Fast Extrinsic Calibration of a Laser Rangefinder to a Camera

Ranjith Unnikrishnan Martial Hebert

CMU-RI-TR-05-09

July 2005
(last updated Nov 2006)

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Keywords: laser rangefinder, calibration, mobile robot, sensors.

Last updated November 10, 2006

Abstract

External calibration of a camera to a laser rangefinder is a common prerequisite on today's multi-sensor mobile robot platforms. However, the process of doing so is relatively poorly documented and almost always time-consuming. This document outlines an easy and portable technique for external calibration of a camera to a laser rangefinder. It describes the usage of the **Laser-Camera Calibration Toolbox (LCCT)**, a MATLAB® -based graphical user interface that is meant to accompany this document and facilitates the calibration procedure. We also summarize the math behind its development.

The software is accessible online at www.cs.cmu.edu/~ranjith/lcct.html, as well as at the VMR Lab Software page at www.cs.cmu.edu/~vmr/software/software.html .

Contents

1 Introduction

Calibration is basic requirement in multi-sensor platforms where data needs to represented in a common reference frame for the purpose of analysis and data fusion. On platforms where a camera provides intensity information in the form of an image and a laser supplies depth information in the form of a set of 3-D points, external calibration allows reprojection of the 3-D points from the laser coordinate frame to the 2-D coordinate frame of the image.

The process of external calibration is often poorly documented and is almost always notoriously laborious. It usually involves some modification of the scene in the form of markers with features visible by both the camera and the laser. These features are often in the form of edges or corners extracted from the laser data with considerable user intervention [3, 6]. In a popular variant of the procedure, the laser is laboriously positioned to fire exactly at a marked point that is highlighted for easy detection in the image [4]. In procedures where the features are edges or corners, the quality of the final estimates is dependent on the accuracy by which features can be localized and so requires the ability to obtain laser data at very high resolution. Note that this document does not assume existence of additional information, like laser reflectance, is available to ease the procedure of establishing correspondences between features [3].

Another variant of the procedure uses 3-D spheres as markers introduced in the scene. Using a fitting procedure along with prior knowledge of the radius of the sphere, the 3-D coordinates of the sphere center can be estimated with sufficiently high accuracy on condition that there are sufficient laser returns off its surface. The sphere can also be easily detected in an image. Using several such observation pairs of 3-D sphere centers in the laser reference frame and corresponding 2-D centers in the image frame, a traditional calibration procedure can be used to determine the mapping between the two coordinate systems. The disadvantages of this method are that: (1) the spheres need to be large enough so that there are sufficient laser returns off the surface, and (2) this requirement of large spheres even with high resolution rangefinder systems makes the process much less portable.

The procedure proposed in this document uses no more than the same checkerboard calibration target commonly used for internal calibration of the camera. An interactive GUI allows the user to select a region of points in a range image which contain the planar calibration pattern. A robust fitting procedure fits a plane to this selection to find estimates of the perpendicular direction and distance to the plane with respect to the coordinate frame of the laser. A separate procedure to internally calibrate the camera provides independent estimates in the coordinate frame of the camera. A two-stage optimization procedure then aligns the two coordinate systems to minimize the difference in observed orientation and distance of the recorded 3-D planar points to the plane observed by the camera. The result gives the 3-D rigid transformation between the coordinate frame of the camera and the laser frame.

As is not uncommon in research, the above technique was conceived and

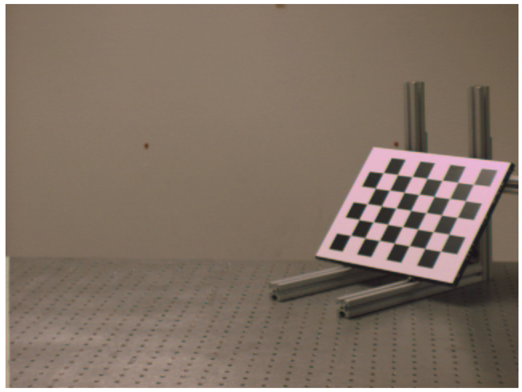

Figure 1: An example of an image observation made of the checkerboard calibration pattern

developed independently of the near identical work by Q. Zhang and R. Pless [7] presented in IROS 2004, although the algorithm in this document has been in regular use since May 2003. The only minor differences are in formulation of the linear solution (Section 5.2) and that our work is geared toward calibrating lasers sensing in full 3-D rather than 2-D. The work in [7] additionally fixed the camera distortion parameters, perhaps to capitalize on a slightly easier optimization problem, and attempted to improve the other parameters. Although a marginally smaller variance in parameter estimates was reported as a result of the procedure, there was no evidence to indicate that the new estimates were any closer to the true answer. We chose not to try to optimize the camera parameters simultaneously due to the high non-linearity of the chosen cost function.

