Data Association for Multi-Object Tracking-by-Detection in Multi-Camera Networks

Michael Bredereck, Xiaoyan Jiang, Marco Körner, and Joachim Denzler
Chair for Computer Vision
Friedrich Schiller University of Jena, Germany
http://www.inf-cv.uni-jena.de
{michael.bredereck,xiaoyang.jiang,marco.koerner,joachim.denzler}@uni-jena.de

Abstract—**Multi-object tracking is still a challenging task in computer vision. We propose a robust approach to realize multi-object tracking using multi-camera networks. Detection algorithms are utilized to detect object regions with confidence scores for initialization of individual particle filters. Since data association is the key issue in Tracking-by-Detection mechanism, we present an efficient greedy matching algorithm considering multiple judgments based on likelihood functions. Furthermore, tracking in single cameras is realized by a greedy matching method. Afterwards, 3D geometry positions are obtained from the triangulation relationship between cameras. Corresponding objects are tracked in multiple cameras to take the advantages of multi-camera based tracking. Our algorithm performs online and does not need any information about the scene, no restrictions of enter-and-exit zones, no assumption of areas where objects are moving on and can be extended to any class of object tracking. Experimental results show the benefits of using multiple cameras by the higher accuracy and precision rates.**

I. Introduction

Multi-object tracking has a variety of applications in the field of computer vision, *e. g.* surveillance, motion analysis, action recognition, etc. Normally, it is quite easy and intuitive for humans to see objects or recognize their actions. However, to establish an automatic system without any intervention by humans is very challenging. Lot of work was done towards this goal. A part of the researches is based on pre-defined restrictions on the special application or scenarios. Common restrictions are the assumption of a flat world where the objects are moving on [1], [2], [3], [4], the existence of enter-and-exit zones where new objects can appear or disappear [1], [3] or a trained classifier that estimates the foreground objects beforehand [1], [4], [5], [6].

Difficulties in multi-object tracking occur when the objects are occluded for a period of time, when they are far away from the observing sensor or if their appearances are quite similar. Lacking of obtained data, single-camera based object tracking may suffer from insufficient information for robust performance. Object tracking in multi-camera systems seems to be a promising trend. With multiple cameras observing the scene, one object that is visible in several cameras can be better handled by the more complete representing model. However, additional challenges arose of how to efficiently fuse the information from multiple cameras, since one camera can not contribute to tracking all the time in the same way.

We intend to realize multi-object tracking in multi-camera systems without the stated restrictions using a general *Tracking-by-Detection*-based framework. With the output of object detectors, individual trackers can be initialized automatically. We aim to show that robust multi-object tracking in multi-camera systems is possible by solely using image information and the calibration data. Additionally, the performance of tracking can be improved in a large extend by utilizing a calibrated multi-camera system. We focus mainly on the data association algorithm through the whole system.

The remainder is organized as follows: after giving a brief overview about related work, the outline of our algorithm will be presented in Sect. I-B. Afterwards, each part of our algorithm will be discussed in Sect. II. In Sect. III, details about the experiments and quantitative analysis will be shown. Finally, we will conclude and give a brief outlook.

A. Related Work

Tracking is the process of finding the corresponding regions of the object in consecutive frames. In essence, tracking is to associate the data representing an object in each time step. Data association is evidently the key issue in tracking approaches.

Detectors, *e. g.* for localizing humans as proposed by Dalal *et al.* [7] or Felzenszwalb *et al.* [8], can provide hypotheses of discrete object regions in separate images. Normally, these detectors contain a large amount of missing detections and false positives, which increases the difficulty of detection-based data association in a large extend.

There are many researches on data association in object tracking algorithms. Some approaches utilize the evaluation of a probabilistic state space. The association might be realized by a Bayesian association as proposed by Mohedano *et al.* [2], while it can also be performed by updating particle filters that represent the tracked objects as shown by Kim *et al.* [4] or more recently by Taj *et al.* [9].

