Toward Accurate Online Multi-target Multi-camera Tracking in Real-time

Andreas Specker[3,1] Jürgen Beyerer[1,2,3]

[1]*Fraunhofer IOSB*, Karlsruhe, Germany; [2]*Fraunhofer Center for Machine Learning*;
[3]*Vision and Fusion Lab, Institute for Anthropomatics and Robotics*,
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{andreas.specker, juergen.beyerer}@iosb.fraunhofer.de

***Abstract*—Multi-target multi-camera tracking is the task of determining the trajectories of objects within a network of cameras. Besides many others, it is a crucial task, e.g., in traffic analysis or law enforcement. Current research mainly focuses on offline algorithms for cross-camera association, which require all video data at once and often lack real-time capability. Thus, such approaches are unsuitable for tracking a criminal's escape route and enabling immediate intervention and arrest. To close this gap, we introduce a novel online multi-target multi-camera tracking method in this work which integrates spatial-temporal information and applies a new probability-based association metric. Moreover, we prove the real-time capability and significantly outperform the baseline on two datasets. In addition, the performance gap to more complex and slower offline methods is discussed to indicate the current state of research.**

***Index Terms*—tracking, multi-camera, online, real-time**

I. Introduction

Multi-target multi-camera tracking (MTMCT) aims at tracking objects such as persons or vehicles across a network of overlapping and non-overlapping cameras. Potential applications include the assistance of law enforcement agencies, but it is also crucial for automatic traffic monitoring and signal time planning. Recent works from the literature mostly focus on solving the task of cross-camera association with offline approaches [1]–[4]. In such a system, single-camera tracks are pre-calculated by an independent component and then clustered in a subsequent step. As a result, offline approaches require all data and information before the actual MTMCT. In contrast, online algorithms process the videos serially, i.e., frame-by-frame, necessary for real-time real-world applications. For instance, in the pursuit of criminals, decisions have to be made in real-time to arrest a suspect immediately. Since there is a lack of online algorithms in the research community, we propose a novel MTMCT framework that is evaluated on pedestrian and vehicle tracking tasks using the MTA [4] and the CityFlow [5] dataset, respectively. A new inter-camera association procedure represents the core contribution of our work. The component manages the life-cycle of multi-camera tracks (MCTs) based on confirmed single-camera tracks (SCTs) from different cameras. In detail, an association procedure is introduced that leverages information from overlapping cameras, spatial-temporal information, and a probability-based matching algorithm. The contributions of our work can be summarized as follows: i) We propose an online and real-time MTMCT component that operates on the frame-by-frame output of multiple tracking-by-detection single-camera trackers. ii) We introduce a novel probability-based matching procedure to connect single-camera tracks to previous occurrences of the same identity and explore the benefits of an implicit homography model to match SCTs from overlapping cameras. iii) Last, we provide the first baseline results of any online approach on the MTA dataset for pedestrian tracking [4] and demonstrate the transferability to different scenarios, such as vehicle tracking.

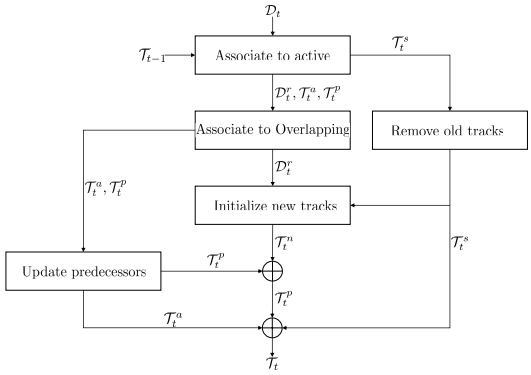

Fig. 1: Overview of the proposed online multi-camera tracker. Tracks are updated frame-by-frame based on current single-camera track detections $\mathcal{D}_t$. SCTs that are not assigned to a known MCT ($\mathcal{D}_t^r$) are either merged into a MCT visible in an overlapping camera, or initialized as a new MCT ($\mathcal{T}_t^n$) and soft-assigned to a set of possible predecessors.

II. Related Work

MTMCT has the goal to capture the trajectories of multiple objects across a network of possibly non-overlapping cameras. So far, the focus of the MTMCT research community has been mainly on offline systems that perform post-processing on the output of several single-camera trackers to merge the obtained SCTs into MCTs.

