Multi-Camera Hand-Eye Calibration for Human-Robot Collaboration in Industrial Robotic Workcells

Davide Allegro†, Matteo Terreran† and Stefano Ghidoni†

***Abstract*—In industrial scenarios, effective human-robot collaboration relies on multi-camera systems to robustly monitor human operators despite the occlusions that typically show up in a robotic workcell. In this scenario, precise localization of the person in the robot coordinate system is essential, making the hand-eye calibration of the camera network critical. This process presents significant challenges when high calibration accuracy should be achieved in short time to minimize production downtime, and when dealing with extensive camera networks used for monitoring wide areas, such as industrial robotic workcells. Our paper introduces an innovative and robust multi-camera hand-eye calibration method, designed to optimize each camera's pose relative to both the robot's base and to each other camera. This optimization integrates two types of key constraints: i) a single board-to-end-effector transformation, and ii) the relative camera-to-camera transformations. We demonstrate the superior performance of our method through comprehensive experiments employing the METRIC dataset and real-world data collected on industrial scenarios, showing notable advancements over state-of-the-art techniques even using less than 10 images. Additionally, we release an open-source version of our multi-camera hand-eye calibration algorithm at https://github.com/davidea97/Multi-Camera-Hand-Eye-Calibration.git**

***Index Terms*—Calibration and Identification, Sensor Networks, Human-Robot Collaboration**

I. Introduction

HUMAN-robot collaboration (HRC) aims to a close and direct interaction between humans and robots to achieve a common objective, leveraging the synergy between human intelligence and manipulation capabilities and robot precision [1], [2], [3]. This collaborative pattern is spreading significantly in industries, fostering greater production flexibility while maintaining efficiency and productivity [4], [5], [6]. Several projects dealing with industrial scenarios, such as Sharework[1] and DrapeBot[2], have recently proposed to supervise the robotic workcell avoiding occlusions problems by means of a multi-camera system positioned around a robot arm, as shown in Figure 1. This enables the continuous monitoring of both the robot workspace and human worker activities throughout the collaboration process [7], [8]. As the robot and each sensor natively defines its own reference system, it is essential to express information provided by each sensor in a common reference frame—a convenient option here is the robot base coordinates system, to make the robot aware of its surroundings. Such process is generally known as hand-eye calibration, and aims to determine the relative transformation between the robot base and a camera by moving a calibration pattern attached to the robot end-effector to different positions in front of the camera [9].

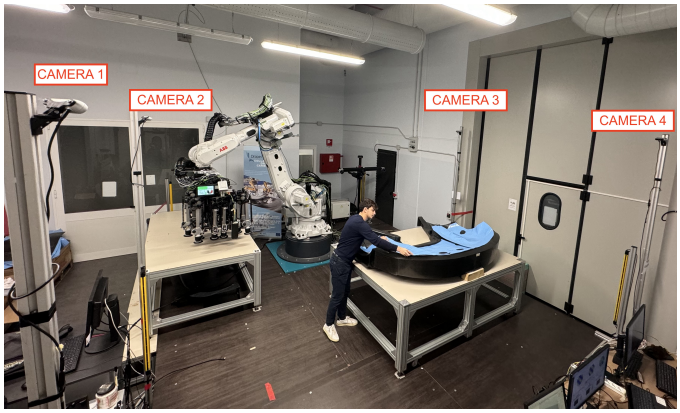

Fig. 1. A large industrial robotic workcell equipped with a camera network around an ABB robot arm, enabling human-robot collaboration task in a carbon fiber draping process as foreseen in the DrapeBot European Project.

When dealing with multiple cameras, calibrating all of them with respect to the robot can be a challenging task: i) the calibration pattern must be compact to avoid collisions during the robot's movement; ii) the calibration often involves a large camera network, necessary to monitor the whole robotic workcell, dealing with rather large distances among cameras and robot; iii) the calibration process needed to be performed in short time to reduce as much as possible production line downtime; iv) accurate calibration must be provided even with a limited number of images, which is a common occurrence in industrial environments due to the difficulty of moving the robot arm safely in a cluttered space.

In existing literature, only a few works have addressed the challenge of hand-eye calibration for camera networks, particularly in industrial scenarios [10]. When dealing with multiple cameras, traditional methods typically perform hand-eye calibration of each camera separately [11]. This process

Manuscript received: April, 19, 2024; Revised August, 7, 2024; Accepted September, 13, 2024.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 101006732 (DrapeBot).

†All the authors are with the Department of Information Engineering (DEI) at the University of Padova, via Gradenigo 6/B, 35131 Padova, Italy. Email: davide.allegro.1@phd.unipd.it, [matteo.terreran; stefano.ghidoni]@unipd.it

Digital Object Identifier (DOI): see top of this page.

[1]https://sharework-project.eu/

[2]https://www.drapebot.eu/

finds the optimal transformation of each sensor with respect to the robot, then derives relative transformations among cameras either by linking transformations together or through additional stereo calibrations between the camera pairs [12]. This approach often leads to methods that are neither robust nor precise, especially in demanding scenarios like industrial ones, where the risk for error propagation to the final calibration is significantly high [13]. Moreover, these existing methods usually focus on relatively small robotic workcells, positioning cameras approximately 1 meter away from both the calibration pattern and each other [14]. This can be considered a significant limitation creating a gap between the capabilities of current calibration techniques and the demands of industrial environments.

In this paper we introduce a non-linear optimization algorithm to address hand-eye calibration in multi-camera setups within industrial robotic workcells. Our method generalizes the work presented in [15] to multi-camera systems, enabling the simultaneous pose estimation of each camera with respect to all other sensors and to the robot's base reference frame. Unlike conventional methods, which usually focus on calibrating each camera independently, our method introduces two main types of constraint to better optimize the mutual pose of the cameras: i) a single board-to-end-effector transformation and ii) the relative rototranslations camera-to-camera. The former allows to streamline the process eliminating the redundancy of determining that transformation for each individual hand-eye calibration; the latter ensures the optimization of the relative transformation among cameras by exploiting the simultaneous detection of multiple cameras of the calibration pattern. This approach guarantees consistency across poses of all cameras and enhances calibration performance by preventing error propagation that can occur with individual calibrations and the need for further calibration steps to determine transformations between cameras. Generally, when considering camera networks for monitoring robotic workcells, cameras are strategically placed to reduce occlusions and simultaneously capture different viewpoints. This setup easily leads to the simultaneous acquisition of images of the same calibration pattern during calibration, allowing our method to concurrently leverage multi-camera information without imposing stringent design requirements on the workcell.

