A Graph-based Optimization Framework for Hand-Eye Calibration for Multi-Camera Setups

Daniele Evangelista[*], Emilio Olivastri[*], Davide Allegro, Emanuele Menegatti and Alberto Pretto

Abstract— **Hand-eye calibration is the problem of estimating the spatial transformation between a reference frame, usually the base of a robot arm or its gripper, and the reference frame of one or multiple cameras. Generally, this calibration is solved as a non-linear optimization problem, what instead is rarely done is to exploit the underlying graph structure of the problem itself. Actually, the problem of hand-eye calibration can be seen as an instance of the Simultaneous Localization and Mapping (SLAM) problem. Inspired by this fact, in this work we present a pose-graph approach to the hand-eye calibration problem that extends a recent state-of-the-art solution in two different ways: i) by formulating the solution to *eye-on-base* setups with one camera; ii) by covering multi-camera robotic setups. The proposed approach has been validated in simulation against standard hand-eye calibration methods. Moreover, a real application is shown. In both scenarios, the proposed approach overcomes all alternative methods. We release with this paper an open-source implementation of our graph-based optimization framework for multi-camera setups.**

I. Introduction

The ability to robustly perceive the surrounding environment is a key factor for the development and deployment of intelligent robotic work cells in industry. Perception devices, for example cameras, are widely used in industrial robotics applications to perform complex tasks that involves robot arms such as visual inspections [1], [2] and bin-picking [3], [4]. They are also involved as assistive devices that continuously monitor the position of human workers in collaborative tasks with robots [5], [6]. However, the information provided by the cameras is generally expressed with respect to their own coordinate systems. To make it possible to control a robot arm based on such visual information, it is first necessary to accurately calculate the position of the camera relative to the robot gripper or relative to the robot base or work cell. This process is called *hand-eye calibration*, and this is defined as the procedure to determine the spatial transformation between a robot reference frame (e.g., robot base or end-effector frames) and the reference frame of the camera. The hand-eye calibration problem can be classified into two categories [7]:

- **Eye-in-hand**: The camera is rigidly mounted on the robot's wrist;
- **Eye-on-base** (or **Eye-to-hand**): One or more cameras are fixed while the robot arm is moving (e.g., Fig. 1)

There exists a variety of methods that solve this problem, many of them based on a global optimization that minimizes
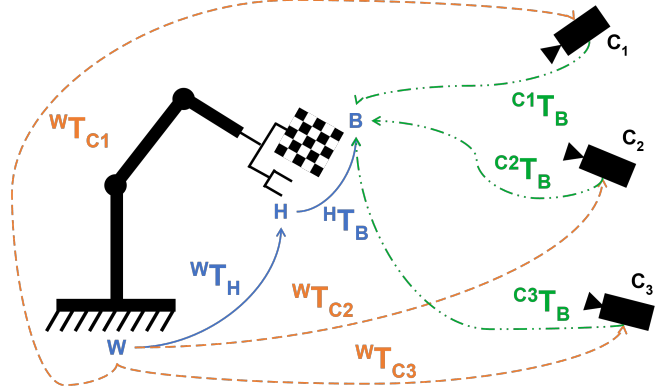

Fig. 1. Multi-camera Eye-on-Base calibration setup. The goal of the proposed hand-eye calibration is to compute the position of each camera in the reference frame W, i.e., the transformations $^{W}T_{C1...C3}$ in the figure.

specific quantities, such as geometric ones like translation and rotation errors or visual ones like reprojection errors. What, instead, is rarely done is to exploit the underlying graph structure of the problem. In fact, as introduced in [8], the hand-eye calibration problem with an eye-in-hand setup can be seen as an instance of the SLAM (Simultaneous Localization And Mapping) problem and can be solved by exploiting standard probabilistic frameworks exploited in SLAM, such as pose graph optimization [9]. In this work, we exploit this intuition by proposing a general graph-based tool to solve the complementary eye-on-base calibration problem. To cast this problem to SLAM, one can consider that the motion is relative: instead of considering the motion of the end-effector in front of the static camera, one can imagine the camera is moving, and the robot's end-effector is static. Thus, taking images (i.e., observations) while the robot moves a checkerboard mounted on the end-effector to different poses (i.e., landmarks) results in observations of landmarks used for the optimization.

Our solution is based on the minimization of reprojection error, which is a well-known technique that has been applied to different types of problems (e.g. camera calibration, bundle adjustment, . . .) but surprisingly rarely in the context of hand-eye calibration problems. This technique also allows us to avoid relying strictly on the PnP (Perspective-n-Point) algorithm, which in our case is simply used to infer an initial guess for the optimization.

