Automatic Extrinsic Multi-Sensor Network Calibration based on Time Series Matching

Sonja Schuster*, Johannes Wetzel‡, Samuel Zeitvogel* and Astrid Laubenheimer*
*Intelligent Systems Research Group (ISRG)
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
sonja.schuster@h-ka.de, samuel.zeitvogel@h-ka.de, astrid.laubenheimer@h-ka.de
‡ Vitracom GmbH
Karlsruhe, Germany
jwetzel@vitracom.de

***Abstract*—We propose an automatic calibration approach to determine the extrinsic (inter-sensor) calibration of a multi-sensor network for people tracking. A plan-view approach is used and pairwise overlapping detection areas of the distributed sensors are assumed. By exploiting intra-sensor tracks of an unknown number of tracking targets, we solve the referencing problem of the sensor fields of view by a matching of time series, avoiding any manual effort for the extrinsic calibration. We realize the automatic calibration exclusively based on intra-sensor tracking information by combining a trackwise *w*-RANSAC with a rotation-invariant distance measure, and an effective pre-filter method based on the walking speed of topological and temporal matched track pairs. Our automatic calibration routine is evaluated on a multi-sensor network, consisting of five depth sensors with a top-down view on an indoor scene, in which five people are randomly walking for approximately one minute. The track mapping accuracy of our automatic calibration method is compared to a calibration based on a manual selection of homologous image points. Therefore, we propose an evaluation method regarding the global track mapping accuracy. By excluding known track matches of our dataset from the calibration process, we derive an assumption about the global tracking performance of the calibrated multi-sensor network.**

***Index Terms*—auto-calibration, self-calibration, multi-sensor fusion, depth sensor networks, people tracking, object tracking, multi-view, time series matching, RANSAC**

I. Introduction

Indoor people tracking has several applications, for example, customer behavior analysis, public security, or ambient assisted living. In order to track people's paths across large indoor areas, multiple top-down depth sensors can be combined into a multi-sensor network [1]. The fusion of the single local fields of view (FOVs) results in a common global (inter-sensor) FOV of the entire scene. The resulting global view can be used for global tracking by determining which tracks, detected by different sensors, are caused by the same person. Referencing the local detection areas with respect to a common coordinate system is a prerequisite of people tracking across multiple sensors and is addressed by the *global extrinsic calibration* of a multi-sensor network, in reference to [2]. The global extrinsic calibration results in transformations from each local coordinate system to a common global coordinate system.

In order to solve the referencing problem, correspondences between neighboring FOVs are required. A manual calibration routine could be used, e.g., by a prior placement of calibration objects in the scene, as done in [3] by using fiducial markers. Although the markers are automatically identified, we consider this kind of calibration as *manual* because it requires the placement of specific calibration objects in the scene. This manual task is not practicable in some cases, e.g., at a trade fair, where an immediate fitting and calibration of the sensors by untrained constructors is requested. In contrast, a *manual-selection* calibration approach can be realized by a manual selection of homologous image points between different FOVs, via a user interface. This has the advantage of not requiring any intervention in the scene, which makes remote usage, e.g., at a trade fair, possible. However, this comes with the disadvantage of the manual human effort, requiring scene understanding and provoking errors. To counteract these disadvantages, we focus on an *automatic* calibration for a multi-sensor network. The presented automatic calibration method is exclusively based on tracking information, requiring neither manual effort nor scene intervention.

As the main contribution, we propose a *trackwise w-RANSAC approach for automatic calibration of a multi-sensor network*. For this purpose, at first, we apply a local tracking routine for each individual sensor to gain information about the local connection of the detection points as local trajectories and take this trajectory information into account for track-pairing, pre-filtering, and subsequently in a random sample consensus (RANSAC) algorithm [4]. In order to find robust track correspondence, we propose a track-based RANSAC, closely related to [5]. Additionally, our trackwise RANSAC is augmented to a weighted (*w*-RANSAC) [6], [7] implementation by combining the algorithm with a rotation-invariant similarity measure for trajectories [8] in order to achieve a more efficient correspondence search. In summary, our approach is a modification of the pointwise RANSAC-based method for auto-calibration of a sensor network proposed by [2]. Accordingly, the point correspondence problem between neighboring sensors can be restated as a track segment correspondence problem between neighboring sensors [9]. The

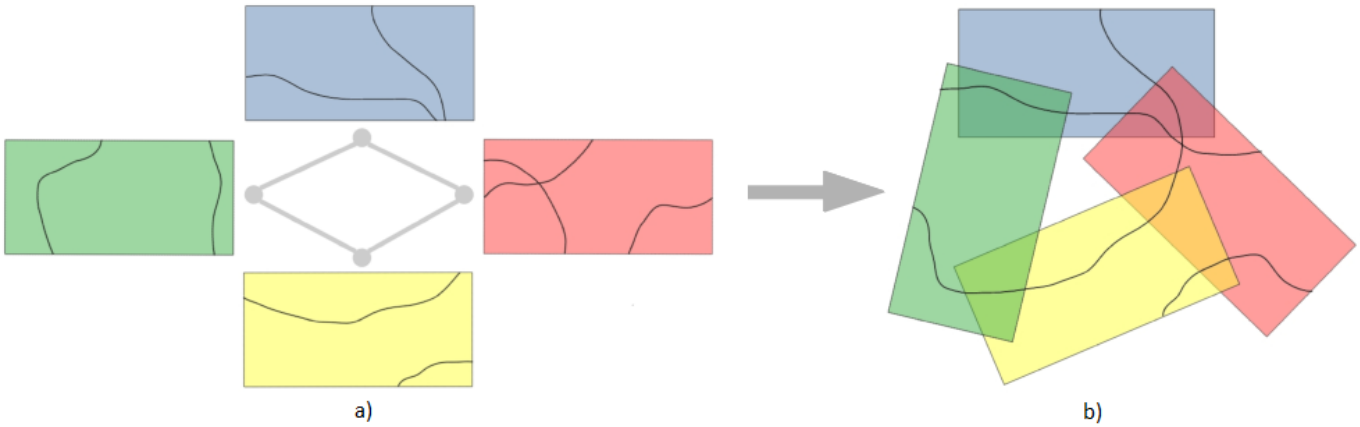

Fig. 1. Extrinsic calibration of a multi-sensor network by solving the referencing problem based on local tracks: a) Example of the given information of the referencing problem: The trajectories on the ground-plane are given by their time-dependent local 2D-coordinates, based on the related local detection area of a sensor. b) Example of the objective of the referencing problem: Relative arrangement of the detection areas to one another. The result provides the transformation from the local coordinate systems to the global (here: blue) coordinate system.

advantages of our automatic calibration method are outlined in section III-C.

