A Joint Optimization Approach of LiDAR-Camera Fusion for Accurate Dense 3-D Reconstructions

Weikun Zhen, Yaoyu Hu, Jingfeng Liu, and Sebastian Scherer

***Abstract*—Fusing data from LiDAR and camera is conceptually attractive because of their complementary properties. For instance, camera images are of higher resolution and have colors, while LiDAR data provide more accurate range measurements and have a wider field of view. However, the sensor fusion problem remains challenging since it is difficult to find reliable correlations between data of very different characteristics (geometry versus texture, sparse versus dense). This letter proposes an offline LiDAR-camera fusion method to build dense, accurate 3-D models. Specifically, our method jointly solves a bundle adjustment problem and a cloud registration problem to compute camera poses and the sensor extrinsic calibration. In experiments, we show that our method can achieve an average accuracy of 2.7 mm and resolution of 70 points/cm^2 by comparing to the ground truth data from a survey scanner. Furthermore, the extrinsic calibration result is discussed and shown to outperform the state-of-the-art method.**

***Index Terms*—Sensor Fusion, Mapping, Calibration and Identification.**

I. Introduction

THIS work is aimed at building accurate dense 3D models by fusing multiple frames of LiDAR and camera data as shown in Fig. 1. The LiDAR scans 3D points on the surface of an object and the acquired data are accurate in range and robust to low-texture conditions. However, the LiDAR data contain limited information of texture (only intensities) and are quite sparse due to the physical spacing between internal lasers. Differently, a camera provides denser texture data but does not measure distances directly. Although a stereo system measures the depth through triangulation, it may fail in regions of low-texture or repeated patterns. Those complementary properties make it very attractive to fuse LiDAR and cameras for building dense textured 3D models.

The majority of proposed sensor fusion algorithms typically augment the image with LiDAR depth. Then the sparse depth image may be upsampled to get a dense estimation, or used to facilitate the stereo triangulation process. However, we observe two drawbacks of these strategies. The first one is that the depth augmentation requires sensor extrinsic calibration, which, compared to the calibration of stereo cameras, is less accurate since matching structural and textural features can be unreliable. For example (see Fig. 2), many extrinsic calibration approaches use edges of a target as the correspondences between point clouds and images, which will have issues: 1) cloud edges due to occlusion are not clean but *mixed*, and 2) edge points are not on the real edge due to data sparsity but only *loosely* scattered. The second drawback is that the upsampling or LiDAR-guided stereo triangulation techniques are based on the local smoothness assumption, which becomes invalid if the original depth is too sparse. The accuracy of fused depth map is hence decreased, which may still be useful for obstacle avoidance, but not ideal for the purpose of mapping. For the reasons discussed above, we choose to combine a rotating LiDAR with a wide-baseline,

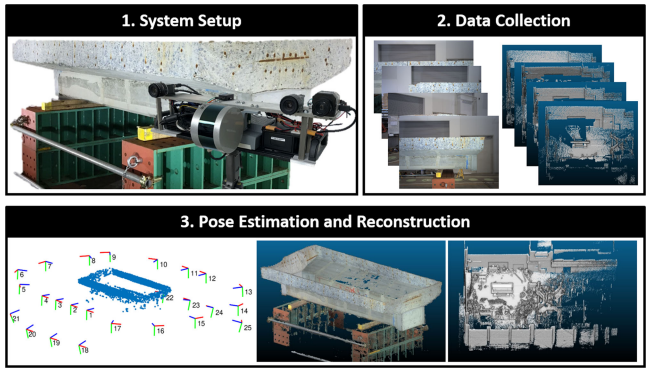

Fig. 1. A customized LiDAR-stereo system is used to collect stereo images (only left images are visualized) and LiDAR point clouds. Our algorithm estimates the camera poses, generates a textured dense 3D model of the scanned specimen and a point cloud map of the environment.

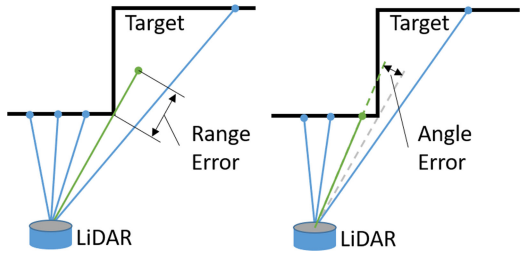

Fig. 2. An illustration of inaccurate edge extraction. *Left:* The *mixed* edge point (green) has range error. *Right:* The *loose* edge point (green) has angular error.

Manuscript received February 24, 2019; accepted June 24, 2019. Date of publication July 12, 2019; date of current version July 24, 2019. This letter was recommended for publication by Associate Editor T. Peynot and Editor E. Marchand upon evaluation of the reviewers' comments. This work was supported by the Shimizu Institute of Technology, Japan. *(Corresponding author: Weikun Zhen.)*

W. Zhen and J. Liu are with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: weikunz@andrew.cmu.edu; jingfenl@andrew.cmu.edu).

Y. Hu and S. Scherer are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: yaoyuh@andrew.cmu.edu; basti@andrew.cmu.edu).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2019.2928261

high-resolution stereo system to increase the density of raw data. Moreover, we aim to fuse multiple sensor data and recover the extrinsic calibration simultaneously.

The main contribution of this letter is an offline method to process multiple frames of stereo and point cloud data and jointly optimizes the camera poses and the sensor extrinsic transform. The proposed method has benefits that:

- it does not rely on unreliable correlations between structural and textural data, but only enforces the geometric constraints between sensors, which frees us from handcrafting heuristics to associate information from different domains.
- it joins the bundle adjustment and cloud registration problem in a probabilistic framework, which enables proper treatment of sensor uncertainties.
- it is capable of performing accurate self-calibration, making it practically appealing.

