

# Automatic Calibration of a Range Sensor and Camera System

Hatem Alismaïl

L. Douglas Baker

Brett Browning

National Robotics Engineering Center,  
Robotics Institute, Carnegie Mellon University  
Pittsburgh, PA 15201 USA

{halismaïl, ldbapp, brettb}@cs.cmu.edu

<http://www.cs.cmu.edu/~halismaïl>

## Abstract

*We propose an automated method to recover the full calibration parameters between a 3D range sensor and a monocular camera system. Our method is not only accurate and fully automated, but also relies on a simple calibration target consisting of a single circle. This allows the algorithm to be suitable for applications requiring in-situ calibration. We demonstrate the effectiveness of the algorithm on a camera-lidar system and show results on 3D mapping tasks.*

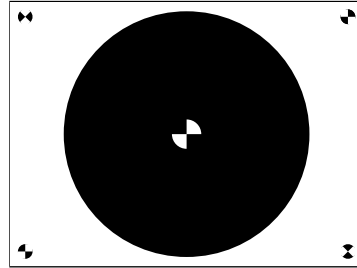


Figure 1. The calibration target used in this work. The fiducial marker at the center marks the circle’s physical center. The other fiducial markers are not used. The circle’s diameter is 33in printed on a standard  $48 \times 36\text{in}^2$  planar board.

## 1. Introduction

Combining cameras with an actuated range sensor is perhaps the most flexible and cost-effective approach to obtaining colorized 3D point clouds with low range noise. Such colorized point clouds provide valuable input to many 3D-based object detection, recognition and scene understanding algorithms [23]. Here, we propose a fully automated method to accurately calibrate the actuated lidar intrinsic parameters and extrinsic pose relationship between the actuated 3D range sensor and intrinsically calibrated camera(s). Contrary to several methods in the literature [5, 17, 15], our method relies on a simple calibration target that does not require a 3D target or a checkerboard pattern [31, 28, 10, 27, 22]. Checkerboards are known to cause range discrepancies for some lidars [14, 30], or algorithms using them rely on intensity/reflectance readings which may not be available or consistent.

Our 2D calibration target, shown in Fig. 1, is comprised of a single circle with a marked center. Using this target, we can obtain an accurate estimate of its supporting plane normal and location in 3D relative to the camera(s). Combining this with the segmented plane points in the lidar’s frame allows us to recover the calibration parameters within a standard point-to-plane optimization framework (i.e. a variant on Iterative

Closest Points [4], or a generalization thereof [26]).

In order to *fully* calibrate a camera-actuated lidar system, one has to estimate the *internal* and *external* calibration parameters. Here, we assume the camera intrinsic parameters are already calibrated (e.g. with the Matlab Camera Calibration Toolbox [3]). Thus, we focus on calibrating the actuated lidar “intrinsic” parameters, and the lidar to camera pose.

### 1.1. Internal Parameters

Internal parameters (or intrinsic) govern the internal workings of the 3D sensor. For instance, time-of-flight (ToF) cameras require the calibration of the camera projection matrix, and phase-delay latency [9]. A Velodyne requires recovering a scale, offset and the elevation angle for each of the rotating laser scanners. In this work, we focus on a line scanning lidar mounted on rotating platform [29]. The internal calibration amounts to estimating a rigid body transform between the instantaneous center of rotation of the lidar’s spinning mirror ( ${}^{\mathcal{L}}ICR$ ) and that of the platform motor ( ${}^{\mathcal{M}}ICR$ ). In the literature, this step is typically performed manually [12, 25], such that an objective/subjective criteria is optimized. In this work, we propose an automated method to recover these parameters in unstructured environments.

## 1.2. External Parameters

External parameters (extrinsic) relate the coordinate frame of the camera to the coordinate frame of the range sensor. Typically, a 6 DOF rigid body transformation. The literature on external calibration is extensive and several methods exist that can be categorized based on the range sensor employed and the method used to establish correspondences between camera imagery and range data. Since the laser spot is often invisible to the cameras of interest, establishing correspondences can be challenging.

Unnikrishnan and Hebert [27] initialize a non-linear minimization algorithm with manual correspondences. The user selects 3D points on a checkerboard calibration plane that is visible in the image and scanner. Pandey *et al.* [22] use a planar checkerboard as well. Their method, similar to Zhang and Pless' work [31], finds the calibration plane using a RANSAC-type framework and then finds the calibration parameters with respect to a 360° monocular camera. Scaramuzza *et al.* [24] do not rely on calibration targets and instead require user input to find the correspondences between an omni-directional camera and the range data. Aliakbarpour *et al.* [1] use an IMU to aid in establishing the correspondences and a light source to relate a stereo camera. While the IMU can be used to increase the robustness of correspondences, it may not be available and greatly reduces the applicability of the algorithm in a general setting. Furthermore, the use of the light source prevents the algorithm from operating outdoors such as may be required in-situ. Mirzaei *et al.* [19] simultaneously calibrate the internal and external sensor parameters, however, their algorithm is specific to the Velodyne. The convenience of circular calibration targets have been previously exploited by Guan *et al.* [11] to calibrate a network of camcorders and ToF cameras. Recent work by Geiger *et al.* [10] uses a fully automated method that is suitable for a variety of range sensors and cameras using a single shot of multiple checkerboard targets distributed around the sensor. This requires the placement of several targets around the scene (hence it is equivalent to multiple position-varied shots of a single target), and relies on corner detection which may not be accurate if the target exhibits strong reflections. Their algorithm also requires intensity information from the range sensor, which may not be readily available or reliable.

