Journal of Electronic Imaging

JElectronicImaging.org

Occlusion handling framework for tracking in smart camera networks by per-target assistance task assignment

Nyan Bo Bo
Francis Deboeverie
Peter Veelaert
Wilfried Philips

SPIE. IS&T imaging.org

Nyan Bo Bo, Francis Deboeverie, Peter Veelaert, Wilfried Philips, "Occlusion handling framework for tracking in smart camera networks by per-target assistance task assignment," *J. Electron. Imaging* **26**(5), 051407 (2017), doi: 10.1117/1.JEI.26.5.051407.

Occlusion handling framework for tracking in smart camera networks by per-target assistance task assignment

Nyan Bo Bo,[a,*] Francis Deboeverie,[a] Peter Veelaert,[b] and Wilfried Philips[a]
[a]imec-IPI-UGent, Ghent, Belgium
[b]Flanders MAKE-IPI-UGent, Ghent, Belgium

Abstract. Occlusion is one of the most difficult challenges in the area of visual tracking. We propose an occlusion handling framework to improve the performance of local tracking in a smart camera view in a multicamera network. We formulate an extensible energy function to quantify the quality of a camera's observation of a particular target by taking into account both person–person and object–person occlusion. Using this energy function, a smart camera assesses the quality of observations over all targets being tracked. When it cannot adequately observe of a target, a smart camera estimates the quality of observation of the target from view points of other assisting cameras. If a camera with better observation of the target is found, the tracking task of the target is carried out with the assistance of that camera. In our framework, only positions of persons being tracked are exchanged between smart cameras. Thus, communication bandwidth requirement is very low. Performance evaluation of our method on challenging video sequences with frequent and severe occlusions shows that the accuracy of a baseline tracker is considerably improved. We also report the performance comparison to the state-of-the-art trackers in which our method outperforms. *© 2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.26.5.051407]

Keywords: occlusion handling; assistance on demand; multitarget tracking; camera selection; smart camera network.
Paper 160982SS received Nov. 23, 2016; accepted for publication Jun. 28, 2017; published online Aug. 4, 2017.

1 Introduction

Nowadays, intelligent visual surveillance is one of the most active research areas in computer vision due to a high demand in automatic extraction of useful information from a large amount of videos captured by surveillance cameras. Computer vision tasks in intelligent visual surveillance include essential low-level tasks, such as detection, segmentation, and tracking of multiple targets as well as high-level tasks such as behavior analysis, activity recognition, and so on. Among these tasks, visual tracking of multiple people in an uncontrolled environment is one of the most challenging tasks due to the nonrigid nature of the human body and object–person/person–person occlusion. When multiple people are moving around in the scene, which usually contains static occluders, a person may sometimes be occluded by another person(s) or object(s) in the scene in a particular camera view. Many occlusion handling techniques have been proposed in visual tracking literature.

Despite being limited to a fixed single view, many single camera tracking systems[1–7] are able to track occluded targets reliably using various occlusion prediction and handling techniques. The limitation of single view tracking systems can be avoided by deploying a network of cameras with overlapping views. Since observations of a target from different viewpoints are available, local tracking in each camera view can take advantage using these observations in a smarter way to handle occlusion. Therefore, as a first contribution in this paper, we propose an occlusion handling framework for local tracking in the view of a smart camera by requesting assistance of other smart cameras that have better observations over occluded targets. In this way, cameras assist each other to improve their local tracking accuracy while reducing computational power requirement. Similar to the aforementioned multicamera trackers,[8,9] local tracking results can be fused to produce more accurate joint position estimates. Some camera networks contain battery operated smart cameras that usually communicate wirelessly with other cameras in the network. If battery life is a constraint, these cameras can only be used as assisting cameras as they can be switched to a standby power saving mode when their assistance is not needed.

The reliability of an observation of a target from a particular camera view must be first quantified to check if a reasonably accurate estimation of the target's state can be made from the current view. If not, the same reliability measurement is made in the viewpoints of assisting cameras where the job of tracking the target is assigned to the camera with the highest reliability. For this purpose, we propose an energy function that incorporates object–person occlusion, person–person occlusion, as well as the coverage of field of view (FOV) into a single number, which is a second contribution of this paper. The proposed energy function can be computed locally in a smart camera at runtime without additional information from other cameras. Therefore, no intercamera communication is required in camera selection process. However, once the best suited camera for assisting in occlusion handling is selected, transmission and receiving of the positions of persons being tracked are required. Since this intercamera communication involves an exchange of the positions of persons (usually two numbers per person), the

*Address all correspondence to: Nyan Bo Bo, E-mail: Nyan.BoBo@ugent.be

bandwidth requirement is significantly lower compared to sending complete images, appearance models, feature points, and so on. Depending on the nature of a baseline tracker, our energy function can be extended by taking into account other terms such as the size of a target in the camera view, the camera pitch angle, and so on. The low computational complexity of the proposed energy function opens the possibility of real-time tracking in a smart camera network.

