Fiducial Marker based Extrinsic Camera Calibration for a Robot Benchmarking Platform

Timo Korthals*, Daniel Wolf, Daniel Rudolph, Marc Hesse, and Ulrich Rückert

Abstract— **Evaluation of robotic experiments requires physical robots as well as position sensing systems. Accurate systems detecting sufficiently all necessary degrees of freedom, like the famous Vicon system, are commonly too expensive. Therefore, we target an economical multi-camera based solution by following these three requirements: Using multiple cameras to track even large laboratory areas, applying fiducial marker trackers for pose identification, and fuse tracking hypothesis resulting from multiple cameras via extended Kalman filter (i.e. ROS's robot_localization). While the registration of a multi-camera system for collaborative tracking remains a challenging issue, the contribution of this paper is as follows: We introduce the framework of Cognitive Interaction Tracking (CITrack). Then, common fiducial marker tracking systems (ARToolKit, AprilTag, ArUco) are compared with respect to their maintainability. Lastly, a graph-based camera registration approach in SE(3), using the fiducial marker tracking in a multi-camera setup, is presented and evaluated.**

I. INTRODUCTION

Tracking systems detecting robots' poses, to perform experiments with necessary accuracy, are in high demand in scientific labs but rarely available. Thus, robotic developments and experiments are done in simulation while a real-world evaluation is just done qualitatively. This approach has two downsides, simulation often does model-simplifications and thus behaves differently from real-world experiments regarding actor and sensor systems. Moreover, bringing the physical robot into simulation requires time and skills in multiple disciplines, and vice versa transferring the developed algorithms back from simulation to real-life makes parameter tuning necessary in general. Therefore, direct evaluation in real-life simplifies the development drastically.

The outline of this work is as follows: we first give an overview of related and fundamental work in Section II. Second, an overview of the *CITrack* systems architecture is given in Section IV. Section III introduces calibration techniques based on graph-optimization. Further, the experimental setup is presented in Section V including discussions on optimizing the camera system for the fiducial marker tracking task as well as an end-to-end localization evaluation using a Vicon Tracking System. Finally, current applications and future prospects are discussed in Section VI.

II. RELATED WORK

Multi-robot test-benches which use vision based fiducial marker tracking for identifying numerous individuals in a scene do exist. Commonly, they are designed with educational purposes in mind. For example, students or professionals can upload their experiment's specification remotely to a test-bench server such, that their experiment is queued, executed, and evaluated automatically. Unfortunately, all approaches suffer from the fact that either the full six dimensional pose can not be retrieved, or the design is not applicable to multi-camera or -modal setups.

While this work has the goal of designing a camera based tracking system, benchmark systems based on other modalities are neglected but can be found in the survey by Jimnez-Gonzlez et. al [1]. The following list is an overview of various benchmark systems on robotic approaches: *VISNET* [2] is a general purpose tracker based on a multi-camera network which jointly tracks an arbitrary object's in R^3. *Emulab* [3] tracks multiple robots on a coarse grid using *Mezzanine* [4] which tracks a marker, but without identification, in $\mathrm{SO}(2)$. *MiNT-m* [5] is analogue to *Emulab*, but got rid of the grid constrained and introduced it's own colored fiducial marker with identifier encoding. The *SSL-Vision System* [6] is the dual-camera based vision system for the RoboCup Small Size League that offers a robot's pose in $\mathrm{R}^3 \times \mathrm{SO}(2)$ based on colored fiducial markers. The downside of this system is, that it does not handle the camera's extrinsic calibration explicitly, nor the fusion of detected markers in the cameras' frustum-intersections. *Teleworkbench* [7] tracks and identifies multiple fiducial markers in $\mathrm{SO}(2)$ in a single camera setup. The *Experimental Testbed for Large Multirobot Teams* [8] uses LED based markers that flash with their corresponding ID, which is rectified within $\mathrm{SO}(2)$ by multiple cameras. The *Robotarium* [9] uses the *ArUco* [10] fiducial marker tracking in a table-top single-camera setup, capable of tracking in $\mathrm{SE}(3)$, where the overall design is limited to the usage of their *GRITSBot*. Further, robotic driven applications are comprised by Lightbody et. al [11].

