2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Targetless Multiple Camera-LiDAR Extrinsic Calibration using Object Pose Estimation

Byung-Hyun Yoon, Hyeon-Woo Jeong, and Kang-Sun Choi

Abstract— **We propose a targetless method for calibrating the extrinsic parameters among multiple cameras and a LiDAR sensor using object pose estimation. Contrast to previous targetless methods requiring certain geometric features, the proposed method exploits any objects of unspecified shapes in the scene to estimate the calibration parameters in single-scan configuration. Semantic objects in the scene are initially segmented from each modal measurement. Using multiple images, a 3D point cloud is reconstructed up-to-scale. By registering the up-to-scale point cloud to the LiDAR point cloud, we achieve an initial calibration and find correspondences between point cloud segments and image object segments. For each point cloud segment, a 3D mesh model is reconstructed. Based on the correspondence information, the color appearance model for the mesh can be elaborately generated with corresponding object instance segment within the images. Starting from the initial calibration, the calibration is gradually refined by using an object pose estimation technique with the appearance models associated with the 3D mesh models. The experimental results confirmed that the proposed framework achieves multimodal calibrations successfully in a single shot. The proposed method can be effectively applied for extrinsic calibration for plenoptic imaging systems of dozens of cameras in single-scan configuration without specific targets.**

I. Introduction

In order to acquire and perceive the external environment around autonomous vehicles or robots, multimodal sensing systems have been widely used. The multimodal sensing system typically consists of light detection and ranging (LiDAR) sensors to obtain sparse geometry information and multiple cameras providing dense color information. By fusing measurements of different characteristics together, the multimodal sensing system is capable of perceiving the environments effectively and tracking dynamic objects accurately from captured scenes. Extrinsic calibration of the multiple sensors is an essential prerequisite for accurate fusion of multimodal measurements.

Approaches to the camera-LiDAR extrinsic calibration are divided into three main categories; target-based methods, targetless methods, and learning-based methods.

As in the camera calibration, the first approach uses various calibration-dedicated targets of specific patterns including a planar checkerboard pattern, a v-shaped target, a cardboard box of three perpendicular sides, ArUco markers [1]–[8]. These target-based methods usually require laborious and time-consuming data acquisitions with the specific target, whenever the calibration process is needed. Considering that the motion of the vehicle or the robot quickly degrades the calibration initially determined, repeating the target-based calibration easily becomes tedious and impractical. To maintain an accurate calibration, an automated system that can recalibrate the sensors using observations made during the normal operations of the vehicle or the robot is required.

More general algorithms have been developed to calibrate sensors without using a specific target object. Targetless methods attempt to search for correspondences between certain geometric features can be found anywhere within artificial structures, such as lines, edges, a trihedron, and planar regions from multimodal data [9]–[12]. Multimodal measurements can be also aligned by using similarity measures including mutual information (MI) [13], normalized mutual information (NMI) [14], and a gradient orientation measure (GOM) [15]. These targetless methods can recalibrate easily as a target is no longer required. However, since the cost function is non-convex, a large number of feature correspondences should be detected to produce accurate calibration. This is why the methods require over 50 panoramic 360° images and a large number of point clouds.

Recently, learning-based methods have been presented to estimate the relation between multimodal sensors [16]–[19]. Although the methods trained in an end-to-end manner estimate the extrinsic parameters regardless of the presence of the target, the quality of the calibration significantly depends on the training data used.

Apart from the calibration approaches, rigid object pose estimation mainly for object tracking has been studied extensively in the literature [20]–[26]. Given both the accurate 3D model and color appearance model of the object to be tracked with its initial pose, the discrepancy between the posterior segmentation of the image and a synthetic object silhouette projection can be estimated [21], [23], [25], [26]. Under the assumption of a small temporal pose change, the object can be tracked in real-time by minimizing the discrepancy with respect to the object pose. Object pose estimation can be transformed to a multimodal extrinsic calibration problem in the sense that it determines the relative pose of the known object with respect to the camera origin.

In this work, we propose an automatic targetless method for calibrating the extrinsic parameters among multiple cameras and a LiDAR sensor using region-based object pose estimation. To do this, the 3D mesh model and color ap-

*This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government [2016-0-00009, Authoring Platform Technology for Next-Generation Plen-Optic Contents. This work was supported by the BK21 plus program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea.]