The presented automatic calibration routine is evaluated on a depth sensor network with five sensors having a top-down view on an indoor scene. The accuracy of the presented automatic calibration approach is compared to a manual-selection calibration method. By considering an evaluation environment for which we have a full scene understanding, the result of the manual-selection calibration provides a competitive solution for the referencing problem. Consequently, the manual-selection calibration also provides knowledge about the correct correspondences of the local tracks in our dataset. Based on the knowledge of correct track correspondences, we also propose an evaluation method regarding the global track mapping accuracy.

II. Related Work

The calibration of camera networks in the context of people tracking is an extensively studied problem in the literature [10]. The vast majority of approaches leverage calibration objects such as visual markers. Another category of approaches focuses on auto- or self-calibration of camera networks. Pätzold et al. [11] employ an automated (marker-free) method for the extrinsic calibration of a multi-camera system. Therefore, an initial measurement of the camera system is required. The estimates are based on appearance-based corresponding key points of recorded people in the camera images. In contrast, in our work, we avoid the usage of appearance-based features with the benefit of a higher data protection and to become independent of the deployed sensors. Munaro et al. [12] propose a framework for people tracking with a network of RGB-D cameras including a calibration procedure. The calibration is based on an initial manual step, utilizing markers, followed by an automatic refinement step which leverages the people detections of the individual sensors. For tracking purpose in outdoor scenes, Stauffer et al. [9] propose a method for an automatic calibration of a camera network to determine a planar tracking correspondence model. Similar to our work, they use local (intra-sensor) tracks. Comparable to our approach, track correspondences are estimated. In contrast to our work, the topology of the cameras is not taken as given and overlapping areas between adjacent sensors are not generally assumed. Because we are considering indoor scenes, the topology of the sensors is considered as implicitly known through the sensor installation process. In [9], it is shown that the lack of overlapping areas is a limiting factor of their method.

Closely related to our work, Korkalo et al. [2] also consider the problem of automatic calibration of a depth-sensor network for people tracking. In addition to solving the referencing problem, [2] identify the topology of the sensor network and refine the camera intrinsics based on a first initialization. As in the present work, a plan-view approach is considered and a RANSAC-based method is used to determine 2D-transformations between the different FOVs. To cope with linear and non-linear depth measurement distortions, [2] utilize non-linear transformations, such as thin plate splines. They complete their calibration procedure with a global optimization routine. The RANSAC-based method in [2] directly relies on the single person detections and uses the temporal proximity and a spatial clustering of the person detections to determine point correspondences. Besides that, they use an additional scoring for validation. The main contrast in using a RANSAC approach for automatic calibration is that [2] use a point-based RANSAC method, while in our work, we rescale the approach to a track-based RANSAC in order to keep the information about the connection between the detection points of a track and use the time series as a whole for model estimation. A closely related trajectory-based sequence matching RANSAC algorithm is proposed by Caspi et al. [5] for feature matching between monocular video-sequences to derive fundamental matrix.

Our approach can also be seen as a modification of the automatic calibration approach by [9]. Equally to [9], we are searching for track correspondences, presupposing local tracks. Differences exist in the search for track correspondences. The authors in [9] and [2] suggest the use of further features. Therefore, our work examines the additional use of information included in the tracks, namely motion pattern and walking speed. Fig. 2 shows a condensed comparison between our calibration pipeline and [2].

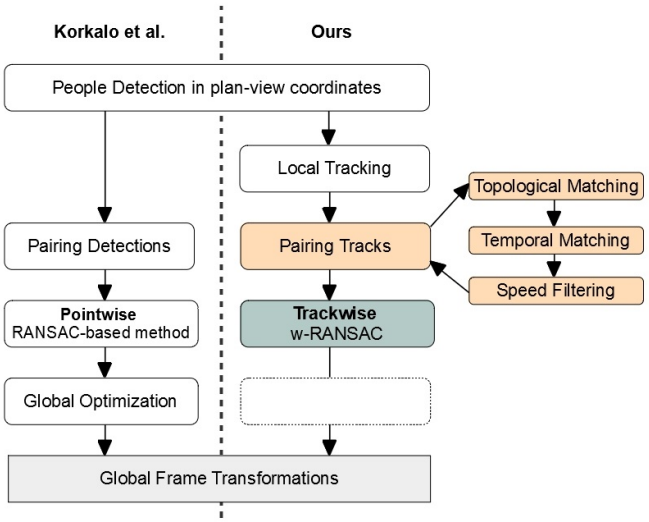

Fig. 2. Overview of our calibration pipeline in comparison to Korkalo et al. [2]: The overall goal is to determine point correspondences between neighboring sensors. Both approaches are applied on a depth sensor network for people detection. The 3D people detections are projected on the ground plane for a plan-view approach. While [2] directly pairs the 2D point detections, our approach firstly applies a local (intra-sensor) tracking routine and subsequently pairs the tracks, before our trackwise *w*-RANSAC method is used. Korkalo et al. conclude with a global optimization, which could be considered as an optional extension of our approach.

III. Approach

A. Assumptions

The presented approach for an automatic calibration is based on the assumption that the FOVs of neighboring sensors have a common overlap area. The resulting network topology, describing the neighborhood of the sensors, is also assumed as known. The topological information can be considered as implicitly known through the sensor installation process. Besides that, an almost top-down view of the sensors is presupposed. As in [2], a plan-view approach [13] is assumed to project the 3D-information of the people detections onto a 2D-ground plane and thereby reducing the parameter count in the system considerably. It is assumed that the sensor clocks are temporally synchronized.

