Camera-LiDAR Data Fusion for Autonomous Mooring Operation

Dipendra Subedi, Ajit Jha, Ilya Tyapin and Geir Hovland
Department of Engineering Sciences
University of Agder
4879 Grimstad, Norway

Abstract **The use of camera and LiDAR sensors to sense the environment has gained increasing popularity in robotics. Individual sensors, such as cameras and LiDARs, fail to meet the growing challenges in complex autonomous systems. One such scenario is autonomous mooring, where the ship has to be tied to a fixed rigid structure (bollard) to keep it stationary safely. The detection and pose estimation of the bollard based on data fusion from the camera and LiDAR are presented here. Firstly, a single shot extrinsic calibration of LiDAR with the camera is presented. Secondly, the camera-LiDAR data fusion method using camera intrinsic parameters and camera to LiDAR extrinsic parameters is proposed. Finally, the use of an image-based segmentation method to segment the corresponding point cloud from the fused camera-LiDAR data is developed and tailored for its application in autonomous mooring operation.**

F.1 Introduction

When thinking of autonomous shipping operations, it is also necessary to consider the autonomous mooring system. One possible solution to this problem is to incorporate a long-reach robot on the ship/vessel (shown in Fig. F.1), which requires feedback from different sensors. To undertake the mooring operations without human intervention using the robotic arm would require the arm to take the mooring rope with a loop and wrap around the bollard on the dock. Autonomous operations rely on an accurate perception of the environment with several complementary sensory modalities.

With the rapid development of range sensor technology and the advancement of machine learning algorithms using data from a camera, the use of camera-LiDAR combination for perception has gained popularity in recent years. The rich and complementary information provided by a camera and LiDAR can be used to sense the environment for autonomous operations. The camera offers better information about the color and features of the surroundings, and LiDAR provides range information. Machine learning algorithms for object detection, identification, and segmentation

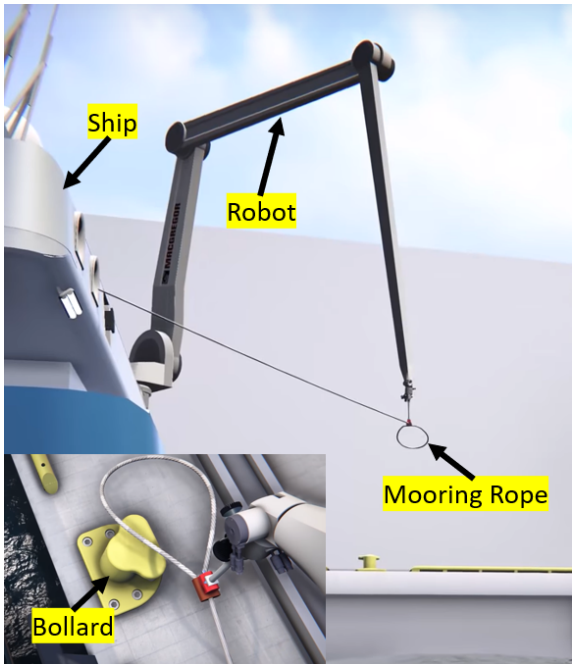


Figure F.1: Autonomous mooring operation (with permission from MacGregor Norway AS)[1]

using the camera data are matured in the literature. In contrary to the stereo camera-based vision system, the LiDAR range measurements have high accuracy for long-range depth measurements [1]. Therefore, the object pose estimation using the LiDAR range measurements is a better alternative to image-based pose estimation.

In order to utilize the information obtained from both the sensors, data from them have to be fused together so that the correspondences between image data and LiDAR point cloud could be made. The environment can be sensed better by using the fused image and point cloud data than by using individual data from each sensor. For fusing camera and LiDAR data, it is necessary to know the relative pose of sensors with respect to each other.

In recent years, the problem of determining the relative pose of the camera and LiDAR has been addressed by many researchers [1–7]. When calibrating LiDAR-LiDAR pair or LiDAR-stereo camera pair, both generating point clouds, the target-based calibration method is widely used for finding corresponding features in both point clouds and using Iterative Closest Point (ICP) to find the transform between two sensors [1]. However, in [2], a markerless online calibration method is proposed for real-time estimation of the camera to LiDAR pose. Another technique to calibrate LiDAR and camera without a need for a specific target is detailed in [5], which is based on finding maximum mutual information between the sensor-measured surface intensities in the LiDAR and the camera data. In [3], ArUco tags are used on the calibration target, and $3D - 3D$ point correspondences are used to determine

[1] `https://youtu.be/Co211gU_J5w`

the transformation between camera and LiDAR. Calibration of RGB camera with Velodyne LiDAR using a 3D marker is presented in [4]. Another method of calibrating multiple RGB cameras with a LiDAR using a spherical object is proposed in [6].

In this work, a single shot calibration method to determine the relative pose of LiDAR with respect to a camera is presented. The proposed calibration method is relatively fast compared to the existing methods.

