Optimizing Long-Term Robot Tracking with Multi-Platform Sensor Fusion

Giuliano Albanese*1 Arka Mitra*1 Jan-Nico Zaech*†1 Yupeng Zhao*1 Ajad Chhatkuli[1]
Luc Van Gool[1,2,3]

[1]ETH Zurich, Zurich, Switzerland, [2]KU Leuven, Leuven, Belgium, [3]INSAIT, Sofia, Bulgaria

Abstract

Monitoring a fleet of robots requires stable long-term tracking with re-identification, which is yet an unsolved challenge in many scenarios. One application of this is the analysis of autonomous robotic soccer games at RoboCup. Tracking in these games requires handling of identically looking players, strong occlusions, and non-professional video recordings, but also offers state information estimated by the robots. In order to make effective use of the information coming from the robot sensors, we propose a robust tracking and identification pipeline. It fuses external non-calibrated camera data with the robots' internal states using quadratic optimization for tracklet matching. The approach is validated using game recordings from previous RoboCup World Cup tournaments.

1. INTRODUCTION

Robust tracking with stable object identification is a crucial component in many robot applications. Previous works in related tasks use robot-mounted sensors in order to achieve the task in various settings [7, 10, 14, 40]. A related task in a different setting is analyzing motions of dynamic agents (*e.g.*, humans in sports) through a fixed external camera [26, 27, 47, 48]. In this work, we propose to fuse information from both types of sensors to robustly track humanoid robots in entire soccer game videos. Although the problem is closely related to automated game analytics, the availability and use of internal robot sensors brings its own unique applications, challenges and opportunities.

We focus on matches in the RoboCup Standard Platform League (SPL), where humanoid NAO robots from two teams compete fully autonomously in soccer matches. Robocup is an international annual competition where teams program different robots to compete in soccer. The long term goal of the project is to have a team of humanoid robots that can win against the winners of the World Cup in compliance with the official rules of FIFA. One of the main platforms in the competition is the SPL where two teams score using five NAO robots each. The actions performed by the robots are autonomous and the first team to score 10 goals or the team with the highest number of goals after twenty minutes are announced the winners. The teams can only make changes to the software present in the NAO robots and no modifications to the hardware are allowed. Making game analytics available in this league can help teams improve their gameplay by providing an objective way of comparing the performance of their algorithms.

Our problem differs in multiple ways from the well-known tracking and identification problem game analytics: RoboCup games are recorded with non-professional uncalibrated camera equipment, robots look identical except for their jerseys, jersey numbers are too small to detect reliably, and human referees often occlude a significant part of the scene. These specifics introduce unique and non-trivial challenges into our long-term tracking task. In particular, the re-identification by recognition becomes virtually infeasible which is not the case in standard game analytics.

Like previous methods, we start with the tracking of individual robots. Different tracking methods can be used, however, our tracklets are obtained solely from visual features and do not extend to the whole duration of the game. We, therefore, opt for the use of the internal states of the robot in order to extract more useful attributes, which we call features. As we show in the paper, these attributes can be efficiently used in order to match the different tracklets with the robot tracks. In this work, we formulate the tracking problem as a biquadratic optimization where the internal states of the robots are used to provide different costs, used to collate the different tracklets. Overall, we propose a long-term tracking pipeline consisting of the following modules:

1. Camera calibration, to estimate camera intrinsics, including distortion, and the extrinsic camera pose relative to the playing field.

*Authors contributed equally and are ordered alphabetically
†Corresponding author. E-mail: nomadz@list.ee.ethz.ch

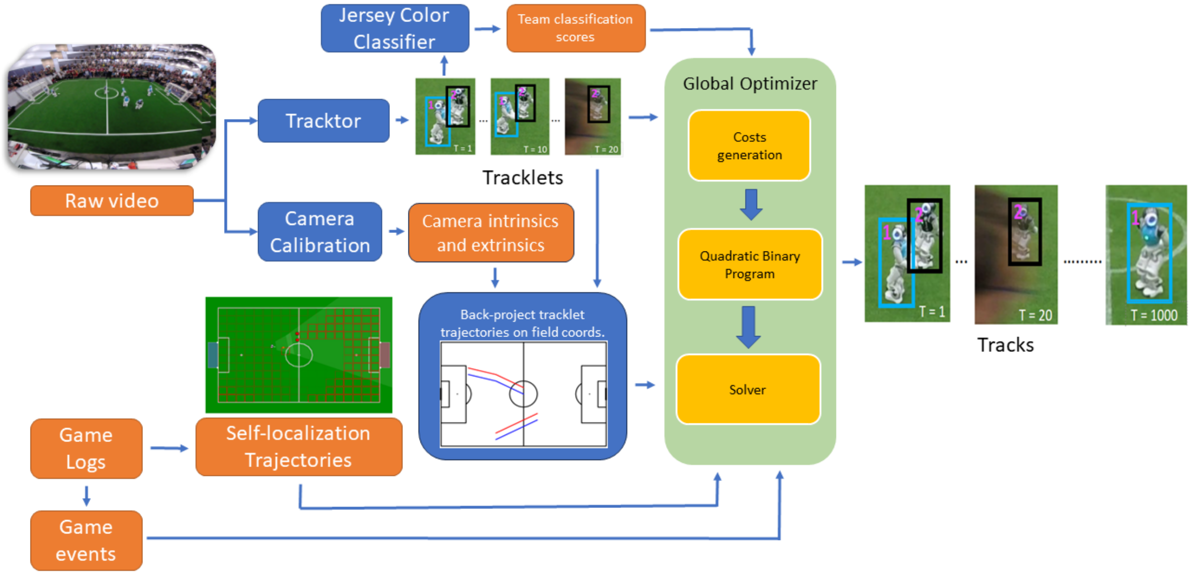

Figure 1. Overview of the proposed approach. The pipeline includes the processed raw video as well as the robot states as the inputs. The processed raw video provides the tracklets from Tracktor and the jersey/team classification as inputs to the optimizer. The robot states used as inputs are the self-localization and fallen state. Another important component that facilitates fusion of these inputs is the camera calibration module. The multi-modal inputs are fed into the global optimizer in order to generate the final track results.

