Multi-camera People Tracking With Projection-Detection Fitting

Szymon Komorowski

March 9, 2025

Abstract

Multi-Camera Multiple Object Tracking (MCMOT) is a complex computer vision problem focused on identifying and continuously tracking multiple objects across multiple camera views. Most MCMOT solutions either perform single-camera tracking within each view, then match objects across cameras or take a global tracking approach, associating objects across all views simultaneously. Few trackers avoid the typical association step, which involves linking object detections across cameras. In this paper, we propose a novel framework for a lightweight, association-free tracker achieving real-time tracking performance without sacrificing accuracy. Our approach is based on the insight that, although estimating 3D positions from detections is challenging, verifying whether a set of hypothetical 3D positions could yield the observed detections is not a demanding task. By exploring a variety of configurations of 3D targets and picking the best candidate at each stage, we generate a scene across the time depicting individuals moving naturally across the plane while considering detections from all cameras. The proposed method, though highly efficient, achieves performance comparable to the state-of-the-art models on ICCV 2021 MMPT dataset, attaining a MOTA score of 94% and an IDF1 score of 90%.

1 Introduction

Multi-camera multi-object tracking (MCMOT) is a core area in computer vision with broad applications in surveillance, autonomous driving, sports analytics and others. Its goal is to maintain a unique, consistent identity for each trajectory across all cameras. This problem poses significant challenges, including object detection, cross-camera identity matching and frequent occlusions.[1]

Traditional approaches to MCMOT typically fall into two main categories. The first category is *Two-Step Hierarchical tracking*, where object detection and tracking are performed independently within each camera view before linking identities across views using re-identification (ReID) algorithms [2] [3]. Many of the leading MCMOT methods currently rely on this approach due to its modularity and its ability to harness specialized ReID algorithms for identity matching across different perspectives.

The second category is *Global Tracking frameworks*, which integrate all camera views into a unified model that associates object identities across the entire multi-camera system [4]. These frameworks often employ graph-based solutions [5] or end-to-end deep learning models [6] to tackle the high complexity of cross-camera associations. While these systems generally offer higher accuracy and more consistent identity tracking, they are computationally intensive, which poses challenges for real-time performance in large-scale environments.

In this paper, we introduce an innovative, lightweight framework that achieves performance comparable to state-of-the-art models while optimizing for computational efficiency. By leveraging camera calibration data, our method evaluates candidate placements of tracked objects across the scene, enabling efficient verification of their 3D projections against each camera's detections. We hypothesize that this verification-centric approach, which assesses potential solutions rather than estimating one, could be broadly applicable to a range of existing MCMOT methodologies, potentially establishing a new paradigm within the field.

Our method demonstrates competitive performance on the ICCV 2021 MMPT dataset, achieving a MOTA score of 94% and an IDF1 score of 90%. The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 details the methodology behind our approach; Section 4 describes the experimental setup and results; and Section 5 discusses findings and avenues for future research.

2 Related Work

2.1 Single-Camera Tracking

Single-camera tracking has gained significant attention due to its applications in surveillance and autonomous systems. Approaches in this field are generally categorized into two main types: tracking-by-detection and end-to-end tracking methods. Tracking-by-detection methods [7, 8] rely on object detection to locate targets in each frame, followed by data association to form tracklets. In contrast, end-to-end methods [9, 10] aim to directly learn temporal relationships between frames, eliminating the need for explicit detection and association steps.

Despite recent advancements, single-camera tracking remains a challenging problem, particularly in complex scenes with frequent occlusions and high clutter. Both object detection and tracklet association suffer from issues related to false positives, missed detections, and difficulties in maintaining consistent identity associations across frames [11]. In our approach, we simplify the problem by removing the tracklet association step, resulting in a more efficient framework that reduces computational complexity while maintaining tracking accuracy.

2.2 Multi-Camera Tracking

A prominent approach to multi-camera tracking (MCT) extends single-camera tracking (SCT), where tracklets are first generated using SCT, followed by an association step to link them across multiple cameras. This method builds on the strengths of SCT, but the challenge lies in accurately associating tracklets between views [4].