Byung-Hyun Yoon, Hyeon-Woo Jeong and Kang-Sun Choi are with Future Convergence Engineering, I.P.C.E., Korea University of Technology and Education, South Korea. E-mails: {dqg0602, shine1606, ks.choi}@koreatech.ac.kr

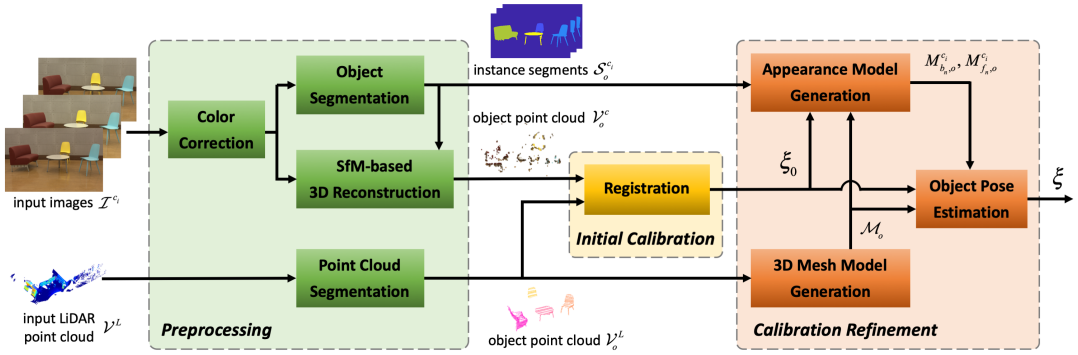


Fig. 1. The overview of the proposed multiple camera-LiDAR extrinsic calibration method.

pearance model of the arbitrary-shaped object in the scene are produced from the LiDAR data and color images, respectively. By estimating the pose of the object with the models obtained, the relative poses among multiple sensors are determined, which means the extrinsic calibration of multimodal sensors. Contrast to the targetless methods requiring certain geometric features and even sometimes manual feature matching, the proposed method exploits any objects of unspecified shapes in the scene to estimate the calibration parameters in single-scan configuration.

Our challenges for successful object pose estimation are to accurately produce the 3D mesh models of the objects in the scene, although the 3D model is partially reconstructed as a one-sided view of the object, since the 3D model is obtained from a sparse LiDAR point cloud obtained at a single location. It is also a serious challenge to infer image regions corresponding to the 3D mesh models for creating an appropriate color appearance model for each object mesh.

The overall process of the proposed method is illustrated in Fig. 1. In the proposed method, objects in the scene are initially extracted from each modal measurement. Specifically, object instance segments $\{\mathcal{S}_o^{c_i}\}$ are obtained from multiple images $\mathcal{I}^{c_i}$, while point clouds $\{\mathcal{V}_o^L\}$ each of which represents an arbitrary-shaped object are segmented from the LiDAR point cloud $\mathcal{V}^L$. The 3D mesh model $\mathcal{M}_o$ is reconstructed using each object point cloud $\mathcal{V}_o^L$. In order to associate each mesh with a corresponding image object segment properly, we generate an inaccurate yet simple point cloud by using the structure-from-motion (SfM) technique with multiple images, which is up-to-scale. Initial calibration is obtained by registering the LiDAR point cloud and the up-to-scale SfM-based point cloud, which provides the initial pose of each object, and thus, links between instance segments in the image and point cloud segments. Based on the correspondence information, the color appearance model for each object can be determined effectively. Using the initial pose, mesh, and color appearance models, iterative region-based object pose estimation is performed to determine the extrinsic parameters of multiple sensors.

Our main contributions are three folds, which are summarized as follows:

- We develop a targetless multiple camera-LiDAR calibration method that estimates the extrinsic parameters successfully using object pose estimation. To the best of our knowledge, the proposed method is the first application of object pose estimation, usually used for object tracking, to the extrinsic calibration of multimodal sensors. It is also noteworthy that the proposed method is the first automatic targetless calibration method in single-scan configuration.
- We propose a simple yet effective method that provides an initial calibration parameters. Based on the initial calibration parameters, corresponding objects obtained from multimodal measurements are matched. Consequently, the color appearance model corresponding to the object mesh model can be created appropriately.
- We address how to apply the correspondences between partially available object models to region-based object pose estimation for object tracking.

II. Proposed Method

The proposed method consists of *preprocessing*, *initial calibration*, and *calibration refinement* as shown in Fig. 1.

Let $\mathcal{I}^{c_i} : \Omega \rightarrow R^3, i = 1 \dots N_c$ denote N_c undistorted images of multiple cameras, where $\Omega \subset R^2$ indicates the image domain. In the preprocessing step, N_o object instance segments, $\mathcal{S}_o^{c_i}, o = 1 \dots N_o$, are extracted from $\mathcal{I}^{c_i}$, while point clouds $\mathcal{V}_o^L, o = 1 \dots N_o$, each of which represents an object, are segmented from the LiDAR point cloud $\mathcal{V}^L$. We assume that the camera intrinsic parameters $K_i, i = 1 \dots N_c$, are predetermined. By corresponding multimodal object segments, an initial pose meaning the relative pose of the cameras with respect to the LiDAR is obtained in the initial calibration step. In the calibration refinement step, the 3D mesh model and the appearance models are generated and utilized in object pose estimation to determine the final pose.