B. Overview

The pipeline of our approach is depicted in Fig. 2. Based on the 2D people detections, the aim is to estimate 2D-transformations to solve the referencing problem. In this work we only consider rigid 2D-transformations. More flexible transformations are used in [2].

In order to solve the referencing problem, shown in Fig. I, we calibrate the sensors $\{S_A, S_B, \ldots\}$ of the network in pairs, according to their topology. In general, we consider two sensors as neighboring if their FOVs have a common overlap area. This can lead to an arbitrary sensor topology of the network. In this work, the sensor topology for the calibration process is restricted to the structure of a tree, considering one sensor as the root node in order to define the global coordinate system. Therefore, a manual edge pruning strategy is applied. Based on the plan-view assumption, between each topological connected sensor pair (S_A, S_B), a rigid 2D-transformation $\hat{T}_{A,B}$ is determined, transforming the local coordinate system of S_A to the local coordinate system of S_B. Considering a tree topology, the pairwise estimated transformations can be chained to transform each local coordinate system along the tree topology to the global coordinate system, defined by the root node sensor. To address a potential occurring error propagation and loop-closure problem, a global optimization routine could be applied, as it is done in [2].

A prerequisite of our proposed approach is the generation of local tracks. For example, the local tracks could be generated by applying the global tracking procedure, intended for the later use in the calibrated sensor network, also locally within each individual sensor view. For an unknown number of tracking targets, for each sensor S, a set $\{D_i^S\}$ of time series (tracks) $D_i^S = \{\boldsymbol{d}^S_{i,t_i^{\text{start}}}, ..., \boldsymbol{d}^S_{i,t_i^{\text{end}}}\}$, consisting of 2D people detections $\boldsymbol{d}^S_{i,t} \in \mathbb{R}^2$, is given. While the subscript i denotes the index of a local track related to the local coordinate frame of sensor S, the subscript $t \in [t_i^{\text{start}}, t_i^{\text{end}}]$ represents the timestamp of a detection point for a track with start timestamp t_i^{start} and final timestamp t_i^{end}. For each sensor pair (S_A, S_B), the input of our calibration routine is defined by two sets of local tracks $(\{D_i^A\}, \{D_j^B\})$ with $|\{D_i^A\}| = N_A$ and $|\{D_j^B\}| = N_B$. For a clear formal distinction between two local tracks from different sensors, in the following, each track captured by sensor S_A is indexed by i with $1 \leq i \leq N_A$, while each track recorded by S_B is indexed by j with $1 \leq j \leq N_B$.

For transformation estimation, the general goal is to determine robust point correspondences between all sensor pairs. Our approach determines the point correspondences by a matching of time series, through searching for corresponding track segments between each sensor pair. As a result, the problem can be restated into finding track correspondences. In detail, two track segments are a correct correspondence if and only if they are caused by the same person during the same period of time.

Given the local 2D tracks, at first, a pairing of the local tracks is performed (section III-D). For our track pairing step, all track segments, detected at the same time by two different neighboring sensors, are combined as track pairs. The matching based on the temporal overlap is highly ambiguous. In order to reduce the combinatorial number of possible track pairs, we pre-filter this track combinations by performing a

validation of the track pairs via their average walking speed deviation during their temporal overlap.

The remaining track pairs are used as input for a track-wise RANSAC algorithm (section III-E) to find robust track correspondences. To exploit the pattern information of the track segments in the overlapping sensor areas, in particular, a *weighted* RANSAC (*w*-RANSAC) approach [6] is applied (section III-G) in combination with a rotation-invariant distance measure (section III-F) to find robust correspondences in fewer RANSAC iterations. To this end, we transform the positional information, based on the Cartesian trajectory representation, into the angle/arc-length space [8] for the rotation-, and translation-invariant distance measure. The pattern information is intentionally only used for a more efficient matching, instead of being used as a strong matching criteria. This is reasoned by the fact that the overlapping areas between the sensors can become very small and consequently the pattern information of the tracks in this area is less discriminative. Considering a small overlapping area, the trajectory pattern of the people tracks in the overlap will be nearly straight. A general use of the trajectory pattern as a strong matching criteria would probably be detrimental, especially because the pattern is not invariant to inaccurate person detections. Slightly inaccurate people detections might be relatively strong penalized when the remaining pattern information is low, probably leading to a higher amount of wrong matched track correspondences.

In summary, we realize the automatic calibration only based on tracking information by constituting a *trackwise w-RANSAC* with a rotation-invariant distance measure and a pre-filtering of topological and temporal matched track pairs.

C. Advantages

1) Advantages of using people detection data:

- Enabling an **automated and remotable calibration** of the sensor network, without the need of an intervention in the scene or any human effort of scene understanding.
- The usage of people detections, instead of features close to the ground, can make it possible to **compensate systematic inaccuracies**, which might be contained in the detections. For example, systematic inaccuracies can occur due to the different perspectives of the sensors, inaccuracies in the calculation of disparities of stereo vision, and weak intrinsic camera calibrations.

2) Advantages of avoiding appearance-based features:

- **Data protection**, e.g., by no necessity to transmit appearance information to a central computing unit, given a distributed local tracking.
- **Independence of the deployed sensors** by not using the image data directly.
- **Independence of the scene** because no exploitable features are necessary.
- **Generalizability** to any moving objects.

3) Advantages of considering track correspondences, instead of disassociated point correspondences:

- **Reduction of the matching possibilities** robustifies the search for correspondences.
- Emerging option to **exploit additional track-associated information** as matching-criteria in order to find corresponding tracks, e.g., the walking speed and the track pattern, both leveraged in our approach.

In summary, by using the tracks to solve the referencing problem, the goal of the referencing problem becomes the same as for the pursued cross-sensor tracking purpose: searching for the best mapping of tracks of the same person, detected by different sensors. Thus, the calibration routine takes the goal of the cross-sensor tracking directly into account.

