignment [9]. These methods are flexible and adaptable to varied environments but are limited by factors such as high computational demands and the need for extensive labelled training data for algorithm training [10].

III. Methodology

The methods section outlines the calibration approach used to determine the translation and rotation matrix that aligns the LiDAR and camera coordinate systems. The process consists of five main steps: data acquisition, converting the LiDAR point cloud into an intensity image, accurate key point detection, key point matching, and transformation calculation.

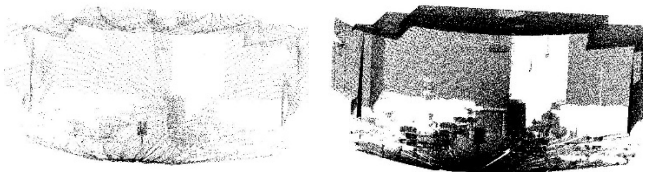
A. Data Acquisition

A checkerboard pattern with a grid size of 7×10 squares, each measuring 50 mm, was used in this experiment and placed in front of a stationary camera and LiDAR sensor setup, as shown in Fig 1.

Fig. 1. Experimental setup with the Livox Avia LiDAR, Lucid Vision camera, and processing unit.

The camera captured a single image, while the LiDAR recorded a point cloud. The duration of LiDAR data capture depended on its channel count and beam divergence. For LiDARs with over 64 channels, a single frame provides sufficient details for calibration. However, LiDARs with fewer channels require multiple frames, which are then stitched together to capture the same level of detail as a higher-channel LiDAR.

In this experiment, the Livox Avia LiDAR was used, operating in non-repetitive circular scanning mode. A one-second recording containing 10 individual frames was captured and merged to create a detailed and densified point cloud. Fig. 2 highlights the importance of densification. The single scan shown on the left is sparse and lacks detail, while the densified point cloud on the right is more detailed and has significantly fewer gaps.



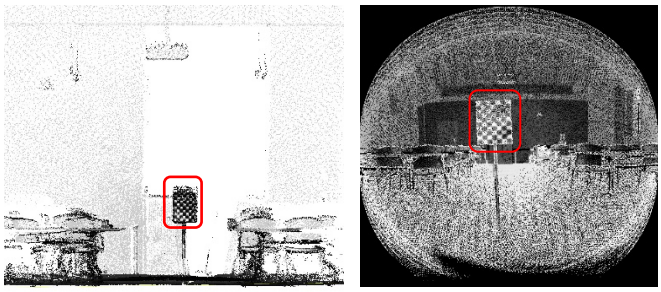
(a) (b)

Fig. 2. LiDAR point clouds (a) Before densification – single frame (b) After densification using 1-second data.

For imaging, the Lucid Vision Phoenix 2.3 MP model camera with a 4mm f/1.8 lens was used, providing a 78° horizontal and vertical field of view (FOV), effectively covering the Livox LiDAR's 70.4°×77.2° FOV. It captured high-resolution (1200p) images.

B. Intensity Image Creation

To make use of existing image processing techniques and to simplify finding correspondences between the two data modalities, the LiDAR 3D point cloud was transformed into a 2D plane using the spherical projection method as shown in Fig. 3. Each 3D point was mapped to a specific pixel on a 2D grid based on its angular position calculated from the horizontal (azimuth) and vertical (elevation) angles relative to the LiDAR. The intensity value of each pixel indicates the strength of the LiDAR signal reflected from objects in that direction. The resulting 600×600 resolution grayscale image was then normalised to enhance contrast and improve feature visibility.



(a) (b)

Fig. 3. Projection of 3D LiDAR point data onto a 2D plane: (a) Raw point cloud, (b) Intensity image, both with highlighted checkerboard pattern in red.

