

Automatic Calibration of Multiple Cameras and Depth Sensors with a Spherical Target

Julius Kümmerle¹, Tilman Kühner¹ and Martin Lauer²

Abstract—In this work we present a novel approach for multi-sensor calibration that significantly outperforms current state-of-the-art. We introduce a new spherical calibration target which has major benefits over existing targets. Those are subresolution detection accuracy in both camera and depth sensor, view invariance and applicability to a wider range of sensor setups than current approaches. With our method a single person achieves high quality calibration in less than a minute. No preparations for setting up the environment for calibration is needed. Our method is fast, easy to use and fully automatic. We evaluate our method in simulation and show high accuracy with an error of less than 3 mm in translation and 0.1° in rotation on real data.

I. INTRODUCTION

Today, robots are deployed in more and more complex environments to support us in daily life. To perceive their environment they are equipped with a variety of sensors. By fusing data from multiple sensors, better robustness and performance can be achieved in tasks like object detection or localization. For successful sensor fusion an accurate calibration of the sensor setup is essential. Sensor fusion is especially effective with diverse sensors such as cameras and depth sensors that measure in different domains. But linking data from different domains is often non-trivial and therefore makes calibration of such a diverse sensor setup hard. In literature, different approaches for creating a link between camera image and depth data are presented. Most calibration methods make use of dedicated targets. A popular calibration target for camera to depth sensor calibration is a checkerboard [1], [2], [3]. Checkerboards have been used for camera-camera calibration for a long time and have proven to be suitable for many camera setups [4]. Corners are detected in the camera image to estimate the pose of the checkerboard by triangulation. In the point cloud of the depth sensor the checkerboard is modeled as a plane which provides a constraint for the calibration problem. Using multiple checkerboards can fully constraint the poses of the sensors. For robustness, the checkerboards should have well distributed orientations. But checkerboards with normals close to orthogonal to the viewing ray of the sensor cannot be detected accurately. This raises a fundamental limitation for practical use of checkerboard based sensor calibration. Furthermore, a known problem for many depth sensors

¹ The authors are with FZI Research Center for Information Technology, Research Department “Mobile Perception Systems”, Karlsruhe, Germany. The authors contributed equally to this work {kummerle, kuehner}@fzi.de

² Martin Lauer is with the Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Germany

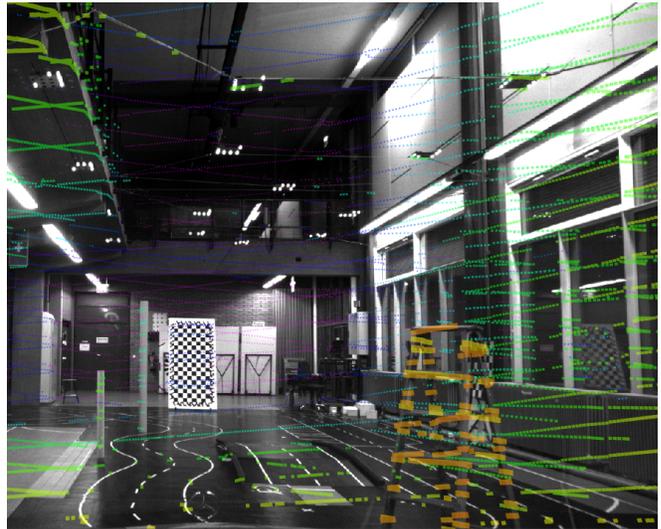
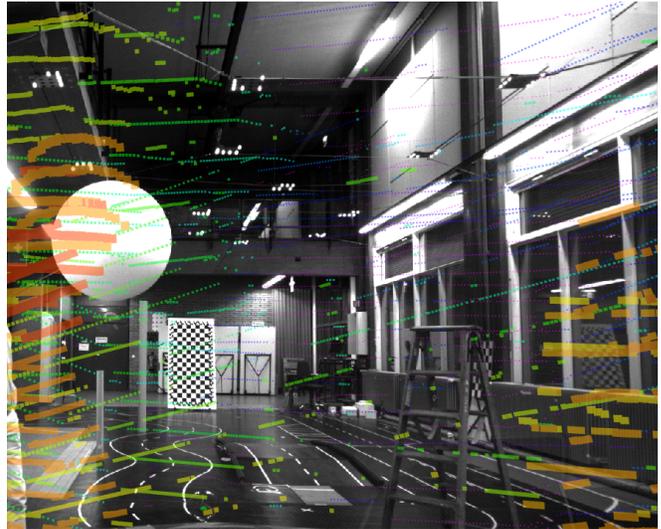


Fig. 1: Projection of LiDAR points into the camera image for an uncalibrated setup (top) and after calibration (bottom). The setup consists of two cameras and four LiDARs (see Fig. 10). The spherical calibration target can be seen in the top image.