In contrast, our algorithm is fully automated and relies on a very simple target, and does not require intensity data. We do not rely on corner extraction, which may not be accurate as we show next. Finally, since the algorithm relies on ICP point-to-plane, it is robust to data sparsity from the range sensor. Missing data is not uncommon as the density of the point clouds is affected by the scanning geometry and the internal settings of the sensor.

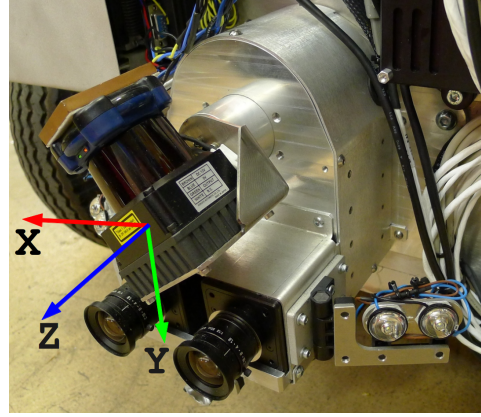


Figure 2. Camera-lidar sensor assembly. The calibration is between the rotating Hokuyo and the left camera (no stereo).

## 2. Sensor & Coordinate Systems

Our camera-lidar setup (shown in Fig. 2) consists of a line scan lidar (a Hokuyo or Sick LMS series) mounted on a spinning or nodding motor. Beneath the scanner, is a 1M pixel stereo camera. In this work, we use the left camera only. The sensor is mounted on a tunnel/mine crawling robot to be used in tunnel mapping tasks. The lidar and the motor are assigned a right-handed coordinate system, with the Z-axis pointing forward (along the line of sight), and the Y-axis pointing downward.

We focus on the Hokuyo sensor, although the approach generalizes to other line scan lidars, multi-beam lidar systems such as the Velodyne, or other range sensors. The Hokuyo scanner covers a FOV of 270° via a rotating mirror about the Y-axis at the rate of 40Hz, with an angular resolution of 0.25° producing 1081 measurements per scan. The maximum guaranteed range is 30m (with 1mm resolution) as quoted by the manufacturer. The motor rotates about the Z-axis orthogonal to the lidar's mirror. Finally, the 1M pixel camera is grayscale with a FOV of  $\approx 90^\circ$  and assumed to be calibrated (*i.e.* known intrinsic matrix  $K$ , and optical distortion parameters). An example image from the camera is shown in Fig. 5.

## 3. Calibration of Internal Parameters

In order to triangulate 3D points from an actuated line scan lidar, the misalignment between  $\mathcal{L}ICR$  and  $\mathcal{M}ICR$  has to be estimated. Line scan lidar records raw measurements  $\rho_i$  in polar coordinates. These can be converted to Cartesian coordinates by applying the appropriate transform:

$$\mathcal{L}\mathbf{Y}_i = (x_i, y_i, z_i, 1)^T = \mathcal{L}\mathbf{R}^Y(\theta_i) (0, 0, s\rho_i, 1)^T, \quad (1)$$

where  $\theta_i$  is the measured mirror angle,  $s$  is a scaling factor to convert raw range into metric units, and  $\mathcal{L}\mathbf{R}^Y(\cdot)$  is a rotation matrix about the Y-axis in the lidar's right-handed coordinate

system  $\mathcal{L}$ . To obtain the final 3D coordinate of the point, we apply the internal calibration between  ${}^{\mathcal{L}}ICR$  and  ${}^{\mathcal{M}}ICR$ , followed by a rotation using the measured *motor* angle  $\phi_j$ :

$${}^{\mathcal{M}}\mathbf{X}_{ij} = {}^{\mathcal{M}}\mathbf{R}^Z(\phi_j) {}^{\mathcal{M}}\mathbf{H}_{\mathcal{L}} {}^{\mathcal{L}}\mathbf{Y}_i, \quad (2)$$

where  ${}^{\mathcal{M}}\mathbf{R}^Z(\cdot)$  is a right-handed rotation about the Z-axis in the motor frame  $\mathcal{M}$ , and  ${}^{\mathcal{M}}\mathbf{H}_{\mathcal{L}}$  is the internal calibration given by:

$${}^{\mathcal{M}}\mathbf{H}_{\mathcal{L}} = \begin{pmatrix} R_{3 \times 3} & \begin{matrix} t_x \\ t_y \\ t_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

The Z component of the translation  $t_z$  is set to zero as scaling along the line of sight does not affect the optimization.

