

Simple Camera-to-2D-LiDAR Calibration Method for General Use^{*}

Andrew H. Palmer, Chris Peterson, Janelle Blankenburg, David Feil-Seifer, and
Monica Nicolescu

University of Nevada, Reno, Reno NV 89557, USA
ahpalmer@unr.edu;chrispeterson@nevada.unr.edu
jjblankenburg@nevada.unr.edu;dave@cse.unr.edu
monica@cse.unr.edu

Keywords: Extrinsic Calibration · 2D LiDAR · Optical Camera · Gradient Descent.

Abstract. As systems that utilize computer vision move into the public domain, methods of calibration need to become easier to use. Though multi-plane LiDAR systems have proven to be useful for vehicles and large robotic platforms, many smaller platforms and low-cost solutions still require 2D LiDAR combined with RGB cameras. Current methods of calibrating these sensors make assumptions about camera and laser placement and/or require complex calibration routines. In this paper we propose a new method of feature correspondence in the two sensors and an optimization method capable of using a calibration target with unknown lengths in its geometry. Our system is designed with an inexperienced layperson as the intended user, which has led us to remove as many assumptions about both the target and laser as possible. We show that our system is capable of calibrating the 2-sensor system from a single sample in configurations other methods are unable to handle.

1 INTRODUCTION

Multi-plane LiDAR works for vehicle platforms due to their ability to carry around the computers capable of processing the data quickly; however, other platforms do not have the same capabilities. Searching through a large point cloud is costly in terms of processing time. For these less capable systems, using a 2D LiDAR with an optical camera gets both the detection and distance information a robot needs to navigate and interact with the world.

Calibration is key to determining which points in the image correspond to which indices in the laser. Several systems, such as [1, 7, 16, 18], exist to derive these transformation values. However, when it comes to applications laypeople are capable of using, these methods require exact target geometries or multiple samples (which take time to position). Some methods, such as [3, 11, 12], have been designed to make calibration easier by either limiting the number of samples

^{*} This work has been supported by NSF Awards #IIS-1719027 and #IIS-1757929.

to one or two, using simpler target shapes, or allowing for wider varieties of configurations between the camera and laser. All of these are improvements over prior systems, yet they too lack the simplicity a calibration system meant for non-experts requires.

A system designed around the ease of use for non-experts should at least have the following traits: 1) the calibration target should be easy to construct and detect, 2) the system should be able to find transformation values from only a single sample, and 3) the system should be able to handle cases where the camera and laser are not aligned in the same relative direction. Current systems of calibrating a 2D LiDAR and an intrinsically calibrated optical camera do not achieve all three of these properties. Therefore, developing a user-friendly system to find the calibration between these sensors is necessary.

The rest of the paper is structured as follows. Section 2 provides background information; Section 3 discusses our target geometry and setup; Section 4 discusses the derivatives used in our method; Section 5 discusses our simulation setup; Section 6 discusses our experimental results; Section 7 discusses our results and future work; and Section 8 presents our conclusions.

2 Background

To perform a calibration, two elements are needed: detected points of correspondence between sensors and an optimization approach that computes a geometric transformation between these points. Determining this correspondence has led to a vast array of calibration targets with specialized properties.

Prior methods for obtaining points of correspondence between an optical camera and 2D LiDAR such as [1, 7, 16, 18] have relied on detecting checkerboards in images and lines in the laser reading. However, with such flat surfaces detecting displacement in the Z-axis is difficult. This led to methods using trihedrons to determine at what height the laser intersects the target [5, 8, 12, 14]. However, there are rotational problems with these targets that led to methods like [3]. That method also has the benefit of calibrating from a single sample. Unfortunately, these prior methods rely on the laser and camera to have relatively the same forward-facing view of the target, which limits the use of those for contexts where the camera and laser have significantly different views of the target.

Once the correspondence is known, methods such as [10, 13, 17] are used to find the optimal

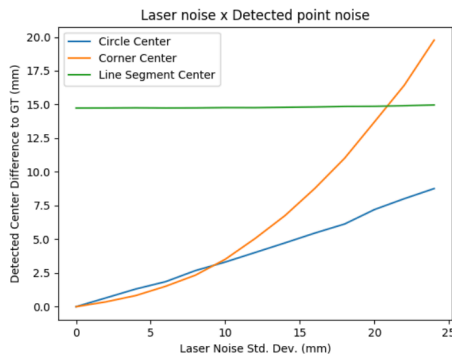


Fig. 1: Comparison of laser noise to detections.

transformation parameters. Most methods of calibration have used either the Levenberg-Marquardt optimization [13] or EPnP [10] after data has been cleaned up with RANSAC [4]. This leads to many systems of calibration requiring multiple samples.