2. Short-term object tracking, to generate tracklets using Tracktor [4] with a Faster-RCNN [38] object detector pretrained on MS-COCO and finetuned on our dataset.
3. Long-term object tracking, to match tracklets to player identity by optimizing a quadratic problem, which fuses visual detections from the external camera and the robot's own self-localization and status messages.
4. Optimizing the long-term tracking performance by fine-tuning the weights associated with the cost terms.

Furthermore, we open-source our code for future research[1].

2. Related Work

Multi Object Tracking (MOT) approaches the task of tracking all objects belonging to a given set of categories. These often include pedestrians [13, 23, 32] and vehicles in datasets containing videos from fixed surveillance cameras [13] or vehicle mounted moving cameras [9, 11, 42]. In joint tracking and detection approaches, the object detector is a fundamental part of the tracking pipeline. One of these approaches is Tracktor [4], which we use as one building block in our pipeline. Tracktor utilizes the bounding box refinement module of Faster R-CNN [38] to propagate the bounding boxes of all tracks through a video sequence. More recently, Meinhardt et al. [31] proposed a transformer architecture in the joint detection and tracking framework. To achieve high performance in challenging scenarios such as scenes with moving cameras or object occlusions, additional motion-compensation [4] and re-identification [20, 29, 57] modules are often integrated in the tracking pipeline.

Another large group of trackers follows the tracking by detection paradigm, where tracking is performed using the detections provided by a separate object detector. In this scenario, a data association problem needs to be solved, wherein approaches include fully deep learning-based methods [8, 12, 54, 58] as well as optimization based approaches [16, 17, 39, 44, 55]. Closest to our data association formulation is Qtrack [55], which proposes to solve a quadratic tracking association problem using quantum computing. While we solve a similar optimization problem, our formulation allows for a larger problem size as tracklets are matched. Furthemore, our formulation at the same time integrates player identification into the tracking problem.

Person re-identification is a core component of many tracking approaches, as it provides strong cues to match pedestrians after occlusions or crossing paths. A common paradigm in this context is metric learning [24, 33, 50], where features are learned together with a metric that measures the similarity between objects. This aims at jointly finding an embedding space and corresponding learned metric to distinguish between different pedestrians. Us-

[1] https://github.com/nomadz-ethz/spl-player-tracking-release

ing a GAN and disentanglement of appearance and pose, Zheng *et al*. [57] are able to extend the domain covered during training.

However, training data for re-identification raises strong privacy protection concerns which have recently led to a movement towards training the module on primarily synthetic data. In this context, PersonX [43] and Bak *et al*. [3] are notable examples that use a small set of models to generate training data. To push to surpass the scale of human-generated data, RandPerson [49] and ClonedPerson [53] further propose to automate this pipeline by generating randomized character clothing. In our setting, re-identification based trackers are strongly limited by the similarity of different robots, as well as the low amount of training data that is available for the resulting fine-grained task. Thus, we employ an extension of DeepSort [50, 51] as our baseline, which combines location information and a deep appearance feature into a single optimization problem.

Game Analytics One of the main problems in game analytics is tracking and identification of players in videos [28, 47, 52]. MOTs are the first key components of the pipeline, which provide candidate detections of the players. Other components include team detection [52] or a combination of team and jersey identification [15]. The work by Maglo *et al*. [30] uses detection followed by association of tracklets in sports videos using player re-identification. In this case, tracklet association is also learned as the method does not have other inputs including spatial locations for the association. [27] on the other hand, uses the estimated spatial image locations of the players for the task. However, as our problem is different from standard game analytics formulations, the solutions presented in previous works [26, 30, 45, 47] are not directly applicable to our task. Specifically, a key problem that is not approached, is the full integration of the 3D environment as well as 3D localization of the players in player tracking and identification.

Camera Calibration Exploiting the known 3D environment during tracking and identity assignment requires accurate camera intrinsics and extrinsics, where identifying these parameters is performed by camera calibration. Standard calibration processes generally provide accurate intrinsic parameters [56] using multiple views of a calibration pattern. Alternative approaches without calibration patterns use minimal point correspondences [36] or a robot's known motion for camera calibration [37] by evaluating 3D-2D correspondences with the Direct Linear Transform [1]. Similarly, Scaramuzza et al. [41] uses a 3D laser sensor to obtain highly accurate camera intrinsics. In contrast to this, our application has to work with a single pose video, where the factory-calibrated intrinsics are further known to be inaccurate. Furthermore, dynamic scenes and texture-less regions lead to poor point correspondences. To alleviate these challenges, our approach utilizes the technique proposed by Alvarez et al. [2], which minimizes an energy objective based on rectifying straight lines that are present on the soccerfield.