In order to carry out the mooring operation autonomously, it is necessary to detect the bollard and estimate the pose of the bollard with respect to the robot coordinate frame. With the rapid development in machine learning methods, deep learning methods, and the boost in computing power, learning-based approaches for object classification, detection, and segmentation have attracted much research attention in recent years. In [8], deep learning-based object detection frameworks are reviewed. In this work, Mask R-CNN is used for bollard detection and segmentation because of its simplicity with promising instance segmentation and object detection results [9].

The presented work in this paper deals with the fusion of camera-LiDAR data in order to use an image-based segmentation method to segment the object of interest (bollard) and corresponding point cloud for pose estimation.

The paper is organized into five sections as follows. Section F.2 describes the intrinsic and extrinsic calibration of the camera and LiDAR. The camera-LiDAR data fusion method is elaborated in section F.3. The results obtained from the data fusion method are presented in section F.4. Conclusions and discussions follow in section F.5.

F.2 Camera and LiDAR Calibration

In computer vision, a generic camera model provides a mapping between the $3D$ world and $2D$ image given by eq. (F.1), where $x = (U, V, W)^T$ is $2D$ image point in the homogeneous form (3×1), $X = (X_w, Y_w, Z_w, 1)^T$ is $3D$ world point in the homogeneous form (4×1), and P is the camera matrix (3×4). Considering the pinhole camera model, the camera matrix P can be written as in eq. (F.2), where K is the intrinsic camera matrix given by eq. (F.3) , R is the $3D$ rotation of camera frame $\{\mathbf{C}\}$ with respect to world frame $\{\mathbf{W}\}$, t is the $3D$ translation of camera frame $\{\mathbf{C}\}$ with respect to world frame $\{\mathbf{W}\}$, (c_x, c_y) is the optical center (the principal point) in pixels, and (f_x, f_y) is the focal length in pixels. Assuming that the camera and world share the same coordinate system (i.e., $R = I_{3\times3}$ and $t = (0, 0, 0)^T$), the camera matrix can be written as in eq. (F.4). The pixel position $x' = (u, v)^T$ can be obtained from the homogeneous representation of image point x using eq. (F.5). The

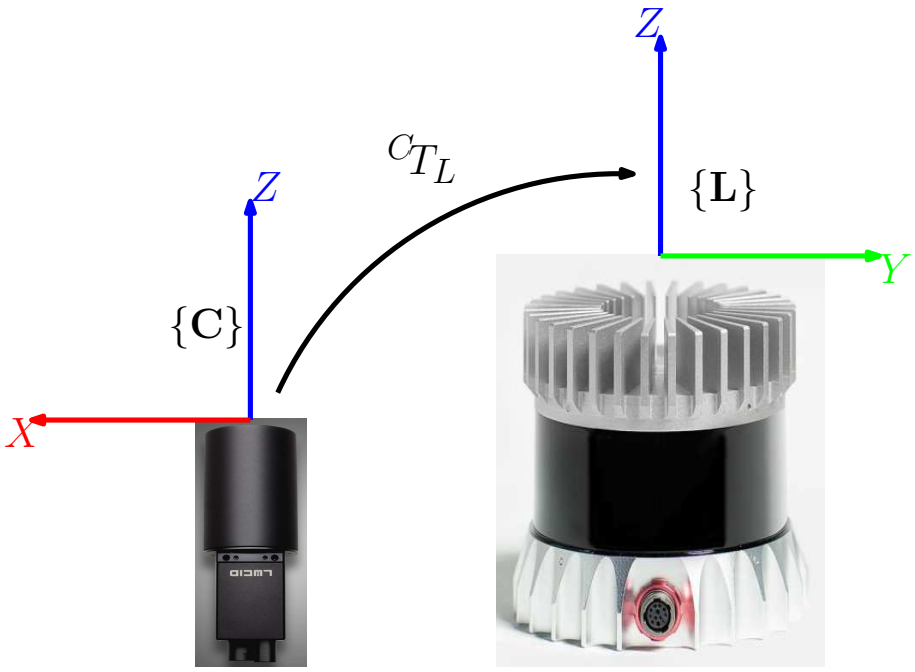


Figure F.2: Camera to LiDAR transformation

intrinsic camera matrix K is obtained from the intrinsic calibration of the camera.

$$x = PX \tag{F.1}$$

$$P = K[R|t] \tag{F.2}$$

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{F.3}$$

$$P = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{F.4}$$

$$x' = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{W} \begin{pmatrix} U \\ V \end{pmatrix} \tag{F.5}$$

To estimate the rigid body transformation (${}^{C}T_{L}$) of LiDAR coordinate frame {**L**} with respect to camera coordinate frame {**C**}, as shown in Fig. F.2, a calibration target with known dimensions, as shown in Fig. F.3, is used.