Alternatively, global or centralized methods associate detections across all cameras without relying on single-camera views. These approaches focus on global consistency and often aim to optimize the overall tracking performance by considering the entire set of detections simultaneously [12].

2.3 3D Multi-Camera Tracking

3D Multi-Target Multi-Camera Tracking (3D-MT-MCT) aims to track multiple targets across multiple camera views while estimating their 3D positions. The challenge lies in associating 2D detections from different cameras and reconstructing the 3D locations of targets. Several methods tackle this by triangulating the 2D detections and optimizing for 3D positions [13, 14]. However, the presence of occlusions, camera calibration errors, and identity switches often complicates the task [15].

Recent approaches employ global multi-target tracking (MCT) algorithms to manage target identities across frames and views [16], with graph-based techniques often used for data association [17]. Learning-based methods, including deep neural networks, are also becoming increasingly popular for improving performance, particularly in complex environments with challenging lighting and camera angles [18]. Despite significant progress, the task remains challenging due to issues like occlusion handling and accurate identity maintenance across different views [19].

3 Methodology

Consider Figure 1 below. Given detections, potential 3D locations and their projections it is clear to the human eye how to evaluate hypothetical solutions. The following section aims to formalize this process, breaking it down into key components: person detection, efficient placement space search, and fitting projections to detections.

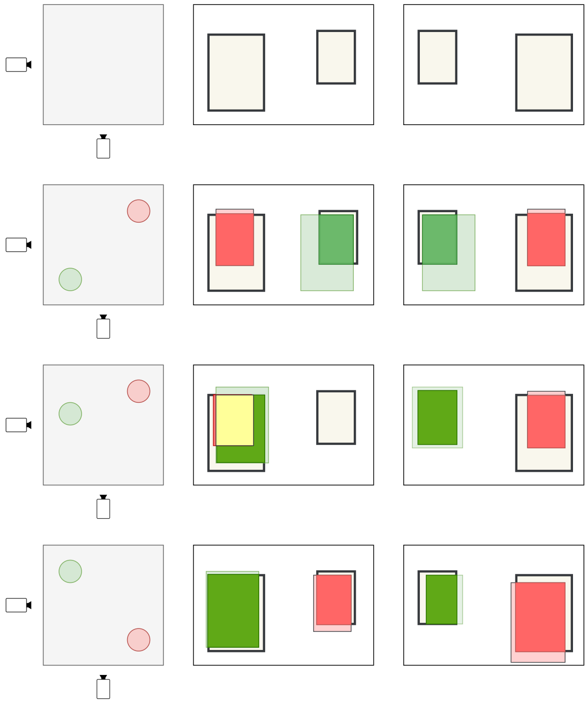

Figure 1: Camera setup with detections and potential candidate placements together with their projections overlaid on detections. The first row shows a scene with two cameras. We do not know where individuals are placed on the topdown view, but we do get detections. Notice that we do not know which detections from both views correspond to each other. Following rows present a potential configuration together with corresponding projections. Those are overlaid on previous detections to show a fit.

3.1 Overview

Our tracker seeks to determine the optimal configuration of individuals on a floor plane such that their 3D-to-2D projections fit the detections as accurately as possible. Thus, person detection is a foundational step, even though the final tracks will be assigned bounding boxes of projections rather the detections.

To fit projections to detections, we must generate candidate 2D placements for individuals on the floor plane. Following the initial step the candidate configurations are derived from previous candidates.

Given a configuration we can generate corresponding projections. Rather than directly estimating the optimal placement from detections, our approach evaluates a wide range of potential projections, selecting the one whose projections best fit the detections. Since we'll be evaluating thousands of potential configurations to make this approach work in real time requires a method that is both fast and computationally efficient, while robustly fitting projections to detections.