Particle Swarm Optimization A core component of our method is the fusion of different sources of information through optimization. In such scenarios, the best objective weights of the optimization problem are often obtained using an exhaustive grid search. However, this process is computationally expensive and requires discretizing the search space. As an alternative, meta-heuristic algorithms such as simulated annealing [21] and particle swarm optimization (PSO) [35] have shown good results in various domains [18, 19]. In our work, the PSO algorithm is used for the constrained optimization of the weights for different cost terms.

3. Method

In this section, we detail our pipeline for consistent player tracking and identification. Figure 1 provides an overview of the key components in our target application. Our pipeline consists of three parts. First, the camera intrinsics and extrinsics are estimated using field features, such as lines and corners, whose dimensions and relative positions are known a priori. Then, player tracklets are generated and the jersey color is estimated for each tracklet. Additional information, such as the players' self-estimated position and game state are extracted from the game logs. The final step associates each tracklet with a specific robot player. We perform this crucial step by optimizing a binary quadratic program. The performance is further improved by finding the best cost weighting using PSO. In the following, each component is described in detail.

3.1. Data and Application

We consider RoboCup Soccer SPL matches between teams of 5 NAO robots, where data is acquired from an external camera as well as the game-log. The game-log is generated by the Game Controller, which communicates the game state (start, end, free-kick, player penalties) to the players through WiFi. Furthermore, each player is required to send a heartbeat network packet including its estimated position to the Game Controller at 1Hz. These are logged by the Game Controller together with the game states. In addition, players can exchange information with their team members by broadcasting network packets at a fixed rate. These are also captured and logged by the Game Controller. Our dataset is composed of 8 annotated 5000-frame sequences recorded with a wide-angle camera at 30 FPS and the Game Controller logs of the corresponding matches. The sequences were extracted from videos recorded at RoboCup 2019 and 2022, and the frame timestamps have been synchronized with the Game Controller logs. The annotations include the bounding box, jersey color and number of each active player which is visible on the field in each frame. The object detection and image classification models and the optimizer's weights are trained on five of these sequences. The remaining three sequences are used for evaluation.

3.2. Camera Calibration and Pose Estimation

Accurate camera calibration and pose estimation is essential in our method to locate robots in the image on the field and to remove false positive detections outside the field boundaries. Estimating the radial distortion coefficients is especially important in this case due to the barrel distortion introduced by the wide angle lens.

We assume a static camera over the sequence, which is the setup used for all RoboCup game recordings. We, therefore, use the known geometry and dimensions of the field lines for the camera calibration. The main pre-requisite for the task is to establish clean images with clear correspondences between the target frame and the known 3D geometry. Due to moving robots and humans on the field, occlusions are present. We resolve these and obtain a clean unoccluded view of the field by computing the median image over the whole sequence.

Widely used calibration algorithms that are implemented in common computer vision toolboxes require either multiple views of a flat calibration target [56] or several accurate 2D-3D correspondences of non-coplanar points on the calibration target. In our application, however, the former approach is not applicable due to the lack of camera motion. We further observe that the later algorithms, based on 2D-3D correspondence fail to jointly estimate distortion coefficients, intrinsics and extrinsics. This is due to the low number of available calibration points and missing good initial estimate of the distortion coefficients. Therefore, we approach the problem in two steps:

First, we estimate radial distortion coefficients by leveraging the fact that field lines should be straight. Groups of points belonging to the same field lines are selected and used to formulate the optimization problem according to Alvarez et al. [2].

In the second step, extrinsics are computed and the focal lengths are refined if needed. To this end, we leverage the known 3D soccer field landmarks, specifically the line intersection positions. After undistorting the median image using the parameters estimated in the previous step, line segments are detected with the SOLD2 [34] line detector. To filter out initial false positives, a mask of the field area is estimated using color thresholding. Lines are then further refined with morphological dilation followed by the Spaghetti algorithm [6] for connected component labeling. The remaining line segments are merged into large and straight field lines by clustering them based on their proximity of endpoints and collinearity [46].

Intersections are computed from the detected and post-processed lines, which provides the required 2D-3D point correspondences to the ground truth 3D field coordinates. Altogether, we obtain 7 reliable point pairs in each of the videos. In order to compute the camera poses, the P3P [22] algorithm followed by a non-linear refinement step is utilized. Although the non-linear refinement can potentially further improve the intrinsics, we find that the intrinsics are already accurate enough for our purpose at this stage.

3.3. Multi Object Tracker

To generate bounding box tracklets, we use Tracktor [4] with Faster-RCNN [38] with Feature Pyramid Networks (FPN) and a ResNet-50 backbone. We initialize the model with MS-COCO [25] pre-trained weights and fine-tune it on the training sequences of our dataset to detect robot players. Since during matches players often occlude each other for several seconds, we set the patience of the tracker to 1 and use conservative thresholds for the NMS step to prevent tracklets from switching from one player to another. In this way, when players cluster in one area of the field and occlude each other, several short-lived tracklets are initialized. Our optimizer is then able to robustly combine these into longer tracks.

Subsequently, the trajectory of each player tracklet is converted to field coordinates, (x, y). We approximate the position of the robots' feet by the midpoint of the lower side of each bounding box. This point is then projected to field coordinates using the camera pose estimated during calibration to obtain 2D positions in field coordinates. Each resulting projected tracklet j is smoothed using a Kalman filter with a constant velocity model.

3.4. Jersey Color Detection

In the SPL, 9 distinct jersey colors are used. These colors, known for each match, provide a strong signal to associate tracklets with players from either team. We thus train a VGG16 network to detect jersey colors for each tracklet and assign a score for each of the team colors. As the colors of the two playing teams are known, only predictions for these are considered at this stage.