A. Preprocessing

The sparse LiDAR point cloud is preprocessed by removing large planes such as walls, the ceiling, and the floor. The remaining points are simply segmented into object point clouds $\mathcal{V}_o^L$ based on Euclidean distance.

Even cameras of the same model have not only their own intrinsic parameters but also unique color characteristics. The

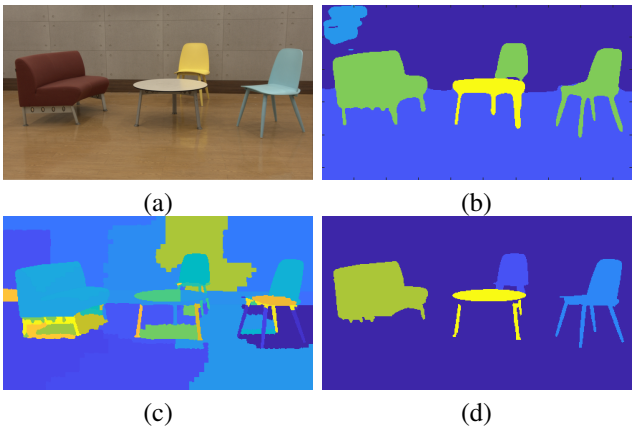


Fig. 2. An example of the results of the proposed object instance segmentation method. (a) An input image. (b) The semantic segmentation. (c) The spectral clustering-based segmentation. (d) The semantic cluster merging.

color inconsistency prevents correct image feature matching. In order to improve the following steps effectively, $\mathcal{I}^{c_i}$ are corrected for color consistency by mapping between their color distributions.

1) Object Instance Segmentation: In order to estimate precise color distributions for the appearance models, contours of objects in the scene should be identified correctly. Recently, various networks have been developed for semantic segmentation [27]–[29]. Although the networks extract object regions with their class information, boundaries of the regions are not aligned accurately to the object boundaries due to the pooling and striding operations in the networks. To overcome this, we combine a semantic segmentation network [28] and a spectral clustering (SC)-based segmentation technique. Well-aligned yet over-segmented clusters obtained by the SC-based segmentation are grouped further based on the semantic object information provided by the semantic network.

In order to capture high-level semantic features and produce segments roughly without small noisy segments, the input image of the i-th camera, $\mathcal{I}^{c_i}$, is segmented by using RefineNet [28] only with the highest level of the image feature pyramid.

In addition, $\mathcal{I}^{c_i}$ is over-segmented by using the accelerated superpixel approach [30]–[32]. A superpixel can be represented as a vector of the average color and position, $[l, a, b, x, y]^\top$. For each superpixel k, five neighboring superpixels l are searched, which are closest to the superpixel in terms of the distance based on the representatives

$$D_{k,l} = \sqrt{D^C_{k,l} + \gamma D^S_{k,l}}, \tag{1}$$

where γ controls the weight of positional difference, and

$$D^C_{k,l} = (l_k - l_l)^2 + (a_k - a_l)^2 + (b_k - b_l)^2, \tag{2}$$

$$D^S_{k,l} = (x_k - x_l)^2 + (y_k - y_l)^2. \tag{3}$$

A similarity matrix is constructed for spectral clustering by measuring similarity of a superpixel to its neighboring superpixels

$$S_{k,l} = \begin{cases} \exp\left(\frac{-D^2_{k,l}}{2\sigma_k\sigma_l}\right), & \text{if } k \text{ and } l \text{ are neighbors,} \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where σ_k indicates the average distance of superpixel k to its neighbors. Spectral clustering is performed with the similarity matrix to merge similar superpixels into a cluster whose boundaries are well aligned to the object. Clusters are merged to yield object segment $\mathcal{S}^{c_i}_o$ by labeling each cluster to the most overlapping semantic segment. Clusters overlapping wall or floor segments are removed.

Fig. 2 illustrates an example of results of the proposed image semantic segmentation method. The semantic segmentation extracts objects roughly, while the spectral clustering-based segmentation yields clusters well aligned to object boundary. By merging clusters overlapping an identical semantic segment, main objects in the scene are well segmented with semantic information.

2) SfM-based 3D Reconstruction: Using the SfM technique with undistorted images [33], a 3D point cloud is produced up to scale. The color correction for color consistency in the multiple images increases the number of inliers significantly. Although this point cloud is usually not sophisticated when the number of cameras, N_c, is small and the baseline between cameras is wide, it can provide rough relations about camera poses. The point cloud can be segmented into object point clouds $\mathcal{V}^c_o$ containing points whose reprojection is within $\mathcal{S}^{c_i}_o$.