The rest of this letter is organized as follows: Section II presents the related work on LiDAR-camera fusion techniques. Section III describes the proposed method in detail. Experimental results are shown in Section IV. Conclusions and future work are discussed in Section V.

II. Related Work

In this section, we briefly summarize the related work in the areas of LiDAR-camera extrinsic calibration and fusion. For extrinsic calibration, the proposed methods can be roughly categorized according to the usage of a target. For example, a single [1] or multiple [2] chessboards can be used as planar features to be matched between the images and point clouds. Besides, people also use specialized targets, such as a box [3], a board with shaped holes [4] or a trihedron [5], where the extracted features also include corners and edges. The usage of a target simplifies the problem but is inconvenient when a target is not available. Therefore target-free methods are developed using natural features (e.g. edges) which are usually rich in the environment. For example, Levinson and Thrun [6] make use of the discontinuities of LiDAR and camera data, and refine the initial guess through a sampling-based method. This method is successfully applied on a self-driving car to track the calibration drift. Pandey *et al.* [7] develop a Mutual Information (MI) based framework that considers the discontinuities of LiDAR intensities. However, the performance of this method is dependent on the quality of intensity data, which might be poor without calibration for cheap LiDAR models. Differently, [8]–[10] recover the extrinsic transform based on the ego-motion of individual sensors. These methods are closely related to the well-known hand-eye calibration problem [11] and do not rely on feature matching. However, the motion estimation and extrinsic calibration are solved separately and the sensor uncertainties are not considered. Instead, we construct a cost function that joins the two problems in a probabilistically consistent way and optimizes all parameters together.

Available fusion algorithms are mostly designed for LiDAR-monocular or LiDAR-stereo systems and assume the extrinsic transform is known. For a LiDAR-monocular system, images are often augmented with the projected LiDAR depth. The fused data can then be used for multiple tasks. For example, Dolson *et al.* [12] upsample the range data for the purpose of safe navigation in dynamic environments. Bok *et al.* [13] and Vechersky *et al.* [14] colorize the range data using camera textures. Zhang and Singh [15] show significant improvement on the robustness and accuracy of the visual odometry if enhanced with depth. For LiDAR-stereo systems [16]–[19], LiDAR is typically used to guide the stereo matching algorithms since a depth prior could significantly reduce the disparity searching range and help to reject outliers. For instance, Miksik *et al.* [17] interpolate between LiDAR points to get a depth prior before stereo matching. Maddern and Newman [18] propose a probabilistic framework that encodes the LiDAR depth as prior knowledge and achieves real-time performance. Additionally, in the area of surveying [20]–[22], point clouds are registered based on the motion estimated using cameras. Our method differs from these work in that LiDAR points are not projected on the image since the extrinsic transform is assumed unknown. Instead, we use LiDAR data to refine the stereo reconstruction after the calibration is recovered.

III. Joint Estimation and Mapping

A. Overview

Before introducing the proposed algorithm pipeline, we clarify the definitions used throughout the rest of this letter. In terms of symbols, we use bold lower-case letters (e.g. $\mathbf{x}$) to represent vectors or tuples, and bold upper-case letters (e.g. $\mathbf{T}$) for matrices, images or maps. Additionally, calligraphic symbols are used to represent sets (e.g. $\mathcal{T}$ stands for a set of transformations). And scalars are denoted as light letters (e.g. i, N).

As basic concepts, an *image landmark* $\mathbf{l} \in \mathbb{R}^3$ is defined as a 3D point that is observed in at least two images. Then a *camera observation* is represented by a 5-tuple $\mathbf{o}_c = \{i, k, \mathbf{u}, d, w\}$, where the elements are the camera id, the landmark id, image coordinates, the depth and a weight factor of the landmark, respectively. In addition, a *LiDAR observation* is defined as a 6-tuple $\mathbf{o}_l = \{i, j, \mathbf{p}, \mathbf{q}, \mathbf{n}, w\}$ that contains the target cloud id i, the source cloud id j, a key point in the source cloud, its nearest neighbor in the target cloud, the neighbor's normal vector and a weight factor. In other words, one LiDAR observation associates a 3D point to a local plane and the point-to-plane distance will be minimized in the later joint optimization step.

The complete pipeline of proposed method is shown in Fig. 3. Given the stereo images and LiDAR point clouds, we first extract and match features to prepare three sets of observations, namely the landmark set $\mathcal{L}$, the camera observation set $\mathcal{O}_c$ and the LiDAR observation set $\mathcal{O}_l$. The observations are then fed to the joint optimization block to estimate optimal camera poses $\mathcal{T}_c^*$ and sensor extrinsic transform $\mathbf{T}_e^*$. Based on the latest estimation, the LiDAR observations are recomputed and the optimization is repeated. After a number of iterations, the parameters converge to local optima. Finally, the refinement and mapping block joins the depth information from stereo images and LiDAR clouds to produce the 3D model. In the rest of this section, each component is described in detail individually.

