GUI App for Automatic Extrinsic Calibration of Range and Visual Sensors

2018 GSoC Project with MRPT

Karnik Ram R.

Abstract

Range and visual sensors are being increasingly used alongside one another in mobile robots. With their increasing use, calibration techniques that can accurately estimate the 6-DoF transformation between them are becoming increasingly important. In this regard, an end-to-end application with an easy-to-use graphical user interface for the extrinsic calibration of different types of sensors is proposed. The app will be able to calibrate the extrinsics of 3D LiDARs, RGB-D cameras, RGB cameras, and any combination between them. Automatic and target-less calibration algorithms based on line matching, plane matching, and trajectory matching will be implemented and integrated into the app. The user will be able to directly visualize the calibration results inside the app and also compare different algorithms wherever possible, and significantly reduce calibration efforts.

Introduction

To gather more information about the environment, robots are being equipped with larger numbers and modalities of sensors these days. For example, in an autonomous car, 3D LiDARs are being increasingly used together with 2D cameras as the dense color information of the latter complements the sparse depth information of the former. For their data to be fused together, however, requires the knowledge of the 6-DoF transformations between the sensors. Extrinsic calibration is the process of determining these transformations. In the same example of an autonomous car, without calibration, the laser distance measurements cannot be accurately projected onto the camera images, which means the color pixels in the images cannot be accurately associated with distance measurements, or in other words their data cannot be fused together meaningfully.

Several calibration techniques have been proposed over the years, and these can broadly be classified into target-based and target-less techniques, depending on whether they require specific calibration targets like checkerboards. These techniques can further be classified based on whether or not they require some amount of manual intervention, like the manual marking of correspondences. Unnikrishnan et al. [1] proposed one of the fist stand-alone applications for extrinsic calibration as a Matlab toolbox. The app had a GUI for manually labelling a checkerboard and could calibrate a laser rangefinder to a camera. Geiger et al. [2] later proposed a web-based toolbox for calibrating camera-to-camera and camera-to-range sensor systems. Several ROS packages have been made available as well, like the one made by Dhall et al. [3]

However, all these applications are still restricted to specific scenes and require considerable human effort, which can significantly frustrate research efforts. With this motivation in mind, an end-to-end GUI application for automatic extrinsic calibration of range and visual sensors is proposed. The app will enable users to test out and compare different automatic calibration algorithms, on generic scenes, and also visulaize the fused sensor output wherever possible.

The specifics of the application along with a proposed timeline, and the details of the algorithms that will be implemented are discussed in the following sections.

Deliverables

- A Qt GUI which integrates three different calibration algorithms, allowing the user to choose the sensor combination, input the .rawlog sensor data and visualize it, choose the calibration algorithm, and initialize the algorithm. The GUI will also show the calibration progress, and present the results along with their visualization.
- An implementation of the line matching technique described by Perez-Yus et al. [4] for calibrating range sensors with an RGB camera (3D LiDAR - RGB, RGB-D - RGB, 3D LiDAR - RGB-D). This includes 3D line feature extraction from 3D LiDAR and RGB-D data, normal extraction from RGB data, RANSAC-based matching followed by non-linear minimization of the geometric cost function.
- An implementation of the plane matching technique described by Fernandez-Moral et al. [5] for calibrating range sensors (3D LiDAR - 3D LiDAR, RGBD- RGBD, RGBD - 3D LiDAR). This includes 3D plane extraction from range sensor data, RANSAC-based matching, and least squares minimization of the geometric cost function.
- An implementation of the trajectory matching technique described by Jarrett Holtz and Joydeep Biswas [6] for calibrating range sensors (3D LiDAR - 3D LiDAR, RGB-D - RGB-D, RGB-D - 3D LiDAR). This includes using ICP for the ego-motion estimation of LiDARs, DifOdometry for RGB-D cameras, and eigen value decomposition for solving for the transforms.
- Detailed documentation on how to use the app, along with a demo video. Report of all the work done during the project duration.

Stretch goals

- Integrate trajectory matching based RGB - range sensor calibration into the app by estimating RGB ego-motion appropriately.
- A separate GUI window for the comparison of different calibration algorithms for a particular sensor combination, wherever possible.
- Extend the implemented algorithms to calibrate multi-sensor arrays (more than two sensors).