This document is organized as follows. Section 2 states the assumptions of the method. The prerequisites are listed in Section 3 followed by a description of the procedure in Section 4. Finally Section 5 outlines the mathematical formulation of the problem and describes how it is solved in MATLAB®.

2 Assumptions

This procedure assumes that the laser data is

1. not very noisy: This is because it would otherwise require a large number of camera-laser observation pairs to get any reasonable degree of accuracy. It is also reasonable in the context of the falling prices of modern laser-based measurement systems.

2. of sufficiently high spatial resolution: The dense data requirement is so that a sufficient number of laser returns are recorded off the calibration target. This is usually not a problem in systems where the user has

control over the sweep rate or operating resolution of the laser device through software. In practice, a number of just 20-30 points recorded off the calibration target are usually sufficient.

3. in available in a stationary coordinate frame (e.g. the laser mount) and the origin is located at or close to the point from where the lasers emanate: Because the selection of points (explained later) is in 2-D and the origin is assumed as the center of projection, large offsets between the origin of the chosen coordinate frame and the true center of projection could lead to unintentional selection of outlying points lying behind the target.

4. in a right-forward-up coordinate system: This means that the vector pointing "up" is assumed to be aligned with the positive Z-axis as $\begin{bmatrix}0 & 0 & 1\end{bmatrix}$. The reason is assumption is made is only so that 2-D projected range images are rendered "right side up" instead of arbitrarily rotated about the origin.

   Important: If your scanner uses a different axis convention, e.g. the up direction is associated with the positive Y-axis as $\begin{bmatrix}0 & 1 & 0\end{bmatrix}$, change the line that reads:

   ```
   handles.up_vector = [0 0 1]';
   ```

   in the file `view_range_image.m` (Line 75) accordingly. Note that the vector is defined as a column vector by convention.

3 Requirements

1. MATLAB® 5.x or higher pre-installed on any platform capable of providing a graphical interface.

2. Installation of the Camera Calibration Toolbox for MATLAB® [2] : The technique described in this document makes heavy use of this toolbox, and relies on it completely for intrinsic calibration of the camera. The project page is at
   www.vision.caltech.edu/bouguetj/calib_doc.
   Make sure that the installation directory of the above toolbox is in your path.

3. Installation of the Laser-Camera Calibration Toolbox which is the software we will use. The install procedure involves nothing more than unzipping a tar file containing .m and .fig files and sample data (laser scan-image pairs).

   Make sure that the installation directories of the above toolboxes are in your MATLAB® path.

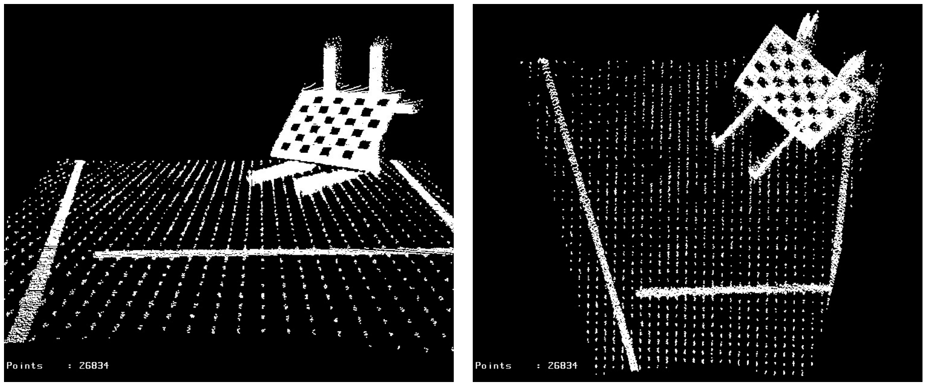

Figure 2: Two views of the 3-D points in the laser scan corresponding to the image observation of Figure 1

4. A checkerboard calibration rig: This is required for intrinsic calibration of the camera as well as extrinsic calibration. The pattern can also be downloaded from the above page as eps or pdf. The larger the size of the calibration target, the better usually. It is highly recommended that the pattern be printed out on plastic or thick poster paper and glued onto a flat board. Printing on normal printer paper is discouraged as it tends to warp and deform over time because of humidity and temperature.