3.2 Person Detection

For person detection, we employ YOLOv7 [20], a high-performing single-stage detector trained on the COCO dataset. Since our primary concern is minimizing false negatives rather than false positives, we apply a relatively low confidence threshold, $c_\epsilon = 0.25$, retaining only boxes with confidence $c > c_\epsilon$.

Algorithm 1 Multi-Camera Placement and Scoring Algorithm

```
Initialize placements ← $\emptyset$
for each frame do
    detections ← GET_DETECTIONS(frame)
    placements ← GET_PLACEMENTS(placements)
    projections ← GET_PROJECTIONS(placements)
    scores ← FIT(projections, detections)
    top_placement ← TOP_FIT(placements, scores)
end for
```

Additionally, YOLOv7 applies Non-Maximum Suppression (NMS) with an Intersection over Union (IoU) threshold. We set this threshold to 0.45 to strike a balance between removing redundant detections and ensuring that each individual is assigned only one bounding box. Although the final tracks do not directly use these detection boxes, they rely on YOLOv7's bounding boxes for initial positioning and to maintain continuity in the tracking process.

3.3 2D Configuration Search Strategy

Given the vast space of potential configurations and the unknown number of individuals, we need an efficient search for candidate placements. We define a configuration (also referred to as a placement) c as a list of 2D coordinates on a floor plane, representing all individuals within an area of interest. Specifically, for a configuration c,

$$c = [(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)] \,,$$

where (x_i, y_i) are the 2D coordinates on the floor plane, and $m \in \mathbb{N}$ represents the number of individuals in the placement.

3.3.1 Number of Individuals

The first key factor is the number of individuals in the scene. We assume that the number of people, m, in the scene remains fixed throughout the entire sequence. The issue of individuals entering and exiting the scene is discussed in the *Discussions* section. For the initial search, we separately consider placements with m individuals for each integer m in the range $[0, M]$, where M is selected to be sufficiently large to exceed the true number of individuals while minimizing bias. After the initial configuration, the number of individuals is implicitly determined by the best-scoring placement found in the initial search.

3.3.2 Initial Search

The initial frame represents a special case in the search for possible placements, as there are no prior placements to build upon. In this case, the search is a variation of Algorithm 1, where we iteratively generate random placements, score them, pick the best one, and then generate random placements around the previous best placement.

Given an area of interest, we select x_l, x_s, y_l, and y_s, ensuring that any person projected onto any view and positioned at coordinates (x, y) satisfies $x_l < x < x_s$ and $y_l < y < y_s$.

To initiate the search, we draw k sets of m (x, y) coordinates from a uniform distribution over the rectangle defined by $[x_l, x_s] \times [y_l, y_s]$. Each of these k placements receives a fit score, after which we pick the top placement. For the placement with the best score, we generate k new placements by drawing m coordinates (x, y) where $(x_i, y_i) \sim \mathcal{N}((x_i, y_i), R \cdot \mathbf{I})$. We then update R by setting $R = R \cdot \alpha$, where $\alpha \in (0, 1)$. This process continues iterating, gradually refining the placement, until $R < 1$ reaches below $1\mathrm{cm}$, at which point the search stops.

3.3.3 Following Searches

After the initial step, all subsequent steps assume that the Euclidean distance between consecutive ground truth placements is small enough to justify local search strategies. We assume that for each pair of consecutive coordinates, the distance will not exceed a certain threshold.

Thus, similar to the iterative process described in the *Initial Search* section, we start with a single placement and generate k new placements by drawing m coordinates (x, y) where $(x_i, y_i) \sim \mathcal{N}((x_i, y_i), R \cdot \mathbf{I})$. From these k placements we pick the one with the highest projection-detection fit. The difference here is that this step is performed only once, with R fixed to reflect small changes in placements. In our case, we set $R = 10\,\mathrm{cm}$.

3.3.4 ID Assignment

Given the assumption of small distances between corresponding coordinates across consecutive configurations it is natural to assign to them same global ID. Therefore for configuration c, individual centered at (x_i, y_i) and all its projections will be mapped to i-th global ID.