3.5. Robot States

The Game Controller logs include several sources of information about the state of the active players at every point in the game. In our formulation, we make use of information from the following states to match tracklets to players:

Self Localization: The robots calculate their position on the field based on the field landmarks they observe with the onboard cameras. The estimated positions are often sufficiently accurate and can be correlated with tracklet trajectories to provide a strong signal for identification. However, relying on this signal alone is not possible, as they can diverge arbitrarily far from the true value due to drastic changes in lighting conditions or other factors like the players losing track when falling over.

Fallen Robot: The robots use the IMU information and heuristics to determine when they fall. In the external camera, when a player falls, its bounding box has an aspect ratio higher than 1. Therefore, a player tracklet whose bounding box has an aspect ratio higher than a given threshold for a certain number of consecutive frames is considered to be a fallen player tracklet, which can be matched to the robots' internal states and provides another strong signal.

Penalties: In RoboCup soccer, the robots are penalized and removed from the field if they fail to follow the game rules, e.g. if they commit a foul or suddenly start leaving the field. These events are used to add constraints to the problem to prevent the optimizer from matching an active tracklet to a penalized player.

3.6. Global optimization

Even though there are at most 10 active robots in the considered soccer matches, occlusions and distractors cause Tracktor to split the tracks into a large number of tracklets. Therefore, we frame the long-term tracking problem as an assignment of *tracklets* to a fixed number of player *tracks*. It is modeled as a constrained quadratic binary optimization problem. We denote the index set of player tracks $I = \{1, ..., N\}$ (with $N = 10$) and generated tracklets $J = \{1, ..., M\}$. The objective is to minimize:

$$\begin{aligned} H(x) = &\sum_{i\in I}\sum_{j\in J} x_{i,j}(O_u + \sum_{l\in L} w^l c^l_{i,j}) \\ &+ \sum_{i\in I}\sum_{j\in J}\sum_{k\in J} x_{i,j}x_{i,k}(\sum_{p\in P} w^p c^p_{j,k}), \end{aligned} \tag{1}$$

where $x_{i,j} \in \{0, 1\}$ are binary optimization variables, with $x_{i,j} = 1$ meaning tracklet j is assigned to track i, L and P is the number of unary and pairwise cost functions, $c^l_{i,j}$ the unary (tracklet-to-track) costs, and $c^p_{i,j,k}$ the pairwise (tracklet-pair-to-track) costs. The scalars w^l, w^p are the *cost weights* and the scalars O_u denote the offsets. The offsets are negative to penalize the trivial solution of assigning nothing ($x_{i,j} = 0 \; \forall i, j$).

To prevent generating invalid tracking solutions, the following constraints are implemented. The first set of constraints

$$\sum_{i\in I} x_{i,j} \leq 1, \forall j \in J, \tag{2}$$

prevents assigning a single tracklet to multiple tracks. The second set of constraints is implemented to avoid merging temporally overlapping tracklets

$$x_{i,j}x_{i,k} = 0 \; \forall i \in I, \; \forall (j, k) \in J \times J : T_j \cap T_k \neq \emptyset,$$

where T_j, T_k represent the set of frames in which detections exist for tracklets j and k respectively.

3.7. Cost terms

Our formulation uses two types of cost terms: 1) Unary cost terms are a measure for the fit between tracklets and tracks. 2) Pairwise cost terms measure the fit between pairs of tracklets. Overall, we utilize the following cost-terms:

Self-localization - During the matches each robot sends its estimated position on the field (x, y) once per second. The signal is linearly interpolated between timestamps and the distance to the position estimated from the external camera is computed. Averaged over each tracklet, this provides a strong prior for the assignment problem. To encourage matching a tracklet to a player's track when its trajectory is close to the player's communicated trajectory, we define the following cost term:

$$c^{loc}_{i,j} = \frac{\beta^{loc}}{|T_j|} \sum_{t\in T_j} ||\hat{\tau}^t_j - \tilde{\tau}_i(t)|| \tag{3}$$

where β^{loc} is a scaling factor.

Jersey color detection - Let $\bar{p}^H_j$ and $\bar{p}^A_j$ be the mean probabilities of tracklet j belonging to a player of the **H**ome or **A**way teams respectively. We encourage matching tracklets to the correct team with:

$$c^{team}_{i,j} = \begin{cases} 1 - \bar{p}^H_j & \text{if } i \in I_H \\ 1 - \bar{p}^A_j & \text{if } i \in I_A \end{cases} \tag{4}$$

Fallen robot state - Fallen player tracklets detected with the heuristic described above can be easily matched to fallen player events in the Game Controller logs. Given a fallen robot event reported by player i recorded in a given time frame, for each fallen robot tracklets detected in the same time frame we add a fixed cost term $c^{fallen} = 1$ to discourage matching these tracklets to other players.

Duration - To filter out false positive tracklets, which are usually short, we use the following cost term to encourage matching with longer tracklets:

$$c^{duration}_{i,j} = \min(1, \frac{\mu}{T_j}) \tag{5}$$

where μ is a tunable threshold.