4 Procedure

4.1 Data Collection

1. Rigidly place the calibration pattern (either by holding it very still or by using a mechanical device such as a mounted clamp) at a point visible from both the laser and the camera. Take a laser scan and an image, and inspect both to see whether:

   (a) there are at least 20-30 laser points recorded on the planar calibration target, and

   (b) the corners of the checkerboard pattern are clearly discernible in the image.

   Depending on your observations, it may be necessary to move the calibration target to be positioned closer to the laser/camera system to get the desired quality of the laser scan/image. Once this is done, repeat the process for different positions of the calibration target to get a feel for the range in position and orientation of the target that will cover the field of view of the camera while ensuring sufficient number of laser returns in each scan.

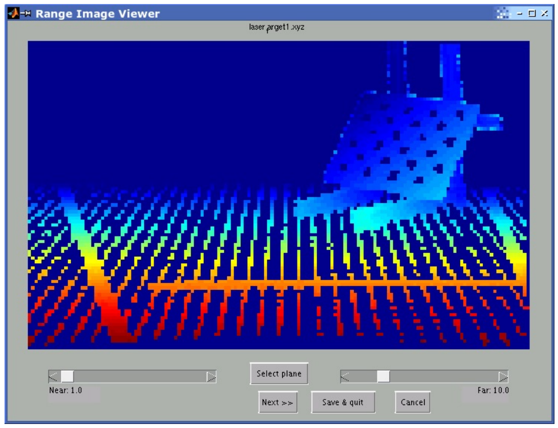

Figure 3: The range image corresponding to the scan of Figure 2 rendered in the 'jet' colormap. Cooler (blue) regions correspond to points further away from the viewer, while warmer (red) regions are closer.

2. Take 15-20 laser scan-image *pairs* for varying valid positions and orientations of calibration target. Let the scans be named as `laser_targetX.xyz` and the images as `img_targetX.jpg`, where `X` is a number indexing the observation and a scan with a given index corresponds to the image having the same index. (Note: the images may be in any MATLAB® -readable format and not necessarily JPEG.)

   Note that it is *crucial* for a scan and an image to have the same index in their respective filenames, e.g. `laser_target1.xyz` and `img_target1.jpg`, to correspond to observations of the calibration target placed at the same physical location.

Figures 1 and 2 show examples of one such observation pair - an image of the calibration target with two views of the corresponding captured laser scan.

4.2 Intrinsic calibration of the camera

The camera intrinsic parameters are calibrated within the MATLAB® environment using the procedure described in the Camera Calibration Toolbox for MATLAB® project page on the images (`imageX.jpg`) obtained in Section 4.1. A good place to start your reading is from the "Doing your own calibration" page.

The procedure essentially involves supplying basic parameters like window size and number of squares in each dimension of the grid, clicking the corners of the calibration grid in each of the images and then repeating part of the procedure in a post-processing step to achieve the desired accuracy in the final estimates.

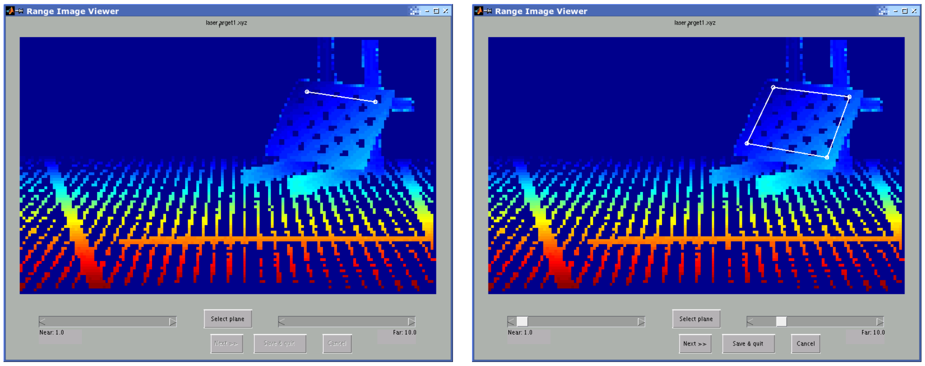

Figure 4: User selecting vertices of a polygonal region (shown in white edges) containing the planar calibration target in the range image of Figure 3

At the end of this process, click the `Save` button to save the results (parameter estimates) of the calibration in a file named `Calib_Results.mat`. This file will be used in later steps for external calibration of the camera to the laser.

