

Optimal Target Shape for LiDAR Pose Estimation

Jiunn-Kai Huang, William Clark, and Jessy W. Grizzle

Abstract—Targets are essential in problems such as object tracking in cluttered or textureless environments, camera (and multi-sensor) calibration tasks, and simultaneous localization and mapping (SLAM). Target shapes for these tasks typically are symmetric (square, rectangular, or circular) and work well for structured, dense sensor data such as pixel arrays (i.e., image). However, symmetric shapes lead to pose ambiguity when using sparse sensor data such as LiDAR point clouds and suffer from the quantization uncertainty of the LiDAR. This paper introduces the concept of optimizing target shape to remove pose ambiguity for LiDAR point clouds. A target is designed to induce large gradients at edge points under rotation and translation relative to the LiDAR to ameliorate the quantization uncertainty associated with point cloud sparseness. Moreover, given a target shape, we present a means that leverages the target’s geometry to estimate the target’s vertices while globally estimating the pose. Both the simulation and the experimental results (verified by a motion capture system) confirm that by using the optimal shape and the global solver, we achieve centimeter error in translation and a few degrees in rotation even when a partially illuminated target is placed 30 meters away. All the implementations and datasets are available at https://github.com/UMich-BipedLab/optimal_shape_global_pose_estimation.

I. INTRODUCTION

Targets have been widely employed as fiducial markers [1]–[9] and for target-based sensor calibration [10]–[21]. Fiducial markers (artificial landmarks or targets) help robots estimate their pose by estimating the target pose and are applied to simultaneous localization and mapping (SLAM) systems for robot state estimation and loop closures. Additionally, it can facilitate human-robot interaction, allowing humans to give commands to a robot by showing an appropriate marker. Extrinsic target-based calibration between sensors (cameras, Light Detection and Ranging (LiDAR) sensors, Inertial Measurement Units (IMU), etc) is crucial for modern autonomous navigation. Particularly in target-based LiDAR-camera calibration, one seeks to estimate a set of corresponding features of a target (e.g., edge lines, normal vectors, vertices, or plane equations) in the LiDAR’s point cloud and the camera’s image plane.

The targets applied in these critical tasks are typically symmetric (square, diamond, rectangle, or circle). A symmetric target, such as a square [1], [3], leads to undefined and ambiguous pose which is often solved by adding an additional pattern to the squared target and then rotating the target four times (four symmetry axes), see [1, Figure 6]. Furthermore, estimating the pose or features of a target of injudicious design suffers from the quantization uncertainty of a sensor, especially for LiDAR sensors. A high-end LiDAR, such as *32-Beam Velodyne ULTRA Puck LiDAR*, still has roughly six centimeters

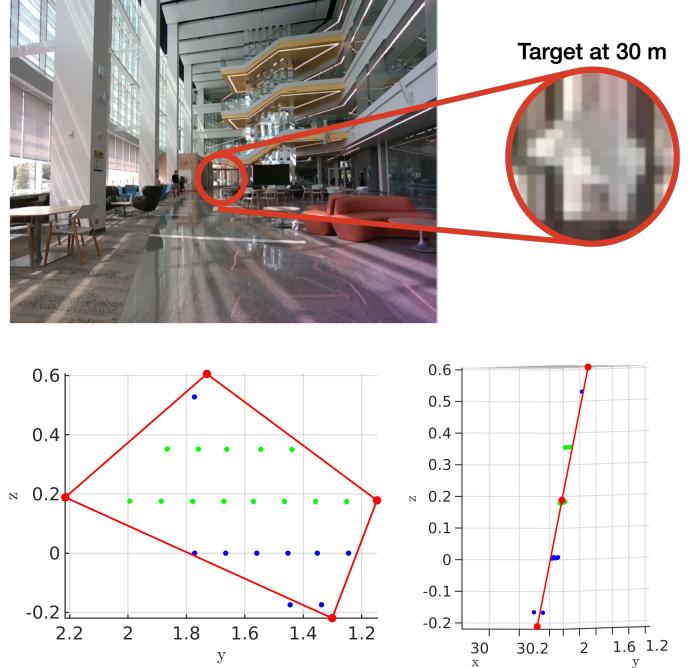


Fig. 1: Illustration of the vertex and pose estimation using the proposed optimal target shape placed 30 meters away from a LiDAR in the atrium of the Ford Robotics Building at the University of Michigan. The red frame is the proposed optimal shape that induces large gradients at edge points under translation and rotation. The pose and vertices of the target are jointly estimated by a global solver that uses known target geometry. The bottom two figures show the front view and a side view of the fitting results, respectively. The blue and the green dots are the LiDAR returns on the target. Only the blue dots are used for pose estimation while to demonstrate robustness of the approach, the green dots are considered missing. If one were to apply RANSAC — a commonly used method — to regress the target boundaries and subsequently estimate the vertices by line intersections, the method would fail due to the sparsity of inliers.

of quantization error at 10 meters, and 18 centimeters at 30 meters. The quantization uncertainty of a LiDAR results in poor pose estimates for targets farther than 15 meters away.

In this paper, we propose the concept of optimizing target shape to ameliorate problems caused by quantization uncertainty and sparsity of the LiDAR image of a target. Specifically, we propose that a “good target shape” is one that possesses large gradients at edge points when the target undergoes rotations or translations. Moreover, we present a means that exploits target geometry to extract target vertices while estimating pose. The pose estimation problem is formulated so that an existing Semidefinite programming (SDP) global solver can be modified to globally and efficiently compute the target’s pose. Figure 1 shows the obtained pose estimation of a partially illuminated target placed 30 meters away and having only nine returns (blue dots) from a *32-Beam Velodyne ULTRA Puck LiDAR*, and three LiDAR rings on the target after the green dots are considered missing.

Jiunn-Kai Huang, William Clark, and J. Grizzle, are with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109, USA. {bjhuang, wiclark, grizzle}@umich.edu.

II. RELATED WORK

To the best of our knowledge, there is no existing work on target shape design for LiDAR point clouds. We instead review some techniques to improve the pose estimation of fiducial markers and to assist in extracting features of calibration targets.

A. Fiducial Markers

Fiducial markers for cameras were originally developed and used for augmented reality applications [4], [5] and have been widely used for object detection and tracking, and pose estimation [22]. Due to their uniqueness and fast detection rate, they are also often used to improve Simultaneous Localization And Mapping (SLAM) systems [23]. CCTag [24] adopts a set of rings (circular target) to enhance pose estimation from blurry images. ChromaTag [6] proposes color gradients on a squared target to speed up the detection process and obtain more accurate pose estimation. More recently, LFTag [9] has taken advantage of topological markers, a kind of uncommon topological pattern, on a squared target to improve pose estimation at a longer distance. However, all the mentioned fiducial markers only work on cameras.

In our prior work on LiDAR [1], we proposed the first fiducial marker for LiDAR point clouds, which can be perceived by both LiDARs and cameras. We achieved millimeter error in translation and a few degrees in rotation. However, due to the quantization error of the LiDAR, the performance of the pose estimation was noticeably degraded when the target was farther than 12 meters. Thus, this work proposes the concept of target shape design to specifically address the quantization uncertainty present in LiDAR returns and push the range of pose estimation to more than 30 meters. In passing, we note that symmetric targets, such as a square or hexagon, suffer from rotational ambiguity. Hence, our designed target will not be symmetric.