(e.g. LiDAR) is that the reflectivity of the observed surface influences the actual measured depth [5]. Therefore, black and white areas on a checkerboard can have a significant offset in depth. Park et al. [5] uses a polygonal planar board with a homogeneous surface to prevent this effect. Vertices

of the polygonal board are detected in both the camera and the depth sensor. Since the resolution of the depth sensor is usually significantly lower than the resolution of the camera, the precision of the vertex detection based on 3D edges in the point cloud is the limiting factor of this calibration method. The problem of inaccurate feature point detection in depth data also occurs for planar targets with holes such as described in [6], [7]. Besides the afore mentioned target based methods there are also targetless alternatives. [8] detects reflections from LiDAR in a darkroom and minimizes their reprojection errors. In [9], [10] depth edges from range sensors are matched to intensity edges in camera images holding the risk of wrong associations since depth and texture are not guaranteed to match. [11] matches point clouds from LiDAR and stereo using ICP. This approach however is limited to setups with at least two cameras with overlapping fields of view and suffers from common problems of ICP e.g. wrong point associations. Another approach is based on the concept of maximizing the mutual information between measured surface intensities [12], [13]. This approach faces the problem of wrong associations for large initial calibration errors and is therefore only applicable for fine tuning [12], [14]. Furthermore, Schneider et al. [15] use a CNN to learn calibration of a camera-LiDAR setup. A downside of the approach is the need for a dataset with a groundtruth calibration for learning which practically does not exist. Moreover, the CNN has to be retrained for different sensor configurations.

To overcome the limitations of the existing approaches this paper introduces a new calibration method for multi-sensor setups with any number of cameras and depth sensors.

II. CALIBRATION TARGET

We recognize major benefits in using a calibration target instead of targetless calibration:

- Dedicated observation features which are designed for high precision detection
- Uniqueness in the calibration environment which is essential for outlier rejection
- Invariance to the scene such that the environment does not influence the calibration

On the search for an ideal calibration target for camera to depth sensor calibration, we focused on following requirements:

The target has to be easily detectable for both sensor types. It should be distinct from common objects in the environment to generate unambiguous observations. Unambiguity is important for reliable target detection in uncontrolled environments. The detection precision should be better than the resolution of the sensor. This is especially important for many depth sensors because of their low resolution. In the camera image the target should be detected with subpixel precision. The baseline is state-of-the-art checkerboard corner detection with subpixel refinement. The target should be of monochromatic color to prevent errors in depth measurements due to varying reflectivity. Further, an ideal target is non-stationary and lightweight. This allows to move the

target around the sensor setup which is in many cases more convenient than the other way round and allows calibration in confined spaces e.g. cars inside garages. Finally, the target should be cheap and easy to set up so that it can be used by everyone. A spherical target meets these requirements. For our experiments we choose an off-the-shelf white styrofoam sphere that can be purchased in a regular hardware store. The radius of the sphere measures 25 cm with precision in the range of one tenth of a millimeter. In the subsequent section we will describe how it can be used for calibration.

III. TARGET DETECTION

A. Sphere detection in camera images

We assume that an intrinsic calibration for the cameras is given. For the commonly used pinhole camera model, a sphere is projected onto an ellipse in the image plane. An ellipse is described by five parameters which are too many for efficiently applying standard methods like RANSAC [16] or Hough Transform [17]. Therefore, we warp the image onto a spherical screen which results in a sphere being projected onto a circle (intersection of the viewing cone with the spherical screen centered on the apex). Our circle detection consists of multiple steps: First, we detect circles using the Hough Transform on Canny edges [18]. Second, the circle position and radius are optimized in a least-squares sense using only those Canny edge pixels that lie close to the Hough circles. This step allows subpixel accuracy for the circle parameters (see chapter V-A.1). In a final step, false detections are rejected based on multiple filter criteria. We define inlier edge pixels as edge pixels which lie on the contour of the estimated circle. Pixels in close neighbourhood which are not inliers we call outlier edge pixels. Our filter criteria are:

- The radius of the estimated circle is within a certain range.
- We compute the ratio of inlier edge pixels to the maximum possible number of inliers. We filter out detections with a ratio less than a threshold which indicates an incomplete circle contour.
- We threshold the ratio of outliers to inliers to discard detections that are supported by clutter.

After filtering, the center position of the best circle detection is warped back to the input image, then its viewing ray is computed with known camera intrinsics. Further, the distance of the sphere d can be computed with its known radius R , the estimated radius $R_{\text{est,px}}$ and the radius of the spheric projection screen $R_{\text{screen,px}}$, both in pixels, as

$$d = \frac{R}{\sin\left(\frac{R_{\text{est,px}}}{R_{\text{screen,px}}}\right)}. \quad (1)$$

B. Sphere detection in range data

We assume that the range data is organized in an ordered structure with rows and columns. This applies for most commonly used range sensors such as LiDAR or the Kinect sensor. We detect depth discontinuities within each scan line

to find free standing segments. Segments that are significantly longer than half of the sphere’s perimeter are filtered out. Next, segments which are close together are associated and form a point cluster. A sphere model is fit to each of the point clusters by minimizing the squared geometric distance of points to the surface of the sphere. We define outliers to be points from the cluster that are further away from the sphere estimate than a threshold. If the radius of the estimation is close to the actual radius and the number of outliers is small then it is a valid detection.