III. MARKER BASED CALIBRATION

Camera network calibration has widely been discussed in literature, whereas the sparse bundle adjustment (SBA) is the fundamental approach to intrinsic and extrinsic calibration, visual SLAM, structure from motion, and scene reconstruction [12]. This holds whether the setup consists out of multiple cameras in a static scene or single camera moving through it. SBA optimizes the cameras' parameters and detected objects at the same time but is also very susceptible with respect to the identified objects on the pixel screen [13]. If range measurements already exists and intrinsic parameters are sufficiently calibrated, graph SLAM

Bielefeld University, Cluster of Excellence Cognitive Interaction Technologies, Cognitronics & Sensor Systems, Inspiration 1, 33619 Bielefeld, Germany, http://www.ks.cit-ec.uni-bielefeld.de/, *tkorthals@cit-ec.uni-bielefeld.de

978-1-7281-3605-9/19/$31.00

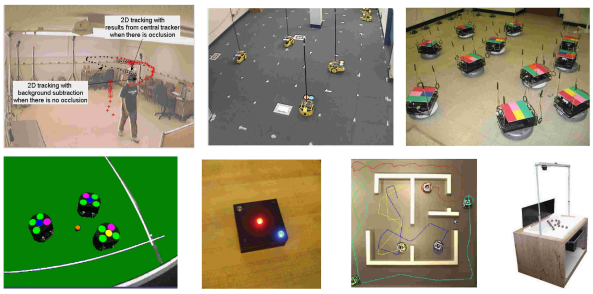

Fig. 1: Tracking and benchmark systems [2], [3], [5], [6], [7], [8], [9] (from left to right and top to bottom)

is the way to optimize just the extrinsics. Both approaches has been recently brought together by Yeguas-Bolivar and Medina-Carnicer proposing a fiducial marker (FM) based graph SLAM approach under the constraints of known FM geometrics [13]. However, every approach builds up a measurement graph which can be optimized, given that the initial parameters are already close to a solution, by the same techniques (e.g. the Levenberg-Marquard (LM) algorithm).

All of the discussed work so far only respects measured objects in R^3 without taking the orientation into account. FM allow the measurements in $\mathrm{SE}(3)$ and thus, the graph optimization can be extended to also respect orientations and therefore simplifies the full SBA approach of being a graph SLAM problem. Therefore, graph SLAM based calibration can be applied not only to camera networks but every heterogeneous multi-sensor setup as long as every sensor provides measurements in $\mathrm{SE}(3)$. However, sensor intrinsics need to be known and measurements needs to be free of systematical error.

Fig. 2 (left) shows the concept of the presented approach, while Fig. 2 (right) represents the corresponding full coordinate transformation (CT) tree of the depicted example. Given a set of cameras $\mathcal{C}$, a set of multiple markers $\mathcal{M}$ is identified at the same time (Fig. 2: $\mathcal{C} = \{i, j\}$ and $\mathcal{M} = \{k\}$). In order to calibrate the camera network, ${}^{\mathrm{O}}\mathbf{T}_i$ and ${}^{\mathrm{O}}\mathbf{T}_j$ need to be optimized, such that the residual transformation ${}^{ik}\mathbf{T}_{jk}$ becomes the identity matrix. The great benefit of the FM approach is, that the identified markers can be directly associated between the camera systems and that the extrinsics of the markers (${}^{i}\mathbf{T}_k$, ${}^{j}\mathbf{T}_k$) are measured. Therefore, visual FM are introduced in Section III-A which is followed by the extended graph SLAM approach to FM based camera network calibration in Section III-B and III-C.

A. Fiducial Marker Detection

Fiducial markers (FM) are objects or patterns in a physical environment which are detectable and localizeable through an exteroceptive sensor [14], [15]. Visual FM had their biggest impact with the advent of augmented reality (AR) applications in camera based systems. The common shape of visual FM for AR in particular is a squared pattern as shown

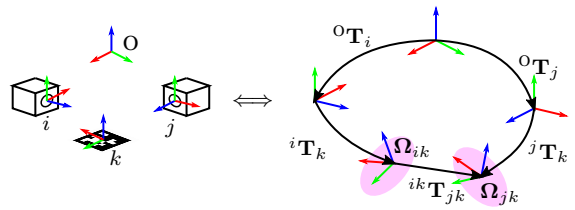

Fig. 2: Camera pair detecting a single FM (left) and corresponding CT tree with erroneous measurement (right).

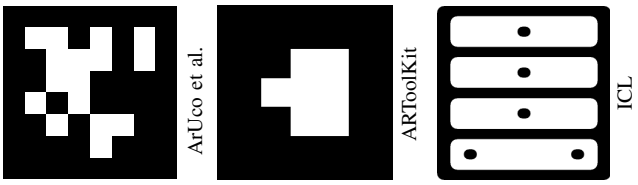

Fig. 3: FM detectable by *ArUco* [10],*ARToolKit* [16], and *ICL* [17] (left to right)

in Fig. 3 where the four edges[1] allow the determining of extrinsic parameters through homography and known intrinsic camera parameters. Although, another famous application is FM-based visual localization systems, since they provide robustness against environmental factors, distinguishability, economical feasibility in production and application, and precision in localization up to $\mathrm{SE}(3)$. Many FM systems has been proposed in the vision community [18]. Among all systems resides a common, two-staged way of how to detect and identify the FM in a scene. The first stage is the hypothese generation which creates a list of regions, together with their transformation parameters (homography or affine), which are likely to contain a marker. The second stage identifies and decodes a hypotheses under transformation, if the region is indeed a marker or just an arbitrary object.