Furthermore, there are several approaches to associate data by solving underlying optimization problems. As shown by Huang *et al.* [3], association can be done hierarchically based on three association steps—*low-level*, *mid-level* and *high-level*—where a maximum a-posteriori estimation problem is solved iteratively and an Expectation-Maximization-like algorithm leads to the final output. This approach was recently used by Henrique *et al.* [5] for formulating the association task

Algorithm 1 Multi-camera multi-object tracking.

```
Input:
  • multi-camera system $\boldsymbol{C} = \{c_1, \ldots, c_M\}$
for every time step $t$ do
    $\boldsymbol{D}_{\boldsymbol{C}}^t \leftarrow \text{detections}(\boldsymbol{C}, t)$
    for every camera $c \in \boldsymbol{C}$ do
        $\tau_c^t \leftarrow \text{single-camera-tracking}(c, \boldsymbol{D}_{f_t}, \boldsymbol{\mathcal{T}}_c^{t-1})$
    end for
    $\boldsymbol{\mathcal{T}}^t = \bigcup_{c \in \boldsymbol{C}} \tau_c^t$
    $\mathfrak{T}^t \leftarrow \text{multi-camera-tracking}(\boldsymbol{C}, \boldsymbol{\mathcal{T}}^t, \mathfrak{T}^{t-1})$
end for
Output: $\mathfrak{T}^t$
```

as a global optimization problem, which bases on the mid-level association of the hierarchical association approach. Additionally, Andriyenko *et al.* [10] proposed a continuous energy minimization algorithm to obtain robust tracking results.

Matching is another approach widely used for data association. Berclaz *et al.* [1] used the output of object detectors to create *Probabilistic Occupancy Maps* (POM) to estimate the maximum number of objects and their locations, which are further used for association by graph matching. Another approach that uses matching was presented by Rudakova *et al.* [11], where the matching operates on bipartite graphs. Matching can be used directly to relate the features of an object with the most likely ones over time as presented by Andersen *et al.* [12]. The matched features can also be used for tracking with an online classifier as shown by Stalder *et al.* [13].

Traditional assignment algorithms like the *Hungarian Algorithm* introduced by Kuhn [14] can be used to find an optimal single-frame assignment [15]. However, good tracking results in single cameras have been shown in [15], where a greedy matching strategy was used for data association. It can be seen that greedy matching algorithms achieve equivalent results to the Hungarian Algorithm but at lower computational complexity. Therefore, we also perform such a greedy matching algorithm for data association.

B. Algorithm Outline

The overview of the algorithm is presented in Alg. 1. For each new frame $f_{c_j}^t$ captured by camera c_j at time step t, the previous trackers, the current image information and the calibration information of the multi-camera network are used for tracking. It starts with object detection in each camera image frame $f_{c_j}^t$ individually. Afterwards, 2D object tracking is performed to obtain a set $\boldsymbol{\mathcal{T}}_{c_j}$ of 2D trackers for each frame from camera c_j. The 2D trackers are then further associated with each other to gain global 3D trackers $\mathfrak{T}^t$ for all the cameras. Finally, the outputs are represented in 2D image space. The framework is general, without any restrictions as in other methods we mentioned before. The details of our single-camera and multi-camera tracking algorithms are presented in the following sections.

Algorithm 2 Tracking in Single Cameras.

```
Input:
  • calibrated camera $c \in \boldsymbol{C}$
  • list of detections $\boldsymbol{D}_c^t = \{d_1^t, \ldots, d_K^t\}$
  • previous list of 2D trackers $\boldsymbol{\mathcal{T}}^{t-1} = \{\tau_1^{t-1}, \ldots, \tau_L^{t-1}\}$
for every $\tau^{t-1} \in \boldsymbol{\mathcal{T}}^{t-1}$ do
    state prediction by particle filters
end for
$\boldsymbol{\mathcal{T}}^t \leftarrow \text{greedy\_matching}(\tau^{t-1}, \boldsymbol{D}_c^t)$ using $L_{\text{single}}$
for every $\tau^t \in \boldsymbol{\mathcal{T}}^t$ do
    if $\tau^t$ does not represent a new object then
        update the associated tracker and excute online classifier
    end if
end for
Output: $\boldsymbol{\mathcal{T}}^t$
```

II. Methodology

A. Tracking-by-Detection in Single Cameras

Since reliable object tracking in single camera views is a basis for object tracking in multi-camera systems, we consider this aspect first. Particle filters aim to represent hypotheses of object states by a set of particles weighted by an importance factor between the candidate model and the target model, which shows good performance in object tracking area. [16] Thus, particle filters are used in our framework to represent object trackers in single cameras. The state of an object includes the object position, velocity and size in 2D image space. Details of 2D tracking in single cameras are outlined in Alg. 2.

