

Article

Not peer-reviewed version

Singular Value Decomposition (SVD) Method for LiDAR and Camera Sensor Fusion and Self-Calibration Algorithm

[Kaigjiao Tian](#) , [KaC Cheok](#) , Micho Radovnikovich , Kazuyuki Kobayashi , [Changging Cai](#) *

Posted Date: 1 October 2024

doi: 10.20944/preprints202410.0081.v1

Keywords: LiDAR and Camera data sensor fusion; Singular Value Decomposition; Self-calibration; error detection



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Singular Value Decomposition (SVD) Method for LiDAR and Camera Sensor Fusion and Self-Calibration Algorithm

Kaiqiao Tian ¹, KaC Cheok ¹, Micho Radovnikovich ¹, Kazuyuki Kobayashi ²
and Changqing Cai ^{3,*}

¹ Electrical and Computer Engineering, Oakland University, USA

² Department of Advanced Sciences, Hosei University, Japan

³ College of Electrical and Information Engineering, Changchun Institute of Technology, 395 Kuan Ping Road, Changchun City 130103, China

* Correspondence: caicq@ccit.edu.cn

Abstract: Lidar and camera sensors are widely utilized in autonomous vehicles (AVs) and robotics due to their complementary sensing capabilities—lidar provides accurate depth information, while cameras capture rich visual data. However, effective fusion of these sensors requires precise calibration, which is often challenging due to the differences in sensing modalities. This paper presents a self-calibration algorithm based on Singular Value Decomposition (SVD) to address this problem. Singular Value Decomposition is a powerful matrix factorization technique that helps identify key calibration parameters by decomposing matrices into singular vectors and values, representing essential calibration data. Our method not only automates the calibration process but also provides a reliable mechanism for detecting calibration errors, ensuring long-term accuracy and robustness in sensor fusion. The algorithm simplifies deployment in real-world applications by eliminating the need for manual calibration or external targets. Experimental results show that our approach improves sensor alignment and fusion accuracy, making it particularly useful for AV and robotic systems where maintaining precise calibration is critical.

Keywords: LiDAR and camera data sensor fusion; singular value decomposition; self-calibration; error detection

1. Introduction

Sensor fusion is a powerful technique that involves combining data from multiple sensors to obtain a more complete and accurate understanding of the environment. The integration of different types of sensors can provide complementary information, improving the overall quality and reliability of the data. One example of sensor fusion is the combination of data from LiDAR and camera sensors. LiDAR sensors use laser beams to detect the distance and position of objects in the environment, while camera sensors capture visual information that provides valuable context about the scene. LiDAR and camera fusion method enhances the safety and reliability of Advanced Driver Assistance Systems (ADAS) [1-4].

By merging LiDAR and camera data, it becomes possible to obtain a more comprehensive and precise understanding of the environment, which in turn enables better object detection and recognition. Several methods have been developed by scholars for data processing, such as fully convolutional neural networks method [5], Deep fusion method [6], data segmentation method [7-9]. These methods aim to improve the accuracy, efficiency, and robustness of data integration for applications such as autonomous driving and robotics. Advanced algorithms help in handling sensor data discrepancies, ensuring real-time synchronization, and enhancing object detection and perception capabilities. Before performing data fusion, the primary challenge is calibrating LiDAR and camera sensors. Calibration ensures that the data from both sensors can be accurately aligned

and interpreted together. To calibrate the LiDAR Camera sensing system, we can use checkerboard for manual calibration [10-14] and increase calibration accuracy using mathematic method [15-18].

However, maintaining accurate calibration over time is a significant issue, as calibration can drift due to factors such as vibrations, temperature changes, and physical wear. Therefore, another major challenge is determining how to verify whether the calibration remains valid in the future and how to detect and correct any calibration errors that arise during long-term use. Addressing these challenges is critical for ensuring reliable sensor fusion and maintaining the overall performance of autonomous systems. Singular Value Decomposition (SVD) is a powerful mathematical tool widely used to address spatial rotation and transformation problems in various applications [19-21]. In this paper, we propose a novel method for self-calibration, self-correction, and error detection, leveraging the capabilities of SVD. Our approach allows for real-time adjustments to the calibration parameters of LiDAR and camera systems, ensuring accuracy over time. By systematically analyzing the calibration data, our method can identify and correct errors, thereby enhancing the reliability of multi-modal perception systems in advanced driver assistance applications.

2. Vehicle Platform Setup and Sensor Calibration

The calibration of LiDAR and Camera sensors is a crucial process that requires aligning their coordinate systems to facilitate precise data fusion. Due to their distinct measurement principles and units, it is necessary to transform their data into a standard coordinate system before fusing it. The calibration process usually involves collecting and preprocessing data, intrinsic and extrinsic calibration, and validation.

Calibrating LiDAR and Camera sensors can pose challenges due to their complexity and variability, and accurately modeling their internal parameters can be complex. Nonetheless, precise calibration is crucial for achieving accurate fusion of LiDAR and Camera data, which is vital for autonomous driving and other applications. Despite its importance, calibrating these sensors can be time-consuming and resource intensive. This chapter introduces a self-calibrating algorithm for LiDAR and Camera sensors to tackle these challenges. Automating the calibration process with this algorithm can significantly reduce the time and effort required for calibration while enhancing accuracy and consistency.

The algorithm development platform features a Chevrolet Cruse equipped with an Ouster 64 LiDAR mounted on the roof and a Logitech Camera attached to the windshield, as shown in Figure 1. The team designed a vehicle simulation model in ROS with identical dimensions and sensor placement to match the vehicle's length, height, and sensor position.