Extensive evaluations on the synthetic and real data of the open source METRIC[3] dataset [16] allowed to investigate the impact of the workcell and pattern sizes on the calibration performances. Additionally, comprehensive experiments on real industrial robotic workcells were necessary to validate the proposed method's robustness, precision, and applicability in demanding industrial environments.

In summary, our work offers three main contributions:

1) A novel multi-camera hand-eye calibration method for calibrating multiple sensors with respect to a robot and to each others, characterized by two key constraints in the optimization procedure: a single board-to-end-effector and relative camera-to-camera transformations;
2) A comprehensive performance evaluation of the proposed method against state-of-the-art hand-eye calibration techniques using the METRIC dataset;
3) A thorough comparative analysis of our approach in real-world industrial settings, outperforming state-of-the-art methods in challenging scenarios of large camera network and limited number of images available for each camera.

[3]https://zenodo.org/records/7976757

II. Related works

Single-camera hand-eye calibration. In the literature, several methods have been proposed to tackle hand-eye calibration in single-camera setups. Some of these approaches evaluate the solutions to the homogeneous equation $AX = ZB$ as shown in Figure 2, with the aim of minimizing translation and rotation errors. Here, A and B denote the camera-to-board and the transformation between the robot's end-effector with respect to its base, respectively. While X and Z are the unknown transformations that have to be estimated. Among these, Tsai *et al.* [17] estimated separately translation and rotation with angle-axis representation, Park *et al.* [18] proposed a solution based on Lie algebra, Daniilidis *et al.* used a dual quaternion parametrization [19], while Liang *et al.* [20] and Andreff *et al.* [21] proposed the Kronecker product parametrization. More recently, Shah *et al.* [22] formulated a closed-form solution using an SVD-based algorithm and the Kronecker product to solve for rotation and translation separately; Li *et al.* [23] employed both Kronecker product and dual quaternions to solve the hand-eye calibration problem. However, all these methods depend on directly estimating the board-to-camera transformation A using the Perspective-n-Point algorithm [24]. This approach can introduce errors, particularly when handling blurred images that hinder precise pattern detection.

On the other hand, alternative methods are based on the minimization of visual quantities, specifically the re-projection error. This process involves minimizing the difference between the observed control points of the calibration pattern attached to the robot end-effector and their corresponding re-projected points, (i.e., the 3D control points of the calibration pattern re-projected back onto the camera's image plane). In this context, multiple methods have been introduced: Evangelista *et al.* presented an hand-eye calibration method for single-camera configurations across different setups [15]. Koide *et al.* proposed an hand-eye calibration method, implementing the minimization of the re-projection error as a pose graph optimization problem, demonstrating high accuracy at a high computational cost [25]. These approaches offer a notable advantage by directly leveraging the calibration pattern images, removing the need for explicit camera pose estimation, which typically requires the PnP algorithms [26], [27].

Multi-camera hand-eye calibration. In scenarios involving multiple cameras, the spatial transformation between cameras is often determined either by performing hand-eye calibration for each individual camera and applying a transformation chain, or by calibrating just one camera using hand-eye calibration followed by stereo camera calibration [12]. However,

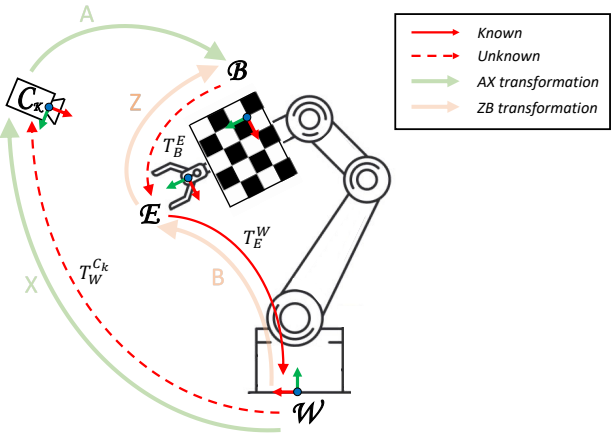

Fig. 2. Transformations chain in a single-camera hand-eye setup, illustrating the re-projection of calibration pattern corners onto the image plane. The robot's end effector pose relative to its base (W) is denoted by T_E^W, along with two unknown matrices: $T_W^{C_k}$, representing the hand-eye transformation, and T_B^E, denoting the transformation between the board and the robot's end-effector pose.

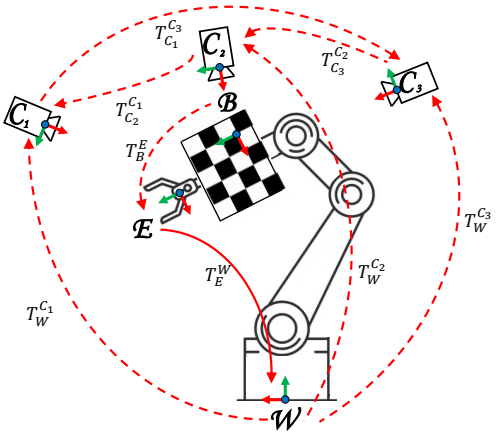

Fig. 3. Multi-camera hand-eye setup, illustrating the geometric transformations optimized through the proposed multi-camera hand-eye calibration method. They include the single board-to-end-effector transformation common to all cameras, the spatial constraints among the cameras, and all the hand-eye transformations.