B. Target-Based LiDAR-camera Calibration

LiDAR-camera calibration [11]–[21] requires feature correspondences from the image pixels and the LiDAR point cloud. However, the representations and inherent properties of camera images and LiDAR point clouds are distinct. An image (pixel arrays) is dense and very structured, with the pixels arranged in a uniform (planar) grid, and each image has a fixed number of data points. On the other hand, each scan of a LiDAR returns a 3D point cloud consisting of a sparse set of (x, y, z) coordinates with associated intensities. In particular, LiDAR returns are not uniformly distributed in angle or distance [2, III-A]. Target-based LiDAR-camera calibration utilizes targets to identify and estimate the corresponding features, such as vertices, 2D/3D edge lines, normal vectors, or the plane equations of the targets. References [10]–[13] have noted that placing the targets so that the rings of the LiDAR ran parallel to its edges led to ambiguity in the vertical position due to the spacing of the rings and thus was detrimental to vertex or feature estimation. References [16], [20] utilize RANSAC [25] and plane fitting to remove the outliers of the LiDAR returns, while [13] proposes a

“denoising process” for LiDAR returns around the target boundaries before applying RANSAC to extract features. When estimating the target vertices, the later references separate the edge points into groups and then apply the RANSAC algorithm. However, regressing the line equation of edge points will fail when there are not enough edge points or inliers, as shown in Fig. 1. Additionally, no target geometry information is used while estimating the features.

C. Contributions

In particular, this work presents the following contributions:

- 1) We propose the concept of target shape optimization for estimating pose and vertices from LiDAR point clouds. Specifically, we design a target so that its edge points induced by LiDAR rings are “highly” sensitive to translation and rotation. This attenuates the effects of quantization uncertainty and sparsity of a target’s LiDAR image. The resulting shape is asymmetric to remove pose ambiguity.
- 2) We present a means that uses target shape to jointly estimate target vertices and pose. Because the cost function of the proposed method can be formulated as an SDP, the target’s pose and vertices can be globally estimated with an open-source solver [26].
- 3) We utilize an open-source LiDAR simulator [27] to provide ground truth of the poses and vertices. In the simulation, we validate that the optimal shape with the global solver achieves centimeter error in translation and a few degrees of error in rotation when the targets are at a distance of 30 meters and partially illuminated. In addition, we conduct experimental evaluations where the ground truth data are provided by a motion capture system, and achieve results similar to the simulation.
- 4) We open-source all the related software for this work, including the generation of the optimal shape, our means for pose estimation, and the simulated/experimental datasets; see [28], [29].

The remainder of this paper is organized as follows. Section III formulates the design of an optimal target shape for LiDAR point clouds. The extraction of the target vertices while globally estimating the pose is discussed in Sec. V. The simulation and experimental results are presented in Sec. VI and Sec. VII. Finally, Sec. VIII concludes the paper and provides suggestions for further work.

III. OPTIMAL SHAPE FOR SPARSE LiDAR POINT CLOUDS

In this section, we propose a mathematical formulation of target shape design. The main idea is for target translation and rotation to result in large gradients at edge points defined by the LiDAR returns. Fig. 2 summarizes a high-level optimization process.

A. Convex Shape Generation

We apply projective transformations on a nominal convex quadrilateral in 3D to generate planar candidate targets. We will see that applying a projective transformation rather than working with a collection of vertices makes it easier to ensure

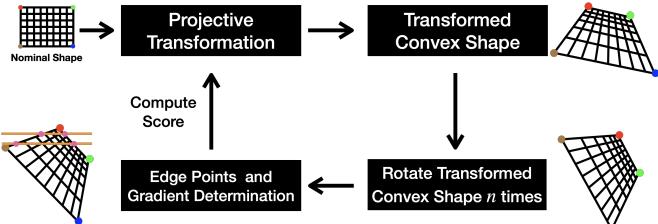


Fig. 2: Optimization process for determining an optimal target shape. Projective transformations applied to a nominal quadrilateral generate candidate convex quadrilaterals (targets). Edge points are intersections of LiDAR rings with the target boundaries. The objective is to maximize the gradient of edge points under actions of $\text{SE}(2)$ applied to the target. To enhance robustness, the gradients are computed for n -discrete rotations of the quadrilateral under partial illumination, and the score is the worst case.

convexity of the target and to generate a cost function that is invariant under scaling and translations.

Let $\mathcal{V} := \{X_i | X_i := (x_i, y_i, 1) | x_i, y_i \in \mathbb{R}\}_{i=1}^4$ denote the 3D vertices X_i of a nominal convex quadrilateral, such as a square. Given \mathbf{P} , a projective transformation defined by a non-singular 3×3 matrix [30, p.33], let \tilde{X}_i denote the new vertices transformed by \mathbf{P} :

$$\tilde{X}_i = \begin{bmatrix} x'_i \\ y'_i \\ \lambda'_i \end{bmatrix} = \mathbf{P}X_i = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & v \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \quad (1)$$

The resulting vertices $\tilde{\mathcal{V}} := \{\tilde{X}_i\}_{i=1}^4$ lie in the projective space \mathbb{P}^2 [30, p.26]. Let \mathcal{V}' be the corresponding transformed vertices in the Cartesian space (\mathbb{R}^2) [30, p.27] and $\Pi : \mathbb{P}^2 \rightarrow \mathbb{R}^2$ be the mapping function

$$\Pi(\tilde{\mathcal{V}}) := \Pi(\mathbf{P}(\mathcal{V})) := \left\{ X'_i | X'_i := \left(\frac{x'_i}{\lambda'_i}, \frac{y'_i}{\lambda'_i} \right) \right\}_{i=1}^4. \quad (2)$$

To summarize, given a nominal convex quadrilateral, \mathcal{V} , and a projective transformation, \mathbf{P} , we construct a new quadrilateral via

$$\mathcal{V} \longmapsto \mathbf{P}\mathcal{V} =: \tilde{\mathcal{V}} \text{ and } \mathcal{V}' := \Pi(\tilde{\mathcal{V}}). \quad (3)$$

Remark 1. From here on, we will abuse notation and pass from Cartesian coordinates to homogeneous coordinates without noting the distinction.

It is important to note that for any desired quadrilateral, there exists a projective transformation yielding \mathcal{V}' from \mathcal{V} . Hence, our procedure for generating candidate targets is without loss of generality.

Theorem 1 ([31, p.274]). *There exists a unique projective transformation that maps any four points, no three of which are collinear, to any four points, no three of which are collinear.*

While Theorem 1 can be used to construct an arbitrary quadrilateral, convexity need not be conserved as shown in Fig. 3. We need the following condition to ensure that the resulting polygon is convex:

Theorem 2 ([32, p.39]). *Let $\Omega : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ by $\Omega(x, y, \lambda) = (x/\lambda, y/\lambda)$. If $\text{dom}\Omega = \mathbb{R}^2 \times \mathbb{R}_{++}$, where $\mathbb{R}_{++} = \{x \in \mathbb{R} | x >$*

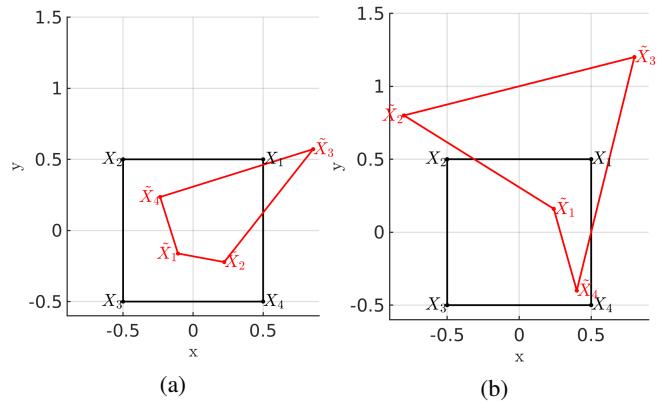


Fig. 3: Equation (5) versus convexity. The red indicates the resulting shapes transformed by projective transformations from the same nominal shape (black). The left shows a case where (5) is satisfied and thus the transformed shape is convex; otherwise, it is non-convex, as shown on the right, where $p_{31}x_3 + p_{32}y_3 + v = -0.5$.

$\} \text{ and the set } C \subseteq \text{dom}\Omega \text{ is convex, then its image}$

$$\Omega(C) = \{\Omega(x) | x \in C\} \quad (4)$$

is also convex.