As a third contribution, we deploy the proposed framework on our previously introduced baseline tracker[10] and assess the improvement in terms of both local tracking accuracy and precision. The entire tracking system is evaluated on a self-captured indoor video as well as on outdoor videos from publicly available datasets, such as CVLab-EPFL,[11] PETS2009,[12] and UvA train station.[13] The evaluation results show that deploying our occlusion handling on the baseline tracker[10] considerably improves the accuracy. On the S2.L1 video of the PETS2009 dataset, which is widely used as a benchmark in literature, our tracker outperforms state-of-the-art techniques[5,14–21] in terms of multiple object tracking accuracy (MOTA).

The remainder of this paper is organized as follows. In Sec. 2, we provide an overview of existing methods for occlusion handling in visual tracking. Then, we give the detailed description of the proposed occlusion handling method in Sec. 3. A systematic evaluation of the performance of our framework and a discussion of results are presented in Secs. 4 and 5, respectively. Finally, our paper is concluded in Sec. 6.

2 Related Work

Despite the large amount of research performed for several decades, a robust solution to visual tracking of multiple targets remains challenging in many realistic scenarios and applications. One of the most difficult problems in visual tracking stems from the fact that even a simple target interacting with other target(s) or static object(s) in the scene may cause severe occlusions that negatively impact the performance of a tracker. In single camera tracking, various approaches have been proposed to handle the occlusion problem. Some trackers[1,2] merge targets that are involved in the occlusion (both occluding and occluded targets) and track them as one target. When occlusion is over, merged targets are split and reidentified using image region association methods, such as temporal template[1] and region covariance.[2]

Another approach to handle occlusion in a monocular camera tracking system is to only track the nonoccluded parts of the target.[3–7] These trackers rely on a robust segmentation of occluding and occluded targets using color histograms[4,7] or single/mixture of Gaussian color models.[3,5,6] In case of total occlusion, i.e., a target is completely covered by other target(s) or objects(s), some methods[3,4,6] just pause tracking occluded targets and continue tracking nonoccluded targets only. The totally occluded targets are reidentified and tracked again once the occlusion is over. These methods are based on the concept of "object permanence,"[22] which suggests that a totally occluded target will reemerge near its occluder. Since totally occluded targets are not tracked (having just a rough idea that a target is somewhere behind the occluder), part of the trajectory of a target where the total occlusion occurred is not available. These missing parts of the trajectory are interpolated using motion models in Refs. 5 and 7. However, due to the lack of actual observations on a totally occluded target, an interpolated/predicted target's position based on a motion model may be far from the actual position of the target.

The problem of missing observations due to a total occlusion can be overcome using a network of cameras with overlapping views, observing targets in the scene from different viewpoints. A target that is completely occluded in a particular camera view may be fully or partially observable in one or more other camera views. Many multicamera tracking systems[14,15,23,24] directly rely on observations of targets in all camera views for an estimation of targets' positions. Despite not having explicit occlusion handling in these methods, their evaluation results show that these trackers are robust to occlusion. This implies that using observations from different viewpoints compensates for the missing observations of targets in occluded views with observations from nonoccluded views. However, these methods try to handle occlusion at a global level rather than at a local level in individual smart cameras. Some multicamera trackers[8,9] perform occlusion detection and handling by fusing local estimates to produce global estimates. The local tracking in the tracker of Gruenwedel et al.[8] uses probabilistic foreground modeling to segment targets involved in an occlusion as individual observations. In our previous work,[9] local tracking in a smart camera view pauses for targets that are occluded in their view, assuming that these targets will not be occluded in at least one of the other cameras. These approaches still do not take the advantage of available observations from other cameras for occlusion handling at a local level.

However, in practice it is not always necessary to track a target in all camera views, since observations in some cameras are sometimes redundant depending on the scene configuration. Based on this fact, Tessens et al. proposed a method to effectively determine a view that contributes most to the desired observation of a scene.[25] However, once the number of targets involved in the occlusion increases, the probability of a single camera observing all targets adequately is very small. Therefore, their later work[26] proposed a method to dynamically assign a subset with limited size of all available cameras to track each target. Their method simulates the observations from each camera view for current time instance using estimated positions of persons from previous time instance. Then, simulated observations from a combination of different cameras are tested using Dempster–Shafer theory of evidence framework[27,28] to find the camera combination that gives the position estimate with the lowest uncertainty. In contrast, our method directly analyzes how a particular person is occluded by other person(s)/object(s) and whether the person is fully/partially inside the view. Their proposed method is deployed on a tracker of Munoz-Salinas et al.,[29] which is also based on Dempster–Shafer framework,[27,28] to demonstrate that only using an automatically selected subset with a limited number of cameras for tracking each target can achieve the same accuracy as when using all cameras.