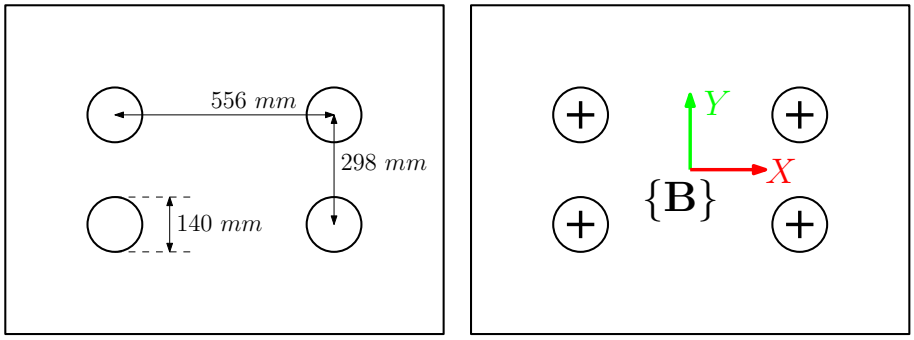


Figure F.3: Calibration target with the target coordinate frame {**B**}

F.2.1 Camera Pose Estimation

To estimate the position of the camera with respect to the calibration target, it is necessary to locate the exact position of the four circular blobs on the calibration target by using the information from the camera only. The Circle Hough Transform (CHT) is used to detect blobs on the calibration target. In order to avoid inaccurate circle detection using CHT, the calibration target should be placed parallel to the camera. The centers of the detected blobs are sorted anticlockwise, starting from the lower left center. The pose of the calibration target with respect to the camera is found using $3D - 2D$ point correspondences (OpenCV SolvePnP algorithm) [10].

F.2.2 LiDAR Pose Estimation

The point cloud obtained from the LiDAR is processed to find the calibration target plane using *PCL RANSAC parallel plane model* [11]. The edges of the planar cloud are detected based on the discontinuities in the range data of the points [2]. From the point cloud containing the planar edges, the circles are detected using the method proposed in [1]. The centers of the detected blobs are sorted anticlockwise, starting from the lower left center. The pose of the calibration target with respect to the LiDAR is found by the least-square rigid motion using the Singular Value Decomposition (SVD) technique [3, 12, 13].

Considering ${}^{L}P$ and ${}^{B}P$ are two sets of corresponding $3D$ points representing the blob center in the calibration target with respect to the LiDAR coordinate frame {**L**} and target coordinate frame {**B**}, respectively. The pose of the calibration target with respect to LiDAR is calculated by finding the optimal/best rotation ${}^{L}R_{B}$ and translation ${}^{L}t_{B}$ between these corresponding points so that they are aligned, which is shown in eq. (F.6) for point i.

$$ {}^{L}P^{i} = {}^{L}R_{B}\,{}^{B}P^{i} + {}^{L}t_{B} \tag{F.6} $$

The optimal rigid body transformation of the target coordinate frame with respect

to the LiDAR coordinate frame is found using the following steps:

1. *Calculate the centroid of both datasets.* The centroids of the points in the LiDAR coordinate frame (${}^{L}P^{c}$) and target coordinate frame (${}^{B}P^{c}$) are calculated by the average of points in each dataset as given by eq. (F.7), where $N = 4$ is the total number of points in each coordinate frame.

$$\left.\begin{aligned} {}^{L}P^{c} &= \frac{1}{N}\sum_{i=1}^{N} {}^{L}P^{i} \\ {}^{B}P^{c} &= \frac{1}{N}\sum_{i=1}^{N} {}^{B}P^{i} \end{aligned}\right\} \tag{F.7}$$

2. *Bring both datasets to the origin and calculate the optimal rotation ${}^{L}R_{B}$.* Based on the centroids computed using eq. (F.7) both datasets are re-centered to the origin, which removes the translation component from the datasets leaving only the rotational part between the datasets. Covariance matrix H is calculated using eq. (F.8), where $N = 4$ is the total number of points in each coordinate frame. The optimal rotation ${}^{L}R_{B}$ is calculated using SVD as given by eq. (F.9), where $V' = V$ if the determinant of $VU^{T} > 0$, otherwise V' is obtained by changing the sign of the third column of V.

$$\begin{aligned} H = &\Big[({}^{B}P^{1} - {}^{B}P^{c}) \cdots ({}^{B}P^{i} - {}^{B}P^{c}) \cdots ({}^{B}P^{N} - {}^{B}P^{c})\Big] \cdot \\ &\cdot \Big[({}^{L}P^{1} - {}^{L}P^{c}) \cdots ({}^{L}P^{i} - {}^{L}P^{c}) \cdots ({}^{L}P^{N} - {}^{L}P^{c})\Big]^{T} \end{aligned} \tag{F.8}$$

$$\left.\begin{aligned} \Big[U, S, V\Big] &= SVD(H) \\ {}^{L}R_{B} &= V'U^{T} \end{aligned}\right\} \tag{F.9}$$

3. *Calculate the optimal translation ${}^{L}t_{B}$.* The translation of the target coordinate frame with respect to the LiDAR coordinate frame is calculated using eq. (F.10).

$${}^{L}t_{B} = {}^{L}P^{c} - {}^{L}R_{B}\,{}^{B}P^{c} \tag{F.10}$$

F.2.3 Camera to LiDAR Transform Estimation

Once the transforms of the calibration target coordinate frame {**B**} with respect to the camera frame {**C**} (${}^{C}T_{B}$) and with respect to the LiDAR frame {**L**} (${}^{L}T_{B}$) are known, the transform of the LiDAR frame with respect to the camera frame (${}^{C}T_{L}$) is calculated using eq. (F.11).