For the domain of our projective transformation to be $\mathbb{R}^2 \times \mathbb{R}_{++}$, and hence for the candidate target to be automatically convex, the following linear inequality should be imposed in (1),

$$p_{31}x_i + p_{32}y_i + v > 0. \quad (5)$$

Because we consider two candidate targets to be equivalent if one can be obtained from the other by translation and scaling, we are led to decompose the projective transformation as follows [33], [34],

$$\mathbf{P} = \mathbf{H}_S \mathbf{H}_{SH} \mathbf{H}_{SC} \mathbf{H}_E \\ = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & \mathbf{1} \end{bmatrix} \begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda & 0 & 0 \\ 0 & 1/\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^\top & v \end{bmatrix}, \quad (6)$$

where $\mathbf{H}_S, \mathbf{H}_{SH}, \mathbf{H}_{SC}, \mathbf{H}_E$ are similarity, shearing, scaling and elation transformations, respectively; see [33] for more details. By setting $s = 1, \mathbf{t} = (0, 0)$ in (6), the degree of freedom (DoF) of the projective transformation drops from eight to five. Our family of candidate targets is now given by (2) with \mathbf{P} satisfying (5) and (6).

In summary, we can describe candidate convex target shapes via projective transformations while reducing the number of degrees of freedom to five.

B. Edge Points Determination

As mentioned in (2), \mathcal{V}' are the 2D vertices of a candidate target. An edge point E_i is defined by the intersection point of a LiDAR ring and the line connecting two vertices of the polygon as shown in Fig. 4. If we let \mathcal{S} be the boundary of the quadrilateral with vertices \mathcal{V}' , the collection of edge points detected by the LiDAR is the set $\mathcal{EP} := \{E_i\}_{i=1}^M$, where M

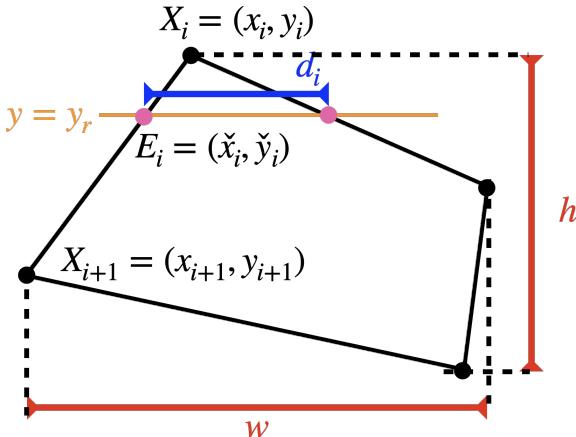


Fig. 4: The pink dots are the edge points determined by a LiDAR ring (orange line) intersecting with the edge line connecting two vertices (X_i, y_i) and (X_{i+1}, y_{i+1}) . The distance between the two edge points on the same LiDAR ring is d_i . The height and width of the shape is w and h , respectively.

is the number of edge points and is given by the intersection of \mathcal{S} with the LiDAR rings, i.e.

$$\mathcal{S} = \partial \text{conv}(\mathcal{V}'), \quad \{E_i\}_{i=1}^M = \mathcal{S} \cap \mathcal{LR}. \quad (7)$$

When the LiDAR rings are horizontal, an edge point $E_i = (\check{x}_i, \check{y}_i)$ can always be computed in closed form,

$$\check{x}_i = x_i + \frac{x_{i+1} - x_i}{y_{i+1} - y_i} (y_r - y_i) \quad \text{and} \quad \check{y}_i = y_r. \quad (8)$$

C. Shape Sensitivity

The sensitivity of a polygon is defined as the gradient of the edge points with respect to rigid-body transformations of the polygon, with the LiDAR rings held constant, as shown in Fig. 5. Hence, the sensitivity captures the horizontal movement of an edge point after the shape is rotated and translated.

For a transformation $\mathbf{H} \in \text{SE}(2)$, let E'_i denote the transformed edge point by

$$E'_i := \mathbf{H} \circ E_i = \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \check{x}_i \\ \check{y}_i \\ 1 \end{bmatrix}, \quad (9)$$

where θ, t_x, t_y are the rotation angle, the translation on x-axis, and the translation on y-axis, respectively. Denote $\text{Exp}(\kappa, \omega, u, v)$ as the exponential map that maps from the Lie algebra $\mathfrak{se}(2)$ to the continuous Lie group $\text{SE}(2)$,

$$\begin{aligned} \text{Exp}(\kappa, \omega, u, v) &= \text{expm} \left(\kappa \begin{bmatrix} 0 & -\omega & u \\ \omega & 0 & v \\ 0 & 0 & 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} \cos(\omega\kappa) & -\sin(\omega\kappa) & \frac{1}{\omega}(v \cos(\omega\kappa) + u \sin(\omega\kappa) - v) \\ \sin(\omega\kappa) & \cos(\omega\kappa) & \frac{1}{\omega}(v \sin(\omega\kappa) - u \cos(\omega\kappa) + u) \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (10)$$

where (ω, u, v) parameterize the unit sphere $S^2 \subset \mathbb{R}^3$, expm is the usual matrix exponential, and κ is a dummy variable for

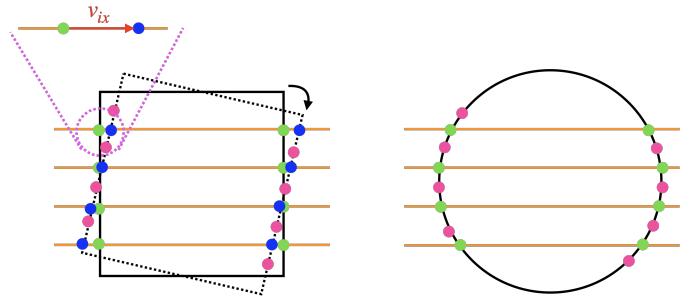


Fig. 5: Shape sensitivity under rotation. The sensitivity of a shape is defined by the gradient (v_{ix}) of each edge point as it moves along the ring lines (orange) under rotations and translations of the shape. The green dots are the original edge points and the pink dots are the edge points on the rotated shape. The same process can be done for translation. The left shows a gradient v_{ix} of a square where the dotted square has been rotated from the nominal positioned square. The right shows that the gradient of an edge point on a circle is zero under rotation about its center. The gradient would be non-zero for translations.

differentiation. Comparing \mathbf{H} in (9) and (10) leads to

$$\begin{cases} \theta = \omega\kappa \\ t_x = \frac{1}{\omega}(v \cos(\omega\kappa) + u \sin(\omega\kappa) - v) \\ t_y = \frac{1}{\omega}(v \sin(\omega\kappa) - u \cos(\omega\kappa) + u). \end{cases} \quad (11)$$

For each triple of values (ω, u, v) , the action of (10) on a candidate target quadrilateral results a path $p_i(\kappa) := p_i(\text{Exp}(\kappa\omega, \kappa u, \kappa v))$ being traced out by the edge points along a LiDAR ring. Using (11) to differentiate the path p_i at the identity of $\text{SE}(2)$ produces an action of \mathfrak{se}_2 ,

$$v_i(\omega, u, v) := \frac{d}{dt} \Big|_{\kappa=0} p_i(\text{Exp}(\kappa\omega, \kappa u, \kappa v)), \quad (\omega, u, v) \in \mathfrak{se}_2. \quad (12)$$

From (8), (10), (11), and (12), the gradient of the edge point with respect to the LiDAR ring is

$$\begin{aligned} v_{i_x} &= \omega \left(\frac{(x_i - x_{i+1})(x_i y_{i+1} - y_i x_{i+1} + x_{i+1} y_r - x_i y_r)}{(y_i - y_{i+1})^2} - y_r \right) \\ &\quad + u + \left(\frac{x_{i+1} - x_i}{y_{i+1} - y_i} \right) v. \end{aligned} \quad (13)$$

Notice that $v_{i_y} = 0$ because the y -coordinates of $\mathbf{H}(\mathcal{V}') \cap \mathcal{LR}$ remain fixed.

