Computer Vision and Image Understanding 128 (2014) 36–50

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Joint multi-person detection and tracking from overlapping cameras [☆]

Martijn C. Liem [a], Dariu M. Gavrila [a,b,*]

[a] Intelligent Autonomous Systems Group, Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
[b] Environment Perception Department, Daimler Research and Development, Wilhelm Runge St. 11, 89081 Ulm, Germany

ARTICLE INFO

Article history:
Received 21 June 2013
Accepted 6 June 2014
Available online 23 June 2014

Keywords:
Person tracking
Person detection
Overlapping cameras

ABSTRACT

We present a system to track the positions of multiple persons in a scene from overlapping cameras. The distinguishing aspect of our method is a novel, two-step approach that jointly estimates person position and track assignment. The proposed approach keeps solving the assignment problem tractable, while taking into account how different assignments influence feature measurement. In a hypothesis generation stage, the similarity between a person at a particular position and an active track is based on a subset of cues (appearance, motion) that are guaranteed observable in the camera views. This allows for efficient computation of the *K*-best joint estimates for person position and track assignment under an approximation of the likelihood function. In a subsequent hypothesis verification stage, the known person positions associated with these *K*-best solutions are used to define a larger set of actually visible cues, which enables a re-ranking of the found assignments using the full likelihood function.

We demonstrate that our system outperforms the state-of-the-art on four challenging multi-person datasets (indoor and outdoor), involving 3–5 overlapping cameras and up to 23 persons simultaneously. Two of these datasets are novel: we make the associated images and annotations public to facilitate benchmarking.

1. Introduction

The ability to visually track persons across a scene is important for many application domains, such as surveillance or sports analysis. In this paper, we are interested in the more challenging scenarios involving multiple persons in complex environments (i.e. dynamic and cluttered backgrounds, varying lighting conditions). Multiple view analysis allows to compensate for the effects of occlusions and noisy observations. Cost and logistics considerations will, however, often limit the number of overlapping cameras that can be employed. We aim for methods that work with as few as 3–4 cameras from very different, diagonal downward viewing directions with overlap (as opposed to ceiling-mounted cameras with a bird-eyes view). These conditions could arise in a sports stadium (e.g. football, basketball), the main lobby of a building (e.g. bank, government) or at a critical infrastructure (e.g. train station or airport hall). Our cameras are synchronized and calibrated offline, but several previous work on self-calibration [1,2] exists that could relax this assumption.

The considered wide-baseline camera set-up makes it difficult to establish individual feature correspondences across views, especially in the presence of sizable inter-person occlusion. We aim for robustness by performing the analysis based on a 3D scene reconstruction, namely, using the visual hulls of objects obtained by volume carving. The main challenge is thus to establish correct correspondence across views at the object level. Matching different objects together across multiple views leads to erroneous 3D objects, so-called ghosts, see Fig. 1. We will address this challenge with a novel, two-step likelihood optimization approach; this is part of a recursive tracker that is meant for online analysis.

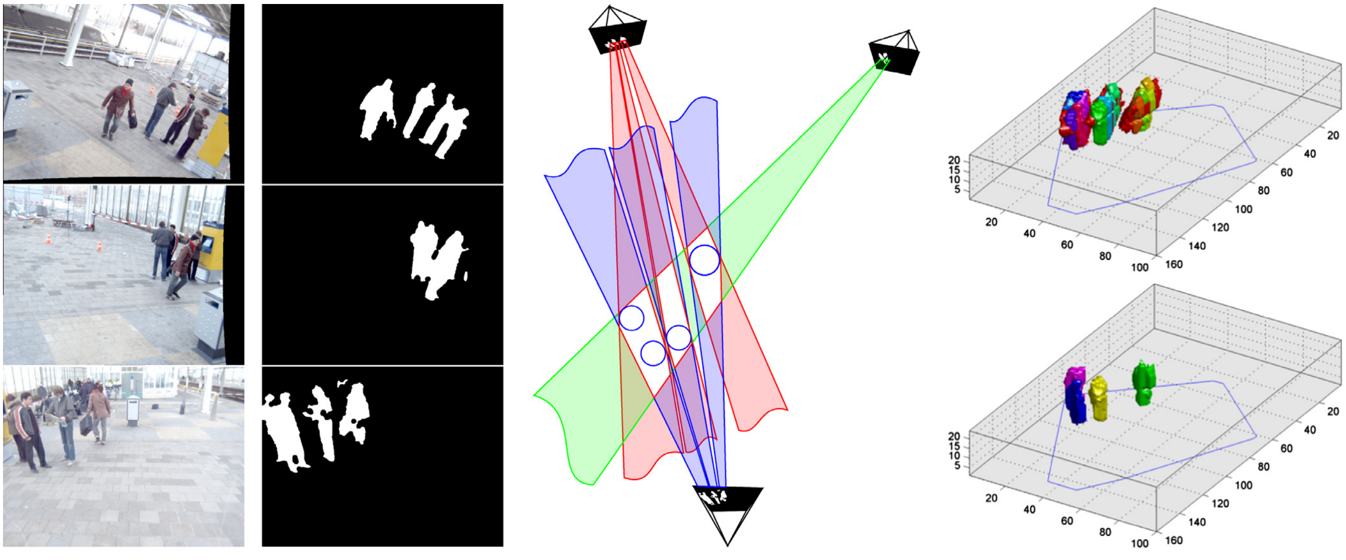
The remainder of this paper is organized as follows. Section 2 covers the most closely related work. Section 3 provides an overview of the proposed approach, while Section 4 contains the technical details. In Section 5, we present the experimental results. We discuss the chosen camera set-up and computational issues in Section 6. The conclusions and suggestions for future work are listed in Section 7.

2. Related work

Extensive research has been performed in the area of person detection and tracking. In this section, we will give an overview of the work most relevant to our paper.

[☆] This paper has been recommended for acceptance by Jordi Gonzàlez.
[*] Corresponding author at: Intelligent Autonomous Systems Group, Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands.
E-mail addresses: martijn@liem.nu (M.C. Liem), d.m.gavrila@uva.nl (D.M. Gavrila).

http://dx.doi.org/10.1016/j.cviu.2014.06.003
1077-3142/© 2014 Elsevier Inc. All rights reserved.



(a) (b) (c) (d)

Fig. 1. (a) Recorded images. (b) Corresponding foreground segmentation images. (c) Volume carving projects foreground masks generated at each camera view into a 3D space, 'carving out' potential object positions (red bounded, white areas). Because of incorrect correspondences, extra volumes are carved out, so-called 'ghosts' (artifacts). Actual person positions are shown as blue circles. (d) (top) Volume carving result (including artifacts), segmented into person hypotheses (colored objects). (bottom) Detected persons. The blue lines represent the area perceived by all cameras. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Much research on person detection and tracking focuses on the use of a single camera view [3–10]. In recent work, pre-trained person detectors are often used to generate detections [3,6,7]. Benfold and Reid [3] use a HOG based head detector to detect heads from a bird's-eye-view camera perspective and extrapolate full body detections using a fixed ground plane. Using multiple instance learning, Yang and Nevatia [6] learn appearance models from person detections making up possibly connected as well as mutually exclusive tracks. Andriyenko et al. [7] combine HOG based person detections with a Gaussian per detection to reason about occlusions and track people through occlusions. The method presented by Leibe et al. [4] uses quadratic boolean programming to solve detection and tracking in coupled manner. Multiple tracking hypotheses are kept and the most likely trajectories are found searching forward as well as backwards in time. Wu et al. [5] perform coupled detection and tracking using graph based flow optimization. Detections are generated using a template based generative model on a discretized ground plane.

Handling occlusions from one perspective is difficult and is often solved by extrapolating tracking results, which is error prone. While single view tracking methods can offer good results, using more cameras makes tracking more robust for more complex and crowded scenes. Furthermore, using multiple views allows modeling persons' appearance from all sides.

Multi-camera person detection has been approached in several ways. Mittal and Davis [11] match colors in different views along epipolar lines to determine the position of people on the ground plane. Occlusions are modeled by learning person presence likelihood maps at these locations. Eshel and Mozes [12] project multiple camera views onto a horizontal plane located at head-hight in the 3D space, comparing pixel values of different views at the same projected location to locate persons' heads. Foreground images are projected onto multiple horizontal planes in the 3D space by Khan and Shah [13] and Arsić et al. [14], detecting objects at ground plane locations where multiple foreground regions intersect in multiple planes. Similarly, Santos and Morimoto [15] use images containing the number of foreground pixels above each pixel to create 3D detections at positions with the highest accumulated score.

Fleuret et al. [16] present a Probabilistic Occupancy Map (POM) for person detection. A generative model using a discretized ground plane and fixed size regions of interest approximates the marginal probability of occupancy by accumulating all evidence received from foreground images from every camera. A similar approach using information acquired by a person detector instead of only using foreground information is presented by Berclaz et al. [17]. Huang and Wang [18] propose a model in which multiple volume carving based scene configuration hypotheses are evaluated. Instead of solving hypothesis selection in 3D, the graph cut algorithm is used to label the pixels of each camera image as background or one of the people in the scene. An iterative model labeling individual voxels of a volume reconstruction as either part of an object, background or static occluder is presented by Guan et al. [19]. Otsuka and Mukawa [20] determine 2D visual angles as seen from the top-down, corresponding to the segmented objects in each camera.

Detections can be combined into long-term tracks in several ways. Methods either use a *recursive* tracking approach [8,9,21,22], useful for real-time applications, or do tracking in *batch mode* [23,24,16,25,26] using a buffer of frames.

Recursive trackers perform on-line tracking on a frame-to-frame basis, often using well known algorithms like Mean-Shift [9], Kalman filtering [18,14] or particle filtering [8]. Calderara et al. [27] and Hu et al. [29] both perform tracking and detection in individual camera views and match persons' principal axis between cameras to consistently label persons across cameras. While Hu et al. [29] use a standard Kalman filter for tracking, Calderara et al. [27] take a tracking-by-detection approach based on foreground segmentations and learned appearance models. When tracking multiple objects simultaneously, the issue of consistently assigning tracks to detections should be solved. Well known solutions are the Joint Probabilistic Data Association Filter [33,30] and Multiple Hypotheses Tracking [34]. A JPDA tracker using appearance, 2D and 3D person positions is presented by Kang et al. [30]. Huang and Russell [35] formulate the track assignment problem as a bipartite matching problem. They apply the Hungarian algorithm to solve the assignment problem, using an association matrix to enumerate all possible assignments. The method is applied to vehicle re-identification using non-overlapping monocular cameras and constrained motion patterns (lanes on a highway), which is significantly different from the context of this paper.

Particle filters have also been extended for multi-target tracking, for example combined with the appearance model from [11] and the projection of people's principal axis onto the ground plane by Kim and Davis [31]. Du and Piater [28] combine multiple particle filters for each camera as well as on the ground plane to track people. Otsuka and Mukawa [20] keep multiple hypotheses for the regions in the scene where a person can be. Position estimation within these regions is done using particle filtering. Possegger et al. [32] compute a volumetric scene reconstruction to generate detections and use particle filtering for tracking. The complexity of the assignment problem is reduced by partitioning the detections using Voronoi partitioning, making the object states independent. Appearance cues are used to distinguish objects in close proximity of each other, after all person locations have been identified. The track creation policy is based on strict entry and exit areas where tracks can be created and deleted.

Since this paper focuses on recursive tracking, the most relevant approaches doing recursive tracking using multiple cameras are highlighted in Table 1. This table also lists whether the approaches treat localization and tracking separately or if they combine the two. In the first case, persons' positions are determined during the localization step while the tracking step connects all locations over time to form tracks [14,27–31,11]. In the second case, localization provides just hypotheses for persons' positions. This information is combined with tracking information from previous time steps to resolve which hypotheses are actual persons and which are false positives [20,32]. This last approach is referred to as joint localization and tracking and is the approach taken in this paper.