Figure 1. Vehicle platform and ROS simulation.

As shown in Figure 2, the TF tree for the vehicle model illustrates the spatial relationship between the vehicle body and its sensors in terms of Roll, Pitch, Yaw, x, y, and z values. The base_footprint frame represents the vehicle's lowest point on the ground, positioned at the midpoint between the rear wheels. The vehicle_cam frame displays the position of the camera in space and is also considered a child frame under the LiDAR frame

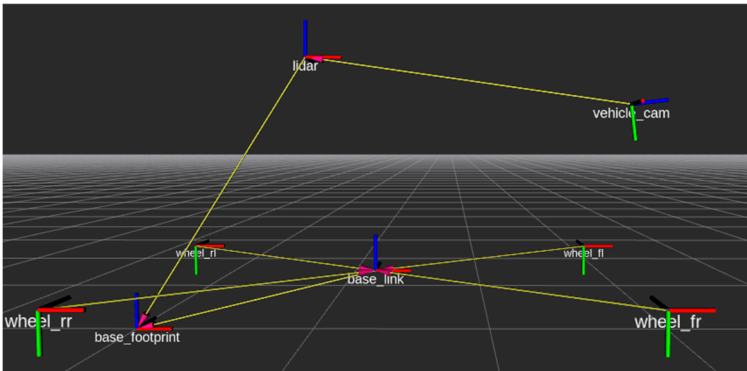


Figure 2. Vehicle platform sensor system transfer frame tree.

The accuracy of the sensor frames' transformation can significantly impact sensor fusion results. As depicted in Figure 3, LiDAR and Camera data fusion were tested using a checkerboard with two holes. In ROS RViz, the LiDAR point cloud data and vehicle TF tree are displayed, and all LiDAR points are aligned with the 2D image. The point cloud data presents the ground, tall lights, and the calibration board.

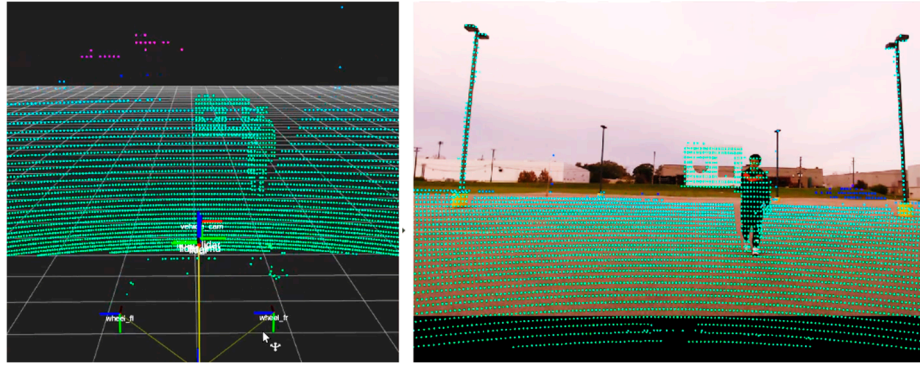


Figure 3. LiDAR and Camera manual calibration.

3. LiDAR and Camera Data Projection

To begin the LiDAR-to-2D image project, the first step involves determining the intrinsic and extrinsic parameters of the LiDAR and Camera sensors. The Camera's intrinsic parameters include the focal length, principal point, and distortion coefficients. In contrast, the extrinsic parameters describe the relative orientation and position of the LiDAR and Camera coordinate systems. With these parameters known, the 3D point cloud obtained from the LiDAR sensor can be transformed into the Camera coordinate system using the extrinsic parameters.

Transforming the 3D point cloud into the Camera coordinate system involves a rigid transformation that considers the position and orientation of the Camera relative to the LiDAR. This transformation enables the LiDAR data to be projected onto the 2D image plane of the Camera, which facilitates the creation of a 2D image that corresponds to the 3D point cloud. The accuracy of this transformation is crucial for ensuring that the resulting 2D image accurately represents the features and geometry of the 3D point cloud. PCD contains space points (x, y, z) , and 2D images are presented by 2D pixels (x, y) . The project 2D LiDAR can be obtained from the Camera intrinsic matrix and multiple extrinsic matrix.

$$P = K \times [R|t] = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & \frac{s}{f_x} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times (I|t) \times \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \quad (1)$$

The 2D rotation matrix for shown as below:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_x & -\sin\phi_x \\ 0 & \sin\phi_x & \cos\phi_x \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \quad (2)$$

$$R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Joint Rotation: } R = R_z \cdot R_y \cdot R_x$$

Joint Rotation

Figure 4 displays the outcome of the 2D projection of the PCD and the CNN classification. The YOLO neural network generated a green bounding box with a "person" label for the detected individual. A green dot on the image represents the 3D point cloud projected onto the 2D plane.

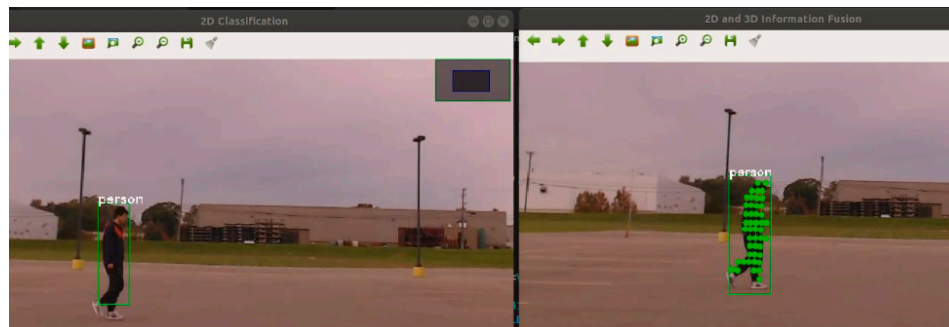


Figure 4. Camera CNN detection and LiDAR PCD 2D projection.