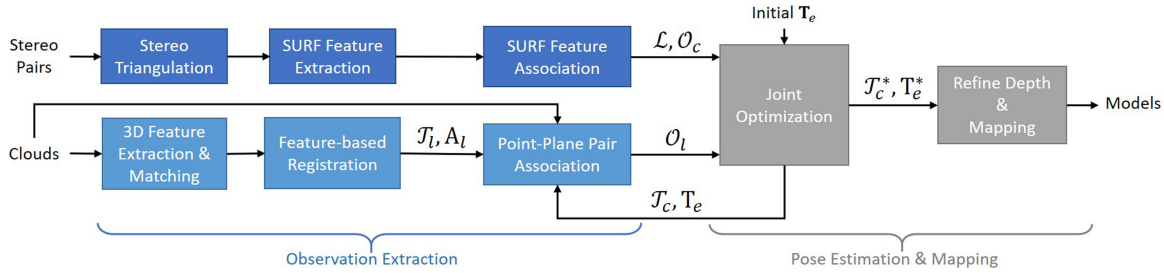

Fig. 3. A diagram of the proposed pipeline. In the observation extraction phase (front-end), SURF features are extracted and matched across all datasets to build the landmark set $\mathcal{L}$ and the camera observations $\mathcal{O}_c$. On the other hand, point clouds are abstracted with BSC features, and roughly registered to find cloud transforms $\mathcal{T}_l$. Then point-plane pairs are found to build the LiDAR observation set $\mathcal{O}_l$. In the pose estimation and mapping phase (back-end), we solve the BA problem and the cloud registration problem simultaneously. Here the $\mathcal{O}_l$ is recomputed after each convergence based on the latest estimation $\mathcal{T}_c, \mathbf{T}_e$ and the optimization is repeated for a few iterations. Finally, local stereo reconstructions are refined using LiDAR data and assembled to build the 3D model.

B. Camera Observation Extraction

Given a stereo image pair, we firstly perform stereo triangulation to obtain a disparity image using Semi-Global Matching (SGM) proposed in [23]. The disparity image is represented in the left camera frame. Then SURF [24] features are extracted from the left image. Note that our algorithm itself does not require a particular type of feature to work. After that, a feature point is associated with depth value if a valid disparity value is found within a small radius (2 pixels in our implementation). Only the key points with depth are retained for further computation. The steps above are repeated for all stations to acquire multiple sets of features with depth. Once the depth association is done, a global feature association block is used to find correlations between all possible combinations of images. We adopt a simple matching method that incrementally adds new observations and landmarks to $\mathcal{O}_c$ and $\mathcal{L}$. Algorithm 1 shows the detailed procedures. Basically, we iterate through all possible combinations to match image features based on the Euclidean distance of corresponding descriptors. $\mathcal{L}$ and $\mathcal{O}_c$ will be updated accordingly if a valid match is found.

Additionally, an adjacency matrix $\mathbf{A}_c$ encoding the correlation of the images can be obtained. Since the camera FOV is narrow, it is likely that the camera pose graph is not fully connected. Therefore, additional connections have to be added to the graph, which is one of the benefits of fusing point clouds.

C. LiDAR Observation Extraction

Although many 3D local surface descriptors have been proposed (a review is given in [25]), they are less stable and not accurate compared to image feature descriptors. In fact, it is preferable to use 3D descriptors for rough registration and refine the results using slower but more accurate methods such as Iterative Closest Point (ICP) [26]. Our work follows a similar idea. Specifically, the Binary Shape Context (BSC) descriptor [27] is used to match and roughly register point clouds to compute the cloud transforms $\mathcal{T}_l$. As a 3D surface descriptor, BSC encodes the point density and distance statistics on three orthogonal projection plane around a feature point. Furthermore, it represents the local geometry as a binary string which enables fast difference comparison on modern CPUs. Fig. 4-left shows

Algorithm 1: SURF Feature Association

```
Given feature sets F_{1:N} from N stations;
for i = 1 : N do
    for j = i + 1 : N do
        for f in F_i do
            find the best match g in F_j;
            if f and g NOT similar then
                continue;
            end
            if f, g both unlabeled then
                create new landmark id k ← |L|;
                label f, g with id k;
                add new landmark l with id k to L;
                add new observations o_f, o_g [1] to O_c;
            else if f labeled, g unlabeled then
                copy label from f to g;
                add new observation o_g to O_c;
            else if f unlabeled, g labeled then
                copy label from g to f;
                add new observation o_f to O_c;
            else
                continue;
            end
        end
    end
end
return O_c, L.
```

[1] $\mathbf{o_f}, \mathbf{o_g}$ are observation tuples filled with information from $\mathbf{f}$ and $\mathbf{g}$.

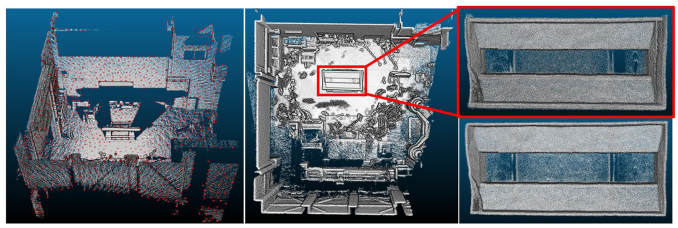

Fig. 4. *Left:* An example of extracted BSC features (red) from a point cloud (grey). *Middle:* Registered point cloud map based on matched features. *Right:* Comparison of rough registration (top-right) and refined registration (bottom-right) in a zoomed-in window.

an example of extracted BSC features. However, feature-based registration is of low accuracy. As shown in the right plots of Fig. 4, misalignment can be observed in the rough registered map. As a comparison, the refined map of higher accuracy obtained by our method is also visualized.

After the rough registration, another adjacency matrix $\mathbf{A}_l$ encoding matched cloud pairs is obtained. We use the merged adjacency matrix $\mathbf{A}_c \vee \mathbf{A}_l$ to define the final pose graph, where $\vee$ means element-wise or logic operation.

To obtain $\mathcal{O}_l$, a set of points are sampled randomly from each point cloud as the key points. Note that the key points to refine the registration are denser than the features. For each pair of connected clouds in $\mathbf{A}_l$, the one with a smaller index is defined as the target while the other one as the source. Then each key point in the source is associated with its nearest neighbor and a local normal vector in the target within a given distance threshold. Finally, all point matches are formatted as a LiDAR observation and stacked into $\mathcal{O}_l$.