Batch mode trackers optimize detection to track assignment over a set of multiple frames together, looking both forward and backward in time. Tracking is often modeled as a linear or integer programming problem or as an equivalent graph traversal problem. Flow optimization is used for tracking by Zhang et al. [26], finding disjoint paths in a cost flow network defined using observation likelihoods and transition probabilities. Berclaz et al. [24] apply standard flow optimization techniques to do tracking in a graph created by stacking POMs for multiple time steps. For comparison, this method (used as a baseline in the experiments) is added to Table 1. This method is extended with an appearance model by Ben Shitrit et al. [23]. Detections created by Khan and Shah [13] are combined into tracks based on their positions and a graph cut segmentation method is used to find individual trajectories in a tracking sequence. Leal-Taixe et al. [36] solve tracking and detection jointly by simultaneously optimizing the flow through graphs representing detections per view as well as graphs representing 3D detections reconstructed from combined camera pairs. In comparison, Hofmann et al. [37] use flow minimization on a single global graph to perform tracking as well as 3D position reconstruction based on one or more views. Vertices describe 3D positions based on different combinations of 2D detections and edges connect these positions over time.

Finally, hybrid tracking approaches based on tracklets [3,6,7,12,38] combine both types of methods. First, recursive methods connect detections to short consistent tracklets, after which batch methods combine these to form long term tracks. Benfold and Reid [3] generate short stable tracklets using KLT features to track their detections over time and use a sliding window to match and combine these into tracks. Andriyenko et al. [7] use a non-convex energy optimization scheme to connect tracklets created from connecting person detection results. Yang and Nevatia [6] learn non-linear motion patterns from tracklets created using a person detector and connect tracklets according to the learned patterns. Baltieri et al. [38] create short term tracks by Kalman filtering detections created using a 3D marked point process model. An appearance model, sampled using a manually created 3D person model, is used to combine corresponding tracklets into more consistent long-term tracks.

3. Overview

There are three main aspects to multi-person, multi-camera detection and tracking: finding correspondences between different views, corresponding detections to tracks and determining false positives. This paper's main contribution is a two-step likelihood optimization approach that solves all these aspects jointly while considering information on object locations, foreground

Table 1
Overview of recursive person localization and tracking methods. For comparison, the batch mode method from [24] that is used as a baseline in our experiments is included as an additional entry. (CA: number of cameras, NP: number of people in one group, App: uses appearance model for tracking).

Method	CA	NP	Localization	Track assignment	Tracking	App
Arsić et al. [14]	4	5	Foreground segmentation, multi-plane homography, false positive reduction	Quadratic programming: position, appearance	Kalman filter	Yes
Calderara et al. [27]	4	3	Homography, epipolar constraints	Position, appearance	Tracking by detection	Yes
Du and Piater [28]	2–4	2	Principal axis, homography	Position	Multiple particle filters	No
Hu et al. [29]	2–3	4	Foreground segmentation, principal axis, homography	Position	Kalman filter	No
Kang et al. [30]	2	5	Foreground segmentation, homography	JPDA: 2D and 3D position, appearance	Kalman filter	Yes
Kim and Davis [31]	4	4	Foreground segmentation principal axis, homography	Multi-hypotheses, position, appearance	Particle filter	Yes
Mittal and Davis [11]	4–16	3–6	Color matching of epipole segments	Position, velocity	Kalman filter	Yes
Otsuka and Mukawa [20]	6	5	Joint localization and tracking: 2D detections based on visual angles, no appearance, no track creation/deletion, measurements independent, multiple occlusion hypotheses, recursive bayesian estimation, position likelihood		Particle filter	No
Proposed method	**3–5**	**2–23**	**Joint localization and tracking: 3D detections, appearance in objective function, entry/exit likelihood map occlusions make assignments dependent, graph assignment problem, position likelihood, foreground likelihood**		**Kalman filter**	**Yes**
Possegger et al. [32]	4–5	4–12	Joint localization and tracking: 3D detections, close proximity appearance after detection, fixed entry/exit areas, assignments conditionally independent, Voronoi partitioning of hypotheses space, position likelihood		Particle filter	Yes
Berclaz et al. [24]	1–5	9–10	Joint localization and tracking: Probabilistic Occupancy Map (POM) [16] detections, no appearance, entry/exit areas at a.o.i. edges, joint optimization of all trajectories using KSP flow optimization in a graph of stacked POMs		Batch mode flow optimization	No

The method proposed in this paper is highlighted in bold.

segmentations and appearances. The task of identifying detected objects as persons and tracking them is formulated as an assignment problem in a bipartite graph. Hypotheses for correlating 2D detections across views are created using volume reconstruction, forming detections in the 3D space. By associating these detections with existing tracks, the proposed method determines whether or not detections are actual persons and thereby whether correlation hypotheses are correct. Occlusions and the uncertainty about which detections are persons and which ones are ghosts make determining which features belong to which object ambiguous and dependent on all assignments made. This is for example the case when taking appearances into account, and makes determining the most likely assignment hypothesis intractable. The proposed two-step approach offers an approximate solution to this problem. Methods like [27,11] only take into account dependencies when correlating 2D detections across views, but do not combine this with the aspect of detection and tracking.

The different viewpoints enable modeling persons' appearances from all sides simultaneously, while the use of occlusion information from the volume reconstruction allows individual appearances to be extracted with minimum pollution from other objects' appearances. Other methods taking a joint approach to multi-camera detection and tracking either do not take appearance into account [20], or only use appearance to resolve conflicting track assignments (i.e. not for detection) [32]. Our joint approach allows the use of appearance information for all aspects of detection and tracking, making it an integral part of the objective function and using it for track continuity as well as to distinguish persons from ghosts. It thus incorporates available information as early as possible. This is not possible when treating different aspects independently. Entry and exit regions are modeled using a combination of a likelihood map (Fig. 5(b)) and the foreground likelihood (Section 4.1.2), offering high flexibility in the creation and deletion of tracks.

Fig. 2 provides an overview of the proposed method for one time step. The volume reconstruction is segmented into individual objects using an *EM* clustering approach discussed in Section 4. The approximation of the objective function used in the *preselection* step is based on a subset of the features, of which the measurement is independent of the assignment hypothesis. *All* objects are projected to the camera planes to compute occlusions and appearances, while foreground likelihood is computed using Kalman filtered person position predictions (Section 4.2). Since the *K preselection* hypotheses are ranked using the approximated objective function, a *verification* step using the full objective function determines the most likely hypothesis. Sections 4.1.2 and 4.1.3 describe foreground likelihood and appearance likelihood computation for this step. This paper is based on earlier work presented in [22,39].

4. Multi-person track assignment

Person detection and tracking is done using a volumetric 3D reconstruction of the scene, computed from the segmented foregrounds of the overlapping camera views $c \in C$. For one of the datasets this overlap area is shown as a blue line in Fig. 1(d) and 5(a) when using a minimum of 3 cameras. The 3D reconstruction is projected vertically onto the ground plane and regions with too little vertical mass to contain a person are removed. The area of each remaining region is compared to the average area expected to be covered by a single person on the ground plane to determine how many measurements (i.e. hypothetical persons) make up each region. Each region is segmented into a number of sub-regions equal to the estimated number of measurements by applying *EM* clustering with a Mixture of Gaussians. Random initialized *K*-means clustering is used to bootstrap the *EM* algorithm. To ensure that each mixture component is a reasonable representation of the average human shape as seen from top-down, the *maximization* step of the *EM* algorithm is adapted as follows to constrain the shapes of each mixture's components. First, after computing the new component weights and covariances, the weighted average

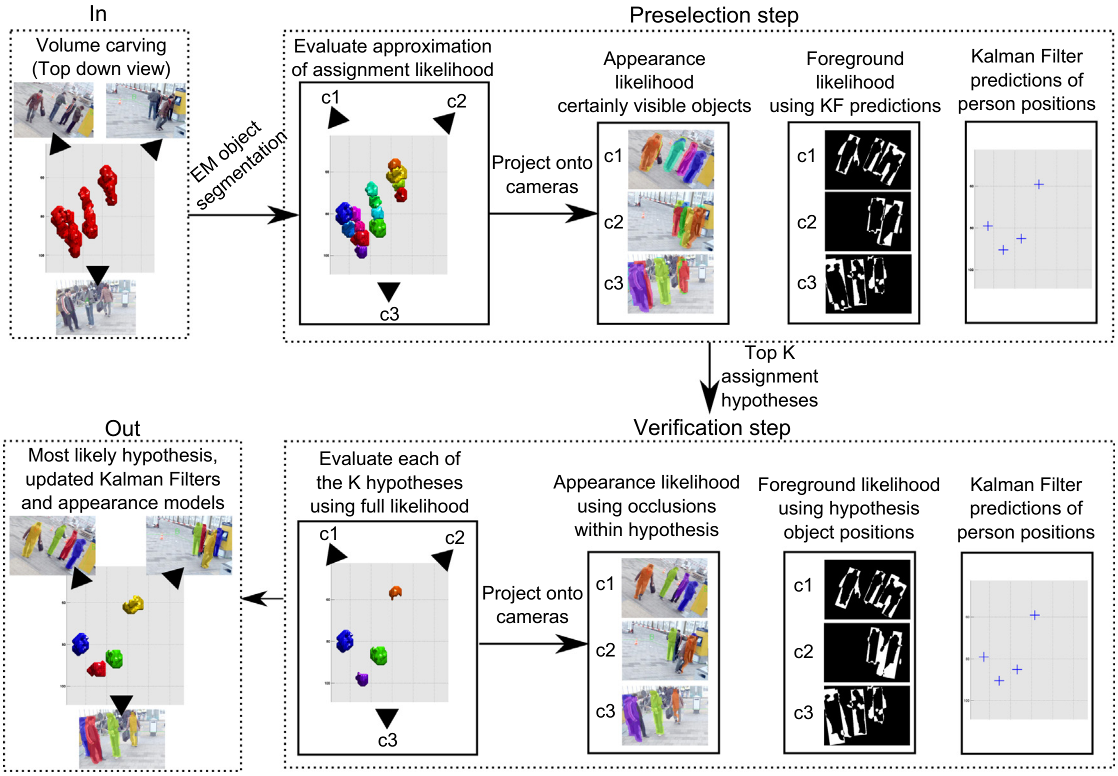

In
Volume carving
(Top down view)
EM object segmentation
Preselection step
Evaluate approximation of assignment likelihood
c1
c2
c3
Project onto cameras
Appearance likelihood certainly visible objects
Foreground likelihood using KF predictions
Kalman Filter predictions of person positions
c1
c2
c3
c1
c2
c3
Top K assignment hypotheses
Out
Most likely hypothesis, updated Kalman Filters and appearance models
Verification step
Evaluate each of the K hypotheses using full likelihood
c1
c2
c3
Project onto cameras
Appearance likelihood using occlusions within hypothesis
Foreground likelihood using hypothesis object positions
Kalman Filter predictions of person positions
c1
c2
c3
c1
c2
c3

Fig. 2. Overview of the proposed method for one time step: input, preselection using approximated likelihood, verification using full likelihood, output. The situation shown is the same as used in Fig. 1. Computation of the foreground likelihood and appearance likelihood are discussed in Sections 4.1.2 and 4.1.3.

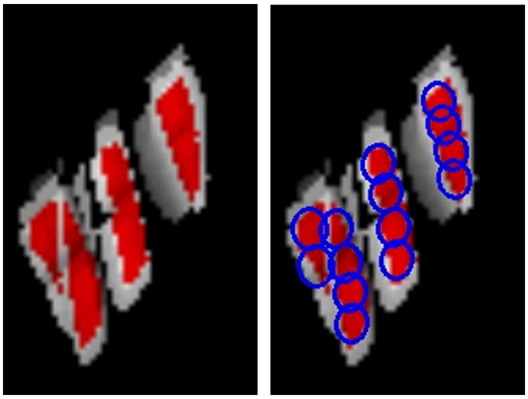


Fig. 3. *EM* clustering of top-down projected volume space. Showing top-down projection of the volume space from Fig. 1. Red: area with enough vertical mass to be a person. White: reconstructed regions with insufficient vertical mass. (left) unclustered top-down projection. (right) *EM* clustered top-down projection (blue circles: mixture components). 14 hypothetical persons locations have been found for the 4 actual persons. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

covariance is computed and used to replace all individual component covariances. Second, the aspect ratio of the covariance's minor and major axis is fixed at 2:3 using the covariance's eigenvalue decomposition. An example of volume carving and object segmentation (i.e. the position measurements) can be found in Fig. 1. Fig. 3 shows the *EM* clustering result for the same time step.