D. Pairing Tracks

The pairing of the tracks starts by pairing the sensors based on a tree topology, pre-set by a manual edge pruning. Then an initial combinatorial matching of all possibly corresponding tracks between each sensor pair is performed. At this level, all temporally overlapping tracks between each sensor pair (S_A, S_B) are considered as possible track pairs. Pairing two tracks D_i^A and D_j^B based on their temporal overlap:

$$(t_{\text{start}}, t_{\text{end}}) = \left(\max(t_i^{\text{start}}, t_j^{\text{start}}), \min(t_i^{\text{end}}, t_j^{\text{end}})\right) \quad (1)$$

results in a temporal *track-intersection pair* (P_i^A, P_j^B) if and only if $t_{\text{start}} < t_{\text{end}}$ with:

$$\{\boldsymbol{d}_{i,t}^A \in D_i^A | t_{\text{start}} \leq t \leq t_{\text{end}}\} \rightarrow P_i^A = \{\boldsymbol{p}_{i,1}^A, .., \boldsymbol{p}_{i,M}^A\} \quad (2)$$

$$\{\boldsymbol{d}_{j,t}^B \in D_j^B | t_{\text{start}} \leq t \leq t_{\text{end}}\} \rightarrow P_j^B = \{\boldsymbol{p}_{j,1}^B, .., \boldsymbol{p}_{j,M}^B\} \quad (3)$$

and $\boldsymbol{p}_{i,m}^A, \boldsymbol{p}_{j,m}^B \in \mathbb{R}^2$. A linear spline interpolation between the detection points of each track and a subsequently equidistant subsampling by a time step s guarantees that $\boldsymbol{p}_{i,m}^A$ and $\boldsymbol{p}_{j,m}^B$ refer to two track-intersection points at the same point in time, represented by their subsampling index $m \in [1..M]$ with M sampled points of each intersecting track segment. We map the timestamps t to the subsampling indices m as following: $t_{\text{start}} \mapsto 1$ and $t'_{\text{end}} \mapsto M$ with $t'_{\text{end}} = t_{\text{start}} + sh \leq t_{\text{end}}$ and $h \in \mathbb{N}$. This track-intersection pairing is repeated for all track pairs (D_i^A, D_j^B) during their common temporal overlap.

To counteract the many ambiguities caused by the temporal matching, in the first place, a speed filter is applied. This pre-filter includes a heuristic to exclude pairs of intersections (P_i^A, P_j^B) that cannot be caused by the same person, using the walking speed as track-associated information. Here the *average walking speed deviation* of a track-intersection pair (P_i^A, P_j^B) is denoted by:

$$\Delta(P_i^A, P_j^B) = \frac{1}{M} \sum_{m=1}^{M} \left| \left\|\boldsymbol{u}_{i,m}^A\right\|_2 - \left\|\boldsymbol{u}_{j,m}^B\right\|_2 \right|, \quad (4)$$

where $\boldsymbol{u}_{i,m}^A, \boldsymbol{u}_{j,m}^B \in \mathbb{R}^2$ denote the velocities corresponding to $\boldsymbol{p}_{i,m}^A$, respectively $\boldsymbol{p}_{j,m}^B$, and are calculated by exponential smoothing [14]. A threshold λ_{speed} is used for the tolerable average difference in walking speeds between two intersecting track segments, which sorts out track-intersection pairs with significant different speeds, resulting for each sensor pair (S_A, S_B) in a pre-filtered set of *presumably corresponding track-intersection pairs* $C^{A,B}$ with:

$$C^{A,B} = \left\{ \left(P_i^A,\ P_j^B\right) \mid \Delta(P_i^A, P_j^B) < \lambda_{\text{speed}} \right\}. \quad (5)$$

E. Trackwise RANSAC

Similar to Korkalo et al. [2], we propose to use RANSAC to determine robust correspondences between sensor pairs for an automatic calibration of a sensor network. In contrast to [2], in our work the local track-information is leveraged for finding correspondences between two sensor views. Therefore, we use a track-based RANSAC, described below. It can be considered as modification of the trajectory-matching algorithms by [5] and [9]. The input of our trackwise RANSAC is a set of pairs of time series. In our case, for each sensor pair (S_A, S_B), the input is the set of presumably corresponding track-intersection pairs $C^{A,B}$ after the pre-filtering step.

In the sense of the general RANSAC algorithm [4], in each iteration, the exact number of corresponding points required to estimate the model are randomly selected. In the present case of rigid 2D-transformations, only two 2D-point correspondences are required for calculation. Therefore, in the algorithm presented here, in each RANSAC iteration, two track-intersection pairs $(P_i^A,\ P_j^B)$ are randomly selected out of the input set of the pre-filtered track correspondences $C^{A,B}$. Then out of each of these two selected track-intersection pairs, a corresponding point pair $(\boldsymbol{p}_{i,m}^A,\ \boldsymbol{p}_{j,m}^B)$ is randomly selected. This ensures that two point pairs out of two different track pairs are considered for the rigid 2D transformation estimation $T_{A,B}^*$ in each RANSAC iteration. Taking the two required points out of two different track pairs comes with the advantage of a probabilistically broader spatial distribution of the chosen point pairs. To additionally allow a scaling through the 2D-transformation between the tracks of different FOVS, e.g., if the sensors are placed at different heights, three point pairs out of three different track pairs could be randomly selected in each iteration.

