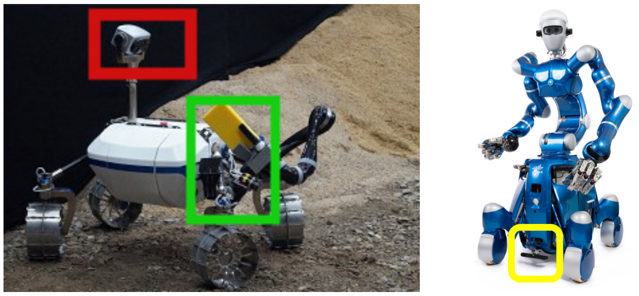
A Method for Hand-Eye and Camera-to-Camera Calibration for Limited Fields of View

Christian Nissler[1] Zoltán-Csaba Márton[1] Hannes Kisner[3] Ulrike Thomas[3] Rudolph Triebel[1,2]

Abstract— **In classical robot-camera calibration, a 6D transformation between the camera frame and the local frame of a robot is estimated by first observing a known calibration object from a number of different view points and then finding transformation parameters that minimize the reprojection error. The disadvantage with this is that often not all configurations can be reached by the end-effector, which leads to an inaccurate parameter estimation. Therefore, we propose a more versatile method based on the detection of oriented visual features, in our case AprilTags. From a collected number of such detections during a defined rotation of a joint, we fit a Bingham distribution by maximizing the observation likelihood of the detected orientations. After a tilt and a second rotation, a camera-to-joint transformation can be determined. In experiments with accurate ground truth available, we evaluate our approach in terms of precision and robustness, both for hand-eye/robot-camera and for camera-camera calibration, with classical solutions serving as a baseline.**

I. Introduction

Whenever a robotic system is to interact autonomously and safely with its physical environment, a mapping between the coordinate frame defining the position of the actuators and the sensors is required. For example, in the case where a robotic arm autonomously grasps an object that is sensed by a camera, it is crucial to know the mapping from the joint positions of the arm to the 3 rotational and 3 translational degrees of freedom (6DOF) pose of its end effector in the camera coordinate frame. To obtain such a mapping, a so-called *hand-eye-calibration* process is needed (more generally, a robot-camera calibration), and in the literature there are a number of techniques presented that can achieve this (see, e.g. [1], [2]). In such methods, images from a precisely known calibration pattern are recorded at different configurations of the robot. From a set of robot poses and detected correspondence pairs between pixels and positions on the calibration pattern, e.g. the corners on a checkerboard, the standard approach finds, apart from the intrinsic camera parameters, extrinsic transformation parameters by optimizing a set of non-linear equations. To obtain a well-conditioned problem, a large range of robot configurations must be available (ideally, covering two half-circles in front of the camera that are perpendicular to each other [3]). Often, however, it is hard or even impossible to reach certain configurations. Furthermore, the standard method gives poor results if the positioning of the robot arm is imprecise.


(a) The DLR Lightweight Rover Unit (LRU) [4] with the pan-tilt unit and base cameras marked.

(b) Justin [5] with the base camera marked.

Fig. 1. Two examples of complex robotic systems where traditional hand-eye and/or camera-camera calibration is difficult to apply.

Visual servoing can be used to cope with an imprecise hand-eye calibration [6], [7], but this brings the additional problem of more or less continuous occlusion-free visibility and a higher computational complexity [8]; as for self-tracking [9]. Therefore, in this paper we propose a different approach. We use detections of AprilTag fiducial markers [10], or any other 6DOF pose estimates of reference objects. These can either be attached to the moving robot and observed by a static camera or, vice-versa, where the camera moves and the targets are static. AprilTags are a good alternative to checkerboards when multiple of them need to be used and distinguished in a small area. Thus different tags can be observed by the same camera during large rotations, and/or different cameras can observe different tags, whereas one or more checkerboard patterns are cumbersome to use. When attached to a robot arm (e.g. on the LRU's, see Fig. 1a), small tags can be even left in place, allowing to check the calibration online and recalibrate if needed.