To treat person localization and track assignment jointly, we formulate the problem as an edge selection task on a bipartite graph and compute the maximum likelihood assignment. This graph consists of disjoint vertex sets $\mathcal{T}$ (tracks) and $\mathcal{P}$ (measurements), connected by an assignment hypothesis $\mathcal{E}$ consisting of a set of edges. Given M measurements, N currently existing tracks and O possible track creations, vertex set $\mathcal{T}=\{t_1,\ldots,t_N,\pi_1,\ldots,\pi_O\}$ contains vertices t_i representing existing person tracks, and π_i for generating new person tracks. Vertex set $\mathcal{P}=\{p_1,\ldots,p_M,\omega_1,\ldots,\omega_N,\bar{\pi}_1,\ldots,\bar{\pi}_O\}$ contains vertices p_j representing measurements, ω_j corresponding to terminated tracks, and $\bar{\pi}_j$ representing cases where no new person track is created. Edges $e_{i,j}\in\mathcal{E}$ are constrained such that: (1) for every vertex in $\mathcal{T}$, there is one edge $e_{i,j}\in\mathcal{E}$ connecting it to a vertex in $\mathcal{P}$ (2) vertices within $\mathcal{T}$ and $\mathcal{P}$ have maximum degree one (i.e. are connected by maximally one edge) and (3) $\mathcal{E}$ does not contain edges connecting a vertex π_i to ω_j or t_i to $\bar{\pi}_j$. From the above, it follows that the number of elements in each set is: $|\mathcal{P}|=M+N+O$ and $|\mathcal{T}|=|\mathcal{E}|=N+O$.

An assignment hypothesis $\mathcal{E}$ can be divided into subsets $\mathcal{E}^{\mathcal{C}},\mathcal{E}^{\mathcal{N}},\mathcal{E}^{\mathcal{D}}$, and $\mathcal{E}^{\mathcal{G}}$, each defined as follows:

- $\langle t_i,p_j\rangle\in\mathcal{E}^{\mathcal{C}}$: p_j is a person continuing track t_i,
- $\langle \pi_i,p_j\rangle\in\mathcal{E}^{\mathcal{N}}$: p_j is a person creating a new track,
- $\langle t_i,\omega_j\rangle\in\mathcal{E}^{\mathcal{D}}$: track t_i can be terminated (deleted),
- $\langle \pi_i,\bar{\pi}_j\rangle\in\mathcal{E}^{\mathcal{G}}$: no new track is created for π_i.

Any measurement p_j not connected by any edge in the assignment hypothesis is considered to be a false detection. A track t_i assigned to a termination node ω_j is not instantly removed, until it has had a termination node assigned for five consecutive frames. While it is not removed, the Kalman filter is used without any updates, to predict the expected position. This makes the positional uncertainty grow each frame, effectively increasing the search radius for matching detections. Vertices $\bar{\pi}_j$ are necessary to ensure that each assignment hypothesis consists of the same number of edges, regardless of whether or not a new track is created, and to balance the likelihood scores defined in the next section.

In our experiments we set $O=1$, thus allowing the addition of maximally one person track per frame. At a frame rate of 20 Hz, this means that 20 persons can be added every second. This constraint forces the algorithm to only create new tracks when it is highly likely that they represent an actual person. When allowing the creation of more tracks per frame, the selection of a suboptimal solution adding multiple tracks to describe a single person becomes more likely. Fig. 4 provides an overview of the graph structure.

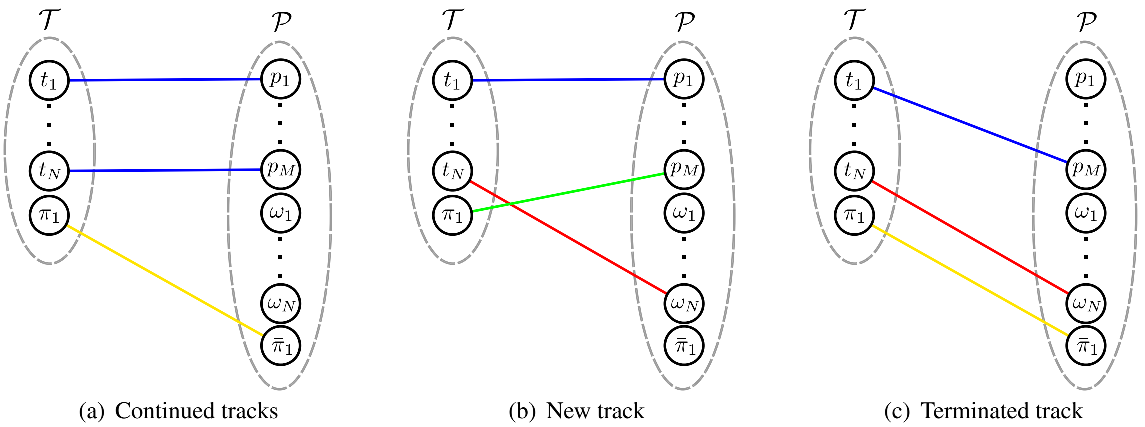
4.1. Likelihood formulation

A set of features $\mathcal{F}$ is measured using the assignments defined by $\mathcal{E}$. This set consists of the position of tracks and measurements on the ground plane $\mathcal{F}^{Pos}$, the foreground image regions $\mathcal{F}^{FG}$, the appearance of tracks and measurements $\mathcal{F}^{App}$ and the age of tracks $\mathcal{F}^{Age}$ (the number of frames a track exists). The joint likelihood $p(\mathcal{F}|\mathcal{E})$ is factorized in terms of the individual cue likelihoods:

$$p(\mathcal{F}|\mathcal{E}) = p(\mathcal{F}^{Pos}|\mathcal{E})\,p(\mathcal{F}^{FG}|\mathcal{E})\,p(\mathcal{F}^{App}|\mathcal{E})\,p(\mathcal{F}^{Age}|\mathcal{E}). \quad (1)$$

4.1.1. Position likelihood

The position likelihood $p(\mathcal{F}^{Pos}|\mathcal{E})$ is the probability of observing tracks and measurements matched in hypothesis $\mathcal{E}$ at their specific



(a) Continued tracks (b) New track (c) Terminated track

Fig. 4. Bipartite graph for $O=1$ showing three assignment hypotheses for vertex sets $\mathcal{T}$ and $\mathcal{P}$. Edges $e_{i,j}$ are color-coded for continued tracks in $\mathcal{E}^{\mathcal{C}}$ (blue), new tracks in $\mathcal{E}^{\mathcal{N}}$ (green), terminated tracks in $\mathcal{E}^{\mathcal{D}}$ (red) and not creating new tracks $\mathcal{E}^{\mathcal{G}}$ (yellow). (a) All tracks connected to measurements. (b) New track created for measurement p_M, track t_N not assigned and terminated. (c) Track t_N terminated, no new track created, unassigned measurement p_1 considered to be an artifact. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

locations on the ground plane. It can be factorized according to (2), splitting the computation over the different subsets of $\mathcal{E}$.

$$p(\mathcal{F}^{Pos}|\mathcal{E}) = \prod_{\langle t_i,p_j\rangle\in\mathcal{E}^{\mathcal{C}}} p(p_j|t_i) \cdot \prod_{\langle \pi_i,p_j\rangle\in\mathcal{E}^{\mathcal{N}}} p(p_j) \cdot \prod_{\langle t_i,\omega_j\rangle\in\mathcal{E}^{\mathcal{D}}} p(t_i) \cdot p_{nPos}^{|\mathcal{E}^{\mathcal{G}}|}. \qquad (2)$$

For edges in $\mathcal{E}^{\mathcal{C}}$, the likelihood of assigning a tracker t_i to a measurement p_j is based on the Kalman filtered estimate of the position of t_i at the current time step. The likelihood $p(p_j|t_i)$ is computed by evaluating the location of p_j under the posterior distribution of the Kalman filter prediction of the position of t_i. For edges in $\mathcal{E}^{\mathcal{N}}, \mathcal{E}^{\mathcal{D}}$ and $\mathcal{E}^{\mathcal{G}}$, the position based likelihood is computed using a distance map representing the distance on the ground plane between a given location in the volume space and the nearest edge of the area visible in all camera views. An example of this edge for the first of the datasets is shown as a blue line in Fig. 5(a). The likelihood of creating or terminating a track decreases linearly with the distance from the edge of this area. Fig. 5(b) shows an example of this likelihood map.

Using this distance map, both the likelihood $p(p_j)$ of the creation of a new tracker at the location of p_j and the likelihood $p(t_i)$ of the termination of a track at the location of t_i are determined. Finally, p_{nPos}, used to balance the likelihood computation when not all O potential new tracks are created, is set to the highest likelihood value in the distance map. Note that the position based likelihood can be evaluated for each $e_{i,j} \in \mathcal{E}$ independently. Changing the assignment of some other edge $e_{k,l} \in \mathcal{E}$, $k \neq i$, $l \neq j$, does not change the position based likelihood for $e_{i,j}$. This distinguishes this likelihood from the foreground likelihood and the appearance likelihood discussed next. Because these either evaluate a full multi-person hypothesis or depend on occlusions between persons, independence between individual edges cannot be assumed.

4.1.2. Foreground likelihood

The likelihood $p(\mathcal{F}_c^{FG}|\mathcal{E})$ of observing a certain foreground $\mathcal{F}_c^{FG}$ for a camera c given an assignment hypothesis $\mathcal{E}$ can be defined as the difference between the segmented foreground and a synthetic binary foreground image $S_c(\mathcal{E})$. Assuming the total binary foreground region $\mathcal{F}^{FG}$ of all camera views is generated by the persons in the scene augmented with some independent noise, an assignment hypothesis $\mathcal{E}$ corresponding to the actual scene situation should generate a good explanation of $\mathcal{F}^{FG}$. Inspired by [16], $S_c(\mathcal{E})$ is constructed by positioning 1.8 × 0.5 m rectangles at the locations of all p_j for which $e_{i,j} \in \mathcal{E}^{\mathcal{C}} \cup \mathcal{E}^{\mathcal{N}}$, and projecting these onto camera c. The difference between $\mathcal{F}_c^{FG}$ and $S_c(\mathcal{E})$ is defined as $|\mathcal{F}_c^{FG} \oplus S_c(\mathcal{E})|$, where $\oplus$ is the pixel-wise *XOR* operator and $|\cdot|$ is the number of foreground pixels in the *XOR* image. A parameter σ is used to weigh this term, controlling how well $\mathcal{F}_c^{FG}$ and $S_c(\mathcal{E})$ should match and determining the importance of the foreground likelihood compared to the other likelihood terms.