Finally, we define the sensitivity \mathcal{M} of the polygon

$$\mathcal{M}(\mathcal{V}, \mathcal{LR}, \omega, u, v) := \frac{1}{h} \sum_{i=1}^M v_{i_x}^2, \quad (14)$$

where M is the number of edge points, and h , defined in Fig. 4 is included because gradients in (13) scale with vertical height.

D. Initial Cost Function

The candidate target's sensitivity defined in (14) does not take into account the discrete nature of the LiDAR returns on a given ring. Let d_i denote the distance between two edge points on the i -th LiDAR ring. We want to encourage targets that have d_i larger than the spatial quantization of the LiDAR returns. We can do this in two ways, by scaling (14) by w and by

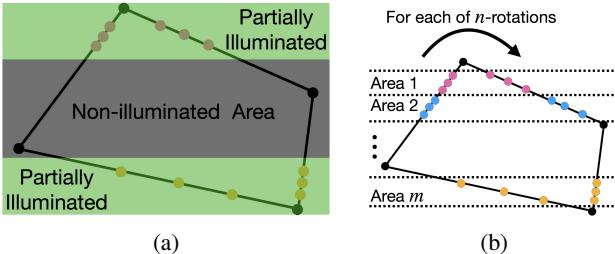


Fig. 6: The left shows a partially illuminated candidate shape. The right shows that the target is divided into m -areas of partial illumination, and that for each of n -rotations of the candidate target, a score is assigned to each subarea based on (15). The final score of the shape is the lowest among the $n \times m$ scores.

including a term in the cost of the form $\sum_{i=1}^K d_i$ (d_i is larger for wider targets). The resulting score of shape (Ψ) becomes

$$\Psi = w\mathcal{M} + \mu \sum_{i=1}^K d_i, \quad (15)$$

where w is the width of the polygon, K is the number of rings illuminating the polygon, and μ is a weight trading off the two terms in the cost function.

IV. ROBUST SHAPE FOR REAL LiDAR SENSORS

In previous section (Sec. III-D), we added an extra term to (15) to account for the discrete measurements in azimuth direction of the LiDAR. Additionally, LiDARs have different ring densities at different elevation angles. For example, 32-Beam Velodyne ULTRA Puck LiDAR has dense ring density between -5° and 3° , and has sparse ring density from -25° to -5° and from 3° to 15° [35]. A target could be partially illuminated in the sparse region, as shown in Fig. 6a. Therefore, assuming edge points are uniformly distributed is not practical and using a distribution of edge points that is similar to reality is critical while maximizing (15). Additionally, LiDAR rings from mobile robots are not always parallel to the ground plane. We account for non-horizontal LiDAR rings by rotating the candidate target.

A. Partial Illumination of Target

To have the shape being robust to illuminated area and the angle of the rings with respect to the target, we first rotate the generated polygon n times, and then divide the rotated polygon into m areas. Only one area is illuminated by LiDAR rings at a time to determine edge points and to compute the score (15), Ψ_{ij} , for $1 \leq i \leq n$ and $1 \leq j \leq m$. Figure 6b shows the edge points being determined for a partially illuminated target. Equation (15) is consequently evaluated $n \times m$ times for illumination of the target and the lowest among the $n \times m$ scores is the final score of the candidate target shape.

B. Optimization for the Optimal Shape

To summarize, the resulting optimization problem depends on the projective transformation parameters that are used to generate a convex polygon, edge points illuminated by horizontal LiDAR rings lied on the rotated quadrilateral, the transformation of the edge points in se_2 , and distances between

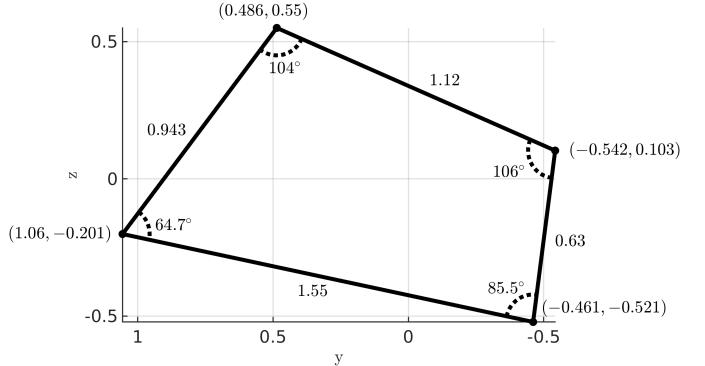


Fig. 7: The resulting optimal shape from (16) in arbitrary units.

two edge points on the corresponding LiDAR rings. Thus, the optimization problem is defined as:

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \min_{\omega, u, v} \max_{i, j} \{-\Psi_{ij}\}. \quad (16)$$

The optimization problem (16) was (locally) solved by fmincon in MATLAB, after the optimization parameters were randomly initialized. We rotated the generated polygon six times. Each rotated polygon was divided into five areas, and four LiDAR rings were used to illuminate one area at a time. The unit sphere of unit vectors in $\text{se}(2)$, mentioned in Sec. III-C, was discretized into 25×25 faces (by normalizing the vectors to have unit length, we can reduce the dimension from three to two). Once the generated polygon was rotated and illuminated, the sensitivity of the resulting edges points were evaluated at each face on the unit sphere. The resulting optimal shape is shown in Fig. 7.

Remark 2. *The main point of this paper is that target shape can be used to enhance the estimation of target vertices and relative pose between a target and a LiDAR. We have proposed one algorithmic means to produce an “optimal target shape”. Different notions of cost will result in different shapes.*

V. GLOBAL POSE AND FEATURE ESTIMATION

In this section, we propose a means to use known target geometry to extract target vertices while globally estimating relative pose between target and LiDAR. For a collection of LiDAR returns $\mathcal{TP} := \{\mathcal{X}_i\}_{i=1}^N$, let $\mathcal{EP} := \{E_i\}_{i=1}^M \in \mathcal{TP}$ be the M target edge points. Given the target geometry, we define a template with vertices $\{\bar{X}_i\}_{i=1}^4$ located at the origin of the LiDAR and aligned with the y - z plane as defined in Fig. 8. We also denote $\bar{\mathcal{L}} := \{\bar{\ell}_i\}_{i=1}^4$ as the line equations of the adjacent vertices of the template. We seek a rigid-body transformation from the template to the target, $\mathbf{H}_L^T \in \text{SE}(3)$, that “best fits” the template onto the edge points. In practice, it is actually easier to project the edge points \mathcal{EP} back to the origin of the LiDAR through the inverse of the current estimate of transformation $\mathbf{H}_T^L := (\mathbf{H}_L^T)^{-1}$ and measure the error there. The action of $\mathbf{H} \in \text{SE}(3)$ on \mathbb{R}^3 is $\mathbf{H} \cdot \mathcal{X}_i = \mathbf{R}\mathcal{X}_i + \mathbf{t}$, where $\mathbf{R} \in \text{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$.