B. Initial calibration

In order to determine initial relative poses of the cameras with respect to the LiDAR, two sets of object point clouds, i.e. $\{\mathcal{V}^L_m\}$ and $\{\mathcal{V}^c_o\}$, are registered using an iterative closest point (ICP) algorithm [34]. It is notable that $\{\mathcal{V}^c_o\}$ was obtained up-to-scale, and we use different subscripts for $\{\mathcal{V}^L_m\}$ and $\{\mathcal{V}^c_o\}$, which represent object indexes, since they are not associated with each other yet.

The SfM technique tends to incorrectly reconstruct points within textureless regions. Since such noisy points prevent the following registration process from performing successfully, noisy points in $\{\mathcal{V}^c_o\}$ are initially removed. In contrast to points representing the surface of an object, noisy points are placed very sparsely especially in terms of distance.

Therefore, if a point has less than k neighboring points within a radius τ, it is removed from $\{\mathcal{V}^c_o\}$. In our experiments, τ is set to one tenth of the average side length of the bounding box for $\mathcal{V}^c_o$, and k is set to $0.01 \cdot \min_o |\mathcal{V}^c_o|$, where $|\mathcal{V}^c_o|$ denotes the number of points in $\mathcal{V}^c_o$.

Letting $\mathbf{c}(\cdot)$ be an operation to obtain a centroid, we compare the average distance between all pairs of object point cloud centroids $\mathbf{c}\left(\mathcal{V}^c_o\right)$ with the average distance between all pairs of $\mathbf{c}\left(\mathcal{V}^L_m\right)$ in order to determine the scale of the point clouds reconstructed by the SfM technique.

After scaling $\{\mathcal{V}^c_o\}$, $\{\mathcal{V}^c_o\}$ is registered to $\{\mathcal{V}^L_m\}$ using ICP algorithm [34] to determine initial poses of cameras, $\xi^{c_i}_0$, with respect to the LiDAR. Through the registration, we can find correspondences between $\{\mathcal{V}^c_o\}$ and $\{\mathcal{V}^L_m\}$. Furthermore,

the object segment $\mathcal{S}_o^{c_i}$ can be associated with $\mathcal{V}_m^L$, in case that the projection of $\exp(\hat{\xi}_0^{c_i})\mathbf{c}\left(\mathcal{V}_m^L\right)$ onto $\mathcal{I}^{c_i}$ is closest to $\mathbf{c}\left(\mathcal{S}_o^{c_i}\right)$.

C. Calibration refinement

The initial poses $\xi_0^{c_i}$ are gradually refined by the rigid object pose estimation framework. In order to adapt the framework to the extrinsic calibration problem, we first produce appropriate 3D mesh and appearance models of objects in the scene.

1) 3D Mesh Model Generation: For each unorganized 3D point cloud $\mathcal{V}_o^L$, a manifold triangular mesh model $\mathcal{M}_o$ consisting of vertices $\mathbf{X}_n := (X_n, Y_n, Z_n)^\top \in R^3, n = 1 \dots N$, is generated using the surface reconstruction methods [35]–[37]. For a point $\mathbf{X}_n$, a weighted least squares plane is estimated with the $\mathbf{X}_n$'s k-neighborhood, and the k-neighboring points are projected onto the plane. The normal of the plane is used as an estimate of the true surface normal. Triangulation with the projected points produces triangular faces.

Small triangles are pruned to alleviate inconsistency in triangulation due to noisy points. Since the LiDAR point cloud is typically sparse, small holes may occur. In order to fill the hole, a tangent plane is estimated with the boundary points of the hole [37]. Then, resampling positions determined on the tangent plane are transformed to the 3D coordinates assuming that the surface is smooth.

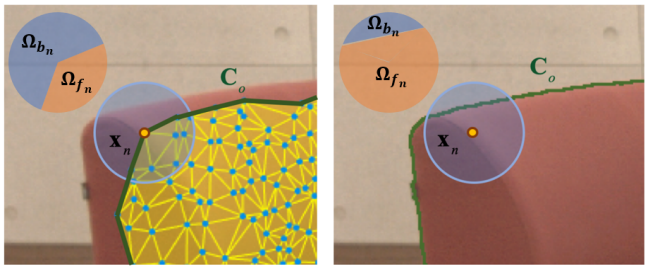
2) Local Appearance Model Generation: For each object o, a synthetic silhouette projection mask $\mathcal{I}_o^s : \Omega \rightarrow \{0, 1\}$ can be rasterized by rendering $\mathcal{M}_o$. The contour of the mask $\mathbf{C}_o$ splits the image into a foreground region $\Omega_f \subset \Omega$ and a background region $\Omega_b = \Omega \backslash \Omega_f$. By defining a circular image region Ω_n, centered at $\mathbf{x}_n \in \mathbf{C}_o$, two local color histogram models M_{f_n} and M_{b_n} are obtained as the appearance models for pixel-wise posterior estimation from $\Omega_{f_n} = \Omega_f \cap \Omega_n$ and $\Omega_{b_n} = \Omega_b \cap \Omega_n$, respectively.