$$ {}^{C}T_{L} = {}^{C}T_{B}({}^{L}T_{B})^{-1} \tag{F.11} $$

F.3 Camera-LiDAR Data Fusion

The point cloud in the LiDAR coordinate frame {**L**} is transformed to the camera coordinate frame {**C**} using the transformation ${}^{C}T_{L}$ obtained after calibration. The transformed point cloud with the negative Z-coordinate is filtered out; the points that are only within the field of view of the camera are kept for coloring. Filtering is needed because the points with the positive and negative Z-coordinate with the same X and Y coordinates are projected in the same image pixel coordinates resulting in the false coloring of the point cloud. The transformed point cloud is projected to the image plane using eq. (F.1). The points that are located outside the image pixel size of the camera are filtered out. The colored point cloud is obtained using the RGB values of the image pixel coordinates obtained using eq. (F.5).

F.3.1 Object Detection and Segmentation

Bollard detection and segmentation are done using Mask R-CNN [9]. The use of Mask R-CNN to detect and segment the bollard is presented in [14]. The bounding box coordinates obtained from the segmentation are used to segment the corresponding point cloud belonging to the bounding box. The overall architecture for segmenting the bollard point cloud using the fused camera and LiDAR data is shown in Fig. F.4.

F.3.2 Object Pose Estimation

Fig. F.5 shows the procedure for bollard pose estimation from the segmented bollard point cloud obtained using the proposed camera-LiDAR data fusion technique. The point cloud corresponding to the bollard in the camera coordinate frame {**C**} is transformed to the robot/world coordinate frame {**W**} with the known transformation ${}^{W}T_{C}$ obtained after the extrinsic calibration of the camera [15]. In the transformed bollard point cloud, all the planes perpendicular to the vertical axis (Z-axis pointing up from the ground) are estimated. To avoid processing of the planar point clouds with a number of inliers less than a threshold, such planar clouds are filtered out,

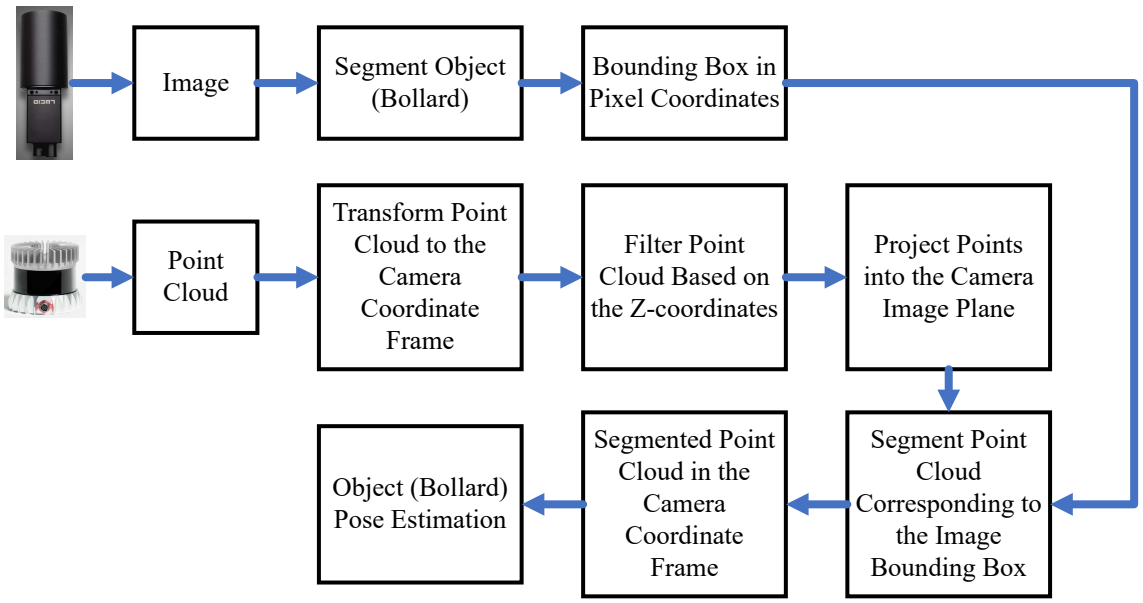


Figure F.4: Block diagram showing the camera-LiDAR data fusion for the object pose estimation