The cost j_i of edge point $E_i \in \mathcal{EP}$ is defined as the point-to-line distance,

$$j_i(E_i; \bar{\mathcal{L}}) = \min_{\bar{\ell}_i \in \bar{\mathcal{L}}} \|E_i - \bar{\ell}_i\|_2^2 \quad (17)$$

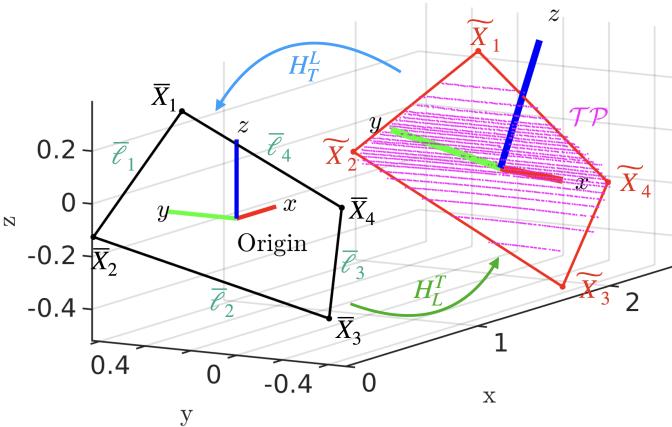


Fig. 8: Pose definition and illustration of template fitting. A coordinate frame for the template (target shown in black) is defined by aligning the plane of the template with the y - z plane of the LiDAR frame and also aligning the mean of its vertices with the origin of the LiDAR. Let H_T^L (blue arrow) be an estimate of the rigid-body transformation from target to LiDAR, projecting the edge points of the target back to the template. The estimated pose of the target is then given by the inverse transformation, H_L^T (green arrow). The optimal H_T^L is obtained by minimizing (18) (based on point-to-line distance). This figure also shows a fitting result of a target at 2 meters in the Ford Robotics Building. The red frame is the template re-projected onto the LiDAR point cloud by H_T^L .

where $\bar{\mathcal{L}}$ is the set of line equations for the target. Let $\{\bar{E}_i\}_{i=1}^M := H_T^L(\mathcal{EP}) = \{H_T^L \cdot E_i\}_{i=1}^M$ denote the projected points by H_T^L . The total fitting error is defined as

$$J(H_T^L(\mathcal{EP})) := \sum_{i=1}^M j_i(\bar{E}_i) \quad (18)$$

To minimize (18), we adopt techniques that were used to globally solve 3D registration or 3D SLAM [26], [36]–[38] where the problem is formulated as a QCQP, and the Lagrangian dual relaxation is used. The relaxed problem becomes a Semidefinite Programming (SDP) and convex. The problem can thus be solved globally and efficiently by off-the-shelf specialized solvers [39]. As shown in [26], the dual relaxation is empirically always tight (the duality gap is zero).

Once we (globally) obtain \mathbf{H}_T^L , the pose of the target is $\mathbf{H}_L^T = (\mathbf{H}_T^L)^{-1}$, and the estimated vertices are $\{\widetilde{X}_i\}_{i=1}^4 := \{\mathbf{H}_L^T \cdot X_i\}_{i=1}^4$. The edge-line equations, the normal vector, and the plane equation of the target can be readily obtained from the vertices.

VI. SIMULATION RESULTS

Before carrying out experiments with the new target shape, we used a MATLAB-based LiDAR simulator introduced in [10] to extensively evaluate the pose and vertex estimation of the optimal shape. Both quantitative and qualitative results are provided. We do not compare against standard targets, such as unpatterned rectangles, diamonds, or circles, because their symmetry properties result in pose ambiguity. At large distances, a pattern would not be discernible.

We simulate a 32-Beam *Velodyne ULTRA Puck LiDAR*, whose data sheet can be found at [35]. A target is placed at distances from 2 to 40 meters in 2 m increments. At each distance, simulation data is collected with a target face-on to the LiDAR as an easy case, and another dataset with the target

rotated by the Euler angles (roll = 20° , pitch = 30° , yaw = 30°) under the XYZ convention as a challenging case. In addition, we induce two different levels of noise to each dataset to examine the robustness of the algorithm.

The results of vertex estimation are reported as the root-mean-square-error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{4} \sum_{i=1}^4 \|\widetilde{X}_i - X_i\|_2^2}, \quad (19)$$

where \widetilde{X}_i is the estimated vertex and X_i is the ground truth vertex from the simulator. The pose on $\text{SE}(3)$ is evaluated on translation e_t in \mathbb{R}^3 and rotation e_r on $\text{SO}(3)$, separately. In particular, e_t and e_r are computed by

$$e_t := \|\mathbf{t} - \widetilde{\mathbf{t}}\| \quad \text{and} \quad e_r := \|\text{Log}(\mathbf{R}\widetilde{\mathbf{R}}^T)\|, \quad (20)$$

where $\|\cdot\|$ is the Euclidean norm, $\widetilde{\cdot}$ is the estimated quantity, \mathbf{R} and \mathbf{t} are the ground truth rotation and translation, respectively, and $\text{Log}(\cdot)$ is the logarithm map in the Lie group $\text{SO}(3)$. Additionally, we also report the percentage of the RMSE and translation error, which are computed by each quantity divided by the centroid of the target. The quantization error e_q is the distance between two adjacent points on the same ring and can be approximated by the azimuth resolution (0.4°) of the LiDAR times the target distance.

Qualitative results are shown in Figure 9. The complete quantitative results of the distances and noise levels are shown as line charts in Fig. 10. Table I shows a subset of quantitative results of the pose and vertex estimation using the noise-free dataset. Based on these results, we were motivated to build the target and run physical experiments.

VII. EXPERIMENTAL RESULTS

We now present experimental evaluations of the pose and vertex estimation of the optimal shape. All the experiments are conducted with a 32-Beam *Velodyne ULTRA Puck LiDAR* and an Intel RealSense camera rigidly attached to the torso of a Cassie-series bipedal robot, as shown in Fig. 11. We use the Robot Operating System (ROS) [40] to communicate and synchronize between the sensors. Datasets are collected in the atrium of the Ford Robotics Building at the University of Michigan, and a spacious outdoor facility, M-Air [41], equipped with a motion capture system.

A. Quantitative Experimental Results in M-Air

The Qualisys motion capture system in M-Air is used as a proxy for ground truth poses and vertices. The setup consists of 33 motion capture cameras with passive markers attached to the target, the LiDAR and the camera, as shown in Fig. 11 and Fig. 12. Datasets are collected at various distances and angles. Each of the datasets contains images (20 Hz) and scans of point clouds (10 Hz). Similar to the simulation environment, the optimal-shape target is placed at distances from 2 to 16 meters (maximum possible in M-Air) in 2 meter increments. At each distance, data is collected with a target face-on to the LiDAR and another dataset with the target roughly rotated by the Euler angles (roll = 20° , pitch = 30° , yaw = 30°)

as a challenging case. The results are shown in Table II. As expected, the results are slightly worse (approximately one degree) than the simulator's due to the white noise of the LiDAR and many missing returns on the targets, as shown in Fig 13.

B. Qualitative Experimental Results and Target Partial Illumination

For distances beyond 16 meters (the distance limit in M-Air), we present qualitative results from the atrium in Fig. 13 to support the simulation-based analysis. The blue dots are LiDAR measurements, and the red frame is the fitting result. Figure 14 illustrates a partially illuminated target (the green dots are assumed missing and only blue dots are used for pose estimation); the resulting fitting results are the red frame.

VIII. CONCLUSION

We presented the concept of optimizing target shape to enhance pose estimation for LiDAR point clouds. We formulated the problem in terms of choosing a target shape that induces large gradients at edge points under translation and rotation so as to mitigate the effects of quantization uncertainty associated with point cloud sparseness. For additional robustness, the cost function or score for a candidate target was defined to be the minimum score under a set of relative rotations of the edge points; this had the effect of breaking symmetry in the candidate target, which also removes pose ambiguity.

For a given target, we used the target's geometry to jointly estimate the target's pose and its vertices. The estimation problem was formulated so that an existing semi-definite programming global solver could be modified to globally and efficiently compute the pose and vertices of the target. A

TABLE I: Pose and vertex accuracy of the simulation results at various distances. The vertex estimation is computed by (19). The quantization error e_q is the distance between two adjacent points on the same ring. The translation error e_t in meters and rotation error e_r in degrees are computed by (20).