As shown in Alg. 3, the greedy matching algorithm we applied during single-camera Tracking-by-Detection is similar to the one proposed in [15]. The main differences are firstly, we use the associated detections to update the target model for the corresponding trackers to handle object appearance changes over time. This depends on the detections and is independent from object positions in the image. Therefore, there is no estimation or knowledge about enter-and-exit zones in the images, which increases the generality of our approach. Secondly, we reformulate the likelihood function

$$L_{\text{single}}(d,\tau) = \alpha_{\text{HSV}} \cdot L_{\text{HSV}}(d,\tau) + \alpha_{\text{pos}} \cdot L_{\text{pos}}(d,\tau) + \alpha_{\text{size}} \cdot L_{\text{size}}(d,\tau) + \alpha_{\text{class}} \cdot L_{\text{class}}(d,\tau), \tag{1}$$

for weighting the possible matches of recent trackers τ and new detections d from a single view. It is composed by three different sources which are fused similar to democratic integration [17]:

- $L_{\text{HSV}}(d,\tau)$ computes the Bhattacharyya distance of the *hue-saturation-value* (HSV) histograms of d and τ,
- $L_{\text{pos}}(d,\tau)$ evaluates the euclidean distance of the position of d and the predicted position of τ according to the last velocity,
- $L_{\text{size}}(d,\tau)$ calculates the ratio of the sizes of their bounding rectangles and
- $L_{\text{class}}(d,\tau)$ is the result of the boosted online classifier [18] of τ on detection d.

The α_* values are the corresponding weights and normalized to sum up to 1. Note that the value domain of the likelihood

Algorithm 3 Greedy matching.

```
Input:
  • bipartite graph $\boldsymbol{\mathcal{G}} = (\boldsymbol{V}, \boldsymbol{E})$
  • vertices $\boldsymbol{V} = \boldsymbol{V}_1 \cup \boldsymbol{V}_2$, $|\boldsymbol{V}_1| = n$, $|\boldsymbol{V}_2| = m$
  • $\nexists \boldsymbol{e} = (v_i, v_j) \in \boldsymbol{E} : v_i \in \boldsymbol{V}_1 \wedge_j \in \boldsymbol{V}_2$
define weighting matrix $\boldsymbol{M} \in \mathbb{R}^{n \times m}$
$m_{i,j} := L(v_i, v_j), \quad v_i \in \boldsymbol{V}_1,\ v_j \in \boldsymbol{V}_2$
Sort $\boldsymbol{M}$ descendingly
while $(\text{rows}(\boldsymbol{M}) > 0)$ do
    $(i^*, j^*) = \text{position}(\text{max-element}(M))$
    link $v_{i^*}, v_{j^*}$, delete row $i^*$, delete column $j^*$
end while
```

is $[0, 1]$, where $L(d, \tau) = 0$ if the considered pair is not a corresponding match and $L(d, \tau) = 1$ if it is a definitely matching pair.

After matching current detections with previous trackers, the object models of particle filters are updated by the associated detections and the online classifier is extended by the newly associated detections meanwhile as shown in Fig. 2.

The result of object Tracking-by-Detection based on single camera views is a set of 2D trackers of all cameras. To increase the robustness and perform accurate tracking especially during occlusions, we associate the trackers from different cameras to have global efficiency by estimating the 3D positions of the objects.

B. Data Association in Multi-Camera Systems

For the sake of computation costs, we only consider the trackers from single-camera tracking from all cameras instead of operating on all camera images directly. The aim is to associate each tracker from one camera with up to one tracker from other cameras. We use the greedy matching algorithm introduced in the previous section for associating the trackers accross multiple cameras. Our proposed algorithm for object tracking in multi-camera systems based on object tracking in single cameras is firmly drafted in Alg. 4.