In this paper, the proposed occlusion handling framework is inspired by aforementioned multicamera trackers and best camera/subset of cameras selection methods in a way that each camera handles occlusion in its view with the help

of a selected subset of other cameras. The main goal of our work is to improve the local tracking accuracy in the view of a smart camera with the assistance of other cameras when the observations of targets in its view become inadequate. Local tracking in the view of a smart camera is limited to observations in its own view as long as the quality of the observations over all targets is sufficient. When the observations of a target become inadequate due to an occlusion, the smart camera decides which one of the other smart cameras has the best observation over the target and assigns the job of tracking the target to it. In our framework, we use previously estimated positions of all persons in the scene to assess the severity of person–person occlusion. The magnitude of object–person occlusion is assessed using object an occlusion map, which can be automatically built by existing techniques.[30–34]

3 Occlusion Handling by Per-Target Assistance Task Assignment

3.1 *System Setup and Notation*

In this paper, we consider tracking in the view a particular smart camera that is interconnected with other smart cameras which are observing a scene where multiple people are moving around. A smart camera can communicate with other cameras via wired/wireless communication. All smart cameras are intrinsically and extrinsically calibrated. In this work, we assume that all cameras know the calibration matrices of the other cameras in the network. To be clear, we will use the following terminology in the remainder of the paper: any camera in the network is called main camera if we are considering local tracking of that camera and all other cameras are called assisting cameras. Depending on the camera network configuration and the targeted application, local tracking can be set to run on all cameras, i.e., all cameras in the network are performing the task of both main camera and assisting camera. Furthermore, we define some basic notations

- C denotes the total number of smart cameras in the network and M denotes the total number of targets being tracked.
- $\mathbf{s} = (x, y)^T$ denotes a position on the ground plane in world coordinates and $\mathbf{r} = (i, j)^T$ denotes a position in image coordinates. Given a projection matrix R_c of a particular smart camera c, a position on the ground plane $\mathbf{s}$ is projected on the image plane as $\mathbf{r} = R_c\mathbf{s}$.

3.2 *Target State Estimation*

Local tracking of a smart camera c is achieved by estimating the positions of all persons in the scene at time t by maximizing the product of the likelihood of the positions $\mathbf{s}_t^1, \ldots \mathbf{s}_t^M$ of M persons, given the foreground image $F_{t,c}$ computed from its image as an observation, and a motion model $P(\mathbf{s}_t^1, \ldots \mathbf{s}_t^M | \mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M)$

$$P(F_{t,c}|\mathbf{s}_t^1, \ldots \mathbf{s}_t^M) P(\mathbf{s}_t^1, \ldots \mathbf{s}_t^M | \mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M). \tag{1}$$

However, not all persons may be adequately observed by the camera c due to occlusions. Let $\mathbf{v} = (v_0, \ldots v_N)^T : N \leq M$ contains the IDs of N targets for which observations from the camera c are adequate to make a reliable position estimation. Each of the remaining positions is estimated by other cameras in the network that have better observations. Positions of persons, whose IDs are in the vector $\mathbf{v}$, are estimated by maximizing the likelihood as follows:

$$\hat{\mathbf{s}}_t^{v_1}, \ldots \hat{\mathbf{s}}_t^{v_N} = \underset{\mathbf{s}_t^{v_1}, \ldots \mathbf{s}_t^{v_N}}{\arg\max} P(F_{t,c}|\mathbf{s}_t^1, \ldots \mathbf{s}_t^M) P(\mathbf{s}_t^1, \ldots \mathbf{s}_t^M | \mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M). \tag{2}$$

When observations of all targets in the view of camera c are adequate, the IDs of all M persons are in the vector $\mathbf{v}$, thus $N = M$.

Local tracking in the view of camera c (main camera) assesses if the quality of observation of a person m from its view is adequate for reliable estimation of the target's position. If the observation is inadequate, the camera checks if one other camera in the network has better observation over the person m. In case that a camera $\hat{c}$ with better observation over the person m is found in the network, the main camera requests the camera $\hat{c}$ to assist in the estimation of the position of the person m. In order to give a clear overview, pseudocode of our proposed method is presented in Algorithm 1. The detailed descriptions on how to quantify the quality of an observation over a target from the viewpoint of a particular camera, i.e., computation of observation energy E in lines 6 and 9 of the Algorithm 1, are presented in the following Sec. 3.3. In Sec. 3.4, the process of how the main camera assigns the tracking task of one or more targets to a particular assisting camera, i.e., lines 30 and 31 of the Algorithm 1 is thoroughly discussed.