D. Joint Optimization

Given the observations $\mathcal{O}_c$ and $\mathcal{O}_l$, we first formulate the observation likelihood as the product of two probabilities

$$P(\mathcal{O}_c, \mathcal{O}_l | \mathcal{T}, \mathcal{L}, \mathbf{T}_e) = P(\mathcal{O}_c | \mathcal{T}, \mathcal{L}) P(\mathcal{O}_l | \mathcal{T}, \mathbf{T}_e) \quad (1)$$

where $\mathcal{T} = \{\mathbf{T}_i | i = 1, 2, \ldots\}$ is the set of camera poses with $\mathbf{T}_1 = \mathbf{I}_4$, and $\mathbf{T}_e$ is the extrinsic transform. Assuming the observations are conditionally independent, we have

$$P(\mathcal{O}_c | \mathcal{T}, \mathcal{L}) = \prod_{\mathbf{o}_c \in \mathcal{O}_c} P(\mathbf{o}_c | \mathbf{T}_i, \mathbf{l}_k) \quad (2)$$

$$P(\mathcal{O}_l | \mathcal{T}, \mathbf{T}_e) = \prod_{\mathbf{o}_l \in \mathcal{O}_l} P(\mathbf{o}_l | \mathbf{T}_i, \mathbf{T}_j, \mathbf{T}_e) \quad (3)$$

where i, j are camera ids and k is the landmark id, which are specified by observation $\mathbf{o}_c$ or $\mathbf{o}_l$. The probability of one observation is approximated with a Gaussian distribution as

$$P(\mathbf{o}_c | \mathbf{T}_i, \mathbf{l}_k) \propto \exp\left(-\frac{1}{2} w_{\mathbf{o}_c} (E_f^2 + E_d^2)\right) \quad (4)$$

$$P(\mathbf{o}_l | \mathbf{T}_i, \mathbf{T}j, \mathbf{T}_e) \propto \exp\left(-\frac{1}{2} w_{\mathbf{o}_l} E_l^2\right) \quad (5)$$

where $w_{\mathbf{o}_c}, w_{\mathbf{o}_l}$ are the weighting factors of camera and LiDAR observations. And the residual E_f and E_d encode landmark reprojection and depth error, while E_l denotes the point-to-plane distance error. Those residuals are defined as

$$\text{feature: } E_f = \frac{||\phi(\mathbf{l}_k | \mathbf{K}, \mathbf{T}_i) - \mathbf{u}||}{\sigma_p} \quad (6)$$

$$\text{depth: } E_d = \frac{\|\psi(\mathbf{l}_k | \mathbf{T}_i)\| - d}{\sigma_d} \quad (7)$$

$$\text{laser: } E_l = \frac{\mathbf{n}^{\mathrm{T}} (\psi(\mathbf{p} | \mathbf{T}_{l,ij}) - \mathbf{q})}{\sigma_l} \quad (8)$$

Here, $\mathbf{u}$ and d are observed image coordinates and depth of landmark k. $\mathbf{T}_{l,ij} = (\mathbf{T}_e \mathbf{T}_i)^{-1} \mathbf{T}_j \mathbf{T}_e$ is the transform from target cloud i to source cloud j. Function $\phi(\cdot)$ projects a landmark onto the image i specified by input intrinsic matrix $\mathbf{K}$ and transform $\mathbf{T}_i$. Function $\psi(\cdot)$ transforms a 3D point using the input transformation. σ_p, σ_d and σ_l denote the measurement uncertainties of extracted features, stereo depths and LiDAR ranges, respectively.

Substituting (2)–(8) back into (1) and taking the negative log-likelihood gives the cost function

$$f(\mathcal{T}, \mathcal{L}, \mathbf{T}_e) = \frac{1}{2} \sum_{\mathbf{o}_c} w_{\mathbf{o}_c} \left(E_f^2 + E_d^2\right) + \frac{1}{2} \sum_{\mathbf{o}_l} w_{\mathbf{o}_l} E_l^2 \quad (9)$$

which is iteratively solved over parameters $\mathcal{T}, \mathcal{L}, \mathbf{T}_e$ using the Levenberg-Marquardt algorithm.

To filter out incorrect observations in both images and point clouds, we check the reprojection error $||\phi(\mathbf{l}_k | \mathbf{K}, \mathbf{T}_i) - \mathbf{u}||$ and depth error $\|\psi(\mathbf{l}_k | \mathbf{T}_i)\| - d$ of camera observations and check the distance error $\mathbf{n}^{\mathrm{T}} (\psi(\mathbf{p} | \mathbf{T}_{l,ij}) - \mathbf{q})$ of LiDAR observations after the optimization converges. The observations whose errors are larger than prespecified thresholds will be marked as outliers and assigned with zero weights. The cost function (9) is optimized repeatedly until no more outliers can be detected. The thresholds can be tuned by hand and in the experiments we use 3 pixels, 0.01m and 0.1m respectively.

Similar to the ICP algorithm, the $\mathcal{O}_l$ is recomputed based on the latest estimation of $\mathcal{T}_c, \mathbf{T}_e$, while the $\mathcal{O}_c$ remains unchanged. Once $\mathcal{O}_l$ is updated, the outlier detection and optimization steps are repeated as mentioned above. The $\mathcal{O}_l$ only needs to be recomputed a few times (4 times in our experiments) to achieve good accuracy.