Global trajectory continuity - A pair of consecutive non-overlapping tracklets (j, k) is more likely to belong to the same track if the earlier tracklet "ends near" the start of the later tracklet. We extrapolate the pose of the earlier tracklet j from its end position using a constant velocity model:

$$\hat{\tau}_j(t) = (\hat{x}_i^{t_{j,\mathrm{f}}}, \hat{y}_i^{t_{j,\mathrm{f}}})^\top + (t - t_{j,\mathrm{f}})(\hat{v}_i^{t_{j,\mathrm{f}}}, \hat{w}_i^{t_{j,\mathrm{f}}})^\top \quad (6)$$

We define the following pairwise cost term based on the distance to the start of any temporally close tracklet:

$$\begin{aligned} &c_{i,j,k}^{cont} = ||\hat{\tau}_j(t_{k,\mathrm{i}}) - \hat{\tau}_k^{t_{k,\mathrm{i}}}|| \\ &\forall i \in I, (j,k) \in J \times J : 0 < t_{k,\mathrm{i}} - t_{j,\mathrm{f}} < \theta_{\mathrm{cont}} \end{aligned} \quad (7)$$

where $t_{j,\mathrm{f}} = \max(T_j)$, $t_{k,\mathrm{i}} = \min(T_k)$, $\hat{\theta}_k^{t_{k,\mathrm{i}}}$ is the earliest pose of tracklet k, and θ_{cont} is a tunable parameter.

3.8. Optimization of Cost Weighting

The weights for each cost term and the offsets define the optimization problem and thus the performance of the tracking results. Assigning a high weight to a cost term ensures that the optimizer pays more attention to that attribute, while the offsets implement an error threshold for tracklet assignment. We use PSO to optimize the weights for the cost terms using the ground truth training data to maximize the metrics. For this work, the initial particles are initialized randomly over the search space. The weights corresponding to the different cost terms and the offsets are the values represented by each particle. At each step, the values of the particles that correspond to the cost terms are updated such that they are non-negative and their sum is normalized to 1. This ensures that no redundant information is modeled. At the same time, the value corresponding to the offset for each particle is kept negative. In this work, 50 particles are initialized randomly and the optimization is run for 100 iterations. The cognitive parameter and the social parameter, which control a particle's affinity to its best position and the global best position are kept at 2 for the whole run. The objective function is the average of the MPIR for all the sequences and the optimization aims to maximize it. The search stops when the number of iterations are completed or when all particles have converged to the same position.

3.9. Reference Method: DeepSORT

While our task provides information beyond what common tracking pipelines are able to utilize, it is important to quantify the performance relative to existing trackers. To fulfill the task of long-term tracking and player identification we augment the DeepSORT tracker [50,51] by a greedy tracklet matching algorithm.

DeepSORT is an extension of the optimization-based SORT algorithm [5] that integrates appearance information from a pre-trained network to create a deep association metric. This metric combines motion and appearance cues to establish measurement-to-track associations during tracking. Motion cues are integrated through Kalman filtering and data association is performed using the Hungarian algorithm.

Time Frame	30	150	300	900	1800	3600	5400
MPIR	42.45	42.31	40.08	43.75	38.40	38.51	38.51

Table 1. DeepSort Performance with different re-identification time.

While DeepSORT can handle occlusions by using the re-identification module, it commonly is only able to do so over short timeframes. We, therefore, combine it with a *greedy tracklet matching* approach. Any tracklet which has not been assigned is matched with the spatialy closest inactive track. Furthermore, constraints are applied to prevent multiple tracklets being assigned to the same track if they have any time overlap. At the start of each run, the total number of robots which are present in that session is provided for initialization. This provides additional privileged information and can bound the maximum number of tracks which are generated.

Furthermore, a pure tracking pipeline like DeepSORT is able to generate long-term tracks, but cannot detect the ID of each robot. To circumvent this issue, we manually assign the first tracklet appearing for each robot to the corresponding ground-truth ID, which forms an oracle approach for identification i.e. a perfect tracker would also perform perfect identification using this approach. While this provides additional information beyond what is used in our method, it allows us to compare our method to a fair tracking-baseline that uses the best-possible identification approach.

Finally, to allow for a well performing baseline and fair comparison, we tune the DeepSort baseline re-identification time over the test set. Table 1 shows the different metrics for different values of re-identification time. The algorithm's performance shows an increase with the increase in the number of frames within the reidentification window. However, when it is too high, the performance degrades, as the initial tracklets are more likely to contain ID-switches and therefore contain errors that cannot be corrected later.Based on this, we select a maximum re-identification time of 900 frames corresponding to 30s of video for our baseline method.

4. Experimental Results

We evaluate our approach over a test set of 3 sequences of 5000 frames recorded at 30 frames per second. Each video covers a different game, thus testing our approach with different levels of player self-localization accuracy, team colors, and environmental conditions. Since we are primarily interested in correctly identifying players in ev-

Method	MPIR
ours	88.11
Oracle Deepsort [51]	43.76

Table 2. Results on the testset for our approach and the extended deepsort baseline. All values are provided in percent.

Full	Self Loc.	Tracklet Duration	Team Det.	Fallen Flag	Penalized Flag	Tracklet Distance
88.11	15.39	51.14	76.48	86.22	76.27	83.33

Table 3. The ablation study evaluates the influence of removing different information used to match tracklets to tracks. All numbers are provided in percent MPIR on the testset.

ery frame, commonly used MOT metrics such as MOTA are not relevant, as they do not measure player identification performance. Therefore, we define an ad-hoc metric more suited to our problem setting, the Mean Player Identification Recall (MPIR). For each frame t in a sequence, we match the bounding boxes predicted by the tracker to the ground truth based on an IoU-threshold of 0.5. We denote with TP_t the number of correctly identified bounding boxes and with FN_t the number of incorrectly identified bounding boxes. The tracking metrics then read as:

$$\mathrm{MPIR} = \frac{1}{T}\sum_{t=1}^{T}\frac{TP_t}{TP_t + FN_t}. \tag{8}$$

Table 3 shows the Mean Player Identification Recall (MPIR), the ratio of times each player has been identified correctly. The first column shows our full approach. Subsequent columns show ablations, with each feature removed separately. The cost weightings are optimized using PSO for each scenario. Figures 2 and 3 provide a visual representation of the results obtained with our algorithm on a sequence from a match played at RoboCup 2019.