As pointed out by Fairfield [7], the misalignment between  ${}^{\mathcal{L}}ICR$  and  ${}^{\mathcal{M}}ICR$  does not depend on the rotation rates of either the mirror's or the motor's. Furthermore, by rotating the scanner a full 360° from a static position, we obtain two sets of scans of the environment:

$$\mathcal{S}_1 = \{ {}^{\mathcal{M}}X_{ij} : \phi_j < \pi \}_j^N, \text{ and} \quad (4)$$

$$\mathcal{S}_2 = \{ {}^{\mathcal{M}}X_{ik} : \phi_k \geq \pi \}_k^M. \quad (5)$$

The two sets of points scan the same surfaces in space and points should lie on these surfaces. In practice, the two sets do not align perfectly (up to the local tangent plane) due to possible non-constant acceleration of the rotating motor, differences between scan rate and rotation rate, as well as the mechanical misalignment we need to calibrate. To resolve this, we employ the point-to-plane ICP [4]. The internal calibration is such that:

$${}^{\mathcal{M}}\mathbf{H}_{\mathcal{L}} = \underset{\mathbf{H}}{\operatorname{argmin}} \left\{ \sum_i^{|S_1|} \sum_k^K \|\eta_k \cdot (X_i - \mathbf{H}Y_i^k)\|^2 \right\}, \quad (6)$$

where  $X_i \in \mathcal{S}_1$ ,  $Y_i^k \in \mathcal{S}_2$  is the set of closest neighbors to  $X_i$  from  $\mathcal{S}_2$ ,  $\eta_k$  is the plane normal at  $Y_i^k$ , and  $K$  is the size of the neighborhood defined by a fixed radius search with radius  $r$ . We solve this nonlinear problem iteratively using Levenberg-Marquardt (LM) [16, 18]. For the optimization, the rotation is parameterized using Euler angles. The algorithm is detailed in Alg. 1.

The most computationally intensive part of the algorithm is the search for neighbors. If the misalignment is not very large (as is commonly the case), we found that the neighbors and plane normals need not to be computed at every iteration. Instead, we run ICP for  $I$  iterations with the same set of normals before recalculating and repeating. Typically, the algorithm converges within a few iterations. We found that this improves the speed of the algorithm without jeopardizing the quality of the results. An example of the results we obtain is shown in Fig. 3.

```

Input:  $\mathcal{S}_1 = \{ {}^{\mathcal{M}}X(j) : \phi_j < \pi \}_j^N$ 
         $\mathcal{S}_2 = \{ {}^{\mathcal{M}}Y(m) : \phi_m \geq \pi \}_m^M$ 
         $r$  maximum radius to search for neighbors
Output:  ${}^{\mathcal{M}}\mathbf{H}_{\mathcal{L}}$  internal calibration transform
// Initialize
 ${}^{\mathcal{M}}\mathbf{H}_{\mathcal{L}} \leftarrow \mathbf{I}_{4 \times 4}$ ;
for  $i \leftarrow I$  do
    // Compute neighbors and normals
    for  $j \leftarrow |X|$  do
         $\{y\}_k^K \leftarrow \text{RadiusSearch}(X_j; \mathcal{S}_2, r)$ ;
         $Y_l^k \leftarrow \text{ClosestPoint}(X_j; \{y\})$ ;
         $\eta_k \leftarrow \text{EstimateNormal}(\{y\})$ ;
    end
    while not converged do
         ${}^{\mathcal{M}}\mathbf{H}_{\mathcal{L}} = \underset{\mathbf{H}}{\operatorname{argmin}} \left\{ \sum_j \sum_k \|\eta_k \cdot (X_j - \mathbf{H}Y_l^k)\|^2 \right\}$ 
    end
end