3.3 *Energy Function for Camera's Observation Quality Assessment*

In a recursive tracking approach, the current position $\mathbf{s}_t^m$ of a target m is estimated based on current observations and prior knowledge of previous position $\mathbf{s}_{t-1}^m$ of the target at time $t-1$. Due to physical constraints on human movement, a person can only move a small distance in a single frame interval. Therefore, the suitability for estimating the position of a person in the next frame from a particular camera view can be assessed from the currently estimated position of the person. The quality of observation of a target in a particular camera view is degraded by the presence of other target(s) as well as static object(s), such as furniture between the target and the camera. Moreover, the observation quality is also considered low if it is partially/fully outside the FOV of the camera, even though nothing is occluding the target. Figure 1 shows all three types of scenarios that degrade the quality of observation. Targets are shown as rectangular boxes in the FOV of a camera. A static object, shown as an octagram, partially occludes target 1 while target 3 is occluding target 2. There is no occlusion on target 4 but more than half of the target is outside the FOV.

In a particular camera view, when static objects such as furniture (indoor), lamp posts (outdoor), etc. partially/fully cover a target, object–person occlusion occurs. The word "static" is used because these objects never move (e.g., lamp post) or infrequently move (e.g., chairs) compared to the movement of targets. These static objects must be located first so that the magnitude of their occlusion over targets can be assessed. Many approaches[30–33] have been

Algorithm 1 Occlusion handling by per-target tracking task assignment.

Input: Estimated positions at previous time instance $S_{t-1}=\{\mathbf{s}_{t-1}^1,\ldots\mathbf{s}_{t-1}^M\}$

Output: Newly estimated position at current time instance $S_t=\{\mathbf{s}_t^1,\ldots\mathbf{s}_t^M\}$

for each person ID m **in** $\{1,\ldots M\}$ **do**

  Set $\hat{c}$ as the main camera c_{main} m as main camera c_{main}

  Compute the observation energy $E(c_{\text{main}},\mathbf{s}_{t-1}^m)$ of the main camera's view

  for each assisting camera c **do**

    Compute the observation energy $E(c,\mathbf{s}_{t-1}^m)$ of the assisting camera's view

    if $E(c_{\text{main}},\mathbf{s}_{t-1}^m) > E(c,\mathbf{s}_{t-1}^m)$ **then**

      Set $\hat{c}$ as the assisting camera c

    end if

  end for

  if $\hat{c}$ **is** c_{main} **then**

    Add m to $\mathbf{v}$

  else

    Add m to $\mathbf{u}_{\hat{c}}$

  end if

end for

if $\mathbf{v}$ **is** not empty **then**

  Estimate the new position of each person in $\mathbf{v}$ on the *main camera* resulting $S_t^{\text{main}}=\{\mathbf{s}_t^v:\forall v\in\mathbf{v}\}$

end if

Initialize S_t^{assist} as empty set: $S_t^{\text{assist}}=\phi$

for each assisting camera c **do**

  if $\mathbf{u}_c$ **is** not empty **then**

    Request the *assisting camera* c to estimate the new position of each person in $\mathbf{u}_c$

    Receive the newly estimated position of each person in $\mathbf{u}_c$ from the *assisting camera* as $S_t^c=\{\mathbf{s}_t^u:\forall u\in\mathbf{u}_c\}$

    Add S_t^c to S_t^{assist}: $S_t^{\text{assist}}=S_t^{\text{assist}}\cup S_t^c$

  end if

end for

$S_t=S_t^{\text{main}}\cup S_t^c$

return S_t

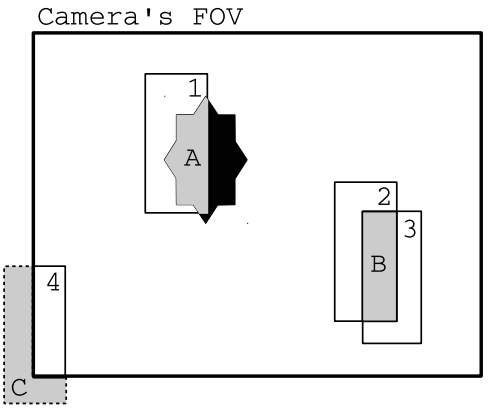


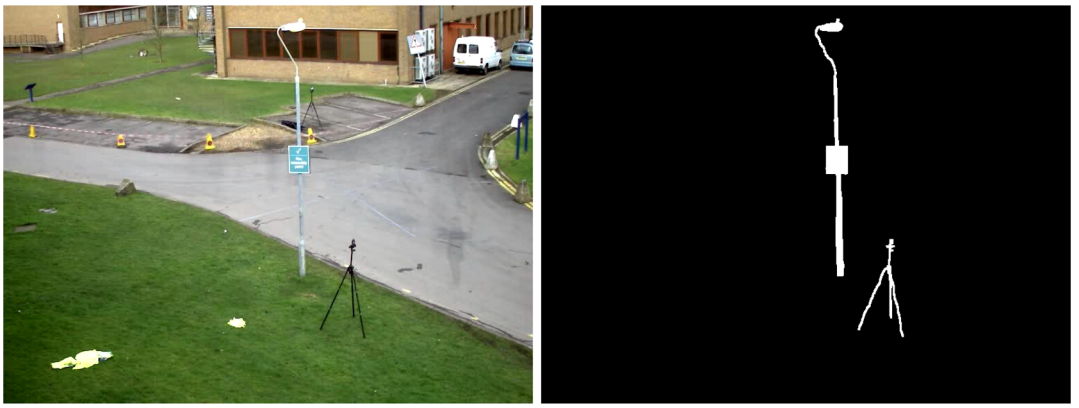
Fig. 1 Example of three scenarios that degrade the quality of camera observation.