4.3 Computing extrinsic parameters

1. Start the gui interface using the **lasercamcalib** command. Note that the directory containing the scan-image pairs should be added to the MATLAB® search path.

2. Click on "Select a camera calib file" and choose the `Calib_Results.mat` file create in the last section.

3. Click on "Search scan files" and enter the basename of the ladar scan files without the numeric index or extension. E.g. The basename for file `laser_target.xyz` is "`laser_target`". Then specify the file extension. The program will read in the names of appropriate ladar scan files.

4. Click on the "Select Planes" button to begin the process of specifying region correspondences for use in extrinsic calibration. In this step, for each scan we will select a region of points that mostly lie on the planar calibration target. To ease the process of 3-D point selection, we will instead select a 2-D region in the range image corresponding to each scan. An example of a range image corresponding to the scan of Figure 2 is shown in Figure 3. The region corresponding to the planar calibration target can be clearly delineated.

   For some scans it may be the case that the 2-D region corresponding to the projection of points on the planar calibration target occupies a very small area in the range image. This can make the selection process difficult even in the 2-D range image. To compensate for this, adjust the sliders labeled "Near" and "Far" to control the volume of the region of interest that is rendered in the display. The numbers next to the labels determine the minimum and maximum distance, respectively, from the origin for a point to be rendered in the range image. The default values are 1.0m and 10.0m respectively. A good set of values for *near* and *far* correspond to when the portion of the range image corresponding to the planar target occupies a large but discernible portion of the image. Note that the values of these parameters do not influence the calibration procedure, but are required solely for user convenience.

5. The user is then sequentially shown range images corresponding to each scan and is asked to select a 2-D region containing the planar target for each range image.

   The selection of vertices of the polygon containing the plane is done sequentially in order around the polygon using the left mouse button. The selection is terminated using the right mouse button. The last selected vertex can be deleted using the `ESC` key.

   Use the "Next" and "Prev" buttons to cycle through all the range images. Click "Save and Quit" to end the selection or "Cancel" to revert to the previous state before leaving.

6. After selecting polygons containing the calibration target in the all the range images, click the "Calibrate" button. This begins the optimization routine to estimate the rigid transformation parameters.

   There are two transformation matrices displayed at the end of this step. The first corresponds to the $[R \quad t]$ matrix after the 1st stage of optimization and the second corresponds to after the 2nd stage of non-linear optimization.

   It sometimes happens that the results obtained after the first stage of optimization work better than those after the second stage. The reason for this will be made more clear in Section 5, where the two-stage optimization procedure is detailed. By default, the program uses the results after the 2nd optimization stage.

7. Click the "Save" button and specify a tag name (e.g. Laser_Cam) and an optional comment line to save two files - in this example, `Laser_Cam_calib_1.m` and `Laser_Cam_calib_2.m` containing results after the 1st and 2nd optimization stage respectively.

4.4 Visually verifying the results

In the absence of ground truth measurements for obtaining error bounds on the calibration estimates, it is useful to be able to visually inspect the quality of calibration using a separate test set of image-laser scan observation pairs.

To render a 3-D point cloud using its corresponding intensity image and the calibration parameters computed in the previous section, click the "Colorize

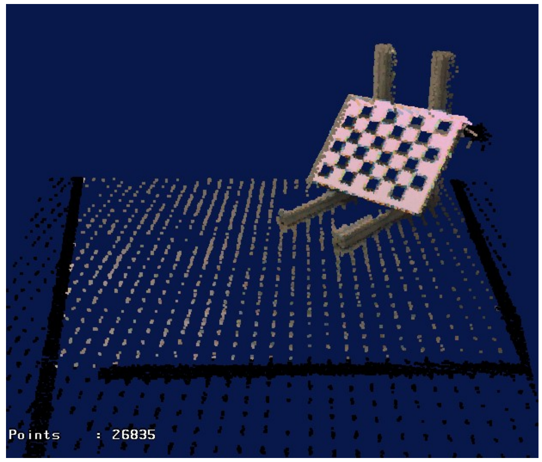

Figure 5: Scan of Figure 2 colored by reprojection onto corresponding image in Figure 1

Test Scan" button. Enter the filename of the ladar scan and image files and the program output a VRML file named rgbScan.wrl . This can be inspected with any VRML viewer (VRMLview, Cosmo Player etc.).