Approach

In this section, the different automatic calibration algorithms that will be implemented and integrated into the app are discussed.

Extrinsic calibration from line observations [4]

- An algorithm to calibrate range sensors viz. 3D LiDARs and RGB-D cameras, with 2D cameras, even with non-overlapping FOVs. The algorithm is based on extracting lines from the scene, and matching them to impose geometric contraints.

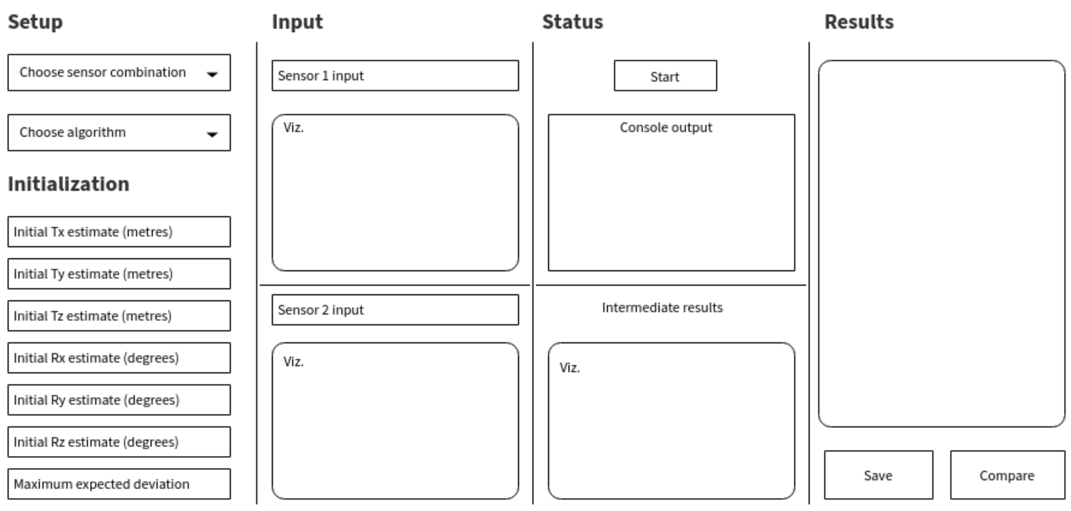
Setup
Choose sensor combination
Choose algorithm
Initialization
Initial Tx estimate (metres)
Initial Ty estimate (metres)
Initial Tz estimate (metres)
Initial Rx estimate (degrees)
Initial Ry estimate (degrees)
Initial Rz estimate (degrees)
Maximum expected deviation
Input
Sensor 1 input
Viz.
Sensor 2 input
Viz.
Status
Start
Console output
Intermediate results
Viz.
Results
Save
Compare

Figure 1: A wireframe of the app that is to be developed.

- 2D line segments are extracted from the images by applying a Canny filter in combination with RANSAC, and these segments are used to extract the corresponding 3D normals as $\mathbf{n} = \mathbf{l} \times \mathbf{r_i}$ where $\mathbf{l}$ is the line segment and r_i is the 3D ray passing through the mean of the line segment's inlier points.

- 3D lines are extracted from the range sensor data as plane intersections. 3D planes are extracted by applying RANSAC for plane fitting. Alternatively, for an RGB-D camera, 2D line segments can be extracted directly using the procedure described above and then transformed to 3D.

- After extraction, a set of line correspondences is established using an initial estimate as $\mathscr{L}_i = \{\mathbf{p}_i^D, \mathbf{v}_i^D, \mathbf{n}_i^C\}$ where $\mathbf{p_i}^D, \mathbf{v_i^D}$ are the point and direction vector corresponding to the 3D line, and $\mathbf{n_i^c}$ is the normal vector in the range sensor's frame and camera's frame respectively.