However, the main challenge in calibrating the Camera and LiDAR sensors for fusion is accurately determining the relative position and orientation of the two sensors. This requires defining the intrinsic parameters of each sensor (such as focal length, principal point, and distortion coefficients for cameras) and the extrinsic parameters that describe their relative positions and orientations. Calibration errors can lead to misalignment between the two-sensor data, resulting in inaccurate fusion and potentially compromising the output quality. Additionally, sensor hardware and mounting configuration differences can introduce additional challenges, requiring careful consideration during the calibration process. Figure 5 shows the LiDAR and Camera mis-calibration example in real data.



Figure 5. Calibration error results in LiDAR and Camera sensor fusion.

Another challenge in calibration is the time and effort required to collect and label data for calibration. This is particularly challenging for dynamic environments, where the scene constantly changes, and calibration data may need to be updated frequently to maintain accuracy.

Therefore, self-calibration algorithms are essential for LiDAR and camera sensor fusion because they help overcome the challenges and limitations of manual calibration methods.

4. Singular Value Decomposition Calibration Algorithm

Manual calibration of LiDAR and camera sensors can be time-consuming and lead to significant output discrepancies due to minor calibration errors. To address these challenges, a self-tuning

algorithm can refine the calibration and automatically correct any mistakes in the transfer matrix. The initial step in this process involves identifying the 2D shapes of objects in both the camera and LiDAR data. To accomplish this, a contour hull creates an outline of the contour by tracing lines around the entire image, while a convex hull creates a convex curve around the object. The main difference between a contour and a convex hull is that a contour represents the boundary of an object, while a convex hull represents the minor convex shape that can contain a set of points. Contours can be irregular and have concave sections, while a convex hull is always a convex shape with straight edges connecting the points.

As shown in Figure 6, the blue line in the picture represents the 2D image contour, while the green line represents the convex hull.

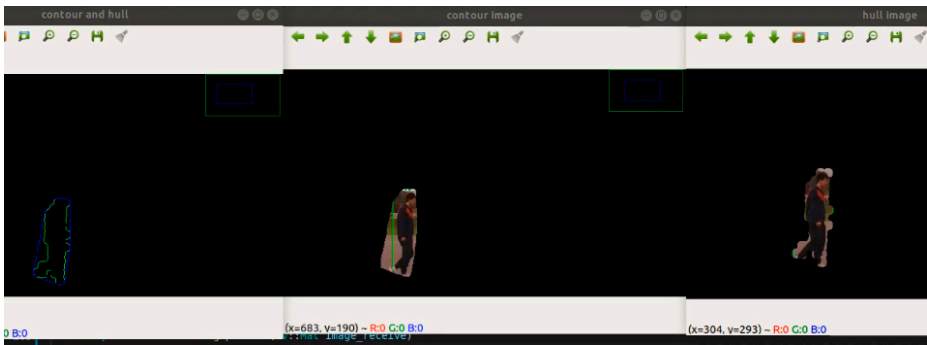


Figure 6. 2D Image pedestrian tracking using contours and convex hull algorithms.

In LiDAR and camera fusion, 2D images track individuals' locations and create contours and convex hulls. Similarly, a 2D point cloud generates an alternative representation of individuals. An algorithm is then employed to compare these shapes in scale and rotation, resulting in an accurate transfer matrix. Figure 7 showcases all the preparatory data. The top left image depicts a 2D classification with people labeled in green. The top right image illustrates object tracking with a binary image decomposed into 2D pixel points. The middle-left picture shows LiDAR points projected onto the RGB image with green dots. The middle right picture shows the hull and contour algorithm that describes the tracked object. Finally, the bottom pictures exhibit the RGB image extracted by the contour and hull algorithms.

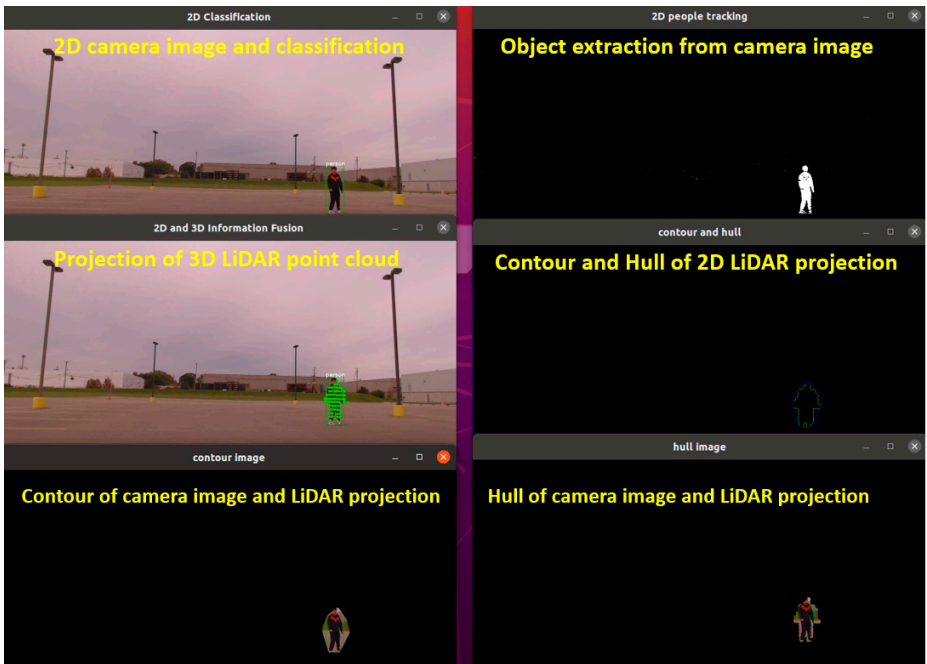


Figure 7. Combining 2D image and 3D point cloud data.