both approaches carry the risk of error propagation. Only few works in the literature address the simultaneous calibration of a camera network with respect to a robot, probably because of the complexity of the task, which involves managing and integrating visual data from multiple perspectives at the same time. Wang *et al.* [13] proposed a multi-camera calibration method to handle non-overlapping camera network setups, however relying on an external motion capture system to achieve precise camera position during the calibration process. Tabb [10] proposed a robot-world hand-multiple-eye calibration for a small robotic workcell, relying on the minimization of the corner re-projection error, considering the board-to-end-effector transformation Z to be unique for all cameras. Evangelista *et al.* [28] proposed an hand-eye calibration method for a multi-camera setup based on a pose-graph optimization, proving to be accurate, but at the same time really time-consuming. A notable limitation of these methods is their effectiveness mainly within small robotic workcells. The underlying assumption of these methods is that the cameras must be positioned relatively close to the calibration pattern, typically within a distance of about one meter, to ensure precise detection of the board. This proximity requirement, however, often does not align with the spatial configurations commonly found in real-world industrial environments. In many practical settings, especially in larger or more complex workcells, the calibration pattern, and thus the robot, needs to be placed further from the cameras to accommodate the operational layout and the movement of humans and robots within the workspace. In this scenario, the detection of the calibration pattern becomes challenging, negatively affecting the final calibration. Additionally, many of aforementioned approaches require a significant number of images to converge to an optimal solution, which is challenging to ensure in industrial scenarios. This discussion highlights a critical gap between existing calibration methods and the real needs of industrial environments, a gap that is addressed in our work.

III. Methodology

This section introduces our novel multi-camera hand-eye calibration method, that generalizes the approach introduced in [15] to configurations involving multiple cameras. As in [15], our calibration method is based on the minimization of the re-projection error by means of non-linear optimization, but it introduces two main constraints: i) a single transformation between the calibration pattern and the robot's end-effector removing the need for calculating that transformation independently for each camera, and ii) the spatial transformation among the cameras ensuring consistency among all relative transformations between the cameras and the robot base. Section III-A briefly summarizes the single-camera hand-eye calibration work proposed in [15], presenting the main notations that will be adopted for the formulation of our method. In Section III-B, a comprehensive and detailed explanation of the proposed multi-camera hand-eye calibration method is described.

A. Single-camera hand-eye calibration

In [15] we presented an hand-eye calibration technique which does not rely on the PnP algorithm for the estimation of the transformation camera-to-board A (see Figure 2), but rather solves the calibration through the minimization of the re-projection error.

Consider the single-camera setup shown in Figure 2. Given a set of M pairs of robot poses T_E^W and images acquired by a camera C_k, we aim to estimate the unknown rototranslations $T_W^{C_k}$ and T_B^E by minimizing the objective function c reported in (1). This cost function represents the euclidean distance between the detected 2D corners $p_{ij}^D = (u_x, u_y)_{ij}^D$ of the calibration pattern and their corresponding 3D corners re-projected on the image plane, denoted by $(u_x, u_y)_{ij}^P$.

$$c = \sum_{j=0}^{M-1} \sum_{i=0}^{L-1} \left\| \begin{pmatrix} u_x \\ u_y \end{pmatrix}_{ij}^{\mathrm{P}} - \begin{pmatrix} u_x \\ u_y \end{pmatrix}_{ij}^{\mathrm{D}} \right\|^2 \tag{1}$$

Here, $j = 0, \ldots, M-1$ denotes the j^{th} pose of the robot's end-effector with respect to its base and $i = 0, \ldots, L-1$ is the i^{th} corner of the calibration pattern.

In particular, consider the 3D coordinates P_i^B of the i^{th} corner in the calibration pattern reference frame B, and the function $\pi_k(P_i)$ that describes the projection of a 3D point in the camera frame onto the image plane of a camera C_k with known intrinsic and distortion parameters. The projection of the 3D corners of the calibration pattern onto the image plane is given by:

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix}_{ij}^{\mathrm{P}} = \pi_k \left(P_i^{C_k} \right) = \pi_k \left(T_W^{C_k} [T_E^W]_j [T_B^E]_k P_i^B \right) \quad (2)$$

where the 3D corners P_i^B are transformed in the camera frame by means of the chain of transformations depicted in Figure 2. While the transformation $\left[T_E^W\right]_j$ describing the j^{th} end-effector pose is known from the robot kinematics, the remaining transformations are unknown: the hand-eye transformation $T_W^{C_k}$, and the rototranslation $[T_B^E]_k$ from the calibration board to the robot's end-effector for camera C_k. Therefore, the cost function for calibrating a single camera C_k can be rewritten as in the equation (3).

$$c_k = \sum_{j=0}^{M-1} \sum_{i=0}^{L-1} \left\| \pi_k \left(T_W^{C_k} \left[T_E^W\right]_j [T_B^E]_k P_i^B \right) - p_{ijk}^D \right\|^2 \quad (3)$$

B. Multi-camera hand-eye calibration

The method presented in Section III-A could easily be used to calibrate a network of cameras by simply applying it separately to each camera in the network. However, in the case of a network of N cameras this approach leads to estimating N hand-eye transformations $T_W^{C_k}$ and N board-to-end-effector transformations $\left[T_B^E\right]_k$ independent of each other. On one hand, this can be very inefficient since N different calibration processes are needed; on the other hand, this can lead to poor calibration performance of the entire camera network since the relative transformation $T_{C_t}^{C_k}$ between two cameras C_k and C_t can be affected by accumulated errors in translation and rotation by chaining their corresponding hand-eye transformations $T_W^{C_k}$, $T_W^{C_t}$. This motivates us to introduce two main types of constraints in the multi-camera calibration process to better exploit the data available in such a scenario, namely: (i) a common single board-to-end-effector transformation T_B^E for all cameras and (ii) the relative camera-to-camera transformation $T_{C_t}^{C_k}$ for each pair of cameras (C_t, C_k) which detects the calibration pattern at the same acquisition step.