Figure 5 shows a screenshot from VRMLview of the scan of Figure 2 colored by reprojecting appropriate colors from the image in Figure 1. Figure 6 shows output corresponding to recolored test scans that accompany the software. The ladar scans supplied with this software have been acquired using a prototype Z+F (Zoller-Fröhlich) laser.

Examples of recolored outdoor scans, taken from a different SICK laser-based platform are shown in Figures 7 and 8.

5 Algorithm

5.1 Camera model

The calibration procedure is detailed in the Camera Calibration Toolbox for MATLAB® page. The parameters for the camera model used in the toolbox are:

- **Focal length**: The focal length in pixels in the 2×1 vector fc.
- **Principal point**: The principal point coordinates in the 2×1 vector cc.
- **Skew coefficient**: The skew coefficient defining the angle between the x and y pixel axes in the scalar *alpha_c*.
- **Distortions**: The image distortion coefficients (radial and tangential distortions) in the 5×1 vector kc.

Figure 6: Examples of supplied test scan-image pairs with scans rendered with the image after calibration

Briefly, if the 3-D coordinate point corresponding to the observed pixel is $P = [X_c; Y_c; Z_c]$, the normalized pinhole projection is given by

$$x_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Let $r^2 = x^2 + y^2$. The normalized point after including lens distortion is:

$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = \left(1 + kc(1)r^2 + kc(2)r^4 + kc(5)r^6\right) x_n + dx$$

Figure 7: Example of a test scan-image observation with scan rendered with the image after calibration

where dx is the tangential distortion vector

$$dx = \begin{bmatrix} 2kc(3)xy + kc(4)(r^2 + 2x^2) \\ kc(3)(r^2 + 2x^2)2kc(4)xy \end{bmatrix}$$

Once distortion is applied, the final pixel coordinates of the point are given by:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} fc(1) & alpha\_c * fc(1) & cc(1) \\ 0 & fc(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix}$$

The ambiguity in scale is resolved by prior knowledge of the size of the squares in the calibration target. Because of this resolution, the toolbox is also able to estimate the location of the plane in 3-D.

More specifically, let the top left corner of the square checkerboard pattern on the target be chosen as the origin of a virtual target reference frame. Let the directions along the left and top edges were chosen as x and y axii of this frame, and the direction perpendicular to the checkerboard be chosen as the z-axis. Then the toolbox also returns the rotation Rc and translation Tc of the target coordinate frame with respect to the camera, for each observation of the calibration target. These values are stored as *Rc_i* and *Tc_i* in the `Calib_Results.mat` file, where i is the index of the observation.

The values of Rc_i and *Tc_i* are extracted from the calibration file for each observation in the laser-camera calibration routine. This gives the orientation ($\theta_{c,i} \in \mathbb{S}^2$) represented as a column 3-vector, and distance ($\alpha_{c,i} \in \mathbb{R}$) of the

Figure 8: Example of a test scan-image observation with scan rendered with the image after calibration

plane $\theta_{c,i}^{\top} x - \alpha_{c,i} = 0$ with respect to the *camera* origin, for each image in the *camera* frame of reference.

5.2 Extrinsic calibration model

For each region selected in a range image, a robust total least squares estimator is used to fit a plane to the set of points in 3-D corresponding to the selection. This gives an estimate of $\theta_{l,i}$ and $\alpha_{l,i}$ in the *laser* coordinate frame. Thus for estimating the rigid transformation between the laser and camera frames, a reasonable approach would be to find the transform that minimizes the "difference" in observations of each plane. However, while a distance metric is easy to define for point features, it is not so obvious to define for planes.

We propose a two-stage estimation process. In stage I, we decouple the problem of finding the best transform into estimating a translation part and a rotation part independently. The result of this stage is used as initialization for stage II in which the minimized objective function is the distance from the user-selected inlier 3-D points to the corresponding plane observed from the image.

Stage I

In stage I, we compute an initial estimate of the transformation by estimating the translation part and the rotation part independently. The translation is the one that minimizes the difference in distance from the camera origin to each plane, represented in the camera coordinate system and the laser coordinate system.

The distance between the origin and the planes wrt the coordinates system of the camera is given by the α_c's. Upon translation to position t, the distance from the planes become $\alpha_c - \theta_c^{\mathsf{T}} t$. We may expect the position of the laser origin in the camera reference frame to be the position at which the distances to the planes are equal to those reported by the laser.