B. Calibration

In order to calibrate the cameras as illustrated in Fig. 2, ${}^{\mathrm{O}}\mathbf{T}_i$ and ${}^{\mathrm{O}}\mathbf{T}_j$ need to be optimized, such that the two transforms ${}^{\mathrm{O}}\mathbf{T}_i\,{}^{i}\mathbf{T}_k$ and ${}^{\mathrm{O}}\mathbf{T}_j\,{}^{j}\mathbf{T}_k$ coincide.

Let $\mathbf{x}_i$ be the state vector consisting of the parameters, which are the extrinsic calibration, of camera i with respect to a reference coordinate system O. Further, let $\mathbf{x}_{ik}$ and $\mathbf{\Omega}_{ik}$ be respectively the mean and the information matrix of measuring FM k via camera i.

To avoid singularities in the over-paramatrized space $\mathrm{SE}(3)$ induced by quaternions, the state vector $\mathbf{x}$ is defined on a manifold expressed by $\mathbf{x} = (x, y, z, q_x, q_y, q_z)$ as proposed by Grisetti et al. [19]. (x, y, z) denote the translatory components, while (q_x, q_y, q_z) being the imaginary components of the unit quaternion $\left(\sqrt{1 - q_x^2 - q_y^2 - q_z^2}, q_x, q_y, q_z\right)$.

[1]the outmost points of a marker provide the greatest number of pixels relatively to the marker's area and thus reducing pose jitter to maximize the accuracy of line equations formed from the border sides

The log-likelihood l, that the measurements of two nodes i and j for one particular FM k coincide, shows the following proportionality:

$$l_{ijk} \propto \mathbf{e}_k\left(\mathbf{x}_i, \mathbf{x}_j\right)^\intercal \mathbf{\Omega}_{ijk} \mathbf{e}_k\left(\mathbf{x}_i, \mathbf{x}_j\right) =: \mathrm{F}_{ijk}\,. \tag{1}$$

Let $\mathbf{e}_k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathrm{t2v}\left({}^{ik}\mathbf{T}_{jk}\right)$ be the state vector of the residual transformation and $\mathrm{t2v}$ be a function that projects a transformation matrix to a state vector on the manifold. All components of $\mathbf{e}$ become zero, if and only if ${}^{ik}\mathbf{T}_{jk}$ is the identity matrix which makes it suitable for gradient descent techniques. Further, let $\mathbf{\Omega}_{ijk}$ be the information matrix of measuring $\mathbf{x}_{jk}$ from $\mathbf{x}_{ik}$ which can be obtained via error propagation between the measurements. Since the full CT tree is known, the residual transformation between i and j can be directly expressed as:

$${}^{ik}\mathbf{T}_{jk} = \left({}^{\mathrm{O}}\mathbf{T}_i\, {}^{i}\mathbf{T}_k\right)^{-1} {}^{\mathrm{O}}\mathbf{T}_j\, {}^{j}\mathbf{T}_k, \tag{2}$$

The goal of a maximum likelihood approach is to find the configuration of the states $\mathbf{x}$ of the cameras that minimizes the negative log-likelihood of all observations:

$$\mathrm{F}(\mathcal{C}, \mathcal{M}) = \sum_{i,j \in \mathcal{C}} \sum_{k \in \mathcal{M}} \underbrace{\mathbf{e}_k(\mathbf{x}_i, \mathbf{x}_j)^\intercal \mathbf{\Omega}_{ijk} \mathbf{e}_k(\mathbf{x}_i, \mathbf{x}_j)}_{\mathrm{F}_{ijk}} \gamma_{ijk}, \tag{3}$$

with γ being an indicator function that is 1, if a FM is seen by camera i and j and 0 otherwise.

C. Error Minimization via Iterative Local Linearizations

If a good initial guess $\hat{\mathbf{x}}$ of the camera poses is known, the numerical solution of Eq. 3 can be obtained by using the Gauss-Newton or Levenberg-Marquardt algorithms. The idea is to approximate the error function by its first order Taylor expansion around the current initial guess $\hat{\mathbf{x}}_{ij} = (\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)$:

$$\mathbf{e}_k(\hat{\mathbf{x}}_i + \Delta\mathbf{x}_i, \hat{\mathbf{x}}_j + \Delta\mathbf{x}_j) \simeq \mathbf{e}_k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) + \mathbf{J}_{ijk}\Delta(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