Face-on LiDAR							
Distance [m]	No. Points	e_q [m]	RMSE [m]	RMSE [%]	e_t [m]	e_t [%]	e_r [deg]
2.00	2530	0.01	0.01	0.49	0.002	0.11	0.71
4.00	1015	0.03	0.03	0.83	0.005	0.12	2.45
6.00	548	0.04	0.02	0.40	0.01	0.12	1.77
8.00	338	0.06	0.02	0.28	0.01	0.15	1.58
16.00	96	0.11	0.04	0.22	0.03	0.21	0.68
30.00	28	0.21	0.03	0.10	0.03	0.08	1.09
32.00	26	0.22	0.06	0.17	0.05	0.17	1.20
Extreme Angle (20, 30, 30)							
Distance [m]	No. Points	e_q [m]	RMSE [m]	RMSE [%]	e_t [m]	e_t [%]	e_r [deg]
1.94	1734	0.01	0.01	0.62	0.01	0.37	0.71
3.93	716	0.03	0.03	0.71	0.03	0.64	0.95
7.93	258	0.06	0.04	0.46	0.03	0.44	0.96
15.93	70	0.11	0.05	0.31	0.05	0.29	1.37
29.93	21	0.21	0.12	0.39	0.09	0.30	5.65
31.93	17	0.22	0.47	1.46	0.06	0.18	34.72

LiDAR simulator generated synthetic ground truth of the target pose and vertices. We validated that the combination of the optimal shape with the global solver achieved centimeter error in vertex estimation, centimeter error in translation, and a few degrees off in rotation in pose estimation when a partially illuminated target was placed 30 meters from the LiDAR . In experiments, when compared to ground truth data collected by a motion capture system with 33 cameras, we achieved results similar to those of the simulations.

In the future, we shall establish a system to automatically

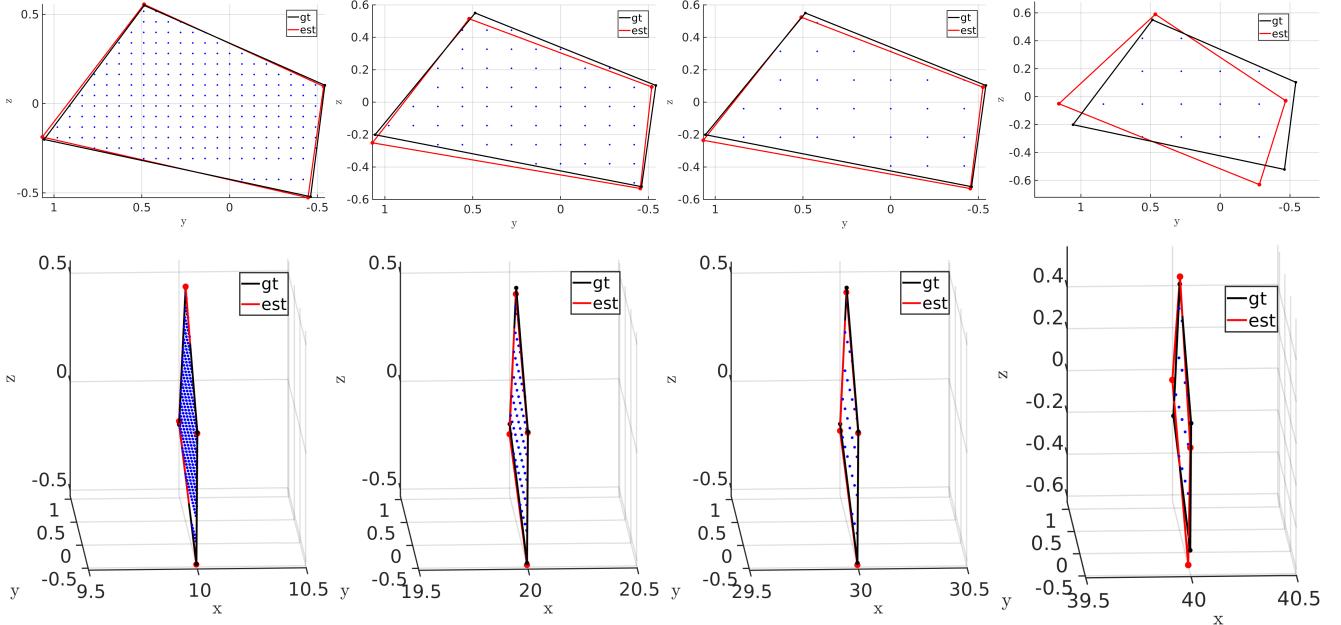


Fig. 9: Simulation results of the noise-free dataset of the pose estimation at various distances (10, 20, 30, 40 m). LiDAR returns (blue dots) on the target are provided by the LiDAR simulator. Black indicates the ground truth pose from the simulator, and red is the estimated pose and vertices. The top and bottom show the front view and a side view of the fitting results, respectively.

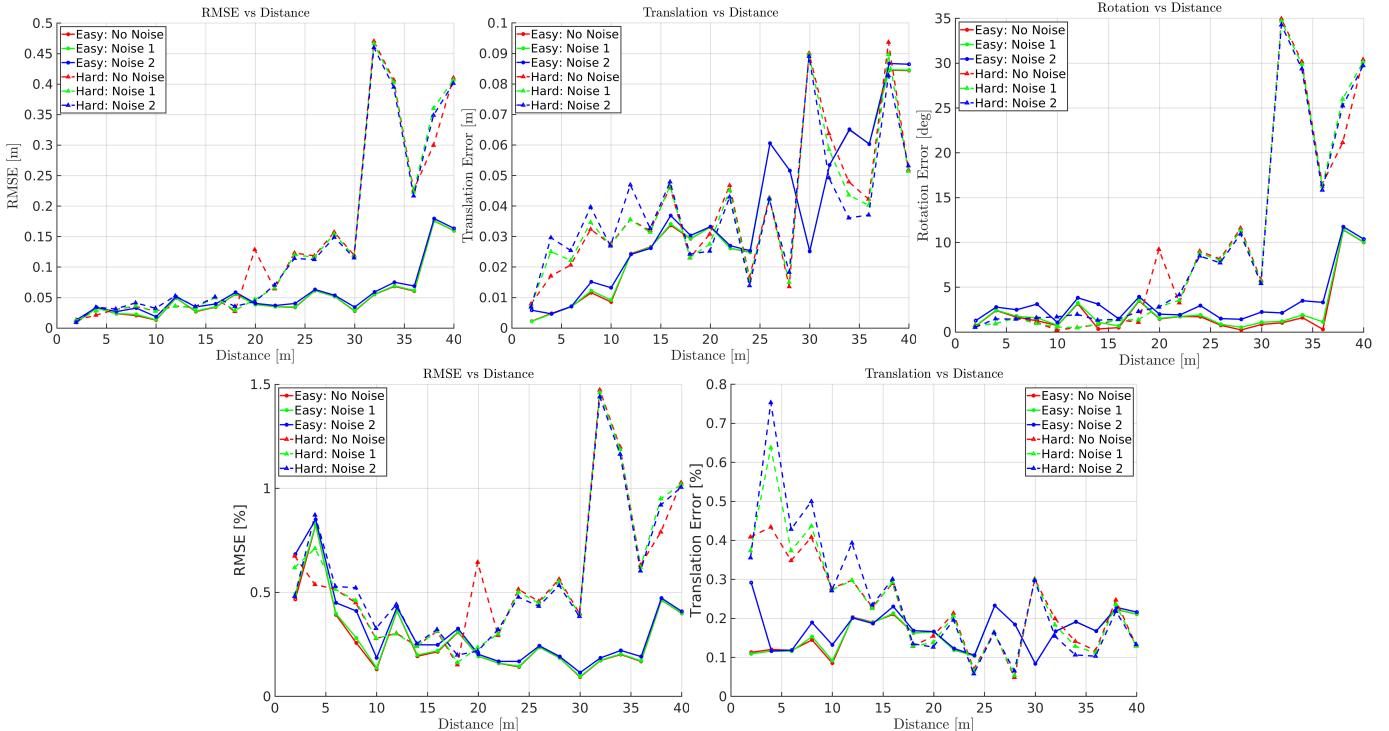


Fig. 10: Simulation results with a target placed at distances from 2 to 40 meters in 2 m increments in the LiDAR simulator. At each distance, the simulation data are collected with the target face-on to the LiDAR as an easy case (solid line), and for the other, the target is rotated by the Euler angle (roll = 20°, pitch = 30°, yaw = 30°) as a challenging case (dashed line). In addition, we induce two different levels of noises to each dataset, as indicated by the different colors.