3.4 Projection-Detection Fitting

3.4.1 Generating Projections

In the previous section, we explained how to generate the 2D floor placements for the individuals. To get their projections we require 3D locations to project them to each camera view. We assume that, given placement p, 3D locations are cylinders with fixed height $h = 175\,\text{cm}$, width $w = 25\,\text{cm}$ and a center at (x_i, y_i) in c.

We then transform the 3D points into pixel coordinates using the following equation, assuming a distortion-free projective transformation based on a pinhole camera model.

$$s \cdot \mathbf{p} = \mathbf{K}\left[\mathbf{R}|\mathbf{t}\right]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where: (X, Y, Z) is a 3D point expressed in the world coordinate system, $\mathbf{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ is the corresponding 2D pixel in the image plane, $\mathbf{K}$ is the camera intrinsic matrix, $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation matrices that convert world coordinates to the camera coordinate system (camera frame), s is an arbitrary scaling factor, part of the projective transformation but not part of the camera model itself.

By expanding $\mathbf{K}$ and $\left[\mathbf{R}|\mathbf{t}\right]$ we arrive at the following equation:

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

Once the 3D object is projected onto the 2D plane, we calculate the bounding box by determining the minimum and maximum x and y coordinates among the projected points. The resulting bounding box coordinates $x_{\text{min}}, y_{\text{min}}$ and $x_{\text{max}}, y_{\text{max}}$ define the rectangular area in the image that encloses the projected 3D object.

3.4.2 Associating Projections with Detections

The core of our tracking approach lies in Projection-Detection Fitting, which evaluates how accurately the 3D-to-2D projections of individuals align with detections. To evaluate the fit between all projections and detections, we first assign a single fit score to each projection-detection pair. To do so we define a function $f : (\mathbb{R}^2 \times \mathbb{R}^2) \times (\mathbb{R}^2 \times \mathbb{R}^2) \to [-1, 1]$, which quantifies the fit between a single projection and a single detection.

Given the function f, a matrix of all pairwise fit scores can be constructed for each view. Given m projections and n detections, the matrix is:

$$\begin{bmatrix} f(p_1, d_1) & f(p_1, d_2) & \dots & f(p_1, d_m) \\ f(p_2, d_1) & f(p_2, d_2) & \dots & f(p_2, d_m) \\ \vdots & \vdots & \ddots & \vdots \\ f(p_n, d_1) & f(p_n, d_2) & \dots & f(p_n, d_m) \end{bmatrix}$$

,

where p_i is a bounding box of an individual from configuration c centered at (x_i, y_i).

If $m > n$ the matrix is extended along detections axis with null detections, ensuring that each projection can be assigned to a detection, even if there is no detection.

Since a single projection should only explain a single detection to get a cumulative score for a single view, we frame it as an assignment problem. We then assign a final score for a configuration c by using the Hungarian algorithm to obtain a score for the matrix from each view and aggregating the results.

3.4.3 Fit Function

To finalize projection-detection fitting we need to quantify the fit function f. The function operates on the assumption that detected individuals correspond to projections within their bounding boxes, while undetected regions should not contain any projections. Although no object detection algorithm can guarantee these assumptions, they are generally sufficient for our framework.

The function is split into two cases based on whether the bounding boxes intersect. When they do, Intersection over Union (IoU) is assigned as a score.

$$\text{IoU(A, B)} = \frac{|A \cap B|}{|A \cup B|}, \tag{3}$$

where A and B represent the two bounding boxes.

With high-quality detection, we expect the bounding box to closely encompass the individual. Therefore, if we consider an individual at the same place as the ground truth IoU should approach 1, but even with low quality detections we should see IoU value positive, when considering the ground truth configuration.

When projection and detection bounding boxes do not intersect (i.e. IoU $= 0$), the projection cannot reasonably explain the detection, even if the detection is of low quality. Thus, a straightforward approach would be to apply a penalty for that pair. To make this penalty more sophisticated and include occlusion treatment at the same time, consider a third row in Figure 1.