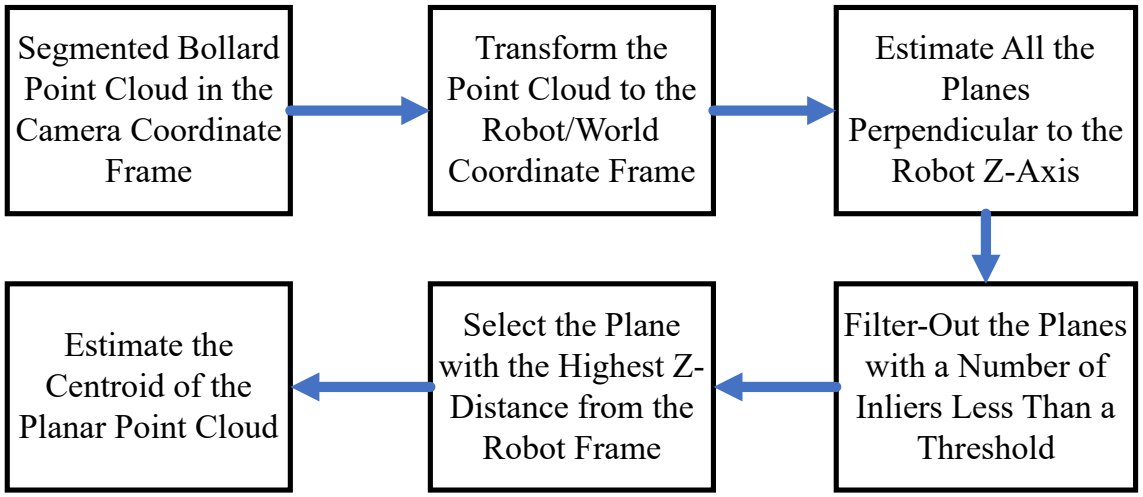
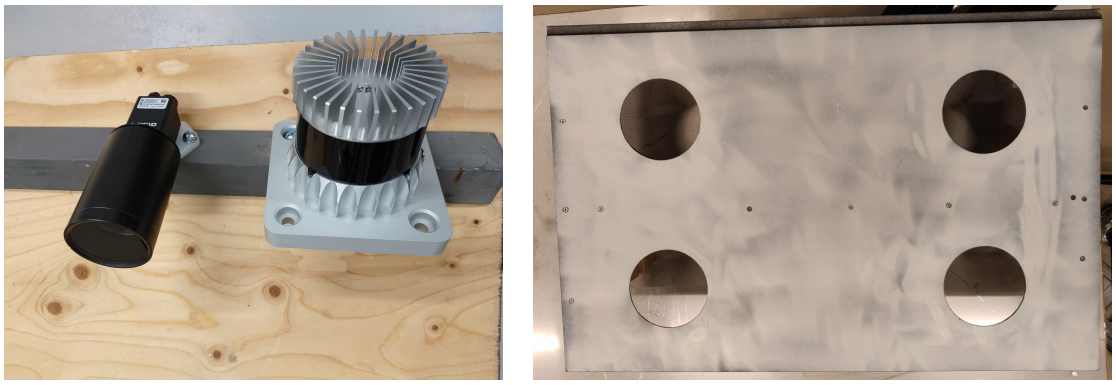


Figure F.5: Block diagram showing the bollard pose estimation

leaving only the planar clouds corresponding to the base and top of the bollard. The planar cloud corresponding to the top of the bollard is selected based on the Z-coordinate of the planes with the known information that the bollard is always located in the direction of the positive Z-coordinate with respect to robot frame. The centroid of thus obtained planar cloud representing bollard's top surface represents the position of the bollard with respect to the robot coordinate frame. The estimation of the orientation of the bollard is not taken into consideration in this work to carry-out the autonomous mooring operation irrespective of the orientation of the bollard.



(**a**) Camera-LiDAR setup (**b**) Calibration target

Figure F.6: Camera-LiDAR setup and calibration target

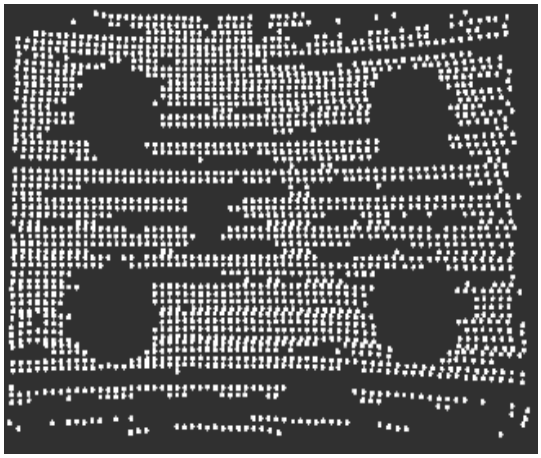
F.4 Results

The performance of the proposed camera-LiDAR data fusion approach is evaluated using 64 channel Ouster OS1-64 LiDAR and 5-megapixel Lucid Triton TRI050S-C color camera. Both sensors are mounted together in a common rig, as shown in Fig. F.6(**a**). Fig. F.6(**b**) shows the calibration target used for the extrinsic calibration of LiDAR with respect to the camera. Dimensions of the calibration target are given in Fig. F.3. The calibration target is placed around 1.4 m away from the camera-LiDAR setup within the overlapping field of view of the camera and LiDAR.