IV. SOLVING THE CALIBRATION PROBLEM

A. Problem Definition

Given is a set of n sensors $\mathcal{S} = \{S_1, \dots, S_n\}$. The transformations $\mathcal{T} = \{T_1, \dots, T_n\}$ with $T_i \in SE(3)$ define the poses of the sensors in a reference frame. Without loss of generality, we assume the reference frame to be the sensor frame of S_1 . Let $\mathcal{O} = \{O_1, \dots, O_n\}$ be the observations of the sensors in the corresponding sensor frame. By linear interpolation, we form m time-synchronized pairs of observations $\mathcal{P} = \{P_1, \dots, P_m\}$ (see Fig. 2). We define our calibration problem as finding transformations \mathcal{T} which minimize the sum of squared distances of all observation pairs \mathcal{P} in the reference frame

$$\arg \min_{\mathcal{T}} \sum_{i=1}^m \text{dist}(P_i, \mathcal{T})^2. \quad (2)$$

B. Distance Measure

The distance measure in Eq. 2 depends on the types of observations within P_i . For a depth sensor we always represent the observation as a 3D point. Therefore, the distance between two observations from depth sensors is the Euclidean point-to-point distance. A camera observation can be interpreted as a ray when only using the 2D projection of the sphere’s center. By additionally using the known radius of the sphere we can estimate the distance of the sphere’s center (see Eq. 1) and thereby get a 3D point observation. We show in chapter V-A.1 that the depth estimation in camera is not as precise as with a depth sensor. Using point-to-point distance in this case is shown to be less robust than using point-to-ray distance. Therefore, we use point-to-ray distance for an observation pair of depth sensor and camera. A ray is defined by its origin \mathbf{p} and its unit direction vector \mathbf{v} in the form of $\mathbf{x}(s) = \mathbf{p} + s\mathbf{v}$, $s \geq 0$. The closest point on the ray to another point \mathbf{q} is at $\mathbf{x}(s_q)$ with $s_q = (\mathbf{q} - \mathbf{p})^\top \mathbf{v}$ thus the point-to-ray distance is $\|\mathbf{q} - \mathbf{x}(s_q)\|$ if $s_q \geq 0$ and $\|\mathbf{q} - \mathbf{p}\|$ if $s_q < 0$.

The third case is an observation pair of two cameras. Ray-to-ray distance is more sensitive to noise due to ray geometry. Furthermore, ray-to-ray distance converges to the trivial solution of all cameras being in one place in case of the setup consisting of only cameras. Therefore, we use point-to-point distance for observation pairs from cameras.

C. Observation Synchronization

Calculating the distance between two observations only makes sense if they have identical timestamps. Especially when using rolling shutter sensors, such as rotating LiDAR, the observations will not be time-synchronized so that an additional synchronization step is needed. The process of synchronization is depicted in Fig. 2. For each timestamp of a real observation all observations from the other sensors are linearly interpolated. We only interpolate in-between observations that are not further apart in time than the known cycle duration of that sensor.

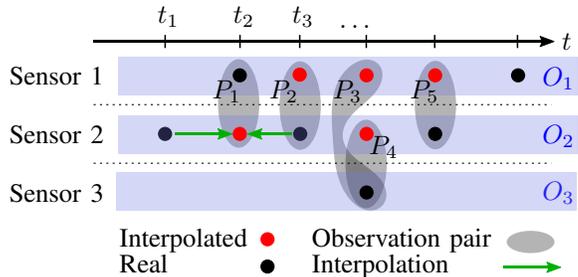


Fig. 2: Time synchronization of observations. Interpolation is shown in detail for $t = t_2$.

D. Optimization

We solve the minimization problem (Eq. 2) by a numeric solver [19]. The initialization of all sensor poses is random.

E. Robustification

To achieve robustness against outliers from the detectors we use RANSAC [16]: Calibration is performed multiple times on randomly sampled minimum subsets of observation pairs. For all observation pairs \mathcal{P} the distance between both observations is computed. Observation pairs having a distance larger than a threshold are outliers. A final calibration is performed only on inliers of the best calibration run carried out previously. To further increase robustness, we use a 3D constant velocity motion model for filtering out false detections.

V. RESULTS AND EVALUATION

We evaluate our calibration framework in multiple stages and separately analyze error propagation in the detectors and the solver (see Fig. 3):

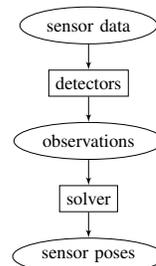


Fig. 3: Calibration dataflow overview.

First, we simulate raw sensor data which is fed to the detectors (V-A.1). Thereby, we learn how errors in raw data affect the quality of the observations. Second, we evaluate how the solver reacts to different types of error sources (V-A.2). Third, real sensor data is used and the calibration results are analyzed in two experiments with different sensor setups (V-B).