Maintaining the approach based on track pairs, the resulting consensus set of each iteration also consists of a set of pairs of time series. To determine the consensus set of a RANSAC iteration, for each track-intersection pair (P_i^A, P_j^B) out of the RANSAC input set $C^{A,B}$, it is determined, whether the track-intersection pair supports the current estimated transformation $T_{A,B}^*$ of the current RANSAC iteration by calculating a *track transformation error*:

$$\mathrm{J_t}(T_{A,B}^*, (P_i^A, P_j^B)) = \frac{1}{M}\sum_{m=1}^{M} \left\| \boldsymbol{p}_{j,m}^B - T_{A,B}^*\ \boldsymbol{p}_{i,m}^A \right\|_2 \quad (6)$$

A threshold λ_{RANSAC} is used to decide, if a track pair supports the current model of a RANSAC iteration. If $\mathrm{J_t}(T_{A,B}^*, (P_i^A, P_j^B)) < \lambda_{\text{RANSAC}}$, then (P_i^A, P_j^B) is added to the current consensus set.

After a pre-defined RANSAC iteration count K, the resulting *maximum consensus set* $C_{\max}^{A,B}$, consisting of multiple track-intersection pairs (P_i^A, P_j^B), is used in a least median of squares:

$$T'_{A,B} = \underset{T_{A,B}}{\operatorname{argmin}} \underset{\substack{(P_i^A,P_j^B)\in C_{\max}^{A,B} \\ \forall m\in[1..M]}}{\operatorname{Median}} \left\| \boldsymbol{p}_{j,m}^B - T_{A,B}\ \boldsymbol{p}_{i,m}^A \right\|_2 \quad (7)$$

All corresponding point pairs $(\boldsymbol{p}_{i,m}^A, \boldsymbol{p}_{j,m}^B)$ included in $C_{\max}^{A,B}$ are used to estimate the point-based inliers:

$$G^{A,B} = \{(\boldsymbol{p}_{i,m}^A, \boldsymbol{p}_{j,m}^B) \in C_{max} \left\| \boldsymbol{p}_{j,m}^B - T'_{A,B}\ \boldsymbol{p}_{i,m}^A \right\|_2 < \lambda_{\text{LM}}\} \quad (8)$$

The final $\hat{T}_{A,B}$ transformation is estimated by an iterative optimization:

$$\hat{T}_{A,B} = \underset{T_{A,B}}{\operatorname{argmin}} \sum_{(\boldsymbol{p}_{i,m}^A, \boldsymbol{p}_{j,m}^B)\in G^{A,B}} \sum_{m=1}^{M} \left\| \boldsymbol{p}_{j,m}^B - T_{A,B}\ \boldsymbol{p}_{i,m}^A \right\|_1 \quad (9)$$

The described robust transformation estimation in (7)-(9) is based on, and solved by the implementation provided by [15].

F. Rotation-invariant Similarity Measure for Trajectories

In the present work, the pattern of the track-intersections, during the temporal overlap between two sensors, is used to find more reliable track correspondences. Therefore, we propose to use a rotation-, and translation-invariant similarity measure for trajectories because of the unknown relative arrangement of the local coordinate systems. A rotation-invariant distance measure for trajectories by Vlachos et al. [8], referred to as *exact angle* approach, is briefly described below.

The basic idea is to transform the trajectories into an *angle/arc-length (AAL) space* to initially achieve translation invariance. Rotation invariance is established using an iterative normalization routine. Scale-invariance can also be enabled.

Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M\}$ be an equidistant sampled 2D time series with $\boldsymbol{p}_m \in \mathbb{R}^2$. First, a so-called *movement vector* $\boldsymbol{v_m} = (v_{m,x}, v_{m,y})^T$, is defined by:

$$\boldsymbol{v}_m = \boldsymbol{p}_m - \boldsymbol{p}_{m-1} \quad (10)$$

In the *exact angle* approach, a *reference vector* is defined, corresponding to the positive x-axis:

$$\boldsymbol{v}_{\text{ref}} = (v_{\text{ref},x}, v_{\text{ref},y})^T = (1, 0)^T \quad (11)$$

Next, the angle α_m between the movement vector and the reference vector is calculated by:

$$\alpha_m = \operatorname{sign}(v_{m,x}v_{\text{ref},y} - v_{m,y}v_{\text{ref},x}) \arccos \frac{\boldsymbol{v}_m^T \boldsymbol{v}_{\text{ref}}}{\|\boldsymbol{v}_m\|\ \|\boldsymbol{v}_{\text{ref}}\|} \quad (12)$$

In order to achieve rotation invariance within this space, an iterative normalization is proposed in [8]:

$$\alpha_m^{l+1} = \alpha_m^l - \frac{1}{M}\sum_{m=1}^{M} \alpha_m \quad (13)$$

By this normalization routine, the average angle of the trajectory is iteratively subtracted from all α_m values. For better intuition, one can imagine this as the rotation, which affects the entire trajectory related to the reference vector, being removed. This leaves only the relative rotations of the movement vectors to each other. In order to avoid an oscillation, the angle values are wrapped to $[-\pi, \pi]$ within the normalization routine [8]. The resulting angle α_m represents the first of two coordinates within the AAL space. In order to completely transform the

movement vector into the new space, the arc-length of $\boldsymbol{v}_m$ has to be determined by its Euclidean length $\|\boldsymbol{v}_m\|$. Thus, the transformation of the movement vector from its original local coordinates $\boldsymbol{v}_m = (v_{m,x}, v_{m,y})^T$ into the AAL space $\boldsymbol{v}_{m,\mathrm{AAL}} = (\alpha_m, \|\boldsymbol{v}_m\|)$ is completed. Scale-invariance is additionally enabled by expressing the arc-length as a proportion of the total length of the track-intersection [8]. Done for each $\boldsymbol{v}_m$, this results in an AAL representation of the entire trajectory.