Here, $\mathbf{J}_{ijk}$ is the Jacobian of $\mathbf{e}_k(\mathbf{x}_i, \mathbf{x}_j)$. For further simplicity of notation, the indices of the measurement are encoded in the residual term as follows: $\mathbf{e}_{ijk} = \mathbf{e}_k(\mathbf{x}_i, \mathbf{x}_j)$. Now substituting Eq. 3 in the residual terms of Eq. 4 leads to:

$$\begin{aligned}
&\mathrm{F}_{ijk}(\hat{\mathbf{x}}_{ij} + \Delta\mathbf{x}_{ij}) \\
&= \mathbf{e}_k(\hat{\mathbf{x}}_{ij} + \Delta\mathbf{x}_{ij})^\intercal \mathbf{\Omega}_{ijk} \mathbf{e}_k(\hat{\mathbf{x}}_{ij} + \Delta\mathbf{x}_{ij}) \\
&\simeq (\mathbf{e}_{ijk} + \mathbf{J}_{ijk}\Delta\mathbf{x}_{ij})^\intercal \mathbf{\Omega}_{ijk} (\mathbf{e}_{ijk} + \mathbf{J}_{ijk}\Delta\mathbf{x}_{ij}) \\
&= \underbrace{\mathbf{e}_{ijk}^\intercal \mathbf{\Omega}_{ijk} \mathbf{e}_{ijk}}_{c_{ijk}} + 2\underbrace{\mathbf{e}_{ijk}^\intercal \mathbf{\Omega}_{ijk} \mathbf{J}_{ijk}}_{\mathbf{b}_{ijk}^\intercal} \Delta\mathbf{x}_{ij} \\
&\quad + \Delta\mathbf{x}_{ij}^\intercal \underbrace{\mathbf{J}_{ijk}^\intercal \mathbf{\Omega}_{ijk} \mathbf{J}_{ijk}}_{\mathbf{H}_{ijk}} \Delta\mathbf{x}_{ij} \\
&= c_{ijk} + 2\mathbf{b}_{ijk}^\intercal \Delta\mathbf{x}_{ij} + \Delta\mathbf{x}_{ij}^\intercal \mathbf{H}_{ijk} \Delta\mathbf{x}_{ij}
\end{aligned} \tag{5}$$

With this approximation for one measurement k between two cameras i and j, the combined log-likelihood in Eq. 3 can be rewritten as

$$\mathrm{F}(\mathcal{C}, \mathcal{M}) = \sum_{i,j \in \mathcal{C}} \sum_{k \in \mathcal{M}} c_{ijk} + 2\mathbf{b}_{ijk}^\intercal \Delta\mathbf{x}_{ij} + \Delta\mathbf{x}_{ij}^\intercal \mathbf{H}_{ijk} \Delta\mathbf{x}_{ij}, \tag{6}$$

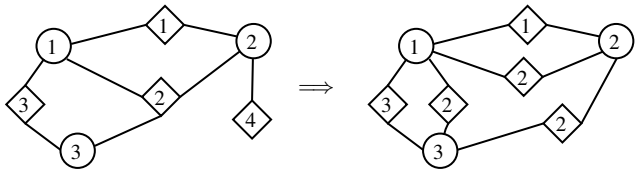

Fig. 4: Pose graphs of cameras (○) and FM measurements (◇). Initial pose graph (left) and marginalized version to fit into linear equation system (right).

and minimized in $\mathbf{x}$ by solving the linear system

$$\underbrace{\sum_{i,j \in \mathcal{C}} \sum_{k \in \mathcal{M}} \mathbf{H}_{ijk} \Delta\mathbf{x}_{ij}}_{\mathbf{H}\Delta\tilde{\mathbf{x}}} = -\underbrace{\sum_{i,j \in \mathcal{C}} \sum_{k \in \mathcal{M}} \mathbf{b}_{ijk}}_{\mathbf{b}}\,. \tag{7}$$

The linearized solution is then obtained by adding to the initial guess the computed increments

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} + \Delta\tilde{\mathbf{x}}. \tag{8}$$

Note, that $\hat{\mathbf{x}}$ and $\mathbf{x}$ now update all parameter in one step.

In order to interpret this approach as a graph optimization, Fig. 2 (right) illustrates the functions and quantities that play a role in defining an edge of the graph. Cameras can be interpreted as nodes in a pose graph, which are connected via FM measurements to each other. Fig. 4 (left) shows an initial pose graph consisting out of three cameras and four fiducial markers. The optimization approach requires a marginalized graph as depicted in Fig. 4 (right). It is shown, that lose edges (e.g. FM 4 was measured by only by camera 2) are removed and multiple edges are expanded (e.g. FM 2 was measured by camera 1, 2, 3). The marginalized graph can then be used to build up the linear system in Eq. 7.

IV. ARCHITECTURE OVERVIEW

The modular and distributed system architecture of the *CITrack* is shown in Fig. 5 and consists out of multiple open-source contributions: The physical *CITrack*, its simulation model[2] and its tools consisting out of grabber, tracker, localization, and calibration tools.