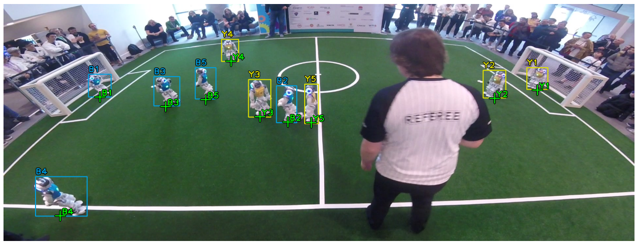

Figure 2. Visualization of robots identified by the tracker. The tracking result is represented by bounding boxes and IDs at their top. Ground truth positions are represented by green crosses and corresponding green IDs.

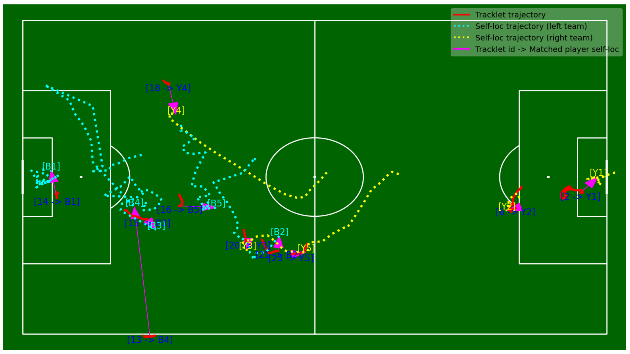

Figure 3. Top-view of Figure 2. The dotted lines represent the players' self-localization trajectories. The red lines are the tracklet trajectories in field coordinates. For each tracklet, the original ID and the matched player ID are shown. The purple arrows connect each tracklet trajectory to the self-localization trajectory of the player to which the tracklet has been matched.

4.1. Ablation Study

We perform an ablation study and depict results in Table 3. With all features, we achieve 88.11% MPIR. Removing the robot self localization has the strongest impact with 15.39% MPIR remaining, while removing the fallen robot flag results in the least performance drop. This is expected since the self-localization is an important attribute that provides information about the position of the robot in the field and consequently the image. The fallen robot flag is noisy, as it relies on the robot's IMU and an approximate heuristic to detect whether the robot has fallen in the video.

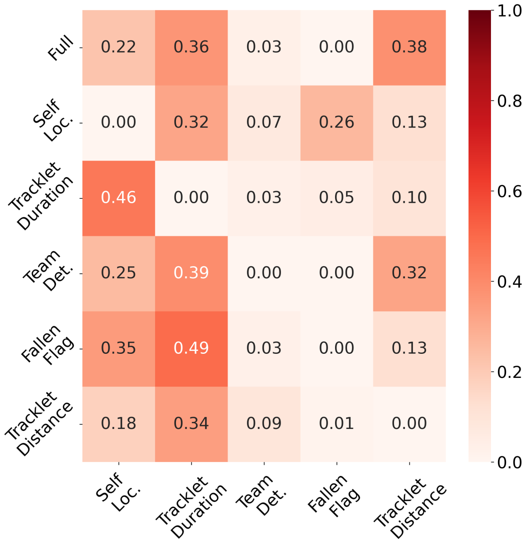

Figure 4. Feature weights for ablated features.

Seq #	Self-Local.	Jersey	Track Time	Fallen	Traj. cont.	Offset
6	0.27	0.16	0.34	0.02	0.22	-0.19
7	0.20	0.0	0.4	0.04	0.36	-0.45
8	0.35	0.28	0.23	0.0	0.14	-0.2

Table 4. Cost weights optimized individually for each sequence in the test set with PSO.

4.2. Feature Importance

We further analyze each feature's importance through the weights obtained from the PSO optimizer, where a higher weight indicates higher importance. Figure 4 shows the importance of the features in each column for the ablations represented by each row of the table. The first row corresponds to the full model and each following row to one of the ablations where a single feature weight is set to zero.

While the weighting of different features needs to be handled with care due to their scaling, we can compare the weights of the same feature in different ablations directly. Strong weights are assigned to the self-localization and tracklet duration, which provide strong indicators for matching and tracklet confidence. Removing these features shows that weighting is redistributed: While in the full model the noisy fallen robot events are not used, they are incorporated when no self-localization information is available. In this case, the primary source of information to match tracklets to robot IDs is missing but can be replaced by matching the fallen robots.

4.3. Explainability

Using several sequences for the optimization of cost weights yields weights which can generalize to new data. However, since the matches are played by different teams which have non-identical algorithms running on the robots, the weights might be suboptimal for some matches. By searching for the parameters which yield the best results on a single sequence, we can further understand the shortcomings and types of noise exhibited by each team. This further allows us to better understand and explain the inner workings of the proposed algorithm. For this purpose, we use PSO to optimize the weights individually for each sequence in the test set to find the optimal cost weights. Then we compare the resulting parameters, reported in Table 4, with qualitative observations on the game-videos itself. We discuss the outcome of this analysis in the following.