For LHS camera view, a green individual obstructs a red one. For that configuration both red and green projections intersect with the same single detection. When assigning projections to detections we see that although both projections explain LHS detection reasonably well, only one can be assigned to it, and the other must incur a penalty. This hypothetical configuration provides an explanation for the missing detection of the red person, as they are occluded by the green one.

To address missing detections and occlusions, we introduce a function $p : \mathbb{R}^2 \times \mathcal{C} \to [0, 1]$ calculating the occlusion level for an individual x and configuration c, where x represents 2D floor coordinates for an individual whose occlusion level is being determined. The function p computes the occlusion level based on the sum of the intersection areas between the bounding box of x and the bounding boxes of individuals $c \in \mathcal{C}$ that are closer to the camera than x, normalized by the area of x's bounding box.

$$p(x, c) = \frac{\sum_{c' \in C_{\text{closer}}(x)} \text{Area}(B_x \cap B_{c'})}{\text{Area}(B_x)}$$

where:

- B_x and $B_{c'}$ are the bounding boxes of individual x and individual c', respectively,
- $\text{Area}(B_x \cap B_{c'})$ is the area of intersection between the bounding boxes of x and c',
- $C_{\text{closer}}(x)$ is the set of individuals c' that are closer to the camera than x,
- $\text{Area}(B_x)$ is the area of the bounding box of x.

With that definition unoccluded individuals occlusion score of 0, while more occluded persons receive higher scores up to 1 for fully occluded ones.

Finally, we define the fit function f as follows:

$$f(p, d) = \begin{cases} \text{IoU}(p, d) & \text{if IoU}(p, d) > 0, \\ o(p) - 1 & \text{otherwise.} \end{cases}$$

This formulation rewards high IOU scores and penalizes missed projections based on their occlusion level.

4 Experiments

4.1 Dataset

The MMPTrack dataset, introduced in the ICCV 2021 Multi-camera Multiple People Tracking Challenge, is the largest publicly available dataset for 3D multi-camera multi-person tracking. This dataset spans over 9.6 hours of video across five environments: Retail, Lobby, Industry, Cafe, and Office. It features 28 subjects and 23 cameras, with four to six cameras per scene. [21]. Each environment's video data is densely labeled with RGB-D-assisted annotations, which are later refined by human labelers to ensure high accuracy.

Ground-truth bounding boxes are included for all views, even when individuals are partially or fully occluded, enabling testing in challenging, occlusion-prone environments like retail. Additionally, the dataset provides calibrated camera parameters, enabling accurate projection of 3D coordinates onto the 2D image plane, a key requirement not only for our approach [22].

4.2 Evaluation Metrics

To assess tracking performance, we use the MOTA (Multiple Object Tracking Accuracy) and IDF1 (IDentity F1 Score) metrics, applied to both 2D camera views and 3D-derived top-down coordinates with a 0.5-meter matching threshold. MOTA evaluates tracking accuracy by penalizing false positives, false negatives, and identity switches, providing a measure of detection and track consistency [23]. IIDF1 measures identity matching quality by calculating the ratio of correctly matched identities to the average number of ground-truth and predicted identities. This makes it particularly useful for evaluating association robustness in environments with frequent occlusions [22].

4.3 Comparison with Existing Trackers

To benchmark our method, we evaluated it against several existing multi-camera tracking approaches on the MMP Tracking dataset, including both baseline and advanced methods. Specifically, we compared with VoxelTrack, DMCT, and top performers from the CCV MMP-Tracking challenge.

The first baseline method we compare is VoxelTrack, built on a state-of-the-art 3D pose estimation method VoxelPose [24]. VoxelTrack generates 2D pose estimation for each view and later performs cross camera pose association. DMCT is a full end-to-end deep multi-camera tracker [25] generating a global heatmap by combining single-view heatmaps.

Participants from the CCV MMP-Tracking challenge, meanwhile, have introduced sophisticated association techniques tailored for multi-camera tracking [26]. Finally, MIO-MMPT has emerged as a recent state-of-the-art model, utilizing 2D-3D projection for robust multi-camera tracking [27].