A. Offline MTMCT

Most commonly, offline MTMCT is treated as a clustering problem [1], [3], [4], [6], [7]. Pairwise distances are calculated

to determine whether a set of SCTs belong to the same object. Based on this procedure, recent advances indicate that the integration of a scene model to leverage spatial-temporal information is highly beneficial [1], [2], [4], [8]–[10]. Such information is either incorporated into the distance function [4] or is directly used as hard constraints to filter out infeasible cross-camera transitions [1], [8], [9]. Liu et al. [2] directly integrate the scene topology into the design of the clustering algorithm by proposing a local clustering method to merge tracklets in adjacent cameras first. Another work [10] goes one step further and only allows associating tracks from adjacent cameras. However, we argue that hard constraints are not feasible for person tracking since the movements of persons do not follow strict rules, and people might be entering buildings or cars and reappear in cameras further away. Thus, we rely on softly integrating spatial-temporal information in this work.

B. Online MTMCT

In contrast to the previously introduced approaches, online algorithms solve the MTMCT task on a frame-by-frame basis, i.e., clustering algorithms are not suitable. If an instance has already been seen, it is also desirable that new SCTs are assigned to the multi-camera trajectory of that instance as soon as possible, which makes hybrid approaches difficult. Therefore, the instance must be re-identified without having all information about the new tracklet available. Due to the more challenging nature, only a few works have dealt with solving it. Zhang et al. [11] introduce a framework for non-overlapping cameras. If a track is lost within a camera view, it is marked as such, and the algorithm tries to re-connect it to new detections solely based on appearance features. After a given period without a match, tracks may only be reactivated by detections from other cameras. Spatial-temporal information is integrated by analyzing usual transition times in the training data. The authors of [12] follow a similar pipeline but only use the feature distance as a similarity metric. Further work [13] deals with special cases such as networks, which solely consist of cameras with overlapping field of views (FOVs). Tracking is entirely performed in the world coordinate system based on visual, spatial, and motion information, i.e., no separate single-camera tracking stages are applied.

Our system builds on [11] but is enhanced by a handling mechanism for overlapping FOVs and an innovative probability-based inter-camera association procedure.

III. Methods

Based on a set of video streams $\mathcal{V} = \{V_1, \ldots, V_n\}$ from n synchronized cameras, detections of confirmed tracks $\mathcal{D}_t$ in each time step t are obtained by a single-camera tracking pipeline consisting of an object detector, a re-identification feature extractor, and a single-camera tracker. Each track detection $D_i \in \mathcal{D}_t$ is represented by a tuple (pos, f, c, ID_{SCT}), i.e., the bounding box position pos, an appearance feature vector f, the camera identifier $c \in \{1, \ldots, n\}$, and a unique single-camera track id ID_{SCT}. $\mathcal{T}$ stands for the set of MCTs currently considered by the tracker. A MCT $T_j \in \mathcal{T}$ is formed by a track id ID_{MCT}, a state s, the running average appearance feature $\tilde{\text{f}}$, a set of track detections $\{D_i | D_i \in \mathcal{D}_t, D_i \text{ shows objects } ID_{MCT}\}$, and the time step t_{last} when the track was lastly seen. In the following, we describe the baseline approach that resembles [12] first and propose our online multi-camera tracker approach afterward.

A. Baseline

In the baseline approach, we only consider two states $s \in \{active, sleeping\}$. In contrast to *sleeping* tracks, *active* tracks are currently visible in at least one camera. In the following, the sets of *sleeping* and *active* tracks are denoted $\mathcal{T}_t^s$ and $\mathcal{T}_t^a$, respectively. Furthermore, $\mathcal{T}_t = \mathcal{T}_t^s \cup \mathcal{T}_t^a$ applies for the baseline approach. In each iteration, the track detections included in $\mathcal{D}_t$ go through the following stages.

Associate to active - First, *active* MCTs are updated by assigning the current detections from corresponding SCTs. If an MCT is no longer tracked in any camera, its state will change to *sleeping*. Note that an MCT can disappear in one camera but remain active in another camera, i.e., the object left the overlapping area of the two cameras.