The main contributions of this work are the following:

(i) The proposed approach extends a recent state-of-the-art approach [8] by implementing a graph-based opti-

[*]These authors contributed equally. All the authors are with the Department of Information Engineering (DEI) at the University of Padova, via Gradenigo 6/B, 35131 Padova, Italy. Emails:`[evangelista; olivastrie; allegrodav; emg; alberto.pretto]@dei.unipd.it`.

mization solution to *eye-on-base* setups with one camera (Fig. 2), not covered in the basic method;

(ii) Our method also covers multi-camera setups (Fig. 1); thus, a general and unified graph-based optimization framework for hand-eye calibration is achieved;

(iii) An exhaustive evaluation of the proposed method is reported. The evaluation is primarily performed in simulated environments, where the robustness to visual and geometric noise can be tested. The proposed approach outperforms several popular calibration approaches, proving to be robust to high levels of noise in the calibration data. We also tested our method on a real setup, confirming the convincing results obtained in the simulations;

(iv) An open-source implementation of the proposed hand-eye calibration method is made publicly available with this paper at:

`https://bitbucket.org/freelist/gm_handeye`

The remainder of this paper is organized as follows. Section II reviews related work on hand-eye calibration. Section III presents and describes the proposed hand-eye calibration methodology based on the formalization of the process through graph optimization guided by minimization of a customized error function. In Section IV we show a comprehensive evaluation, performed in simulation, and demonstrate the applicability of the proposed method in real hand-eye calibration problems. Finally, in Section V we draw our conclusions.

II. RELATED WORK

In the literature, several methods have been proposed for hand-eye calibration. Shiu and Ahmad [10] initially proposed a method for the estimation of the hand-eye transformation based on the solution of a homogeneous equation. Tsai and Lenz [11] proposed a widely adopted calibration method (e.g., implemented in [12]) that first estimates the translation and then the rotation of a hand-eye transformation. Similarly to Shiu and Tsai, Chou and Kamel [13] solved an estimation problem based on a normalized quaternion representation to transform the kinematic equation into two simple and structured linear systems with rank-deficient coefficient matrices. Daniilidis and Bayro-Corrochano [14] proposed a similar framework that is based on dual quaternion parameterization and singular value decomposition, and Srobl and Hirzinger [15] introduced a novel metric to optimize the estimation problem by appropriately weighting translation and rotation errors in the optimization problem. Park and Martin [16] estimated the hand-eye transformation in the Euclidean group using Lie group and least-square optimization. Gwak *et al.* [17] proposed a cyclic coordinate descent algorithm to optimize objective functions in SE(3) and applied it to a class of robotic problems such as hand-eye calibration. In this direction, Horaud and Dornaika [18] proposed a linear formulation of the same optimization problem. The same method has been further applied to online calibration by Andreff *et al.* [19].

More recently Shah [20] formulated a closed-form solution for the hand-eye problem by using an SVD-based algorithm and the Kronecker product to solve for rotation and translation separately, while LI *et al.* [21] instead used dual quaternions
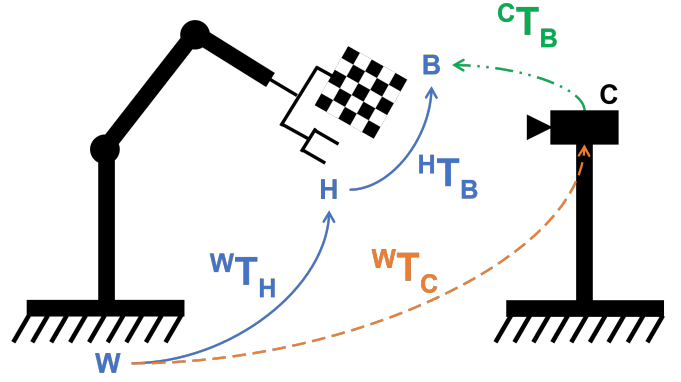

Fig. 2. Single-camera Eye-on-Base calibration setup. The goal of hand-eye calibration here is to compute the position of the camera in the W reference frame, namely the transformation $^{W}T_C$ in the figure.

to solve them simultaneously to overcome the limitations of the Kronecker product.

More general formulations of the hand-eye calibration problem have been done in [22] and [23]. The latter, in particular, mathematically formulates the problem of camera-to-camera calibration using the geometric constraints of hand-eye calibration when a common reference frame (e.g., robot base frame) is taken into the loop. Although this method formally extends the formulation of [11] to multi-camera setups, it requires an external motion capture system to accurately recover the position of cameras during calibration, making the whole approach not easily scalable or applicable to small setups.

In our previous work [8], we proposed a more general approach based on pose graph optimization that achieves high levels of accuracy and introduces a general solution that can be easily applied to different robotic setups and camera projection models (e.g., both standard pinhole camera projection and X-ray source detector models [24]). Although the proposed method is quite general, it does not cover eye-on-base setups in single-camera setups, and it does not handle multi-camera robotic systems. In this work, we overcome this limitation by extending the approach to eye-on-base setups; moreover, we give a general formulation applicable to multi-camera setups where the position of multiple cameras has to be computed w.r.t. a robot base frame.