A. Simulation

We simulate a sphere which moves along a trajectory with a velocity of 0.4 m/s. We generate three different groundtruth trajectories for simulation. The distance to the sensors varies between 2-8 m. The simulated sensor setup consists of two cameras and two laser scanners with 16 scan lines each (see Fig. 4). The cameras have a resolution of 2000 by 974 and a focal length of 1222 px. All sensors run at 10 Hz. We

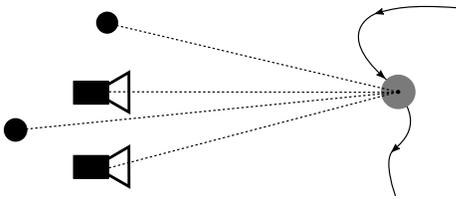


Fig. 4: Simulated sensor setup with two cameras and two laser scanners.

simulate twelve datasets of one to two minutes. For each we perform 50 calibrations with randomly initialized sensor poses.

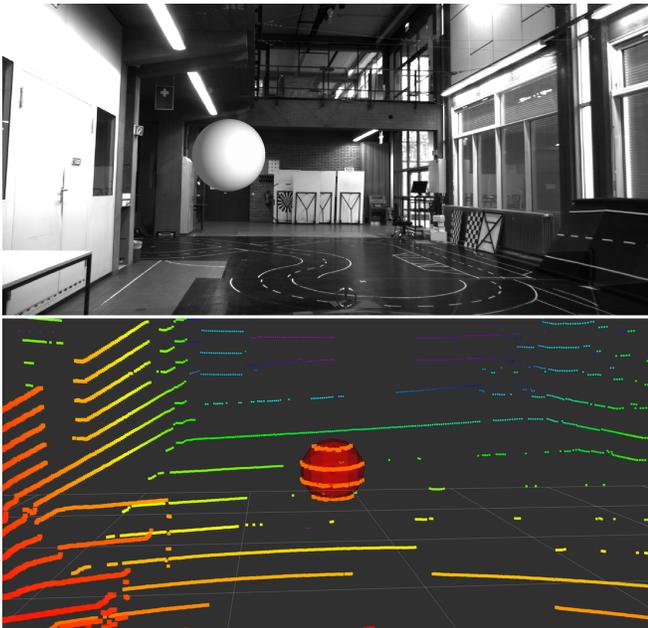


Fig. 5: Simulated sensor data with virtual sphere for camera (top) and LiDAR (bottom).

1) *Evaluation of Detector Accuracy:* We simulate raw sensor data as shown in Fig. 5 to find out the accuracy of the detectors. For camera images we render a sphere

on a background image. For simulation of range data we use a pointcloud recorded by a real sensor. By ray tracing we get the intersections of the virtual laser beams with the groundtruth sphere. The following error sources are simulated and analyzed:

- Cluttered background
- Lighting
- Camera defocus
- Noise in pixel intensities
- Error in focal length
- Noise in range measurements

As a baseline for camera an ideal white sphere in front of a black background is simulated. First, we add real background images and lighting to put the sphere into a realistic environment. Next, camera defocus is simulated by blurring the contour of the sphere in the image with a Gaussian filter. By comparison with real data we came up with a Gaussian kernel standard deviation of $\sigma_{\text{blur}} = 0.5 \text{ px}$. Zero mean Gaussian noise is added to the intensity of each pixel. We found that a standard deviation of $\sigma_I = 4$ provides realistic results for images with 8-bit intensity resolution. Next, an erroneous focal length of 0.3 % is simulated. This is a realistic result from good intrinsic calibration [20]. For LiDAR data we add zero mean Gaussian noise to each range measurement with a standard deviation of $\sigma_\delta = 12.5 \text{ mm}$ which was also determined experimentally.

For evaluation we calculate the distance from each detection to its corresponding groundtruth point as explained in chapter IV-B. The results of the simulation are summarized in Fig. 6. The simulations show that calibrating in front of a

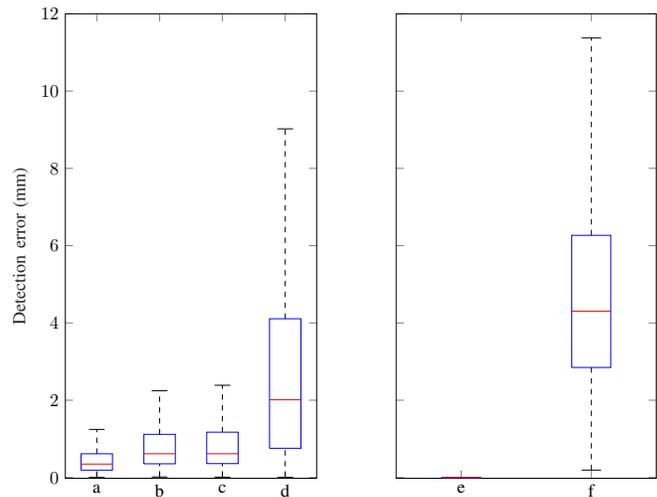


Fig. 6: Effect of different error sources on sphere detection. Camera: (a) ideal sphere, (b) realistic background image and lighting, (c) background, lighting, defocus and noise, (d) 0.3 % error in focal length. LiDAR: (e) ideal depth measurements, (f) noisy depth measurements.