The former constraint derived from the fact that all cameras are calibrated using the same pattern rigidly attached to the robot, and all images are acquired simultaneously moving just the robot arm: the estimated board-to-end-effector transformation should then be the same for all cameras in the network. Generalizing (3) to a multi-camera setup, the overall cost function to be minimized is given by the sum of the re-projection errors of the N cameras, imposing the same transformation T_B^E for all cameras:

$$c_{rpj} = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \sum_{i=0}^{L-1} \left\| \pi_k (T_W^{C_k} [T_E^W]_j T_B^E) P_i^B - p_{ijk}^D \right\|^2 \quad (4)$$

Generally, cameras are positioned to minimize occlusions and capture different viewpoints of the same scene simultaneously. The same calibration pattern can thus be observed by multiple cameras simultaneously during the data collection phase. As shown in Figure 3, this introduces an additional path to project the pattern's 3D corners onto the image plane of camera C_k: we can either use the transformation $T_W^{C_k}$ as in (2) or the transformation $T_{C_t}^{C_k} T_W^{C_t}$ passing through a camera C_t which concurrently detected the pattern.

Based on such observation, we proposed an additional cost function c_{cross} which aims to minimize the difference between corners detected by camera C_k and their re-projection onto this camera's image plane through camera C_t:

$$c_{cross} = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \sum_{t=0}^{N-1} \sum_{i=0}^{L-1} \left\| p_{ijkt}^{cross} - p_{ijk}^D \right\|^2 \quad (5)$$

This process is crucial for refining the relative transformations $T_{C_t}^{C_k}$ between the cameras, exploiting the occurrence of cross-detections—when the checkerboard is simultaneously detected by more cameras. A cross-detection matrix $\mathbf{X}_j$ is used within the optimization framework to describe when two cameras are jointly detecting the calibration pattern. This matrix consists of binary values in each cell, $\mathbf{X}_j(k,t)$, where $k \neq t$, indicating the concurrent detection of the calibration pattern by cameras k and t at the j^{th} time step. Consequently, the re-projected corners p_{ijkt}^{cross} can be computed as shown in (6).

$$p_{ijkt}^{cross} = \pi_k \left(T_{C_t}^{C_k} T_W^{C_t} [T_E^W]_j T_B^E P_i^B \right) \cdot \mathbf{X}_j(k,t) \quad (6)$$

Overall, the cost function to be optimized can be summarized as the sum of two main contributions, the term c_{rpj} aiming to minimize the re-projection error for each individual camera and the term c_{cross} to impose constraints on the relative pose of pairs of cameras:

$$\underset{T_{C_t}^{C_k}, T_W^{C_t}, T_B^E}{\operatorname{argmin}} \quad c_{rpj} + c_{cross} \quad (7)$$

Note how the proposed approach does not require that two or more cameras detect the pattern simultaneously, but rather it is able to take advantage of that opportunity if it occurs. When no pair of cameras simultaneously detects the pattern, the matrix $\mathbf{X}_j$ becomes a matrix of all zeros and no additional constraints between cameras are used in the optimization process: the term c_{cross} is neglected and the collected images are used only for the calculation of the reprojection error of each individual camera in the term c_{rpj}.

To our knowledge, this is the first multi-camera hand-eye calibration that takes advantage of both constraints through a non-linear optimization algorithm. The proposed method is implemented using the Ceres Solver [29] library, adding a Cauchy loss function to enhance resilience against outliers.

IV. Performance evaluation procedure

To comprehensively assess the effectiveness and the robustness of our method, we carried out a series of calibration experiments. These experiments compared our method against other state-of-the-art calibration techniques using the publicly available METRIC dataset and data acquired in two real industrial environments designed for human-robot collaboration. By employing the METRIC dataset, we aimed to validate the calibration method's precision, leveraging the dataset's ground truth data provided for both synthetic and real scenarios. Notably, the dataset features images captured by a network of cameras surrounding the robot arm, with three different workcell sizes. This enables a thorough evaluation of our method's performance considering the variations in distances between the cameras and the calibration pattern. The experiments conducted in industrial environments aimed to assess our method's suitability and robustness in complex and challenging industrial contexts. These settings are particularly demanding due to their larger robotic workcells and the limited availability of data due to the difficulties associated with capturing numerous images within such cluttered areas.

In the experiments on METRIC we consider the average translation error (e_t^{GT}) and rotation error (e_θ^{GT}), defined as:

$$e_t^{GT} = \frac{\sum_{i=0}^{N-1} \left\| t - \hat{t} \right\|_2}{N} \tag{8}$$

$$e_\theta^{GT} = \frac{\sum_{i=0}^{N-1} angle(R^T \hat{R})}{N} \tag{9}$$

where N represents the number of sensors belonging to the camera network. In these equations, t and R represent the translation vector and rotation matrix provided in the ground truth data, while $\hat{t}$ and $\hat{R}$ are the values estimated through the calibration process, all related to the hand-eye transformations $T_W^{C_k}$. Note that rotation error is defined considering the angle of the relative rotation between R and $\hat{R}$ using the axis-angle representation, which is computed as $angle()$ in (9). For the industrial performance evaluation of our method, since the ground truth data are not available, we adopt the metric used in several hand-eye calibration papers [10], [13], [30], which is obtained from the decomposition of the homogeneous equation $AX = ZB$ described in Section II. Specifically, the translation error e_t and the rotation error e_θ are computed as shown in (10) and (11).

$$e_t = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left\| (R_{A_j} t_{X_i} + t_{A_j}) - (R_Z t_{B_j} + t_Z) \right\|_2 \tag{10}$$

$$e_\theta = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} angle\left((R_{A_j} R_{X_i})^T (R_Z R_{B_j}) \right) \tag{11}$$

Here, $j = 0, \ldots, M-1$ denotes the j^{th} robot pose achieved during the image acquisition. The error is evaluated as the average over all the M captured images for all the N cameras. In scenarios with no ground truth, as illustrated in the industrial settings discussed in Section VI, only the equations related to the second error metric were employed for evaluation. Instead, when experiments were conducted on the METRIC dataset, the primary metrics for assessment were (8) and (9), with the additional metrics (10) and (11) used to demonstrate the correctness of those metrics and its consistency with the ground truth results. Moreover, for each experiment, we computed the runtime of each hand-eye calibration method used to calibrate the robotic workcell. Experiments were performed on a MSI Stealth 15M PC running Linux Ubuntu 20.04, equipped with an Intel Core i7-1280P 20-core CPU clocked at 3.20 GHz and 16 GB of DDR4 RAM. With such specifications an iteration of the optimization process takes on average 0.3 seconds. In all experiments, the identity matrix was used as the initial guess for the unknown rototranslation matrices in the optimization process as a worst-case performance estimate (i.e., no a priori information on camera locations).