Assume that we have a set of previous 3D trackers $\mathfrak{T}^{t-1}$. For each 3D tracker $T^{t-1} \in \mathfrak{T}^{t-1}$, there is a set of previously assigned 2D trackers $\boldsymbol{\mathcal{T}}^{t-1}$ with up to one tracker per camera. If at least two trackers $\tau_i, \tau_j \in \boldsymbol{\mathcal{T}}^{t-1}$ have been updated during single-camera tracking, T^t in 3D space can be updated by estimating the 3D position

$$\boldsymbol{p}_{T^t} = \frac{\sum\limits_{\tau_i, \tau_j \in \boldsymbol{\mathcal{T}}^t, \tau_i \neq \tau_j} \text{triangulate}\left(\tau_i, \tau_j\right)}{\text{card}\left(\boldsymbol{\mathcal{T}}^t\right)} \tag{2}$$

which is the centroid of all the 3D positions from assigned tracker pairs obtained by the triangulation relationship between two cameras [19]. Afterwards, we back-project the 3D positions of each 3D tracker T^t into each camera c that has no 2D tracker assigned to it. If this projection is within a bounding region of a 2D tracker τ_j, the minimal likelihood L_{min} of τ_j and the 2D trackers that are assigned to T^t is computed. Finally, τ_j is assigned to T^t if $L_{\text{min}} > \theta_{\text{bp}}$ holds. Therefore, lost 2D trackers are re-assigned to T^t, while new 2D trackers in cameras that could not observe the objects before are newly assigned to T^t. Each updated 3D tracker T^t is added to the current set of trackers $\mathfrak{T}^t$ and the assigned 2D trackers are not considered in the following matching process.

Algorithm 4 Tracking in multi-camera systems.

```
Input:
  • multi-camera system $\boldsymbol{C} = \{c_1, \ldots, c_M\}$
  • list of all 2d-trackers in each camera $\boldsymbol{\mathcal{T}}^t_{\boldsymbol{C}} = \{\boldsymbol{\mathcal{T}}^t_{c_1}, \ldots, \boldsymbol{\mathcal{T}}^t_{c_M}\}$
  • previous list of 3d-trackers $\mathfrak{T}^{t-1}$
for $T^{t-1} \in \mathfrak{T}^{t-1}$ do
    if $(\forall \tau \in \boldsymbol{\mathcal{T}}^t_{c_i} \in \boldsymbol{\mathcal{T}}^t_{\boldsymbol{C}}$ : tracker $\tau$ was assigned to $T^{t-1})$ then
        update $T^{t-1}$ and delete $\tau$ from $\boldsymbol{\mathcal{T}}^t_{c_i}$
    else
        delete $T^{t-1}$ from $\mathfrak{T}^{t-1}$
    end if
end for
$\mathfrak{T}^t \leftarrow \mathfrak{T}^{t-1} \cup \text{greedy-matching}(\boldsymbol{\mathcal{T}}^t_{\boldsymbol{C}}, \boldsymbol{\mathcal{T}}^t_{\boldsymbol{C}})$ using $L_{\text{multi}}$
Output: $\mathfrak{T}^t$
```

Afterwards, the unassigned 2D trackers from each camera are associated to new 3D trackers by the greedy matching algorithm. The assignment between the two 2D trackers τ_i and τ_j depends on the likelihood function

$$L_{\text{multi}}(\tau_i, \tau_j) = \begin{cases} \alpha_{\text{epi}} \cdot L_{\text{epi}}(\tau_i, \tau_j) + \\ \alpha_{\text{HSV}} \cdot L_{\text{HSV}}(\tau_i, \tau_j) & c_k \neq c_l \\ 0 & c_k = c_l \end{cases}, \tag{3}$$

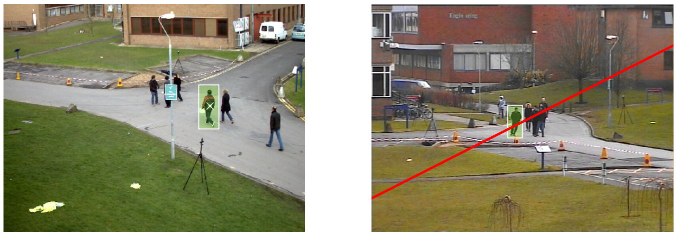
where τ_i is a tracker from camera c_k, τ_j is a tracker from camera c_l and the α_* values are the corresponding weights. Here, the atomic likelihood functions are defined as

- $L_{\text{epi}}(\tau_i, \tau_j)$ evaluates the Euclidean distance of τ_i to the epipolar line induced by τ_j,
- $L_{\text{HSV}}(\tau_i, \tau_j)$ computes the Bhattacharyya distance between the HSV color histogram of τ_i and τ_j.