Thus, the objective function for the translation t may then be written as

$$\min_t \sum_i \left(\alpha_{l,i} - (\alpha_{c,i} - \theta_{c,i}^{\mathsf{T}} t)\right)^2$$

We may write this in a convenient matrix form with some added notation. Let:

$$\theta_c = \begin{bmatrix}\theta_{c,1} & \theta_{c,2} & \dots & \theta_{c,n}\end{bmatrix} \qquad \theta_l = \begin{bmatrix}\theta_{l,1} & \theta_{l,2} & \dots & \theta_{l,n}\end{bmatrix}$$
$$\alpha_c = \begin{bmatrix}\alpha_{c,1} & \alpha_{c,2} & \dots & \alpha_{c,n}\end{bmatrix}^{\mathsf{T}} \qquad \alpha_l = \begin{bmatrix}\alpha_{l,1} & \alpha_{l,2} & \dots & \alpha_{l,n}\end{bmatrix}^{\mathsf{T}}$$

where n is the number of scan-image observation pairs, and the primary subscript *c* (or *l*) denotes the camera (or laser) frame of reference respectively.

Then the objective function for t is equivalent to

$$t_1 = \arg\min_t ||\theta_c^{\mathsf{T}} t - (\alpha_l - \alpha_c)||_{\mathcal{F}}^2$$

which has a closed form solution given by:

$$t_1 = (\theta_c \theta_c^{\mathsf{T}})^{-1} \theta_c (\alpha_c - \alpha_l)$$

We also find the rotation between the reference frames that minimizes the angular difference between the normal from the origin to the corresponding planes as reported in the laser and camera frames. This objective function may be written as the rotation that maximizes the sum of cosines of the respective angles, or

$$R_1 = \arg\max_R \sum_i \theta_{c,i}^{\mathsf{T}} (R\theta_{l,i})$$

such that R is a rotation matrix with $R^{\mathsf{T}} R = I_3$ and $\det(R) = 1$. This is equivalent to

$$R_1 = \max_R \operatorname{trace}(\theta_c^{\mathsf{T}} R \theta_l) = \max_R \operatorname{trace}(R\theta_l \theta_c^{\mathsf{T}})$$

This problem is an instance of the well-studied Orthogonal Procrustes Problem (OPP) [5, 1] and has the closed form solution given by

$$R_1 = VU^{\mathsf{T}} \tag{1}$$

where $\theta_l \theta_c^{\mathsf{T}} = USV^{\mathsf{T}}$ is the associated singular value decomposition. We outline a basic proof of this result in Appendix A.

Stage II

In the stage II of the optimization, the objective function to be minimized is chosen to be the distance from the user-selected inlier 3-D points to the corresponding plane observed from the image. The inliers can be estimated as a byproduct of the least squares fitting procedure in stage I. The minimization of the objective function can then be done through an iterative optimization procedure with initial estimates chosen as the results from stage I.

Let $x_{l,i}$ be the matrix of 3-D points by the user from the i^{th} range image ordered as:

$$x_{l,i} = \begin{bmatrix} x_{l,i}^{(0)} & x_{l,i}^{(1)} & \ldots & x_{l,i}^{(m)} \end{bmatrix}$$

where $x_{l,i}^{(j)} \in \mathrm{R}^{3\times1}$ and $m = m(i)$ is the number of inliers for the i^{th} laser scan (range image).

Then the problem takes the form:

$$\arg\min_{R,t} \quad \sum_{i=1}^{n} \frac{1}{m(i)} \sum_{j=1}^{m(i)} \left(\theta_{c,i}^{\mathrm{T}} \left(R x_{l,i}^{(j)} + t \right) - \alpha_{c,i} \right)^2$$

To increase robustness to outliers, the implementation uses only the points corresponding to the inliers of the plane fitted to each selected region in the range image. The inlier points are defined to be those whose distance to the plane are less than the median distance. The rotation matrix is efficiently coded as a quaternion in the implementation, giving an intermediate parameter vector of length 7 (4 for rotation + 3 for translation). The result obtained from invoking `fminsearch` is transformed back into a rotation matrix and a translation vector, and finally returned as R_2, t_2.

