Single Frame Lidar and Stereo Camera Calibration Using Registration of 3D Planes

Ashutosh Singandhupe, Hung Manh La *IEEE Senior Member*

2021 Fifth IEEE International Conference on Robotic Computing (IRC) | 978-1-6654-3416-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IRC52146.2021.00026

Abstract—**This work focuses on finding the extrinsic parameters (rotation and translation) between the lidar and the stereo camera setups. We use a planar checkerboard and place it inside the Field-of-View (FOV) of both the sensors, where we extract the 3D plane information of the checkerboard acquired from the sensor's data. The planes extracted from the sensor's data are used as reference data sets to find the relative transformation between the two sensors. We use our proposed method Correntropy Similarity Matrix Iterative Closest Point (CoSM-ICP) algorithm to estimate the relative transformation. In this work, we use a single frame of the point cloud data acquired from the lidar sensor and a single frame from the calibrated Stereo camera point cloud to perform this operation. We evaluate our approach on a simulated dataset since it has the freedom to evaluate under multiple configurations. Through results, we verify our approach under various configurations.**

I. Introduction

Primitive approaches for an autonomous multi-sensor system involve a predefined setup where the sensors are placed at known locations relative to each other. This setup does not necessarily involve any calibration methodology since the sensors' relative translation and rotation components are known. However, with the advent of unique designs of autonomous systems in the market and the research community, it has become essential for automatic and efficient calibration methods for multi-sensor setups. Calibration in a multi-sensor system is essentially finding the relative transformation between the sensors. With our focus on autonomous navigation, lidar and stereo camera configurations are explicitly designed to the requirement of the autonomous system. These sensors, however, need to be calibrated (finding the relative rotation and translation between the two sensors) for efficient autonomous navigation [8], [11], [13]–[15].

Early methods that involve extrinsic calibration between a lidar and a camera (or a stereo camera) made use of a calibration card or some well-defined calibration objects [2], [3], [6], [8], [16]. A planar checkerboard is one of the most widely used calibration objects since it can easily extract features from both the lidar and the camera data. Most of these approaches have highlighted the importance of placing the calibration objects in the Field-of-View (FOV) of both the sensors. Finding the correspondences between the lidar and the stereo data points is crucial for efficient lidar and stereo camera calibration. The correspondences are calculated either manually or automatically using feature extraction algorithms. The accurate estimation of the correspondences is essential for efficient calibration.

In this work, we propose an easy and efficient process to perform the lidar-stereo camera calibration using a checkerboard calibration target. Our work is similar to the work proposed by [5], where we compute the plane coefficients using the data from both the lidar and the stereo camera 3D point data. Later on, these coefficients are used to construct a well-structured set of 3D points residing in that plane. Our work differs from the above mentioned works, where we use our own proposed Correntropy Similarity Matrix (CoSM ICP) approach [12] for aligning the points in the plane and compute the relative transformation between the sensors. In essence, the significant contributions of our work are outlined as follows:

- We develop our algorithm based on finding planes acquired from both the lidar and the stereo camera's data.
- We compute the plane coefficients separately from both the sensor's data.
- From the plane coefficients acquired from both the sensors, we determine the plane's location and populate the plane with structured data points.
- Our proposed algorithm only needs to populate a limited number of points for the plane to plane matching.
- We use our CoSM ICP approach to find the relative transformation between the points present in the plane.

The remaining paper is organized as follows: Section II describes our implementation of the proposed methodology. The results and evaluation of our proposed methodology are discussed in Section III, and the conclusions are given in Section V.

II. Proposed Methodology

In this section, we describe the steps involved in our lidar-stereo calibration. We perform different steps corresponding to the data acquired from the 3D lidar sensor and the stereo camera data. The overall procedure involved in our work is shown in Fig. 1. In this work, use only a single frame of the lidar's 3D data and the stereo camera's 3D data, which contains the 3D points of the calibration target. For the 3D lidar sensor, we manually select the region containing the 3D points corresponding to the calibration target (checkerboard). We do the same for the stereo camera data, and we assume that the camera is calibrated, and we know the intrinsic parameters of the cameras. It was convenient for us to use the same calibration target for camera calibration as well. Again, the key

This work is supported by the U.S. National Science Foundation (NSF) under grants NSF-CAREER: 1846513 and NSF-PFI-TT: 1919127, and the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) under Grant No. 69A3551747126 through INSPIRE University Transportation Center. The views, opinions, findings and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NSF and USDOT/OST-R.

The authors are with the Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA. Corresponding author: Hung La, email: hla@unr.edu.