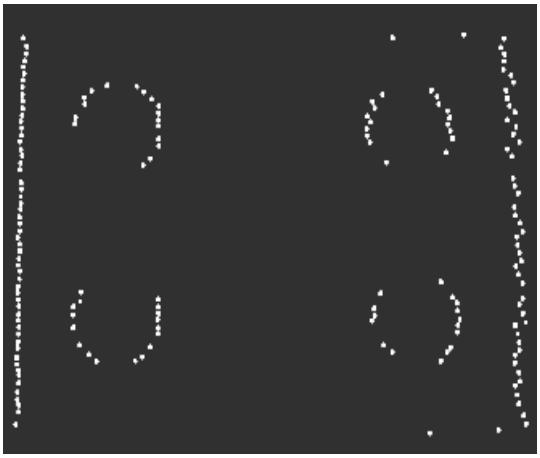
To estimate the pose of the calibration target frame with respect to the LiDAR coordinate frame, four blobs in the calibration target are detected in the LiDAR point cloud. The point cloud corresponding to the calibration target plane is shown in Fig. F.7(**a**). The edges detected in the planar cloud based on the range discontinuities of the points are shown in Fig. F.7(**b**). The blobs detected in the cloud containing the edges of the calibration target and their centroids are shown in Fig. F.7(**c**). The pose of calibration target with respect to the LiDAR calculated using the least-square rigid motion estimation is shown in Fig. F.7(**d**).

The blobs detected in the image and their centroids are shown in Fig. F.8(**a**). The pose of the calibration target with respect to the camera calculated using $3D - 2D$ point correspondences is shown in Fig. F.8(**b**).

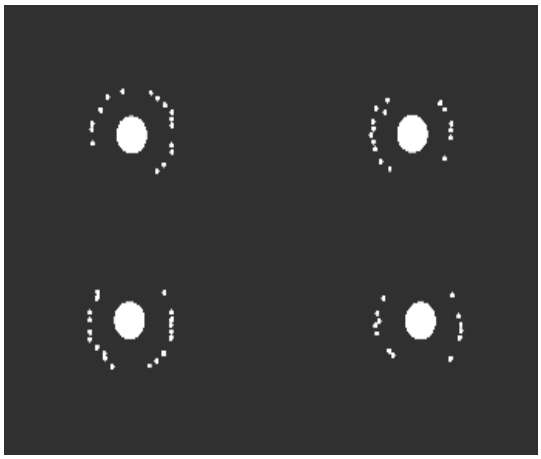
Eq. (F.11) is used to calculate the pose of the LiDAR with respect to the camera after determining the pose of the calibration target with respect to the LiDAR and camera. In order to evaluate the calibration accuracy, the re-projection error is calculated. First, the $3D$ circle centers (^{L}P) detected in the LiDAR coordinate frame are transformed to the camera coordinate frame using the transformation ($^{C}T_{L}$) obtained using the proposed calibration method. Thus transformed points (^{C}P) are projected into the image plane using eq. (F.1) to obtain the re-projected circle centers