Associate to overlapping - Subsequently, remaining track detections $\mathcal{D}_t^r$ are associated to overlapping tracks. The Hungarian algorithm determines the best matching between new SCTs and existing MCTs based on appearance features for each overlapping camera pair. If the distance is below the threshold τ_o, an SCTs will be associated with the corresponding MCT. Next, the same procedure is applied to the residual track detections from the previous step and *sleeping* tracks. In this case, the threshold τ_{ic} is used to decide the association.

Initialize new tracks - Leftover SCTs show previously unseen identities and are thus initialized as new *active* MCTs.

Remove finished tracks - *Sleeping* tracks are finally closed when the time without update exceeds A_{max}, i.e., $t - t_{last} > A_{max}$ applies.

B. Improvements

One major drawback of current online multi-camera trackers is that SCTs are associated with MCTs directly when appearing in a camera or after a specific time interval, e.g., ten time steps. The first option is fast, but decisions must be made with little information. Appearance features might be biased since persons entering a camera are often only partly visible. Waiting for a fixed time interval increases the computational overhead and is often unnecessary since only a few tracks are possible predecessors. We address these issues by proposing an association mechanism that automatically determines the optimal decision time by taking possible predecessors into account. Moreover, we leverage spatial-temporal information to decrease the number of possible predecessors and apply an improved homography model.

In addition to the baseline, we introduce the *pending* state. The set of *pending* MCTs is referred to as $\mathcal{T}_t^p$ in the following and represents tracks with multiple possible predecessors. Fig. 1 provides an overview of the proposed online multi-camera tracker.

Associate to active - Analogous to the baseline approach, *active* tracks are updated in the first pipeline stage. Please note that this also applies to *pending* tracks. When the single-camera trackers no longer provide evidence for a *pending* track, it is lastly compared to its most likely predecessor. If the appearance feature distance is smaller than the threshold τ_{ic} the *pending* track is merged into its predecessor. Otherwise, a new *sleeping* identity hypothesis is initialized. To prevent the creation of too many hypotheses, short *pending* tracks, i.e., when $t - t_{last} < A_{min}$ applies, are deleted.

Associate to overlapping - The remaining SCTs in $\mathcal{D}_t^r$ are new to the multi-camera tracker, meaning one of three things. First, the SCT shows an object that is currently not visible in any other camera of the camera network. Second, a new SCT is currently visible in another overlapping camera, or third, an object enters two overlapping cameras simultaneously, thus producing two new SCTs. The mechanism to determine whether two tracks overlap is improved to handle these cases. A homography score is incorporated instead of solely relying on visual information about the objects' appearances and the information that the positions are located in an overlapping area. To do so, we train a fully-connected binary classifier for the camera network. Positions pos_1 and pos_2 with corresponding camera identifiers c_1 and c_2 serve as input and the training is supervised by the cross-entropy loss and the information whether the two track detections D_1 and D_2 belong to the same identity. Then, the homography score h is calculated as the inverse confidence P of the classifier that both tracks show the same object, as shown in Eq. 1.

$$h(D_1, D_2) = 1 - P(pos_1, pos_2, c_1, c_2), \tag{1}$$

The cost matrix in this associate to overlapping stage is constructed as the sum of the homography score and the appearance feature distance. Analogous to the baseline, cost matrices are calculated, and best agreements are determined with the Hungarian algorithm. This is done separately for the second and third cases. Similar to the baseline, τ_o is applied to decide whether two tracks should be associated.

Initialize new tracks - Track detections in $\mathcal{D}_t^r$ that are still not associated with an MCT either show a known identity last seen in the same camera, a known identity that lastly occurred in another camera, or an identity that has not been previously tracked. Albeit within-camera tracking is the task of the single-camera tracker, our multi-camera tracker handles the first option since especially persons might, e.g., enter and leave buildings or disappear for an extended period due to large obstacles. Current single-camera trackers only handle short-term occlusions lasting for a few frames. We treat the first two cases differently since camera-related characteristics such as illumination, viewpoint, and calibration influence the appearance features. For instance, the feature distance of similar objects within the same camera might be smaller than the distance of the same identity in different cameras. Therefore, an SCT is only matched to an MCT that was last visible in the same camera if the feature representations of the tracks are very similar measured by a separate within-camera threshold τ_{wc}. All SCTs not matched in the previous step are initialized as new MCTs but not linked to a unique identity hypothesis. Instead, they are connected to a set of possible predecessors $\mathcal{P}$. Possible predecessors are *sleeping* MCTs that were last active in another camera and meet the spatial requirement that a direct path between the cameras exists without crossing through the FOV of a third camera. In addition, temporal information is included to filter impossibly fast transition times Δt_{trans} by applying a time penalty tp. Eq. 2 shows the calculation based on the mean and standard deviation of all transitions between the two cameras observed in training data and a factor λ to balance the contribution.