To be compliant with *Robot Operating System* (ROS), all applications are available as ROS-packages. Further, all simulation models are available for the Gazebo simulation.

A. CITrack

The *CITrack* comprises a main experiment area of $6\,\mathrm{m} \times 6\,\mathrm{m} \times 1.5\,\mathrm{m}$ that is rectified by five cameras as depicted in Fig. 6 (left). The operative hight of $1.5\,\mathrm{m}$ is explained by the cameras' overlapping fields of view, such that a $10\,\mathrm{cm} \times 10\,\mathrm{cm}$ fiducial marker (FM) does never go out of sight. The experiment area can also be partitioned into four sub-fields running up to four independent experiments in parallel[3]. Robots and objects are attached with FM for position and orientation detection as well as for identification. Four SP-5000M-GE2 grayscale cameras with $8\,\mathrm{mm}$

[2] https://github.com/cognitiveinteractiontracking

[3] VR experience: https://youtu.be/ezJA2EgBLyk

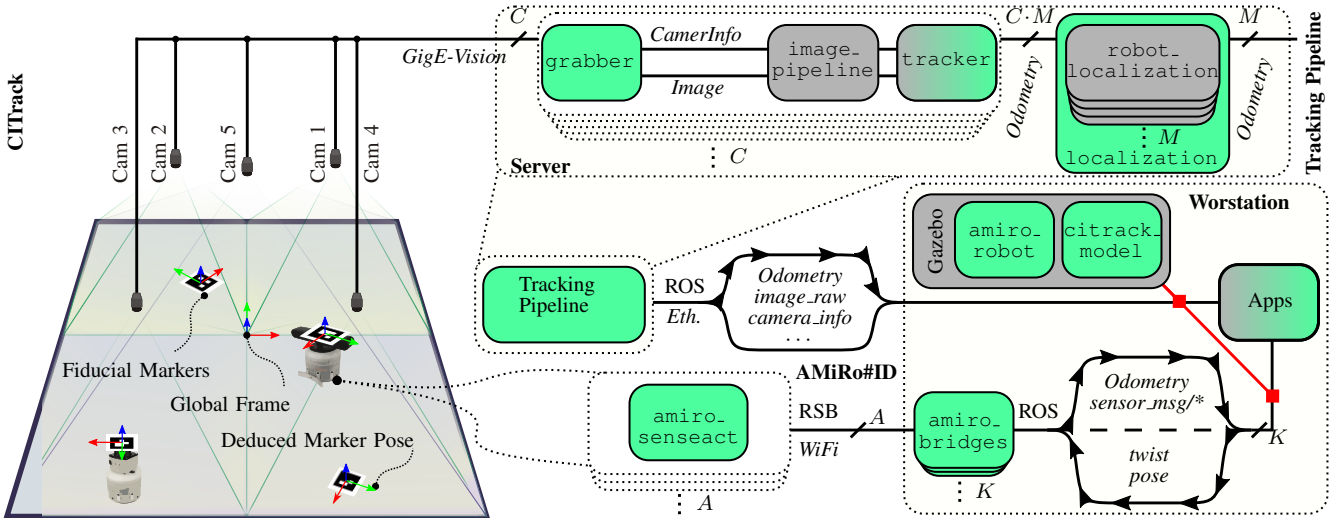

Fig. 5: Architecture overview of the *CITrack* environment. Left: Physical *CITrack* overhanged with $C = 5$ cameras observing the area. Top-Right: Tracking pipeline as applied in the experiments with $C = 5$ FM tracker and $M = \#\text{Marker}$ Kalman filter for each FM ID. Bottom-Right: RSB interfaces of $A = 2$ AMiRo [20] is advertised in the university network which assigns domain names by MAC addresses. A workstation PC, running *ROS* applications (Apps), allocating $K \leq A$ robots via `amiro_bridges` that advertise ROS compliant sensor messages and control interfaces. The whole physical setup can be substituted by the Gazebo simulator and the provided models. Worth mentioning, multiple workstations can run the setup in parallel and all ROS topics are automatically namespaced by the robots domain name. Major open-source contributions are highlighted in green, minor contributions and implementations of third parties in partial green, and third party implementation necessary for the setup in gray. Transport types are written in *italic* and package names in `teletype`.