Our approach has three major benefits over existing methods: First, it works with a comparably low number of detections. Second, it is easier because no expert knowledge of good camera to calibration pattern movements is needed. And third, we can use our approach also to calibrate the extrinsics between a camera and a kinematic chain, or two cameras mounted without overlapping fields of view when no joint encoder information is available, requiring only two rotational joints. This paper builds on [11], where instead of testing the method on the Justin robot, where traditional calibration was not possible (see Fig. 1b). We obtained various ground truth estimates and evaluated our method

[1] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Oberpfaffenhofen, Germany `name.lname@dlr.de`

[2] Dep. Computer Science, Technical University of Munich, 85748 Garching, Germany `rudolph.triebel@in.tum.de`

[3] Robotics and Human Machine Interaction Lab, Technical University Chemnitz, 09126 Chemnitz, Germany `ulrike.thomas@etit.tu-chemnitz.de`

based on that. Moreover, we performed experiments with different cameras and performed camera-to-camera extrinsic calibration, comparing our method with a Simultaneous Localization and Mapping (SLAM) based one as well.

II. Related Work

Besides the mentioned classical supervised (i.e., with a known calibration pattern, placed according to expert knowledge) hand-eye calibration methods [1], [2], there exist several other methods to obtain an intrinsic or extrinsic calibration with limited or without supervision.

For the intrinsic calibration of depth sensors, Teichmann *et al.* present a method called CLAMS (calibration, localization and mapping simultaneously) [12]. The authors use the fact that certain intrinsic parameters of depth sensors are *myopic*, meaning the error gets larger with increasing distance to the sensor. The main idea is to collect raw sensor data over time, then iteratively estimate the ego-motion of the sensor during this recording by SLAM methods, compute a near-sight map from which training examples are extracted, with which a model for the distance-variant parameters can be estimated.

For extrinsic unsupervised calibration Kümmerle *et al.* [13] propose an approach based on Graph-SLAM methods, where the camera position explicitly gets incorporated. This however requires information about robot odometry and an initial guess about the forward kinematics parameters of the robot is needed. In the case of robots with imprecise forward kinematics, like the iCub, a camera to robot calibration can also be learned through experience. Leitner *et al.* [14] propose an artificial neural network that is trained by ground truth positions provided by an industrial robot, that are detected by the iCub's cameras and reached for by the hands. While extremely deep neural nets can be efficiently learned with today's GPUs, the training required by another robot makes only sense in such difficult to control robots like the tendon-driven iCub. Levine *et al.* [15] follow another approach to use deep learning methods in order to overcome hand-eye coordination inaccuracies for grasping objects. By training a large convolutional network with many grasp attempts, robots can learn the spatial relationship between gripper and objects. Thereby grasps with a high success rate can be achieved without the need of a calibration. However, much data is needed and the solution is robot configuration specific, thus challenging to transfer to other target poses.

AprilSLAM is another approach which performs camera pose estimations[1]. It is originally developed by Kumar Robotics with the aim to estimate camera pose from one AprilTag without initilization steps. The library uses the AprilTag algorithm based on [10], while the mapping system is build on the GTSAM library [16] and is available as a ROS-Package. The method detects an AprilTag in the first provided image that confirms the predefined AprilTag parameters. However, one can define only one tag-family and tag-size. Therefore, all other tag-families are not considered. This leads to problems if there are AprilTags with the same ID and tag-family. Moreover the program only considers AprilTags that are in the center of the image which has to be known during the image acquisition process. With the detected AprilTag the approach associates the 2D points in the image with 3D points in the world frame. The first detected AprilTag is set as origin. The approach tries to find the relative pose by using further detected tags. Therefore, at least one AprilTag has to be continuously visible in two consecutive images. Furthermore the algorithm experiences problems if two consecutive images have a big relative difference between their poses, in which case the SLAM approach can probably not converge.