black background is most accurate. A realistic environment results in the detection error to approximately double. Additionally adding effects like noise and defocus only leads

to subtle deteriorations. The dominant error in the camera detector is caused by an erroneous focal length. Therefore, we see no need to calibrate in front of a black background. Sphere detection on perfect depth data leads to zero detection error as we expect. The sphere detection with our depth sensor has a median position error of approximately 4.5 mm.

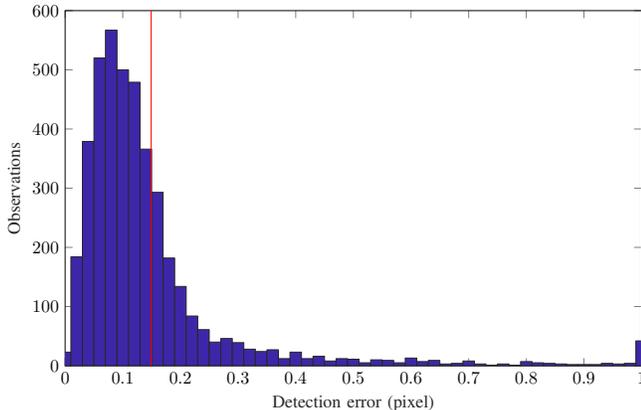


Fig. 7: Histogram of reprojection errors for sphere in realistic scene. The mean error is marked by a red line.

For comparison to checkerboard corner detection we calculate reprojection errors of the sphere detections in the camera image. Fig. 7 shows that we reach an accuracy comparable to state-of-the-art checkerboard corner detectors [2]. Simulating a realistic scenario, we achieve a mean error of 0.15 px.

We also considered estimating the 3D center point of the sphere via Eq. 1. However, for our setup better results were achieved when using point-to-ray distance for camera-depth-sensor observation pairs in the optimization problem. This can be explained by a higher depth precision of the LiDAR measurements compared to radius-based depth measurements in the camera image. By using point-to-ray distances we effectively rely only on depth from LiDAR. Results from calibrations using both distance measures are shown in Fig. 8.

2) *Evaluation of Sensitivity:* We analyze sensitivity of the calibration problem by simulating the following error sources:

Noise on the position: We simulate errors in detection of the sphere by adding zero mean Gaussian noise to its ground truth position. For cameras we set the standard deviation to $\sigma_{\text{pos,cam}} = 5$ mm and for LiDAR we use $\sigma_{\text{pos,lidar}} = 10$ mm. We set these parameters as upper estimations based on the evaluation of detector accuracy (V-A.1). From the noisy position we generate a ray or a point observation depending on the simulated sensor type.

Observation interpolation: Observations in observation pairs need to have identical timestamps. Due to untriggered sensors or rolling shutter effects, this cannot be guaranteed and therefore must be obtained by interpolation (chapter IV-C). We analyze the errors from interpolating by triggering the virtual sensors at different times.

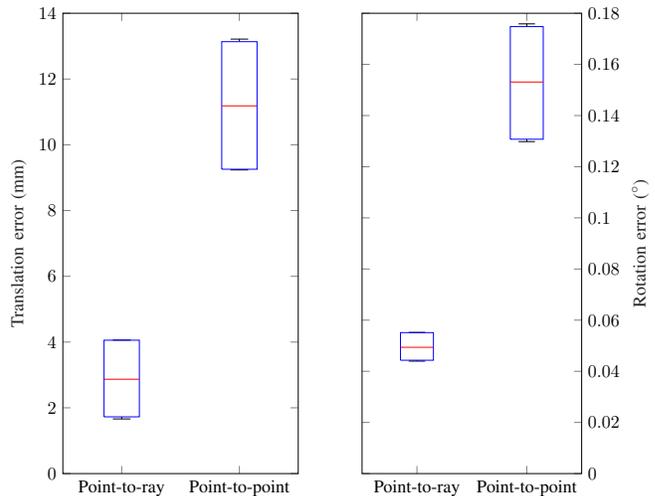


Fig. 8: Comparison of calibration results between point-to-ray and point-to-point distance for camera-depth-sensor observation pairs.

Erroneous timestamps: Professional sensor setups are usually time synchronized and provide exact timestamps for their measurements. We want to analyze the effect of erroneous timestamps which occur in cheap and provisional setups. We add Gaussian noise to the timestamps with a mean of up to $\mu_{\text{ts}} = 30$ ms and a standard deviation of $\sigma_{\text{ts}} = 10$ ms. We determined these parameters based on an experiment with a setup consisting of a camera and a LiDAR for which the data is stamped when arriving at the main system.