There was some ambiguity regarding whether a distance threshold of 0.5m or 1m was used for top-down view evaluation. Thus, we used 0.5m as the threshold for our evaluation.

4.4 Results

Table 1 shows top-down view MOTA and IDF1 scores for our tracker, comparing them to other state-of-the-art approaches.

Method	MOTA	IDF1
Alibaba [26]	96.0	97.6
Hikvision [26]	96.0	91.1
MIO-MMPT [27]	94.5	96.3
Ours	**94.2**	**90.0**
DMCT [25]	88.8	56.0
VoxelTrack [21]	76.8	58.0

Table 1: Performance comparison of various tracking methods on MOTA and IDF1 metrics for 3D topdown view.

Analogous results for 2D camera views MOTA and IDF1 scores are shown in Table 2.

Method	MOTA	IDF1
Hikvision [26]	87.0	86.4
MIO-MMPT [27]	86.2	92.2
Alibaba [26]	78.0	88.2
Ours	**72.7**	**80.2**

Table 2: Performance comparison of various tracking methods on 2D MOTA and IDF1 metrics.

For the 3D evaluation Projection-Detection Fitting proved robust by scoring MOTA of 94.2 and 90.0 for IDF1. The 2D results score lower for both metrics compared to 3D topdown view and with a significant gap to state-of-the-art models. To break down the results we present all metrics for our tracker across each five different scenes.

Scene	2D MOTA	2D IDF1	3D MOTA	3D IDF1
Cafe Shop 0	49.6	71.9	98.6	93.3
Lobby 0	94.2	97.1	99.5	99.7
Industry Safety 0	85.2	92.3	92.2	96.1
Office 0	69.6	84.1	97.9	98.6
Retail 0	64.8	55.7	82.8	62.1

Table 3: Performance comparison across different scenes for both 2D and 3D MOTA and IDF1 metrics.

We first address the drop in 2D performance compared to 3D. As mentioned in section 3.4.1 we model each individual as a 3D cylinder with a fixed height and width. Although effective for estimating 3D foot coordinates, the projections can deviate from the ground truth due to variations between individuals and their poses.

The results show that *Lobby* and *Industry Safety* perform well in 2D metrics, while *Cafe Shop* and *Office* show a significant drop in performance compared to 3D. A common theme between *Lobby* and *Industry Safety* is that most individuals are standing throughout the video, whereas *Cafe Shop* and *Office* often feature sitting individuals. Upon investigating the videos with overlaid projections, we find that while the projections do intersect with sitting individuals, they often do not tightly encompass them in the 2D views.

Retail warrants separate discussion, as it ranks last in three metrics and second last in the fourth. The issues stem from the scene setup. An obstacle in the middle of the floor effectively separates the area into multiple subscenes. Even though *Retail* has six cameras, compared to four for the other scenes, many subareas are effectively monitored by just three or even two cameras. Its results highlight the impact of the number of cameras on the performance of our method. It is worth noting that our tracker scales well, with computational complexity growing linearly with the number of cameras.

5 Discussion

As a novel framework, the tracker has significant room for improvement, along with open areas for future research. Simultaneously, as any method, it has its limitations.

5.1 Area of Research

Projection-Detection Fitting proved extremely effective for top-down location estimation. However, the results for 2D camera views were significantly behind. Although only a conjecture, we believe this issue could be addressed with improved 3D modeling of individuals. We also suggest other potential improvements that were not explored in this work.

5.1.1 3D modeling of Individuals

Currently, each individual is represented as a cylinder of fixed height and width. This is a simplistic approach, where the intrinsic characteristics of individuals and their pose do not influence their projections. The framework can be improved by incorporating pose estimation for all individuals in the scene.

5.1.2 Different Detections Set Up

Detections are fundamental to tracking, playing a critical role in our method, where projections are directly compared to detections. Small detection variations—such as false positives, false negatives, or minimal overlaps with ground truth—can significantly affect tracking accuracy. These variations may lower the fit score of the correct placement compared to less accurate ones.