Sequence 6 - In this sequence, the self-localization cost is given a relatively high weight. This sequence was extracted from the 2022 championship final between the top teams in the league and it was played under ideal lighting conditions, so the players' self-localization is accurate. Thus, a higher weight on this feature is expected. However, throughout this sequence, several players are penalized and manually moved outside of the field, which causes their internal position estimate to diverge and match less closely the tracklet trajectories. The jersey color detection is also assigned a comparatively high weight. The jersey colors of the two teams differ strongly and are detected accurately.

Finally, we can observe that tracklet duration is also given high importance. In this sequence, the players are completely occluded by the referees at several points in the sequence, during which many short-lived false-positive tracklets are created. However, since the players are distributed evenly on the field the rest of the time, there are also several long-lived tracklets. Since matching these correctly can significantly affect the metrics, prioritizing this feature helps the identification process.

Sequence 7 - In this sequence, the self-localization weight bears the lowest weight of all three sequences. This is because the self-localization is accurate for one of the teams, but is often very noisy and incorrect for the other. The players are rarely occluded by the referees and the players do not often cluster in one part of the field as often seen these matches. As a result, there are several long lived and very accurate tracklets, hence the higher importance given to tracklet duration. The large offset term is most likely related to the low occurrence of tracklet switching and detection false positives, the optimization. Hence, this term discourages the optimizer from discarding tracklets.

Sequence 8 - In this sequence, the self-localization has a high weight. Self-localization is accurate for one team (black jersey), but is rather unreliable for the other team (yellow jersey) because of a software malfunction causing the players to often report their position to be in the center of field. However, several players of the later team are penalized for the game for most of the sequence, which means no tracklets are assigned to them due to the constraints. As a result, most of the tracklets represent the black team, resulting in a good self-localization performance, which makes it an important feature. Jersey color detection also carry high importance in this sequence. This is because the jersey colors of the two teams are strongly distinct and easy to detect, such that team detection can easily help differentiate tracklets belonging to players of different teams.

5. Conclusion

In this work, we presented a sensor fusion based method for tracking multiple similar humanoid robots. We utilize information from both visual data and their own sensors by combining tracklets using a quadratic optimization technique. The method allows automated tracking of robots over a long time on a stationary video sequence. Open points that we will investigate in the future include the evaluation in more complex environments as well as the interpolation of tracks during occlusions.

References

[1] YI Adbel-Aziz. Direct linear transformation from comparator coordinates into object space in close-range photogrammetry. In *ASP Symp. Proc. on Close-Range Photogrammetry, American Society of Photogrammetry, Falls Church, 1971*, pages 1–18, 1971. 3

[2] Luis Alvarez, Luis Gómez, and J Rafael Sendra. An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision*, 35(1):36–50, 2009. 3, 4

[3] Slawomir Bak, Peter Carr, and Jean-Francois Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking Without Bells and Whistles. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 941–951, Seoul, Korea (South), Oct. 2019. IEEE. 2, 4

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple Online and Realtime Tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, Sept. 2016. 6

[6] Federico Bolelli, Stefano Allegretti, Lorenzo Baraldi, and Costantino Grana. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. *IEEE Transactions on Image Processing*, 29:1999–2012, 2019. 4

[7] Tara Boroushaki, Isaac Perper, Mergen Nachin, Alberto Rodriguez, and Fadel Adib. Rfusion: Robotic grasping via rf-visual sensing and learning. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, page 192–205, New York, NY, USA, 2021. Association for Computing Machinery. 1

[8] Guillem Braso and Laura Leal-Taixe. Learning a Neural Solver for Multiple Object Tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6246–6256, Seattle, WA, USA, June 2020. IEEE. 2

[9] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020. 2

[10] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 1

[11] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3D tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[12] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a Proposal Classifier for Multiple Object Tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2452, Nashville, TN, USA, June 2021. IEEE. 2

[13] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003 [cs]*, Mar. 2020. 2

[14] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing. 1

[15] Sebastian Gerke, Karsten Muller, and Ralf Schafer. Soccer jersey number recognition using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 17–24, 2015. 3

[16] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted Disjoint Paths with Application in Multiple Object Tracking. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4364–4375. PMLR, Nov. 2020. 2

[17] Andrea Hornakova, Timo Kaiser, Paul Swoboda, Michal Rolinek, Bodo Rosenhahn, and Roberto Henschel. Making Higher Order MOT Scalable: An Efficient Approximate Solver for Lifted Disjoint Paths. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, page 11, 2021. 2

[18] Md. Alamgir Hossain, Hemanshu Roy Pota, Stefano Squartini, and Ahmed F. Abdou. Modified pso algorithm for real-time energy management in grid-connected microgrids. *Renewable Energy*, 2018. 3

[19] Gourhari Jana, Arka Mitra, Sudip Pan, Shamik Sural, and Pratim Kumar Chattaraj. Modified particle swarm optimization algorithms for the generation of stable structures of carbon clusters, cn (n = 3-6, 10). *Frontiers in Chemistry*, 7, 2019. 3

[20] Shyamgopal Karthik, Ameya Prabhu, and Vineet Gandhi. Simple Unsupervised Multi-Object Tracking. *arXiv:2006.02609 [cs]*, June 2020. 2

[21] Scott Kirkpatrick, Charles D. Gelatt, and Michelle Vecchi. Optimization by simulated annealing. *Science*, 220:671 – 680, 1983. 3

