Grant agreement no: FP7-600877

SPENCER:

Social situation-ware perception and action for cognitive robots

Project start: April 1, 2013

Duration: 3 years

DELIVERABLE 6.1

Calibration tool for sensory setup

Due date: month 7 (November 2013)

Lead contractor organization: TUM

Dissemination Level: PUBLIC

Contents

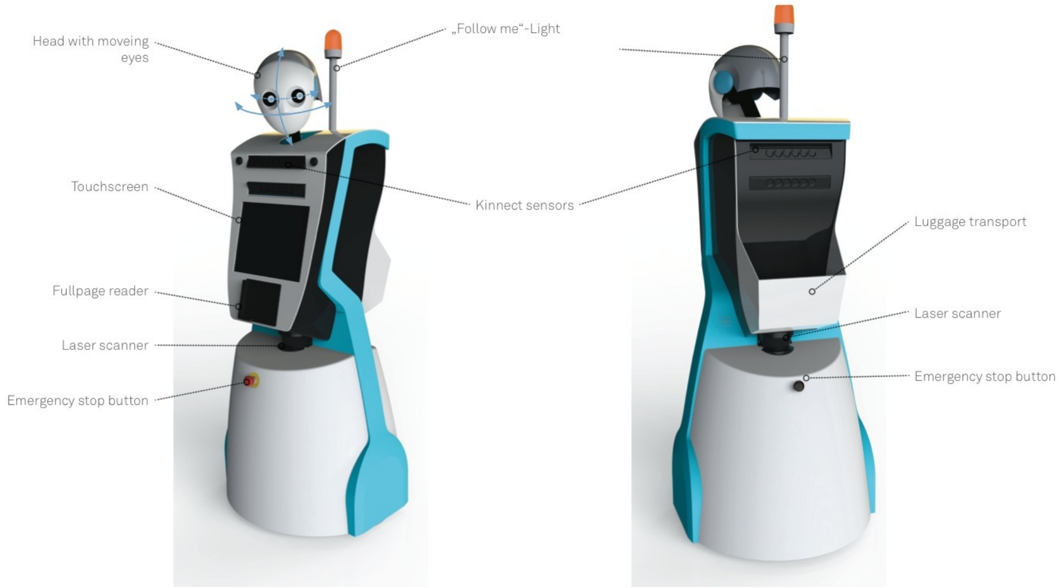

Figure 1: The sensors and their positions on the robot platform. This is an illustration, the final concept is still to be defined.

Abstract

We describe the calibration tool developed for the SPENCER project. In particular, this includes software tools to obtain parameters of the mappings between the different sensor coordinate frames, i.e. from one camera to another one, or from a laser to a camera. In addition to the tools provided in ROS, we reimplemented a method to find the mapping between a 2D laser frame and a 3D camera frame. We also present a Graphical User Interface (GUI) which we developed here, and we show reprojection errors obtained from our calibration method.

1 Introduction

As reported in D1.2, the SPENCER robot platform will be equipped with passive and active exteroceptive sensors. In particular, these are four RGB-D cameras, two monocular cameras – which will be combined into a stereo system, and two 2D laser range finders mounted parallel to the ground. Fig. 1 shows an illustration of the platform with the positions of the sensors, as they are designed at this time. Given this many sensors, which differ in modalities, position, orientation and noise level, it is very important to obtain accurate models with which the data acquired from one sensor can be related to the data acquired from another sensor. These models are obtained using sensor calibration, which is the aim of the work reported here.

Calibration between different sensor modalities requires different methods, and we will present the different methods we used here. Before, we give details of the hardware (i.e. sensors and robot platform) we used, as it is slightly different from the one that will be used later in the project.

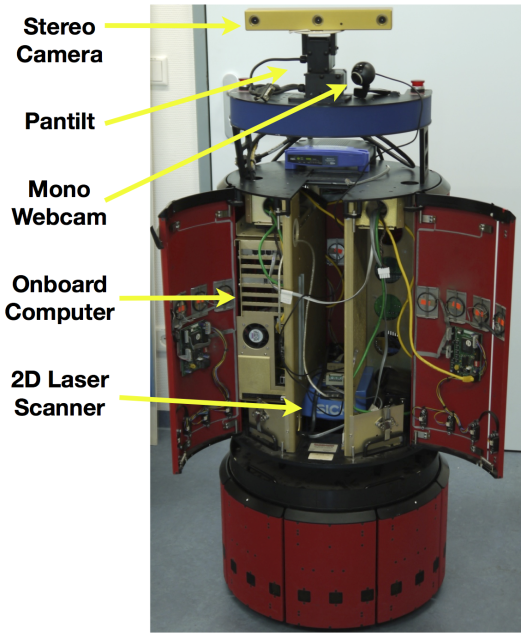

Figure 2: The Robot and the sensor setup used to develop the calibration tool. In addition to the sensors shown we also carried out experiments with an RGB-D sensor (see Fig.3), while the monocular webcam was not used here.