To understand how different target shapes can be detected in the laser, we examined the noise in the laser sensor and its effect on the detection of centers of line segments, corners, and circle centers. We examined 13 noise settings from 0 to 24 mm in increments of 2 mm. This noise was applied to the sensor depths as a Gaussian noise. Figure 1 shows the results of this analysis. 1000 samples were collected at each noise level. As is evident from Figure 1, circle fitting is more robust at high levels of noise. This suggests that circle fitting is preferred for setups in large areas.

3 Setup

Our setup for finding a calibration in the real world uses a target made out of spheres. We color the spheres to stand out against the background and show an image with correspondence assigned to the target for validation by the user. Currently we manually place the spheres at laser height. A sphere allows us to calibrate with viewing directions where the camera is on opposite sides of the target from the laser. Our set up in the real world is shown in Figure 2a.

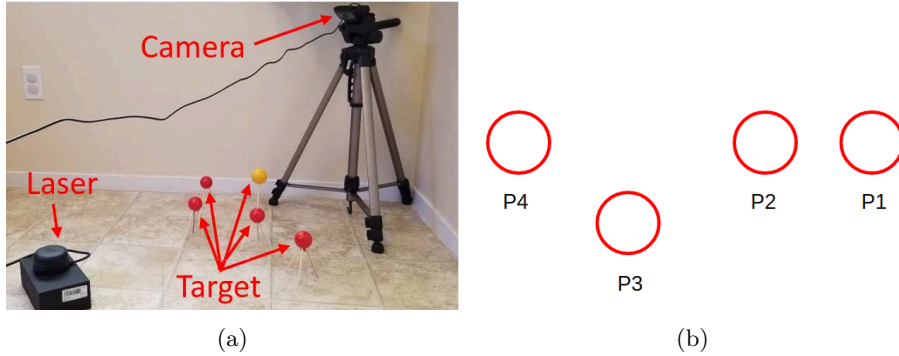


Fig. 2: (a) A real world setup of our laser, camera, and target. (b) Top down view of calibration target shape.

To have a target with unknown dimensions, we needed to define a shape that could be detected from relative positions alone. We defined our target as follows. Points 1, 2, and 4 lie on a line within the plane. Point 1 is closer than point 4 to point 2. Point 3 resides outside of the line formed by the other points, and its projection onto that line resides between points 2 and 4. A top down of points 1, 2, 3, and 4 is shown in Figure 2b. Point 5 has the same (x, y) position as point 2 with a displacement in the Z -axis.

Because we did not want to rely on fixed dimensions for the target, we had to develop a method to estimate the height of point 5 for the purposes of both the camera projection and the laser frame. We cover this process in Section 4.

Our system can handle undefined lengths for the distances between target points; thus, it is possible to make a target out of common household materials such as ping pong balls, skewers, and paint. To detect the target, we determine which point corresponds to point 2 and then determine points 1, 3, and 4 based on the geometry of the target. Point 5 is above point 2; for the laser this makes it easier to determine the (x, y) location of point 5. To make it simple to detect the non-planar point, we used a different color as our identifier for point 5.

We also rotated the camera around the Z axis to have it face back toward the laser when the random initial guess put the camera in front of the laser. For cases with the camera behind the target this second initial guess should reduce the burden of modifying the rotation components.

The Euclidean distance is the best error metric suited to our method because the camera points should be perfectly projected onto the laser plane at the detected laser locations with correct transform values.

4 Derivations

Due to the unknown Z -coordinate of point 5, choosing gradient descent allows us to update an estimate at each iteration. To derive this gradient, we need to project the points onto the laser plane from the camera given a transformation matrix. The projection and rotation between the frames is demonstrated in Figure 3a. From there we can find the error between those points and the points detected in the laser frame and use the error to find the partial derivatives of the transformation matrix.

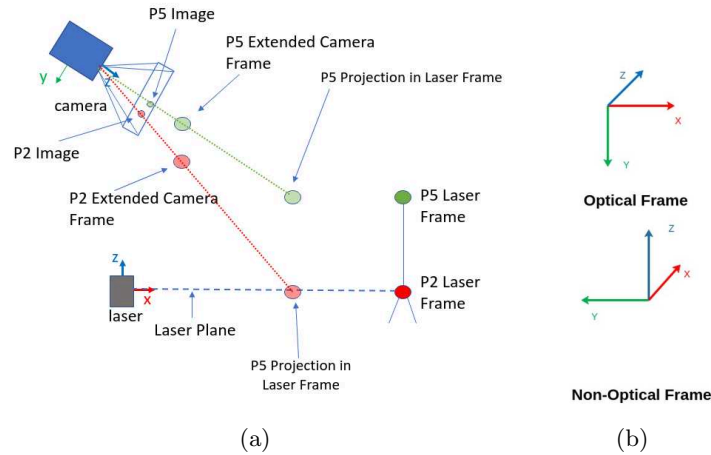


Fig. 3: (a) Diagram of camera and laser with projections and detections of point 2 and point 5 into the laser frame. (b) Axis representations of optical and non-optical frames.