Additionally, the strategy of specifying the uncertainty parameters is as follows. Based on the stereo configuration, the triangulation depth error e_d is related to the stereo matching error e_p by a scale factor as in $e_d = (d^2/bf)e_p$, where b is the baseline, f is the focal length and d is the depth. Assuming the uncertainties of feature matching and stereo matching are equivalent, we have $\sigma_d = (d^2/bf)\sigma_p$. Therefore, we can now set σ_p to be the identity (i.e. 1) and set σ_d by multiplying the scale factor. On the other hand, the value of σ_l is tuned by hand so that the total cost of camera and LiDAR observations are roughly at the same magnitude. In the experiments, setting $\sigma_p = 1$, $\sigma_d = 5 \times 10^3$ and σ_l between [0.02, 0.1] can generate sufficiently good results.

E. Mapping

With the camera poses estimated, building a final 3D model could be simply registering all stereo point clouds together. However, the stereo depth maps typically contain outliers and holes due to triangulation failure. In order to refine the stereo depth maps, we further perform a simple but effective two-fold fusion of LiDAR and camera data for each frame or station. In the first fold, the stereo depth is compared with the projected LiDAR depth and will be removed if there is a significant difference. In the second fold, LiDAR depth is selectively used to fill holes in the stereo depth. Particularly, we only use the regions that are locally flat (such that the local smoothness assumption is valid), and well observed (avoiding degenerated view angle). The curvature of the local surface is used to measure the flatness. And the normal vector is used to compute the view angle. Fig. 5 shows an example of refining the stereo point cloud. It

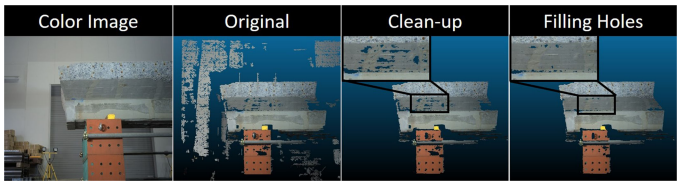

Fig. 5. An example of refining the stereo depth. The outliers are first filtered out by limiting its difference to the LiDAR depth within a maximum range threshold. Then the holes are filled with the surrounding LiDAR depth only if the local surface has a near-zero curvature.

can be observed that holes lying on a flat surface can be filled successfully, while the missing points close to the edges are not treated to avoid introducing new outliers.

F. Conditions of Uniqueness

The proposed approach relies on the ego-motion of individual sensors to recover the extrinsic transform $\mathbf{T}_e$, making it possible that $\mathbf{T}_e$ is not fully observable if the motion degenerates. It turns out to be the same problem encountered in hand-eye calibration, where the extrinsic transform between a gripper and a camera is estimated from two motion sequences. Here we discuss conditions for a fully observable $\mathbf{T}_e$ by borrowing knowledge from the hand-eye calibration, whose classical formulation is given by

$$\mathbf{T}_c\mathbf{T}_e = \mathbf{T}_e\mathbf{T}_h \tag{10}$$

where $\mathbf{T}_h, \mathbf{T}_c$ represent the relative motion of the hand and the camera w.r.t. their own original frames. Incorporating multiple stations will result in a set of (10) and then $\mathbf{T}_e$ can be solved. According to [11], the following two conditions must be satisfied to guarantee a unique solution of $\mathbf{T}_e$:

1) At least 2 motion pairs $(\mathbf{T}_c, \mathbf{T}_h)$ are observed. Equivalently, at least 3 stations are needed, with one of them to be the base station.
2) The rotation axes of $\mathbf{T}_c$ are not colinear for different motion pairs.

In our case, the robot hand frame is substituted by the LiDAR frame. Therefore, the configuration of each station must also satisfy the above conditions of uniqueness. This provides formal guidance to collect data effectively. From our experience of deploying the developed system, an operator without adequate background knowledge in computer vision, particularly in structure from motion, is likely to miss the second condition and only rotates the sensor about the vertical axis, which will make the extrinsic calibration unobservable.

IV. Experiments

A. The Sensor Pod

To collect data for experiments, we developed a sensor pod (as shown in Fig. 6) which has a pair of stereo cameras (global shutter, resolution 4112×3008, baseline 38 cm), a Velodyne Puck (VLP-16), an IMU and a thermal camera. This work only uses the stereo image pairs and LiDAR clouds for reconstruction. Particularly, the VLP-16 is mounted on a continuously rotating (180° per second) motor to increase the sensor FOV.

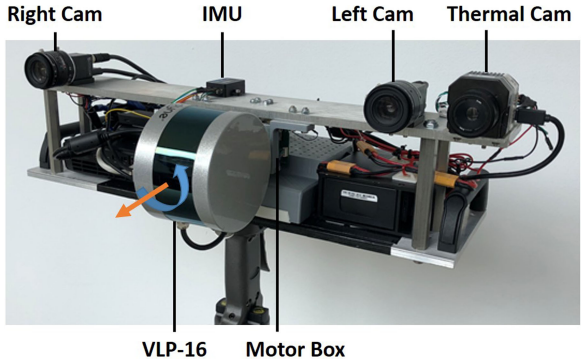

Fig. 6. The sensor pod developed for data collection.

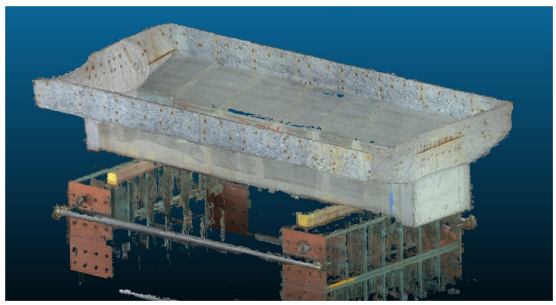

Fig. 7. Built point cloud model of the T-shaped specimen.