```

**Algorithm 1:** Spinning lidar internal calibration.

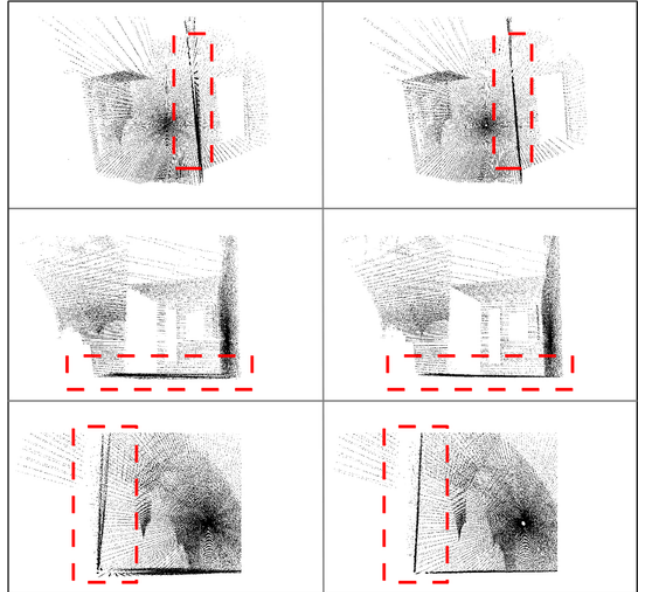


Figure 3. Example results of the internal calibration for a scan of room. The left column shows the result prior to calibration. Red boxes highlight the areas of improvement. Notice how the “double wall” effects due to misalignment have been eliminated post-calibration.

## 4. Calibration of External Parameters

It is well known that a circle in space projects onto an ellipse under colineation [20]. Using this knowledge, the circle parameters in space (center and supporting plane) can be recovered up to two-fold ambiguity [13, 8]. Using the derivation by Kanatani and Liu [13], let  $Q$  be the projection of a circle in space with radius  $r$  in normalized image coordinates. Let the Eigenvalues of  $Q$  be  $(\lambda_3 < 0 < \lambda_1 \leq \lambda_2)$  with corresponding Eigenvectors  $\mathbf{u}_3, \mathbf{u}_2$  and  $\mathbf{u}_1$ . The unit

normal of the circle’s supporting plane in space is given by:

$$\mathbf{n} = \mathbf{u}_2 \sin \theta \pm \mathbf{u}_3 \cos \theta, \quad (7)$$

where  $\sin \theta = \sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_2 - \lambda_3}}$  and  $\cos \theta = \sqrt{\frac{\lambda_1 - \lambda_3}{\lambda_2 - \lambda_3}}$ . The distance of the circle’s center from the camera is given by:

$$d = r \lambda_1^{3/2}. \quad (8)$$

Note that  $Q$  is scaled such that  $\det(Q) = -1$  to resolve the scale ambiguity. If  $\lambda_1 \neq \lambda_2$ , then two solutions exist and cannot be resolved without further information.

Given a method disambiguate the two solutions, we can use the recovered normals  $\mathbf{n}_i$  and centers of the circle  $\mathbf{c}_i$  with the correspondences 3D scans of the supporting plane  $P_i$  from several views to automatically calibrate the camera-lidar system. While it is possible to detect the location of the projected circle center  $\mathbf{m}$  in the image directly and then compute the normal ( $\mathbf{n} = Q\mathbf{m}$ ), we found low resolution images cause large biases when trying to localize corners accurately. This case is illustrated in Fig. 4.

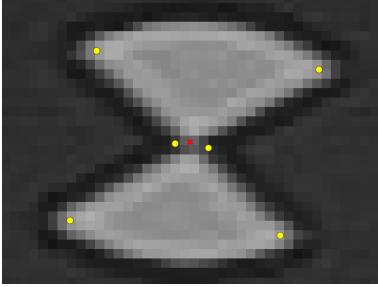


Figure 4. Accuracy of corner localization vs. estimating center from ellipse geometry. The figure shows a zoomed-in view of the marker at the center of the target. Circles show the location of sub-pixel refined Harris corners, while the cross shows the estimated location of the circle center from the ellipse geometry.

#### 4.1. Circle Pose from a Single View

In order to compute the circle’s pose from a single view, we need to detect its elliptical projection. For this task, we use the algorithm proposed by Ouellet and Hebert [21]. The algorithm typically finds several ellipses in the scene. By construction, the ellipse of interest is the largest and consists of mostly black pixels. Hence, it is easy to extract it by filtering the list of ellipses based on their size and the distribution of color in the interior. The two possible centers of the ellipse are tested and the one that is closest to the marker is selected as the correct center-normal pair. This is done by extracting a small patch ( $11 \times 11$  px<sup>2</sup>) around each projection and computing a zero-mean normalized correlation score with a template of the expected shape of the target’s center. Fig. 5 shows an example of ellipse detection.

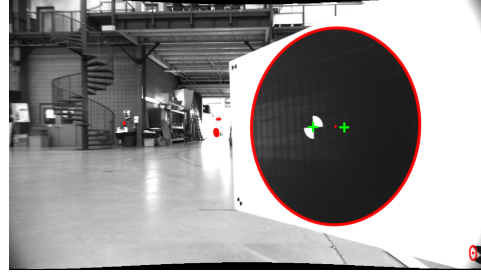


Figure 5. Example of ellipse detection. Detected ellipses are shown in red. The red dot shows the *ellipse* center, the green crosses show the two possible projections of the centers of the *circle*. The correct solution coincides with the marker on the target.