V. Results on METRIC dataset

In this section we report the experimental results obtained on the METRIC dataset [16], considering the error metrics defined in Section IV. Subsections V-A and V-B analyze the performance achieved with synthetic and real images from the dataset, respectively. Note that the METRIC dataset considers workcells with different sizes, all equipped with four sensors and an A4 paper checkerboard as calibration pattern, whose inner corners are arranged in a 4×3 grid with a spacing of about 5 cm. This allows to investigate how limited sizes of calibration patterns affect calibration performances, which is one of the main limitation of calibrating cameras in large robotic workcells designed for human-robot collaboration tasks.

A. METRIC: synthetic data

The synthetic data used in METRIC comprises images acquired by 4 cameras in simulated robotic workcells with various sizes, encompassing small, medium, and large workcells, covering an area of approximately 6 m^2, 12 m^2, and 20 m^2 respectively. In Table I the errors and the time required for running each algorithm are reported, distinguishing the single-camera and multi-camera methods.

The results clearly prove that the robotic workcell size has a significant impact on the calibration accuracy for various hand-eye calibration methods. Notably, as the mean distance between sensors and the calibration pattern extends, introducing a more complex scenario for corner detection, consistently both translation errors (e_t, e_t^{GT}) and rotation errors (e_θ, e_θ^{GT}) exhibit a decline, in accordance with our previous results [16]. It is observed that numerous methods [10], [15], [18], [19] experience a notable reduction in calibration efficacy within larger robotic workcells, while others [17], [21], [25] may even diverge from the ideal solution in the large robotic workcell. However, the multi-camera hand-eye calibration method presented in this paper emerges as the most robust and consistently accurate in all three scenarios, demonstrating minimal sensitivity to variations in robotic workcell size across all error metrics.

In particular, in the case of the large robotic workcell, the proposed method achieves remarkable results, ensuring translation errors of approximately 1 mm with respect to the ground

TABLE I
AVERAGE ERRORS FOR ALL CAMERAS ACHIEVED BY HAND-EYE CALIBRATION TECHNIQUES IN METRIC SIMULATED WORKCELLS. BOLD AND UNDERLINED VALUES INDICATE THE 1^{st} AND 2^{nd} TOP-PERFORMING CALIBRATION METHODS FOR EACH METRIC, RESPECTIVELY.

Method	**Small workcell**					**Medium workcell**					**Large workcell**				
	Ground truth		AX=ZB		Time [s]	Ground truth		AX=ZB		Time [s]	Ground truth		AX=ZB		Time [s]
	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]		e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]		e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]	
Evangelista [15]	1.93	0.03	1.98	0.05	14.99	5.46	0.07	6.43	0.08	15.46	19.98	0.29	35.21	0.53	25.10
Tsai [17]	427.16	2.06	40.26	0.62	0.08	820.90	6.72	47.29	1.12	0.11	587.59	0.21	35.47	0.56	**0.09**
Park [18]	3.27	0.04	2.35	0.12	0.12	5.27	0.03	4.47	0.18	0.16	12.06	0.20	4.97	0.18	0.13
Danilidis [19]	42.43	0.58	15.34	0.37	0.12	100.09	3.27	39.24	1.08	0.17	57.58	0.17	24.9	0.70	0.12
Andreff [21]	10.05	0.05	8.72	0.33	0.26	91.83	0.38	80.84	1.30	0.33	149.53	0.34	116.54	2.89	0.26
Shah [22]	1.90	0.05	2.63	0.13	0.07	3.03	0.05	4.37	0.19	**0.08**	5.35	0.08	4.51	0.16	0.12
Li [23]	1.93	0.05	2.71	0.14	**0.06**	3.15	0.05	4.55	0.19	0.09	5.87	0.08	4.58	0.16	**0.09**
Koide [25]	1.68	0.03	1.46	0.07	116.1	3.94	**0.02**	2.31	0.11	88.82	1236.30	0.04	773.11	5.38	77.38
Tabb [10]	2.52	0.20	2.86	0.32	69.45	5.79	0.36	7.35	0.38	87.92	13.52	0.89	17.45	0.65	87.98
Evangelista [28]	1.02	**0.02**	1.14	0.05	237.97	3.38	**0.02**	2.02	0.10	311.67	1.43	0.02	4.02	0.15	249.89
Ours III-B	**0.71**	**0.02**	**0.45**	**0.03**	4.67	**0.75**	0.02	**0.83**	**0.05**	13.78	**1.08**	**0.01**	**2.02**	**0.09**	27.13

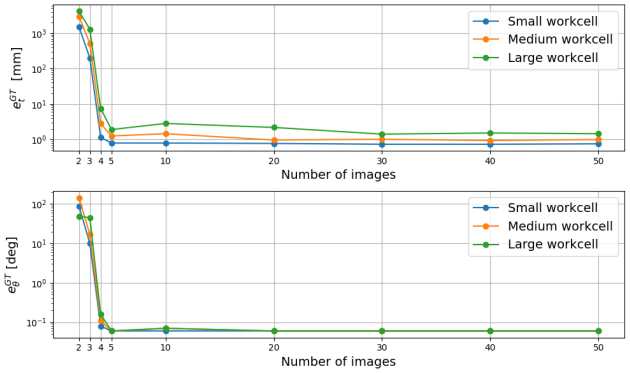

Fig. 4. Analysis of translation error e_t^{GT} and rotation error e_θ^{GT} for the METRIC simulated workcells as the number of images per camera used in the calibration process varies. Errors are expressed in a logarithmic scale.