In the context of LiDAR and the Camera calibration algorithm, Singular Value Decomposition (SVD) is a powerful mathematical technique that can identify the relationship between LiDAR coordinates and camera coordinates by decomposing a matrix into three separate matrices. This helps uncover any patterns or features in the data and can be a valuable tool for understanding how the LiDAR and camera data relate to each other and optimizing the fusion algorithm. Additionally, the Gradient Descent (GD) algorithm can be used to optimize the SVD algorithm by minimizing the cost function of the machine learning model. By breaking the sensor fusion process into smaller, smoother steps, GD can help identify the distance vector and rotation matrix between LiDAR coordinates and Camera coordinates more efficiently, ultimately improving the accuracy and speed of the sensor fusion process. The LiDAR and camera calibration algorithm is a mathematical process that helps to align and synchronize the data from both sensors. This algorithm is developed and implemented using MATLAB software.

In the beginning, the algorithm takes the raw data from both LiDAR and camera sensors at time, which is shown in Figure 8. The red outline data represents the people detected by the camera in 2D, while the pink outline data represents the people detected by LiDAR. However, due to some random errors or misalignments, there might be a discrepancy between the two data sets, leading to a mis-calibration situation. This discrepancy can be in the form of incorrect scaling or rotation values.

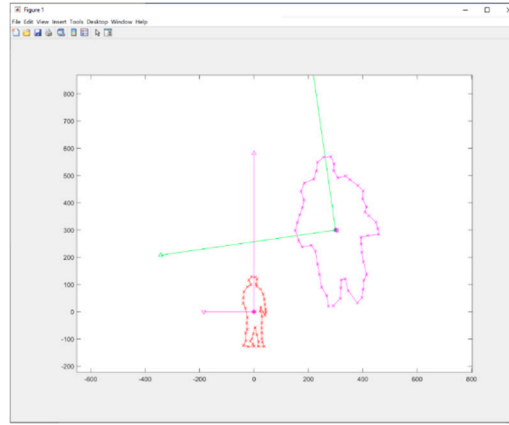


Figure 8. Camera and LiDAR contours calibration process at time t_0 .

The LiDAR and camera calibration algorithm is designed to identify and correct these random values, which will make the two systems well-matched and calibrated. By applying this algorithm, the discrepancies between the two data sets can be minimized, leading to more accurate and precise sensor fusion results.

Let's denote and respective LiDAR data as matrix p_i and Camera data as matrix q_i :

$$p_i = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_i = \begin{bmatrix} p_{x_i} \\ p_{y_i} \\ p_{z_i} \end{bmatrix} \quad q_i = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}_i = \begin{bmatrix} q_{x_i} \\ q_{y_i} \\ q_{z_i} \end{bmatrix} \quad (1)$$

With reference to the p-frame, the distance vector and rotation matrix from the p-frame to the q-frame is:

$$d_q^p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_q^p$$

$$R_q^p = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (1)$$

$$= \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\theta + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & -c\psi s\theta + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

The forward kinematics relationship between the position vector p and q is given by

$$p_i = d_q^p + R_q^p q_i \quad (1)$$

If a collection of m sets of data at each frame can be stored as matrix

$$P = [p_1 \ p_2 \ \dots \ p_m] \quad Q = [q_1 \ q_2 \ \dots \ q_m] \quad (1)$$

Then, the following matrix relationship follow:

$$P = D_q^p + R_q^p Q \quad D_q^p = d_q^p \mathbf{1}_{1 \times m} \quad (1)$$

The relationship between the means of the data is similarly related as

$$\begin{aligned} \bar{p} &= d_q^p + R_q^p \bar{q} \\ \text{where, } \bar{p} &= \frac{1}{m} \sum_{i=1}^m p_i, \bar{q} = \frac{1}{m} \sum_{i=1}^m q_i \end{aligned} \quad (1)$$

Let H be the product $H = [P - \bar{P}][Q - \bar{Q}]'$, where $[P - \bar{P}]$ be the variations of positions P from the mean position $\bar{p}$, where $\bar{P} = \bar{p} \mathbf{1}_{1 \times m}$, and $[Q - \bar{Q}]'$ where $\bar{Q} = \bar{q} \mathbf{1}_{1 \times m}$. The goal is decomposing $H = USV'$, where U and V are unitary matrices in the minimal singular value decomposition and S is a matrix of singular values of H .