Fig. 11: The sensor suite consists of a LiDAR, a camera and several motion capture markers.

detect the shape in both images and LiDAR point clouds. If a dataset has been collected and labeled, automatic detection using deep-learning architectures is also an exciting future direction. Currently, the proposed algorithm assumes the point cloud has been motion compensated; how to adopt motion distortion into the algorithm is another direction for future work. Applying it as a fiducial marker system or as an automatic calibration system also offers another interesting area for further research.

ACKNOWLEDGMENT

Toyota Research Institute provided funds to support this work.

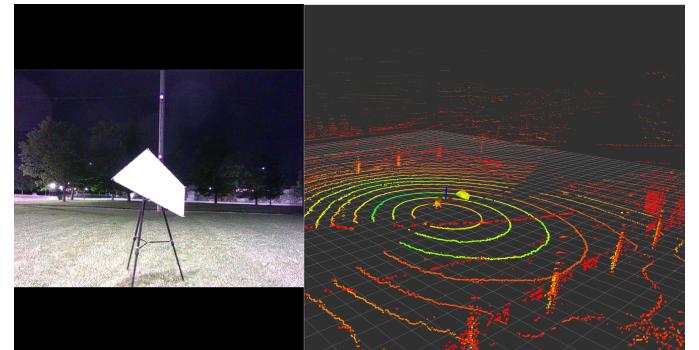


Fig. 12: The Experimental setup. The left shows passive markers are attached to the four corners of the optimized target shape and the right shows a LiDAR scan in M-Air.

Funding for J. Grizzle was in part provided by NSF Award No. 1808051 and 2118818. The first author thanks Wonhui Kim for useful conversations.

REFERENCES

- [1] J. K. Huang, S. Wang, M. Ghaffari, and J. W. Grizzle, “LiDARTag: A Real-Time Fiducial Tag System for Point Clouds,” *IEEE Robotics and Automation Letters*, pp. 1–1, 2021.
- [2] Jiunn-Kai Huang, Shoutian Wang, Maani Ghaffari, and Jessy W. Grizzle, “LiDARTag: A real-time fiducial tag using point clouds,” *arXiv preprint arXiv:1908.10349*, 2020.
- [3] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *Proc. IEEE Int. Conf. Robot. and Automation*. IEEE, 2011, pp. 3400–3407.
- [4] D. Wagner and D. Schmalstieg, “Artoolkit on the pocketpc platform,” in *IEEE International Augmented Reality Toolkit Workshop*. IEEE, 2003, pp. 14–15.

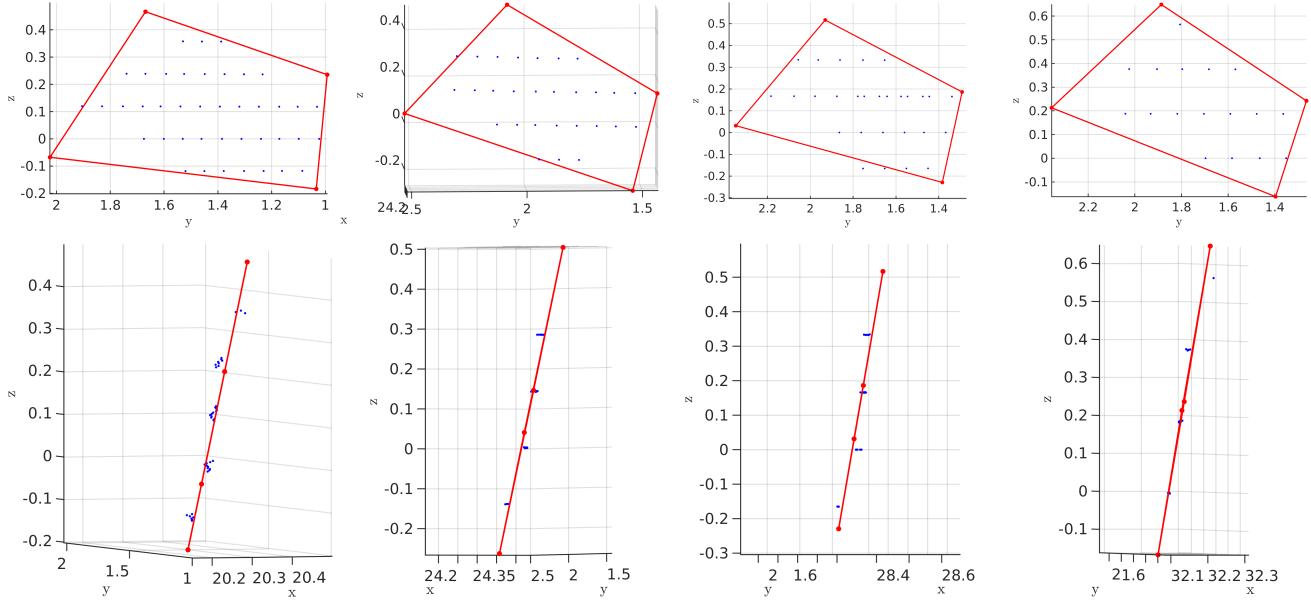


Fig. 13: Fitting results of the optimal shape at various distances (20, 24, 28, 32 meters) in the atrium of the Ford Robotics Building at the University of Michigan. The blue dots are the LiDAR returns on the target and the red frame are the fitting results. The top and bottom show the front view and a side view of the results, respectively.

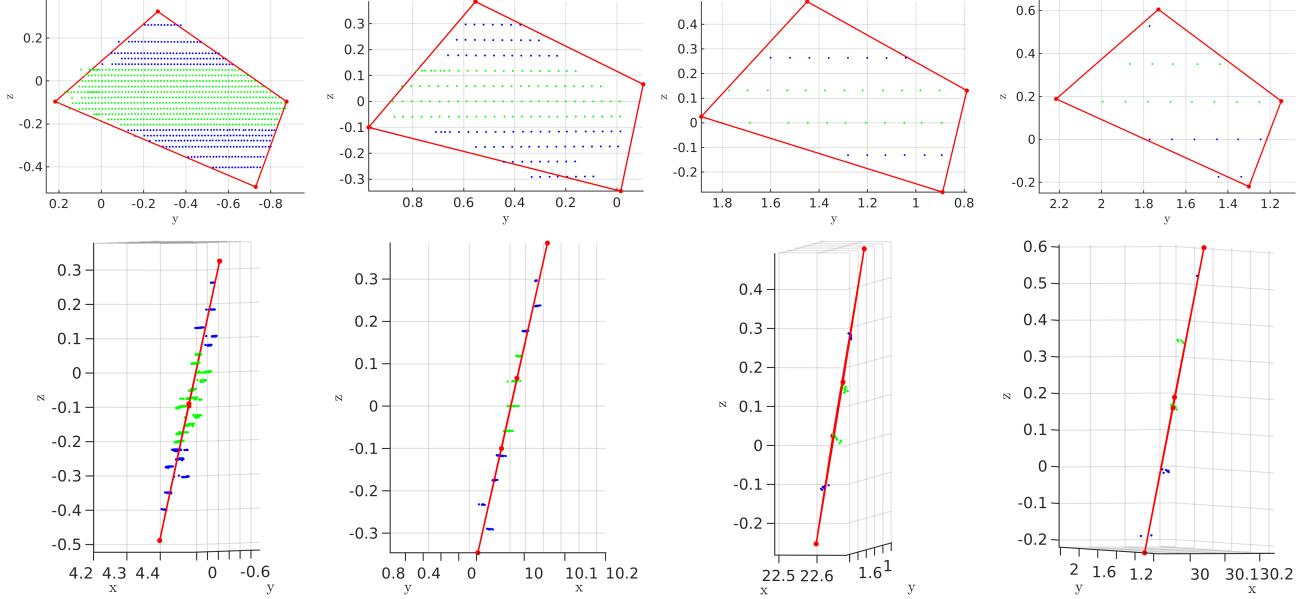


Fig. 14: Fitting results of the partially illuminated target at various distances (4, 10, 22, 30 meters) in the atrium of the Ford Robotics Building at the University of Michigan. The selected distances are different from Fig. 13 to show more results. The red frames are the fitting results. The blue dots are the LiDAR returns on the targets while the green dots are considered missing. The top and bottom show the front view and a side view of the fitting results, respectively.