In contrast to [8], the distance calculation using *Dynamic-Time-Warping* is knowingly omitted in this work. In the present case, the temporal simultaneity of the tracks is explicitly used as a strong matching criterion between two point correspondences. Flexibility along the time axis, respectively along the arc-length, would contradict the matching assumption of simultaneous detections. In our case, the time series are already grouped in corresponding track-intersection pairs (P_i^A, P_j^B), wherein P_i^A and P_j^B consist of the same number of equidistant sampled points $\boldsymbol{p}_m$. Therefore, the arc-length can be rejected because only the difference between the corresponding angles $\{(\alpha_{i,m}^A,\ \alpha_{j,m}^B)\}$ are expressive for the rotation-invariant pattern comparison. The pattern distance between a track-intersection pair (P_i^A, P_j^B) is calculated by the average distance of their *angle-feature vectors* $(\alpha_{i,1}^S, \ldots, \alpha_{i,M}^S)$ by:

$$\mathrm{L}(P_i^A, P_j^B) = \frac{1}{M} \sum_{m=1}^{M} \left\| \alpha_{i,m}^A - \alpha_{j,m}^B \right\|_2 \tag{14}$$

G. Using the Similarity Measure in a weighted RANSAC

We use a rotation-invariant similarity measure between paired track-intersections to weight the trackwise RANSAC in order to obtain a combination of the *w*-RANSAC approach [6], [7] and a rotation-invariant similarity measure for time series. The idea behind this approach is to use the pattern information of the track-intersections, within the overlapping areas of the sensors, for a more efficient correspondence search. If two tracks have a similar pattern during their overlapping time period, then the probability, that both tracks were caused by the same person, is increased. The track-intersection pairs with low pattern similarities, i.e., with a high distance in the AAL space, are less weighted in the *w*-RANSAC algorithm. This makes it less likely that within a *w*-RANSAC iteration a track-intersection pair with poor pattern similarity will be chosen.

One possibility for such weighting is presented in the following. We construct a partitioned probability distribution Ψ for the *w*-RANSAC based on a weighting function proposed in [6]:

$$\phi(P_i^A, P_j^B) = \mathrm{e}^{-L(P_i^A, P_j^B)^2} \tag{15}$$

For each sensor pair, the set of all track-intersection pairs $C^{A,B}$ is divided by a threshold λ_L into two disjoint subsets, based on their pattern distance, resulting in a partitioned probability distribution $\Psi : C^{A,B} \longrightarrow [0, 1]$ with:

$$\Psi((P_i^A, P_j^B) \in C^{A,B}) \propto \begin{cases} \phi(P_i^A, P_j^B) & \text{if } L(P_i^A, P_j^B) > \lambda_\mathrm{L} \\ \omega\phi_{max} & \text{else} \end{cases} \tag{16}$$

and:

$$\phi_{max} = \max_{(P_i^A, P_j^B) \in C_{A,B}} \phi(P_i^A, P_j^B) \tag{17}$$

The parameter $\omega \geq 1$ specifies the relation between the two distribution partitions, resulting for $L(P_i^A, P_j^B) \leq \lambda_\mathrm{L}$ in an uniformly distributed partition of the overall distribution Ψ. In this way ω expresses, how reliable the classification of the track-intersection pairs to the two weighting functions is, based on the threshold value λ_L.

The intention for the partitioned probability distribution is caused by the fact that for short track intersections, the pattern information, within this intersections, will be minimal and therefore, resulting in high similarities also for wrong track correspondences. To prevent correct correspondences with slight inaccuracies from being less weighted than wrong correspondences with short intersections, the distance-based weighting starts beyond a specific pattern distance λ_L.

IV. Evaluation

A. Evaluation Setting

We evaluated our automatic calibration routine on real tracking data of an indoor setting of five depth sensors with top-down view at the same height. Our network, consisting out of five sensors, results in four concatenated sensor pairs with pairwise overlapping FOVs. The sensors are temporally synchronized using the Network Time Protocol [16]. Since the sensors have a frame rate of 30 frames per second, the maximum phase shift error between the sensors amounts to $\frac{1}{60}$s. During the recording for approximately one minute, five people walk through the detection area of the sensor network. The resulting local sensor tracking data is used as input of our automatic calibration approach.

B. Manual-Selection Calibration Routine for Comparison

We use a *manual-selection* calibration routine for comparison to our proposed *automatic* calibration approach. For each sensor pair, a user manually selects circa seven homologous image points of the floor via a user interface, displaying the 2D camera images of the overlapping FOVs. Considering an evaluation environment for which we have a full scene understanding, the result of the manual-selection calibration provides a competitive solution for the referencing problem and it also provides knowledge about the *certain correct track correspondences* (CCC) of the local tracks in our dataset, which we verified visually.

C. Evaluation Method

To compare the 2D-transformations estimated by our automatic calibration with the 2D-transformations resulting from the manual-selection calibration, we use a *mean transformation error* (19) based on the tracking data. By using the tracking data for the evaluation of the calibration procedure, we make an assumption about the performance of our automatic calibration routine for the global tracking purpose. An improvement of the global tracking would mean that tracks transitioning between two sensors break down less often,

which can be expected by a better mapping of the track-intersection pairs in the overlapping areas, represented by the *track* transformation error $\mathrm{J_t}$ in (6).

To ensure that the tracks which we use to calculate our *mean* transformation error do not contribute to the transformation estimation, we split the set $\mathrm{CCC}^{A,B} \subseteq \{(P_i^A, P_j^B)\}$ into two disjoint subsets: $\mathrm{CCC}^{A,B}_{\mathrm{train}} \cup \mathrm{CCC}^{A,B}_{\mathrm{eval}} = \mathrm{CCC}^{A,B}$. This is done in five cross-validation rounds for each sensor pair, by a test-split percentage of 20 %. We want to emphasize that the cross-validation is commonly used regarding supervised learning, while in our approach the labels are only exploited for the evaluation step and not for the model estimation. The input set of our automatic calibration routine for a sensor pair $(S_A,\ S_B)$ is defined by:

$$\mathrm{Input}^{A,B}_{\mathrm{preFilter}} = \{(P_i^A,\ P_j^B)\} \setminus \mathrm{CCC}^{A,B}_{\mathrm{eval}} \tag{18}$$

To determine the mapping accuracy of a transformation $\hat{T}_{A,B}$ between the local coordinate frames of a sensor pair (S_A, S_B), the *mean transformation error* over all track-intersection pairs of the evaluation set $\mathrm{CCC}^{A,B}_{\mathrm{eval}}$ of the sensor pair is calculated by:

$$\mathrm{J_{mean}}(\hat{T}_{A,B}) = \frac{1}{|\mathrm{CCC}^{A,B}_{\mathrm{eval}}|} \sum_{(P_i^A, P_j^B) \in \mathrm{CCC}^{A,B}_{\mathrm{eval}}} \mathrm{J_t}(\hat{T}_{A,B}, (P_i^A, P_j^B)) \tag{19}$$

The comparison between our automatic calibration approach and the manual-selection calibration is done by considering the difference of their mean transformation errors for each pair of sensors.