If the dependencies between the foreground segmentations for different cameras are assumed to be caused by the persons in the scene, $\mathcal{F}_c^{FG}$ can be considered conditionally independent of the other cameras given an assignment hypothesis $\mathcal{E}$. Therefore, $p(\mathcal{F}^{FG}|\mathcal{E})$ can be computed as follows:

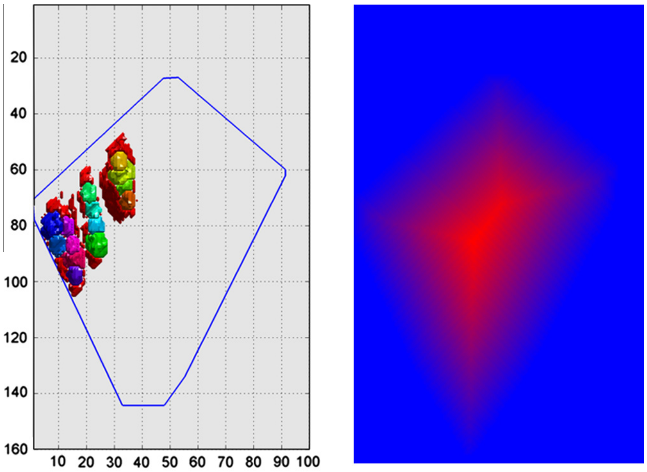
$$p(\mathcal{F}^{FG}|\mathcal{E}) = \prod_{c=1}^{C} p(\mathcal{F}_c^{FG}|\mathcal{E})$$

with

$$p(\mathcal{F}_c^{FG}|\mathcal{E}) = \frac{1}{Z_f} e^{-\frac{1}{\sigma}\frac{|\mathcal{F}_c^{FG} \oplus S_c(\mathcal{E})|+\alpha}{|\mathcal{I}|}}, \qquad (3)$$

where $|\mathcal{I}|$ represents the total number of pixels in a single image. The term α is added for hypotheses in which not all O potential new tracks π_i are created (i.e. $\mathcal{E}^{\mathcal{G}} \neq \emptyset$). This parameter puts a minimum on the number of extra pixels in $\mathcal{F}^{FG}$ that need to be explained by a hypothesis adding an extra track. The system is not particularly sensitive to the specific value of α and σ, which are generally set to be small (for more details, see Section 5.5). The normalization factor Z_f can be computed using the range of possible values for $|\mathcal{F}_c^{FG} \oplus S_c(\mathcal{E})|$, which is $[0, |\mathcal{I}|]$. However, because Z_f is constant and we will be maximizing the full likelihood, this normalization can be omitted. An example of a synthetic image $S_c(\mathcal{E})$ and the result of computing $\mathcal{F}_c^{FG} \oplus S_c(\mathcal{E})$ for one camera can be found in Fig. 6.

Since $\mathcal{F}^{FG}$ cannot be segmented per individual person, it is not possible to factorize $p(\mathcal{F}^{FG}|\mathcal{E})$ into independent terms for all $e_{i,j} \in \mathcal{E}$. On the other hand, evaluating $p(\mathcal{F}^{FG}|\mathcal{E})$ for all different $\mathcal{E}$ when M and N are large quickly becomes intractable. A solution is to start evaluation using an approximation $\hat{p}(\mathcal{F}^{FG}|e_{i,j})$, assuming all $e_{i,j} \in \mathcal{E}$ are independent. This step is discussed in more detail in Section 4.2.



(a) (b)

Fig. 5. (a) Topdown view of the volume space shown in Fig. 1(d), containing persons and noise. The blue line is the edge of the area visible by all cameras. (b) Likelihood map for determining addition/removal likelihood. Blue: high likelihood, red: low likelihood. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.3. Appearance likelihood

The likelihood $p(\mathcal{F}^{App}|\mathcal{E})$ of observing a set of appearance differences $\mathcal{F}^{App}$ between tracks and measurements given an assignment hypothesis $\mathcal{E}$ is defined as a distribution over the Hellinger distances [40] between the learned appearance models of $t_i \in \mathcal{T}$ and the measured appearances of $p_j \in \mathcal{P}$, connected by $e_{i,j} \in \mathcal{E}^{\mathcal{C}}$. Note that the Hellinger distance must be computed between appearances, and therefore can only be computed for $e_{i,j} \in \mathcal{E}^{\mathcal{C}}$.

Appearances are modeled using three RGB color histograms per camera using 5 × 5 × 5 bins (for R, G and B). They are computed by vertically dividing a person's volume into three height slices, as in [41]. This allows for more accurate appearance updates of partially occluded persons. Furthermore, some spatial information which would be lost when using one histogram per camera is retained. Making use of the 3D volume reconstruction of the scene, occlusions between different p_j (for which $e_{i,j} \in \mathcal{E}^{\mathcal{C}} \cup \mathcal{E}^{\mathcal{N}}$) are taken into account when sampling appearance information from the images. Occlusions are detected by labeling the voxels of each segmented object in the 3D scene reconstruction and projecting them onto all camera planes. Projection is done taking into account the ordering of the objects with respect to the camera. By selecting only the pixels of a single projected object in each camera, a mask of the unoccluded part of the person is acquired.

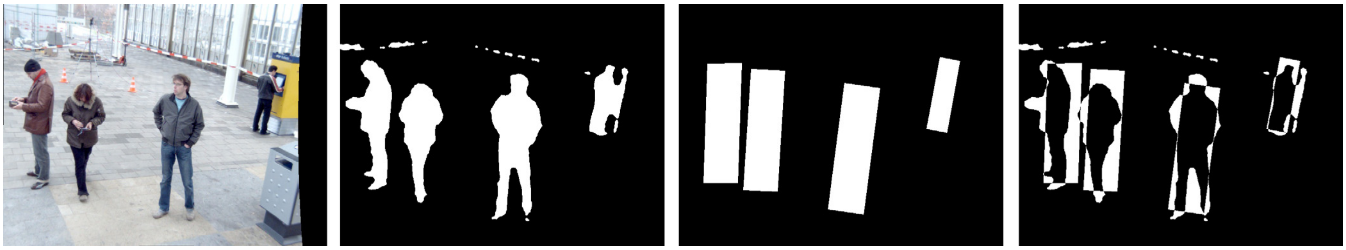


Fig. 6. Left to right: captured image, foreground segmentation $\mathcal{F}_c^{FG}$, synthetic image $S_c(\mathcal{E})$ created by putting rectangles at measurement locations, *XOR* image.

We assume a person's appearance can differ between viewpoints, for example due to an open jacket, a shawl or differently colored sleeves. Furthermore, we use viewpoints with a wide baseline. Therefore, we assume independence between the appearances recorded per camera and factorize the overall appearance likelihood over all individual cameras:

$$p(\mathcal{F}^{App}|\mathcal{E}) = \prod_{e_{i,j}\in\mathcal{E}^{\mathcal{C}}} \left[p(\mathcal{F}^{App}_{i,j}|\mathcal{E})\right] \cdot p_{nApp}^{|\mathcal{E}\setminus\mathcal{E}^{\mathcal{C}}|}$$

with

$$p(\mathcal{F}^{App}_{i,j}|\mathcal{E}) = \prod_{c=1}^{C} p(\mathcal{F}^{App}_{i,j,c}|\mathcal{E}). \tag{4}$$

In (4), $\mathcal{F}^{App}_{i,j,c}$ is the Hellinger distance between the appearances of t_i and p_j, averaged over the three histograms per camera. The distribution used for $p(\mathcal{F}^{App}_{i,j,c}|\mathcal{E})$ is discussed in Section 5.5. The factor p_{nApp} is used to compensate for the missing likelihood values for p_j not connected to any t_i (i.e. $e_{i,j} \notin \mathcal{E}^{\mathcal{C}}$). This factor is also used to replace the measured Hellinger distance for person parts that are occluded for more than 25%, since they give unstable appearance measurements.

Like the computation of the foreground likelihood described in Section 4.1.2, factorizing $p(\mathcal{F}^{App}|\mathcal{E})$ into independent $e_{i,j} \in \mathcal{E}$ is not possible. Due to occlusions between different p_j, only full hypotheses can be evaluated for which it is clear which p_j should be taken into account and which should not. This makes computation of $p(\mathcal{F}^{App}|\mathcal{E})$ for all possible $\mathcal{E}$ quickly intractable. Our solution to this issue is described in Section 4.2.

4.1.4. Age likelihood

The age likelihood $p(\mathcal{F}^{Age}|\mathcal{E})$ defines the probability of assigning tracks with a certain age to measurements as specified by assignment hypothesis $\mathcal{E}$. It ensures more stable tracking results over time, making the continuation of an existing track (assigning it to a nearby measurement) more likely than assigning a newer track to the same measurement when other likelihoods are similar. This is especially useful when measurements have been temporarily unavailable or when a nearby artifact that was erroneously tracked disappears. In the first case, the age likelihood prevents the creation of a new track when the measurement reappears and the original track is still available close by. In the second case, it prevents the erroneous tracker from 'pushing away' the real tracker when the appearance and position of the artifact were similar to the real track. The likelihood can be computed for individual edges $e_{i,j} \in \mathcal{E}$ and is factorized for the different subsets of $\mathcal{E}$ as follows:

$$p(\mathcal{F}^{Age}|\mathcal{E}) = \prod_{e_{i,j}\in\mathcal{E}^{\mathcal{C}}\cup\mathcal{E}^{\mathcal{D}}} \left[p(\mathcal{F}^{Age}|e_{i,j})\right] \cdot p_{nAge}^{|\mathcal{E}^{\mathcal{N}}\cup\mathcal{E}^{\mathcal{G}}|}$$

with

$$p(\mathcal{F}^{Age}|e_{i,j}) = \frac{1}{Z_a} e^{-1/A(t_i)}. \tag{5}$$

Here, $A(t_i)$ is the age of track t_i in frames and p_{nAge} replaces the age likelihood when $A(t_i)$ is 0 ($e_{i,j} \in \mathcal{E}^{\mathcal{N}}$) or no track is involved in the assignment ($e_{i,j} \in \mathcal{E}^{\mathcal{G}}$). Since p_{nAge} is defined proportional to $p(\mathcal{F}^{Age}|e_{i,j})$ (Section 5.5), the normalization factor Z_a is constant and can be omitted (as in (3)).

4.2. Likelihood optimization

A brute-force approach to finding the most likely set of edges $\mathcal{E}$ for (1) would quickly become intractable due to the combinatorial nature of the assignment problem, especially for large M and N. Instead, the idea is to split the process into a *preselection* step and a *verification* step. In the *preselection* step, the top-K most likely hypotheses are selected based on an approximation of the full likelihood function from (1), using a subset of features measured independent of occlusions. In the *verification* step, this top-K is evaluated using the full likelihood function from (1) to determine the most likely hypothesis. For our experiments we use a value of $K = 40$.

Since (3) and (4) contain terms dependent on the complete assignment $\mathcal{E}$ (e.g. due to occlusion), an approximation $\hat{p}(\mathcal{F}|\mathcal{E})$ of the likelihood $p(\mathcal{F}|\mathcal{E})$ is used in the *preselection* step. It uses the approximated likelihoods $\hat{p}(\mathcal{F}^{FG}|\mathcal{E})$ and $\hat{p}(\mathcal{F}^{App}_{i,j,c}|\mathcal{E})$, where each edge's likelihood is independent of the other edges:

$$p(\mathcal{F}^{FG}|\mathcal{E}) \approx \hat{p}(\mathcal{F}^{FG}|\mathcal{E}) = \prod_{e_{i,j}\in\mathcal{E}} \hat{p}(\mathcal{F}^{FG}|e_{i,j},\mathcal{T}) \tag{6}$$

$$p(\mathcal{F}^{App}_{i,j,c}|\mathcal{E}) \approx \hat{p}(\mathcal{F}^{App}_{i,j,c}|\mathcal{E}) = \prod_{e_{i,j}\in\mathcal{E}} \hat{p}(\mathcal{F}^{App}_{i,j,c}|e_{i,j},\mathcal{P}). \tag{7}$$

Murty's K-best assignment algorithm [42] is used to find the top-K most likely hypotheses according to $\hat{p}(\mathcal{F}|\mathcal{E})$. It solves the assignment problem in low polynomial time by iteratively minimizing the sum over the edge costs in a bipartite graph.