- [5] M. Fiala, “Artag, a fiducial marker system using digital techniques,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 2. IEEE, 2005, pp. 590–596.
- [6] J. DeGol, T. Bretl, and D. Hoiem, “Chromatag: a colored marker and fast detection algorithm,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1472–1481.
- [7] B. Atcheson, F. Heide, and W. Heidrich, “Caltag: High precision fiducial markers for camera calibration.” in *VMV*, vol. 10. Citeseer, 2010, pp. 41–48.
- [8] M. Fiala, “Comparing artag and artoolkit plus fiducial marker systems,” in *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*. IEEE, 2005, pp. 6–pp.
- [9] B. Wang, “Lftag: A scalable visual fiducial system with low spatial frequency,” *arXiv preprint arXiv:2006.00842*, 2020.
- [10] J.-K. Huang, C. Feng, M. Achar, M. Ghaffari, and J. W. Grizzle, “Global Unifying Intrinsic Calibration for Spinning and Solid-State LiDARs,” *arXiv preprint arXiv:2012.03321*, 2020.
- [11] J. Huang and J. W. Grizzle, “Improvements to Target-Based 3D LiDAR to Camera Calibration,” *IEEE Access*, vol. 8, pp. 134 101–134 110, 2020.
- [12] Q. Liao, Z. Chen, Y. Liu, Z. Wang, and M. Liu, “Extrinsic calibration of LiDAR and camera with polygon,” in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 200–205.
- [13] 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.
- [14] X. Gong, Y. Lin, and J. Liu, “3D LiDAR-camera extrinsic calibration

TABLE II: Pose and vertex accuracy of the experimental results. The ground truth is provided by a motion capture system with 33 cameras. The quantization error e_q is the distance between two adjacent points on the same ring. The translation error e_t in meters and rotation error e_r in degrees are computed by (20).

Face-on LiDAR							
Distance [m]	No. Points	e_q [m]	RMSE [m]	RMSE [%]	e_t [m]	e_t [%]	e_r [deg]
2.1487	3009	0.015	0.026	1.193	0.024	1.133	0.619
4.0363	1120	0.028	0.029	0.718	0.024	0.603	1.906
6.0877	527	0.042	0.033	0.544	0.03	0.5	1.426
7.9145	589	0.055	0.033	0.413	0.03	0.384	1.765
10.093	197	0.07	0.034	0.334	0.03	0.294	1.846
12.016	136	0.084	0.035	0.288	0.03	0.247	2.081
13.987	192	0.098	0.03	0.216	0.028	0.201	1.737
15.981	155	0.112	0.03	0.187	0.028	0.176	1.237
17.971	119	0.125	0.031	0.173	0.028	0.156	1.387
Extreme Angle (20, 30, 30)							
Distance [m]	No. Points	e_q [m]	RMSE [m]	RMSE [%]	e_t [m]	e_t [%]	e_r [deg]
2.0448	2708	0.014	0.048	2.364	0.027	1.304	4.577
4.1225	898	0.029	0.04	0.971	0.034	0.835	2.641
5.966	504	0.042	0.033	0.549	0.029	0.486	1.826
7.9452	276	0.055	0.052	0.656	0.031	0.395	4.633
9.9484	155	0.069	0.041	0.412	0.034	0.345	3.52
12.006	105	0.084	0.06	0.498	0.036	0.299	6.064
14.106	75	0.098	0.051	0.364	0.045	0.318	2.714
16.05	54	0.112	0.057	0.355	0.041	0.253	4.634
18.087	48	0.126	0.055	0.305	0.044	0.241	3.593

- using an arbitrary trihedron,” *Sensors*, vol. 13, no. 2, pp. 1902–1918, 2013.
- [15] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, “LiDAR-camera calibration using 3D-3D point correspondences,” *arXiv preprint arXiv:1705.09785*, 2017.
- [16] S. Verma, J. S. Berrio, S. Worrall, and E. Nebot, “Automatic extrinsic calibration between a camera and a 3D LiDAR using 3D point and plane correspondences,” *arXiv preprint arXiv:1904.12433*, 2019.
- [17] J. Jiao, Q. Liao, Y. Zhu, T. Liu, Y. Yu, R. Fan, L. Wang, and M. Liu, “A novel dual-LiDAR calibration algorithm using planar surfaces,” *arXiv preprint arXiv:1904.12116*, 2019.
- [18] E.-S. Kim and S.-Y. Park, “Extrinsic calibration of a camera-LiDAR multi sensor system using a planar chessboard,” in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2019, pp. 89–91.
- [19] 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)*. IEEE, 2017, pp. 1–6.
- [20] S. Mishra, G. Pandey, and S. Saripalli, “Extrinsic calibration of a 3d-lidar and a camera,” *arXiv preprint arXiv:2003.01213*, 2020.
- [21] B. Xue, J. Jiao, Y. Zhu, L. Zheng, D. Han, M. Liu, and R. Fan, “Automatic calibration of dual-lidars using two poles stickered with retro-reflective tape,” *arXiv preprint arXiv:1911.00635*, 2019.
- [22] M. Klopschitz and D. Schmalstieg, “Automatic reconstruction of wide-area fiducial marker models,” in *IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2007, pp. 71–74.
- [23] J. DeGol, T. Bretl, and D. Hoiem, “Improved structure from motion using fiducial marker matching,” in *Proc. European Conf. Comput. Vis.*, 2018, pp. 273–288.
- [24] L. Calvet, P. Gurdjos, C. Griwodz, and S. Gasparini, “Detection and accurate localization of circular fiducials under highly challenging conditions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 562–570.
- [25] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [26] J. Briales and J. Gonzalez-Jimenez, “Convex global 3d registration with lagrangian duality,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4960–4969.
- [27] J.K. Huang, C Feng and Jessy W. Grizzle, “LiDAR Simulator Package,” 2020. [Online]. Available: https://github.com/UMich-BipedLab/lidar_simulator
- [28] J.K. Huang and Jessy W. Grizzle, “Optimal Shape Design for LiDAR,” 2021. [Online]. Available: https://github.com/UMich-BipedLab/optimal_shape_generation.git
- [29] —, “Global Pose Estimation using Optimal Shape,” 2021. [Online]. Available: https://github.com/UMich-BipedLab/optimal_shape_global_pose_estimation.git
- [30] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision second edition*, 2000.
- [31] J. N. Cederberg, *A course in modern geometries*. Springer Science & Business Media, 2013.
- [32] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [33] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [34] D. Lenton, “Projective transformations in 2d,” 2019. [Online]. Available: <https://mc.ai/part-ii-projective-transformations-in-2d/>
- [35] Velodyne Lidar, “Velodyne Ultra Puck: VLP-32C User Manual,” 2019. [Online]. Available: <https://icave2.cse.buffalo.edu/resources/sensor-modeling/VLP32CManual.pdf>
- [36] R. Tron, D. M. Rosen, and L. Carlone, “On the inclusion of determinant constraints in lagrangian duality for 3d slam,” in *Robotics: Science and Systems (RSS) in the workshop “The Problem of Mobile Sensors*, 2015.
- [37] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, “Lagrangian duality in 3d slam: Verification techniques and optimal solutions,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 125–132.
- [38] C. Olsson and A. Eriksson, “Solving quadratically constrained geometrical problems using lagrangian duality,” in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–5.
- [39] M. Grant and S. Boyd, “Cvx: Matlab software for disciplined convex programming, version 2.1,” 2014.
- [40] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, 2009.
- [41] “M-air at the university of michigan, ann arbor,” 2018. [Online]. Available: <https://robotics.umich.edu/about/mair/>