2 Used Hardware

For the development of our sensor calibration tool we first needed some realistic data set. As the final SPENCER platform is not yet available at this time, we choose to use two other kinds of data sources. The first one is a publicly available data set, as described in [1]. As a second source of data, we used a robot platform that was already available at TU Munich: a B21R robot that was equipped with sensors of the same modalities as there will be on the SPENCER platform. Fig.2 shows a picture of our test platform. Both the robot and the sensors shown are taken from earlier projects, but the setup here was designed to be as close to the final SPENCER setup as possible. The RGB-D sensor used can be seen in Fig.3.

To obtain data that can be mapped from one sensor modality onto another one, we decided to use a calibration pattern. We will give more details below. Here, we only note that the calibration pattern was printed onto a metal plate with $4mm$ thickness, and the dimensinos of the pattern are $0.9m \times 0.6m$.

3 Tasks of the Calibration Tool

From the sensor setup given in the previous section (see also Fig. 2 and 3), we see that there are basically four different tasks required from a calibration tool. These are

Figure 3: The Robot collects data using the stereo camera and the laser range finder. The calibration pattern is visible from both sensor modalities.

- **Intrinsic Calibration:** This refers to the task of obtaining an accurate model of the mapping between the 3D coordinate frame onto the 2D camera frame. This mapping is defined by parameters such as the focal length of the camera, the shift of the center point, and the distortion of the camera lense.
- **Extrinsic Camera-to-camera Calibration:** As there will be many cameras, we need a model of the mapping between different 3D camera frames. This is done using known correspondences between 3D points observed from different cameras.
- **Camera-to-laser Calibration:** Additionally, there will be a 2D laser range finder on the SPENCER platform. Therefore, we need to find a mapping from a given 3D point acquired with the laser into one of the camera frames. This is a substantially different task than the camera-to-camera calibration and requires a different calibration method.
- **Depth-to-color Calibration:** The RGB-D sensor measures both color and depth simultaneously, but to accurately assign one information to the other, a mapping from depth values to color pixels is required. This is similar to the camera-to-laser calibration, as it also relates depth with color, however it requires a slightly different method.

In the next section, we will give details on our solutions to each of these tasks.

4 The Developed Calibration Tool

In the literature, many different calibration methods have been proposed for different kinds of sensor setups and modalities, and the research in automatic calibration methods is still ongoing. In particular, calibration methods are currently investigated that do not require a predetermined calibration pattern or object. However, as the focus of the SPENCER project is not mainly on these research topics, we preferred reliable and established methods based on a standard checkerboard pattern for calibration (shown in Fig. 3) over a pattern-free approach. The advantage is that there is already a big set of software tools available, which we could adapt to our particular purpose. Nevertheless, it turned out that some given software needed to be reimplemented, as we describe below. Before that we give more details on the particular tasks of our calibration and on the already available tools we used.

4.1 Available Software Tools Used

As was decided by the SPENCER consortium, we will use the ROS system (www.ros.org) for the communication between all different modules on the robot and for many low-level robotic tasks. Therefore, we decided to use the calibration tool that is already available in ROS, namely the tool `cameracalibrator.py`. This program receives two camera images where the calibration pattern is completely observed in both images and computes the intrinsic and extrinsic parameters of the mapping. In our experiments, we found that the intrisic parameters obtained from calibration are not significantly different from the ones provide by the camera manufacturer. Therefore, we decided to use the tool only for obtaining the extrinsic parameters.

For camera-to-2D-laser calibration, there is no tool in ROS that can be used as easily as the camara calibrator. We therefore relied on some MATLAB software provided by Zhang and Pless [1]. However, as describe next this code needed to be reimplemented for our purpose.