III. METHODOLOGY

This section presents a detailed overview of the proposed approach. As already mentioned, this work extends [8] in two main directions, explained hereafter: eye-on-base for single- and multi-camera setups. In this way, we are able to provide a complete framework that holds the necessary instruments to fulfil all the calibration needs that arise in robotic industrial applications.

A. Eye-on-Base Calibration for Single Camera

Here, we report the necessary notation for eye-on-base single camera setups.
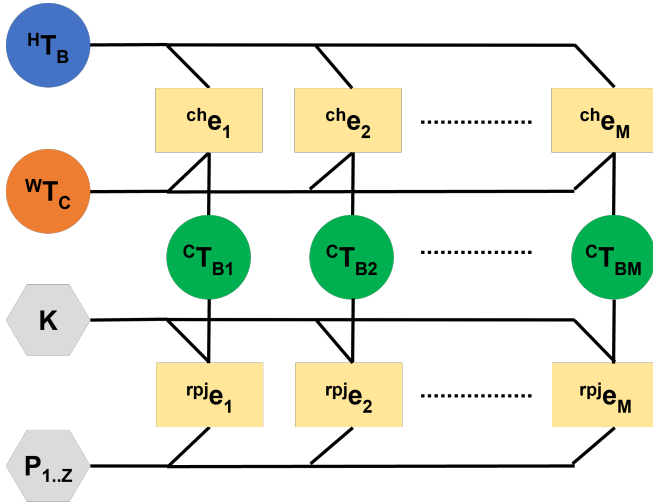
Fig. 3. Graphical representation of the optimization problem for eye-on-base single camera setups. The representation contains circles, rectangles and hexagons that respectively represent the estimated variables, error functions, and fixed parameters. The error functions are defined following the equations 3 and 4.

- **W** : world reference frame, usually placed in the robot's base;
- **H** : hand reference frame, placed in the robot's end-effector;
- **B** : checkerboard reference frame;
- **C** : camera reference frame;

While the transformations are :

- $^{W}\mathbf{T_H}$: Isometry representing the pose of the end-effector in the world reference frame;
- $^{H}\mathbf{T_B}$: Isometry representing the pose of the checkerboard in the end-effector reference frame;
- $^{C}\mathbf{T_B}$: Isometry representing the pose of the checkerboard in the camera reference frame;
- $^{W}\mathbf{T_C}$: Isometry representing the pose of the camera in the world reference frame.

Fig. 3 shows the graph structure for the eye-on-base calibration. Using the notation from [8]: circles represent the state variables that need to be estimated, while hexagons are fixed parameters such as K and $P_{1..Z}$, that represent the camera matrix and the set of corners points of the checkerboard in their reference system, respectively. M represents the number of observations, i.e., images and robot poses, used for the calibration. The rectangles represent the different error terms. The error is generally defined as

$$e_i(x) = h_i(x) \boxminus z_i, \qquad (1)$$

where $h_i(x)$ is the measurement function and z_i is the i-th observation and $\boxminus$ is the difference operator for the manifold domain.

The first type of edge that is going to be presented is the one that enforces the equality :

$$^{W}\mathbf{T_H} \cdot {}^{H}\mathbf{T_B} = {}^{W}\mathbf{T_C} \cdot {}^{C}\mathbf{T_B}, \qquad (2)$$

that is one of the classic constraints for eye-on-base calibration [25]. And its corresponding error representation:

$$^{ch}\mathbf{e}_i = t2v([{}^{H}\mathbf{T_W}]_i \cdot {}^{W}\mathbf{T_C} \cdot [{}^{C}\mathbf{T_B}]_i \cdot {}^{B}\mathbf{T_H}), \qquad (3)$$

where the observation $\mathbf{z}_i = [{}^{W}\mathbf{T_H}]_i^{-1}$ is the i-th pose of the end-effector in the world. The remaining transformations are the ones that need to be estimated, while the $t2v(\cdot)$ is a function that converts the transformation to the corresponding minimal manifold representation in order to prevent optimization degradation due to over parametrization. The second type of edge instead enforces the reprojection error constraint:

$$^{rpj}\mathbf{e}_i = \begin{bmatrix} \pi([{}^{C}\mathbf{T_B}]_i \cdot {}^{B}\mathbf{P_1}) - [\mathbf{u_1}]_i \\ \vdots \\ \pi([{}^{C}\mathbf{T_B}]_i \cdot {}^{B}\mathbf{P_j}) - [\mathbf{u_j}]_i \\ \vdots \\ \pi([{}^{C}\mathbf{T_B}]_i \cdot {}^{B}\mathbf{P_Z}) - [\mathbf{u_Z}]_i \end{bmatrix}, \qquad (4)$$

where the observation $\mathbf{z}_i = [\mathbf{u_1} \dots \mathbf{u_j} \dots \mathbf{u_Z}]_i$ is a vector of tuples $\mathbf{u_j} = (u_x, u_y)_j$ that are the pixel coordinates of the j-th detected corner of the checkerboard at the i-th observation and π is the projection function of the camera.