Use the forward kinematics to express H as:

$$\begin{aligned} H &= [D_q^p + R_q^p Q - D_q^p + R_q^p \bar{Q}][Q - \bar{Q}]' \\ &= R_q^p [Q - \bar{Q}][Q - \bar{Q}]' \end{aligned} \quad (1)$$

Use similarity transformation for symmetric matrix to decompose relationship into:

$$\begin{aligned} H &= R_q^p T_{QQ} \Lambda_{QQ} T_{QQ}' \\ \Lambda_{QQ} &= [\text{diag}\{\lambda_j([Q - \bar{Q}][Q - \bar{Q}]'), j = 1, 2, 3; \lambda_1 \geq \lambda_2 \geq \lambda_3\}] \\ \lambda_j &= \text{eigenvalue of } [Q - \bar{Q}][Q - \bar{Q}]' \\ T_{QQ} &= \text{modal matrix of } [Q - \bar{Q}][Q - \bar{Q}]' \\ T_{QQ}' &= T_{QQ} \text{ or } T_{QQ}' = I \end{aligned} \quad (1)$$

Decomposing H using SVD:

$$H = USV' \quad (1)$$

Where,

$$\begin{aligned} S &= [\text{diag}\{\sqrt{\lambda_j(H'H)}, j = 1, 2, 3; \lambda_1 \geq \lambda_2 \geq \lambda_3\}] \\ UU' &= U'U = I \\ V'V &= VV' = I \end{aligned} \quad (1)$$

It is seen that,

$$\begin{aligned} H'H &= [Q - \bar{Q}][Q - \bar{Q}]' R_q^{p'} R_q^p [Q - \bar{Q}][Q - \bar{Q}]' \\ &= [Q - \bar{Q}][Q - \bar{Q}]'^2 \end{aligned} \quad (1)$$

So, it turns out that,

$$\begin{aligned} S &= [\text{diag}\{\sqrt{\lambda_j(H'H)}, j = 1, 2, 3\}] \\ &= [\text{diag}\{\sqrt{\lambda_j([Q - \bar{Q}][Q - \bar{Q}]'^2)}, j = 1, 2, 3\}] \\ &= [\text{diag}\{\lambda_j([Q - \bar{Q}][Q - \bar{Q}]'), j = 1, 2, 3\}] = \Lambda_{QQ} \end{aligned} \quad (1)$$

From the relationship $H = R_q^p T_{QQ} \Lambda_{QQ} T_{QQ}' = USV'$, one finds that

$$\begin{aligned} H &= R_q^p T_{QQ} \\ V' &= T_{QQ}' \end{aligned} \quad (1)$$

Therefore, the rotation matrix between LiDAR and Camera can be present as:

$$R_q^p = UV' \quad (1)$$

Once we have obtained the rotation matrix that describes the orientation between the LiDAR and Camera, we can input this result into the Gradient Descent function. This function allows us to iteratively optimize the values of rotation and the distance scale vector by minimizing the cost function. This helps us to obtain more accurate results. Once we have calculated the optimal values for these parameters, we can apply the new frame transfer tree to get a more precise estimate of the orientation and distance between the LiDAR and Camera.

The designed Gradient Descent state present as:

$$\theta = [p_{x_i}, p_{y_i}, s, radMajb] \quad (1)$$

Where, p_{x_i}, p_{y_i} are the LiDAR data (able switch to Camera data); s is the scale change between LiDAR and Camera data, initial as 1; $radMajb$ is the major rotation calculated from SVD algorithm, $radMajb = atan2(V'(2,1)V'(1,1))$

Gradient Decent iteration process:

$$\theta_{new} = \theta_{old} - \gamma \nabla J(\theta) \quad (1)$$

The derivative cost function present as:

$$\begin{aligned} \nabla J(\theta) &= [2q_1(p_{x_i} - \bar{p}_{x_i}), 2q_2(p_{y_i} - \bar{p}_{y_i}), -2q_3(radMaja - s * radMaja) * s, \\ &\quad -2q_4(radMaja - radMajb)] \end{aligned} \quad (1)$$

Where, $q = [q_1, q_1, q_1, q_1]$ is an adjust matrix.

After deriving the necessary equation for the calibration process, MATLAB can be used to implement and execute the required computations. Figure 9 provides a visual representation of the LiDAR data changing as a result of rotation, scale, and position adjustments during the gradient descent iteration from t_0 to t_n . As the calibration process iteratively optimizes these parameters, the LiDAR data eventually matches the target data with minimal error in rotation, scale, and position. This figure allows us to see the calibration process in action and visualize the changes to the LiDAR data as the calibration progresses.

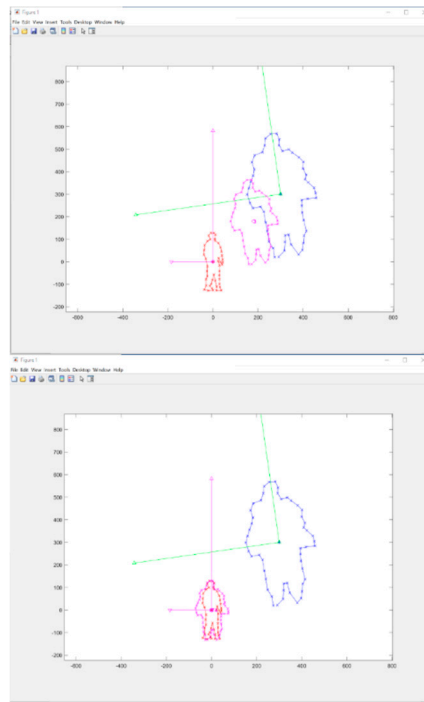


Figure 9. Camrea and LiDAR calibration process at time $t_{n/2}$ and t_n

The combination of the Singular Value Decomposition (SVD) and Gradient Descent (GD) algorithms provides a highly effective approach for completing the LiDAR and Camera calibration task. We can achieve fast and accurate results by tuning the parameters of both algorithms. Additionally, the designed algorithm is not limited to a specific type of sensor or calibration system, making it versatile and applicable in various contexts.

5. Results

Figure 10 illustrates the mis-calibration of the LiDAR-Camera sensor fusion, and correction from t_0 to t_n . The LiDAR data is represented by blue points, the Camera data by red points, and the processing state by pink points.