III. Calibration by Modeling Rotational Joints

The basic idea of our approach is to track visible features, in our case AprilTags fiducial markers [10], while performing a rotation around two axes with the robot. Each visible AprilTag yields a 6D transformation from the camera's frame of reference to the marker's frame of reference. If a marker is visible in consecutive camera images a relative rotation is calculated (see Section III-B). The key idea of our approach is to first rotate the camera only around one axis and collect all detected orientation vectors represented as unit quaternions. Then, we fit a *Bingham* distribution [17] into this data, from which the most consistent axis of rotation can be estimated. This axis corresponds directly to the physical rotation axis of the robot. The centers of rotations are obtained by jointly optimizing the centers of multiple concentric circles that are formed by the tag's path in the camera's image. To obtain a second axis, and thus build a frame of reference describing the rotated joint of the robot, we perform a second rotation of the camera, fit again a Bingham distribution, and determine the intersection of the two estimated 3D lines.

In the case of the humanoid robot platform Justin, a camera is attached to the robot, see Fig. 1b, and is rotated around its vertical axis. By performing several rotational motions of the robot (e.g. tilted in different directions), we can estimate the intersection of the rotational axes, which corresponds to the center point of the base [11]. But also the other way around is possible e.g. in a mobile platform like the LRU, see Fig. 1a. If markers are attached to the arm, which is then rotated in front of the camera, the rotation axis can be estimated in the same way. If this rotation is repeated under a different pose, an intersection of the rotation axes can be calculated. If the different poses only differ by a rotation of one adjacent joint, the intersection point found has a physical correspondence. If the joints lie on the axis around which the rotation occurred, the intersection point will be the joint adjacent to the joint which rotated. Even if those joints do not lie on the rotational axis, the intersection point corresponds to a known "virtual" point of the robot, which can be obtained e.g. by CAD data.

A. Bingham Distribution for ML Estimation of Rotations

We describe rotations with unit quaternions, utilizing their computational advantages in comparison to other represen-

[1] https://github.com/ProjectArtemis/aprilslam

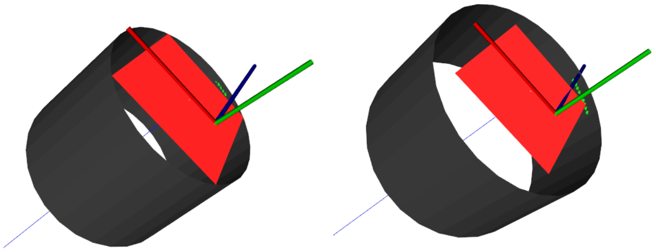


Fig. 2. Example results of estimated rotation centroids and axes, depicted as 3D cylinders and the plane of rotation of the samples around the centroid (red), as well as the rotation axis (thin blue line) and the found markers in green. The coordinate frame shows the frame of reference of the camera.

tations and their lack of singularities [18]. The Bingham distribution describes an antipodally symmetric distribution on a sphere [17]. This makes it very useful to describe a distribution of 3D rotations with unit quaternions, which represent a double mapping of $SO(3)$ with two antipodally symmetric half-spheres (i.e., $\mathbf{x}$ and $-\mathbf{x}$ represent the same rotation). The general probability density function of a Bingham distribution on a hyper-sphere is given by

$$B(\mathbf{x}; K, V) := \frac{1}{F(\kappa_1, \kappa_2, \kappa_3)} \exp\left(\sum_{i=1}^{3} \kappa_i (\mathbf{v}_i^T \mathbf{x})^2\right) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^4$ with $\|\mathbf{x}\|_2 = 1$ is a unit quaternion (i.e. $\mathbf{x} \in S^3$), $\kappa_1, \kappa_2, \kappa_3$ are concentration parameters and $V = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ are orthogonal 4D basis vectors. $F(\kappa_1, \kappa_2, \kappa_3)$ is a normalization term, ensuring that the distribution integrates to 1 on the surface of the hypersphere S^3.