To determine the likelihood $\hat{p}(\mathcal{F}^{FG}|\mathcal{E})$, instead of constructing $S_c(\mathcal{E})$ by drawing rectangles in each camera for all assignments made in $\mathcal{E}$, we create $S_c(e_{i,j},\mathcal{T})$ for each $e_{i,j}$ by putting rectangles at the Kalman filter predicted tracker locations of all $t_k \in \mathcal{T},\ k \neq i$. For $e_{i,j} \in \mathcal{E}^{\mathcal{C}} \cup \mathcal{E}^{\mathcal{N}}$, an extra rectangle is added at the location of p_j while for $e_{i,j} \in \mathcal{E}^{\mathcal{D}} \cup \mathcal{E}^{\mathcal{G}}$ no rectangle is added. The foreground likelihood $\hat{p}(\mathcal{F}^{FG}|e_{i,j},\mathcal{T})$ from (6) is now computed as

$$\hat{p}(\mathcal{F}^{FG}|e_{i,j},\mathcal{T}) = \prod_{c=1}^{C} e^{-\frac{1}{\sigma}\frac{|\mathcal{F}_c^{FG}\oplus S_c(e_{i,j},\mathcal{T})|+\alpha}{|\mathcal{I}|}}. \tag{8}$$

To make the extracted appearance of a measurement p_j independent of $\mathcal{E}$, only parts of p_j sure to be unoccluded in camera c by any $p_k \in \mathcal{P},\ p_k \neq p_j$, are used for computing the appearance for camera c. Therefore, all $p_j \in \mathcal{P}$ are assumed to be objects when computing occlusions during preselection. The likelihood $\hat{p}(\mathcal{F}^{App}_{i,j,c}|e_{i,j},\mathcal{P})$ of the appearance difference between track t_i and measurement p_j for an assignment $e_{i,j}$ can then be computed as described in Section 4.1.3.

With these likelihood approximations, the full likelihood $p(\mathcal{F}|\mathcal{E})$ from (1) can be approximated as the preselection likelihood $\hat{p}(\mathcal{F}|\mathcal{E})$ according to (9).

$$\begin{aligned}\hat{p}(\mathcal{F}|\mathcal{E}) = & \prod_{e_{i,j}\in\mathcal{E}^{\mathcal{C}}} p(p_j|t_i)\,\hat{p}(\mathcal{F}^{FG}|e_{i,j},\mathcal{T})\,\hat{p}(\mathcal{F}^{App}_{i,j}|e_{i,j},\mathcal{P})\,p(\mathcal{F}^{Age}|e_{i,j}) \\ & \cdot \prod_{e_{i,j}\in\mathcal{E}^{\mathcal{N}}} p(p_j)\,\hat{p}(\mathcal{F}^{FG}|e_{i,j},\mathcal{T})\,p_{nApp}\,p_{nAge} \\ & \cdot \prod_{e_{i,j}\in\mathcal{E}^{\mathcal{D}}} p(t_i)\,\hat{p}(\mathcal{F}^{FG}|e_{i,j},\mathcal{T})\,p_{nApp}\,p(\mathcal{F}^{Age}|e_{i,j}) \\ & \cdot \prod_{e_{i,j}\in\mathcal{E}^{\mathcal{G}}} p_{nPos}\,\hat{p}(\mathcal{F}^{FG}|e_{i,j},\mathcal{T})\,p_{nApp}\,p_{nAge}. \end{aligned} \tag{9}$$

All factors in (9) define the likelihood for one specific edge $e_{i,j} \in \mathcal{E}$, independent of all other edges. After taking the log of this expression, it is used to compute the K most likely assignment hypotheses using Murty's algorithm. After evaluating these top-K hypotheses using the full likelihood $p(\mathcal{F}|\mathcal{E})$, the overall most likely hypothesis is selected.

5. Experiments

In this section, we compare the proposed method to three state-of-the-art approaches: two batch mode approaches from [24,23] and one recursive method from [32]. For this last comparison, results presented in [32] are compared to results of our method on the same dataset. Furthermore, a comparison is made with the version of the proposed method presented in [22] which will be referred to as Recursive Combinatorial Track Assignment (RCTA). The main differences between the method presented here and RCTA have been described in [39]. The most important changes are: the use of a more general way to compute the foreground likelihood, using Kalman filtered tracker position estimates for the foreground likelihood during *preselection* and using the Kalman filter's posterior distribution for the distance likelihood of $e_{i,j} \in \mathcal{E}^{\mathcal{C}}$. Experiments are performed on three different challenging real-world datasets.

5.1. K-shortest paths

The two batch mode approaches in the comparison are versions of the K-Shortest Paths (KSP) tracker presented by Berclaz et al. [24]. KSP does tracking by minimizing the flow through a graph constructed by stacking POMs [16] from a batch of sequential frames. Each POM location is a graph node, connected to its 9 neighbors in the next frame. Tracks are modeled as flows through this graph with costs defined by the POM probabilities at locations connected by the tracks. Finding the optimal set of disjoint tracks with minimum cost is formulated as a linear programming problem. Like [24], we use consecutive batches of 100 frames. Track consistency between batches is created by adding the last frame of the previous batch in front of the current batch. Flows are forced to start at the track locations from the last frame of the previous batch.

KSP-App [23] extends KSP, incorporating appearance information into KSP's linear programing formulation. For this purpose, the KSP graph is stacked L times, creating L 'groups'. Each of these groups is assigned a predefined appearance template and the number of objects that can have a path in each group is limited. Each appearance consists of one color histogram per camera. Using a KSP iteration, the graph is pruned and appearances are extracted at locations along the tracks and at locations where tracks are separated by at most 3 nodes. The extracted appearance information is compared to the templates using KL divergence and the graph's edges are reweighed using these values. KSP is run a second time using the new graph to determine the final paths. More detailed descriptions of KSP and KSP-App are found in [24,23].

5.2. Background estimation

The datasets used in our experiments contain significant amounts of lighting changes and background clutter. An *adaptive* background estimation method, compensating for changes in the scene over time by learning and adapting the background model on-line, would be preferred in this case. The method presented in [43] uses a Mixture of Gaussians per pixel to model the color distribution of the background and is to some extent robust with respect to illumination. However, in our scenarios persons tend to stand still for some time (up to a minute). Preliminary experiments have shown that the adaptive nature of the method causes them to dissipate into the background, creating false negatives. The proposed method solves this as suggested in [44], adding tracker feedback into the learning process by updating the background model only at locations without tracks.

For the KSP methods, this type of feedback is not straightforward since tracking results are only available after processing the full batch, preventing frame-to-frame reinforcement of the learned background model. Therefore, we use the foreground segmentation method from [45], implemented by [46], as a second, *static* background estimation method. It models the empty scene using eigenbackgrounds constructed from images of the empty scene under different lighting conditions. Nevertheless, foreground segmentations created by this method show more noise than foregrounds generated by our adaptive method. For comparison, the static background model is used for both KSP and our methods. Furthermore, the foreground segmentations from our adaptive background model is used as input to the KSP methods, effectively cascading KSP and the proposed method.

5.3. Datasets

Experiments were done on two new datasets[1] as well as on two public benchmark datasets. Fig. 7 shows an example frame from the two novel datasets. The outdoor 'train station data' has 14 sequences of in total 9925 frames, recorded on a train platform. Between two and five actors enact various situations ranging from persons waiting for a train to fighting hooligans. The scenes have dynamic backgrounds with trains passing by and people walking on the train platform. Lighting conditions vary significantly over time. The area of interest (a.o.i.) is 7.6 × 12 m and is viewed by three overlapping, frame synchronized cameras recording 752 × 560 pixel images at 20 fps. Ground truth (GT) person locations are obtained at each frame by labeling torso positions, annotating the shoulder and pelvis locations of all persons in all cameras and projecting these onto the ground plane.

The indoor 'hall data' is one 9080 frame sequence recorded in a large central hall. During the first half, actors move in and out of the scene in small groups. After this, two groups of about eight people each enter one by one and start arguing and fighting. Fig. 8(b) shows the number of people in the scene over time. The 12 × 12 m a.o.i. is viewed by four overlapping, frame synchronized cameras recording 1024 × 768 pixel images at 20 fps. GT positions are generated every 20th frame by annotating every person's head location in every camera, triangulating these points in 3D and projecting them onto the ground plane. This data is considerably more difficult than the previous dataset, since it contains more, similarly clothed people forming denser groups, and the cameras are placed

[1] The datasets are made available for non-commercial research purposes. Please follow the links from http://www.gavrila.net or contact the second author.

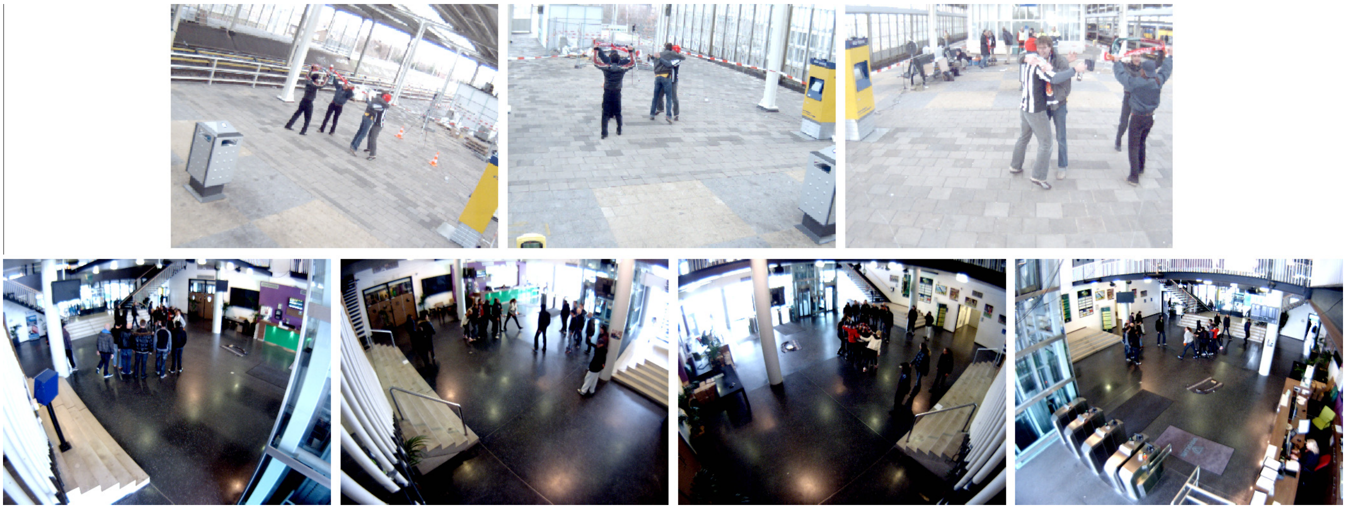
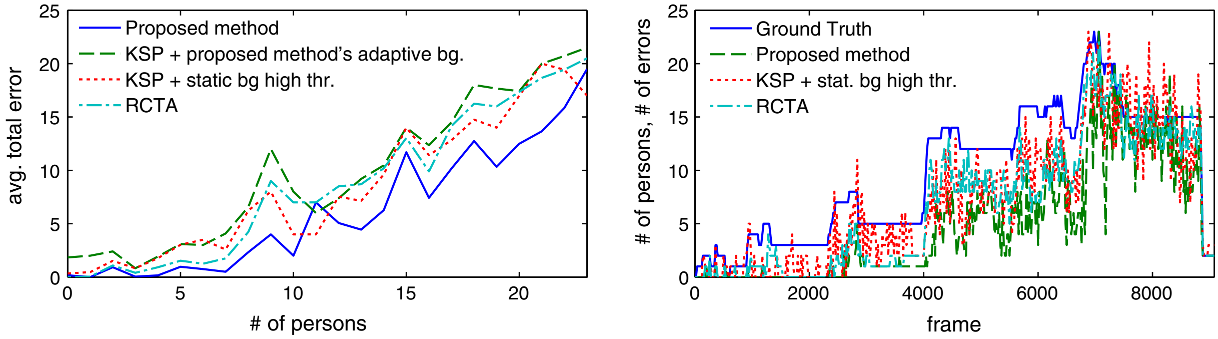


Fig. 7. All viewpoints of the train station data (top) and the hall data (bottom).