C. Checkerboard point detection

The next step involved detecting the checkerboard pattern within the grayscale LiDAR intensity images. Standard image processing algorithms, such as OpenCV's inbuilt "findChessboardCorners" and the checkerboard 0.2.4 Python package, work well on regular camera images but failed in this case due to several factors: sparse and uneven point density, a grainy texture, limited contrast, and the absence of RGB colour — all made it challenging for patterns to stand out. This challenge led to the development of customised solutions to detect the checkerboard corners within the intensity images.

The process began with smoothing the image by applying a Gaussian filter, which reduced noise and provided a clean foundation for further analysis. First and Second derivative features were calculated to highlight the potential points of interest. Directional derivatives were then applied to set criteria

to identify edges and corners. Potential checkerboard points are subsequently detected. The checkerboard pattern was then formed by expanding these detected points into a grid layout. Different pattern configurations were tested to find the one with the best alignment, and the selected grid was adjusted to match the actual position of the checkerboard in the scene. This method proved effective in extracting checkerboard points from LiDAR-generated intensity images and was developed using only OpenCV, with no additional dependencies, making it easy to deploy. In most instances, it successfully detects the checkerboard pattern and all points within it. However, in some cases, it misses a few points in the pattern, as shown in Fig. 4 (left).

The same method was applied to the image captured by the camera, successfully detecting all the checkerboard points within its structure. This is shown in the Fig. 4 (right).

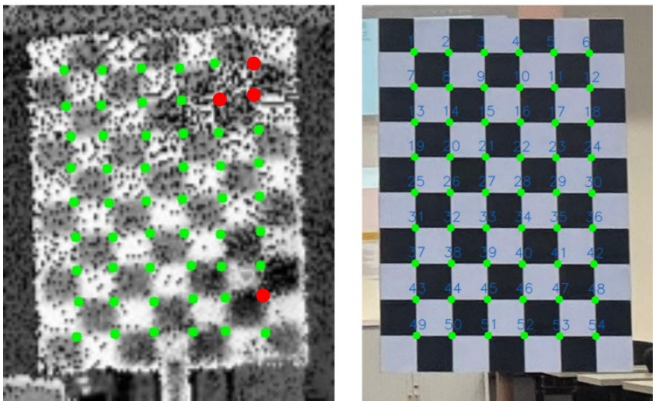


Fig. 4. Checkerboard point detection on LiDAR intensity image (Left) and the camera image (Right). In the intensity image, undetected points are shown in red, while in the camera image, all points are correctly detected and numbered.

D. Key Point Correspondence

With the corners detected, a challenge arose due to incomplete point detection in the LiDAR intensity image; some checkerboard corners are missing, making direct one-to-one matching with the camera image corners difficult. To address this, it was necessary to identify which points were detected and which were missing in the intensity image.

A virtual lattice representing the checkerboard pattern was constructed based on the detection results from the camera-captured image. The detected points from the LiDAR intensity image were then spread across this virtual grid. A KDTree (k-dimensional tree) structure was used to match the detected points to the grid. This data structure organises points to facilitate efficient nearest-neighbour searches. Each detected point from the intensity image was then mapped to its closest lattice point on the virtual grid. If the distance between an expected grid point and an actual detected point exceeded the defined tolerance, the lattice point was flagged as missing. Each corresponding image point and its LiDAR counterpart were identified and recorded. Finally, the intensity points were transformed back into their original 3D space using the inverse of the initial transformation applied to create the intensity image.

E. Transformation Calculation

After identifying the matching 3D and 2D points, the optimisation process started with Efficient PnP (E-PnP) to get the initial estimation for the rigid transformation between the LiDAR and the camera. This estimate is further refined using bundle adjustment with the Levenberg-Marquardt algorithm.

IV. Results

The checkerboard pattern was placed in both indoor and outdoor settings. Ten point clouds, each captured with an acquisition time of 1 second, along with their corresponding images, were recorded. 20-25 random corresponding points across the scene were manually selected from each pair of LiDAR and camera data. Their recorded positions served as ground truth for assessing the reprojection error. Results were calculated separately for each data pair.