Biased range measurements: The range measurements of some laser scanners are affected by the reflectivity of the scanned object [5]. Further, the sensor might have imperfect depth calibration or measurement drift which also results in biased range measurements. To simulate this effect a constant offset of $\Delta\rho = 10$ mm along the line of sight is added to the detections. The parameter is set based on a depth measurement analysis for a common LiDAR [21].

Erroneous focal length: We also analyze the influence of an error in focal length on the calibration result. We simulate an error of 0.3%.

The results for the error sources are shown in Fig. 9. The estimated sensor poses are compared to the known groundtruth poses and the error is split up into translation error e_t and rotation error e_r .

$$e_t = \|t - t_g\| \quad (3)$$

$$e_r = \angle(R^{-1}R_g), \quad (4)$$

with t denoting the translation part and the rotation matrix R . Groundtruth is indicated by index g .

First, we evaluate the effect of zero mean Gaussian noise on the observations' position (Fig. 9a). This error depends on the quality of the sensor data and the detectors. To validate the noise parameters we simulate realistic raw data and used our detectors to create observations (Fig. 9b). By comparing both results, we show that the assumed position noise on

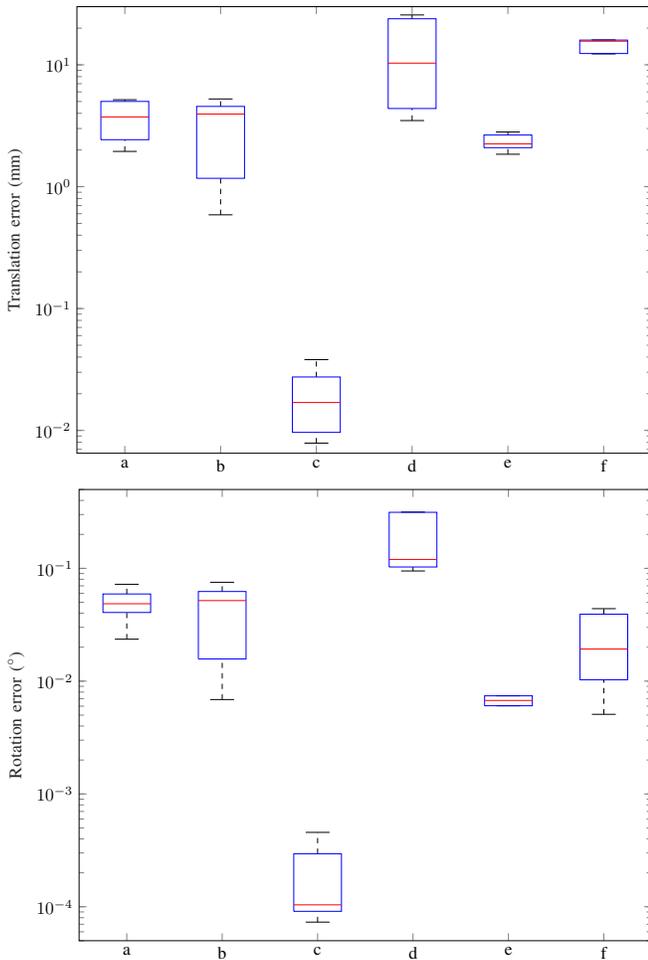


Fig. 9: Influences of different error sources on calibration results: (a) zero mean Gaussian noise in observations' position, (b) realistic raw data with detectors, (c) interpolation, (d) erroneous timestamps, (e) biased range measurements, (f) erroneous focal length

the observations is suitable for our setup. Further, we learn that noise on the observations' position is noticeable in the calibration result but with our data and detectors it is not the dominating source of errors. The least significant error source we analyze is interpolation. The pose error is less than 0.1 mm in translation and less than 0.001° in rotation. This shows that when moving the sphere with moderate velocity errors from linear interpolation of the observations have minor influence on the calibration result. A significant error is introduced by erroneous timestamps. The median pose error in translation is more than 10 mm and more than 0.1° in rotation. By slowing down the sphere's movement one can reduce this error. As mentioned before, this type of error can be prevented by using a synchronized sensor setup. Next, we notice that biased range measurements only introduce a small error. Errors in focal length show a significant effect on the calibration result. It is in the range of errors from noisy timestamps.