Again, the value domain of these functions is normalized to $[0, 1]$, where $L_{\text{multi}}(\tau_i, \tau_j) = 0$ if τ_i and τ_j do not correspond and $L_{\text{multi}}(\tau_i, \tau_j) = 1$ if the correspondence of τ_i and τ_j is evident. A 2D tracker pair $\varkappa_{i,j} = (\tau_i, \tau_j)$ is only matched if its $L_{\text{multi}}(\tau_i, \tau_j) > \theta_{\text{min}}$.

The accuracy of camera calibration has a high impact on the likelihood function L_{epi}, since it depends on the epipolar geometry between two cameras. The more reliable the calibration is, the higher the influence of L_{epi} in L_{multi} might be. We also consider situations where multiple objects are near to one corresponding epipolar line of an object in another camera, as presented in Fig. 1. The use of such a HSV histogram based likelihood function L_{HSV} is promising to evaluate the likelihood of the two trackers that may be matched, which has shown a good performance in our experiments.

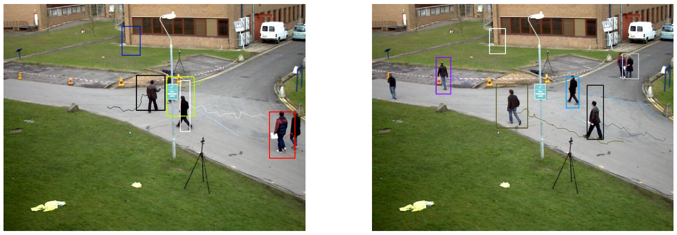
Therefore, for each matched pair of 2D trackers $\varkappa^t_{i,j} = (\tau_i, \tau_j)$, we define a new 3D tracker T^t and assign τ_i and τ_j to it. Afterwards, we project the 3D position of T^t back into the cameras that have no 2D trackers assigned to T^t in order to add missing 2D trackers if possible. Each T^t is finally added to the current set $\mathfrak{T}^t$. The process iterates over every frame in the sequence.



(a) source image (view 001)

(b) corresponding image (view 003)

Figure 1: Multiple candidates near the red epipolar line in (b) for the query object (white rectangle in (a)).



(a) Frame 020

(b) Frame 258

Figure 2: Examplary results for 2D tracking with online-learnt re-classification of the tracking targets.

Table I: Evaluation results of single-camera object tracking

Parameter	Evaluation						
α_{class}	MOTA	MOTP	FP	IDS	FN	PRE	REC
0.0	0.78	0.88	123	125	782	0.97	0.83
0.1	0.80	0.85	421	142	388	0.91	0.92

Table II: Evaluation of tracking in multi-camera systems.

Detector	Evaluation						
	MOTA	MOTP	FP	IDS	FN	PRE	REC
HOG	0.65	0.82	1088	121	416	0.80	0.91
PBM	0.75	0.85	201	106	858	0.95	0.82
HOG+PBM	0.61	0.86	1448	216	168	0.76	0.96

III. Experiments

A. Dataset

Since there are not many suitable datasets for multi-object tracking in multi-camera systems and PETS'09 [20] dataset is sufficient and challenging, we tested our algorithm on the this dataset for evaluation. PETS'09 contains many scenes with occlusions and variations in the camera images, such as the distance of the objects to the cameras, illuminations in the images and the camera resolutions. Furthermore, this dataset has been used to evaluate different state-of-the-art approaches, which enables us to compare our method with them.

PETS'09 offers three types of applications: *person counting and density estimation*, *people tracking* and *flow analysis and event recognition*. We use the scene `S2.L1 walking` that includes 795 frames and about $7\,{}^{f}\!/{}_{s}$ for each of the 7 cameras. Together with the images, PETS'09 offers the camera calibration parameters. Since they did not offer ground truth data, we use the manually labeled ground truth data for the first camera provided by [10].

B. Evaluation Metric

We utlize the CLEAR MOT metrics [21] for evaluating the tracking results. They define the *multi-object tracking accuracy* (MOTA) to consider the number of *false positives* (FP), *false negatives/missing detections* (FN) and *identity switches* (IDS). Besides, *multi-object tracking precision* (MOTP) is used for evaluating the precision of localization of targets. For further details about the evaluation parameters we refer to [21]. The evaluation program is provided by [10].