truth, and rotation errors as low as 0.01 deg. The method's robustness to potential misdetections in large workcells is due to its optimization process implementation, which not only focuses on minimizing re-projection errors on a single camera but also leverages minimization across other cameras, particularly when they simultaneously capture the calibration pattern. Conversely, as expected in terms of optimization time, single-camera calibration methods (i.e., first group of rows in Table I) prove significantly faster, benefiting from solving smaller sets of homogeneous equations and generally exhibiting lower algorithmic complexity. However, among multi-camera calibration methods, the presented approach stands out as the fastest, offering a balance between accuracy and efficiency. To further investigate the robustness of our method, we analyze its performance as the number of images used in the calibration process varies (i.e., from 2 to 50 images per camera, randomly selected). As shown in Fig. 4, our method achieves good performance for all workcell sizes even in the case of a limited number of available images: with at least 5 images for each camera, the method achieves performance very similar to that achievable with a larger number of images, while above 30 images performance saturates.

B. METRIC: real data

The real images of METRIC were captured in two robotic workcells of different size—one small, covering an area of about 7 m^2 typically designed for tasks that involve transferring small objects between human operators and robots in minor assembly applications, and the large one, approximately spanning an area of 15 m^2, configured for human-robot collaboration applications involving several people within the workcell, e.g. for the collaborative transport of large objects. For each robotic workcell layout, three different sets of images were collected by means of different camera networks, each characterized by the use of a particular type of sensor: Intel RealSense Lidar camera L515, Intel RealSense Depth D455 sensor and the Microsoft Kinect V2. As discussed in [16], the size of the workcell is not the only factor influencing calibration; the type of sensor and its characteristics also play a significant role in the calibration pattern detection and, consequently, in the calibration process. As illustrated in Table II, the multi-camera hand-eye calibration method proposed in this work consistently outperforms other methods and it ensures convergence to an optimal solution, not achievable by some single-camera methods [19], [22], [23]. This can primarily be attributed to the incorporation of two additional constraints, which enhance the algorithm's robustness in real-world scenarios where corner detection may be imprecise. In particular, our method achieves a translation error e_t^{GT} lower than 52 mm and a rotation error e_θ^{GT} lower than 0.3 deg, demonstrating effectiveness even within the larger workcell. Notably, even with the lower-resolution Intel RealSense D455 sensor, the proposed method achieves an average error comparable to that obtained with the other sensors, guaranteeing superior performance with respect to all other state-of-the-art methods. In general, multi-camera calibration methods demonstrate greater accuracy compared to running N times the single-camera methods, which do not have the possibility of mitigating the impact of inaccurate pattern detection through other points of view that are simultaneously considered. Our optimization process, while not as rapid as single-camera methods that address calibration problem through closed-form solutions [17], [18], [19], [21], [22], [23]—which complete in less than 1 second— is comparable with single-camera approaches [15], [25] that employ non-linear and graph optimization techniques. Moreover, it stands out as the quickest among other multi-camera methods, achieving calibration in some instances over ten times faster. It's important to note that while some methods may achieve faster results, they often do so at the cost of accuracy. Our approach, however, maintains an optimal balance, offering both speed and precision, thereby positioning it among the top-performing strategies. The availability of ground truth on real data in METRIC makes it possible to verify the reliability of the metrics (10) and (11) in a real-

TABLE II
AVERAGE ERRORS ACHIEVED BY HAND-EYE CALIBRATION TECHNIQUES ON METRIC REAL WORKCELLS. BOLD AND UNDERLINED RESULTS INDICATE THE 1^{st} AND 2^{nd} TOP-PERFORMING CALIBRATION METHODS, RESPECTIVELY ("−" DENOTES THE NON-CONVERGENCE).

Method	Small real workcell														
	Microsoft Kinect V2					Intel RealSense Depth D455					Intel RealSense LiDAR L515				
	Ground truth		AX=ZB		Time [s]	Ground truth		AX=ZB		Time [s]	Ground truth		AX=ZB		Time [s]
	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]		e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]		e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]	
Evangelista [15]	42.79	0.57	13.21	0.81	7.54	45.14	0.43	72.91	1.22	13.89	26.20	0.39	50.33	3.58	31.27
Tsai [17]	75.13	0.15	12.11	0.90	**0.07**	1215.45	14.32	224.68	10.65	0.09	396.83	0.59	53.51	**0.86**	**0.10**
Park [18]	56.61	0.14	10.42	0.86	0.11	167.70	3.48	69.20	2.13	0.08	72.40	0.12	40.32	1.79	0.13
Daniilidis [19]	49.62	0.14	10.32	0.94	0.11	1425.42	0.43	541.38	1.69	0.09	316.34	0.30	71.11	1.77	0.16
Andreff [21]	235.17	0.58	127.61	5.51	0.22	1101.90	11.95	759.66	11.76	0.19	596.57	6.78	443.67	4.75	0.27
Shah [22]	27.22	0.75	10.37	0.94	0.08	23.90	0.54	65.90	2.31	0.08	18.51	0.34	30.93	1.68	0.12
Li [23]	66.67	0.73	13.89	0.93	0.10	51.03	0.72	100.29	2.40	**0.07**	23.70	0.31	34.27	1.64	0.11
Koide [25]	46.52	0.12	12.84	0.68	125.14	72.87	0.28	22.30	2.91	53.30	36.67	0.78	82.95	2.96	122.62
Tabb [10]	51.57	0.73	11.20	0.81	145.31	63.98	1.36	81.32	2.76	65.22	34.59	0.94	67.23	3.56	89.21
Evangelista [28]	42.42	**0.09**	7.31	0.56	215.23	41.80	0.34	11.10	1.13	210.61	24.11	0.44	33.75	1.70	198.42
Ours III-B	**22.01**	**0.09**	**6.54**	**0.55**	16.79	**13.21**	**0.07**	**9.21**	**0.66**	20.11	**13.68**	**0.02**	**24.95**	1.32	42.21