B. Experiments on real data

A known problem for evaluating calibration on real data is that there is usually no groundtruth available. Therefore, we perform two different experiments:

First, we analyze repeatability of calibration results on different datasets which were recorded by a setup of four Velodyne Puck and two cameras (Fig. 10). We vary the

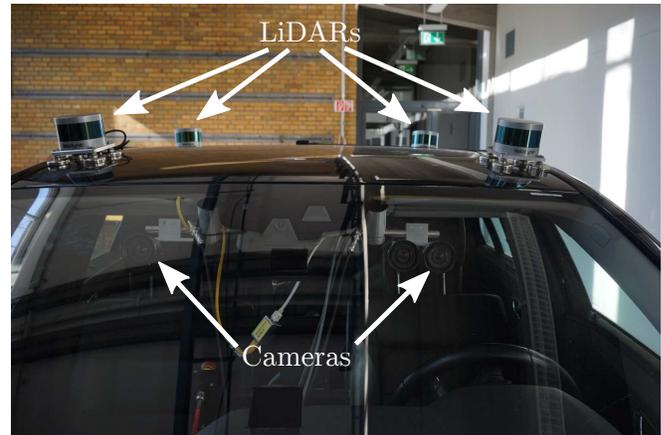


Fig. 10: Sensor setup with two cameras and four Velodyne Puck.

scene, the trajectory of the sphere and the duration of the recording. We calibrate indoors and outdoors under different lighting conditions and in front of changing background. The trajectory varies between helical, zick-zack and random smooth movements. Based on eight datasets on which we run 50 calibrations each, we calculate average transformations for the sensors. The average translation is calculated by averaging the translation vectors and the averaged rotation is determined as described in [22]. The differences between

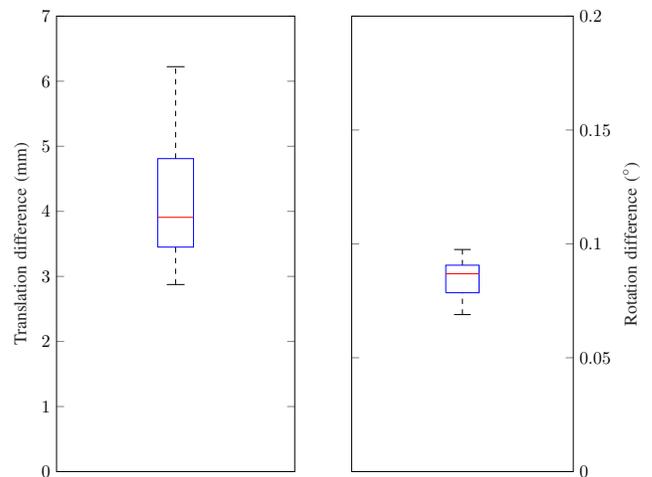


Fig. 11: Results of the repeatability analysis for a sensor setup with two cameras and four Velodyne Puck.

the averaged transformations and each calibration run is visualized in Fig. 11. The median translation difference is 3.9 mm and the median rotation difference is 0.087° . This

proves high repeatability of the method and robustness to different scenes and trajectories.

The second experiment on real data uses another setup of a single camera and a Velodyne Puck (see Fig. 12). The



Fig. 12: Sensor setup with translation and rotation table.

camera is fixed but the LiDAR is mounted on a moving and rotating table. The setup allows controlled translation in the direction of the baseline and rotation around the symmetry axis of the Velodyne. The procedure for the experiment is as follows: First, we run the calibration on a starting state S . In state S , the sensors have a baseline of roughly 180 mm. Then, a known translation or rotation is applied. We rerun the calibration for the new state. Finally, we calculate the translation and rotation difference to the calibration result of state S . This is the estimate for the manually applied pose change of the LiDAR. As in the previous experiment, we calculate mean transformations based on multiple recorded datasets. In Fig. 13, the results for the starting state T_0 , a state T_t with 120 mm increased baseline and a state T_R with 20° rotation are shown. Again, we notice high repeatability over all states. The repeatability is comparable to the results of the previous experiment with a different sensor setup. Comparing the result of state T_0 to the result of state T_t gives the estimate of relative translation (Fig. 13 bottom left). The median translation difference is 122.7 mm which is 2.7 mm off from the applied translation. The relative rotation between state T_0 and state T_R is estimated with a mean rotation difference of 19.905° which is 0.095° off from the applied rotation (Fig. 13 bottom right). This experiment shows that our calibration framework provides accurate results in both translation and rotation.

VI. CONCLUSIONS AND FUTURE WORK

We presented a new approach for calibrating multisensor setups of cameras and depth sensors. Our main contribution is the introduction of a spherical calibration target. By simulation and in experiments with real data we proved that our target is superior to currently used targets in multiple ways: We showed that with simple methods a sphere can be detected at least as accurate as corners on checkerboards in camera images. Furthermore, our sphere detector for range data shows high accuracy even for low resolution LiDAR data. Our calibration target can be detected from every viewing angle which makes calibration of complex sensor