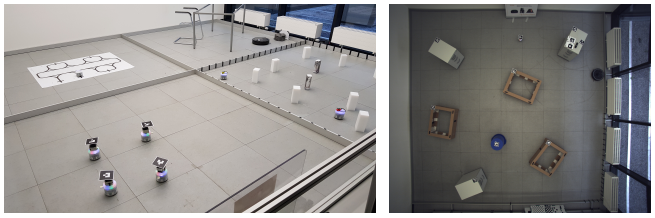

Fig. 6: Exemplary setup of the *CITrack*: four different experiments running in parallel (left) and birds-eye view from color camera with labeled objects and robot.

lenses and one SP-5000C-GE2 color camera with 6 mm lens, with a resolution of 2560×2048 pixels each, are mounted above the experiment area. Each camera is connected via *Ethernet* to the university network and is grabbed via *GigE-Vision* by a common server running *Ubuntu 16.04* and ROS Kinetic. Furthermore, all computer based systems are synchronized via *Network Time Protocol* (NTP) while the cameras are synchronized via *Precision Time Protocol* (PTP) and synchronously hardware triggered to achieve exact time stamping which is crucial for any later fusion. The server also runs the *multimaster_fkie*[21] to advertise ROS communication in the network. Thus, experiments and recordings can be conducted by any common PC in the network.

B. CITrack Tools

Three different calibration tools for setting the extrinsic parameters of the cameras in the *CITrack* are available: `tf_dyn` for manual online calibration, `oneshot_calib` which averages poses of a single static FM, and `graph_calib` that realizes the approach from Section III.

Once the camera system is calibrated, localization is performed as follows: Images of each camera are grabbed and processed separately to provide the IDs and poses of all detected FM in the current frames. To be fully ROS compliant and to make use of the `image_pipeline`-implementation the camera drivers are written, such that they provide the undistorted raw camera image via the `#camera/image_raw`-topic, and all corresponding information via the `#camera/camera_info`-topic. Each camera frame is then processed by a FM tracker to provide the ID and pose of every detected marker via odometry messages on a single `#camera/odom/#ID`-topic. Currently, *ArUco2/3*, *ArToolKit5*, *AprilTag*, and *ICL* are implemented. Further, the Kalman filter provided by the `robot_localization`-package from Moore and Stouch [22] is applied to fuse odometry of equal IDs from different cameras.

V. EXPERIMENTAL SETUP & EVALUATION

In this section, various setups are evaluated. First, the different fiducial marker (FM) tracker *ArToolkit*, *ArUco*, and *AprilTag* are analyzed regarding their accuracies in Section V-A. Further, the FM based calibration error and convergence speed of our proposed approach is evaluated in Section V-B. At last, the FM based calibration and the end-to-end error of the tracking pipeline, as shown in Fig. 5, is evaluated against a *Vicon* tracking system, consisting out of

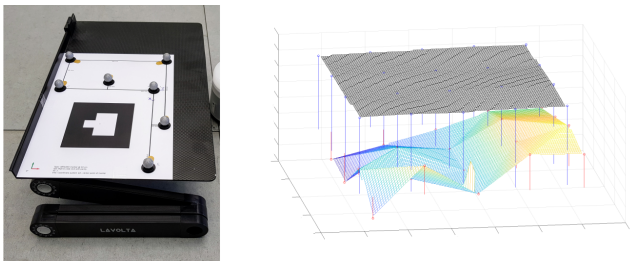

Fig. 7: Combined FM-Vicon Marker (left) and qualitative evaluation (right) of various measurements of the FM-Vicon marker on the plane floor of the *CITrack* laboratory for a single camera. Black surface is measured by Vicon while the other is measured by *ArUco*.

	ArUco	*ICL*	*AprilTag*	*ARToolKit*
RMSE (m)	.009162	.143602	.018296	.042388
μCOS (1)	.000186	.020259	.009214	0.009622

TABLE I: Root Mean Squared Error (RMSE) and mean cosine similarity (μCOS) for chosen FM tracker. RMSE is the remaining error of fitting a plane into the performed measurements, while μCOS is calculated wrt. the reference frame parallel to the plane.

eight *MX T20* which are interfaced by the *Vicon Nexus 1.8.4* software, in Section V-B.

A. FM Error Evaluation

To evaluate which FM tracking system is sufficient for the calibration task, we measure a plane surface under the camera by passing around a combined FM-Vicon marker on the floor (c.f. Fig. 7). Figure 7 qualitatively reveals the discrepancy between a straight plane measured by Vicon versus a FM tracker. While the planes are not perfectly aligned, due to naive extrinsic calibration, the FM tracker shows comparable high noise in measuring the plane.

However, fitting the planes into each other, to assume a perfect extrinsic calibration, results in quantitative evaluation for all FM tracker. *ArUco* outperforms all other FM based systems by at least one magnitude in measuring a straight plane as shown in Table I. All FM tracker were applied per frame over at least 250 measurements to avoid artifacts by any tracker based filtering. However, while no FM tracker system performs perfectly, we stick to *ArUco* for further experiments.

B. FM Based Calibration Evaluation

We use the combined marker from V-A and perform a random trajectory captured by all four cameras of the *CITrack* as depicted in Fig. 8. Due to the labeled marker and hardware-trigger based synchronization, we achieve a perfect association between all measurements. Furthermore, we can directly setup any *Sparse Bundle Adjustment* (SBA) toolbox for camera network calibration, since all FM tracker provide pixel and pose location. Thus, we use Matlab2018a `bundleAdjustment` as our baseline.