The source code of our implementation is available at the ARA lab's github: https://github.com/aralab-unr/Lidar_Stereo_Calibration

978-1-6654-3416-4/21/$31.00 ©2021 IEEE
DOI 10.1109/IRC52146.2021.00026

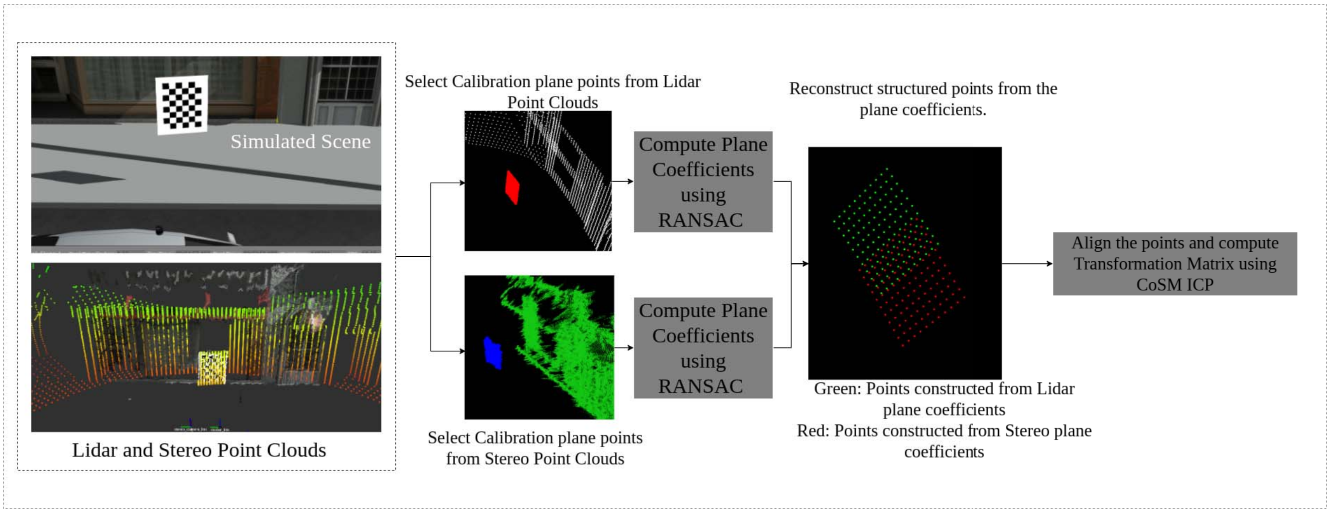

Simulated Scene
Lidar and Stereo Point Clouds
Select Calibration plane points from Lidar Point Clouds
Select Calibration plane points from Stereo Point Clouds
Compute Plane Coefficients using RANSAC
Compute Plane Coefficients using RANSAC
Reconstruct structured points from the plane coefficients.
Green: Points constructed from Lidar plane coefficients
Red: Points constructed from Stereo plane coefficients
Align the points and compute Transformation Matrix using CoSM ICP

Fig. 1: Flowchart of our approach.

point is to determine the plane coefficients acquired from both the lidar's 3D data and the stereo camera's computed 3D data. Fig. 1, as read from left to right, initially shows the simulated scene setup that includes a lidar and a stereo camera separated by a certain transformation. The left camera center defines the stereo camera's frame of reference. We directly show the 3D computed points from the stereo sensor setup for simplicity (3D points are computed using the well-known Semi-Global Block Matching (SGBM) [4] of the image pairs). We manually select the region, which contains the points corresponding to the calibration target from both the lidar 3D point clouds and the stereo computed 3D point cloud. We compute the plane coefficients for each of these selected regions using Random Sample Consensus (RANSAC). We construct a well-structured set of points from the computed plane coefficients (from the lidar and the stereo sensor data). We use our CoSM ICP approach to compute the transformation between these point sets. The transformation computed by the CoSM ICP returns the relative transformation between the lidar and the stereo sensors.

A. Lidar data processing

Using a single calibration target is relatively easy and intuitive for an extrinsic calibration between multiple sensors. As mentioned earlier, the first step for our process is to capture the 3D points from the lidar sensor and the 3D points from the stereo camera that contain the calibration target. From the 3D data, we manually pick the region containing the points corresponding to the calibration target and ignore the rest. This selected data contains points of the calibration target, which is effectively a plane. Other automatic approaches, like distance filtering [10], can be used for this process, but we let the user pick the region for complete control. Now, from this 'selected' point set, we intend to get the corresponding plane coefficients. RANSAC is our choice for this process, which determines the best-fit plane of 3D points using inliers. We use the following steps for our work.