No false positive points were identified in the checkerboard points detection algorithm, neither in the LiDAR intensity image nor in the RGB camera images. The optimisation algorithm was chosen based on the results shown in Table 1. Bundle adjustment with the Levenberg-Marquardt (LM) algorithm was applied to further improve the initial estimate.

TABLE I. Comparison of Average Reprojection Errors

Algorithm	Initial Reprojection Error (Pixel)	
	Initial	Final
PnP Iterative	1.110	1.110
E-PnP	1.600	1.060
PnP RANSAC	1.188	1.117

RANSAC PnP offers no significant advantage, as there are no outliers in the checkerboard detection. Results indicate that bundle adjustment is not required for iterative and RANSAC-based PnP methods. Both methods are iterative, whereas E-PnP is single-pass. E-PnP was selected, because it achieved the lowest reprojection error and faster execution time compared to the other methods, even without LM. With this method, the final average reprojection error was recorded as 1.060 pixels.

Moreover, the final results from the proposed method were then compared against the Livox Calibrator Toolkit [11], MATLAB Camera Calibration Toolbox [6], and a recent method by Koide, Oishi et al. (2023) [12]. Attempts to calibrate using the MATLAB toolbox were unsuccessful, as it could not detect the checkerboard in the LiDAR frames. The MATLAB toolbox requires the checkerboard to stand out distinctly from the scene and relies on clear detection of its four corners. Setting up a scene that allows MATLAB to accurately identify the checkerboard from LiDAR images proved to be a tedious and challenging task. In contrast, the proposed method successfully detected the checkerboard area and the grid points within it, regardless of the scene.

The single-shot results from the Livox calibrator were below acceptable standards, with reprojection errors exceeding 4 pixels. However, combining all data pairs achieved a reprojection error of 1.11 pixels. This process is entirely manual, from point-picking LiDAR frames to recording the data and only relies on the 04 corners of the checkerboard.

The method by Koide et al. relies on direct key point matching between the camera-captured image and the intensity image. However, it failed in 8 out of 10 test samples, frequently resulting in mismatches between features detected in the intensity and RGB images, as shown in Fig. 5. Additionally, it required more than 120 seconds to process a single data pair. This underscores the importance of using an external object, such as the checkerboard pattern in the proposed method, which adds significant robustness to the algorithm.

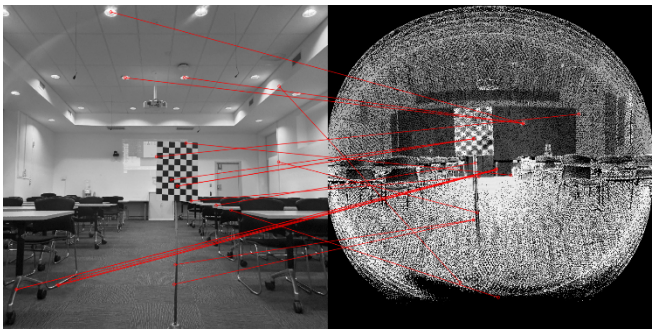


Fig. 5. The calibration results using the method by Koide, Oishi et al. (2023) with direct key point matching: RGB image (left) and intensity image (right).

To further validate the results and visually confirm the proper alignment of the two data sources, the image points were projected into the LiDAR coordinate system using the calculated final transformation matrix. This created a colourised point cloud, as shown in Fig. 6. The colours accurately correspond to the actual physical features of the scene. The zoomed view of the checkerboard area further highlights the successful integration of the data sources.

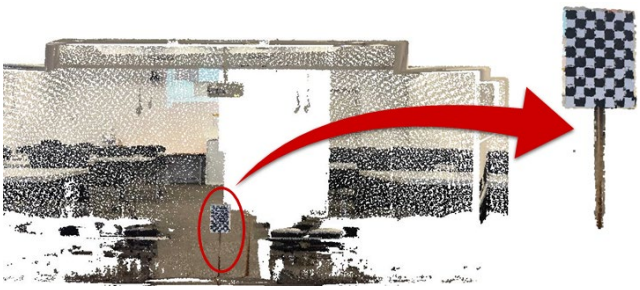


Fig. 6. Calibration results: The colourised point cloud obtained using the transformation matrix along with the zoomed view of the checkerboard pattern, illustrates the precise overlap.