The optimization aims to minimize the following function:

$$\min \sum_{i=1}^{M} \left(\lambda \|{}^{rpj}\mathbf{e}_i\|^2 + \|{}^{ch}\mathbf{e}_i\|^2 \right), \qquad (5)$$

where λ is a weighting factor that accounts for the non homogeneity of the two error terms. The minimization is carried out by using the Levenberg-Marquardt [26] algorithm. Our method proved to be robust also if provided with an identity transformation as initial guess, showing a rather large basin of convergence. This means that the system can be correctly calibrated even if we do not provide a good initial guess, which could be difficult to compute.

B. Eye-on-Base Calibration for Multi-Camera

If the robot is observed by multiple static cameras, one could repeat the above algorithm several times. On the other hand, running an algorithm that performs single-camera calibration N times, with N the number of cameras that are present in the workspace, is a sub-optimal solution because, in the optimization, the cross-observations between the different cameras are not factorized.

Given N cameras in the workspace, as shown in Fig. 1, there is the need to introduce a little more notation:

- $^{C_d}\mathbf{T_{C_a}}$: Isometry representing the pose of the a-th camera in the d-th camera reference frame;
- $^{C_a}\mathbf{T_B}$: Isometry representing the pose of the checkerboard in the a-th camera reference frame;
- $^{W}\mathbf{T_{C_a}}$: Isometry representing the pose of the a-th camera in the world reference frame.

The starting point for the multi-camera model is simply stacking the single-camera model N times as the number of the cameras. In Fig. 4 some of the vertices and edges have been grouped together in order to facilitate the graphical representation.
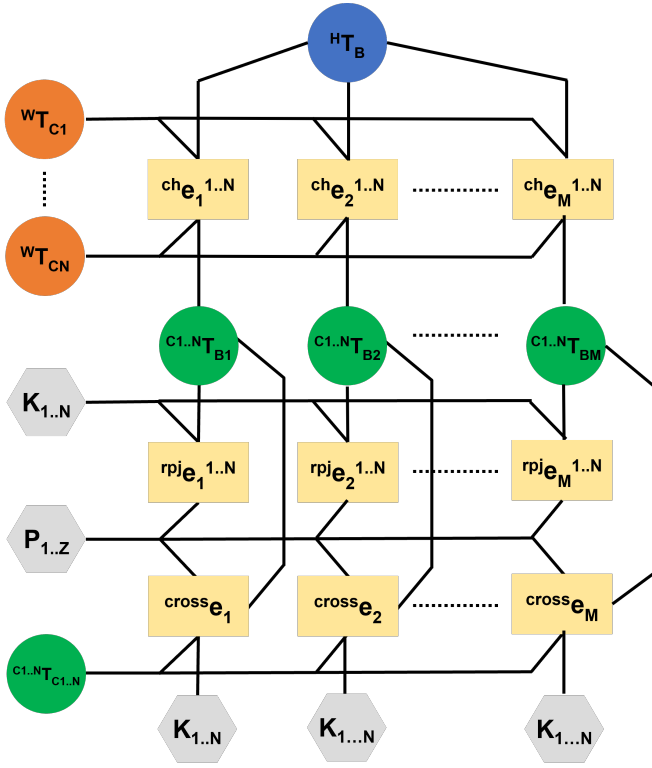
Fig. 4. Evolution of the graph representation in multi cameras setups. The figure reports the error terms of Eq. (6-7) that are the stack of the errors in Eq. (3-4), respectively. Moreover, the graph includes an additional error defined in Eq. 10 which enforces constraints between the cameras with overlapping field of view.

The error term $^{ch}\mathbf{e}_i^{1..N}$ enforces all the chained equalities of the system :

$$^{ch}\mathbf{e}_i^{1..N} = \left[^{ch}\mathbf{e}_i^1 \ldots ^{ch}\mathbf{e}_i^a \ldots ^{ch}\mathbf{e}_i^N \right]^T, \qquad (6)$$

where $^{ch}\mathbf{e}_i^a$ represents the chained equality for the a-th camera at the i-th time step.