- The maximum likelihood estimate of the rigid body 6-DoF transformation T_D^C between the 2D camera and the range sensor is determined by decoupling the rotation and translation. Using the orthogonality condition the rotation is computed as the matrix that satsifies $(n^{C_2}).R.\mathbf{v}^{C_1}$. This is equivalent to the following non-linear least squares minimization:

$$\arg\min_{\mu} \sum_{i=1}^{N_L} (\mathbf{n}_i^T . e^{\mu} R \mathbf{v}_i)^2$$

- Similarly the translation vector can be solved from $(\mathbf{n}^C)^T.(R\mathbf{p}^D + t) = 0$ This is equivalent to the non-linear least squares minimization:

$$\arg\min_{t} \sum_{i=1}^{N_L} \left(\mathbf{n}_i^T . \frac{\mathbf{R}\mathbf{p}_i + \mathbf{t}}{\|R\mathbf{p_i} + \mathbf{t}\|} \right)^2$$

  The least squares problem in both the cases is solved using Gauss-Newton.

- A minimal set of three random correspondences is pulled from the set $\mathscr{L}$ to perform the calibration, and repeated using RANSAC to obtain consistent correspondences. The final calibration is computed using the inlier correspondences.

Extrinsic calibration from plane observations [5]

- An algorithm to calibrate range sensors viz. 3D LiDARs and RGB-D cameras, or their combination, by extracting and matching planes. The algorithm requires only the co-observation of planar surfaces.

- Planes from depth images are extracted using a region-growing approach, and the extracted planes are represented as $\mathbf{n}.\mathbf{p}+d = 0$ where $\mathbf{n}$ is the normal vector, d is the distance of the plane from the camera centre, and $\mathbf{p}$ is any point lying on the plane. These plane parameters are estimated along with their covariences assuming noise only affects the distance measurements.

- A set of plane correspondences is then established using an initial estimate for the pose. The correspondences are further refined by applying simple heuristics such as: a) the angle between the normal vectors of both planar regions is smaller than a threshold b) the distances of both the planes from the camera center are smaller than a threshold c) the sizes of the planes are large enough. The process is performed automatically as the sensor setup is moved. It is also ensured that the correspondences fulfill the observability condition.

- After the correspondences have been gathered, the rotation and translation are solved for separately. The maximum likelihood estimate of the rotation is solved for as the following least sqaures problem

  $$\arg\min_{R} \sum_{i=1}^{N} \omega_i \|\mathbf{n}_i - R\mathbf{n}_i^{'}\|^2$$

  where $\omega_i = \frac{1}{|\Sigma_i|}$ considering independent errors of the correspondences. This can further be simplified to obtain R as the singular value decomposition

  $$R = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(VU^T) \end{pmatrix} U^T$$

  where V and U are 3×N matrices with the normal vectors $\mathbf{n'}_i$,$\mathbf{n}_i$ as their columns.

- The maximum likelihood estimate of the translation is solved for as the following least squares problem

  $$\arg\min_{t} \sum_{i=1}^{N} \omega_i (d_i - d_i^{'} + \mathbf{t}.\mathbf{n}_i)^2$$

  with $\omega_i = 1/\sigma_i$

Extrinsic calibration from motion observations [6]

- An algorithm that solves for the calibration using ego-motion estimates from the sensors, and the perceived changes in ego-motion known as " delta-transforms".

- Delta-transform, written as D_i^t, is defined as the transform that takes points from sensor i's frame of reference at timestep $t + \Delta t$ to timestep t. orresponding sets of delta-transforms from a pair of sensors are used to compute the extrinsic calibration.

- The Delta-transforms can be related using $D_1^t A = A D_2^t$ where

$$D_1^t = \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}, \; D_2^t = \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix}, \; A = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

  Considering the rotation and translation components separately we have,

$$R_1 R_2 = R R_2, \; R T_2 + T = R_1 T + T_1$$

- The rotation component can be equivalently represented in quaternion form as $q_1 q = q q_2$. Using rules of quaternion multiplication we can obtain a linear system $M_1 q = M_2 q$ where M_1 is a matrix given by quaternion post-multiplication and M_2 is a matrix given by quaternion pre-multiplication. The quaternion vector q can then be solved for by computing the following eigenvalue decomposition.

$$\arg\min_q \sum_t [(M_1^t - M_2^t)q]^T [(M_1^t - M_2^t)q]$$

- Using the calculated rotation component, the translation T can be expressed as $(a^T a)^{-1} a^T b$ where $a = (I - R_1)$ and $b = T_1 - R T_2$. A solution to T is then computed using multiple delta-transforms as

$$T = \left(\sum_t a_t^T a_t \right)^{-1} \left(\sum_t a_t^T b_t \right)$$

- Best results are obtained when the delta-transforms are time synchronized across the sensors, are calculated over keyframes, and cover all the axes of motion.