τ_O	τ_{WC}	τ_{IC}	λ	Age_{Max}	Age_{Min}
1.6	1.2	0.7	0.1	4000	10

TABLE I: Overview of the hyper-parameters used.

$$tp = \frac{|\Delta t_{trans} - \mu_{trans}|}{\sigma_{trans}} \cdot \lambda \tag{2}$$

The primary motivation for this soft assignment of possible predecessors is that observing a tracklet for a more extended period allows more stable features. If an object enters the camera, it might be detected before it becomes entirely visible, thus strongly distorting the appearance feature. Additionally, suppose two MCTs are highly similar to the new SCT, and a decision based on the feature distance is difficult. In that case, one of the two possible predecessors may be merged with another SCT shortly after, leaving only one option, thus making the decision easier.

Update predecessors - *Pending* MCTs are updated in each tracker iteration to reduce the number of predecessors. The appearance feature distance and time penalty are computed for all remaining possible predecessors, using the current representation of the active tracklet. Subsequently, the Softmax function is applied to the negative distances to obtain matching probabilities for each possible predecessor. Unlikely predecessors are then removed from $\mathcal{P}$ if the matching probability is lower than $\frac{1}{|\mathcal{P}|}$. As the set becomes smaller, fewer comparisons have to be carried out in later iterations, making the procedure significantly faster than keeping a connection to all possible predecessors and making a decision after a specific time. Additionally, the decision process is way more flexible. If a possible predecessor has a very high matching probability, more competitors are eliminated, but when multiple tracks are similar, the decision is postponed until more information is available. When only one predecessor remains, it is merged with the MCT in *pending* state if their distance is smaller than the inter-camera threshold τ_{ic}. If the distance is too large, or if the set is empty at any other point, the MCT switches from *pending* into *active*, creating a new unique identity hypothesis.

Remove finished tracks - MCTs in $\mathcal{T}_t^s$ are closed and removed when the duration without update exceeds A_{max}.

IV. Evaluation

This section discusses the experiments conducted within the scope of this work. The experiments were performed with

Tracker	Online	IDF1	IDP	IDR	FPS
WDA [4]		28.9	32.0	26.3	-
Baseline	✓	22.5	24.2	21.0	21.3
Ours	✓	26.8	28.8	25.1	37.7

TABLE II: Comparison of our tracker with the baseline and the offline WDA tracker on the whole MTA dataset.

Distance Metric	τ_o	τ_{ic}	τ_{wc}	IDF1	IDP	IDR
Euclidean distance	1.50	1.20	0.70	36.9	39.3	34.8
Euclidean distance	1.60	1.20	0.70	**37.7**	**40.1**	**35.6**
Euclidean distance	1.70	1.20	0.70	36.1	38.4	34.1
Euclidean distance	1.60	1.15	0.70	36.8	39.1	34.8
Euclidean distance	1.60	1.20	0.70	**37.7**	**40.1**	**35.6**
Euclidean distance	1.60	1.30	0.70	35.8	38.1	33.9
Euclidean distance	1.60	1.20	0.65	36.7	39.0	34.7
Euclidean distance	1.60	1.20	0.70	**37.7**	**40.1**	**35.6**
Euclidean distance	1.60	1.20	0.75	36.5	38.8	34.4
Cosine similarity	1.50	0.25	0.25	36.8	39.1	34.6

TABLE III: Evaluation of the influence of threshold parameters on the ext-short subset of the MTA dataset.

an Nvidia GeForce GTX TITAN X and 10 Intel Xeon CPU e5-2630 v4. We evaluate our approach on two datasets from different domains: the Multi-camera Track Auto (MTA) [4] dataset for person and the second validation scenario from the CityFlow [5] dataset for vehicle tracking, respectively. Regarding evaluation metrics, the identity F1 (IDF1), identity precision (IDP), and identity recall (IDR) scores proposed by Ristani et al. [14] established themselves as the standard metrics for evaluating MTMCT systems. The single-camera tracking pipeline consists of well-established components that can run in real-time on affordable hardware [15]. The YOLOv5m [16] detector is followed by the ABD-Net [17] as feature extractor and DeepSORT [18] as the single-camera tracker. Hyper-parameters are provided in Tab. I.