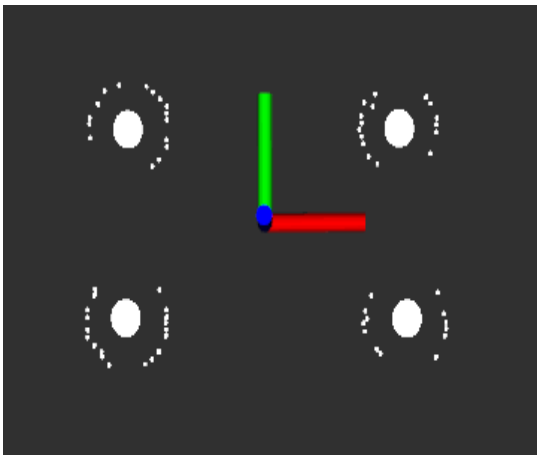
(**a**) Calibration target plane



(**b**) Calibration target edges

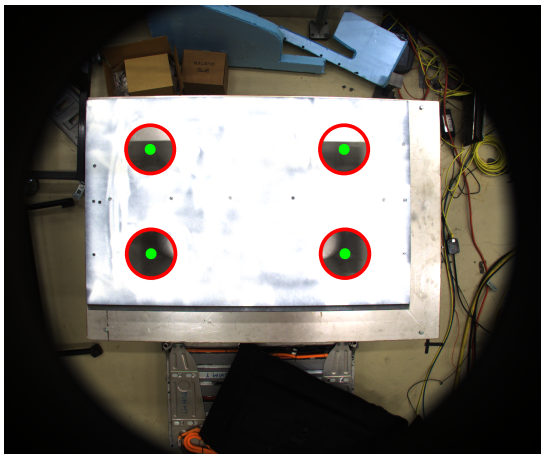


(**c**) Blob detection in the point cloud

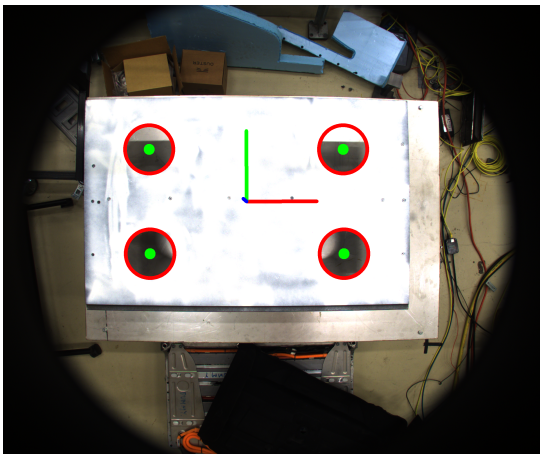


(**d**) Calibration target transform

Figure F.7: Calibration target (blobs) detection in the point cloud

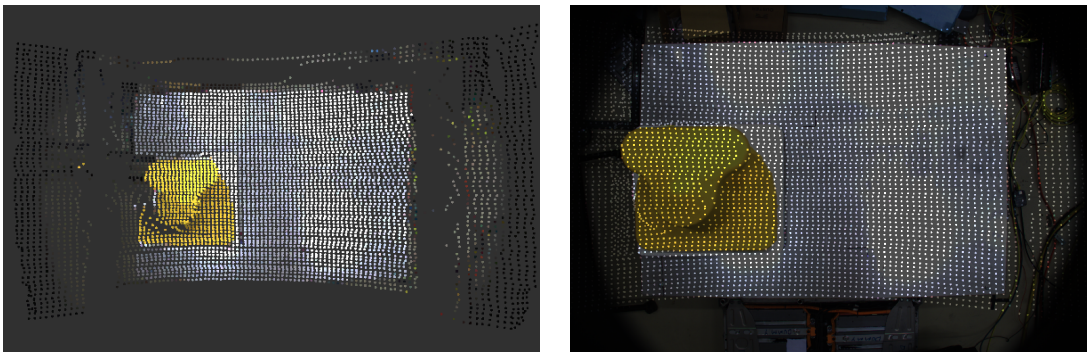


(**a**) Blob detection in the image



(**b**) Calibration target transform

Figure F.8: Calibration target (blobs) detection in the image



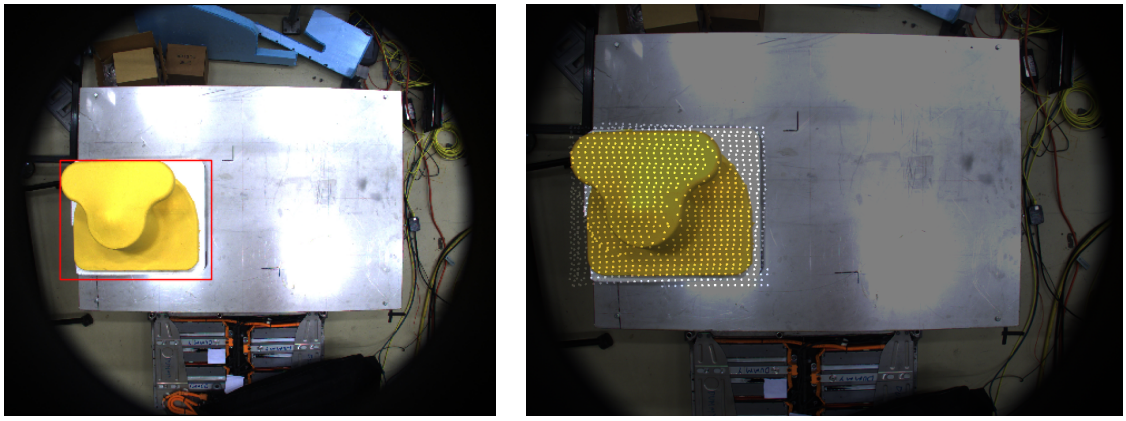
(**a**) Colored point cloud (**b**) Colored point cloud overlaid on the image

Figure F.9: Colored point clouds

(p_L) in pixel coordinates. Then, the re-projection error is calculated using eq. (F.12), where p_{Li} is the i^{th} re-projected circle center, p_{Ci} is the corresponding circle center in the image obtained from the CHT-based blob detection explained in section F.2, and $N = 4$ is the number of circles [7]. The re-projection error (e_{reproj}) obtained in the calibration when placing the calibration target at a distance of approximately 1.4 m is 1.87 $pixels$.

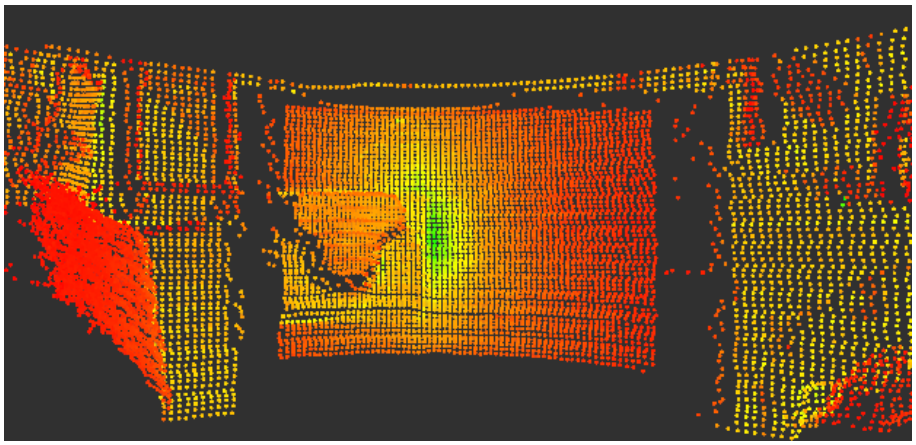
$$e_{reproj} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(||p_{Li} - p_{Ci}||_{2-norm})^2} \tag{F.12}$$