The error term relative to the reprojection error evolves into:

$$^{rpj}\mathbf{e}_i^{1..N} = \left[^{rpj}\mathbf{e}_i^1 \ldots ^{rpj}\mathbf{e}_i^a \ldots ^{rpj}\mathbf{e}_i^N \right]^T, \qquad (7)$$

where $^{rpj}\mathbf{e}_i^a$ is the reprojection error term relative to the camera a-th at the i-th time step. Up to this point, we have simply stacked the single-camera graph N times one on top of the other, in fact the derivation of the new error terms was straightforward. The additional error term that is added to the graph model is the one that represents the cross-observation between cameras, from which information about their relative spatial relation can be extracted. Cross-observations occur when more cameras are able to see the checkerboard at the same time step i. In order to add it in an automatic manner to the optimization, it is needed to build a co-visibility matrix $\mathbf{X}_i$. At each time step i, each cell of this matrix is represented by $\mathbf{X}_i(a, d)$, where a and d are two different cameras (i.e., $a \neq d$). The cell contains a binary value that encodes whether or not the checkerboard is in the field of view of both cameras

a and d at the i-th time step:

$$\mathbf{X}_i(a, d) = \begin{cases} 0, & \text{if } a = d \text{ or no cross-observation} \\ 1, & \text{otherwise} \end{cases} \qquad (8)$$

Thus, using the co-visibility matrix we are able to enforce the cross-observation constraints only where they are actually aiding the optimization.

To capture the spatial relation between cameras the transformation $^{C_d}\mathbf{T}_{C_a}$ is added as a state variable that needs to be estimated.

The new error term that models the cross-observation is defined as follows:

$$^{cross}\mathbf{e}_{iad} = \begin{bmatrix} \pi_a(^{C_a}\mathbf{T}_{C_d} \cdot [^{C_d}\mathbf{T}_B]_i \cdot \mathbf{P}_1^B) - [\mathbf{u}_1^a]_i \\ \vdots \\ \pi_a(^{C_a}\mathbf{T}_{C_d} \cdot [^{C_d}\mathbf{T}_B]_i \cdot \mathbf{P}_j^B) - [\mathbf{u}_j^a]_i \\ \vdots \\ \pi_a(^{C_a}\mathbf{T}_{C_d} \cdot [^{C_d}\mathbf{T}_B]_i \cdot \mathbf{P}_Z^B) - [\mathbf{u}_Z^a]_i \end{bmatrix}, \qquad (9)$$

where the indices a and d refer to the cameras and $[\mathbf{u}_j^a]_i$ represents the j-th corner of the checkerboard detected by the camera a at the i-th time step. This error accounts for the current estimate of the relative transformation between cameras. Checkerboard points are projected into the camera c_a using the camera c_d as starting point.

To simplify the notation all the error terms at time step i will be marginalized:

$$^{cross}\mathbf{e}_i = \sum_{a=1}^{N} \sum_{d=1}^{N} \|^{cross}\mathbf{e}_{iad}\|^2 \cdot \mathbf{X}_i(a, d), \text{ with } a \neq d. \quad (10)$$

Fig. 4 shows the complete graphical model that includes these final constraints. The resulting function that needs to be minimized is the following:

$$\min \sum_{i=1}^{M} \left(\lambda_1 \|^{rpj}\mathbf{e}_i\|^2 + \|^{ch}\mathbf{e}_i\|^2 + \lambda_2 {}^{cross}\mathbf{e}_i \right) \qquad (11)$$

The effect that the cross-observations have on the optimization is to make it more robust against noise, and it will be effectively shown in the following section.

The proposed algorithm has been implemented using the g2o [27] framework. The weighting factors λ_1 and λ_2 have been experimentally tuned and set to the values $[10^{-6}, 10^{-3}]$, respectively, and kept fixed during all experiments.

IV. EXPERIMENTS

In Section IV-A we show the robustness and estimation accuracy of the proposed method by testing it under visual and geometric perturbations. In particular, we both added noise to the detection of the checkerboard corners, thus simulating wrong detections of the calibration pattern (e.g., when using low-resolution cameras), and we also added noise to the robot poses $[^H\mathbf{T}_W]_i$ used for calibrating.

The proposed approach has been tested against the following methods (i.e., implementation from OpenCV[1]): [14], [16],