Optimal settings for detection confidence and IoU thresholds remain open questions, as they were not extensively examined in this work. Improvements could include testing different thresholds or considering multiple thresholds and aggregating results for projection-detection fits. Another approach could involve bypassing Non-Maximum Suppression (NMS) to retain valuable detections, fitting projections to densely clustered detections.

5.1.3 Expanded 2D Configuration Search Method

2D configuration search strategy uses previous best placement for next candidate configuration. Leaving multiple high-scoring and diverse configurations may lead to better representation across the time. Additionally, more sophisticated methods could be used to generate candidate configurations based on the previous highest-scoring configuration. One such enhancement could involve using a Kalman filter to estimate the initial coordinates for subsequent candidates. After obtaining this estimation, random draws could be made around the predicted coordinates with a smaller deviation. This approach could provide more targeted candidates, thus improving the overall efficiency and accuracy of the placement search.

5.1.4 Fitting Projections to the Ground Truth

Our method differs from black-box approaches by being highly transparent and based on elementary mathematical operations. This simplicity ensures interpretability but also comes with a limitation: the fit between projected bounding boxes and ground truth (GT) bounding boxes is not optimized.

Neural network could be trained to minimize the discrepancy between projections and corresponding GT bounding boxes. This could enhance the accuracy of the projections by learning optimal transformations from the projected coordinates to the GT coordinates, refining the overall fit.

While this would increase computational complexity, such an enhancement could significantly improve tracking performance. Future work could explore integrating this approach, providing more accurate projections and closer alignment with ground truth data.

5.2 Limitations

Several limitations in our current implementation have already been identified. For example, we model each individual as a variable-height cylinder, consider only top-down occlusions, and do not account for fixed obstacles within the scene. While these limitations could be mitigated through more sophisticated modeling and additional effort, certain constraints are inherent to our current methodology and cannot be resolved without fundamental changes to the approach.

5.2.1 ReID

Though rare, our tracker may occasionally experience identity switches. The underlying assumption behind global ID assignment is that the i-th coordinate in any configuration corresponds to the i-th track, based on spatial proximity. This assumption is reasonable because each placement is generated based on the previous one, with small changes in position, leading to coordinates in close proximity being naturally assigned to the same track.

However, we cannot entirely prevent scenarios where the ground truth coordinates of two individuals become sufficiently close. In such cases, the i-th and j-th coordinates may effectively switch places between configurations. To address this issue, a ReID (Re-Identification) algorithm could be employed, though we did not incorporate such methods in our tracker.

5.2.2 Handling Severe Misalignment

In our tracking framework, we assume continuous movement between frames, with corresponding coordinates in consecutive placements staying within a bounded range. However, when there is severe misalignment between the track and detections—such as when no detections correspond to a particular track across all views—our current framework lacks an effective method for recovery.

This misalignment presents a challenge, as the tracker cannot resolve the absence of valid detections. Expanding the search would not resolve the fundamental issue of lost correspondence between the track and the real world individual. One potential solution would be to consider non-continuous configurations, though this would result in unrealistic motion estimates. This could be considered an optional treatment, used only when severe misalignment is identified.

6 Conclusions

The presented tracker deviates from traditional methods that focus on associating detections directly. Instead, our approach reconstruct the scene by fitting hypothetical projections to detections. This work demonstrates that evaluating potential configurations can be both efficient and robust. It allows to estimate an optimal placement by choosing the candidate where projections align with detections the most.

Results on the ICCV 2021 MMP Tracking dataset further demonstrate the effectiveness and robustness of the proposed Projection-Detection Fitting framework. These results highlight its potential for deployment in real-world applications, offering reliable and accurate 2D and 3D multi-camera multiple people tracking. Our method stands out for its lightweight design, robustness in complex environments, and real-time performance, making it well-suited for practical use in dynamic scenarios.