The calibration between the involved sensors are performed separately. We use the OpenCV library [28] to obtain camera intrinsic and extrinsic parameters. The transform between the motor and the LiDAR frame is obtained by placing the sensor pod in a conference room, and carefully tuning the transform until the accumulated points on walls and ceiling form thin surfaces in the fixed motor base frame. From now on, we use the term *LiDAR frame* to denote the fixed motor base frame instead of the actual rotating Velodyne frame, and assume all point clouds have been transformed into the LiDAR frame.

B. Reconstruction Tests

The first reconstruction test is carried out at the Shimizu Institute of Technology in Tokyo to scan a T-shaped concrete specimen that is under structural tests. In total, 25 stations of data are collected around the specimen at a distance of about 2.5 meters. Each station contains a stereo image pair, a point cloud that accumulates scans for 20 seconds and contains approximately 1.6 million points. For station 1–17, the sensor pod is placed on a tripod and pointed to the specimen. Station 18–25 are collected with the sensor pod on the ground, tilted up to capture the bottom of the specimen. Fig. 7 shows the reconstructed model and Fig. 8 visualizes the camera poses and landmarks. In the lower plots of Fig. 8, correlations found between images (blue lines) and point clouds (grey lines) are visualized. Since the cameras have narrow FOV (48° horizontal), it is likely that adjacent images don't have enough overlap, which makes the pose graph not fully connected. Fortunately, LiDAR clouds have much wider FOV and therefore guarantees a fully connected graph.

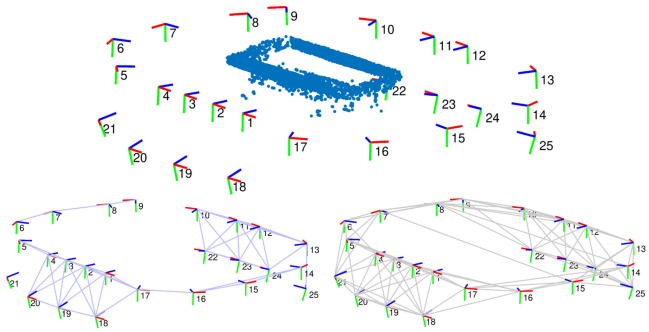

Fig. 8. *Top:* Estimated camera poses (numbered in the order of capture) and visual landmarks (blue points). We follow the convention to define camera frame z (blue) forward, y (green) downward. *Bottom:* Pose graph connections from images (blue) and poing clouds (gray).

As to the computation statistics, we provide a rough measure of the processing time of the major components. On a standard desktop (i7-3770 CPU, 3.40 GHz $\times$ 8), it takes less than 2min to remove vignetting effects and triangulate a stereo pair (40–50 min for the whole dataset). The feature-based cloud registration takes about 15 min in total and the joint pose estimation and map refinement can be finished in about 15 min and 20 min respectively.

In addition to the T-shaped specimen, we tested our algorithm in different environments, where the shapes of reconstructed objects vary from simple squared and cylinder pillars to more complex bridge pillars (see Fig. 9). Table I summarizes the model statistics. The averaged error is obtained by comparing to a ground truth model and more details are provided in Section IV-E.

C. LiDAR-Camera Calibration

In this section, we evaluate the accuracy of the recovered extrinsic transform. As a comparison, we implemented a target-free calibration method [6] which uses discontinuities in images and point clouds to iteratively refine an initial guess. The key steps of this method are shown in Fig. 10a–d. Basically, the initial guess is perturbed in each dimension (x, y, z, roll, pitch, yaw) separately and then moved towards the direction that increases the correlation between image edges and projected cloud edges. Eventually, a locally optimal solution can be found if any further changes will decrease the edge correlation.

Since it is difficult to get ground truth calibration, we choose to compare the extrinsic parameters computed from two methods. The extracted point cloud edges are projected on to the image plane and the projection is visualized in Fig. 10e and 10f. However, the edges are both well aligned and no obvious difference can be identified. We then compare the overlay of LiDAR clouds and stereo clouds (see Fig. 11). It can be observed that with our results, the models are aligned consistently while there exists an offset if calibrated using [6]. Further investigation shows that the offset happens along the camera's optical axis, in which direction the motion will generate less flow on the image. As a result, the total correlation score becomes less sensitive to the motion of the LiDAR along the optical axis. This observation suggests that calibration methods using direct feature alignment, including target-based and target-free, may require wide angle lenses.

D. Observability of Extrinsic Transform

The uniqueness conditions stated in Section III basically requires the sensor pod to change its position and orientation for different stations. In this section, we aim at providing more intuition behind the formal statements. Specifically, the conditions are experimentally demonstrated by perturbing the extrinsic parameters around their optimal values. Three tests are designed to clarify the situations of degeneration.

1) Rotation is Fixed: In this case, the sensor pod is placed at 3 different positions but keeps its orientation unchanged. Specifically, station 1-3 are used for optimization. The total cost after the perturbation is visualized in the left 2 plots of Fig. 12. It can be seen that perturbing the translation won't affect the cost value at all, meaning unobservable. Besides, since the 3 frames are almost collinear, the pitch angle is also under-constrained (flat orange curve).

2) Rotation About One Axis: In this case, stations 1–17 are used, where the sensor pod is placed around the T-shaped specimen and all rotations are about the camera's y-axis. As shown in the middle plots of Fig. 12, position y is under-constrained.

3) Rotation About Two Axes: For reference, we show the perturbed cost with all 25 available datasets in the right plots of Fig. 12. In this case, the rotations can be about x- or y-axis. As expected, the extrinsic transform is well constrained.