- Randomly selects 3 points from the 'selected' points.
- Compute the plane equation using these 3 points.
- Compute inliers using the computed plane with all other 3D points.
- Repeat the process with the highest inlier ratio.

For this setup, we set the maximum iterations for our RANSAC algorithm as 1000. 3D points that are within 10mm from the plane are considered inliers. The inlier ratio, which crosses 90% or more, is considered as a best-fit plane. The plane equation computed from the lidar points is given as $a_l x + b_l y + c_l z + d_l = 0$, where a_l, b_l, c_l and d_l represent the coefficients of the plane.

B. Stereo camera data processing

This section details the process of computing the 3D plane equations of the calibration target using the point clouds generated from the stereo camera setup. In this work, we assume that the camera is already calibrated, and we know its intrinsic parameters. For our experiments, we perform the calibration steps that are implemented in Robot Operating System (ROS). For camera calibration, we use the same checkerboard calibration target that we have used to perform the lidar-camera calibration as mentioned in this work. To compute the point clouds from a pair of stereo images, we use the popular Semi-Global Block Matching (SBGM) algorithm [4] to generate the depth map from the stereo image pairs. Based on the quality of the images (resolution, frame rate), we can tune multiple parameters of the SGBM algorithm (block size, speckleRange, speckleWIndowSize, etc.) to get the desired quality of the depth map for the further point cloud generation. In our framework we use the well known ***StereoSGBM()*** function provided by OpenCV [1]. The reader is suggested to refer to [4] and [1] to explore further options in point cloud generation from stereo images. We manually select the 3D points computed from the stereo camera and compute the plane equation corresponding to the 3D points using RANSAC. We follow the same steps as mentioned in the lidar data plane computation:

- Randomly selects 3 points from the 3D point clouds computed from the stereo data.
- Compute the plane equation using these 3 points.

- Compute inliers using the computed plane with all other 3D points.
- Repeat the process with the highest inlier ratio.

The plane equation computed from the camera points is given as $a_s x + b_s y + c_s z + d_s = 0$, where a_s, b_s, c_s and d_s represent the coefficients of the plane.

C. Transformation estimation

We compute the transformation between these sensors from the plane equations calculated from both the lidar and the stereo camera data. For this step, we use the computed plane equations to populate the planes with a fixed number of points separated by a known distance (e.g., 100 points). The point sets generated from this process can be 'aligned'. The resultant transformation gives us the transformation between the lidar and the camera sensor. The primary reason for this process is that the number of points in the lidar point set is different from the number of points in the computed 3D points from the stereo camera. Our CoSM ICP needs to have an equal number of points for alignment.

III. Results

We perform our initial evaluation in a simulated environment provided by Open Robotics [9]. To demonstrate our results, we start with a basic simulated dataset containing simulated lidar data and a simulated stereo camera setup (mounted on a simulated Prius model car) established in a simulated environment in Gazebo, which is shown in Fig.2 (a) and (b) [7]. The primary reason for selecting this simulation setup is that we can compare our results since we know the ground truth, and this setup contains other complexities in the environment like buildings and cars (beyond the calibration card). Our experiments test the results with a linear transformation ranging from $0.05m$ to $2.5m$ (with x, y, z axes). We set up a simple test case of lidar-stereo setup where the stereo camera faces the calibration target and is placed near the lidar sensor under various configurations (e.g., 5cm along the y axis of the lidar sensor or $\mathbf{t} = [0, 0.05, 0.0]$). We intend to calculate the transformation between the stereo's left camera (C_c) with respect to the lidar's frame L_c. The stereo camera has a baseline of $7cm$ between the left and the right cameras. Both the cameras have the resolution of 1280×720, and since we use a simulated setup, we ignore the radial and tangential distortion (both were set to 0). In this scenario, we can evaluate our approach on multiple configurations where the ground truth is already known, as it can be seen in Fig. 2. So, in this case, our problem statement is defined as finding the transformation between the lidar (L_o) and the stereo camera (S_l) setup using our proposed methodology. This setup provides a base test case to verify our algorithm. Thus it can be extended to complex real-time test cases.

A. Evaluation on Simulated data

The various configurations under which we performed our experiments are shown in Table I. It shows the original ground truth transformations between the lidar and the stereo sensor. We place the calibration target at an average distance of 2m to 3m from the calibration target throughout our experiments (this distance is with respect to the lidar coordinate frame (L_c)).