proposed to detect static occluding objects automatically. These approaches construct a pixel-based occlusion mask, similar to a binary image, shown in Fig. 2(b), for a camera view that is a binary decision for each pixel if it is occluded by a static object or not. A scene in which a lamp post and a tripod located between the road (where people usually walk) and the camera is shown in Fig. 2(a). An occlusion mask as shown in Fig. 2(b) can be automatically[30–33]/manually constructed from a scene as shown in Fig. 2(a) to locate the static objects in the image.

Once static objects in the scene are located, we can verify that a person m at position $\mathbf{s}_t^m$ is occluded by any static object. It is done by projecting a three-dimensional (3-D) model of a person (cuboid, cylinder, etc.) placed at $\mathbf{s}_t^m$ on the image plane of a camera c using calibration matrices. This projected region on the image plane of camera c is denoted as $\omega(\mathbf{s}_t^m,c)$. If $\omega(\mathbf{s}_t^m,c)$ overlaps with any image region where static objects are detected, then the person m is considered as a target occluded by a static object. The energy $E_{\text{obj}}(c,\mathbf{s}_t^m)$ of this object–person occlusion must be quantified in order to compare with other views. Moreover, the energy $E_{\text{obj}}(c,\mathbf{s}_t^m)$ must increase as more body parts of the person m is covered by a static object, i.e., directly proportional to the number of pixels of a static object within $\omega(\mathbf{s}_t^m,c)$. Therefore, it is defined as the ratio of the number of pixels occluded by a static object (lightly shaded region "A" of the octagram in Fig. 1) to the total number of target's pixels (rectangular region of target 1). Therefore, the energy of static occlusion $E_{\text{obj}}(c,\mathbf{s}_t^m)$ is computed as

$$E_{\text{obj}}(c,\mathbf{s}_t^m)=\frac{\sum_{r\in\omega(\mathbf{s}_t^m,c)}O(r,c)}{|\omega(\mathbf{s}_t^m,c)|},\tag{3}$$

where $O(r,c)$ is a binary occlusion mask for a camera c in which pixels occluded by one or more static objects have value $O(r,c)=1$ and otherwise $O(r,c)=0$. An operator $|.|$ computes the area of any geometric shape. When there is no static object occluding a person m in the view of camera c, the energy of static occlusion is the lowest, i.e., $E_{\text{obj}}(c,\mathbf{s}_t^m)=0$. It reaches its maximum value, i.e., $E_{\text{obj}}(c,\mathbf{s}_t^m)=1$, if a person m is completely occluded. A more detailed voxel-based (3-D) occlusion map can



(a) (b)

Fig. 2 Example of (a) original image and (b) occlusion mask O.

automatically be constructed using state-of-the-art methods such a method proposed by Slembrouck et al.[34]

A person–person occlusion occurs when one person is fully or partially blocking an observation of another person. Therefore, the energy of person–person occlusion over a person for a particular camera view is clearly depending on the position of other persons in the scene. If a scene contains other moving objects such as cars, these must also be tracked to know their positions for dynamic object occlusion detection and handling. There exist many state-of-the-art vehicle trackers[35,36] in the literature. Once the positions of all persons and other moving objects (for example, cars) are known, we can simulate how these targets may appear in the view of each camera to check whether persons are occluding each other or a car is occluding one or more persons. However, in this work, we consider a scenario that persons are the only moving targets in the scene. To assess person–person occlusion, 3-D models of a person (cuboid, cylinder, etc.) are placed at positions of all M persons being tracked $\mathbf{s}_t^1, \ldots \mathbf{s}_t^M$, and these 3-D models are projected on the image plane of a camera c as $\omega(\mathbf{s}_t^1, c), \ldots \omega(\mathbf{s}_t^M, c)$. For simplicity, we denote $\Omega(c) = \{\omega(\mathbf{s}_t^1, c), \ldots \omega(\mathbf{s}_t^M, c)\}$ as a set of projections of 3-D models of all M persons.