D. Parameter Set

We set the parameters of our calibration routine as following: $\lambda_{\mathrm{speed}} = 41\,\mathrm{cm/s}$, $\lambda_{\mathrm{RANSAC}} = 30\,\mathrm{cm}$, $\lambda_{\mathrm{L}} = 0.7\,\mathrm{rad}$, $\omega = 4$, and $K = 1000$.

E. Evaluation Result of Speed Filtering

The pre-filtering by the average track-intersection speed deviation shows very good results. For all sensor pairs, no correct track correspondences were excluded, while the wrong track correspondences are reduced by over 93 % (see Table I), providing good conditions for the subsequent RANSAC-based algorithm.

F. Evaluation Results of Trackwise w-RANSAC

By our automatic calibration routine, we achieved in our evaluation setting the referencing of the sensor views to each other, depicted in Fig. 3. The averaged results of all cross-validation runs for each sensor pair are presented in Table I. The differences of the mean transformation errors between the automatic and the manual-selection calibration of the sensor pairs $(S_A,\ S_B)$ and $(S_A,\ S_C)$ show that it is possible to achieve better results of the mapping of the tracks in the overlapping areas through our automatic calibration, compared to a manual-selection calibration routine, even for relatively small overlapping detection areas, e.g., depicted in Fig. 4. The track mapping accuracy of our automatic calibration routine lies, with one exception, between approx. 10 to 16 cm. An exception is represented by sensor pair $(S_B,\ S_D)$, where the mapping for both calibration routines is significant less accurate.

Considering the results, we assume that the spatial coverage of the overlap, by the used detection points for calibration, is important for generalization to any unseen tracks within the overlap area. Whereas, the mere size of the overlap turned out to be less crucial for an accurate extrinsic calibration, by considering that we achieved the lowest mean transformation error for the sensor pair $(S_A,\ S_B)$, which has the smallest overlapping detection area.

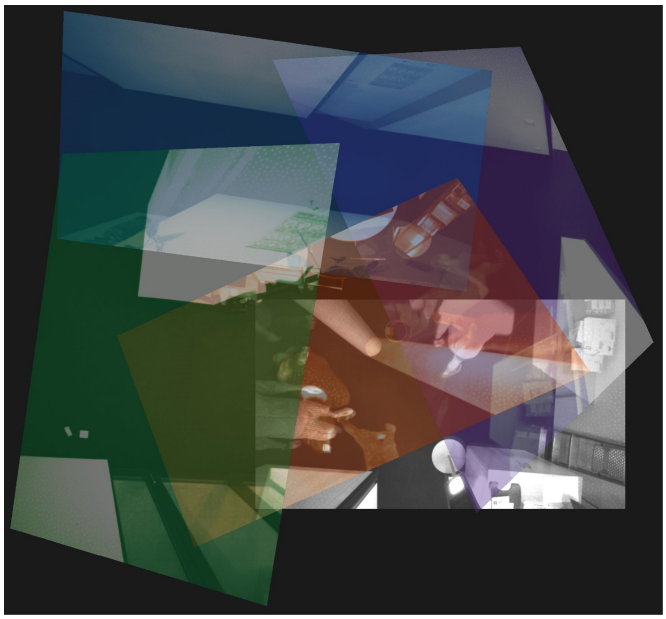

Fig. 3. Referencing of a multi-sensor network consisting of five depth sensors by automatic calibration based on local tracks. The estimated rigid transformations and the homography from each sensor to the root ground plane in image coordinates are applied. Sensors in reference to Table I: S_A: gray (root node); S_B: violet; S_C: red; S_D: blue; S_E: green.

V. Conclusion

We realized the automatic calibration only based on tracking information by a *trackwise w-RANSAC* with pre-filtering. By considering all point correspondences of a track pair as one unit, fewer combinatorial possibilities for composing the consensus set exist, compared to the multitude of combinatorial possibilities that point-based combinations would arise. Additionally, a very effective pre-filter, with an outlier detection rate above 93 %, also simplifies the problem, making the RANSAC-based model estimation more robust. Although not for all sensor pairs, it is shown that in some cases an improved track mapping can be achieved by the automatic calibration, in comparison to a manual selection of homologous point correspondences of the floor plane. The advantages of the self-calibration feature will in most cases outweigh slightly less mapping accuracies. For a planned indoor installation of a multi-sensor network, including an implicit known sensor

Sensor Pair	$(\mathbf{S_A}, \mathbf{S_B})$	$(\mathbf{S_A}, \mathbf{S_C})$	$(\mathbf{S_B}, \mathbf{S_D})$	$(\mathbf{S_C}, \mathbf{S_E})$
$\lvert \text{Input}_{\text{preFilter}} \rvert$ [track-intersection pairs]	339	538	318	446
Number of *wrong* track-intersection correspondences *before* pre-filtering	333	514	306	435
$\lvert \text{CCC}_{\text{train}} \rvert$	**6**	**24**	**12**	**11**
Number of *wrong* track-intersection correspondences *after* pre-filtering	**23**	**29**	**19**	**27**
Size of **RANSAC input set** [track-intersection pairs]	**29**	**53**	**31**	**38**
Percentage of CCC track-intersection pairs in RANSAC-input [%]	20.69	45.28	38.71	28.95
Size of maximum consensus set [track-intersection pairs]	6	22	11	11
Percentage of CCC track-intersection pairs in maximum consensus set [%]	100	100	100	100
$\lvert \text{CCC}_{\text{eval}} \rvert$	1	6	2	2
Mean error J_{mean} of automatic calibration [cm]	10.11	12.77	50.01	16.06
Mean error J_{mean} of manual-selection calibration [cm]	16.51	13.76	29.43	10.34
(automatic J_{mean}) - (manual-selection J_{mean}) [cm]	-6.4	-0.99	20.58	5.75