A. MTA Dataset

We compare our tracker to the baseline and the state-of-the-art offline tracker on the MTA dataset in Tab. II. For a fairer comparison, all methods utilize the same single-camera tracking pipeline. Thus, differences are purely caused by the respective MTMCT component.

The results show that the presented algorithm achieves significantly better results with faster runtime than the baseline implementation, which resembles the algorithm by Gaikwad et al. [12]. Although the assignment of an SCT to possible predecessors seems more effortless in the baseline implementation as it is done immediately and involves fewer processing steps, it is still less efficient when there are many sleeping MCTs. First, spatial-temporal information is not used to filter possible predecessors. Instead, the feature distance to every *sleeping* MCT is calculated to create a cost matrix, which is then used by the Hungarian algorithm to calculate a minimum assignment. In addition, there are also more MCT hypotheses overall because false-positive SCTs that are too short are not removed as in the presented MTMCT component, and fewer

Homography	Time filtering	IDF1	IDP	IDR
No	No	33.0	35.1	31.2
No	Yes	33.4	35.5	31.5
Yes	No	37.3	39.7	35.2
Yes	Yes	**37.9**	**40.3**	**35.7**
Distance [4]	Yes	35.3	37.5	33.4

TABLE IV: Results with and without the homography model and time filtering on the ext-short subset of the MTA dataset. Additionally, the homography model is exchanged with a distance by transforming the position of a detection from one to the other camera.

λ	IDF1	IDP	IDR
0	37.4	39.8	35.2
0.05	37.4	39.9	35.3
0.10	**37.9**	**40.3**	**35.7**
0.15	37.5	39.9	35.3

TABLE V: Evaluation results for different values of the time penalty factor λ on the ext-short subset of the MTA dataset.

tracks are matched in the baseline approach. Moreover, the results prove the real-time capability of our tracker. 37.7 FPS are achieved for six parallel camera streams on the evaluation system. However, the results also show that the performance of the online tracking component is behind the state-of-the-art offline tracker. This finding was expected since offline algorithms perform global optimization strategies with all information available. In contrast, the online algorithm has to make decisions on the fly. In numbers, the difference between current state-of-the-art online and offline trackers is about 6.5% points or 28% (Baseline vs. WDA) regarding IDF1. We close this gap thereby reducing the difference to about 2% points or 7.8% (Ours vs. WDA).

Several ablation studies and justifications of parameter choices are presented in Tab. III, Tab. IV, and Tab. V. The results are achieved on the ext-short subset of the MTA dataset. One can observe that the use of homographies has the greatest impact on performance followed by the thresholds. Furthermore, in comparison with another homography approach from literature [4], our binary classifier achieves significantly better results. The use of the time filtering and the time penalty is beneficial as well, but improvements are smaller.

Fig. 2 shows a selection of trajectories through multiple cameras captured by our multi-camera tracker. Due to our handling of overlapping areas, persons are accurately tracked across multiple cameras, even if they are tiny and in the background. Moreover, the multi-camera tracker correctly re-identifies persons after interruptions caused by occlusions.

B. CityFlow

This experiment aims to evaluate the applicability of the proposed MTMCT component within a different context. Instead of adapting all components of the whole pipeline to the CityFlow dataset, the object detector, the feature extractor, and the single-camera tracking components are taken from [1]. In addition, some changes had to be made to the multi-camera tracking component. Due to a lack of training data, it is impos-

Fig. 2: Qualitative results.The tracks cross multiple camera and are correctly assigned to their ground truth identity. Ground truth is shown in green, and all other colors represent single-camera tracks assigned to the same identity hypothesis.