Method	Large real workcell														
	Microsoft Kinect V2					Intel RealSense Depth D455					Intel RealSense LiDAR L515				
	Ground truth		AX=ZB		Time [s]	Ground truth		AX=ZB		Time [s]	Ground truth		AX=ZB		Time [s]
	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]		e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]		e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]	
Evangelista [15]	77.26	0.77	123.12	3.21	58.32	136.39	1.33	57.32	4.32	14.32	60.59	0.43	54.36	1.66	14.37
Tsai [17]	1924.88	18.78	318.14	15.30	**0.09**	2759.10	14.91	461.62	15.25	**0.06**	1881.75	11.53	222.23	7.50	**0.08**
Park [18]	336.86	1.08	178.91	4.69	0.14	355.10	5.33	226.85	5.10	0.09	233.57	3.98	70.63	1.60	0.13
Daniilidis [19]	1757.97	13.23	629.15	6.07	0.14	–	–	–	–	–	8359.06	0.32	4176.45	1.60	0.13
Andreff [21]	2370.61	12.34	1639.04	12.20	0.29	2554.29	35.89	1680.34	7.25	0.18	1881.23	12.79	1310.25	10.94	0.26
Shah [22]	54.92	0.73	75.34	2.62	0.12	–	–	–	–	–	26.15	0.31	38.12	2.02	0.09
Li [23]	129.39	0.72	111.44	2.61	0.13	–	–	–	–	–	26.39	0.30	66.42	2.49	0.12
Koide [25]	69.38	0.35	20.72	1.04	201.45	65.39	0.32	45.21	2.13	28.41	59.58	0.11	15.05	1.05	25.86
Tabb [10]	105.01	0.75	154.21	5.32	165.22	153.20	1.74	167.34	6.35	78.23	75.33	0.77	45.23	1.21	78.23
Evangelista [28]	63.54	**0.24**	15.79	0.71	275.87	59.33	0.22	48.50	**2.10**	105.91	50.18	**0.09**	**11.63**	0.89	260.59
Ours III-B	**51.18**	0.30	**12.17**	**0.65**	77.99	**45.18**	**0.16**	**35.98**	2.12	35.43	**19.34**	**0.09**	12.45	**0.85**	17.33

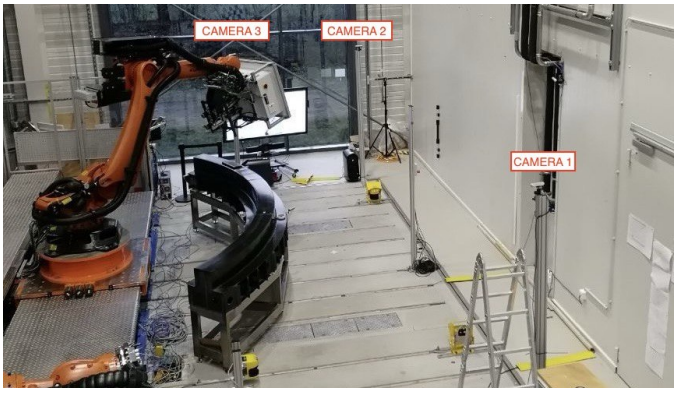

Fig. 5. Kuka industrial workcell equipped with three Intel RealSense Depth D455 surrounding a Kuka manipulator at an average distance of 6 meters.

world scenario, as shown in Table II where they follows the same trend of GT-based metrics (8) and (9) for the methods tested.

VI. RESULTS ON INDUSTRIAL SCENARIOS

To comprehensively evaluate the robustness of our calibration method even in real industrial environments, we performed calibration experiments in two industrial robotic workcells developed as use cases in the DrapeBot project [31]. Both workcells have been designed for human-robot collaboration and collaborative transportation of flexible materials, covering an area of about 20 m^2. The former, illustrated in Figure 1 and identified as the ABB robotic workcell, was equipped with four Intel RealSense Depth D455 sensors surrounding an ABB industrial manipulator at an average distance of 4 meters from the base of the robot. The second workcell, designated as the Kuka industrial workcell, was equipped with a Kuka industrial manipulator surrounded by three RealSense Depth D455 sensors at an average distance of 6 meters from the robot base (Figure 5). Within these two robotic workcells, the image collection comprised 8 images per camera for the first workcell and 15 images per camera for the second one. This limited number was attributed to the challenges associated with moving the robot to various locations and orientations in front of the sensors, due to the presence of large objects, such as the mold depicted in Figure 1 and Figure 5, near the robot, which hinders its movement. The calibration pattern was composed of a checkerboard with 4×3 inner corners, spaced approximately 6 cm apart. The results of the calibration process are reported in Table III. These results are assessed using the metrics defined in (10) and (11), due to the absence of ground truth data.

TABLE III
AVERAGE ERROR ON INDUSTRIAL WORKCELLS.

Method	ABB industrial workcell (8 images per camera)			Kuka industrial workcell (15 images per camera)		
	AX=ZB		Time [s]	AX=ZB		Time [s]
	e_t [mm]	e_θ [deg]		e_t [mm]	e_θ [deg]	
Evangelista [15]	718.94	7.52	6.32	485.92	6.15	12.97
Tsai [17]	172.07	10.15	**0.09**	30.28	1.27	0.13
Park [18]	83.54	11.93	0.12	27.82	2.21	0.17
Daniilidis [19]	77.09	11.14	0.13	36.33	1.42	0.14
Andreff [21]	1378.25	6.06	0.12	71.37	1.66	0.21
Shah [22]	106.99	10.76	0.10	72.94	1.52	0.15
Li [23]	–	–	–	652.44	1.23	**0.11**
Koide [25]	131.08	9.60	7.12	1737.68	10.23	9.45
Tabb [10]	53.76	3.18	8.98	22.65	1.23	17.67
Evangelista [28]	11.95	2.65	11.32	17.38	0.78	36.78
Ours III-B	**6.31**	**0.98**	3.12	**14.12**	**0.67**	7.15

Table III highlights that our proposed multi-camera hand-eye calibration method achieves high accuracy with a translation error of approximately 1 cm and a rotation error lower than 1 deg. In contrast, some single-camera calibration methods (e.g., [15], [23], [25]) struggle to converge to an optimal solution, proving to be unsuitable for such challenging scenarios. Notably, our method shows exceptional accuracy with a limited number of images (8 per camera for the ABB workcell and 15 for the Kuka workcell). This efficiency is significant, especially in the context of industrial robotic workcells, where acquiring images is challenging due to obstacles that restrict

flexible robot movement. Consequently, this approach significantly reduces calibration times and minimizes interruptions in production lines.