Given the correct projection of the circle’s center and the corresponding normal, the projection of the center in the image, and its location in 3D are given by:

$$\mathbf{m} \equiv Q^{-1}\mathbf{n}; \quad (9)$$

$$C = (d\mathbf{m}) / (\mathbf{n} \cdot \mathbf{m}). \quad (10)$$

#### 4.2. Calibration Plane Extraction from 3D data

While extracting planes from 3D data is a relatively easy task, determining the correct calibration plane is more challenging. This is in part due to the large number of planes one can find in indoor environment, noise, and incomplete points on the plane.

We first extract scene planes using RANSAC, where points are considered inliers to the plane hypothesis with a distance defined by the known range uncertainty of the sensor (around 1cm for the Hokuyo). We filter planes with too few points ( $\kappa = 2000$ ) to avoid low quality solutions and small segments given the target is expected to be relatively visible in the scene. This step is run iteratively on the dataset by removing plane candidates and their inliers at every iteration until the number of remaining points is below a significant fraction of the original dataset size (80%).

Given the initialization of the camera and lidar, we utilize the normal estimate for the targets observed in the camera to narrow the search for the target in the 3D data. Essentially, we rank and select plane hypothesis based on the agreement between the plane normal estimates from the camera and from the lidar, based on the current camera to lidar transform. Even with a very poor initialization, we find that this produces robust target identification.

#### 4.3. Optimizing External Parameters

Given a set of plane normals in the camera frame from several views  $\mathbf{n}_i$ , circle centers  $C_i$  and their corresponding 3D points on the calibration plane extracted from the range data  $P_i$ , we seek to find a 6 DOF calibrating transform. Similar to the internal calibration, we use the point-to-plane



ICP with nonlinear optimization via LM [16]:

$${}^c\mathbf{H}_{\mathcal{M}} \leftarrow \underset{\mathbf{H}}{\operatorname{argmin}} \{ \mathbf{E}(\mathbf{H}) \} \quad (11)$$

$$\mathbf{E}(\mathbf{H}) = \sum_i^{\#views} \sum_j^{|P|} \| \mathbf{n}_i \cdot (C_i - \mathbf{H}P_j) \|^2. \quad (12)$$

Given this calibration, points in the range sensor coordinate frame  ${}^{\mathcal{M}}\mathbf{X}_{ij}$  can be projected onto the image by:

$$\mathbf{x}_{ij} \equiv {}^c\mathbf{M}_i {}^c\mathbf{H}_{\mathcal{M}} {}^{\mathcal{M}}\mathbf{X}_{ij}, \quad (13)$$

where  ${}^c\mathbf{M}_i = K_i [R_i | t_i]$  is camera  $i$  projection matrix. Similar to the internal calibration approach, the rotation is parametrized with Euler angles.

## 5. Experiments & Results

Assessing the accuracy of calibration is somewhat challenging given the lack of direct ground truth. Indeed, much of the related work cited here provides only crude measures of true system performance. We evaluate precision and accuracy in two ways: first by measuring the consistency of the solution and estimating the solved parameter distribution through a bootstrap-like mechanism [6]. Second, we evaluate the results in the context of a 3D mapping system – the target application that motivated this work.

### 5.1. Solution Repeatability & Reliability

To assess the solution repeatability, in lieu of ground truth, we perform several runs of the optimization in a bootstrap-like fashion on a randomly selected sub-samples of a dataset. Specifically, we sample  $T$  times with replacement  $N$  views from the set of observations of the calibration target. For each sample, we run the calibration algorithm described above. We then report on the mean and standard deviation for each parameter estimate. Table 1 shows the external calibration results for  $N$  ranging from 6 to 17 from a set of 18 views. The results were obtained from 20 independent runs.

### 5.2. Calibration for 3D Robotic Mapping

The algorithm was used to calibrate the internal and external parameters for the Hokuyo-Camera assembly shown in Fig. 2. The assembly was mounted on a robot to be used for 3D odometry and mapping. The mapping algorithms operates by registering lidar point clouds at every frame using point-to-point ICP. Since the motor spinning rate ( $\approx 0.25\text{Hz}$ ) is relatively slow given the robot's average speed ( $\approx 0.3\text{m/s}$ ), a fast and accurate 6 DOF pose estimation method is needed to rectify/warp 3D point cloud collected while the robot is in motion. For this task, we use P3P with RANSAC and two-frame SBA [2].

In this mapping approach, the calibration between the lidar and camera plays a vital role and hence mapping results are a good indication of the accuracy of calibration. Examples of the result for a relatively large indoor area of an industrial building are shown in Figs. 7 and 8.

## 6. Conclusions & Future Work

In this work, we presented an accurate and fully automated method to calibrate the intrinsic parameters of an actuated lidar and the extrinsic pose between the lidar and a camera system. Our approach utilizes a simple 2D planar calibration target that is easy to manufacture and to use in-situ or in the lab. We evaluated the performance of the algorithm using a bootstrap style of approach, and using a 3D mapping system. Results indicate that our algorithm produces high quality calibration with minimal human effort.