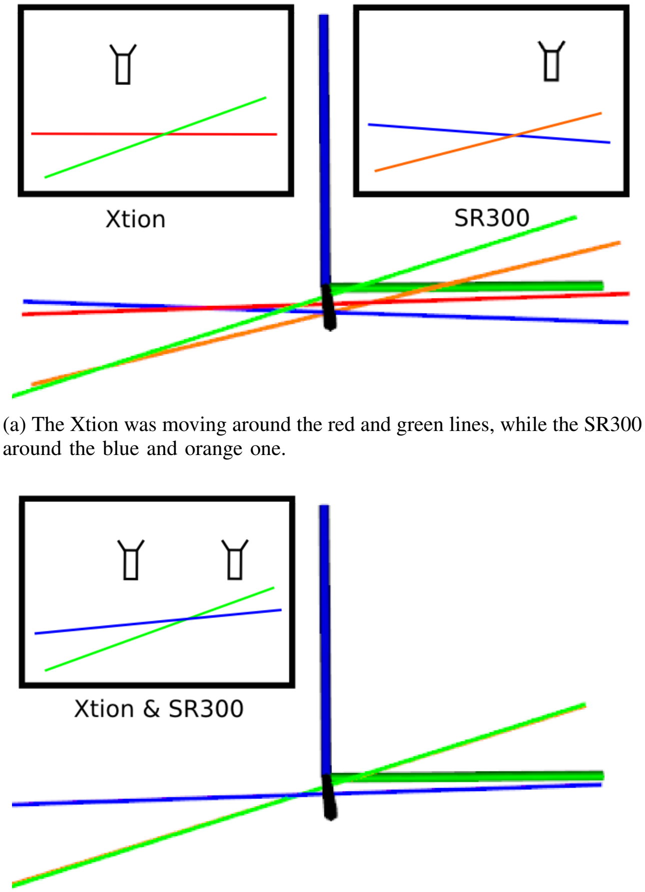
B. Robot-to-Camera Calibration

A rotation axis can be uniquely defined by an axis vector (i.e a direction in 3D) and a support point for this axis. Intuitively one can imagine a marker to be rotated lying on a 3D circle. The perpendicular to this circle's plane is the rotation's axis vector, whereas its center is the support point. As evaluated in [11], to accurately find the direction vector, it is more robust to consider the relative rotations in consecutive frames of all AprilTags jointly instead (as they all undergo the same rotation wrt the camera), and fit a Bingham to them by maximizing the observation likelihood

$$\arg\max_{K,V} p(\mathbf{x} \mid K, V) = \arg\max_{K,V} \prod_{i=1}^{N} p(\mathbf{x}_i \mid K, V), \quad (2)$$

where the likelihood $p(\mathbf{x}_i \mid K, V)$ of a single relative orientation $\mathbf{x}_i$ is given by (1). For more details on how to find the parameters K and V, we refer to [19]. Then, the rotation axis can be computed from the mode of the distribution, and by finding a support point for it. To locate one, and concurrently refine this direction, a non-linear optimization is performed to fine a line crossing the centers of all the 3D circles formed by the tags' detections [11] (see Fig. 2).

Then, the camera is tilted differently, and performs a rotation of the same joint. For each tilt the rotation axis is estimated like described above. The intersection of these bundle of rotation axes is then corresponds to a fixed point of the joint around which all the rotations occur (see Figure 3a).



(a) The Xtion was moving around the red and green lines, while the SR300 around the blue and orange one.

(b) Corresponding rotation axes are overlaid, resulting in a camera-to-camera calibration without assuming overlapping fields of view.

Fig. 3. Estimated rotation axes and intersections (the large central coordinate axes denote the local sensors frames of the two cameras). See Fig. 4 and Fig. 5 for details on the cameras' mounting and joint movements.

C. Camera-to-Camera Calibration

When two or more cameras are attached to the robot, all cameras can be calibrated to a common frame of reference of the robot as described above, thus the transformation from one camera to another can be easily obtained. This allows a camera to camera calibration even in cases with no overlap in the field of view of the cameras, a case where traditional checkerboard based camera-camera calibration fails.

Fig. 3 shows the principle used: in Fig. 3a the resulting rotation axes from two cameras can be seen, whereas the red and green line depict the rotation axes from two robot movements of camera 1 (Asus Xtion), and the blue and orange one show the axes of camera 2 (Intel SR300). By estimating the transformation of the intersection of one line bundle to the other, one can obtain the camera-camera calibration. This can be seen in Fig. 3b, where corresponding rotation axes are overlaid, showing only a minimal mismatch.