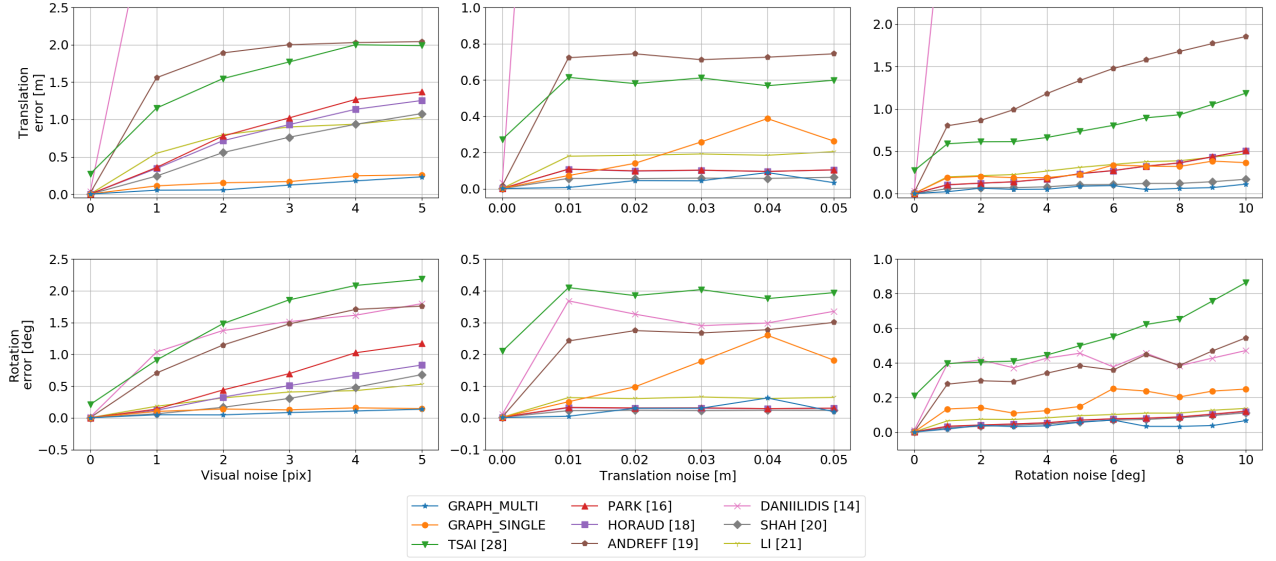[1]http://docs.opencv.org/4.5.4/d9/d0c/group__calib3d.html

Fig. 5. Result of the simulated evaluation. The proposed approach, both in *single-* and *multi-camera* settings has been tested against Tsai [28], Park [16], Horaud [18], Andreff [19], Daniilidis [14], Shah [20] and Li [21].
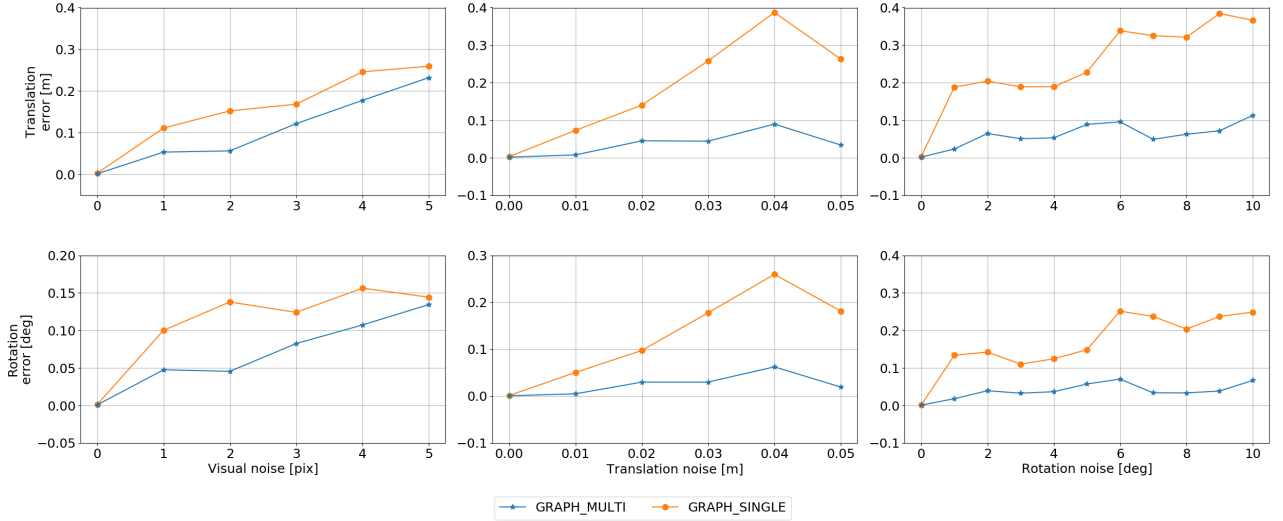

Fig. 6. More in depth focus of the simulated evaluation regarding the proposed method only in *single-* and *multi-camera* settings.