The transformation matrix between the coordinate systems of the camera and the laser is shown in equation 1. The upper left nine values, $r_{11}, r_{12}, \dots, r_{33}$,

allow us to rotate the point around the center of the camera frame before we translate by the the translation values, t_x, t_y , and t_z . We have to include a fourth element, 1, in our 3D points to allow us to multiply the matrix to the point. In equation 1, cp is an extended point in the camera frame (P5 Projection Camera Frame in Figure 3a) and lp is a projection into the laser frame (P5 Projection in Laser Frame in Figure 3a).

To get a 3D point for transformation, we take a target detection camera pixel and extend it forward from the image plane to get an x, y , and z location in the camera frame. This is done by applying the inverse projection matrix defined by the camera properties found prior to calibration. Applying the inverse is shown in equation 2. The pixel locations are specified by i_x, i_y . The f_x, f_y correspond to the focal distance

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{cp} \\ y_{cp} \\ z_{cp} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{lp} \\ y_{lp} \\ z_{lp} \\ 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} i_x \\ i_y \\ 1 \end{bmatrix} = \begin{bmatrix} x_{cp} \\ y_{cp} \\ z_{cp} \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} x_{lp} \\ y_{lp} \\ z_{lp} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11}x_{cp} + r_{12}y_{cp} + r_{13}z_{cp} + t_x \\ r_{21}x_{cp} + r_{22}y_{cp} + r_{23}z_{cp} + t_y \\ r_{31}x_{cp} + r_{32}y_{cp} + r_{33}z_{cp} + t_z \\ 1 \end{bmatrix} \quad (3)$$

in pixels with respect to the x and y directions of the image. The p_x, p_y represent the principal point of the image, most often the center of the image.

The first matrix in equation 2 applies a rotation to adjust our axis representations from the optical frame to the non-optical frame. The rotation is shown in Figure 3b. This rotation transforms the X axis into the negative Y axis, the Y axis into the negative Z axis, and the Z axis into the X axis. We can examine the correctness of the transformation matrix once the points have been extended into 3D and transformed into the laser frame. The transformation is restated differently in equation 3 to demonstrate how the projection onto the laser plane functions.

The camera center, when transformed to $[x_c, y_c, z_c, 1] \rightarrow [t_x, t_y, t_z, 1]$, resolves to the camera position in the laser frame, or just the translation values. Forming a line through the camera center and the transformed extended camera point we can find the projection onto the laser plane as seen in Figure 3a. The changes of x, y , and z for each point relative to the camera center are shown in equations 4, 5, and 6. This is a vector from the camera center to the transformed extended point. χ, γ , and ζ represent the changes in the three axes directions. The line can be described by x and y in terms of z . This is shown in the equations in 7, where we solved for the x and y values of the intersection, X_i and Y_i , of the line with the plane.

$$\chi = x_{lp} - x_c = r_{11}x_{cp} + r_{12}y_{cp} + r_{13}z_{cp} \quad (4)$$

$$\gamma = y_{lp} - y_c = r_{21}x_{cp} + r_{22}y_{cp} + r_{23}z_{cp} \quad (5)$$

$$\zeta = z_{lp} - z_c = r_{31}x_{cp} + r_{32}y_{cp} + r_{33}z_{cp} \quad (6)$$

As stated above, we use the Euclidean distance between the projection onto the plane and the laser detection as the value of our error. This is shown in equation 8, where X_i, Y_i , and Z_i are the camera projection points.

The $Z_L - Z_i$ component of the error is only relevant for point 5 due to the other points lying within the laser plane. To solve for the point 5 un-

$$X_i = t_x - \frac{\chi}{\zeta} t_z \quad Y_i = t_y - \frac{\gamma}{\zeta} t_z \quad (7)$$

known z value, we take the angle between point 5 and point 2 and project this angle onto the Z -axis above/below point 2. To get a difference in the Z component difference of the error, we need a laser estimate and a projection estimate. We get this by picking the position of point 2 for the laser and then the projected camera location respectively. To get these estimations we take the angle between points 2 and 5 extended camera points, shown in equation 9. This comes from solving for the angle between two vectors.

The angle between the Z axis and point 2 is shown in 10. We subtract from π because the angle we are looking for is the supplementary angle.

We get the third angle of the triangle by equation 11. Using the Law of Sines we can solve for the point 5 Z shown in equation 12. The distance from the camera center to point 2 is d_1 . We can call these z estimations using point 2 from the laser detection and from camera projection z_{lp5} and z_{cp5} respectively. We perform this estimation in each iteration of gradient descent.

$$\theta_1 = \cos^{-1} \left(\frac{p_5 \cdot p_2}{\|p_5\| \|p_2\|} \right) \quad (9)$$

$$\theta_2 = \pi - \cos^{-1} \left(\frac{[0, 0, 1] \cdot p_2}{\|p_2\|} \right) \quad (10)$$

$$\theta_3 = \pi - (\theta_1 + \theta_2) \quad (11)$$

$$z_{p5} = \frac{\sin(\theta_1) d_1}{\sin(\theta_3)} \quad (12)$$

The (x, y) position of point 5 from the laser detection does not change at each iteration, but the position of point 5 from the camera projection completely changes each iteration. Thus, we also update point 5's x, y , and z coordinates at each iteration, as shown in equation 13.