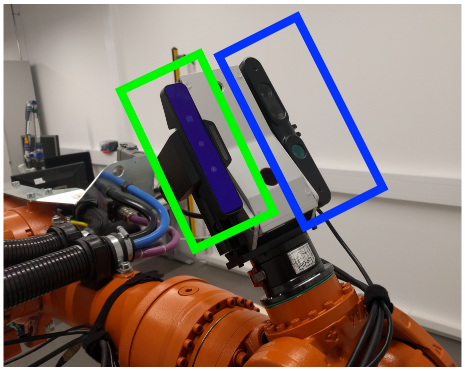


Fig. 4. The two sensors used, an Asus Xtion (blue) and an Intel SR300 (green) RGBD camera, attached to the end effector of a KR16. Note that we only use the RGB image of the sensor, not the depth image.

IV. Results

For performing experiments with ground truth estimates, we used an industrial 6 axis manipulator (KUKA KR16). We attached two RGB-D sensors at the end effector of the robot (Fig. 4). Please note that for our method we only require monocular images and therefore don't use the depth information of the sensors. In the surroundings of the robot mutiple AprilTags are placed. We then performed two different motions with the robot, in which we rotate the axis 5 of the robot for different tilts of axis 6. During those motions we track the surrounding AprilTags and obtain 6D poses describing the rotations and translation of each marker relative to the camera frame. See also Fig. 5, where (a) and (b) show two time frames from the first movement, and (c) from the second movement, i.e. for another tilt of axis 6. Please refer to the accompanying video[2] for details.

A. Rotation Modeling and Marker Selection

In a first pre-experiment a linear motion of the robot is performed, during which AprilTags are tracked. This allows a collection of rotational and translational error measures because we can assume that during this linear movement no rotational changes occur. Fig. 6 shows the distribution of translational and angular errors for each marker.

This can be used for ranking the detected markers according to their errors. Fig. 7 shows the effect of removing certain markers on the error of the estimated rotation axis during a rotational move. It shows two cases, a consecutive reduction of the markers with the highest errors (solid blue line), leading to reduction of the estimation error until only one marker is left, which is not enough for a precise estimation of the rotation axis. On the other side, when "good" markers are removed, i.e. with a small rotational and translational error in the linear motion, the rotation axis error increases continuously (dashed green line).

Please note that this ranking and pruning step is not required, but shows the robustness of the method to markers of different quality. If there are at least two visible markers available, a precise estimation of the rotation axis is possible. On the other hand, if a robot is available that allows for precise linear movements, the overall quality of the method can be improved by ranking the markers based on their quality. An example result can be seen in Fig. 2.

TABLE I

Translational and rotational errors of the estimated robot-camera calibrations compared to the ground truth obtained by classical hand-eye calibration [20]

	angular error (deg)	trans. error (mm)
A5 to Xtion	0.422435	39.5144
A5 to SR300	0.220287	40.4163

TABLE II

Errors of estimated camera-camera calibration compared to the ground-truth stereo calibration [20] for: classical hand-eye calibration and kinematics, AprilSLAM, our results on the markers used by the SLAM approach and on all markers.

	angular error (deg)	trans. error (mm)
classical (kin)	0.30390	0.3813
SLAM	0.69997	29.8605
our on SLAM	1.01798	15.6916
our on all	0.25334	3.9196

B. Evaluation of Robot-to-Camera Calibration

Table I shows the result of the estimated hand-eye calibration, compared to the classical hand-eye calibration based on the inverse kinematics of the robot, represented as the angular and translational error from axis 5 (A5) to the sensors.

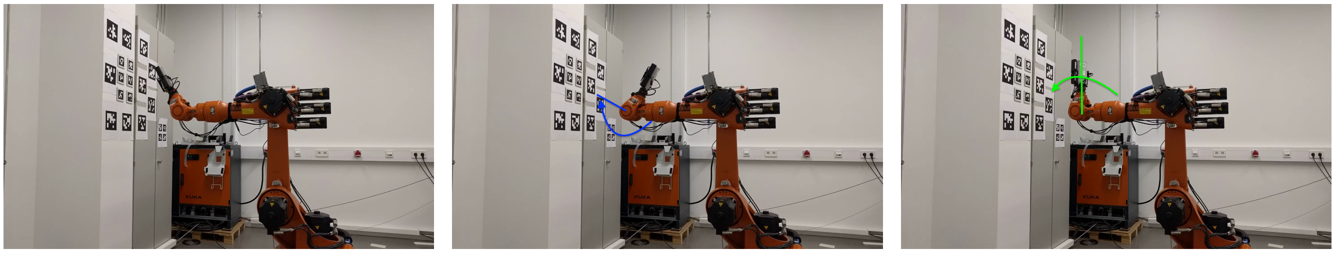
Note that a classical hand-eye calibration is not possible for every robot, for example if no inverse kinematics are available, whereas our method works for any manipulator which can be rotated around two non-parallel axes. It can be also seen that the Xtion produces slightly higher errors, which can be explained by its lower resolution.