(a) Average total error vs. # of persons

(b) #persons (Ground Truth) and #errors (methods) over time

Fig. 8. People and error statistics over time for the hall dataset.

further away from the scene. Furthermore, many people wear dark clothing, which combined with the dark floor of the hall and multiple badly lit regions complicates foreground segmentation.

To facilitate benchmarking the proposed method, additional experiments are done on the publicly available APIDIS dataset[2] [47]. An example frame for this dataset can be found in Fig. 10. The APIDIS dataset contains 1500 frames showing part of a basketball match. A total of 7 cameras with partially overlapping views record 1600 × 1200 pixel images at 25 fps. The scene contains 12 persons (2 referees and two teams of 5 players each) showing fast, abruptly changing motion patterns and heavy occlusion. Furthermore, players in the same team have highly similar appearances and the scene has strong highlights and shadows from the overhead lights. As in [32], the a.o.i. is set to a 15 × 15 m region and tracking is only performed on the left half of the basketball court which is covered by 5 out of the 7 cameras, i.e. cameras 1, 2, 4, 5 and 7. A number of these cameras have very similar viewpoints, showing the scene from two sides of the field and from top-down. This similarity limits the amount of additional information per view.

GT annotations for this data are generated from the 2D per-camera bounding boxes provided with the dataset. The center point of each bounding box is triangulated into the calibrated 3D space using all viewpoints in which the bounding box is visible. The 3D triangulated point is projected onto the ground plane to get each person's GT position. As was done by [32], tracking performance is evaluated on every 10th frame and all GT positions have visually been verified.

Finally, experiments were done using the first EPFL terrace sequence[3] which was also used for experiments in [24]. This 'terrace' dataset consists of 5000 frames recorded using 4 cameras with overlapping fields of view. The frame-synchronized cameras record 360 × 288 pixel images at 25 fps. The outdoor scene consists of 9 persons entering the scene one-by-one, walking randomly around the scene. There are multiple illumination changes and persons' appearances are similar. Segmenting persons can be challenging at times when illumination conditions cause very little contrast between persons and the background. The GT for this dataset has been annotated every 25th frame. GT positions are given on the ground plane, in a 30 × 44 cell grid with 25 × 25 cm cells. The grid cell positions have been converted to ground plane coordinates and tracking precision is evaluated by comparing tracked positions to these ground plane coordinates.

5.4. Evaluation measures

Tracking performance is evaluated using the same metrics as in [23]. A missed detection (*miss*) is generated when no track is found within 0.5 m of a GT location, while a false positive (*fp*) is a track without a GT location within 0.5 m. The mismatch error (*mme*) counts the number of identity switches within a track and is

[2] http://www.apidis.org/Dataset.

[3] http://cvlab.epfl.ch/cms/site/cvlab2/lang/en/data/pom.

increased when a track switches between persons. The global mismatch error (*gmme*) is increased for every frame a track follows a different person than the one it was created on. GT persons outside the area covered by all cameras and GT persons associated with a tracks outside this area (but within 0.5 m of the GT person) are regarded as optional (i.e., associated tracks are neither credited nor penalized). All other GT persons are considered required and are aggregated in *gt*. This results in small differences between *gt* for different experiments.

The widely used Multi Object Tracking Precision (MOTP) and Multi Object Tracking Accuracy (MOTA), defined by [48], summarize performance. MOTP describes the average distance between tracks and GT locations for correctly tracked objects. Since MOTP describes the distance in cm, lower values mean better precision. MOTA describes the fraction of errors w.r.t. the number of GT locations and is defined as follows:

$$\text{MOTA} = 1 - \frac{fp + miss + mme}{gt}. \tag{10}$$

Higher accuracy scores are better, with 1 as the maximum MOTA score and no minimum (more errors then GT gives a negative score).

5.5. Implementation details

For the train station data, POM settings are taken from [16], using 20 cm grid cells and person ROI of 175 × 50 cm. For the hall data, 40 cm grid cells are used. Smaller cells cause POM to detect too few people in the dense second half of the scenario. POM's person prior was set to 0.002 in all experiments. Appearance templates for KSP-App are sampled at manually selected POM locations for the train station dataset. For the hall dataset, running KSP-App turned out to be problematic. The high number of persons and the crowd density, combined with the 40 cm grid cells enlarging the spatial neighborhood of each cell, limit graph pruning, increasing the problem complexity. Even when using a reduced set of 5 instead of 23 templates, the complexity of the linear programming problem to be solved by KSP-App becomes too high, resulting in a processing time of more than a month for the scenario. Processing times like these are not applicable in real-time usage scenarios such as a surveillance context. Because of the long processing time needed, we were unable to get KSP-App results on the hall dataset.

RCTA and the proposed method both use the same parameter settings where applicable. RCTA parameters not used in the proposed method are set according to [22]. For the train station and hall data, 8 × 8 × 8 cm voxels are used. To better match the setup in [32], voxel sizes for the APIDIS data were set to 5 × 5 × 8 cm. Because of memory limitations in the implementation of the proposed method, the resolution used is about 5 times less accurate than the one used in [32]. For the train station data and the hall data, only objects segmented in all cameras (resp. 3 and 4 cameras) are reconstructed to limit artifacts. Because of the limited number of viewpoints in the APIDIS data, the number of cameras necessary to 'carve out' a voxel is made dependent on the number of views overlooking a specific part of the a.o.i. A voxel is 'carved out' when all cameras covering that part of the a.o.i. show foreground. This enables robust detections in the largest possible part of the a.o.i., despite the limited number of viewpoints. To prevent very noisy detections, the region still has to be covered by at least 3 cameras.

The factor α from Eqs. (3) and (8) depends on the expected size of a person entering the scene per camera. The 'train station data' with cameras relatively close to the scene, uses 1% of the number of pixels in the image. The 'hall data', with cameras further away from the scene, uses 0.8% and the APIDIS data uses 0.3%. The weighting factor σ is set to 0.003.

A good value for the default appearance likelihood p_{nApp} was experimentally found to be at a Hellinger distance of 0.5 (the Hellinger distance has range $[0, 1]$). The distribution for $p(\mathcal{F}^{App}_{i,j,c}|\mathcal{E})$ is learned from examples of appearance differences for correct track-to-measurement assignments. A log-normal distribution with mean -2.21 and standard deviation 0.55 is used. The default age likelihood p_{nAge} is set to e^{-20}.

Most 'train station' scenarios start with persons in the scene. Because of the distance map used to determine likelihood of creating new tracks as described in Section 4.1.1, track creation likelihood in the middle of the scene is low for RCTA and the proposed method. To bootstrap the proposed method, the tracker creation likelihood is set to p_{nPos} for the first 10 frames. This is similar to KSP and KSP-App, where the nodes of all ground plane locations in the first frame are connected to the source node. Furthermore, because appearance measurements are inaccurate when not all people have been detected, the appearance factors in the objective function are not evaluated and appearance models are not updated during the first 10 frames. Since RCTA makes a strong assumption that scenes start without any persons present, we used GT initialization for the first frame in experiments with RCTA on the train station data, as was done in [39]. The experimental results show that even with GT initialization, RCTA is outperformed by the other methods.

To reason about measurements occluded by static objects, we opted to use manually created foreground masks to reconstruct a volume space containing static objects (see [19] for an automatic method). The foreground segmented images at each time step are augmented with the static object foregrounds before volume reconstruction. After reconstruction, the static object volume is subtracted. For the hall dataset, pillars in the scene are marked as static occluders. For the APIDIS dataset, the baskets hanging above the field and the rotating billboards on the side of the field are marked.

In order to show the benefit of the proposed two-step approach, we evaluated the proposed approach using only the *preselection* step and skipping the *verification* step. In this setup K is set to 1, adjusting the *preselection* step to only return the most likely hypothesis based on the approximation of the likelihood function. Person positions and appearances are updated based on this hypothesis and used in the next time step. The results based on this version of the proposed method are reported as 'Prop. w/o verify'.

5.6. Tracking results

5.6.1. Train station dataset

Table 2 shows the results for the train station and hall datasets. For the train station data, scores are accumulated over all scenarios. Table 2(a)'s top 7 rows show results of the stand-alone trackers combined with each background estimation method. Compared to [39], a more representative set of training backgrounds is used for the static background model, reducing segmentation noise for this dataset. This results in improved performance for the experiments using the static background model.

The proposed method shows overall improvement over RCTA. Combining it with the adaptive background model gives the highest MOTA. Furthermore, of the three error types making up the MOTA (*miss* rate, *fp* and *mme*), only the proposed method using static backgrounds is able to produce lower *fp* and *mme*. However, this is a direct result of the much higher *miss* rate (creating no tracks produces no *fp* and *mme*). Static background experiments suffer from foreground segmentation errors. For the train station data, the static background model is configured to minimize false positives from strong illumination changes and shadows, classifying as few people as possible as background. This trade-off results

Table 2
Performance of all methods and background models, on the train station and hall datasets. MOTA: accuracy, higher is better. MOTP: precision (cm), lower is better. *miss*: misses. *fp*: false positives. *mme*: mismatch error. *gmme*: global *mme*. *gt*: ground truth persons.

(a) Results on the train station data

method	background	MOTA	MOTP	miss	fp	mme	gmme	gt
Proposed method	adaptive	**0.95**	14	704	960	27	7166	32800
Proposed method	static	0.64	**13**	11402	453	6	6920	32785
Prop. w/o verify	adaptive	0.76	17	1655	5991	109	10226	32795
KSP-App	static	0.92	16	1291	1353	33	4139	32735
KSP	static	0.91	16	1344	1405	47	9508	32732
RCTA	adaptive	0.74	15	3407	5075	59	10130	32756
RCTA	static	0.76	**13**	3879	3788	62	9668	32750
KSP-App	Prop. method, adaptive	0.92	16	1178	1303	39	6909	32683
KSP	Prop. method, adaptive	0.92	16	1236	1360	48	9451	32685

(b) Results on the hall data

method	background	MOTA	MOTP	miss	fp	mme	gmme	gt
Proposed method	adaptive	**0.49**	18	1085	947	212	1799	4419
Proposed method	static, low thr.	0.37	19	1262	1303	237	1900	4419
Proposed method	static, high thr.	0.24	20	3145	164	47	1045	4419
Prop. w/o verify	adaptive	0.42	19	1148	1248	188	2205	4419
KSP	static, low thr.	-0.16	29	2702	2286	126	1364	4414
KSP	static, high thr.	0.28	28	1996	917	252	1966	4411
RCTA	adaptive	0.30	17	2654	361	62	1536	4419
RCTA	static, low thr.	0.24	**16**	3085	250	27	1085	4419
RCTA	static, high thr.	0.24	17	3084	199	58	1248	4419
KSP	Prop. method, adaptive	0.21	28	1925	1195	362	2265	4415

The best MOTA and MOTP scores are highlighted in bold.

in higher *miss* rates for methods using static backgrounds. Because both the proposed method and RCTA assume a person to be well segmented in all cameras for reliable volume carving, segmentation errors have most effect here. The POM detector has no such assumption, making it more robust to these artifacts.

MOTP is worse for the KSP methods since volume carving, allowing higher spatial resolutions than POM, offers more positional flexibility. KSP-App's main improvement over plain KSP is in the *mme* and *gmme*. This is to be expected, since KSP-App performs extra processing of the KSP tracks to correct id switches.

The proposed method with the *verification* step disabled shows significantly lower performance on this data. This shows that, while in the two-step approach the correct hypothesis is in the top 40 hypotheses returned by the *preselection* step, it is not the top ranking hypothesis in this step. Computing the more informed full likelihood in the *verification* step is necessary to re-rank the hypotheses and get the best result. Because information of occluded objects is not taken into account during *preselection*, the method without *verification* is more susceptible to tracking ghosting artifacts, resulting in a higher *fp* rating. This also causes more identity switches.

The last two rows of Table 2(a) show the result of using the adaptive backgrounds generated by the proposed method as input to the KSP methods. Since person segmentations are more emphasized in the adaptive backgrounds and show less holes, the *miss* and *fp* rates improve w.r.t. the KSP methods using static backgrounds. However, since KSP is less sensitive to segmentation errors and static background performance is already reasonable, changing the background model has limited effect.