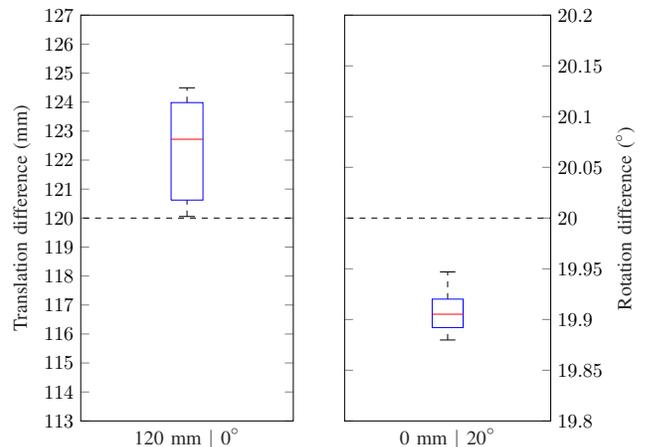
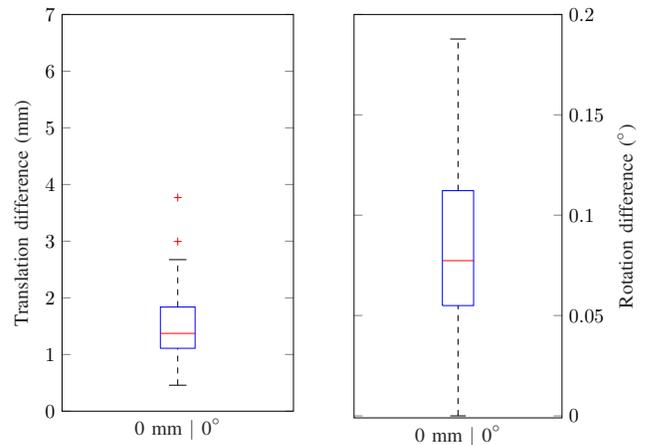


Fig. 13: Results for estimating manually applied pose differences.

setups possible e.g. cameras facing each other. Our calibration framework achieves higher accuracy than current state-of-the-art techniques while being computationally cheap and easy to use. We further evaluated sensitivity to multiple error sources and showed the robustness and repeatability of our method. The applicability was proven on real data with different setups and datasets.

For the future, we want to further increase accuracy of our camera detector by an dedicated refinement step based on subpixel edges. Also, we plan to improve the detector for depth sensors by using a maximum likelihood estimator that considers specific sensor characteristics. Additionally, we plan to extend our calibration framework by adding detectors for further types of sensors. In a next version of our calibration framework we will estimate camera intrinsics since we identified focal length to have a significant influence on the calibration result.

REFERENCES

- [1] Qilong Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2301–2306.
- [2] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic Camera and Range Sensor Calibration using a single Shot," *IEEE International Conference on Robotics and Automation*, pp. 3936–3943, 2012.
- [3] A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," *Proceedings of the 2010 Australasian Conference on Robotics & Automation*, 2010.
- [4] T. V. Strauß, "Kalibrierung von Multi-Kamera-Systemen : Kombinierte Schätzung von intrinsischem Abbildungsverhalten der einzelnen Kameras und deren relativer Lage zueinander ohne Erfordernis sich überlappender Sichtbereiche," Karlsruhe, [2015]. [Online]. Available: <http://dx.doi.org/10.5445/IR/1000051877>
- [5] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3D LIDAR instruments with a polygonal planar board," *Sensors (Switzerland)*, vol. 14, pp. 5333–5353, 2014.
- [6] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of RGB Camera With Velodyne LiDAR," *Comm. Papers Proc. International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, pp. 135–144, 2014.
- [7] S. A. Rodriguez F., V. Frémont, and P. Bonnifait, "Extrinsic calibration between a multi-layer lidar and a camera," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 214–219, 2008.
- [8] J. Gräter, T. Strauss, and M. Lauer, "Photometric laser scanner to camera calibration for low resolution sensors," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1552–1557, 2016.
- [9] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, pp. 2862–2866, 2016.
- [10] J. Levinson and S. Thrun, "Automatic Online Calibration of Cameras and Lasers," *Robotics: Science and Systems IX*, 2013. [Online]. Available: <http://www.roboticsproceedings.org/rss09/p29.pdf>
- [11] M. Hassanein, A. Moussa, and N. El-Sheimy, "A new automatic system calibration of multi-cameras and lidar sensors," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 2016-January, no. July, pp. 589–594, 2016.
- [12] 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," *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 2053–2059, 2012.
- [13] Z. Taylor and J. Nieto, "Automatic calibration of lidar and camera images using normalized mutual information," *IEEE Conference on Robotics and Automation (ICRA)*, 2013.
- [14] H. J. Chien, R. Klette, N. Schneider, and U. Franke, "Visual odometry driven online calibration for monocular LiDAR-camera systems," *Proceedings - International Conference on Pattern Recognition*, pp. 2848–2853, 2017.
- [15] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal Sensor Registration Using Deep Neural Networks," 2017.
- [16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: <http://doi.acm.org/10.1145/358669.358692>
- [17] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972. [Online]. Available: <http://doi.acm.org/10.1145/361237.361242>
- [18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.
- [19] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>.
- [20] Z. Zhang, "A Flexible New Technique for Camera Calibration (Technical Report)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2002.
- [21] C. Glennie and D. D. Lichti, "Temporal stability of the Velodyne HDL-64E S2 scanner for high accuracy scanning applications," *Remote Sensing*, vol. 3, no. 3, pp. 539–553, 2011.
- [22] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging Quaternions."