In [26], the contour is precisely determined by manually placing the mesh model elaborately, and consequently, the appearance models are also accurately estimated. In our framework, however, the mesh model can be placed using the initial pose of a little less accuracy obtained in the initial calibration step. The contour of the silhouette mask will not be well aligned to the actual boundary of the object for the mesh model as shown in Fig. 3(a) where both Ω_{f_n} and Ω_{b_n} are defined over the red couch. Such misalignment causes to mix color distributions for actual foreground and background regions, and the consequent histograms become useless for distinguishing the couch and the background.

In order to deal with this problem, we exploit the object segment $\mathcal{S}_o^{c_i}$ associated with $\mathcal{V}_o^L$ for defining local foreground and background regions, Ω_{f_n} and Ω_{b_n}, instead of the silhouette mask.

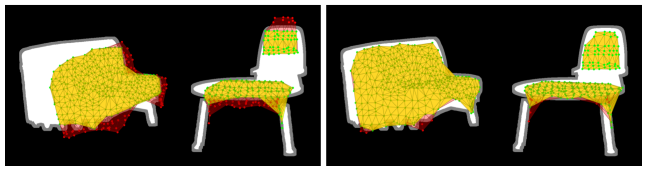
As the camera pose is improved through iterations, the projected point $\mathbf{x}_n$ becomes closer to the actual object boundary, and the appearance models also improve.

3) Object Pose Estimation: Although the mesh model $\mathcal{M}_o$ and the appearance model M_{f_n} and M_{b_n} are determined, it is possible that $\mathcal{M}_o$ is inconsistent with $\mathcal{S}_o^{c_i}$. For example, $\mathcal{S}_o^{c_i}$ may indicate a part of actual object because of either segmentation error or occlusion, while $\mathcal{M}_o$ contains parts outside $\mathcal{S}_o^{c_i}$. An error in the initial pose can cause the projection to deviate much from the correct position. In these cases, the optimization for object pose estimation would fail even with perfect $\mathcal{M}_o$.



(a) (b)

Fig. 3. Contour modification for accurate appearance model estimation. (a) The misaligned contour of the silhouette projection defines wrong foreground and background regions. (b) The contour of the object segment associated with the mesh model.



(a) (b)

Fig. 4. Mesh pruning for improving consistency between $\mathcal{M}_o$ to $\mathcal{S}_o^{c_i}$. (a) The 1st iteration. (b) The 3rd iteration.

To overcome this problem, we fit $\mathcal{M}_o$ to $\mathcal{S}_o^{c_i}$ at every iteration by pruning $\mathcal{M}_o$ as shown in Fig. 4. Initially, $\mathcal{S}_o^{c_i}$ in white is expanded to the gray region by using the morphological dilation operation in order to include external points near the object boundaries. Points outside the expanded segment, indicated in red, are excluded from $\mathcal{M}_o$ at current iteration. As the camera pose is improved through iterations, the excluded points are incrementally included again as shown in Fig. 4(b).

The iterative pose optimization is performed hierarchically within a three level image pyramid generated with a downscale factor of 2 in order to speed up and to find a global minimum solution. The optimization is performed once on the coarsest level, followed by one iteration on the second and finally six iterations on the first level i.e. the original full image resolution.

III. Experimental Result

Our proposed method was first tested on synthetic data that were generated as shown in Fig. 5 to evaluate its performance easily for various different configurations. Then, the proposed method was compared to conventional methods subjectively with real data.

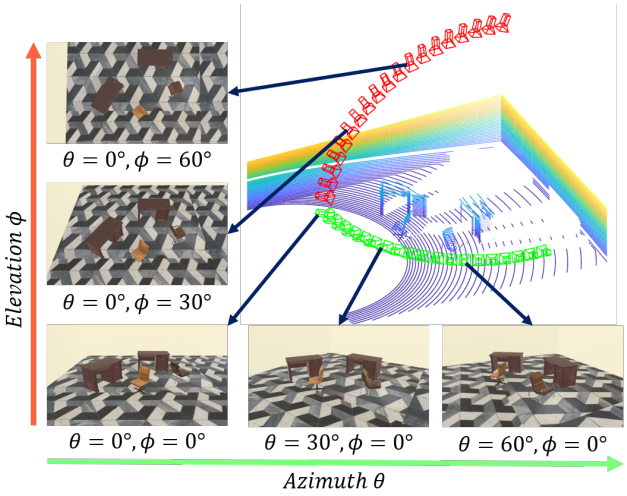


Fig. 5. Configuration for synthetic data generation. A room with office furniture was captured using a LiDAR and three cameras. The LiDAR sensor was fixed and the cameras moved along the elevation and azimuth direction with the same distance to the scene center to generate datasets 1 and 2 for experiments, respectively.