C. Quantitative Analysis

For all following experiments, we directly use the camera calibration data provided by PETS'09 dataset. We do not implement any preprocessing, such as learning the foreground of the expected scene or training the detectors specially for the used data. We use *Histograms of Oriented Gradients* (HOG) [7] and *Part-Based Appearance Models* (PBM) [8] as human body detectors. Parameters used in the experiments are $\alpha_{\text{HSV}} = 0.1$, $\alpha_{\text{pos}} = 0.4$, $\alpha_{\text{size}} = 0.2$ for L_{single} $\alpha_{\text{HSV}} = 0.25$, $\alpha_{\text{epi}} = 0.75$ for L_{multi}.

We start with the evaluation of the implemented single-camera Tracking-by-Detection as it is the basis for the whole object tracking system. To figure out the influence of the online classifier, we evaluate the tracking results with and without using classifiers as presented in Tab. I.

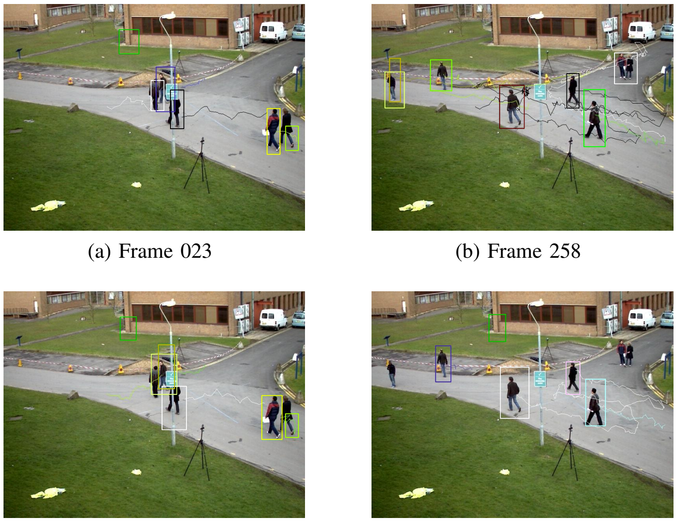
As one can see, the single-camera multi-object tracking is robust. However, there are a lot of id switches. One reason for the high number of missed objects in single-camera tracking is that there is no handling of occlusion. Furthermore, the usage of online classifiers decreases the probability that a tracker loses its target. Therefore, the number of missing objects (FN) increases $(+102\%)$ when we track without using online classifiers. In addition, MOTP is higher probably because there are more situations when an object is assigned to a newly created tracker, while improve MOTA as the trackers renew the localization of the targets.

Tab. II presents the results of object tracking in multi-camera systems, while several intuitive results are shown in Fig. 3. Occlusions are handled relying on the assumption that the occluded objects in some views are not occluded in at least two other cameras. To show the strong dependency of the performance of the detectors, we also evaluate the results using different detectors separately.

Note that the combination of both detectors at the same time reduces the number of missed objects a lot, since there are long periods with almost no detections using one individual detector. However, it suffers from a higher number of false alarms. An additional handling of false positive trackers could increase the overall performance. Furthermore, we can see

Table III: Result of multi-camera multi-object tracking, where 2D trackers are added, if not assigned to a 3D tracker.

	Evaluation						
	MOTA	MOTP	FP	IDS	FN	PRE	REC
without fusion	0.57	0.81	1600	154	249	0.73	0.95
with fusion	0.49	0.85	2119	186	71	0.75	0.97



(a) Frame 023

(b) Frame 258

(c) Frame 023

(d) Frame 258

Figure 3: Exemplary results for 3D tracking: (a) and (b) without re-classification, (c) and (d) with re-classification

that MOTA increases when using the PBM detector since it decreases the number of false alarms, although there are more missing objects.

The Location of the objects can be obtained by 2D trackers or the back projected positions from the 3D trackers. We evaluate the results from 3D trackers with and without fusing the tracking results from 2D trackers as presented in in Tab. III. It can be seen that the combination of the two trackers reduces the number of missing objects by (-71.5%). Simultaneously, the number of false alarms $(+32.4\%)$ and id switches $(+21.8\%)$ increase because there are also false 2D trackers added to 3D trackers.

It still remains to be seen how robust our approach is in comparison to other approaches.

D. Comparison

As we might use ground truth data different from those that are used by other approaches, the results of our experiments can not be directly compared to them. However, since the data is similar, the comparison presented in Tab. IV at least offers tendencies for the robustness of different approaches.