E. Model Accuracy Evaluation

Since the ground truth data are not available during the test in Tokyo, we evaluate the reconstruction accuracy on the squared concrete pillar instead. A FARO FOCUS^{3D} scanner (see Fig. 13) with ± 3 mm range precision is used to obtain the ground truth. The comparison is performed by measuring the point to plane distance between the reconstructed model and the ground truth after precise ICP registration. Furthermore, we compare the results of three models reconstructed using: (1) stereo images only (standard stereo BA), (2) both LiDAR and stereo data but extrinsic calibration is pre-calibrated using [6], and (3) both LiDAR and stereo data with extrinsic calibration being adjusted jointly (proposed in this work). Comparisons (1) and (2) share the same cost function in (3). However, in comparison (1) LiDAR observations are set to have zero weights and $\mathbf{T}_e$ is fixed, and in comparison (2) only $\mathbf{T}_e$ is fixed during optimization.

The error maps and histograms are visualized in Fig. 13. It can be observed that fusing LiDAR data helps to reduce the model error from 6 mm to 2.7 mm, which already lies in the precision range of the ground truth. In fact, due to the limited number of matches between some image frames, the pure image-based model does not align well, resulting in multiple layers of the surface. Compared with the pre-calibrated case, jointly optimizing the calibration improves the overall model accuracy and we also benefit from the convenience of self-calibration. Additionally, since our model is reconstructed from multiple sets of data and each station is collected close to the wall (2–3

Fig. 9. From top to bottom, the results of three tests are visualized: a squared pillar (top), a cylinder pillar (middle) and a bridge pillar (bottom). From left to right, we visualize the camera poses and landmarks (blue points), a sample of the image data, complete LiDAR point cloud, overlaid LiDAR and stereo point cloud, dense stereo point cloud.

TABLE I
DATASET AND MODEL STATISTICS

Datasets	Stations (Frames)	# of LiDAR points ($\times 10^6$)	# of stereo points ($\times 10^6$)	Error (mm)
T-shaped	25	32.4	78.4	N/A
squared	29	39.1	210.3	2.7
cylinder	54	66.5	111.7	N/A
bridge	32	38.6	168.7	3.9

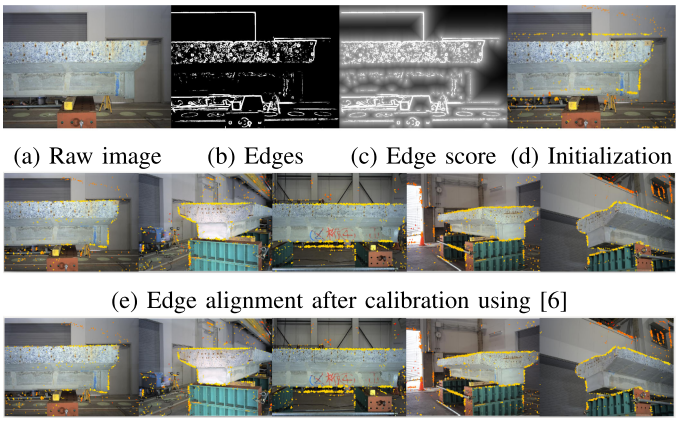

Fig. 10. (a)–(d) The key steps of [6]. (e)–(f) Comparison of extrinsic calibration results from [6] (e) and ours (f). The color of projected cloud edge points encodes the correlation score: yellow means high while red means low.

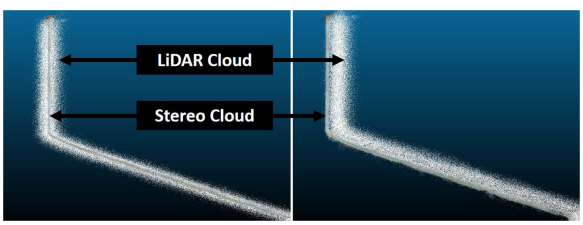

Fig. 11. Cutaway view of the overlaid LiDAR clouds (white) and stereo clouds (textured). *Left:* Jointly optimized. *Right:* Calibrated using [6].

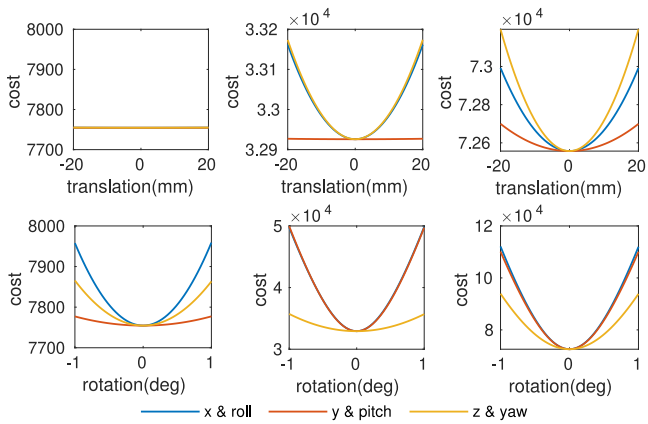

Fig. 12. Changes of cost values w.r.t. perturbed extrinsic transform. From left to right, the three columns show the cost changes in three tests: with rotation fixed, with rotation about one axis, and with rotation about two axes. Within each test, translation (top plots) and rotation (bottom plots) perturbations are visualized separately.