C. Evaluation of Camera-to-Camera Calibration

Table II shows the error of a camera to camera calibration based on our method compared to a classical (stereo) camera-camera calibration [20]. Note that a classical stereo calibration is only possible with overlapping cameras, or more cumbersome solutions, e.g., using custom built rigid mounts with checkerboard patterns to be visible from the cameras. Our method doesn't need this, because it obtains a hand-eye calibration for both non overlapping cameras and obtain the calibration between the cameras through this common robot frame. While this is possible with two traditional hand-eye calibrations through the kinematic chain as well (see "kin" results in the table), specific motions need to be executable by the robot, while the requirement for rotating a joint (interleaved with a small movement of another one) is arguably easier to fulfill.

It can be seen, that our method can estimate the rotation between the cameras very well (and even better than using the individual hand-eye calibrations), but has a small shift in the translational estimation. However, our method is the only

[2] https://youtu.be/75zRNwA95xQ

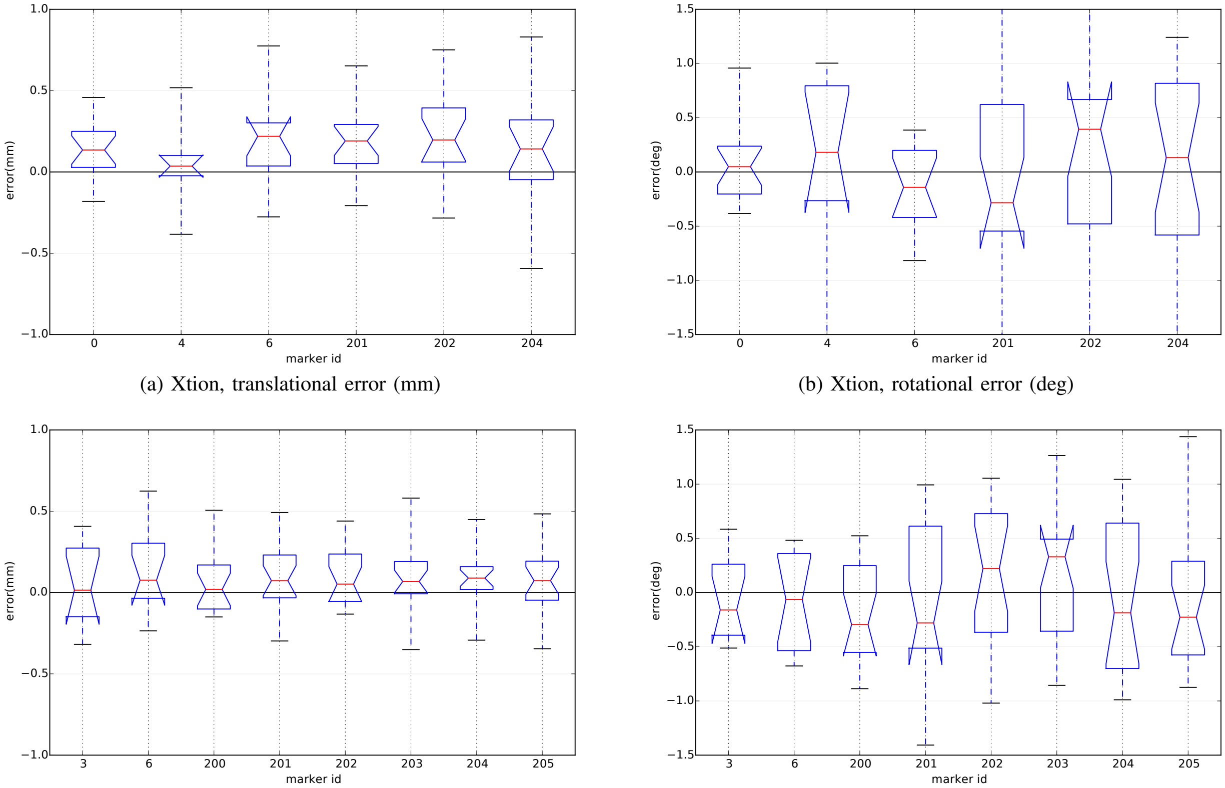


(a)

(b)

(c)

Fig. 5. The experiment setup, consisting of a KUKA KR16 industrial manipulator, with two cameras attached to the end effector (see Fig. 4 for a closeup) and AprilTags placed in the surroundings of the robot. The robot is shown at three stages of the experiment, whereas (a) and (b) show the start and end of the first movement, which is a rotation around axis 5 (around the blue rotation axis), and (c) one frame of the second movement, in which the last axis of the robot is tilted differently, shown as a rotation around the green rotation axis.



(a) Xtion, translational error (mm)

(b) Xtion, rotational error (deg)

(c) SR300, translational error (mm)

(d) SR300, rotational error (deg)

Fig. 6. Box plot of translational errors (left column) and rotational errors (right column) for both used cameras (upper: Xtion, lower: SR300)

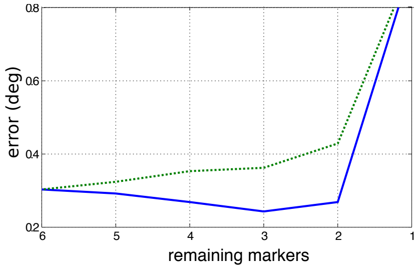


Fig. 7. Error when continuously removing “bad” markers , i.e., ordered by increasing quality (solid blue line) and when continuously removing “good” markers, i.e., ordered by decreasing quality (dashed green line).

method of the presented (and we know of), that is capable of producing a rather precise calibration even when no inverse kinematics are available and cameras do not overlap.