6 Conclusions

The document outlined a fast and portable technique for the traditionally tedious task of extrinsic calibration of a camera to a laser-rangefinder. The accompanying software has been tested on a variety of rangefinder platforms, both indoors as well as outdoors. The technique has repeatedly provided output with accuracy sufficient to meet the demands of outdoor mobile robot perception.

One area of potential improvement is in the use of more quantitative measures to replace visual inspection for assessing calibration quality. While this is easy to do by using additional equipment and rigorously obtaining ground-truth, it would be desirable to provide quick quantitative user feedback using measures other than reprojection error, so that the practitioner may be in a better position to accept/reject scan-image pairs. A quantitative analysis of algorithm stability would also be beneficial.

The Laser-Camera Calibration Toolbox has secured a permanent place in the VMR lab and we hope to keep maintain and add more features to it over time. We hope that by making this software publicly available, we may ease the burden typically associated with calibration. Current as well as future releases will available at `www.cs.cmu.edu/~ranjith/lcct.html` as well as at the VMR Lab Software page at `www.cs.cmu.edu/~vmr/software/software.html`. We value and welcome your feedback and comments.

7 Acknowledgments

Nicolas Vandapel played a crucial role in the interfacing and setup of the laser rangefinder on a mobile platform for data collection. Jean-Francois Lalonde calibrated the Z+F (Zoller-Fröhlich) laser in the lab and generously supplied the sample data that accompanies this document. Thanks to both also for their rigorous testing of the software and for reviewing this document. Thanks also to Caroline Pantofaru for much appreciated help with data collection and experiments in the early stages of this work.

References

[1] B. Arun, Huang. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 9, 1987. 12

[2] J.-Y. Bouguet. Camera calibration toolbox for Matlab®. 3

[3] D. Cobzas, H. Zhang, and M. Jägersand. A comparative analysis of geometric and image-based volumetric and intensity data registration algorithms. In *IEEE Intl. Conference on Robotics and Automation (ICRA)*, volume 3, pages 2506–2511, 2002. 1

[4] E. Krotkov. Laser rangefinder calibration for a walking robot. Technical Report CMU-RI-TR-90-30, Carnegie Mellon University, 1990. 1

[5] P. H. Schonemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966. 12, 15

[6] S. Wasielewski and O. Strauss. Calibration of a multi-sensor system laser rangefinder/camera. In *Proc. of the Intelligent Vehicles Symposium*, pages 472–477, 1995. 1

[7] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder. In *IEEE Intl. Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2301– 2306, 2004. 2

APPENDICES

A The Orthogonal Procrustes Problem (OPP)

We specify the problem as

$$\max_R \text{ trace}(MR)$$

subject to the constraints that $R^{\mathsf{T}}R = I$ and $\det(R) = 1$

We can solve this by forming the associated Lagrangian

$$\max_R \text{ trace}(RM) + \text{trace}\left(L(R^{\mathsf{T}}R - I)\right) + g \text{ trace}\left(\det(R) - 1\right)$$

where L and g are the associated Lagrange multipliers.

Taking the derivative wrt R gives the condition

$$\begin{aligned} & M^{\mathsf{T}} + R(L^{\mathsf{T}} + L) + gR = 0 \\ \Rightarrow \qquad & R^{\mathsf{T}}M^{\mathsf{T}} = -(L^{\mathsf{T}} + L) - g = MR \end{aligned}$$

where the last equality is due to the middle expression being symmetric. This is turn gives

$$\begin{aligned} \Rightarrow \qquad & M^{\mathsf{T}} = RMR \\ \Rightarrow \qquad & M^{\mathsf{T}}M = RMRR^{\mathsf{T}}M^{\mathsf{T}}R^{\mathsf{T}} = RMM^{\mathsf{T}}R^{\mathsf{T}} \end{aligned}$$

If we define the singular value decomposition (SVD) of $M = USV^{\mathsf{T}}$, then we may rewrite the first and last terms in the above expression as

$$VS^2V^{\mathsf{T}} = RUD^2U^{\mathsf{T}}R^{\mathsf{T}}$$

which can satisfied by setting

$$RU = V$$

or

$$R = VU^{\mathsf{T}}$$

Note that this is a particular instance of the more general proof [5] and does not include the case where $\det(VU^{\mathsf{T}}) = -1$. In this case, the solution is given by

$$R = VQU^{\mathsf{T}}$$

where Q is a diagonal matrix with diagonal entries given by $[1 \; 1 \; \ldots \; 1 \; -1]$. In our application, on the reasonable assumption of planes in general position and that the laser and the camera are on the same "side" as the observed planes, this scenario does not occur in practise.