Error [pix] Camera	TSAI [28]	PARK [16]	HORAUD [18]	ANDREFF [19]	DANIILIDIS [14]	SHAH [20]	LI [21]	GRAPH_MULTI
Camera 1	443.763	0.520446	0.432767	1.40595	196253	0.503051	0.603978	**0.337803**
Camera 2	0.416451	0.346938	0.35017	1.98045	0.538384	0.439182	0.717876	**0.284408**
Camera 3	148.89	26.2883	25.0635	165429	38.9004	97.8753	417.908	**4.4232**
Average	197.69	9.0519	8.61548	55144.2	65430.8	32.9392	139.743	**1.6818**

TABLE I

EVALUATION OF THE REPROJECTION ERRORS USING THE REAL SETUP DATA. ALL THE VALUES IN THE TABLE ARE IN PIXEL UNIT.

[18]–[20], [28] and [21]. In the graphs and in the table we name them by reporting the first author of each one. The alternative methods have been tested by calibrating every single camera independently and then averaging the errors. This experimental methodology has been adopted because the alternative approaches are not directly usable for multi-camera

calibration. The evaluation also reports an in-depth focus on the comparison of our multi-camera calibration method against the repetition of the proposed single-camera calibration applied to the individual cameras, this is done to demonstrate the robustness and improved accuracy of the multi-camera calibration framework.
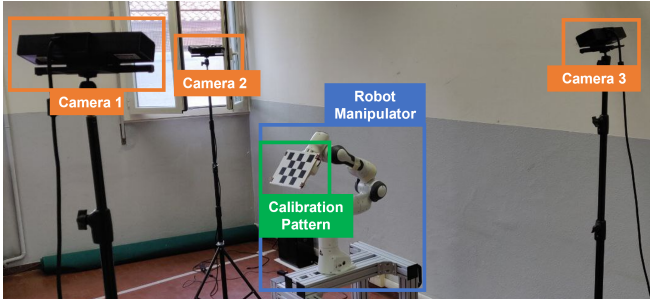
Fig. 7. Front view of the real setup where the proposed calibration method has been tested.

Fig. 8. Rear view of the real setup. Main reference frames are depicted.

A. Experiments with Synthetic Data

Using the same evaluation protocol proposed in [8], [29], our method has been tested, under simulated conditions, using three different sources of noise: visual, translation, and rotation. This evaluation protocol aims to test the robustness of the calibration under perturbed visual conditions (i.e., introducing noise in the detection of the corners of the calibration pattern), and it is also used to test the optimization accuracy and convergence capabilities when additional geometric noise is added to the robot poses.

We collected simulated data using the *Gazebo*[2] simulator, in which a simulated work cell has been implemented. The proposed work cell is composed of an industrial manipulator, mounting the calibration pattern, surrounded by five external cameras as in typical multi-camera eye-on-base setups. A set of 150 robot poses has been selected by randomly sampling the robot workspace to uniformly cover each camera field of view.

The results of this evaluation are reported in Fig. 5 and Fig. 6. In particular, in Fig. 5 the plots show the performance of all the calibration methods, while in Fig. 6 only the results of our single- and multi-camera graph-based calibration approaches are reported. The results show how our proposed method achieves good results in terms of translation and rotational errors even in strong perturbance conditions, demonstrating that it is more robust compared to other methods. The proposed eye-on-base single-camera calibration (i.e., GRAPH_SINGLE in Fig. 5 and Fig. 6) generally outperforms all the other tested approaches. Moreover, our multi-camera calibration hand-eye algorithm (i.e. GRAPH_MULTI in Fig. 5 and Fig. 6) is the best performing approach, in particular when high levels of noise are present in the calibration data.

B. Experiments on a Real Setup

The proposed algorithm has also been validated in a real robotic environment. The setup consisted of a Franka Emika Panda[3] robot manipulator and three Kinect2 cameras (we use only the RGB images) placed around the robot, as shown in Fig. 7 and Fig. 8. Unlike the simulated evaluation, where ground truth is available, in the real environment the absolute pose of each camera is not known a priori. Thus, to perform the
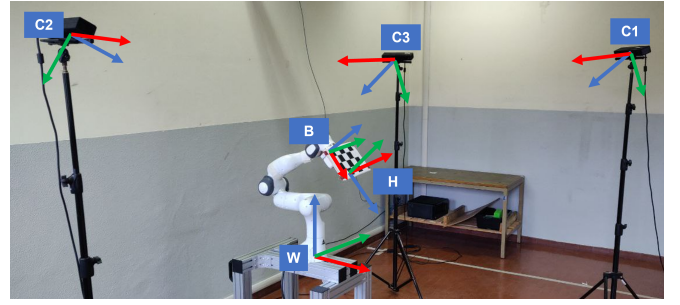
[2]https://gazebosim.org/home
[3]https://www.franka.de/

test in the real setup, the reprojection error has been considered as a metric. In Tab. I the results of all approaches are reported.

Our approach is capable of correctly calibrating all cameras in the setup, also in challenging lighting condition, e.g., for cameras #1 and #3 (see Fig. 7). Most of the tested methods were unable to converge to a low reprojection error solution, while our method provided the lowest error by a large margin.