A person m at position $\mathbf{s}_t^m$ is possibly occluded by one or more other persons if the projection $\omega(\mathbf{s}_t^m, c)$ of the person m overlaps with the union of the projections of all other persons. The degree of possible occlusion is

$$\omega_{\text{per}}(\mathbf{s}_t^m, c) = \omega(\mathbf{s}_t^m, c) \cap \bigcup_{\omega \in [\Omega(c) \setminus \mathbf{s}_t^m]} \omega. \tag{4}$$

The area of $\omega_{\text{per}}(\mathbf{s}_t^m, c)$ increases as more body parts of the person m is covered by other persons in the view of a camera c. In Fig. 1, this overlapping region $\omega_{\text{per}}(\mathbf{s}_t^m, c)$ is shown as a shaded region "C" where the rectangles of targets 2 and 3 intersect. Since the energy of the person–person occlusion $E_{\text{per}}(c, \mathbf{s}_t^m)$ must be directly proportional to the area of $\omega_{\text{per}}(\mathbf{s}_t^m, c)$, it is defined as the ratio of the number of pixels in the projection overlap to the projection of a person m

$$E_{\text{per}}(c, \mathbf{s}_t^m) = \frac{|\omega_{\text{per}}(\mathbf{s}_t^m, c)|}{|\omega(\mathbf{s}_t^m, c)|}. \tag{5}$$

When a person m is completely occluded by one or more other persons, the energy of person–person occlusion is at its highest, i.e., $E_{\text{per}}(c, \mathbf{s}_t^m) = 1$. The person–person occlusion energy is at its lowest, i.e., $E_{\text{per}}(c, \mathbf{s}_t^m) = 0$ if no potential person–person occlusion exists for person m.

In a particular camera view, although a target is free from both static and dynamic occlusion, the quality of observation can still be inadequate if the target is not fully inside a camera's FOV. This scenario is shown in Fig. 1 where more than half of target 4 is outside the camera's FOV. A systematic way of penalizing must be formulated to avoid estimating a target's position from a view in which the target is partially out of the FOV, whenever possible. Moreover, the calibration accuracy is usually higher in the center of a view. The energy of a target observed outside the FOV $E_{\text{FOV}}(c, \mathbf{s}_t^m)$ must increase as more body parts of the person m are outside of the FOV of the camera c. Thus, it is defined as the ratio of the number of target's pixels outside of the camera's FOV (shaded region "B" of target 4 outside the camera's FOV in Fig. 1) to the total number of target's pixels. It is computed as

$$E_{\text{fov}}(c, \mathbf{s}_t^m) = 1 - \frac{|\omega(\mathbf{s}_t^m, c) \cap R_{\text{img}}(c)|}{|\omega(\mathbf{s}_t^m, c)|}, \tag{6}$$

where $R_{\text{img}}(c)$ is an FOV rectangle on the image plane of a camera c. When a target is completely outside the FOV, $E_{\text{FOV}}(c, \mathbf{s}_t^m) = 1$. If the entire target is inside the FOV, $E_{\text{FOV}}(c, \mathbf{s}_t^m) = 0$.

All three energies discussed so far must be aggregated into a single number so that the observation quality from different view points over a target can be compared for a selection of the most suitable view. In this paper, the proposed aggregated energy $E(c, \mathbf{s}_t^m)$ is the sum of three individual energies

$$E(c, \mathbf{s}_t^m) = E_{\text{obj}}(c, \mathbf{s}_t^m) + E_{\text{per}}(c, \mathbf{s}_t^m) + E_{\text{FOV}}(c, \mathbf{s}_t^m). \tag{7}$$

In practical applications, it is possible to multiply each energy term with different weights to fine tune the tracking performance for a specific environment. However, in this paper, we use an equally weighted sum of the energies terms as in Eq. (7) to keep our framework more general and maintain lesser parameters to be tuned. If a target is completely inside the FOV of a camera with no occlusion, the aggregated energy is the lowest; i.e., $E(c, \mathbf{s}_t^m) = 0$. A high $E(c, \mathbf{s}_t^m)$ indicates a low quality of observation over a target

m from the view of a camera c. Given the position $\mathbf{s}_t^m$ of a person m, the optimal camera view for tracking a person m can be found as

$$\hat{c} = \arg\min_{c} E(c, \mathbf{s}_t^m). \tag{8}$$

Once a suitable assisting camera $\hat{c}$ for tracking a target m is found by Eq. (8), a tracking task for the person m is transferred from the main camera to the assisting camera $\hat{c}$. Assuming that calibration matrices and static occlusion mask of all assisting cameras are preloaded, Eq. (7) can be computed locally in the main camera without additional information from any of the assisting cameras. Therefore, our proposed camera selection does not put any additional load to intercamera communication. The following section discusses how an intercamera tracking task transfer is performed.