Our future work will explore the effect of the target size, distance and distribution of views on the accuracy of the calibration. We will also explore evaluating this approach to other range sensor types including multi-lidar systems, such as the Velodyne, and RGB-D sensors.

## Acknowledgments

This Project Agreement Holder (PAH) effort was sponsored by the U.S. Government under Other Transaction number W15QKN-08-9-0001 between the Robotics Technology Consortium, Inc, and the Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

The authors would like to thanks Carnegie Robotics LLC. (David LaRose, Dan Tascione and Adam Farabaugh) for their help in data collection.

## References

- [1] H. Aliakbarpour, P. Nuez, J. Prado, K. Khoshhal, and J. Dias. An efficient algorithm for extrinsic calibration between a 3d laser range finder and a stereo camera for surveillance. In *Int'l Conf. on Advanced Robotics*, pages 1–6, 2009. 2
- [2] H. Alismail, B. Browning, and M. B. Dias. Evaluating pose estimation methods for stereo visual odometry on robots. In *Intelligent Autonomous Systems*, 2010. 5
- [3] J. Y. Bouguet. Camera calibration toolbox for matlab. 1
- [4] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *ICRA*, 1991. 1, 3
- [5] D. Cobzas, H. Zhang, and M. Jagersand. A comparative analysis of geometric and image-based volumetric and intensity data registration algorithms. In *ICRA*, 2002. 1
- [6] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. 5

$N$	$t_x$	$t_y$	$t_z$	$\alpha$	$\beta$	$\gamma$	mean iter.
17	$-0.02 \pm 0.002$	$-0.05 \pm 0.003$	$0.09 \pm 0.001$	$62.70 \pm 0.121$	$15.10 \pm 0.148$	$27.37 \pm 0.151$	$9.00 \pm 0.000$
16	$-0.02 \pm 0.004$	$-0.04 \pm 0.003$	$0.09 \pm 0.001$	$62.75 \pm 0.098$	$15.06 \pm 0.191$	$27.41 \pm 0.132$	$9.00 \pm 0.000$
15	$-0.02 \pm 0.004$	$-0.05 \pm 0.004$	$0.09 \pm 0.001$	$62.72 \pm 0.142$	$15.05 \pm 0.237$	$27.36 \pm 0.218$	$9.00 \pm 0.000$
14	$-0.02 \pm 0.004$	$-0.05 \pm 0.005$	$0.09 \pm 0.002$	$62.61 \pm 0.190$	$15.06 \pm 0.277$	$27.10 \pm 0.421$	$9.17 \pm 0.383$
13	$-0.02 \pm 0.006$	$-0.04 \pm 0.005$	$0.09 \pm 0.002$	$62.72 \pm 0.175$	$14.95 \pm 0.328$	$27.29 \pm 0.292$	$9.17 \pm 0.383$
12	$-0.03 \pm 0.005$	$-0.05 \pm 0.010$	$0.09 \pm 0.002$	$62.80 \pm 0.348$	$14.99 \pm 0.417$	$27.19 \pm 0.345$	$9.17 \pm 0.383$
11	$-0.02 \pm 0.007$	$-0.05 \pm 0.005$	$0.09 \pm 0.002$	$62.76 \pm 0.200$	$14.95 \pm 0.259$	$27.15 \pm 0.445$	$9.33 \pm 0.485$
10	$-0.02 \pm 0.008$	$-0.05 \pm 0.011$	$0.09 \pm 0.002$	$62.67 \pm 0.369$	$14.99 \pm 0.379$	$26.91 \pm 0.460$	$9.50 \pm 0.514$
09	$-0.02 \pm 0.010$	$-0.05 \pm 0.009$	$0.09 \pm 0.004$	$62.64 \pm 0.221$	$15.02 \pm 0.345$	$26.90 \pm 0.465$	$9.39 \pm 0.502$
08	$-0.02 \pm 0.011$	$-0.05 \pm 0.012$	$0.09 \pm 0.003$	$62.87 \pm 0.432$	$14.92 \pm 0.485$	$26.93 \pm 0.489$	$9.78 \pm 0.428$
07	$-0.03 \pm 0.009$	$-0.05 \pm 0.011$	$0.09 \pm 0.003$	$62.74 \pm 0.365$	$14.83 \pm 0.460$	$26.88 \pm 0.442$	$9.78 \pm 0.428$
06	$-0.02 \pm 0.020$	$-0.04 \pm 0.014$	$0.09 \pm 0.006$	$62.97 \pm 0.672$	$14.65 \pm 0.628$	$26.78 \pm 0.476$	$9.78 \pm 0.428$

Table 1. Solution statistics for 20 runs on randomly sampled views. The first column denotes the number of views used in the optimization. Note that the last row (6 views) is the minimum samples size needed to constraint the problem. Translation units are in meters.  $\alpha$ ,  $\beta$  and  $\gamma$  are the Euler angles in degrees about the  $Z$ ,  $Y$ , and  $X$ -axes respectively. The algorithm was initialized with the identity  $\mathbf{H} = \mathbf{I}_4$ . The results show low variance on the estimated parameter with consistent results even when using the minimum sample size.