4.2 Re-implementation of the Laser-to-Camera Tool

Fig. 4 shows the laser-camera calibration node. The left part of the window shows the image acquired from the camera. We can see that the software detected the calibration pattern correctly. On the right, we see the visualisation of the laser data. The red arrow in the center represents the central beam in the laser scan, and the red dots correspond to the positions where the calibration pattern has been detected. For the laser point detection to work, the middle laser beam has to hit the checkerboard. Then, a simple edge detection is used to find the left and right end of the pattern in the laser scan.

Our tool furthermore provides a functionality to save a pair of laser/camera data, where the pattern was detected in both modalities, and to add this information to a stack of data, where the calibration pattern is placed in different locations for every data frame. Once a user-defined number of camera/laser recordings is available, the calibration process is started by hitting the "Calibrate" button, visible in the bottom of the lower screen shot. When the calibration terminates, the result is visualized by projecting the laser points into the camera image. This is shown in Fig. 6.

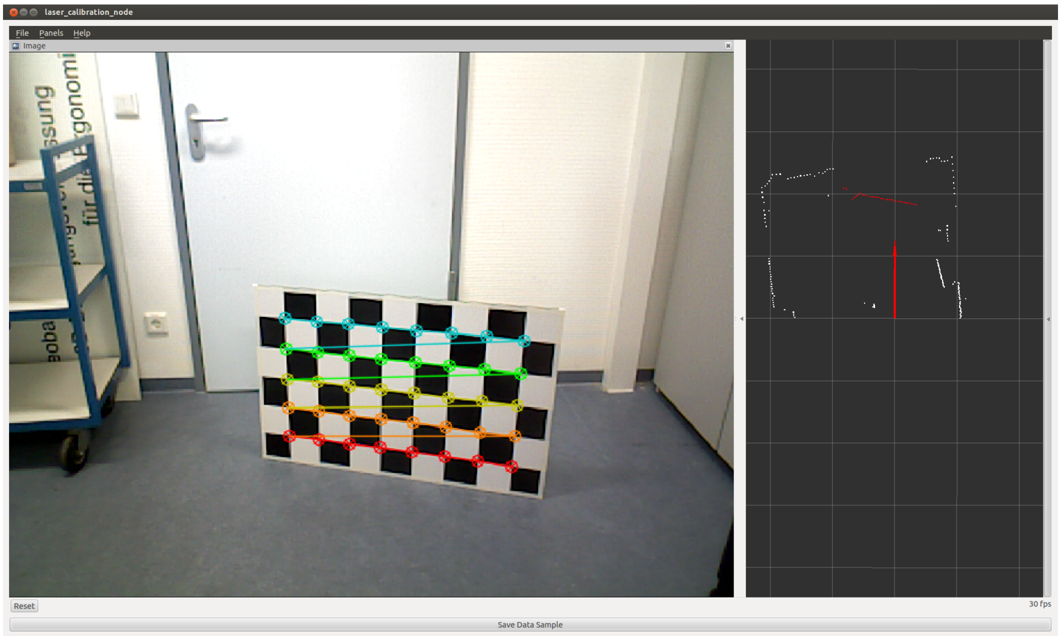

Figure 4: Screenshot of the developed laser-to-camera calibration tool. On the left side, the the detected calibration points in the camera data are visualized. On the right, we show the laser points detected as points on the pattern in red. The red arrow depicts the center beam of the 2D laser.

4.3 The Graphical User Interface

For convenience, we also developed a graphical user interface (GUI) of our calibration tool. The idea is to provide a common method to start the different calibration tools according to the sensor modalities that are to be calibrated. Fig. 5 shows a screenshot of our GUI. On the left, we see a photograph of our robot platform with arrows and descriptions of the important parts of the platform. This image (it is the same as shown in Fig.2) will be replaced by an image of the SPENCER platform, once it is available. On the right side of the GUI, there are edit boxes, where the user can specify the parameters of the calibration pattern. Also the sensor modalities (or topics as they are denoted in ROS) can be specified here. Then the data collection for the specified sensor topics starts by hitting the corresponding button.

4.4 Usage of the Calibration Tool

To make sure that our calibration tool is useful for other users within SPENCER, we provide a set of instructions here. More details of this will be available on the project Wiki.

Instruction 1 (calibration pattern): We recommend to use the calibration that we provide for the project. However, if a different one is used, it should be made sure that the margin around the checkerboard pattern is uniformly white. This is important for the corner points to be detected.