3.4 *Intercamera Tracking Task Assistance*

In our framework, intercamera tracking task assistance and data exchange involve sending and receiving a set of persons' positions with respective IDs, i.e., $\mathbf{s} = (x, y)^T$ and $m \in \{1 \ldots M\}$. The main camera also sends IDs of the persons whose positions are to be estimated by an assisting camera $\hat{c}$. Let's recall Eq. (2) where the vector $\mathbf{v} = (v_0, \ldots v_N)^T : N \leq M$ contains IDs of N persons whose positions can be reliably estimated by a camera c. Suppose that positions of the remaining persons whose IDs are in a vector $\mathbf{u} = (u_0, \ldots u_{M-N})^T$ can be reliably estimated by an assisting camera $\hat{c}$ according to Eq. (8). Then, the camera c sends previously estimated positions $\{\mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M\}$ and IDs in the vector $\mathbf{u}$ to the assisting camera $\hat{c}$. Using previously estimated positions $\{\mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M\}$ as prior, the assisting camera $\hat{c}$ estimates new positions of persons whose IDs are in $\mathbf{u}$ as

$$\hat{\mathbf{s}}_t^{\mathbf{u}_0}, \ldots \hat{\mathbf{s}}_t^{\mathbf{u}_{M-N}} = \arg\max_{\mathbf{s}_t^{\mathbf{u}_0}, \ldots \mathbf{s}_t^{\mathbf{u}_{M-N}}} P(F_{t,\hat{c}} | \mathbf{s}_t^1, \ldots \mathbf{s}_t^M) P(\mathbf{s}_t^1, \ldots \mathbf{s}_t^M | \mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M), \tag{9}$$

where $F_{t,\hat{c}}$ is a foreground image captured and computed by the assisting camera $\hat{c}$. Since only positions and IDs are exchanged between the main camera and the assisting cameras, communication requirement in our framework is very low.

3.5 *Implementation*

The proposed occlusion handling framework is integrated into our previously implemented recursive tracker[10] which tracks multiple persons in real time by recursively maximizing the likelihood of an observation given the positions of all persons. Foreground detected binary images computed by a texture-based foreground detection method[37] are used as observations in the likelihood computation. We simply use the likelihood function together with the default parameters reported in Ref. 10 as

$$\mathcal{L}_c(S_t) = P(F_{c,t} | \mathbf{s}_t^1, \ldots \mathbf{s}^M) = \prod_{\mathbf{r} \in \Omega(S_t)} (1 - \epsilon_f)^{F_{c,t}(\mathbf{r})} \epsilon_f^{1 - F_{c,t}(\mathbf{r})} \prod_{\mathbf{r} \notin \Omega(S_t)} (1 - \epsilon_b)^{1 - F_{c,t}(\mathbf{r})} \epsilon_b^{F_{c,t}(\mathbf{r})}, \tag{10}$$

where ϵ_f and ϵ_b are the probability of a true foreground pixel is detected as background and a true background pixel is detected as foreground, respectively. The search space for maximizing $P(F_{c,t} | \mathbf{s}_t^1, \ldots \mathbf{s}_t^M)$ is defined based on the known positions of all persons at the previous frame $\mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M$ and the fact that a person cannot move very far between two consecutive frames. Then, the real-time likelihood maximization is performed by applying a greedy search algorithm.

4 Performance Evaluation and Analysis

4.1 *Video Sequences for Evaluation*

To evaluate the performance of our tracker in an indoor environment, we use a video sequence captured at 20 fps in a room of 8.8×9.2 m^2. This video is captured using four cameras with overlapping views and it has a total duration of ~6 min. The resolution of the video is 780×580 pixels. Up to four people are walking in the scene and they often occlude each other. This video sequence will be called indoor video sequence in the rest of the paper. For the performance evaluation in an outdoor environment, we use the campus 1 video sequence, which is publicly available.[11] The video sequence is captured by Fleuret et al. for the performance evaluation of their tracker based on occupancy mapping.[38] Three DV cameras are used to capture the video sequence at 25 fps with a resolution of 360×288 pixels. Up to four people are simultaneously walking in front of the cameras. Due to the low camera pitch angle, this sequence is very challenging for accurate position estimation in a local tracking of a smart camera. For both video sequences, ground plane positions for each person have been manually annotated every 20 frames.

For performance comparison with the state-of-the-art trackers, we evaluate our tracker on the PETS2009 S2.L1 video sequence since it is widely used as a benchmark in multiperson tracking literature for both monocular and multicamera approaches. This video sequence is captured at a quite low frame rate of ~7 fps for 1.5 min and up to eight persons are present in the scene. Although the length of this sequence is relatively short, it contains various kinds of multipeople tracking challenges, such as frequent object–person/person–person occlusions and close proximity between persons. Manually annotated ground truth[39] is publicly available for this video. However, users are warned that the bounding boxes around people are not always perfectly aligned due to articulation, interpolation (only key frames are annotated and the bounding boxes for frames between them are interpolated), and annotator mistakes.

Moreover, we also evaluate our tracker on a more recent publicly available[13] Train Station dataset that is recorded on a platform of a train station. Up to four persons appear in the scene having dynamic background with trains passing by and people walking on the platform. Since, the lighting conditions vary significantly over time, the dataset becomes more challenging. This dataset is captured with three

cameras with overlapping views at 20 fps with the resolution of 752×560. The ground truth positions are manually annotated every five frames. Both intrinsic and extrinsic parameters of cameras in all video sequences used for performance evaluation are available. Furthermore, all video sequences are frame synchronized.