References

[1] T. I. Amosa, P. Sebastian, L. I. Izhar, O. Ibrahim, L. S. Ayinla, A. A. Bahashwan, A. Bala, and Y. A. Samaila, “Multi-camera multi-object tracking: A review of current trends and future advances,”

Neurocomputing, vol. 552, p. 126558, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223006811 1

[2] S. Tang, M. Andriluka, B. Andres, and B. Schiele, “Multiple people tracking by lifted multicut and person re-identification,” IEEE, 2017. 1

[3] Z. Tang, G. Wang, H. Xiao, A. Zheng, and J.-N. Hwang, “Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 1

[4] A. Dehghan, S. Modiri Assari, and M. Shah, “Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4091–4099. 1, 2

[5] W. Liu, O. Camps, and M. Sznaier, “Multi-camera multi-object tracking,” *arXiv preprint arXiv:1709.07065*, 2017. 1

[6] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, “Deep learning for visual tracking: A comprehensive survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3943–3968, 2021. 1

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468. 2

[8] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649. 2

[9] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, “Tracking without bells and whistles,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 941–951. 2

[10] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 474–490. 2

[11] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, “Towards real-time multi-object tracking,” in *European conference on computer vision*. Springer, 2020, pp. 107–122. 2

[12] W. Chen, L. Cao, X. Chen, and K. Huang, “An equalized global graph model-based approach for multicamera object tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2367–2381, 2016. 2

[13] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, “Multicamera people tracking with a probabilistic occupancy map,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 267–282, 2008. 2

[14] A. Andriyenko and K. Schindler, “Multi-target tracking by continuous energy minimization,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1265–1272. 2

[15] P. Baque, A. Alahi, K. Schindler, P. Fua, and L. Van Gool, “Deep occlusion reasoning for multi-camera multi-target detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2213–2226, 2017. 2

[16] H. Fan, T. Zhao, Q. Wang, B. Fan, Y. Tang, and L. Liu, “Gmt: A robust global association model for multi-target multi-camera tracking,” *arXiv preprint arXiv:2407.01007*, 2024. 2

[17] Y. Pu, S. Fan, Y. Yang, and R. Tang, “Graph-based cross-camera continual tracking: Improvements to graph structure and feature extraction,” in *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*, ser. CAICE ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 422–429. [Online]. Available: https://doi.org/10.1145/3672758.3672827 2

[18] E. Luna, J. C. S. Miguel, J. M. Martínez, and M. Escudero-Viñolo, “Graph convolutional network for multi-target multi-camera vehicle tracking,” *arXiv preprint arXiv:2211.15538*, 2022. 2

[19] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “Mot16: A benchmark for multi-object tracking,” in *arXiv preprint arXiv:1603.00831*, 2016. 2

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” 2022. [Online]. Available: https://arxiv.org/abs/2207.02696 3

[21] X. Han, Q. You, C. Wang, Z. Zhang, P. Chu, H. Hu, J. Wang, and Z. Liu, “Mmptrack: Large-scale densely annotated multi-camera multiple people tracking benchmark,” 2021. [Online]. Available: https://arxiv.org/abs/2111.15157 6, 7

[22] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 17–35. 6, 7

[23] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear mot metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008. 7

[24] H. Tu, C. Wang, and W. Zeng, “Voxelpose: Towards multi-camera 3d human pose estimation in wild environment,” 2020. [Online]. Available: https://arxiv.org/abs/2004.06239 7

[25] Q. You and H. Jiang, “Real-time 3d deep multi-camera tracking,” *CoRR*, vol. abs/2003.11753, 2020. [Online]. Available: https://arxiv.org/abs/2003.11753 7

[26] ICCV 2021 MMP Tracking Challenge, “Iccv 2021 multi-camera multiple people tracking challenge,” https://iccv2021-mmp.github.io/, accessed: 2024-11-14. 7

[27] H.-M. Hsu, Z. Cheng, X. Yuan, and L. Chen, “2d-to-3d mutual iterative optimization for 3d multi-camera multiple people tracking,” in *2024 IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2024, pp. 1–7. 7