The approach of Henriques *et al.* [5] has the highest MOTA, percentage of detected and percentage of correctly detected objects. A reason for the high accuracy could be the usage of an optimization method for tracking over time. Furthermore, optimization increases the percentage of correctly detected objects as they only need to consider objects that can be tracked for some time steps. However, their approach is only applicable to offline tracking scenarios, while our approach could also be used online. According to the accuracy parameters, both our approaches are competitive with other single-camera based tracking approaches like [15]. We can outperform others regarding the value of MOTP, which may be influenced by the usage of detectors and particle filters. Compared to the single-camera tracking by continuous energy minimization [10], we see that our approach has a higher MOTP but a lower MOTA value. Their accuracy might benefit from optimization algorithms, which reduces the number of id switches. They may also suffer from lower accuracy by not using detections for object localization. Besides, our approach suffers from a higher number of id switches.

Table IV: Evaluation results of our proposed multi-object tracking methods compared to other approaches.

Approach	Evaluation				
	MOTA	MOTP	IDS	Prec	Rec
Henriques *et al.* [5] *(offline)*	**0.966**	*n/a*	**10**	**0.985**	**0.986**
Berclaz *et al.* [1]	0.760	0.630	*n/a*	*n/a*	*n/a*
Andriyenko *et al.* [10]	0.814	0.761	15	*n/a*	*n/a*
Breitenstein *et al.* [15]	0.797	0.563	*n/a*	*n/a*	*n/a*
Yang *et al.* [22]	0.759	0.538	*n/a*	*n/a*	*n/a*
our single-camera approach	0.795	0.846	142	0.910	0.916
our multi-camera approach	0.749	**0.854**	106	0.949	0.815

We also compare our approaches with multi-camera based methods, such as Berclaz *et al.* [1] and Yang *et al.* [22]. Their algorithms have lower MOTPs in PETS'09 than ours, while we localize the objects more accurate using detectors and particle filters than their optimization on segmentation-based POMs. Furthermore, both approaches have higher accuracy (MOTA) than our approach. One reason for this might be the high number of false alarms in our experiments, as mentioned before.

E. Run-time Performance

The system is implemented in C++ using the classification framework of Stalder *et al.* [13] and the open-source library *OpenCV*. All experiments implemented on one single core of a Intel Core2Quad$^{\text{TM}}$ CPU with 2.4 GHz and 8 GB of memory. In average, the runtime for the first 50 frames in the first camera is $2.98\,{}^{s}\!/\!_{f}$ with and $2.52\,{}^{s}\!/\!_{f}$ without using online classifiers. If we consider when new trackers are created and online classifiers are trained, the runtime for the single-camera tracking increases to $8.2\,{}^{s}\!/\!_{f}$. The average runtime of the multi-camera multi-object tracking system is $38.0\,{}^{s}\!/\!_{f}$ with and $36.6\,{}^{s}\!/\!_{f}$ without the usage of online classifiers in single-camera tracking. Note that the system performs tracking in each camera without any parallelism. In addition to that, the calculation of the likelihood values is not using parallel computing as well. Therefore, the runtime could decrease a lot by doing the single-camera tracking of all cameras simultaneously and by parallel computing of the likelihood values for the multi-camera tracking. However, the runtime of the single-camera approach of [15], $0.5 - 2.5\,{}^{s}\!/\!_{f}$ without considering the time for detection, is lower than ours. Unfortunately, Henriques *et al.* [5] did not report runtimes for their offline multi-camera approach.

IV. Conclusion

A. Summary

We presented an approach for multi-object tracking in multi-camera systems in this paper. In contrast to our proposed method, many multi-object tracking approaches rely on a variety of constraints, which reduces their applicability. A common restriction is the assumption, that all the tracked objects are moving on a flat ground plane. Approaches that use this restriction could fail on scenarios where objects can move on uneven terrain, for example when tracking people in multi-decked supermarkets. Some other approaches use a foreground segmentation for detecting objects, which requires a trained classifier and special knowledge about the expected scene. Besides this, some approaches utilize known enter and exit zones in the scene to fix problems with suddenly appearing or disappearing objects, which could fail when tracking objects in a wide area.