The projected location of point 5 is established by replacing the estimated height into the equations of x and y in terms of z for the projected line.

All of the above leads to an expanded error equation, shown in 14.

$$p_5 = \begin{bmatrix} z_{cp5} \frac{\chi_5}{\zeta_5} + x_{ip5} \\ z_{cp5} \frac{\gamma_5}{\zeta_5} + y_{ip5} \\ z_{cp5} \\ 1 \end{bmatrix} \quad (13)$$

$$e = \sqrt{\left(X_L - \left(t_x - \frac{\chi}{\zeta} t_z \right) \right)^2 + \left(Y_L - \left(t_y - \frac{\gamma}{\zeta} t_z \right) \right)^2 + (z_l - z_c)^2} \quad (14)$$

We solve for the derivatives in steps. First, we find the changes in distance to the camera, d_1 , shown in 15.

$$\frac{\partial d_1}{\partial t_x} = \frac{-(X_{p2l} - t_x)}{d_1} \quad \frac{\partial d_1}{\partial t_y} = \frac{-(Y_{p2l} - t_y)}{d_1} \quad \frac{\partial d_1}{\partial t_z} = \frac{-(Z_{p2l} - t_z)}{d_1} \quad (15)$$

However, because the height of the laser detection of point 2 is 0, the t_z equation of 15 becomes 16. Since there is no z component to any of the other points except point 5, we can get the derivatives of z_{p5} , shown in 17.

$$\frac{\partial d_1}{\partial t_z} = \frac{t_z}{d_1} \quad (16)$$

$$\frac{\partial z_{p5}}{\partial t_x} = z_{p5} \frac{-(X_{p2l} - t_x)}{d_1^2} \quad \frac{\partial z_{p5}}{\partial t_y} = z_{p5} \frac{-(Y_{p2l} - t_y)}{d_1^2} \quad \frac{\partial z_{p5}}{\partial t_z} = z_{p5} \frac{t_z}{d_1^2} \quad (17)$$

Now that we have finished the first two sets of derivatives, we can take the derivatives of the error with respect to the translation. To make the equations slightly more readable, we use the convention $\bar{X} = X_L - X_i$ and $\bar{Y} = Y_L - Y_i$. We get the partial derivatives of the translation elements as in 18, 19, and 20.

$$\frac{\partial e}{\partial t_x} = \frac{-\bar{X}}{e} + \frac{z_l - z_c}{e} \left(\frac{-z_l(X_{p2l} - t_x)}{d_{1l}^2} - \frac{-z_c(X_{p2c} - t_x)}{d_{1c}^2} \right) \quad (18)$$

$$\frac{\partial e}{\partial t_y} = \frac{-\bar{Y}}{e} + \frac{z_l - z_c}{e} \left(\frac{-z_l(Y_{p2l} - t_y)}{d_{1l}^2} - \frac{-z_c(Y_{p2c} - t_y)}{d_{1c}^2} \right) \quad (19)$$

$$\frac{\partial e}{\partial t_z} = - \left(\frac{\chi \frac{\partial e}{\partial t_x} + \gamma \frac{\partial e}{\partial t_y}}{\zeta} \right) + \frac{z_l - z_c}{e} \left(\frac{z_l t_z}{d_{1l}^2} - \frac{z_c t_z}{d_{1c}^2} \right) \quad (20)$$

Knowing that the height is 0 for the points 1-4, we can change 18, 19, and 20 to the equations in 21 for all points other than point 5.

$$\frac{\partial e}{\partial t_x} = \frac{-\bar{X}}{e} \quad \frac{\partial e}{\partial t_y} = \frac{-\bar{Y}}{e} \quad \frac{\partial e}{\partial t_z} = - \left(\frac{\chi \frac{\partial e}{\partial t_x} + \gamma \frac{\partial e}{\partial t_y}}{\zeta} \right) \quad (21)$$

Solving for the derivatives of the vector components in the transformed extended points, we get 22 for χ , γ , and ζ respectively.

$$\frac{\partial e}{\partial \chi} = \frac{t_z \bar{X}}{\zeta e} \quad \frac{\partial e}{\partial \gamma} = \frac{t_z \bar{Y}}{\zeta e} \quad \frac{\partial e}{\partial \zeta} = \frac{-t_z}{\zeta^2 e} (\bar{X} \chi + \bar{Y} \gamma) \quad (22)$$