V. Conclusions

This work presents a pose-graph solution to the hand-eye calibration problem that extends previous state-of-the-art work [8] in two different ways: i) by formulating the solution to *eye-on-base* setups with one camera; ii) by extending the *eye-on-base* calibration to multi-camera setups. To validate the proposed system, we used the same experimental protocol proposed in [8] and [29].

The proposed approach has shown to be robust when the calibration data are affected by visual and geometric errors. Compared to traditional solutions, the implemented pose-graph optimization achieves better accuracy in terms of geometric errors (i.e., translation and rotation).

The proposed approach also proves to be robust and accurate when tested in real hand-eye calibration benchmarks, where absolute ground truth is not available and accuracy is measured in terms of reprojection error.

It should be noticed that the proposed approach, as in the other hand-eye calibration methods used for the experiments, does not take into account the influence of dynamic effects given by gravity or joint and link flexibility. This may affect the accuracy and precision of robot manipulators and must be taken into account with proper dynamic calibration techniques.

The authors believe that this kind of calibration can be formulated with the same graph-based structure as in the proposed hand-eye calibration approach, and it may be considered as future work in this field to reach a more general calibration procedure. Moreover, the proposed graph-based formulation can be further extended in future research to address more complex problems such as calibration of *multi-robot-multi-camera* setups.

REFERENCES

[1] S. Zambal, W. Palfinger, and C. Eitzinger, "Robotic inspection of 3d cfrp surfaces," *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*, pp. 197–202, 2016.

[2] D. Evangelista, M. Terreran, A. Pretto, M. Moro, C. Ferrari, and E. Menegatti, "3d mapping of x-ray images in inspections of aerospace parts," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1223–1226.

[3] A. Pretto, S. Tonello, and E. Menegatti, "Flexible 3d localization of planar objects for industrial bin-picking with monocamera vision system," in *Proc. of: IEEE International Conference on Automation Science and Engineering (CASE)*, 2013, pp. 168 – 175.

[4] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 437–451, 2018.

[5] R.-J. Halme, M. Lanz, J. Kämäräinen, R. Pieters, J. Latokartano, and A. Hietanen, "Review of vision-based safety systems for human-robot collaboration," *Procedia CIRP*, pp. 111–116, 2018.

[6] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco, "Prediction of human activity patterns for human–robot collaborative assembly tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3934–3942, 2019.

[7] J. Jiang, X. Luo, Q. Luo, L. Qiao, and M. Li, "An overview of hand-eye calibration," *The International Journal of Advanced Manufacturing Technology*, vol. 119, 03 2022.

[8] K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1021–1028, 2019.

[9] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[10] Y. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[11] R. Tsai and R. Lenz, "Real time versatile robotics hand/eye calibration using 3d machine vision," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1988, pp. 554–561 vol.1.

[12] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

[13] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *The International Journal of Robotics Research*, vol. 10, no. 3, pp. 240–254, 1991.

[14] K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 1, 1996, pp. 318–322 vol.1.

[15] K. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *IEEE International Conference on Intelligent Robots and Systems*, 11 2006, pp. 4647 – 4653.

[16] F. Park and B. Martin, "Robot sensor calibration: solving ax=xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[17] S. Gwak, J. Kim, and F. Park, "Numerical optimization on the euclidean group with applications to camera calibration," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 65–74, 2003.

[18] R. Horaud and F. Dornaika, "Hand-eye calibration," *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, 1995.

[19] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)*, 1999, pp. 430–436.

[20] M. Shah, "Solving the robot-world/hand-eye calibration problem using the kronecker product," *Journal of Mechanisms and Robotics*, vol. 5, no. 3, p. 031007, 2013.

[21] A. Li, L. Wang, and D. Wu, "Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product," *International Journal of the Physical Sciences*, vol. 5, no. 10, pp. 1530–1536, 2010.

[22] M. Ulrich and M. Hillemann, "Generic hand–eye calibration of uncertain robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 060–11 066.

[23] Y. Wang, W. Jiang, K. Huang, S. Schwertfeger, and L. Kneip, "Accurate calibration of multi-perspective cameras from a generalization of the hand-eye constraint," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1244–1250.

[24] D. Evangelista, M. Terreran, A. Pretto, M. Moro, C. Ferrari, and E. Menegatti, "3d mapping of x-ray images in inspections of aerospace parts," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1223–1226.

[25] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

[26] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*. Springer, 1978, pp. 105–116.

[27] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3607–3613.

[28] R. Tsai and R. 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, 1989.

[29] D. Evangelista, D. Allegro, M. Terreran, A. Pretto, and S. Ghidoni, "An unified iterative hand-eye calibration method for eye-on-base and eye-in-hand setups," in *2022 27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022.