A. Testing on synthetic data

Three cameras were placed about 50 cm to 1 m apart from each other to view main objects. Each dataset has three 1920×1080 undistorted images and the LiDAR point cloud of about 300k points. The synthetic LiDAR data were created by imitating HDL-64E sensor, and the proposed algorithm was evaluated changing the LiDAR sensor noise and the number of objects in the scene. As shown in Fig. 5, the LiDAR sensor was fixed and the cameras moved from 0 to 90 degrees along the elevation and azimuth direction with the same distance to the scene center.

The translation error of the extrinsic parameters obtained was represented in Euclidean distance, and the rotation error was calculated as in [38]

$$\alpha = cos^{-1}((trace(R_{GT}^{T} R) - 1)/2), \tag{5}$$

where R_{GT} is the rotation matrix of the ground truth obtained from simulator and R is the rotation matrix obtained by the proposed method.

1) LiDAR Sensor Noise: LiDAR sensors have sensing errors with different reflectance and distances. We added Gaussian noises with different standard deviations from 0 to 8 cm to the virtually generated LiDAR data as shown in Fig. 6. The standard deviation of the LiDAR sensing noises is typically about 2 cm.

Fig. 7(a) plots the calibration results of the proposed method for datasets 1 and 2 having four objects in the scene.

The calibration accuracy is affected by the amount of the added noises, but not closely correlated. The graphs show that the proposed method deals with azimuthal variations much better than elevation changes. That is because the semantic segmentation network usually trained with images of normal viewpoints yields poor segmentation results for high elevated, top-view images. In controlled environments

Fig. 6. Noisy point cloud for the orange colored chair in Fig. 5.

of any azimuthal rotation or within 20° in elevation, the noise amount does not affect the calibration accuracy.

2) The Number of Objects: Since the proposed method exploits the object pose estimation framework, it is expected that the performance of the method depends on the number of objects. To confirm this, the experiment was conducted by changing the number of objects from two to four with LiDAR noises of $\sigma = 0.02$.

As shown in Fig. 7(b), rotational and translational errors reduce as increasing the number of objects. The proposed method yielded erroneous results when using only two objects, since two objects are not satisfied with the 3 degrees of freedom of rotation as in the PnP algorithm. If there are three objects, they satisfy the three degrees of freedom of rotation and reduce calibration errors significantly.

B. Testing on real data

In order to demonstrate the performance of the proposed method, the proposed method was compared with two target-based methods [5], [6] and one targetless method using line features [12], which are referred to as "Target 1", "Target 2", and "Targetless". We implemented all the methods including the proposed method in MATLAB. Although a pre-trained CalibNet [18] was also tested as a deep learning-based calibration method, it produced very unsatisfactory results for our indoor datasets, as compared with results for KITTI datasets, due to quite different characteristics of depth maps generated in the network.

Four datasets of indoor environments including several objects were acquired in single-scan configuration. Each scene contains rectangular cardboards and a box for the methods "Target 1" and "Target 2", respectively. We manually marked the corner points of the cardboards in the image and segmented boundary points of the cardboards in the point cloud for "Target 1". Points of the box were also manually segmented for "Target 2". For "Targetless", the 2D-3D line correspondences were manually matched also. The expansion size for mesh pruning was set to 6, 12, and 24 pixels for the 3rd, 2nd, and 1st levels, respectively.

Fig. 9 shows the projection of the LiDAR point cloud with determined ξ_0 and ξ. Although the initial calibration was not accurate, the relation among multimodal sensors was roughly obtained. The final calibration achieved satisfactory results, and shows that our proposed framework performs effectively.

Fig. 8 illustrates the calibration results of the various extrinsic calibration methods by projecting the LiDAR point cloud on the images for subjective comparison. The two