The AprilSLAM approach, at least in this setting, did not improve the results, as seen in Table II. Compared to using only the markers that were usable by AprilSLAM as input to our method, the rotational error slightly decreases through SLAM, whereas the translational error increases.

V. Discussion and Limitations

This method is unsupervised and can run automatically without the need to mount or move a checkerboard calibration pattern. Multiple AprilTag markers can be placed in the surroundings of the robot; or, if the camera is fixed, on the robot arm, as in the case of the LRU (see Fig. 1a). In the latter case typically multiple smaller tags are used, that can stay on the arm for future verification and re-calibration. Another benefit of our method is that we do not need any prior knowledge about the robot or its surroundings (except that all seen AprilTags have a known size). The expected camera intrinsics are in most cases easy to obtain.

While we investigated the effect of the number and quality of the observed markers, an important remaining open question is the amount of rotation s.t. we obtain high-quality results. As a marker can be observed the longest when it passes diagonally the complete camera image, that is an upper limit. Since we are estimating the axis only from consecutive detections, i.e., small relative angles, the major limitation here is the accurate estimation of the center of rotation. Finding the centers of the detected circles, which lie on the rotation axis, is more accurate if large sections of the circles are sampled. With a typical opening angle of around 60°, we ran into the problem of low quality detections at the edges introducing a bias towards larger circles, thus displaced centers towards the back of the camera. In our initial investigation this caused the relatively large translational errors that we observe. We will address this issue via a more accurate optimization of the center point of the rotations, by considering the orientations of the detections in this step as well and their estimated qualities.

VI. Conclusions

We presented a simple-to-use and novel method to obtain the rigid body transformation between a robot frame of reference and a camera frame of reference. This is classically called hand-eye calibration, but we are performing camera to camera calibration this way as well, even if there is no overlap between the field of view of different cameras.