However, we introduced an object-tracking framework without using these restrictions to realize robust multi-object tracking in multi-camera systems. It bases on a single-camera Tracking-by-Detection algorithm by associating data according to a greedy matching algorithm in each camera. This reduces the complexity while its expected results are comparable with the Hungarian Algorithm that finds an optimal matching. Furthermore, the matching-based algorithm can be easily extended by defining likelihood functions that represent relationships between the objects to be matched. It was further extended by a likelihood function that evaluates the distance to the epipolar line of one object in the camera of another object. Finally, we evaluated our proposed method and compared it to other state-of-the-art approaches. We showed that our single-camera tracking can compete with the approach of [15], which also uses a data association technique based on greedy matching. We see that our approach, especially in the case of multi-camera tracking, suffers from a high number of false alarms, which could be caused by illumination changes of the images from different cameras. Hence, the multi-object tracking accuracy of our approach is worse compared to some other multi-camera tracking approaches, such as [1], [22]. However, the multi-object tracking precision, the percentage of detected objects and the percentage of correctly detected objects of our approach is high. Considering object tracking precision, we can even outperform the mentioned tracking approaches. Most importantly, our framework is general and can easily be extended to other applications.

B. Outlook

It remains an open question, whether our approach could run in realtime by parallely runing the single-camera tracking and the calculation of the likelihood functions for matching between different cameras. Besides this, it is promising to extend our approach by an EM-like strategy to estimate the best set of parameters, such as the weights for the used likelihood functions. Furthermore, as our approach does not use any restrictions beside the camera calibration information, it would be interesting to compare the results to other approaches on a dataset where objects can move up and down. More robust algorithm during occlusion would be a further research direction for us. Additionally, we plan to record such dataset for testing since there are currently no other suitable datasets for evaluating multi-object tracking approaches in such scenarios.

References

[1] J. Berclaz, E. Tretken, F. Fleuret, and P. Fua, "Multiple Object Tracking using K-Shortest Paths Optimization," *TPAMI*, vol. 33, pp. 1–16, 2011.

[2] R. Mohedano and N. Garcia, "Robust Multi-Camera 3D Tracking from Mono-Camera 2D Tracking using Bayesian Association," *IEEE Trans. on Consumer Electronics*, vol. 56, pp. 1–8, 2010.

[3] C. Huang, B. Wu, and R. Nevatia, "Robust Object Tracking by Hierarchical Association of Detection Responses," in *ECCV*, 2008, pp. 788–801.

[4] K. Kim and L. S. Davis, "Multi-Camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering," *ECCV*, vol. 3953, pp. 98–109, 2006.

[5] J. F. Henriques, R. Caseiro, and J. Batista, "Globally Optimal Solution to Multi-Object Tracking with Merged Measurements," *ICCV*, pp. 2470–2477, 2011.

[6] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multi-Camera People Tracking with a Probabilistic Occupancy Map," *TPAMI*, vol. 30, pp. 267–282, 2008.

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *CVPR*, 2005, pp. 886–893.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, pp. 1627–1645, 2010.

[9] M. Taj and A. Cavallaro, "Distributed and Decentralized Multicamera Tracking," *SPMag*, vol. 28, no. 3, pp. 46–58, 2011.

[10] A. Andriyenko and K. Schindler, "Multi-target Tracking by Continuous Energy Minimization," *CVPR*, pp. 1265–1272, 2011.

[11] V. Rudakova, "Probabilistic framework for multi-target tracking using multi-camera: applied to fall detection," Master's thesis, Gjøvik University, 2010.

[12] M. Andersen and R. S. Andersen, "Multi-Camera Person Tracking using Particle Filters based on Foreground Estimation and Feature Points," Master's thesis, Aalborg University, 2010.

[13] S. Stalder, H. Grabner, and L. V. Gool, "BeyondSemi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition," in *OLCV*, 2009.

[14] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, pp. 83–97, 1955.

[15] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, "Online Multi-Person Tracking-by-Detection from a Single, Uncalibrated Camera," *TPAMI*, vol. 33, pp. 1–14, 2010.

[16] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29(1), pp. 5–28, 1998.

[17] D. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6 –23, 1997.

[18] H. Grabner and H. Bischof, "Online boosting and vision," in *CVPR*, 2006, pp. 260–267.

[19] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2006.

[20] Winter-PETS 2009 Workshop, http://www.cvg.rdg.ac.uk/WINTERPETS09.

[21] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics," *IVP*, pp. 1–10, 2008.

[22] J. Yang, Z. Shi, P. Vela, and J. Teizer, "Probabilistic multiple people tracking through complex situations," *PETS*, 2009.