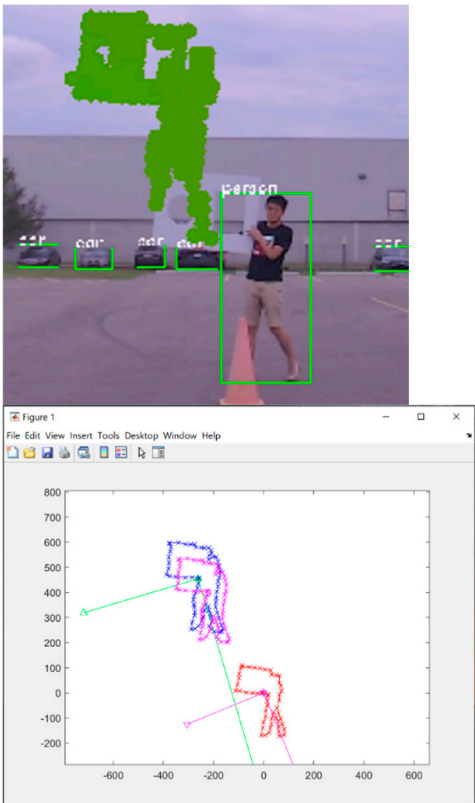


Figure 10. Schemes follow the same formatting.

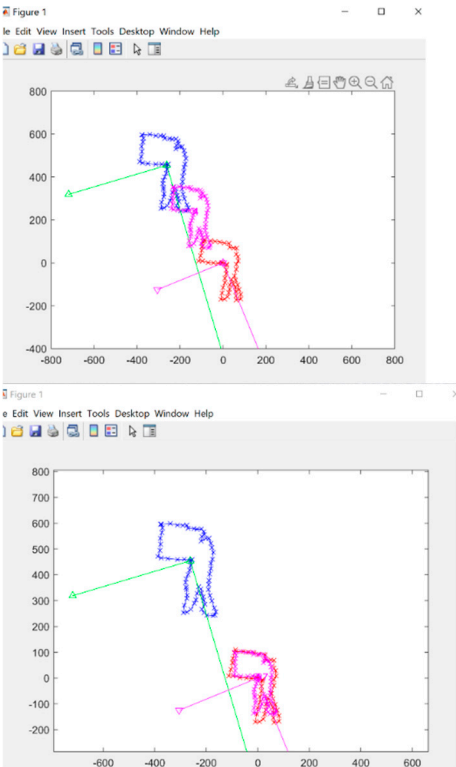


Figure 10. Self-calibration example 1 at time $t_{n/2}$ and t_n

Figure 12 displays an example of mis-calibration where the LiDAR data exhibits scale change and offset in relation to the Camera data. This may be caused by errors in the Camera's TF tree or distortion calibration. On the other hand, this figure also illustrates how the self-calibration algorithm can effectively solve this issue.

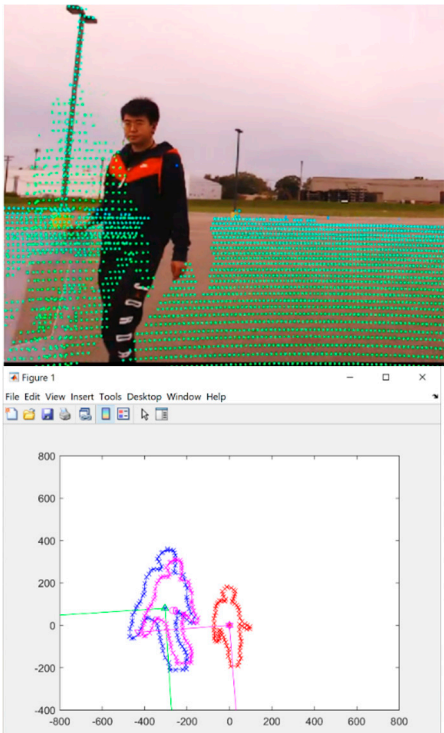


Figure 12. Schemes follow the same formatting.

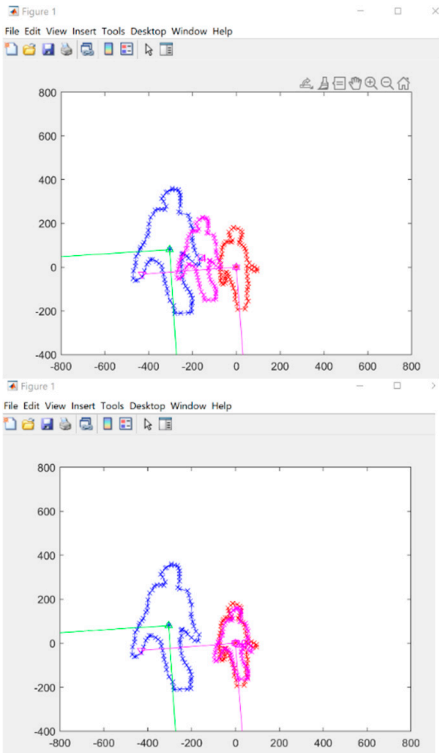


Figure 13. Self-calibration example 2 at time $t_{n/2}$ and t_n

When self-driving vehicles operate on the highway, the position of their sensors may be affected by strong winds or vibrations. As illustrated in Figure 14, the LiDAR data appears to be positioned above the Camera detection. This figure also demonstrates the algorithmic processing involved in addressing this issue.