The remaining derivatives of the rotation components, shown in 23, are found using the vector component derivatives from 22.

$$\begin{aligned} \frac{\partial e}{\partial r_{11}} &= x_{cp} \frac{\partial e}{\partial \chi} & \frac{\partial e}{\partial r_{12}} &= y_{cp} \frac{\partial e}{\partial \chi} & \frac{\partial e}{\partial r_{13}} &= z_{cp} \frac{\partial e}{\partial \chi} \\ \frac{\partial e}{\partial r_{21}} &= x_{cp} \frac{\partial e}{\partial \gamma} & \frac{\partial e}{\partial r_{22}} &= y_{cp} \frac{\partial e}{\partial \gamma} & \frac{\partial e}{\partial r_{23}} &= z_{cp} \frac{\partial e}{\partial \gamma} \\ \frac{\partial e}{\partial r_{31}} &= x_{cp} \frac{\partial e}{\partial \zeta} & \frac{\partial e}{\partial r_{32}} &= y_{cp} \frac{\partial e}{\partial \zeta} & \frac{\partial e}{\partial r_{33}} &= z_{cp} \frac{\partial e}{\partial \zeta} \end{aligned} \quad (23)$$

This results in the gradient for our transformation matrix shown in 24.

This gradient is used to update our transformation matrix at each iteration in gradient descent. We augmented RMSProp [15] by adding a maximum history to the equation. This is represented by the indexing of the sum in equation 26. We chose 1000 for this max history value because it appeared to work well. We used 1×10^{-5} for the value of δ as a way to prevent division by zero. The ϵ term in equation 26 represents how much we either rely on the history or the new gradient for our expected value. We initially adjusted this as a confound and determined that 0.1 worked well.

$$\nabla F = \begin{bmatrix} \frac{\partial e}{\partial r_{11}} & \frac{\partial e}{\partial r_{12}} & \frac{\partial e}{\partial r_{13}} & \frac{\partial e}{\partial t_x} \\ \frac{\partial e}{\partial r_{21}} & \frac{\partial e}{\partial r_{22}} & \frac{\partial e}{\partial r_{23}} & \frac{\partial e}{\partial t_y} \\ \frac{\partial e}{\partial r_{31}} & \frac{\partial e}{\partial r_{32}} & \frac{\partial e}{\partial r_{33}} & \frac{\partial e}{\partial t_z} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (24)$$

$$F_{t+1} = F_t - \frac{\ell}{G} \nabla F \quad G = \sqrt{E[g^2]_t + \delta} \quad (25)$$

$$E[g^2]_t = \epsilon \sum_{i=t-1}^{t-n} \frac{g_i^2}{n} + (1 - \epsilon) |\nabla F|^2 \quad : n = \min(\text{history}, 1000) \quad (26)$$

Since all the points together make the sample, taking the average of the gradients over all the points per update is more meaningful. Thus, we average the gradients of all points together before updating. But the division by e in all the derivatives has the effect of normalizing the gradients. Therefore, if a single point is causing issues, say point 3, then all the other points could overshadow the significance of that point and keep the camera matrix from moving toward a smaller error. To avoid this, we introduce a weighted sum of the gradients for each point proportional to the magnitude of the error at that point. This means that for $t_x, t_y, t_z, \chi, \gamma$, and ζ gradients we multiply by the error. This adjustment results in the updated gradients below, equations 27 - 30.

$$\frac{\partial e}{\partial t_x} = -\bar{X} + (z_l - z_c) \left(\frac{-z_l(X_{p2l} - t_x)}{d_{1l}^2} - \frac{-z_c(X_{p2c} - t_x)}{d_{1c}^2} \right) \quad (27)$$

$$\frac{\partial e}{\partial t_y} = -\bar{Y} + (z_l - z_c) \left(\frac{-z_l(Y_{p2l} - t_y)}{d_{1l}^2} - \frac{-z_c(Y_{p2c} - t_y)}{d_{1c}^2} \right) \quad (28)$$

$$\frac{\partial e}{\partial t_z} = - \left(\frac{\chi \frac{\partial e}{\partial t_x} + \gamma \frac{\partial e}{\partial t_y}}{\zeta} \right) + (z_l - z_c) \left(\frac{z_l t_z}{d_{1l}^2} - \frac{z_c t_z}{d_{1c}^2} \right) \quad (29)$$

$$\frac{\partial e}{\partial \chi} = \frac{t_z \bar{X}}{\zeta} \quad \frac{\partial e}{\partial \gamma} = \frac{t_z \bar{Y}}{\zeta} \quad \frac{\partial e}{\partial \zeta} = \frac{-t_z}{\zeta^2} (\bar{X} \chi + \bar{Y} \gamma) \quad (30)$$

The same simplification for points 1-4 can be applied to 27, 28, and 29 as they were above to get 21 from 18, 19, and 20 respectively.