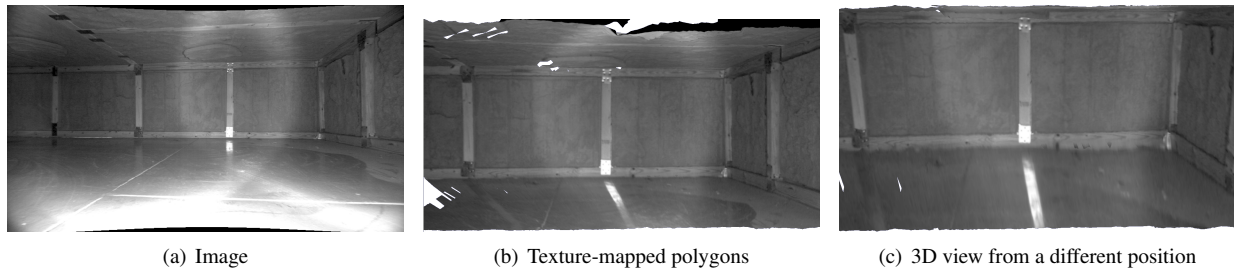


Figure 6. Calibration results inside a tunnel for a mapping application.

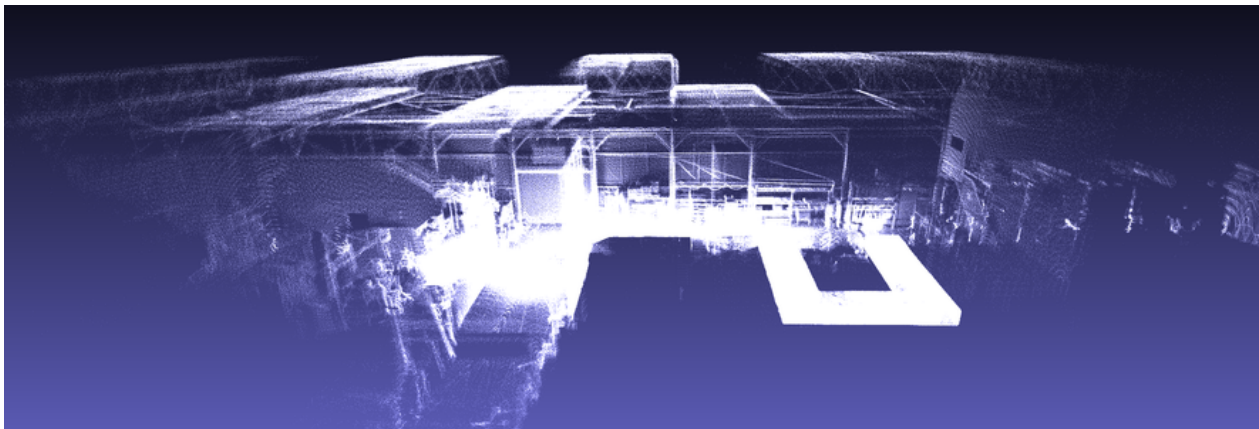


Figure 7. Example of mapping result inside a large indoor area. The tunnel shown in Fig. 6 is the U shaped area on the right. The total number of points is  $\approx 5.4$  million. The robot was driving slower in the tunnel and hence the higher density of points.

- [7] N. Fairfield. *Localization, Mapping, and Planning in 3D Environments*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 2009. 3
- [8] D. Forsyth, J. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3d object recognition and pose. *PAMI*, 13(10):971–991, oct 1991. 3
- [9] S. Fuchs and G. Hirzinger. Extrinsic and depth calibration of tof-cameras. In *CVPR*, pages 1–6, june 2008. 1
- [10] A. Geiger, F. Moosmann, O. Car, and B. Schuster. Automatic calibration of range and camera sensors using a single shot. In *ICRA*, 2012. 1, 2
- [11] L. Guan and M. Pollefeys. A Unified Approach to Calibrate a Network of Camcorders and ToF cameras. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms*