Project Timeline

Days	**Task**
May 1 - 14	Complete the skeleton of the GUI without adding any functionality, and validate with mentors
May 15 - 21	Add functionality to input range sensor data, visualize the data, and extract planes from them
May 22 - 28	Establish correspondences among the extracted planes using a set of heuristics and RANSAC, check for observability condition
May 29 - June 4	Solve for rotation and translation using least sqaures, test performance
June 5 - 11	Add GUI functionality to visualize the results as a fused 3D reconstruction
First evaluation	Fully functional GUI with an implementation of the plane matching algorithm integrated into the app
June 12 - 18	Input 2D camera data, extract lines and normals from range sensor and camera data
June 19 - 25	Solve for the rotation and translation using Gauss-Newton with an initial set of correspondences

June 26 - July 2	Refine set of correspondences and calibration results using RANSAC, test performance
July 3 - 9	Visualize the results, provide documentation for work dones so far with examples and summaries
Second evaluation	Implementation of the line matching algorithm integrated into the app, with initial documentation, and examples
July 10 - 16	Obtain ego-motion estimates for range and RGB-D data using ICP and DifOdometry, and integrate into the app
July 17 - 23	Compute delta-transforms from the ego-motion estimates, establish correspondences, and solve for rotation
July 24 - 30	Solve for translation using the computed rotation, visualize the results
July 31 - Aug 6	Buffer week for any un-anticipated problems / bug fixes, and stretch goals
Aug 7 - 13	Complete documentation with examples, summaries, and a video demo on how to use the app
Final evaluation	Final pull request and merge, with complete documentation, reports, and examples

Development of the GUI has begun, and is hosted under the forked repository:
https://github.com/karnikram/autocalib-sensor-extrinsics

About me

I'm presently a graduate research assistant in the Robotics Research Center at the International Institute of Information Technology Hyderabad, India (IIIT-H). I completed my bachelor's in Electronics and Communication Engineering, last May, from SSN College of Engineering, India. I'll be starting my master's this fall. **MRPT is the only organization that I'm applying to, and hence it's my top preference.** I'll be operating from India (GMT +5:30) for the entire duration of this project.

My background for this project

As a part of my research here at IIIT-H, we've developed a deep network that can automatically estimate the extrinsic calibration parameters between a 3D LiDAR and a 2D camera. The work has been submitted to IROS 2018 for review. I'm also a contributor to lidar_camera_calibration, a popular ROS package for the extrinsic calibration of a 3D LiDAR to a 2D camera.

I have sufficient experience in most of the required technologies required for this project including C++, ROS, OpenCV, Eigen, and PCL. Please check out the following links for more details about me and my projects.

Github: https://github.com/karnikram
Webpage: http://karnikram.info
Resume: http://karnikram.info/resume.pdf
Contact: karnikram@gmail.com

Other commitments

This project will be my main focus during the summer. Since extrinsic calibration is also the focus of my research here, I will have no problems in putting the minimum number of required hours of work per week. However there might be a little dip in productivity sometime in the end of July when I will be getting ready to join my master's program in August. I will appropriately compensate for any lost hours.

References

[1] Ranjith Unnikrishnan and Martial Hebert. "Fast extrinsic calibration of a laser rangefinder to a camera". In: (2005).

[2] Andreas Geiger et al. "Automatic camera and range sensor calibration using a single shot". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 3936–3943.

[3] Ankit Dhall et al. "LiDAR-Camera Calibration using 3D-3D Point correspondences". In: *arXiv preprint arXiv:1705.09785* (2017).

[4] Alejandro Perez-Yus et al. "Extrinsic calibration of multiple RGB-D cameras from line observations". In: *IEEE Robotics and Automation Letters* 3.1 (2018), pp. 273–280.

[5] Eduardo Fernandez-Moral et al. "Extrinsic calibration of a set of range cameras in 5 seconds without pattern". In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 429–435.

[6] Jarrett Holtz and Joydeep Biswas. "Automatic extrinsic calibration of depth sensors with ambiguous environments and restricted motion". In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE. 2017, pp. 2235–2240.