[22] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR 2011*, pages 2969–2976. IEEE, 2011. 4

[23] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv:1504.01942 [cs]*, Apr. 2015. 2

[24] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z. Li. Person re-identification by Local Maximal Occurrence representation and metric learning. In *2015 IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 2197–2206, Boston, MA, USA, June 2015. IEEE. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*, May 2014. 4

[26] Hongshan Liu, Colin Adreon, Noah Wagnon, Abdul Latif Bamba, Xueshen Li, Huapu Liu, Steven MacCall, and Yu Gan. Automated player identification and indexing using two-stage deep learning network. *Scientific Reports*, 13(1):10036, 2023. 1, 3

[27] Hengyue Liu and Bir Bhanu. Pose-guided r-cnn for jersey number recognition in sports. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 3

[28] Wei-Lwun Lu, Jo-Anne Ting, James J Little, and Kevin P Murphy. Learning to track and identify players from broadcast sports videos. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1704–1716, 2013. 3

[29] Liqian Ma, Siyu Tang, Michael J. Black, and Luc Van Gool. Customized Multi-person Tracker. In C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, volume 11362, pages 612–628. Springer International Publishing, Cham, 2019. 2

[30] Adrien Maglo, Astrid Orcesi, and Quoc-Cuong Pham. Efficient tracking of team sport players with few game-specific annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3461–3471, 2022. 3

[31] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-object tracking with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[32] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A Benchmark for Multi-Object Tracking. *arXiv:1603.00831 [cs]*, May 2016. 2

[33] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton Van Den Hengel. Learning to rank in person re-identification with metric ensembles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1846–1855, Boston, MA, USA, June 2015. IEEE. 2

[34] Remi Pautrat, Juan-Ting Lin, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Sold2: Self-supervised occlusion-aware line description and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11368–11378, June 2021. 4

[35] Riccardo Poli, James Kennedy, and Tim M. Blackwell. Particle swarm optimization. *Swarm Intelligence*, 1:33–57, 1995. 3

[36] Marc Pollefeys, Luc Van Gool, and Andre Oosterlinck. The modulus constraint: a new constraint self-calibration. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 349–353. IEEE, 1996. 3

[37] Thomas Probst, Kevis-Kokitsi Maninis, Ajad Chhatkuli, Mouloud Ourak, Emmanuel Vander Poorten, and Luc Van Gool. Automatic tool landmark detection for stereo vision in robot-assisted retinal surgery. *IEEE Robotics and Automation Letters*, 3(1):612–619, 2017. 3

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017. 2, 4

[39] Weihong Ren, Xinchao Wang, Jiandong Tian, Yandong Tang, and Antoni B. Chan. Tracking-by-Counting: Using Network Flows on Crowd Density Maps for Tracking Multiple Targets. *IEEE Transactions on Image Processing*, 30:1439–1452, 2021. 2

[40] Ashutosh Saxena, Lawson Wong, Morgan Quigley, and Andrew Y. Ng. A vision-based system for grasping novel objects in cluttered environments. In Makoto Kaneko and Yoshihiko Nakamura, editors, *Robotics Research*, pages 337–348, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. 1

[41] Davide Scaramuzza, Ahad Harati, and Roland Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4169. IEEE, 2007. 3

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451, Seattle, WA, USA, June 2020. IEEE. 2

[43] Xiaoxiao Sun and Liang Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *CVPR*, 2019. 3

[44] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple People Tracking by Lifted Multicut and Person Re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3701–3710, Honolulu, HI, July 2017. IEEE. 2

[45] Rajkumar Theagarajan and Bir Bhanu. An automated system for generating tactical performance statistics for individual soccer players from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(2):632–646, 2020. 3

[46] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs, 1998. 4

[47] Kanav Vats, William McNally, Pascale Walters, David A Clausi, and John S Zelek. Ice hockey player identification via transformers and weakly supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3451–3460, 2022. 1, 3

[48] Kanav Vats, Pascale Walters, Mehrnaz Fani, David A Clausi, and John Zelek. Player tracking and identification in ice hockey. *arXiv preprint arXiv:2110.03090*, 2021. 1

[49] Yanan Wang, Shengcai Liao, and Ling Shao. Surpassing real-world source training data: Random 3d characters for

generalizable person re-identification. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 3422–3430, New York, NY, USA, 2020. Association for Computing Machinery. 3

[50] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018. 2, 3, 6

[51] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017. 3, 6, 7

[52] Taiki Yamamoto, Hirokatsu Kataoka, Masaki Hayashi, Yoshimitsu Aoki, Kyoko Oshima, and Masamoto Tanabiki. Multiple players tracking and identification using group detection and player number recognition in sports video. In *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, pages 2442–2446. IEEE, 2013. 3

[53] Xuezhi Liang Yanan Wang and Shengcai Liao. Cloning Outfits from Real-World Images to 3D Characters for Generalizable Person Re-Identification. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[54] Jan-Nico Zaech, Dengxin Dai, Alexander Liniger, Martin Danelljan, and Luc Van Gool. Learnable online graph representations for 3d multi-object tracking. In *IEEE International Conference on Robotics and Automation, ICRA*, 2022. 2

[55] Jan-Nico Zaech, Alexander Liniger, Martin Danelljan, Dengxin Dai, and Luc Van Gool. Adiabatic quantum computing for multi object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8811–8822, June 2022. 2

[56] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 3, 4

[57] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[58] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking Objects as Points. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 474–490, Cham, 2020. Springer International Publishing. 2