After performing the steps mentioned in our approach, we collect the computed transformations returned by our approach. Table II shows the individual estimated transformation computed using our method. The reason for separating the components is to evaluate each component separately.

TABLE I: Configuration setups used in our experiments (or Ground Truth).

Setting	$t_x(m)$	$t_y(m)$	$t_z(m)$	roll (rad)	pitch (rad)	yaw (rad)
1	0	0.4	0.0	0.0	0.0	0.0
2	0	0.1	0.0	0.0	0.0	0.0
3	0	0.05	0.0	0.0	0.0	0.0
4	0	0.6	0.0	0.0	0.0	0.0
5	0	1.2	0.0	0.0	0.0	0.0
6	0	2.2	0.0	0.0	0.0	0.0
7	0.5	0.5	0.0	0.0	0.0	0.0
8	−0.5	1.5	0.3	0.0	0.0	0.0
9	−0.5	0.5	0	0.0	0.0	−0.0
10	0.5	0.5	0	0	0	−0.523599
11	0.5	0.5	0.3	0	0.349066	−0.523599
12	0.3	0.6	0.4	0	0.349066	−0.523599
13	0.2	0.3	0.2	0.261799	0	−0.523599
14	0.7	0.2	0.9	0	0.349066	0

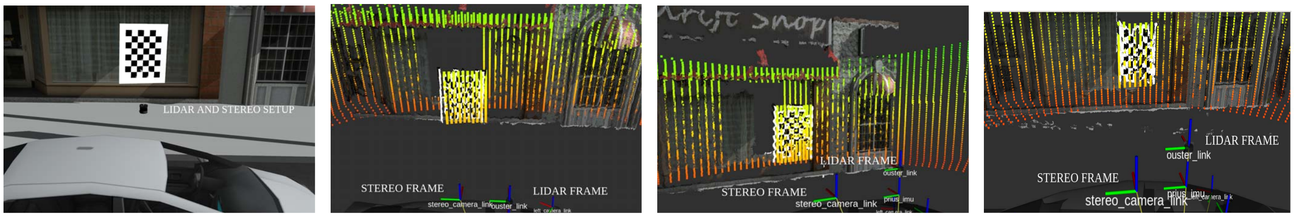
TABLE II: Individually computed transformation based on our approach.

Setting	$t_x(m)$	$t_y(m)$	$t_z(m)$	roll (rad)	pitch (rad)	yaw (rad)
1	0.019	0.342	−0.074	−0.108	−0.032	−0.0090
2	−0.044	−0.091	0.031	−0.0280	0.01700	0.0089
3	0.004	0.046	0.009	−0.0129	0.0069	0.00999
4	0.004	−0.655	0.082	0.046	0.0430	0.0100
5	−0.015	−1.147	0.036	−0.016	0.0039	0.0109
6	0.025	−2.165	0.013	0.0169	0.014	0.0099
7	−0.504	−0.468	0.004	−0.0360	0.0040	0.0059
8	0.541	−0.494	0.091	0.033	0.039	0.0090
9	−0.476	0.549	0.401	−0.093	−0.0049	−0.001993
10	0.488	0.545	0.045	−0.532	0.009	−0.00099
11	0.445	0.567	0.267	−0.5341	0.2085	−0.166
12	0.345	0.571	0.431	−0.2740	0.1023	−0.0663
13	0.163	0.291	0.201	−0.174	0.0209	−0.166
14	0.661	0.242	1.03	−0.331	0.0989	−0.123

For further evaluation, we show the average error in multiple configurations as done in the experiments. Fig.3 shows the average error of the individual components of the rotation and translation components under various configurations. From the data as given in Table II, we can see that under simple translation along x, y, z axes (no angular transformation), our algorithm performs well (with individual RMSE's $\sim$ 0.01 along with all the individual components). Even under small rotations, our approach returns relatively close values (RMSE $\sim$ 0.05) compared to the ground truth. One can see from Table II that if we have large rotations along the x, y and z axes (30 degrees in settings 10-13), the computed transformations are quite far from the ground truth.