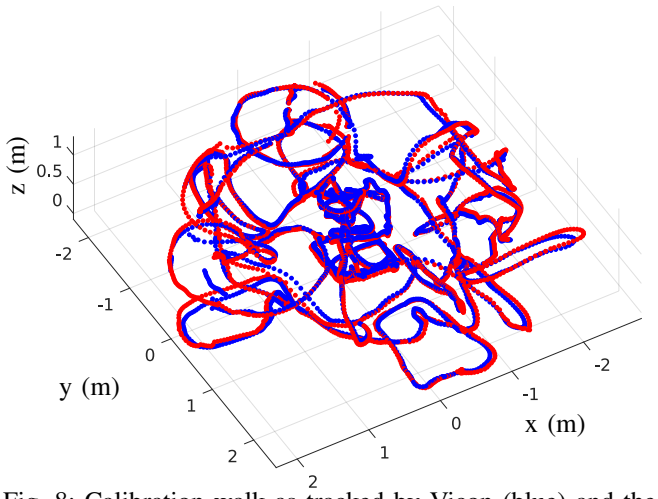

Fig. 8: Calibration walk as tracked by Vicon (blue) and the final calibrated *CITrack* in an operative hight between 0 m and ∼1 m wrt. the floor. The two point clouds do overlap sufficiently well with little disturbances at the *CITrack* boarders (e.g. (x,y)=(1,-3) m).

We apply our proposed approach as follows: First, we initialize the camera positions using `tf_dyn` (c.f. Sec. IV-B). Second, we associate all FM detections to a pairwise detection as depicted in Fig. 2 and 4. Third, we build up the linear equation systems for R^3 (i.e. common graph SLAM w/o measuring orientations) and the proposed SE(3). We solve Eq. 7 using Levenberg–Marquardt algorithm and apply loss specific damping parameters λ for translation (λ_t) and rotation (λ_q) error. We found that constant $\lambda_\mathrm{t} = 10^{-3}$ and $\lambda_\mathrm{q} = 10^{-1}$ over 1000 iteration performed sufficiently well in all experiments, and that parameter change during optimization was not necessary. Since FM measurements of our calibration walk (c.f. Fig. 8) are not equally distributed, biased calibration due to possible systematical errors of the FM tracker is possible. Therefore, we introduce a k-means++ inspired *refinement step* (ref.), where we sample a new data set for calibration from the old one, by assigning a sampling weight to every measurement that is the reciprocal sum of distances to all adjacent measurements. Finally, the progress and end results are depicted in Fig. 9 which reveal that the proposed approach with refinement performs best.

C. FM End-to-End Evaluation

The end-to-end error is evaluated on the calibration walk as depicted in Fig. 8. We applied the *CITrack* as shown in Fig 5 and recorded the Kalman filtered pose of the FM trackers for the calibration marker. With the known temporal association between the FM tracking and the Vicon system we are able to evaluate the exact error of our approach wrt. the Vicon system. Table II reveals that our proposed calibration approach even outperforms other solutions in an end-to-end evaluation and is therefore the technique of choice for FM based camera calibration.

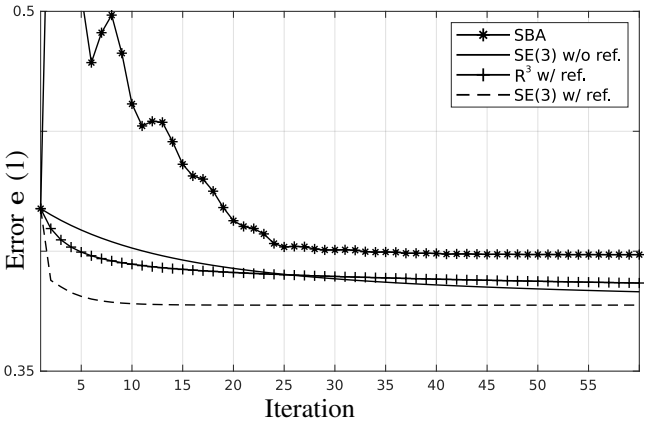

Fig. 9: Calibration error $\mathbf{e}$, as defined in Sec. III-B, over number of iterations. The first 60 episodes show that the proposed approach on SE(3) with refinement of the measurements outperforms the standard Matlab `bundleAdjustment` (SBA), calibrations in R^3 and SE(3) without refinement of the measurements respectively. Final mean errors $\mathbf{e}$ after 1000 iterations are: SBA: .398, SE(3) w/o ref.: .380, R^3 w/ ref.: .384, SE(3) w/ ref.: .377. While all curves start with the same error, we assume that SBA starts with low damping factors λ, which are annealed over the first iterations, causing the increasing error.