Fig. F.9(**a**) shows the colored cloud obtained after fusing the data from the camera and LiDAR within the overlapping field of view. The colored point cloud, shown in Fig. F.9(**a**), is overlaid on the image and shown in Fig. F.9(**b**). The bounding box corresponding to the bollard obtained from Mask R-CNN is shown in Fig.F.10(**a**). After fusing the camera and LiDAR data, the segmented colored point cloud overlaid on the image corresponding to the bounding box in the image is shown in Fig.F.10(**b**). To summarize, the segmented colored point cloud belonging to the bollard, shown in Fig.F.11(**b**), is extracted from the raw point cloud from the LiDAR, shown in Fig.F.11(**a**), using the proposed sensor fusion technique. Hence, the image-based segmentation method is used to segment the object, and the corresponding segmented point cloud is extracted using the sensor fusion technique presented in this paper. Thus obtained point cloud is processed to estimate the object pose.

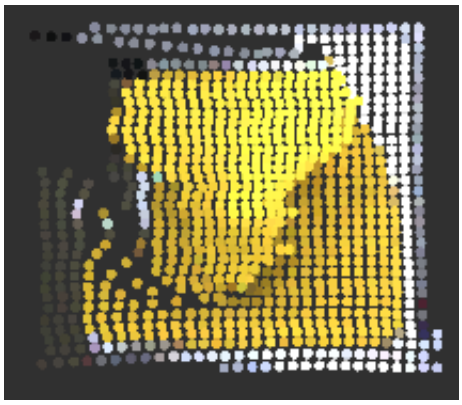


(**a**) Bollard segmentation in the image (**b**) Segmented point cloud overlaid on the image

Figure F.10: Bollard segmentation



(**a**) Raw point cloud from the LiDAR



(**b**) Segmented (colored) bollard point cloud

Figure F.11: Point cloud segmentation

F.5 Conclusions and Discussions

The extrinsic calibration of the LiDAR with respect to the camera is presented. The proposed calibration method is used with the dense LiDAR (64 channel Ouster OS1-64) in this paper. The calibration method is suitable for the sparse LiDAR (such as 16 channel Velodyne VLP-16) as well. It is because the circle detection method used in this paper only requires two LiDAR beams to intersect with each of the four circles.

The intrinsic camera parameters and camera to LiDAR extrinsic parameters are used to fuse the data obtained from the camera and LiDAR. The image-based segmentation method is used to segment the object of interest, and the corresponding point cloud is obtained from the presented data fusion technique.

The work will be extended to perform the autonomous mooring operation using the fused data from the camera and LiDAR mounted on a long-reach robotic arm, as shown in Fig. F.1. In addition, it is worth comparing the performance of the detection and pose estimation methods proposed in this paper with other state-of-the-art methods.

Acknowledgment

The work was funded by the Norwegian Research Council, project number 261647/O20, and SFI Offshore Mechatronics, project number 237896.

References – Paper F

[1] 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, Oct 2017. doi: 10.1109/ITSC.2017.8317829.

[2] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, volume 2, 2013.

[3] Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnan, and K Madhava Krishna. Lidar-camera calibration using 3d-3d point correspondences. *arXiv preprint arXiv:1705.09785*, 2017.

[4] Martin Velas, Michal Spanel, Zdenek Materna, and Adam Herout. Calibration of rgb camera with velodyne lidar. In *WSCG 2014 Communication papers proceedings*, pages 135–144, 2014.

[5] Gaurav Pandey, James R McBride, Silvio Savarese, and Ryan M Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[6] Geun-Mo Lee, Ju-Hwan Lee, and Soon-Yong Park. Calibration of vlp-16 lidar and multi-view cameras using a ball for 360 degree 3d color map acquisition. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 64–69. IEEE, 2017.

[7] Jun Zhang, Ran Zhang, Yufeng Yue, Chule Yang, Mingxing Wen, and Danwei Wang. Slat-calib: Extrinsic calibration between a sparse 3d lidar and a limited-fov low-resolution thermal camera. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 648–653. IEEE, 2019.

[8] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 2019.

[9] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[10] Gary Bradski and Adrian Kaehler. Opencv. *Dr. Dobb's journal of software tools*, 3, 2000.

[11] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.

[12] Olga Sorkine. Least-squares rigid motion using svd. *Technical notes*, 120(3):52, 2009.

[13] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.

[14] Ajit Jha, Dipendra Subedi, Per-Ove Løvsland, Ilya Tyapin, Linga Reddy Cenkeramaddi, Baltasar Beferull-Lozano, and Geir Hovland. Autonomous mooring towards autonomous maritime navigation and offshore operations. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, page to appear. IEEE, 2020.

[15] Morris Antonello, Andrea Gobbi, Stefano Michieletto, Stefano Ghidoni, and Emanuele Menegatti. A fully automatic hand-eye calibration system. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2017.