B Additional GUI Features and Options

- **List active**: This button lists the indices of the scan-image pairs to be used in the optimization routine. It is analogous to its counterpart in the camera calibration toolbox.
- **Add/Remove scans**: This button allows the user to add/remove scan-image pairs from consideration in the optimization routine. It is analogous to its counterpart in the camera calibration toolbox.
- **Preview**: This button allows user to preview the range images but does not allow plane selection.
- **Load Calib File**: This button permits loading of a laser-camera calibration file.
- **Plot Error**: Not yet implemented

C Example terminal output

```
>> lasercamcalib
Selected file: /home/ranjith/camera_calibration/Calib_Results.mat
Enter base name of scans (without numbers or suffix): laser_target
Enter suffix of scan files ([] = "xyz"): (Pressed Enter)
--- Loaded list of active scan/image pairs ---
1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
Enter indices of range images to preview ([]=all active):
Starting range image viewer... please wait
 Viewing file: laser_target1.xyz
(Choosing vertices of polygon)
Selected point (103.963,18.629)
Selected point (132.742,22.099)
Selected point (122.333,44.347)
Selected point (93.962,40.264)
Ending
Selected 5842 points
 Viewing file: laser_target2.xyz
(Choosing vertices of polygon)
Selected point (67.836,17.812)
Selected point (99.064,20.670)
Selected point (87.838,43.938)
Selected point (57.426,40.060)
Ending
Selected 6687 points
 Viewing file: laser_target3.xyz
(Choosing vertices of polygon)
Selected point (40.281,18.516)
Selected point (66.203,21.170)
Selected point (56.201,43.826)
```

```
Selected point (27.218,39.132)
Ending
Selected 5672 points
 Viewing file: laser_target4.xyz
(Choosing vertices of polygon)
Selected point (93.962,17.812)
Selected point (133.151,17.812)
Selected point (135.192,42.510)
Selected point (95.799,42.510)
Ending
Selected 9027 points
.
.
...
.
.
 Viewing file: laser_target14.xyz
(Choosing vertices of polygon)
Selected point (69.673,17.782)
Selected point (119.883,19.415)
Selected point (119.067,56.563)
Selected point (67.836,56.971)
Ending
Selected 24652 points
 Viewing file: laser_target15.xyz
(Choosing vertices of polygon)
Selected point (27.626,17.374)
Selected point (77.429,17.782)
Selected point (76.816,56.767)
Selected point (25.993,55.747)
Ending
Selected 21608 points

(Clicked Calibrate button)
Processing laser_target1.xyz
Median error was 0.000767
Processing laser_target2.xyz
Median error was 0.000744
Processing laser_target3.xyz
Median error was 0.000698
Processing laser_target4.xyz
Median error was 0.000879
Processing laser_target5.xyz
Median error was 0.000845
Processing laser_target6.xyz
Median error was 0.000849
Processing laser_target7.xyz
Median error was 0.000760
Processing laser_target8.xyz
Median error was 0.000729
```

```
Processing laser_target9.xyz
Median error was 0.000726
Processing laser_target10.xyz
Median error was 0.000852
Processing laser_target11.xyz
Median error was 0.000892
Processing laser_target12.xyz
Median error was 0.000879
Processing laser_target13.xyz
Median error was 0.001716
Processing laser_target14.xyz
Median error was 0.001740
Processing laser_target15.xyz
Median error was 0.001404
-- Optimization: Stage I
Computed RMS error in distance to planes: 0.003009
-- Optimization: Stage II
Initial RMS distance of points to planes: 0.001774
... running non-linear optimization routine...
RMS distance of points to planes after search: 0.001365
-- Result:
ans =
   -0.5270    0.8498   -0.0111   -0.1980
    0.1112    0.0560   -0.9922    0.0167
   -0.8425   -0.5242   -0.1240   -0.0917
ans =
   -0.5239    0.8517   -0.0128   -0.1945
    0.1174    0.0574   -0.9914    0.0319
   -0.8436   -0.5209   -0.1300   -0.0938
```

*(Clicked the **Colorize Test Scan** button)*

```
Enter xyz scan filename to colorize: laser_target1.xyz
Enter image filename to use for colorizing points:
img_target1.jpg
Writing VRML file rgbScan.wrl
This may take a minute, so please wait... Done!
Elapsed time is 2.307733 seconds
```