VII. Conclusion

This work proposed a multi-camera hand-eye calibration, incorporating in the optimization process two additional constraints not previously considered together in the existing literature, namely the common board-to-end-effector transformation across all cameras and relative camera-to-camera transformations among all sensors. The proposed method was evaluated on the publicly available METRIC dataset, allowing for a thorough assessment of calibration accuracy and robustness in robotic workcells of various sizes equipped with different sensors. Through a comprehensive analysis, our method significantly outperformed other state-of-the-art calibration methods, showcasing an excellent balance between speed and calibration accuracy. The proposed method was also tested in two real-world industrial scenarios, considering robotic workcells designed for human-robot collaboration: it provided the highest accuracy also in such scenarios, resulting once again the most robust approach. Notably, despite the reduced number of images for calibration, it achieved outstanding results compared to other state-of-the-art methods, minimizing execution time. As future work, we plan to further investigate the advantages of the cross-detection constraints, such as analyzing performance as the number of cameras changes and using motion capture systems for in-depth performance analysis with accurate ground truth data.

Acknowledgment

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 101006732 (DrapeBot).

References

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, pp. 957–975, 2018.

[2] W. Kim, L. Peternel, M. Lorenzini, J. Babič, and A. Ajoudani, "A human-robot collaboration framework for improving ergonomics during dexterous operation of power tools," *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102084, 2021.

[3] M. Lorenzini, M. Lagomarsino, L. Fortini, S. Gholami, and A. Ajoudani, "Ergonomic human-robot collaboration in industry: A review," *Frontiers in Robotics and AI*, vol. 9, p. 262, 2023.

[4] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[5] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, p. 100, 2019.

[6] A. C. Simões, A. Pinto, J. Santos, S. Pinheiro, and D. Romero, "Designing human-robot collaboration (hrc) workspaces in industrial settings: A systematic literature review," *Journal of Manufacturing Systems*, vol. 62, pp. 28–43, 2022.

[7] L. Orsag, T. Stipancic, and L. Koren, "Towards a safe human–robot collaboration using information on human worker activity," *Sensors*, vol. 23, no. 3, p. 1283, 2023.

[8] M. Terreran, E. Lamon, S. Michieletto, and E. Pagello, "Low-cost scalable people tracking system for human-robot collaboration in industrial environment," *Procedia Manufacturing*, vol. 51, pp. 116–124, 2020.

[9] S. Su, S. Gao, D. Zhang, and W. Wang, "Research on the hand–eye calibration method of variable height and analysis of experimental results based on rigid transformation," *Applied Sciences*, vol. 12, no. 9, p. 4415, 2022.

[10] A. Tabb and K. M. Ahmad Yousef, "Solving the robot-world hand-eye (s) calibration problem with iterative methods," *Machine Vision and Applications*, vol. 28, no. 5-6, pp. 569–590, 2017.

[11] J. Miseikis, K. Glette, O. J. Elle, and J. Torresen, "Automatic calibration of a robot manipulator and multi 3d camera system," in *2016 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2016, pp. 735–741.

[12] A. Rashd, W. Hardt, A. Kolker, M. Bdiwi, and M. Putz, "Open-box target for extrinsic calibration of lidar, camera and industrial robot," in *2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA)*. IEEE, 2020, pp. 121–125.

[13] 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)*. IEEE, 2022, pp. 1244–1250.

[14] A. Tabb, H. Medeiros, M. J. Feldmann, and T. T. Santos, "Calibration of asynchronous camera networks: Calico," *arXiv preprint arXiv:1903.06811*, 2019.

[15] 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 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2022.

[16] D. Allegro, M. Terreran, and S. Ghidoni, "Metric—multi-eye to robot indoor calibration dataset," *Information*, vol. 14, no. 6, p. 314, 2023.

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

[18] F. C. Park and B. J. 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.

[19] 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. IEEE, 1996, pp. 318–322.

[20] R.-h. Liang and J.-f. Mao, "Hand-eye calibration with a new linear decomposition algorithm," *Journal of Zhejiang University-SCIENCE A*, vol. 9, no. 10, pp. 1363–1368, 2008.

[21] 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)*. IEEE, 1999, pp. 430–436.

[22] 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.

[23] A. Li, L. Wang, and D. Wu, "Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product," *Int. J. Phys. Sci*, vol. 5, no. 10, pp. 1530–1536, 2010.

[24] G. Schweighofer and A. Pinz, "Globally optimal o (n) solution to the pnp problem for general camera models." in *BMVC*, 2008, pp. 1–10.

[25] 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.

[26] A. Malti, "Hand–eye calibration with epipolar constraints: Application to endoscopy," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 161–169, 2013.

[27] A. Tabb and K. M. A. Yousef, "Parameterizations for reducing camera reprojection error for robot-world hand-eye calibration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3030–3037.

[28] D. Evangelista, E. Olivastri, D. Allegro, E. Menegatti, and A. Pretto, "A graph-based optimization framework for hand-eye calibration for multi-camera setups," *arXiv preprint arXiv:2303.04747*, 2023.

[29] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 10 2023. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[30] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, "A comparative review of hand-eye calibration techniques for vision guided robots," *IEEE Access*, vol. 9, pp. 113 143–113 155, 2021.

[31] M. Terreran, S. Ghidoni, E. Menegatti, V. Enrico, P. Nicola, N. Castaman, A. Gottardi, E. Christian, V. Luca, S. Giuseppe *et al.*, "A smart workcell for cooperative assembly of carbon fiber parts guided by human actions," in *Atti della 4ª Conferenza Italiana di Robotica e Macchine Intelligenti*, 2022.