	SBA	SE(3) w/o	R^3 w/	SE(3) w/
RMSE	.06739	.04991	.0536	**.04692**
μCOS	.00157	.00134	.00136	**.00128**

TABLE II: End-to-end error of the calibration walk. w/ and w/o refer to the refinement (ref.)

VI. CONCLUSIONS AND FUTURE WORK

This publication presents a novel graph-based multi-camera calibration based via fiducial marker tracking and evaluates the tracking performance in an end-to-end approach against a Vicon system. Our approach gives anyone the tools to build a vision and fiducial marker based tracking benchmark system with the introduced, sufficient quality. The final calibrated *CITrack* system allows us to perform crucial upcoming tasks which are necessary to induce robotic benchmarking (c.f. application video[4]): multi-robot tracking, real-life data-annotation, and model-identification. With human-robot interaction in mind, the next effort of extending the presented *CITrack* is to add more tracking modalities, which can be seamless calibrated using our proposed approach, as long as they provide measurements in R^3 or SE(3).

ACKNOWLEDGMENT

This research was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University and by the Federal Ministry of Education and Research under grant number 57388272. The responsibility for the content of this publication lies with the author.

[4] `https://youtu.be/obG8V_426zE`

REFERENCES

[1] A. Jiménez-González, J. R. Martinez-De Dios, and A. Ollero, "Testbeds for ubiquitous robotics: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1487–1501, 2013.

[2] M. J. Quinn, R. Mudumbai, T. Kuo, Z. Ni, C. D. Leo, and B. S. Manjunath, "VISNET: A distributed vision testbed," *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1–8, 2008.

[3] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, "Mobile Emulab: A Robotic Wireless and Sensor Network Testbed," in *Proceedings IEEE INFOCOM 2006*, pp. 1–12.

[4] A. Howard, "Mezzanine User Manual," Tech. Rep., 2002.

[5] P. De, A. Raniwala, R. Krishnan, K. Tatavarthi, J. Modi, N. A. Syed, S. Sharma, and T.-c. Chiueh, "MiNT-m: An Autonomous Mobile Wireless Experimentation Platform," *Proceedings of the 4th international conference on Mobile systems, applications and services - MobiSys 2006*, p. 124, 2006.

[6] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso, "SSL-Vision: The shared vision system for the RoboCup Small Size League," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5949 LNAI, 2010, pp. 425–436.

[7] A. Tanoto, H. Li, U. Ruckert, and J. Sitte, "Scalable and flexible vision-based multi-robot tracking system," *2012 IEEE International Symposium on Intelligent Control*, pp. 19–24, oct 2012.

[8] N. Michael, J. Fink, and V. Kumar, "Experimental Testbed for Large Multirobot Teams: Verification and Validation," *IEEE Robotics and Automation Magazine*, vol. 15, no. 1, pp. 53–61, 2008.

[9] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The Robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1699–1706.

[10] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[11] P. Lightbody, T. Krajník, and M. Hanheide, "A versatile high-performance visual fiducial marker detection system with scalable identity encoding," in *Proceedings of the Symposium on Applied Computing - SAC '17*, 2017, pp. 276–282.

[12] H. Aghajan and A. Cavallaro, *Multi-Camera Networks principles and applications*. Elsevier Inc., 2009.

[13] E. Yeguas-Bolivar and R. Medina-Carnicer, "Mapping and Localization from Planar Markers," no. October, 2017.

[14] B. Morrison Richard, "Fiducial marker detection and pose estimation from LIDAR range data," Ph.D. dissertation, 2010.

[15] J. Rekimoto and Y. Ayatsuka, "CyberCode: designing augmented reality environments with visual tags," *Science*, vol. Vol 303, no. 9, pp. 1–10, 2000.

[16] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top AR environment," in *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, 2000, pp. 111–119.

[17] C. Elbrechter, R. Haschke, and H. Ritter, "Bi-manual robotic paper manipulation based on real-time marker tracking and physical modelling," in *IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 1427–1432.

[18] M. Fiala, "Designing highly reliable fiducial markers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1317–1324, 2010.

[19] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," pp. 1–11, 2010.

[20] S. Herbrechtsmeier, T. Korthals, T. Schöpping, and U. Rückert, "AMiRo: A modular & customizable open-source mini robot platform," in *2016 20th International Conference on System Theory, Control and Computing, ICSTCC 2016 - Joint Conference of SINTES 20, SACCS 16, SIMSIS 20 - Proceedings*, 2016.

[21] A. Koubaa, *Robot Operating System (ROS): The Complete Reference*. Springer International Publishing, 2016, vol. 1, no. 1.

[22] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," *Advances in Intelligent Systems and Computing*, vol. 302, pp. 335–348, 2016.