When overall tracking performance is good, a slightly higher *mme* can reduces the number of *gmme*, since part of the extra id changes can switch the tracker back to its original target. A person who's tracker switches once at the start of the scene will have his *gmme* increasing for the rest of the scene, while another switch reassigning the original tracker to this person increases its *mme* but can significantly reduce its *gmme*. The *gmme* for the proposed method is only outperformed by the two KSP-App versions and the proposed method using static backgrounds. For this last method, the low *gmme* can be explained by its bad overall performance. For the KSP-App methods, the higher *mme* contributes towards lower *gmme*.

The top row of Fig. 9 shows some examples of tracking results from one of the dataset's viewpoints.

5.6.2. Hall dataset

Table 2(b) shows the results on the hall dataset. The large number of people and their close proximity in the second half of the scenario results in lower performance compared to the train station data. RCTA's failure creating tracks is seen in the high *miss* rate but lower number of *fp*. This can to a large extent be blamed on the way the foreground likelihood is computed by RCTA. The synthetic foreground image used in RCTA's *preselection* stage contains only one measurement, making it a bad representation of the actual foreground and resulting in low likelihoods. The proposed method using its adaptive background model again outperforms the other methods. Compared to the train station dataset, the performance difference between the proposed method and KSP is more significant, using either the static backgrounds or the adaptive backgrounds from the proposed method. This shows a more fundamental issue of KSP and the POM detector with crowded scenarios. When persons' foreground segmentations are not separated in any view, POM will detect too few persons, assuming the rest of the foreground regions are noise. Enlarging the POM grid to 40 cm cells partially compensates this, but causes missed detections when people are very close together and lowers detection precision. The proposed method's volume carving and clustering approach has less problems splitting dense groups, but also creates incorrect detections when the volume space is incorrectly clustered. KSP's lack of appearance model makes it is more prone to track switches as well.

Because of the challenging conditions of the hall dataset described earlier, using the same configuration for the static background model as for the train station data results in many missing

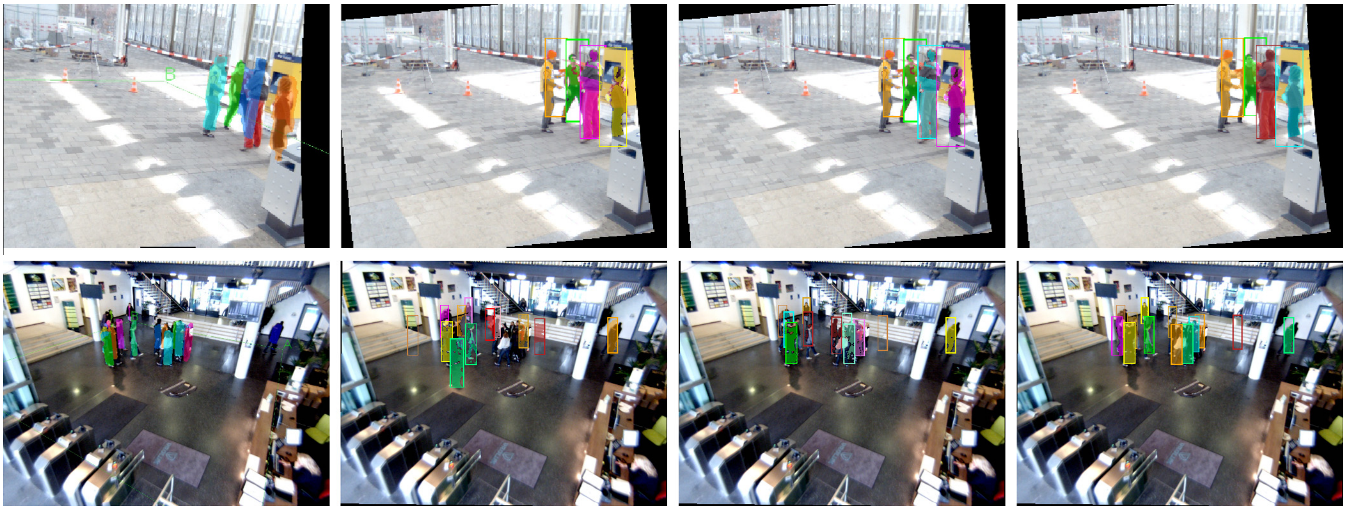


Fig. 9. Examples of tracking results. (top) Train station data: Proposed method, KSP-App, KSP and proposed method/KSP-App cascade. (bottom) Hall data: Proposed method, KSP with low background threshold, KSP with high background threshold and proposed method/KSP cascade.

foreground segments. Therefore, additional experiments are done using a lower segmentation threshold, detecting more people but increasing the foreground noise from illumination and shadows. Results using the 'high' and 'low' threshold settings are marked as resp. 'high thr.' and 'low thr.' in Table 2(b). Again, the proposed method shows sensitivity to missing detections, resulting in a lower MOTA for the high threshold static backgrounds. KSP shows bad performance when using the low threshold however, producing more errors than the *gt*, resulting in a negative MOTA. When using the high threshold KSP shows better results but also suffers from the missing detections.

As with the train station data, disabling the *verification* step reduces performance. However, because the proposed method's benefit in crowded scenarios still exists, it shows better performance than the KSP based methods on this dataset. In the crowded parts of this data where many people stand close together, the GT solution is not always included in the top 40 *preselection* hypotheses of the two-step approach. Therefore, the difference in performance between the two-step approach and the proposed method without *verification* is smaller than on the train station data.

The bottom row of Fig. 9 shows some examples of tracking results from one of the dataset's viewpoints. In Fig. 8(a) the average total error ($fp + miss + mme$) per frame containing a certain number of people is shown for the hall data for the best performing versions of each method. The figure shows a relatively constant error up to 7 people, after which it starts to increase linear with the number of people in the scene. Fig. 8(b) shows the evolution of both the actual number of people during the scene (the blue[4] line), and the error per frame for the proposed method, KSP with static background and high threshold and RCTA.

5.6.3. APIDIS dataset

Table 3 shows the performance of the proposed method, the method from [32] and KSP on the APIDIS dataset. Note that the results on the method from [32] and KSP have been taken from Table 2 in [32]. The difference in the number of GT positions is due to the way persons on the border of the a.o.i. are handled (i.e. if GT positions are required or optional). Even though slightly more GT positions have been evaluated, our method significantly outperforms the other two methods and shows less errors in absolute terms. The main reason for this performance difference between the proposed method and the method from [32] is in the way new tracks are handled. Comparing tracking footage provided by [32] and our own results, a number of persons entering the scene is missed by [32]. In the proposed method, not all persons are detected as soon as they enter the scene, but because of the smooth likelihood map and the foreground likelihood, they are still created at a later point in time. Furthermore, because the proposed method always takes into account the appearance likelihood, trackers that get stuck onto a ghosting artifact can more easily recover when the ghost disappears and the tracker drifted away from the person. The same is true for identity changes that occur while persons are occluded and re-appear at some distance from each other. Finally, the proposed method has the ability of deleting the stray tracker and creating a new one for the actual person in the middle of the scene. As with the hall dataset, disabling the *verification* step decreases performance while still outperforming KSP as well as the method from [32]. Tracking results of the proposed method on the APIDIS dataset can be found in Fig. 10. The images show results at one time instant, seen from all five cameras used.

Table 3
Performance of the proposed method, KSP and the method proposed by [32] on the APIDIS dataset. Results from [32] and KSP have been taken from Table 2 in [32]. MOTA: accuracy, higher is better. MOTP: precision (cm), lower is better. *miss*: misses. *fp*: false positives. *mme*: mismatch error. *gmme*: global *mme*. *gt*: ground truth persons.

method	MOTA	MOTP	miss	fp	mme	gmme	gt
Proposed method	**0.91**	**12**	51	22	3	269	875
Prop. w/o verify	0.84	13	94	45	4	203	876
Possegger [32]	0.68	21	172	88	9	-	828
KSP (from [32])	0.49	23	220	156	46	-	827

The best MOTA and MOTP scores are highlighted in bold.

5.6.4. EPFL terrace dataset

Table 4 shows the performance of the proposed method and KSP on the terrace dataset. The methods show comparable performance, with slightly higher scores for the proposed method. Both methods make use of adaptive backgrounds created by the proposed method. Due to difficult lighting conditions and low contrast between the appearance of some persons and the background, generating good quality foreground segmentations is hard. The static

[4] For interpretation of color in Fig. 8, the reader is referred to the web version of this article.

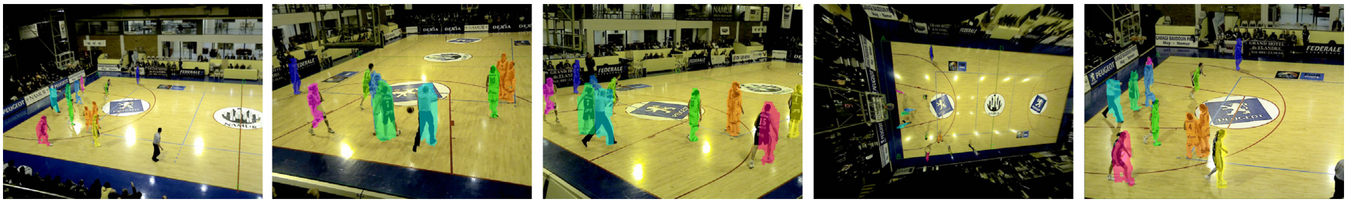


Fig. 10. Example frame from the APIDIS dataset with tracking results from the proposed method. Showing cameras 1, 2, 4, 5 and 7.

Table 4
Performance of the proposed method and KSP on the EPFL terrace dataset. MOTA: accuracy, higher is better. MOTP: precision (cm), lower is better. *miss*: misses. *fp*: false positives. *mme*: mismatch error. *gmme*: global *mme*. *gt*: ground truth persons.

method	background	MOTA	MOTP	miss	fp	mme	gmme	gt
Proposed method	adaptive	**0.86**	**18**	21	85	20	670	878
KSP	Prop. method, adaptive	0.82	21	66	60	35	678	877

The best MOTA and MOTP scores are highlighted in bold.

background method used for earlier experiments was unsuccessful in generating segmentations with sufficient quality and was therefore not used here.

Compared to the results presented in Fig. 5(b) of [24], in which the terrace dataset is referred to as 'laboratory', performance in our experiments is lower. The MOTA score of KSP presented in [24] is about 0.95. This performance difference most likely results from the foreground segmentation used. The segmentation used in our experiments contains several artifacts when strong illumination changes occur and people cannot be distinguished from the background. Since parameter settings for KSP are as similar to the ones used in [24] as possible, it is likely that the segmentations used in [24] are of higher quality, resulting in better tracking performance. Unfortunately, we were unable to acquire the segmentations used by the original authors, making it impossible to verify this. However, since better segmentations would benefit KSP as well as the proposed method, we assume performance between both methods would still be comparable.

6. Discussion

A volume carving approach to person tracking requires in principle only two overlapping cameras from different viewpoints. In practice, the resulting uncertainty regarding where the persons might be becomes too large in this case, even for a low person density scenario (e.g. train station dataset). We consider 3–4 strategically placed cameras the minimum for the proposed approach. The use of additional cameras will improve reconstruction performance, albeit with decreasing added benefit. While the time needed for volume carving increases linearly with the number of cameras, the reduced number of hypotheses can compensate for this effect. Missing foreground regions in one camera can furthermore result in holes in the 3D reconstruction. In the experiments, this effect is minor for the adaptive backgrounds, but can severely impact performance for the static backgrounds. Reconstruction can be improved when taking a probabilistic approach to volume carving as was done in [49].