TABLE I
AVERAGED CROSS-VALIDATED RESULTS FOR EACH SENSOR PAIR

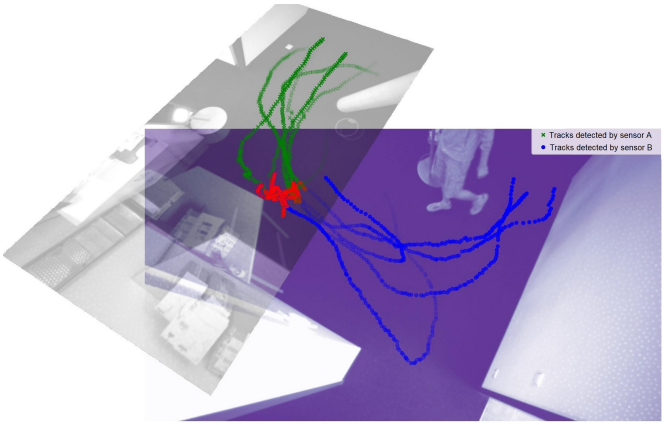

Fig. 4. Referencing of sensor pair $(S_A,\ S_B)$ by automatic calibration: The tracks depicted were used for the illustrated transformation estimation. More precisely, the track-intersections within the overlapping area (red points) constitute the maximum consensus set and were used for the transformation estimation. (Note that, caused by pyramidal FOV, the overlapping area regarding people detection is much smaller than the overlap of the FOVs.)

topology, a robust automatic extrinsic calibration, exclusively based on the local tracks, can be achieved.

VI. FUTURE WORK

Since our automatic extrinsic calibration based on tracks could also be beneficial for compensating non-linear systematic inaccuracies of a depth-sensor system, a combination with non-linear transformations could be promising. For example, [2] use thin plate splines in combination with their pointwise RANSAC-based method to compensate non-linear depth measurement distortions. Especially for large sensor networks, it should be considered that by searching for pairwise transformations between neighboring sensors, a significant error propagation along the chained transformations can occur. Therefore, a global optimization for the multi-sensor network calibration could be used, e.g., done by [2]. In addition, future work could extend our automatic calibration approach for non-overlapping FOVs, e.g., by a motion model to bridge the detection gaps between the sensors.

REFERENCES

[1] J. Wetzel, A. Laubenheimer and M. Heizmann, "Joint probabilistic people detection in overlapping depth images." in *IEEE Access*, vol. 8, pp. 28349-28359, 2020, doi: 10.1109/ACCESS.2020.2972055.

[2] O. Korkalo, T. Tikkanen, P. Kemppi and P. Honkamaa, "Auto-calibration of depth camera networks for people tracking," *Machine Vision Applications*, vol. 30, pp. 671–688, 2019. doi: 10.1007/s00138-019-01021-z.

[3] A. Aalerud, J. Dybedal, G. Hovland "Automatic Calibration of an Industrial RGB-D Camera Network Using Retroreflective Fiducial Markers", *Sensors* (Basel). 19(7). 2019. doi: 10.3390/s19071561.

[4] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Commun. ACM*, vol. 24(6). Oxford: Clarendon, pp. 381–395, 1981. doi: 10.1145/358669.358692.

[5] Y. Caspi, D. Simakov, & M. Irani, "Feature-Based Sequence-to-Sequence Matching.", *Int J Comput Vision* 68, 53–64, 2006. doi: 10.1007/s11263-005-4842-z.

[6] D.Zhang, W. Wang, H. Qingming, J. Shuqiang and W. Gao, "Matching images more efficiently with local descriptors", *2008 19th International Conference on Pattern Recognition* pp. 1-4 2008. doi: 10.1109/ICPR.2008.4761304.

[7] J. Wetzel, "Image Based 6-DOF Camera Pose Estimation with Weighted RANSAC 3D", *Pattern Recognition. GCPR 2013. Lecture Notes in Computer Science*, vol. 8142, pp. 249–254, 2008.

[8] M. Vlachos, D. Gunopulos and G. Das, "Rotation Invariant Distance Measures for Trajectories", in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp.707–712, 2004. doi: 10.1145/1014052.1014144.

[9] C. Stauffer and K. Tieu, "Automated multi-camera planar tracking correspondence modeling", *IEEE Conf. on Comput. Vis. Pattern Recognit. (CVPR)*, 2003. doi: 10.1109/cvpr.2003.1211362.

[10] A. S. Olagoke, H. Ibrahim and S. S. Teoh, "Literature Survey on Multi-Camera System and Its Application.", in *IEEE Access*, vol. 8, pp. 172892-172922, 2020, doi: 10.1109/ACCESS.2020.3024568.

[11] B. Pätzold, S. Bultmann and S. Behnke, "Online Marker-Free Extrinsic Camera Calibration Using Person Keypoint Detections", *Pattern Recognition, DAGM GCPR*, pp. 300–316, 2022.

[12] M. Munaro, F. Basso, and E. Menegatti, "OpenPTrack: Open source multi- camera calibration and people tracking for RGB-D camera networks", in *Robotics and Autonomous Systems 75*, pp. 525–538, 2016.

[13] M. Harville, " Stereo person tracking with adaptive plan-view statistical templates", *Image Vis. Comput.*, vol. 22, pp. 127–142, 2002.

[14] R.G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*. Englewood Cliffs, NJ: Prentice-Hall. 1963.

[15] G. Bradski, "The OpenCV Library". *Dr. Dobb's Journal of Software Tools*, 2000.

[16] J. Martin, J. Burbank, W. Kasch, and D. L. Mills, "Network Time Protocol Version 4: Protocol and Algorithms Specification", *RFC Editor*, 2010, doi: 10.17487/RFC5905.