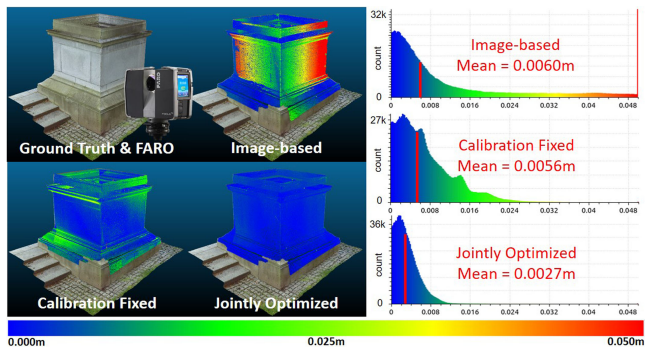

Fig. 13. Comparing the reconstructed models with the ground truth model built by the FARO scanner. On the left are visualizations of the ground truth model and the distance map of reconstructed models, where the color encodes the distance error between two point clouds. On the right are the distance histograms corresponding to each comparison and the averaged errors are marked by the red vertical bar.

meters), it measures about 70 points/cm^2, which is much denser than the ground truth (10–15 points/cm^2). The evaluation results are obtained using the *CloudCompare* software.

V. Conclusions

This letter presents a joint optimization approach to fuse LiDAR and camera for pose estimation and dense reconstruction. It is shown to be able to build dense 3D models and recover camera-LiDAR extrinsic transform accurately. Besides, the accuracy of the reconstructed model is evaluated by comparing to a ground truth model and it shows our method can achieve accuracy similar to a survey scanner.

The proposed method requires data to be collected station by station, which can be time consuming and inconvenient if the viewpoint is difficult to access. For example, the I-shaped beams supporting the deck of a bridge are usually too high to reach. Therefore, future work will be focused on handling sequential data with the sensor pod moving in the environment. Micro Aerial Vehicles (MAVs) may also be used to carry the sensor pod. Another thread of future work is to improve the quality of stereo reconstruction. For instance, given the LiDAR-camera extrinsic calibration obtained from our method, probabilistic fusion methods such as [18] can be applied to recover a dense local map.

Acknowledgment

The authors would like to thank D. Hayashi for his help with experiments in Japan and H. Yu, H. Zhang, and R. Liu for building the sensor pod and helping with data collection.

References

[1] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3-d lidar using line and plane correspondences," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 5562–5569.

[2] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. IEEE/RSJ Int. Conf. Robot. Autom.*, 2012, pp. 3936–3943.

[3] Z. Pusztai and L. Hajder, "Accurate calibration of lidar-camera systems using ordinary boxes," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2017, pp. 394–402.

[4] M. Vel'as, M. Španěl, Z. Materna, and A. Herout, "Calibration of rgb camera with velodyne lidar," in *Proc. WSCG Commun. Papers*, 2014, vol. 2014, pp. 134–144.

[5] 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.

[6] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Proc. Robot.: Sci. Syst.*, 2013, pp. 29–36.

[7] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information." in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 2053–2059.

[8] R. Ishikawa, T. Oishi, and K. Ikeuchi, "Lidar and camera calibration using motions estimated by sensor fusion odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7342–7349.

[9] S. Schneider, T. Luettel, and H.-J. Wuensche, "Odometry-based online extrinsic sensor calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1287–1292.

[10] J. Brookshire and S. Teller, "Extrinsic calibration from per-sensor egomotion," in *Proc. Robot.: Sci. Syst.*, 2013, pp. 504–512.

[11] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[12] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 1141–1148.

[13] Y. Bok, D.-G. Choi, and I. S. Kweon, "Sensor fusion of cameras and a laser for city-scale 3d reconstruction," *Sensors*, vol. 14, no. 11, pp. 20882–20909, 2014.

[14] P. Vechersky, M. Cox, P. Borges, and T. Lowe, "Colourising point clouds using independent cameras," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3575–3582, Oct. 2018.

[15] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2174–2181.

[16] H. Badino, D. Huber, and T. Kanade, "Integrating lidar into stereo for fast and improved disparity computation," in *Proc. Int. Conf. 3D Imag., Modeling, Process., Visualization Transmiss.*, 2011, pp. 405–412.

[17] O. Miksik, Y. Amar, V. Vineet, P. Pérez, and P. H. Torr, "Incremental dense multi-modal 3d scene reconstruction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 908–915.

[18] W. Maddern and P. Newman, "Real-time probabilistic fusion of sparse 3d lidar and dense stereo," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2181–2188.

[19] H. Courtois and N. Aouf, "Fusion of stereo and lidar data for dense depth map computation," in *Proc. Res., Educ. Develop. Unmanned Aerial Syst. Workshop*, 2017, pp. 186–191.

[20] W. Moussa, M. Abdel-Wahab, and D. Fritsch, "Automatic fusion of digital images and laser scanner data for heritage preservation," in *Proc. Eur.-Mediterranean Conf.*, 2012, pp. 76–85.

[21] W. Neubauer, M. Doneus, N. Studnicka, and J. Riegl, "Combined high resolution laser scanning and photogrammetrical documentation of the pyramids at Giza," in *Proc. CIPA 20th Int. Symp.*, 2005, pp. 470–475.

[22] A. Abdelhafiz, B. Riedel, and W. Niemeier, "Towards a 3d true colored space by the fusion of laser scanner point cloud and digital photos," in *Proc. ISPRS Working Group V/4 Workshop (3D-ARCH)*, 2005, pp. 135–144.

[23] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[24] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. Euro. Conf. Comput. Vision*, 2006, pp. 404–417.

[25] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *Int. J. Comput. Vision*, vol. 116, no. 1, pp. 66–89, 2016.

[26] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Proc. Sensor Fusion IV: Control Paradigms Data Struct.*, 1992, vol. 1611, pp. 586–607.

[27] Z. Dong, B. Yang, Y. Liu, F. Liang, B. Li, and Y. Zang, "A novel binary shape context for 3d local surface description," *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 431–452, 2017.

[28] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Softw. Tools*, vol. 120, pp. 122–125, 2000.