The requirement of camera synchronization can be relaxed. For the novel datasets presented in this paper (i.e. train station, hall), frame-synchronized cameras were used. However, the cameras recording the APIDIS dataset were not synchronized; synchronization was done artificially, duplicating frames to match frame numbers and timestamps between cameras. While this is less accurate, results show it is sufficient given the resolution of the volume space and the amount of motion between frames. In a real-time setting, using the most recent frame from each camera should be adequate synchronization for the purpose of person tracking.

All methods are implemented in C^{++}.[5] Experiments were performed on a 2.1 GHz CPU and 4 GB RAM. Computation time was measured on a 960 frame sequence with 4 persons. KSP took 8.9 seconds per frame (s/f) while KSP-App needed 9.7 s/f. Of these times, 8.8 s/f is used by POM (this is longer than stated in [16] because we use more ground plane locations and higher resolution images).

RCTA and the proposed method perform detection and tracking at 6.5 s/f. Of this time, about 0.6 s is spent on preprocessing (loading images, removing lens distortion, foreground segmentation, volume carving). The volume space resolution used is low to decrease computation time. A more advanced (GPU) implementation (e.g. [49]) could make computation of the volume space more efficient and enable higher resolution volume spaces, improving tracking accuracy. The *preselection* step takes about 0.4 s to compute. Computation of object visibility and appearance information for all hypotheses in the *verification* step requires the majority of processing time. Efficiency of this computation has been improved by rendering the 2D silhouettes of all individual objects used during *verification* only once. Object visibility and appearance are computed by combining the 2D silhouettes using precomputed depth information. This takes about 0.11 s per hypothesis (4.4 s total with $K = 40$). Since these operations are highly parallelizable, a GPU implementation could significantly improve performance. The remaining 1.1 s is spent on the computation of the remaining features and overhead such as generating image output.

7. Conclusion

We proposed a novel two-step method for the joint estimation of person position and track assignment in the context of a multi-person tracking system. The method leverages the possibilities offered by an overlapping camera setup, using multi-view appearance models and occlusion information.

The proposed method was compared to several state-of-the-art methods and different types of background estimation. In scenes with lower person densities, when all methods use the same static background, KSP-based methods have an edge over the proposed recursive method. This not surprising given batch-methods can take advantage of information available from multiple time instants, possibly extending into the future. It turns out however,

[5] A KSP implementation was kindly provided by the original authors. For KSP-App we used our own implementation as it could not be made available.

that by pairing the proposed recursive tracker up with adaptive backgrounds obtained from a feedback loop, it can outperform the KSP-based batch methods with static backgrounds. Interestingly, for higher person-density scenarios, KSP-based methods suffer from the limitations of the POM detector when persons overlap in many cameras. The proposed method with adaptive backgrounds outperforms the latter by a MOTA score of 0.21.

Furthermore, experiments on the publicly available APIDIS dataset [47] show that the proposed method can also outperform a very recent volume reconstruction-based method by [32]. These results show the benefit of using a flexible position likelihood map in combination with the proposed foreground likelihood, offering more flexibility when creating and deleting tracks. Improved track consistency shows the benefit of making appearance an integral part of the objective function.

Results could be improved by taking into account multiple track-to-detection hypotheses instead of using the maximum likelihood solution over time. Furthermore, the 3D scene reconstruction could be used to create more discriminative appearance models for better track disambiguation with techniques such as in [50,51].

Acknowledgments

This research has received funding from the EC's Seventh Framework Programme under Grant agreement number 218197, the ADABTS project.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cviu.2014.06.003.

References

[1] R. Pflugfelder, H. Bischof, People tracking across two distant self-calibrated cameras, in: Proc. of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2007, pp. 393–398.

[2] M. Pollefeys, R. Koch, L.V. Gool, Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters, Int. J. Comput. Vis. 32 (1) (1999) 7–25.

[3] B. Benfold, I. Reid, Stable multi-target tracking in real-time surveillance video, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3457–3464.

[4] B. Leibe, K. Schindler, N. Cornelis, L.V. Gool, Coupled object detection and tracking from static cameras and moving vehicles, IEEE Trans. Pattern Anal. Mach. Intell. 30 (10) (2008) 1683–1698.

[5] Z. Wu, A. Thangali, S. Sclaroff, M. Betke, Coupling detection and data association for multiple object tracking, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1948–1955.

[6] B. Yang, R. Nevatia, Multi-target tracking by online learning of non-linear motion patterns and robust appearance models, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1918–1925.

[7] A. Andriyenko, S. Roth, K. Schindler, An analytical formulation of global occlusion reasoning for multi-target tracking, in: Proc. of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2011, pp. 1839–1846.

[8] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool, Online multiperson tracking-by-detection from a single, uncalibrated camera, IEEE Trans. Pattern Anal. Mach. Intell. 33 (9) (2011) 1820–1833.

[9] R. Collins, Mean-shift blob tracking through scale space, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2, IEEE, 2003, pp. II-234–240.

[10] D.M. Gavrila, The visual analysis of human movement: a survey, Comput. Vis. Image Understand. 73 (1) (1999) 82–98.

[11] A. Mittal, L. Davis, M 2 tracker: a multi-view approach to segmenting and tracking people in a cluttered scene, Int. J. Comput. Vis. 51 (3) (2003) 189–203.

[12] R. Eshel, Y. Moses, Tracking in a dense crowd using multiple cameras, Int. J. Comput. Vis. 88 (1) (2010) 129–143.

[13] S. Khan, M. Shah, Tracking multiple occluding people by localizing on multiple scene planes, IEEE Trans. Pattern Anal. Mach. Intell. 31 (3) (2009) 505–519.

[14] D. Arsić, E. Hristov, N. Lehment, B. Hornler, B. Schuller, G. Rigoll, Applying multi layer homography for multi camera person tracking, in: Proc. of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), IEEE, 2008, pp. 1–9.

[15] T.T. Santos, C.H. Morimoto, Multiple camera people detection and tracking using support integration, Pattern Recognit. Lett. 32 (1) (2011) 47–55.

[16] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multicamera people tracking with a probabilistic occupancy map, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 267–282.

[17] J. Berclaz, F. Fleuret, P. Fua, Principled detection-by-classification from multiple views, in: Proc. of the International Conference on Computer Vision Theory and Application (VISAPP), vol. 2, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008, pp. 375–382.

[18] C.-C. Huang, S.-J. Wang, A Bayesian hierarchical framework for multitarget labeling and correspondence with ghost suppression over multicamera surveillance system, IEEE Trans. Autom. Sci. Eng. 9 (1) (2012) 16–30.

[19] L. Guan, J.-S. Franco, M. Pollefeys, Multi-view occlusion reasoning for probabilistic silhouette-based dynamic scene reconstruction, Int. J. Comput. Vis. 90 (3) (2010) 283–303.

[20] K. Otsuka, N. Mukawa, Multiview occlusion analysis for tracking densely populated objects based on 2-d visual angles, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2004, pp. I-90–I-97.

[21] R. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1) (1960) 35–45.

[22] M. Liem, D.M. Gavrila, Multi-person localization and track assignment in overlapping camera views, in: Proc. of the DAGM Symposium on Pattern Recognition, No. 6835 in Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2011, pp. 173–183.

[23] H. Ben Shitrit, J. Berclaz, F. Fleuret, P. Fua, Tracking multiple people under global appearance constraints, in: Proc. of the IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 137–144.

[24] J. Berclaz, F. Fleuret, E. Turetken, P. Fua, Multiple object tracking using K-shortest paths optimization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (9) (2011) 1806–1819.

[25] J.K. Wolf, A.M. Viterbi, G.S. Dixon, Finding the best set of K paths through a trellis with application to multitarget tracking, IEEE Trans. Aerospace Electron. Syst. 25 (2) (1989) 287–296.

[26] L. Zhang, Y. Li, R. Nevatia, Global data association for multi-object tracking using network flows, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2008, pp. 1–8.

[27] S. Calderara, R. Cucchiara, A. Prati, Bayesian-competitive consistent labeling for people surveillance, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 354–360.

[28] W. Du, J. Piater, Multi-camera people tracking by collaborative particle filters and principal axis-based integration, in: Proc. of the Asian Conference on Computer Vision (ACCV), Lecture Notes in Computer Science, vol. 4843, Springer, Berlin Heidelberg, 2007, pp. 365–374.

[29] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, S. Maybank, Principal axis-based correspondence between multiple cameras for people tracking, IEEE Trans. Pattern Anal. Mach. Intell. 28 (4) (2006) 663–671.

[30] J. Kang, I. Cohen, G. Medioni, Tracking people in crowded scenes across multiple cameras, in: Proc. of the Asian Conference on Computer Vision (ACCV), vol. 7, Asian Federation of Computer Vision Societies, 2004, p. 15.

[31] K. Kim, L. Davis, Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering, in: Proc. of the European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, vol. 3953, Springer, Berlin Heidelberg, 2006, pp. 98–109.

[32] H. Possegger, S. Sternig, T. Mauthner, P. Roth, H. Bischof, Robust real-time tracking of multiple objects by volumetric mass densities, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 2395–2402.

[33] T. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, IEEE J. Ocean. Eng. 8 (3) (1983) 173–184.

[34] D. Reid, An algorithm for tracking multiple targets, IEEE Trans. Autom. Control 24 (6) (1979) 843–854.

[35] T. Huang, S. Russell, Object identification in a Bayesian context, in: Proc. of the International Joint Conference on Artificial Intelligence, vol. 97, Morgan Kaufmann Publishers Inc. San Francisco, 1997, pp. 1276–1282.

[36] L. Leal-Taixe, G. Pons-Moll, B. Rosenhahn, Branch-and-price global optimization for multi-view multi-target tracking, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1987–1994.

[37] M. Hofmann, D. Wolf, G. Rigoll, Hypergraphs for joint multi-view reconstruction and multi-object tracking, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3650–3657.

[38] D. Baltieri, R. Vezzani, R. Cucchiara, A. Utasi, C. Benedek, T. Sziranyi, Multi-view people surveillance using 3D information, in: Proc. of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2011, pp. 1817–1824.

[39] M.C. Liem, D.M. Gavrila, A comparative study on multi-person tracking using overlapping cameras, in: Proc. of the International Conference on Computer Vision Systems, No. 7963 in Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2013, pp. 203–212.

[40] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[41] W. Zajdel, Z. Zivkovic, B.J. Krose, Keeping track of humans: Have I seen this person before? in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2005, pp. 2081–2086.

[42] K. Murty, An algorithm for ranking all the assignments in order of increasing cost, Oper. Res. 16 (3) (1968) 682–687.

[43] Z. Zivkovic, F. van der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern Recognit. Lett. 27 (7) (2006) 773–780.

[44] M. Harville, A framework for high-level feedback to adaptive, per-pixel, Mixture-of-Gaussian background models, in: Proc. of the European Conference on Computer Vision (eccv), No. 2352 in Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2002, pp. 543–560.

[45] N.M. Oliver, B. Rosario, A.P. Pentland, A Bayesian computer vision system for modeling human interactions, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 831–843.

[46] A. Sobral, BGSLibrary: an OpenCV C++ background subtraction library, in: IX Workshop de Visão Computacional (WVC), 2013. <http://code.google.com/p/bgslibrary/>.

[47] F. Chen, D. Delannay, C. De Vleeschouwer, An autonomous framework to produce and distribute personalized team-sport video summaries: a basketball case study, IEEE Trans. Multimedia 13 (6) (2011) 1381–1394.

[48] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the CLEAR MOT metrics, EURASIP J. Image Video Process. 2008 (2008) 1–10.

[49] T. Haufmann, A. Brodtkorb, A. Berge, A. Kim, Real-time online camera synchronization for volume carving on GPU, in: Proc. of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2013, pp. 288–293.

[50] M. Hofmann, D.M. Gavrila, 3D human shape model adaptation by automatic frame selection and batch-mode optimization, Comput. Vis. Image Understand. 115 (11) (2011) 1559–1570.

[51] M.C. Liem, D.M. Gavrila, Person appearance modeling and orientation estimation using spherical harmonics, in: Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition, IEEE, 2013, pp. 1–6.