This is achieved by rotating the last axis of the robot and sampling a high number of visual features, in our case AprilTag markers. By fitting a Bingham distribution to the markers' relative rotations around several configurations of the arm, a mathematical model connecting the kinematics of the robot to the camera frame of reference can be found. We have shown in [11] that this method can be applied where a classical hand-eye calibration is problematic, e.g. a humanoid robot with limited movability.

Additionally, the kinematic chain does not necessarily have to be an arm, it can be for example also just a mobile base. In our case the robot was able to rotate around multiple axes, which is not usually the case. Therefore, unless the camera is mounted on a movable part (typically sensors are placed on a pan-tilt unit), there is only one possible axis of rotation. In such cases, determining the ground plane and obtaining the height of the sensor above it (either by placing specific markers on the ground, or using depth information), and performing a linear motion in a robot-specific direction will create a plane and a direction that can be intersected with the axis of rotation to find the robot-camera calibration.

We plan to investigate such challenging situations by applying our method to different real-word robotic problems and evaluating it on more robot platforms.

Acknowledgment

We would like to thank Simon Kriegel, Michal Smisek and Klaus Strobl for their help with the experiments. This work has partly been supported by the European Commission under contract number H2020-ICT-730014-INFUSE.

References

[1] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *Intern. Conf. on Intell. Robots and Systems (IROS)*. IEEE, 2006.

[2] ——, "More accurate camera and hand-eye calibrations with unknown grid pattern dimensions," in *Intern. Conf. on Robotics and Automation (ICRA)*. IEEE, 2008.

[3] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, Jun 1989.

[4] M. J. Schuster, C. Brand, S. G. Brunner, P. Lehner, J. Reill, S. Riedel, T. Bodenmüller, K. Bussmann, S. Büttner, A. Dömel, *et al.*, "The LRU rover for autonomous planetary exploration and its success in the spacebotcamp challenge," in *Intern. Conf. on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2016, pp. 7–14.

[5] B. Bäuml, F. Schmidt, T. Wimböck, O. Birbach, A. Dietrich, M. Fuchs, W. Friedl, U. Frese, C. Borst, M. Grebenstein, O. Eiberger, and G. Hirzinger, "Catching flying balls and preparing coffee: Humanoid rollin' justin performs dynamic and sensitive tasks," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2011.

[6] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 313–326, 1992.

[7] P. I. Corke, "Visual control of robot manipulators-a review," *Visual servoing*, vol. 7, pp. 1–31, 1993.

[8] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 1998, pp. 66–78.

[9] T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox, "Depth-based tracking with physical constraints for robot manipulation," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 119–126, Best Robotic Vision Paper Award Finalist.

[10] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3400–3407.

[11] C. Nissler and Z.-C. Marton, "Robot-to-Camera Calibration: A Generic Approach Using 6D Detections," in *1st Intern. Conf. on Robot Computing (IRC)*. IEEE, 2017, short paper, presented at a workshop.

[12] A. Teichman, S. Miller, and S. Thrun, "Unsupervised intrinsic calibration of depth sensors via slam." in *Robotics: Science and Systems*, vol. 248, 2013.

[13] R. Kümmerle, G. Grisetti, and W. Burgard, "Simultaneous calibration, localization, and mapping," in *Intern. Conf. on Intell. Robots and Systems (IROS)*. IEEE, 2011, pp. 3716–3721.

[14] J. Leitner, S. Harding, M. Frank, A. Forster, and J. Schmidhuber, "Transferring spatial perception between robots operating in a shared workspace," in *Intern. Conf. on Intell. Robots and Systems (IROS)*, 2012, pp. 1507 –1512.

[15] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *arXiv preprint arXiv:1603.02199*, 2016.

[16] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[17] C. Bingham, "An antipodally symmetric distribution on the sphere," *The Annals of Statistics*, pp. 1201–1225, 1974.

[18] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[19] J. Glover, R. Rusu, and G. Bradski, "Monte Carlo pose estimation with quaternion kernels and the Bingham distribution," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[20] K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. (2010, July) DLR CalDe and DLR CalLab. Institute of Robotics and Mechatronics, German Aerospace Center (DLR). Oberpfaffenhofen, Germany. [Online]. Available: http://www.robotic.dlr.de/callab/