Tracker	Online	IDF1	IDP	IDR
Baseline from [1]		34.4	22.1	78.4
+ BG filtering + Exclude BG boxes + Exclude overlapping boxes		48.2	34.8	78.1
Baseline	✓	17.1	10.8	42.2
Ours	✓	40.6	39.7	41.6

TABLE VI: Evaluation results on Scenario 2 of the CityFlow dataset.

sible to train a homography model or derive temporal information. However, homography transformations are provided for each camera of the dataset to transform pixel coordinates into GPS positions, and one can approximate the distance in meters from two GPS positions. Such approximations lead to worse results for the MTA dataset than a learned implicit model. Still, the results are, in any case, better than without including any spatial information in the decision process. Since there was no way to obtain the required temporal information, we removed both the time filtering and the time penalty from the multi-camera tracking component.

Tab. VI compares our tracker with the baseline and [1]. Our tracker outperforms the baseline by even a large margin in comparison with the MTA dataset. The offline clustering baseline from [1] delivers worse results than our online tracker but exceeds it if additional preprocessing steps are carried out.

V. Conclusion

This work proposed an online multi-camera tracker that uses a novel cross-camera association method and spatial-temporal information to reduce the gap to more complex offline tracking algorithms. Our tracker significantly outperforms the baseline approach on two datasets and achieves real-time processing. We hope that our work will spark further research to close the gap between online and offline multi-camera trackers.

References

[1] Andreas Specker, Daniel Stadler, Lucas Florin, and Jurgen Beyerer, "An occlusion-aware multi-target multi-camera tracking system," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 4173–4182.

[2] Chong Liu, Yuqi Zhang, Hao Luo, Jiasheng Tang, Weihua Chen, Xianzhe Xu, Fan Wang, Hao Li, and Yi-Dong Shen, "City-scale multi-camera vehicle tracking guided by crossroad zones," *CoRR*, vol. abs/2105.06623, 2021.

[3] Ergys Ristani and Carlo Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6036–6046.

[4] Philipp Kohl, Andreas Specker, Arne Schumann, and Jurgen Beyerer, "The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[5] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David C. Anastasiu, and Jenq-Neng Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," *CoRR*, vol. abs/1903.09254, 2019.

[6] Peng Li, Jiabin Zhang, Zheng Zhu, Yanwei Li, Lu Jiang, and Guan Huang, "State-aware re-identification feature for multi-target multi-camera tracking," *CoRR*, vol. abs/1906.01357, 2019.

[7] Hung-Min Hsu, Jiarui Cai, Yizhou Wang, Jenq-Neng Hwang, and Kwang-Ju Kim, "Multi-target multi-camera tracking of vehicles using metadata-aided re-id and trajectory-based camera link model," 05 2021.

[8] Jin Ye, Xipeng Yang, Shuai Kang, Yue He, Weiming Zhang, Leping Huang, Minyue Jiang, Wei Zhang, Yifeng Shi, Meng Xia, and Xiao Tan, "A robust mtmc tracking system for ai-city challenge 2021," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 4044–4053.

[9] Yuhang He, Jie Han, Wentao Yu, Xiaopeng Hong, Xing Wei, and Yihong Gong, "City-scale multi-camera vehicle tracking by semantic attribute parsing and cross-camera tracklet matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[10] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann, "Electricity: An efficient multi-camera vehicle tracking system for intelligent city," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2511–2519.

[11] Xindi Zhang and Ebroul Izquierdo, "Real-time multi-target multi-camera tracking with spatial-temporal information," in *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019, pp. 1–4.

[12] Bipin Gaikwad and Abhijit Karmakar, "Smart surveillance system for real-time multi-person multi-camera tracking at the edge," *Journal of Real-Time Image Processing*, 02 2021.

[13] Elena Luna, Juan C. SanMiguel, José M. Martínez, and Marcos Escudero-Viñolo, "Online clustering-based multi-camera vehicle tracking in scenarios with overlapping fovs," *CoRR*, vol. abs/2102.04091, 2021.

[14] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," 2016.

[15] Andreas Specker, Lennart Moritz, Mickael Cormier, and Jürgen Beyerer, "Fast and lightweight online person search for large-scale surveillance systems," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 570–580.

[16] Glenn Jocher, Alex Stoken, Ayush Chaurasia, Jirka Borovec, NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang Hajek, and "ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support," Oct. 2021.

[17] Guangyi Chen, Chunze Lin, Liangliang Ren, Jiwen Lu, and Jie Zhou, "Self-critical attention learning for person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[18] Nicolai Wojke, Alex Bewley, and Dietrich Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, 2017.