During the averaging process of all the points we need to also divide by the sum of errors in addition to the number of points. This correction for a weighted sum provides a much faster convergence.

5 Simulated Environment

We used Gazebo [9] to simulate an environment where we generated different target configurations and camera positions. Figure 4 shows the base target used for points 1, 3, and 4 in addition to the 4 different height settings between points 2 and 5.

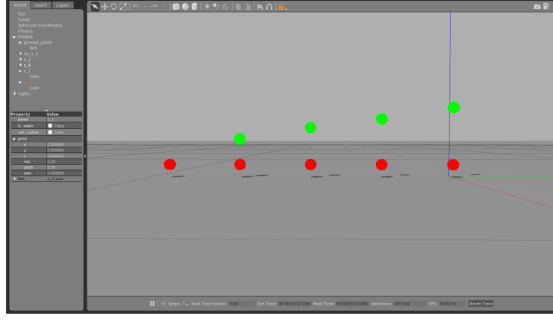


Fig. 4: Simulated targets with transparency. Left to right: base target, point 5 location at 9 cm, 13 cm, 16 cm, and 20 cm.

The target has colored spheres for visual detection differences between point 5 and the other points. The center of the sphere is the point of correspondence between the laser and the camera.

We tested orientations of the camera both below and above the laser plane. Each target configuration had random lengths within the target geometry. The height of point 5 was also randomly chosen from a pool

of models having either a 0.09, 0.13, 0.16, or 0.20 meter distance between points 2 and 5. We randomly placed point 3 on either side of the 1-2-4 points line.

We checked each target placement and configuration to see if the laser could get 4 detected points and the camera 5 detected points. In the same way a user can make sure the real world detections matched in the sensor display, we checked to see if the assignments matched the actual target when projected with the ground truth transformation. If the assignments did not match we generated a new configuration and placement. As soon as both sensors had a detection of the target, we ran our gradient descent method on the detected points. The laser stayed stationary at the origin across all configurations. We generated new target configurations and positions as well as camera locations for each group of samples.

The camera positions were random within a rectangular prism of size 5.5m x 3m x 4m centered on a point 1.25m in front of the laser to generate possible positions behind the target. The orientation of our camera was semi-random. We oriented the camera toward the target in each configuration.

We measured the distance of our resulting transformation matrix to our ground truth (GT) by the Euclidean distance. To evaluate the difference in angle, we used the chord distance to evaluate the difference in rotation angle defined in [6].

$$\theta = 2\sin^{-1}\left(\frac{1}{2\sqrt{2}}\|\hat{R} - R_{GT}\|_{\mathcal{F}}\right) \quad (31)$$

The chord distance is shown in Equation 31, where $\hat{R}$ is the found rotational matrix, R_{GT} is the rotation matrix of the ground truth, and $\mathcal{F}$ refers to the Frobenius Norm, or Euclidean Norm, of the matrices. We applied a Gaussian random value to the detected centers in each sensor to shift their locations in the image and the laser.

We also compared our method to the output from three other methods of solving the PnP problem implemented in OpenCV [2]; the Levenberg-Marquardt optimization [13], EPnP [10], and P3P [17]. In order to enable comparisons with these algorithms, which require the height of point 5 to be known, we provide this information in our evaluation.

Each target configuration resulted in 32 samples of output, 4 camera noise settings, 4 laser noise settings, and point 5 height known/unknown.

6 Results

We tested our calibration method on a variety of different configurations; 118 unique camera-laser positions. Approximately 42.3% (50 configurations) had the camera positioned behind the target (on the opposite side of the target from the laser). With our results as consistent as they are, we have shown that our system can handle both forward- and rear-facing target calibration.

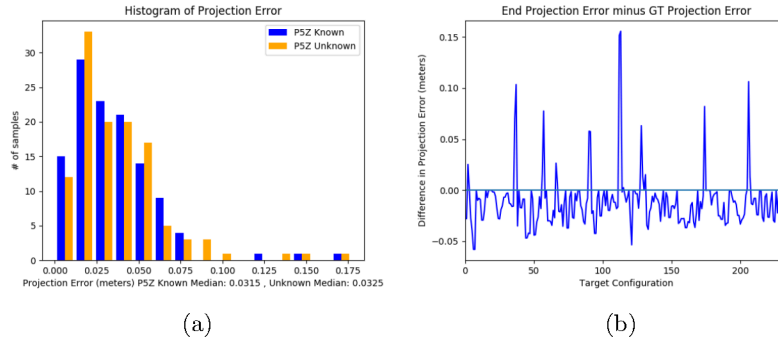


Fig. 5: (a) Histogram of projection errors. (b) Comparison of projection error and GT error per sample.

We examined the distribution of projection error for samples without noise in both point 5 Z known and unknown cases, Figure 5a. We can see from this histogram that the error values are centered on 0.018. There is no significant difference in the distributions of point 5 Z known and unknown.