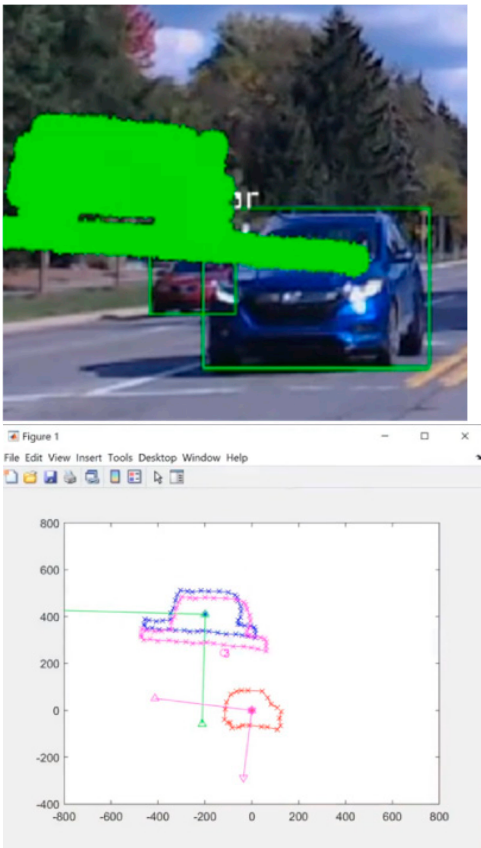


Figure 14. Self-calibration example 3 at time $t_{n/2}$ and t_n .

To test the limitations of the designed LiDAR and Camera self-calibration algorithm, we deliberately introduced manual offsets and rotations to the LiDAR data and subsequently applied the algorithm. The results are visually represented in Figure 18. Upon examining the outcomes, it is evident that the algorithm is capable of accurately identifying the correct transformation matrix from LiDAR to Camera, even in the presence of these intentional distortions

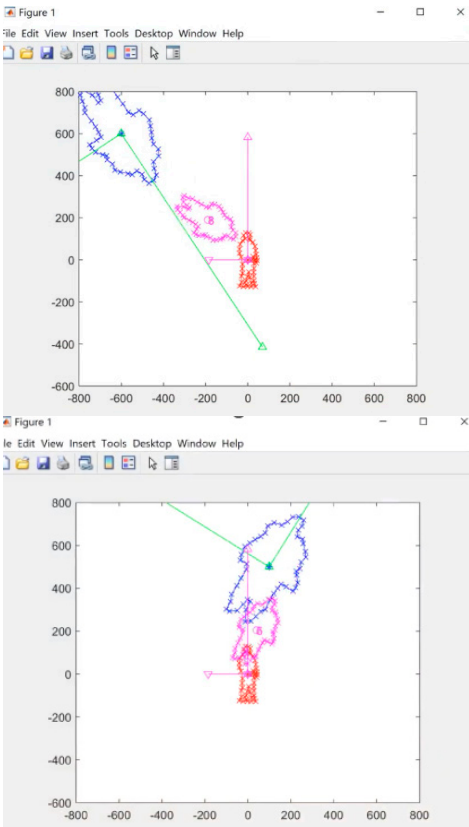


Figure 15. Self-calibration example 5 and 6 at time $t_{n/2}$

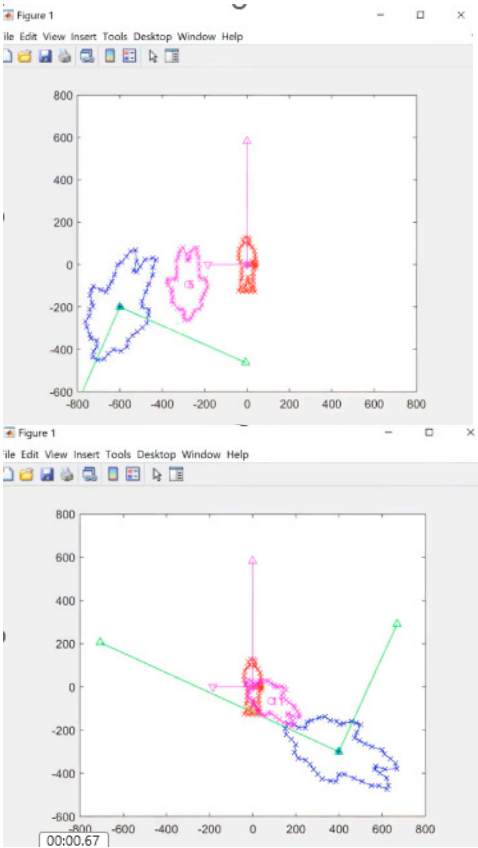


Figure 16. Self-calibration example 7 and 8 at time $t_{n/2}$

6. Conclusions

This paper addresses the implementation of a LiDAR-Camera perception system for standard home-use vehicles. It discusses the significant challenges involved in calibrating LiDAR and camera sensors, as potential miscalibration can lead to dangerous inaccuracies in object detection, which is critical for the safety and reliability of autonomous vehicles. The chapter introduces a robust self-calibration algorithm specifically designed for the LiDAR-Camera system. This algorithm leverages 2D image classification results obtained from Convolutional Neural Networks (CNN) alongside point cloud 2D projection data to ensure precise alignment of sensor data. The integration of CNNs enhances the system's ability to accurately classify and identify objects within the captured images, thereby improving the effectiveness of the calibration process.