The proposed method stands out for being fully automated, and highly reliable, with no false detections—even when the checkerboard is only partially visible. It requires only a single pair of LiDAR and image data to achieve near-pixel reprojection accuracy. Moreover, the method is easy to deploy, requiring only OpenCV and NumPy, making it well-suited for devices with limited computational resources. Additionally, it is the fastest among the tested methods, completing execution in under 10 seconds.

V. Conclusion

This paper presents a robust and efficient single-shot LiDAR and camera calibration method that prioritises speed, accuracy, and ease of use and deployment. The lightweight pipeline, built using only OpenCV and NumPy, converts the LiDAR point cloud into the image domain to leverage established image processing techniques. It achieves near-pixel reprojection accuracy with no false detections, even with partially visible checkerboards. The method completes execution in under 10 seconds. Its reliability, simplicity, and compatibility with a wide range of hardware platforms make it ideal for real-time applications in robotics, autonomous systems, and field environments.

References

[1] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, vol. 3, pp. 2301-2306.

[2] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2012, pp. 3936-3943, doi: 10.1109/ICRA.2012.6224570.

[3] J. Kümmerle and T. Kühner, "Unified Intrinsic and Extrinsic Camera and LiDAR Calibration under Uncertainties," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 31 May-31 Aug. 2020 2020, pp. 6028-6034, doi: 10.1109/ICRA40945.2020.9197496.

[4] Y. Wang, J. Li, Y. Sun, and M. Shi, "A Survey of Extrinsic Calibration of LiDAR and Camera," in *Lecture Notes in Electrical Engineering*, 2022, vol. 861 LNEE, pp. 933-944, doi: 10.1007/978-981-16-9492-9_92.

[5] J. Kang and N. Doh, "Automatic targetless camera–LIDAR calibration by aligning edge with Gaussian mixture model," *Journal of Field Robotics,* vol. 37, 08/01 2019, doi: 10.1002/rob.21893.

[6] MathWorks. "Lidar and Camera Calibration." www.mathworks.com/help/lidar/ug/lidar-and-camera-calibration.html

[7] W. Wang, S. Nobuhara, R. Nakamura, and K. Sakurada, *SOIC: Semantic Online Initialization and Calibration for LiDAR and Camera*. 2020.

[8] Y. Zhu, C. Li, and Y. Zhang, "Online Camera-LiDAR Calibration with Sensor Semantic Information," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 31 May-31 Aug. 2020 2020, pp. 4970-4976, doi: 10.1109/ICRA40945.2020.9196627.

[9] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal Camera-LiDAR Calibration: A Targetless and Structureless Approach," *IEEE Robot. Autom.,* vol. 5, pp. 1556-1563, 2020.

[10] G. Iyer, K. R, K. Jatavallabhula, and M. Krishna, "CalibNet: Self-Supervised Extrinsic Calibration using 3D Spatial Transformer Networks," 03/21 2018, doi: 10.48550/arXiv.1803.08181.

[11] Z. Gong, C. Wen, C. Wang, and J. Li, "A Target-Free Automatic Self-Calibration Approach for Multibeam Laser Scanners," *IEEE Transactions on Instrumentation and Measurement,* vol. 67, no. 1, pp. 238-240, 2018, doi: 10.1109/TIM.2017.2757148.

[12] K. Koide, S. Oishi, M. Yokozuka, and A. Banno, "General, Single-shot, Target-less, and Automatic LiDAR-Camera Extrinsic Calibration Toolbox," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2023, vol. 2023-May, pp. 11301-11307, doi: 10.1109/ICRA48891.2023.10160691.