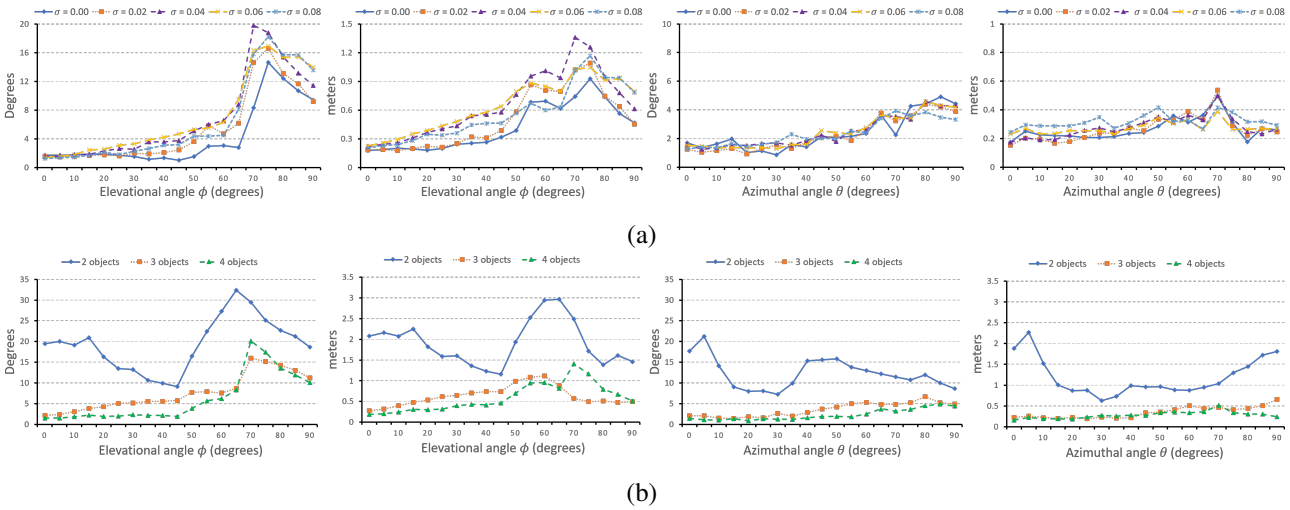


Fig. 7. Extrinsic calibration results of the proposed method for various configurations of the synthetic datasets 1 and 2. (a) Results obtained varying the sensing noise of the LiDAR. (b) Results obtained varying the number of objects. The first and second columns represent the rotation and translation error for the dataset 1, respectively, while the third and fourth columns for the dataset 2.

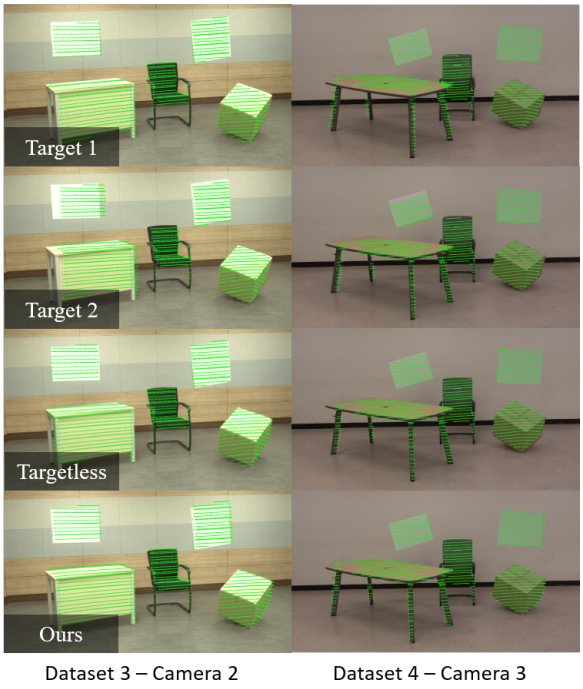


Fig. 8. Subjective comparison of multimodal sensor extrinsic calibration methods by projecting the LiDAR point cloud on the images. The first row: The target-based method using cardboard corners [5]. The second row: The target-based method using a box [6]. The third row: The targetless method using line features [12]. The fourth row: The proposed method.

target-based methods and the targetless method aligns well near the target objects or geometric features, while the misalignment tends to increase for objects farther away from the targets. Specifically, in the result of "Target 1" using the corner points of rectangular cardboards, the projection of the box was deviated severely. In the result of "Target 2" using the box corner points, the farthest cardboard were plotted far away.

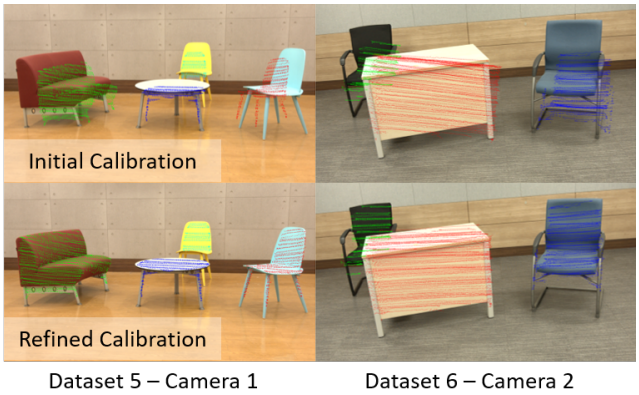


Fig. 9. The initial calibration is refined by iteratively estimating the object poses with partially reconstructed 3D models and color appearance models of the objects.

Since the rectangular cardboards were detected as wall by the image segmentation network in the proposed method, three objects were extracted as object from Dataset 3 and Dataset 4, respectively. The proposed method produces satisfactory results that are well aligned overall as shown in the last row of Fig. 8. This is because the proposed method exploits objects placed at various locations in the scene.