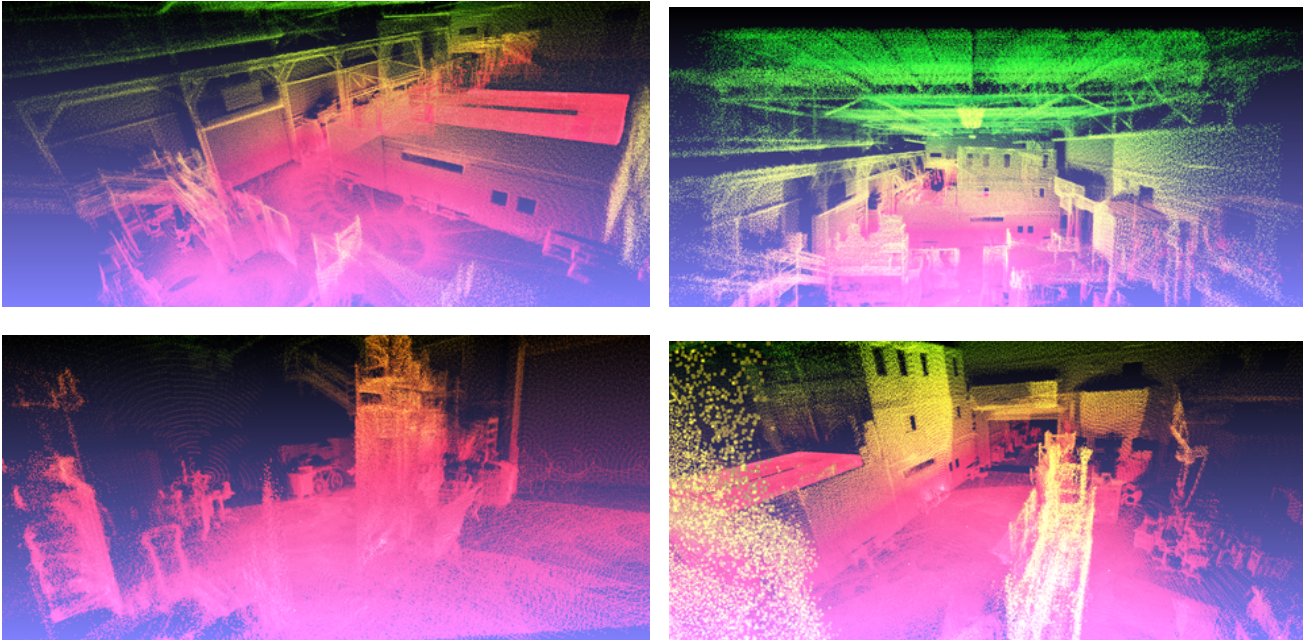


Figure 8. Different close up views colorized by height. Notice the sharpness of the reconstructed structure.

- and Applications, 2008. 2
- [12] W. J. Feature-based 3D SLAM. PhD thesis, EPFL, 2006. 1
- [13] K. Kanatani and W. Liu. 3d interpretation of conics and orthogonality. *CVGIP: Image Understanding*, 58(3), 1993. 3
- [14] N. Kirchner, D. Liu, and G. Dissanayake. Surface type classification with a laser range finder. *Sensors Journal, IEEE*, 9(9):1160–1168, sept. 2009. 1
- [15] K. H. Kwak, D. Huber, H. Badino, and T. Kanade. Extrinsic calibration of a single line scanning lidar and a camera. In *IROS*, 2011. 1
- [16] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. J. Appl. Maths.*, II(2):164–168, 1944. 3, 5
- [17] G. Lisca, P. Jeong, and S. Nedeveschi. Automatic one step extrinsic calibration of a multi layer laser scanner relative to a stereo camera. *Int'l Conference on Intelligent Computer Communication and Processing*, 0:223–230, 2010. 1
- [18] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):pp. 431–441, 1963. 3
- [19] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis. 3d lidar-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *International Journal of Robotics Research*, 31(4):452–467, 2012. 2
- [20] J. L. Mundy and A. Zisserman. Appendix - projective geometry for machine vision, 1992. 3
- [21] J.-N. Ouellet and P. Hebert. Precise ellipse estimation without contour point extraction. *Machine Vision and Applications*, 2008. 4
- [22] 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 *AAAI*, 2012. 1, 2
- [23] S. Savarese, T. Tuytelaars, and L. V. Gool. Special issue on 3d representation for object and scene recognition. *Computer Vision and Image Understanding*, 113(12), 2009. 1
- [24] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In *IROS*, 2007. 2
- [25] S. Scherer, J. Rehder, S. Achar, H. Cover, A. D. Chambers, S. T. Nuske, and S. Singh. River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 32(5), May 2012. 1
- [26] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: Science and Systems*, June 2009. 1
- [27] R. Unnikrishnan and M. Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. Technical Report CMU-RI-TR-05-09, Robotics Institute, July 2005. 1, 2
- [28] F. Vasconcelos, J. P. Barreto, and U. Nunes. A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *PAMI*, 99, 2012. 1
- [29] I. Wulf and B. Wagner. Fast 3D Scanning Methods for Laser Measurement Systems. In *Int'l Conf. on Control Systems and Computer Science*, volume 1, 2003. 1
- [30] C. Ye. Mixed pixels removal of a laser rangefinder for mobile robot 3-d terrain mapping. In *Information and Automation*, pages 1153–1158, june 2008. 1
- [31] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IROS*, volume 3, pages 2301–2306, 2004. 1, 2