We also compared the GT error to the resulting projection error of solutions from our method, Figure 5b. For most of the configurations we have negative values in this difference, with most found transformations having a smaller error than the GT. This is an example of how noise in the detection of point centers effects our projections.

Our method’s average distance to the GT is 0.0361 meters, with a median value of 0.0180 meters. The average rotational error is 0.7515 degrees, with a median value of 0.4149 degrees. Figure 6 shows the distribution of angular and distance to GT values.

We also examined how other methods implemented in OpenCV [2] compare in solving the Perspective-n-Point problem on our samples. The results are shown in Table 1. The correction for optical to non-optical rotations was applied to the other methods for comparison. When we look at the other methods of solving the Perspective-n-Point problem, it becomes clear that the OpenCV methods suffer greatly without the knowledge about point 5’s Z value. This issue is because they cannot use a point with partially unknown location in the world space.

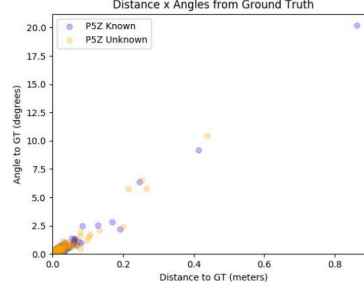


Fig. 6: Translation vs rotational error.

Method	No Point 5		Point 5	
	Translation (m)	Rotation (deg)	Translation (m)	Rotation (deg)
Levenberg-Marquardt	3.7858	122.11	1.2251×10^9	118.42
EPnP	1.0027×10^{92}	113.35	3.5279	112.98
P3P	3.4361	115.39	3.4003	115.24
Our Method	0.0361	0.7515	0.0396	0.8269

Table 1: Our method of solving the Perspective-n-Point problem compared to Levenberg-Marquardt [13], EPnP [10], and P3P [17] for both point 5 height known and unknown.

We evaluated our system with different levels of noise. We used 4 settings of camera Gaussian noise to shift the camera detection locations in both x and y location: 0, 1, 2, and 3 pixels standard deviation. We also used 4 settings for laser Gaussian noise to shift the detected centers in both x and y : 0, 3, 6, and 9 mm standard deviation. The translation error is shown in Figure 7b, and the rotation error in Figure 7b.

Target	Known Result	SD	Unknown Result	SD
1 (9 cm)	0.000396 m	0.002251	-0.000098 m	0.002058
2 (13 cm)	-0.000023 m	0.002273	0.000066 m	0.002012
3 (16 cm)	-0.001080 m	0.003715	-0.001712 m	0.003464
4 (20 cm)	0.000577 m	0.001723	0.000491 m	0.001450

Table 2: Averages and standard deviation of point 5 height estimated error for each target with and without known height by the laser a priori.

Our method relies on estimating the height of point 5 during its iterative process. The average error and standard deviation for each target’s estimated

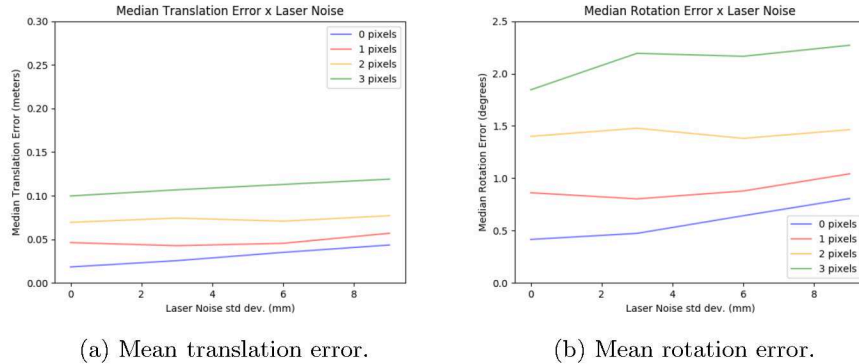


Fig. 7: Mean translation (a) and rotation (b) error for different modes of laser and camera noise.

height with height known and unknown in the laser frame points are shown in Table 2 (height of each target specified next to target number).



Fig. 8: Camera above the laser plane observing the calibration target from the front (a) and behind (b) with laser visualization applying our method’s calibration result.

Additionally, we tested our method on a real-world situation with colored ping-pong balls, kebab skewers, a Logitech C920 (resolution of 640 x 360), and a Slamtec RPLIDAR A3. The results for both front and back facing are shown in Figure 8. From these images we can see that our method is capable of calibrating in the real world in addition to a simulated environment.

7 Discussion and Future Work

The results from our real-world configurations show that our calibration system is capable of using everyday materials in a flexible geometry, which leads us to believe that users not skilled in calibration can reconstruct targets without feeling intimidated.

Through all configurations both in simulation and in the real world, our system found a calibration with a single image and laser sample. The above results indicate that each of the qualities that make a system user friendly were

achieved: 1) simple to construct target, 2) single sample calibration, and 3) diverse camera-to-laser positions.