Moreover, this algorithm incorporates self-calibration, self-correction, and error detection capabilities, collectively enhancing overall accuracy and robustness. The self-calibration process continuously adjusts the sensor parameters based on real-time data, while self-correction actively compensates for any detected discrepancies in sensor alignment. Error detection mechanisms further ensure that any potential inaccuracies are identified and addressed promptly, minimizing risks during operation. To determine the optimal transformation matrix for the sensors, the algorithm employs the Singular Value Decomposition (SVD) method in conjunction with gradient descent techniques. This powerful combination allows for effective integration of data from both LiDAR and camera sources, ensuring accurate sensor fusion and improving detection reliability across varying conditions.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER, grant number XXX" and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at <https://search.crossref.org/funding>. Any errors may affect your future funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: All authors would like to sincerely thank the reviewers and editors for their suggestions and opinions for improving this article.

Conflicts of Interest: The authors declare no conflicts of interest

References

1. Zhang, F., Clarke, D. and Knoll, A., 2014, October. Vehicle detection based on LiDAR and camera fusion. In 17th International IEEE Conference on Intelligent Transportation Systems (ITSC) (pp. 1620-1625). IEEE. Author 1, A.; Author 2, B. Title of the chapter. In Book Title, 2nd ed.; Editor 1, A., Editor 2, B., Eds.; Publisher: Publisher Location, Country, 2007; Volume 3, pp. 154–196.
2. Caltagirone, L., Bellone, M., Svensson, L. and Wahde, M., 2019. LIDAR–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111, pp.125-131. Author 1, A.B.; Author 2, C. Title of Unpublished Work. Abbreviated Journal Name year, phrase indicating stage of publication (submitted; accepted; in press).
3. Zhao, X., Sun, P., Xu, Z., Min, H. and Yu, H., 2020. Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications. *IEEE Sensors Journal*, 20(9), pp.4901-4913.
4. Zhong, H., Wang, H., Wu, Z., Zhang, C., Zheng, Y. and Tang, T., 2021. A survey of LiDAR and camera fusion enhancement. *Procedia Computer Science*, 183, pp.579-588. Title of Site. Available online: URL (accessed on Day Month Year).
5. Caltagirone, L., Bellone, M., Svensson, L. and Wahde, M., 2019. LIDAR–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111, pp.125-131.
6. Li, Y., Yu, A.W., Meng, T., Caine, B., Ngiam, J., Peng, D., Shen, J., Lu, Y., Zhou, D., Le, Q.V. and Yuille, A., 2022. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 17182-17191).
7. Zhao, L., Zhou, H., Zhu, X., Song, X., Li, H. and Tao, W., 2023. Lif-seg: Lidar and camera image fusion for 3d lidar semantic segmentation. *IEEE Transactions on Multimedia*, 26, pp.1158-1168.
8. Berrio, J.S., Shan, M., Worrall, S. and Nebot, E., 2021. Camera-LIDAR integration: Probabilistic sensor fusion for semantic mapping. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), pp.7637-7652.

9. Li, J., Zhang, X., Li, J., Liu, Y. and Wang, J., 2020. Building and optimization of 3D semantic map based on Lidar and camera fusion. *Neurocomputing*, 409, pp.394-407.
10. Huang, J.K. and Grizzle, J.W., 2020. Improvements to target-based 3D LiDAR to camera calibration. *IEEE Access*, 8, pp.134101-134110.
11. Pusztai, Z. and Hajder, L., 2017. Accurate calibration of LiDAR-camera systems using ordinary boxes. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 394-402).
12. Velas, M., Španěl, M., Materna, Z. and Herout, A., 2014. Calibration of rgb camera with velodyne lidar.
13. Grammatikopoulos, L., Papanagnou, A., Venianakis, A., Kalisperakis, I. and Stentoumis, C., 2022. An effective camera-to-LiDAR spatiotemporal calibration based on a simple calibration target. *Sensors*, 22(15), p.5576.
14. Tsai, D., Worrall, S., Shan, M., Lohr, A. and Nebot, E., 2021, September. Optimising the selection of samples for robust lidar camera calibration. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (pp. 2631-2638). IEEE.
15. Li, X., Xiao, Y., Wang, B., Ren, H., Zhang, Y. and Ji, J., 2023. Automatic targetless LiDAR-camera calibration: a survey. *Artificial Intelligence Review*, 56(9), pp.9949-9987.
16. Li, X., Duan, Y., Wang, B., Ren, H., You, G., Sheng, Y., Ji, J. and Zhang, Y., 2024. Edgecalib: Multi-frame weighted edge features for automatic targetless lidar-camera calibration. *IEEE Robotics and Automation Letters*.
17. Tan, Z., Zhang, X., Teng, S., Wang, L. and Gao, F., 2024. A Review of Deep Learning-Based LiDAR and Camera Extrinsic Calibration. *Sensors*, 24(12), p.3878.
18. Yamada, R. and Yaguchi, Y., 2024. Probability-Based LIDAR-Camera Calibration Considering Target Positions and Parameter Evaluation Using a Data Fusion Map. *Sensors*, 24(12), p.3981.
19. Bisgard, J., 2020. Analysis and linear algebra: the singular value decomposition and applications (Vol. 94). American Mathematical Soc..
20. Jaradat, Y., Masoud, M., Jannoud, I., Manasrah, A. and Alia, M., 2021, July. A tutorial on singular value decomposition with applications on image compression and dimensionality reduction. In *2021 international conference on information technology (ICIT)* (pp. 769-772). IEEE.
21. He, Y.L., Tian, Y., Xu, Y. and Zhu, Q.X., 2020. Novel soft sensor development using echo state network integrated with singular value decomposition: Application to complex chemical processes. *Chemometrics and Intelligent Laboratory Systems*, 200, p.103981.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.