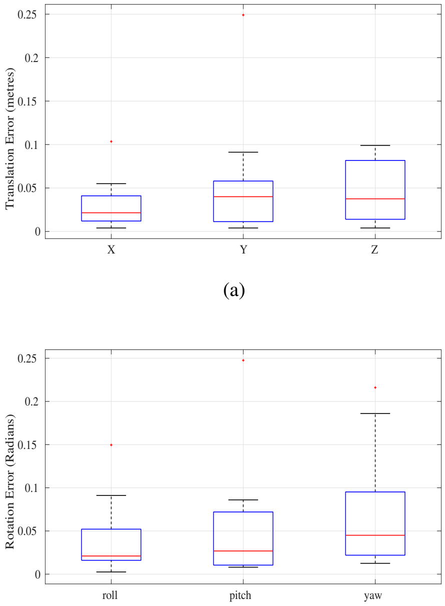
IV. Conclusions

This work proposes a novel algorithm for an efficient lidar-stereo calibration using a single frame of lidar data and the stereo camera data (3D points estimated from the stereo camera). In this work, we propose estimating the plane coefficients from both the lidar and the stereo camera data. From the computed plane coefficients (from both the sensor's data), we construct a well-spaced 3D point structure. Later, we propose to use our methodology called CoSM ICP to compute the transformation between the 'structured' points, thereby accomplishing the purpose of lidar-stereo calibration. CoSM ICP maintains one-to-one relationship between each point in the *Source* dataset and the *Target* dataset. CoSM ICP is also robust to large rotations and translations, which makes it a right



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

Fig. 2: Simulation setup for evaluating lidar-stereo calibration under multiple configurations: (a) denotes a gazebo simulation of Prius car model with lidar and stereo camera; (b) denotes the TF-frames of the lidar and the stereo camera. The link $ouster\_link$ is the reference frame for lidar; (c) and (d) denote TF-frames of multiple configurations of lidar-stereo setup. $\mathbf{t} = [t_x, t_y, t_z]$ denotes the translation component along x, y, z. It essentially denotes how the stereo camera is transformed with respect to the lidar sensor along x, y, z. For (c) the translation between the lidar and the stereo sensor is $\mathbf{t} = [0, 0.5, 0.0]$, and for (d) it is $\mathbf{t} = [-0.5, 0.5, 0]$.



(a)

(b)

Fig. 3: Translation and rotation errors of individual components (x, y, z and roll,pitch and yaw) under various configurations.

choice for this approach to be implemented in this work. One of the primary challenges we still face is the efficient detection of plane coefficients from the 3D points of the stereo data. We still face failure in estimation if we have inadequate data from the stereo point cloud. Our future work is focused on using other algorithms for accurate 3D point estimation from the stereo sensor. On the other hand, we also focus on computing the 3D plane equations using a calibrated monocular camera.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna. Lidar-camera calibration using 3d-3d point correspondences. *arXiv preprint arXiv:1705.09785*, 2017.

[3] C. Guindel, J. Beltrán, D. Martín, and F. García. Automatic extrinsic calibration for lidar-stereo vehicle sensor setups. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[4] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[5] E.-s. Kim and S.-Y. Park. Extrinsic calibration between camera and lidar sensors by matching multiple 3d planes. *Sensors*, 20(1), 2020.

[6] O. Naroditsky, A. Patterson, and K. Daniilidis. Automatic alignment of a camera with a line scan lidar system. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3429–3434, 2011.

[7] Open Source Robotics Foundation. *Vehicle and city simulation*, 2017-10-26. link:http://gazebosim.org/blog/car_sim.

[8] Qilong 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) (IEEE Cat. No.04CH37566)*, volume 3, pages 2301–2306 vol.3, 2004.

[9] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[10] R. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4, May 2011.

[11] A. Sehgal, A. Singandhupe, H. M. La, A. Tavakkoli, and S. J. Louis. Lidar-monocular visual odometry with genetic algorithm for parameter optimization. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, D. Ushizima, S. Chai, S. Sueda, X. Lin, A. Lu, D. Thalmann, C. Wang, and P. Xu, editors, *Advances in Visual Computing*, pages 358–370, Cham, 2019. Springer International Publishing.

[12] A. Singandhupe, H. La, T. D. Ngo, and V. Ho. Registration of 3d point sets using correntropy similarity matrix. *arXiv cs.CV 2107.09725*, pages 1–23, 2021.

[13] A. Singandhupe and H. M. La. A review of slam techniques and security in autonomous driving. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 602–607, Feb 2019.

[14] A. Singandhupe and H. M. La. Mcc-ekf for autonomous car security. *Proceedings of the 4th IEEE International Conference on Robotic Computing (IRC)*, Nov 2020.

[15] R. Unnikrishnan and M. Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*, 01 2005.

[16] S. Verma, J. S. Berrio, S. Worrall, and E. Nebot. Automatic extrinsic calibration between a camera and a 3d lidar using 3d point and plane correspondences. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3906–3912, 2019.