IV. Conclusions

We have presented the extrinsic calibration method for multiple camera-LiDAR sensors in single-scan configuration that is a novel approach to targetless extrinsic calibration in that any objects of unspecified shapes can be utilized. Even though the 3D meshes for objects are partially generated from the LiDAR point cloud and the initial pose may be inaccurate, our proposed method successfully estimates appearance models by exploiting object segments.

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)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[2] K. Kwak, D. F. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning LiDAR and a camera," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 3283–3289.

[3] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 3936–3943.

[4] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3D LiDAR instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[5] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "LiDAR-camera calibration using 3D-3D point correspondences," *arXiv preprint arXiv:1705.09785*, 2017.

[6] Z. Pusztai, I. Eichhardt, and L. Hajder, "Accurate calibration of multi-LiDAR-multi-camera systems," *Sensors*, vol. 18, no. 7, p. 2139, 2018.

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

[8] E.-s. Kim and S.-Y. Park, "Extrinsic calibration between camera and LiDAR sensors by matching multiple 3D planes," *Sensors*, vol. 20, no. 1, p. 52, 2020.

[9] X. Gong, Y. Lin, and J. Liu, "3D LiDAR-camera extrinsic calibration using an arbitrary trihedron," *Sensors*, vol. 13, no. 2, pp. 1902–1918, 2013.

[10] R. Frohlich, L. Tamas, and Z. Kato, "Absolute pose estimation of central cameras using planar regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[11] J. Kang and N. L. Doh, "Automatic targetless camera-LiDAR calibration by aligning edge with gaussian mixture model," *Journal of Field Robotics*, vol. 37, no. 1, pp. 158–179, 2020.

[12] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 3685–3691.

[13] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and LiDAR by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[14] Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and LiDAR in natural environments," in *Australian Conference on Robotics and Automation (ACRA)*, 2012, pp. 3–5.

[15] Z. Taylor, J. Nieto, and D. Johnson, "Multi-modal sensor calibration using a gradient orientation measure," *Journal of Field Robotics*, vol. 32, no. 5, pp. 675–695, 2015.

[16] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "Regnet: Multimodal sensor registration using deep neural networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1803–1810.

[17] H. Liu, Y. Liu, X. Gu, Y. Wu, F. Qu, and L. Huang, "A deep-learning based multi-modality sensor calibration method for usv," in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2018, pp. 1–5.

[18] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "Calibnet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1110–1117.

[19] M. Rofail, A. Alsafty, M. Matousek, and F. Kargl, "Multi-modal deep learning for vehicle sensor data abstraction and attack detection," in *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2019, pp. 1–7.

[20] B. Rosenhahn, T. Brox, and J. Weickert, "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 243–262, 2007.

[21] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 831–844.

[22] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers, "Combined region and motion-based 3D tracking of rigid and articulated objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 3, pp. 402–415, 2009.

[23] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, "A geometric approach to joint 2D region-based segmentation and 3D pose estimation using a 3D shape prior," *SIAM Journal on Imaging Sciences*, vol. 3, no. 1, pp. 110–132, 2010.

[24] C. Schmaltz, B. Rosenhahn, T. Brox, and J. Weickert, "Region-based pose tracking with occlusions using 3D models," *Machine Vision and Applications*, vol. 23, no. 3, pp. 557–577, 2012.

[25] V. A. Prisacariu and I. D. Reid, "PWP3D: Real-time segmentation and tracking of 3D objects," *International Journal of Computer Vision*, vol. 98, no. 3, pp. 335–354, 2012.

[26] H. Tjaden, U. Schwanecke, and E. Schomer, "Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 124–132.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[28] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1925–1934.

[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

[30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[31] K.-S. Choi and K.-W. Oh, "Subsampling-based acceleration of simple linear iterative clustering for superpixel segmentation," *Computer Vision and Image Understanding*, vol. 146, pp. 1–8, 2016.

[32] K.-W. Oh and K.-S. Choi, "Acceleration of simple linear iterative clustering using early candidate cluster exclusion," *Journal of Real-Time Image Processing*, vol. 16, no. 4, pp. 945–956, 2019.

[33] A. M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, 2001.

[34] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[35] Z. C. Marton, R. B. Rusu, and M. Beetz, "On fast surface reconstruction methods for large and noisy point clouds," in *2009 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 3218–3223.

[36] M. Gopi and S. Krishnan, "A fast and efficient projection-based approach for surface reconstruction," in *Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2002, pp. 179–186.

[37] J. Wang and M. M. Oliveira, "A hole-filling strategy for reconstruction of smooth surfaces in range images," in *16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003)*. IEEE, 2003, pp. 11–18.

[38] D. Eberly, "Rotation representations and performance issue," 2001.