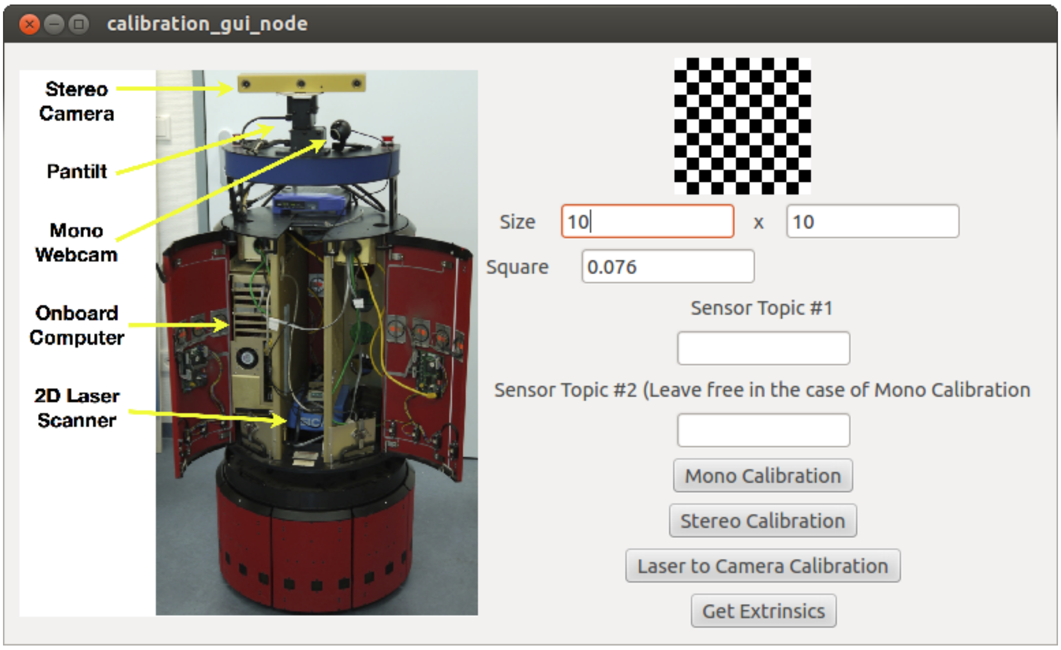

Figure 5: Screenshot of the developed Graphical User Interface. The left part shows a picture of the robot platform with details about the sensor modalities. In the final version, this picture will be replaced by a picture of the SPENCER platform. The right part consists of the interface to specify the dimensions of the calibration pattern and the particular modality used for calibration.

Instruction 2 (Camera Calibration): When calibrating with the pattern make sure that:

- the pattern is moved and NOT the camera (because of motion blur)
- the pattern is moved slowly
- the pattern is held horizontal
- the pattern is moved to every edge of the camera(s)
- the pattern is rotated about each axis
- there are at least 100 automatically saved poses before starting the optimization

For further information we refer to
$http://wiki.ros.org/openni\_launch/Tutorials/IntrinsicCalibration$ and
$http://wiki.ros.org/camera\_calibration/Tutorials/StereoCalibration$.

Instruction 3 (Camera-to-laser Calibration): Before starting this mode of the tool, the intrinsic parameters of the camera should be available, either from calibration or from the specification in the manual from the manufacturer. The following conditions should be met:

- The pattern has to stand on the ground to guarantee it is completely horizontal
- The central beam of the laser must hit the pattern. This is visualized in the tool, so it is easy to verify.
- The calibration pattern should be recorded for at least 20 different poses.

4.4.1 Depth-to-color Calibration

The calibration between depth and color modalities of an RGB-D sensor has mostly theoretical importance. In practice, it turns out that RGB-D sensors do not need to be calibrated. For more details we refer to $http://wiki.ros.org/kinect\_calibration/technical$. Therefore, we do not provide a particular module for the depth-to-color mapping of RGB-D sensors, but we leave the option to add this later in case it should be required.

5 Results

To assess the quality of our calibration tool, we provide visual and quantitative results. Fig. 6 shows the laser scan reprojected into the camera image. Note that the projected points are parallel to the ground and that they fit well onto the objects in the environment. This is a very good indication that the calibration is correct. In addition, we plot the residual error after each iteration of the optimization method (we use the Levenberg-Marquardt algorithm) in Fig. 7. The resulting error is 0.0346, which is comparable to the error reported in [1]. Finally, we report the epipolar error we obtained from the camera-to-camera calibration using over 100 point correspondences, which is 0.2 pixels. This can be considered as very accurate.