One limitation for our calibration process is a camera position that places the camera at the same height as the laser plane. This results in an indistinguishable target assignment in the camera for the target. We plan to address this in the future.

Currently our system still requires the user to specify on which side of the laser plane the camera resides. In the future, we plan to add checks into the calibration process to remove this limitation.

In future work, we will add a graphical interface to allow the user to specify the colors of the spheres instead of defaulting to red-green or red-yellow.

8 Conclusions

We developed a novel routine for calibrating a camera and 2D LiDAR using gradient descent. In order to find a gradient capable of adjusting the camera's height, we proposed a method of estimating a point's height off the laser plane to keep the camera from finding a solution that places the image plane parallel to the laser plane. We also showed that our estimations are close to the actual ground truth height of that point. We showed that our method is robust to laser noise and to a lesser extent camera noise.

We developed a system that could solve for the transformation matrix using only a single sample and a target with unknown dimensions. We showed that our method is one of only three that is capable of generating calibration parameters from a single sample, and our method is one of two that can calibrate while not viewing the target from the same general direction. Most importantly, our method is the only one that can do both. In addition, we showed that our method works with everyday materials, making it easier for users to remake targets without a large degree of expertise.

References

1. Ahmad Yousef, K., Mohd, B., Al-Widyan, K., Hayajneh, T.: Extrinsic calibration of camera and 2d laser sensors without overlap. *Sensors* **17**(10), 2346 (Oct 2017). <https://doi.org/10.3390/s17102346>, <http://dx.doi.org/10.3390/s17102346>
2. Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000)
3. Dong, W., Isler, V.: A novel method for the extrinsic calibration of a 2d laser rangefinder and a camera. *IEEE Sensors Journal* **18**(10), 4200–4211 (2018)
4. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (Jun 1981). <https://doi.org/10.1145/358669.358692>, <https://doi.org/10.1145/358669.358692>
5. Gomez-Ojeda, R., Briaes, J., Fernandez-Moral, E., Gonzalez-Jimenez, J.: Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 3611–3616 (2015)

6. Hartley, R.I., Trampf, J., Dai, Y., Li, H.: Rotation averaging. *International Journal of Computer Vision* **103**, 267–305 (2012)
7. Hillemann, M., Jutzi, B.: Ucalmicel – unified intrinsic and extrinsic calibration of a multi-camera-system and a laserscanner. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* **IV-2/W3**, 17–24 (08 2017). <https://doi.org/10.5194/isprs-annals-IV-2-W3-17-2017>
8. Hu, Z., Li, Y., Li, N., Zhao, B.: Extrinsic calibration of 2-d laser rangefinder and camera from single shot based on minimal solution. *IEEE Transactions on Instrumentation and Measurement* **65**(4), 915–929 (2016)
9. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 2149–2154. Sendai, Japan (Sep 2004)
10. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnnp: An accurate $o(n)$ solution to the pnp problem. *International Journal Of Computer Vision* **81**, 155–166 (2009). <https://doi.org/10.1007/s11263-008-0152-6>, <http://infoscience.epfl.ch/record/160138>
11. Li, N., Hu, Z., Zhao, B.: Flexible extrinsic calibration of a camera and a two-dimensional laser rangefinder with a folding pattern. *Applied Optics* **55**, 2270 (03 2016). <https://doi.org/10.1364/AO.55.002270>
12. Li, Y., Hu, Z., Li, Z., Cai, Y., Sun, S., Zhou, J.: A single-shot pose estimation approach for a 2d laser rangefinder. *Measurement Science and Technology* **31**(2), 025105 (nov 2019). <https://doi.org/10.1088/1361-6501/ab455a>, <https://doi.org/10.1088/1361-6501/ab455a>
13. Moré, J.J.: The levenberg-marquardt algorithm: Implementation and theory. In: Watson, G.A. (ed.) *Numerical Analysis*. pp. 105–116. Springer Berlin Heidelberg, Berlin, Heidelberg (1978)
14. Tian, Z., Huang, Y., Zhu, F., Ma, Y.: The extrinsic calibration of area-scan camera and 2d laser rangefinder (lrf) using checkerboard trihedron. *IEEE Access* **8**, 36166–36179 (2020)
15. Tieleman, T., Hinton, G.: Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning* (2012)
16. Vasconcelos, F., Barreto, J.P., Nunes, U.: A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE transactions on pattern analysis and machine intelligence* **34** (01 2012). <https://doi.org/10.1109/TPAMI.2012.18>
17. Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, Hang-Fei Cheng: Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(8), 930–943 (2003)
18. Zhou, L.: A new minimal solution for the extrinsic calibration of a 2d lidar and a camera using three plane-line correspondences. *IEEE Sensors Journal* **14**(2), 442–454 (2014)