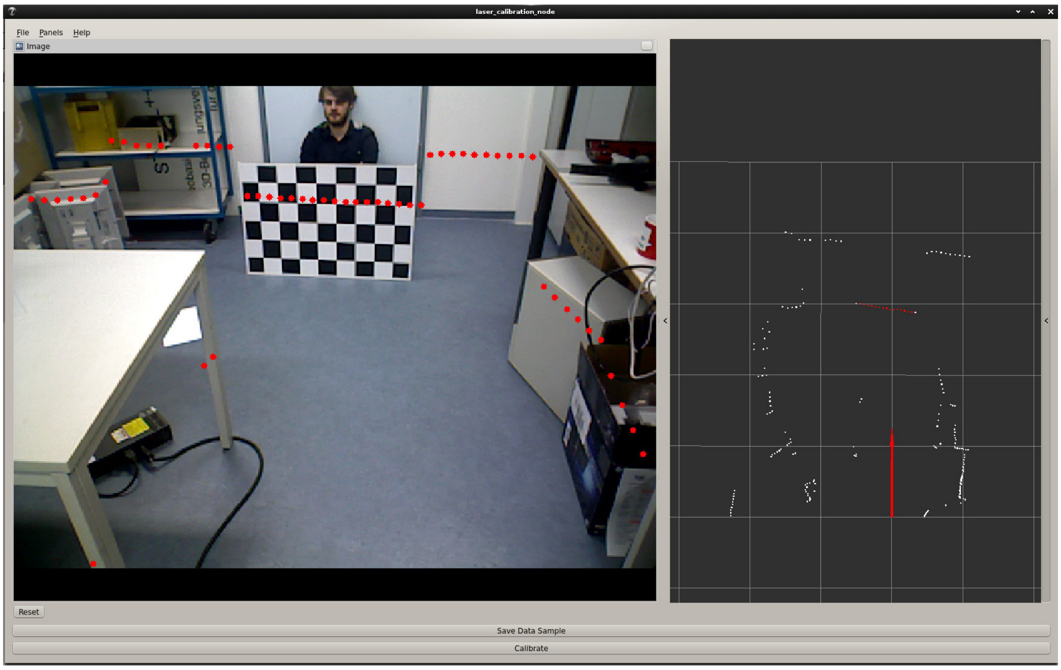

Figure 6: Reprojection of the laser readings into the camera image (left). This is to verify that the reprojection error is small.

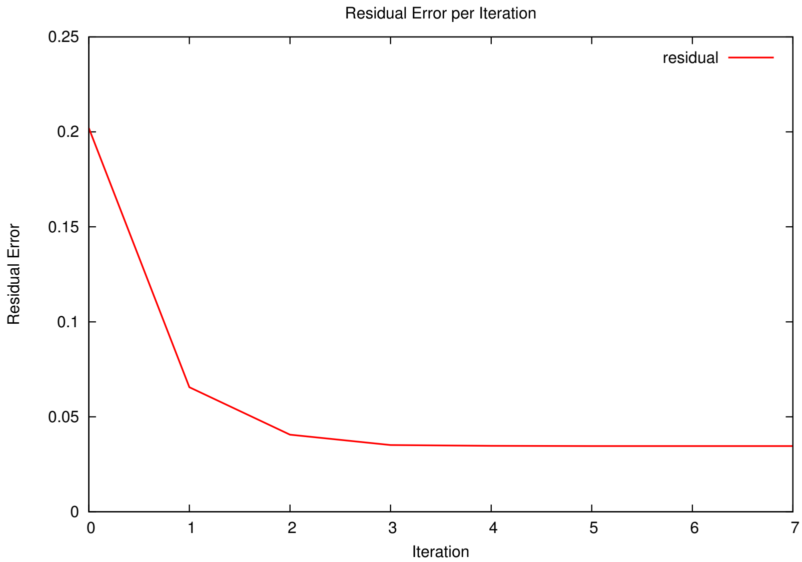

Figure 7: Residual error after each iteration of the optimization method used for laser-camera calibration. After only a small number of iterations, the method converged to a low error.

References

[1] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder. In *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2301–2306, 2004.