4.2 *Evaluation Criteria*

To the best of our knowledge, there is no single standardized evaluation metric to measure the performance in multitarget tracking up to now. We select CLEAR MOT metrics[40] out of many available metrics since they are the most widely used systematic evaluation metrics nowadays. Many state-of-the-art trackers[5,15–17,41] use these metrics to report their performance and using the same criteria allow us to make a straightforward comparison. CLEAR MOT metrics take into account all types of errors produced by multiple object tracking systems and summarize them into two metrics: the multiple object tracking precision (MOTP) and the MOTA. MOTP measures the total position error between the ground truth and the tracker's estimate pairs over all frames. This error measurement is averaged by the total number of ground truth and tracker's estimate pairs.

MOTA summarizes the number of object misses, false positives, and identity mismatches as a single number. The tracked person is considered as lost when the Euclidean distance between the ground truth and the tracker's estimate exceeds a threshold T_d. Ideally, it is desirable to have a tracker with high MOTP and MOTA. However, in some cases, one of the two metrics may be more important depending on the higher level applications that use the trajectories of the tracker. In human behavior analysis based on trajectories, MOTA is considered more important than MOTP, since tracking loss, identity switches, and false positives may impose negative effects on the trajectory analysis. On the other hand, in assistive applications such as automatic assisting visually impaired people MOTP may be more important performance indicator than MOTA since the system must be able to locate a person with a high precision for obstacle avoidance. However, MOTA is more robust to bias and mistakes in manual ground truth annotation. Therefore, we compare the performance of our tracker to state-of-the-art trackers in terms of MOTA only.

5 Results and Discussion

5.1 *Quantitative Evaluation*

We run our baseline tracker[10] on the aforementioned test video sequences once with and once without the proposed occlusion handling framework, to observe how much the framework improves the performance of the baseline tracker. The second and third columns of Table 1 show that MOTA of the baseline tracker is increased by ~3% on average for all test video sequences by deploying the proposed occlusion handling framework. However, there is only a small improvement in MOTP as shown in the last two columns of Table 1. The reason why occlusion problems have more effect on MOTA than on MOTP is that when severe occlusion occurs the tracker often loses the target rather than making wrong position estimates. Tracking losses increase the number of object misses (miss_t), which causes a decrease in MOTA ($\text{MOTA} \propto \frac{1}{\text{miss}_t}$). Moreover, lost targets are not included in the calculation of MOTP. Therefore, occlusion handling is not improving MOTP significantly. The MOTA improvement due to occlusion handling is a bit higher in campus 1 than in PETS2009 S2.L1 and indoor. The performance evaluation on the train station dataset shows the highest improvement in MOTA: from 82% to 92%. The reason is that the train station dataset contains very frequent and severe person–person occlusions and the main camera alone is not sufficient to estimate the positions of persons involved in most occlusion cases. Getting assistance from appropriate assisting cameras based on our method provides more accurate position estimates resulting in significant MOTA improvement.

Table 1 Comparison of MOTA and MOTP for the single camera and the assisted tracking by our proposed framework.

	MOTA		MOTP	
Video	Single	Assisted	Single	Assisted
Indoor	95%	98%	72%	71%
Campus 1	78%	83%	60%	61%
PETS2009 S2.L1	92%	94%	76%	77%
Train station	82%	92%	67%	71%

When we visually examine the tracking results, we find that our framework sometimes fails to handle the occlusion due to a low calibration accuracy of the cameras. An example of how assisted occlusion handling of our framework sometimes fails due to the low calibration accuracy of involving cameras is shown in Fig. 3. In the view of the main camera [Fig. 3(a)] target 3 is occluded by target 14 near the center of the scene. Therefore, the main camera searches for a camera with better view to track target 3 and finds that the assisting camera 3 is the best camera for the task. Then, the main camera assigns the tracking task to the assisting camera 3 using projective geometry. Unfortunately, due to the low calibration accuracy of the assisting camera 3, the projected bounding boxes in its view are not well aligned with the targets. Only about half of the bodies of targets 3 and 14 fall within the respective bounding boxes. This kind of misalignment is usually corrected by position estimation. However, in more severe case, the tracker sometimes drifts from a target's real position and completely loses the target. Although a numerical indication of the calibration accuracy of the video sequences is not available, we found that camera calibration accuracy of indoor and campus 1 video sequences is higher than PETS2009 S2.L1 by visually checking the projection of positions estimated by a particular camera onto an image plane of other cameras.

The expected calibration accuracy of the cameras for successful tracking task transfer is highly dependent of how well local tracker on the cameras can correct the bounding box misalignments. Moreover, as the size of both projected bounding box and the target varies depending on their position with respect to the position of the camera, it is not feasible to express expected accuracy (projecting error) in pixel. For instance, the projection error of 10 pixels may cause slight bounding box misalignment for the targets closer to

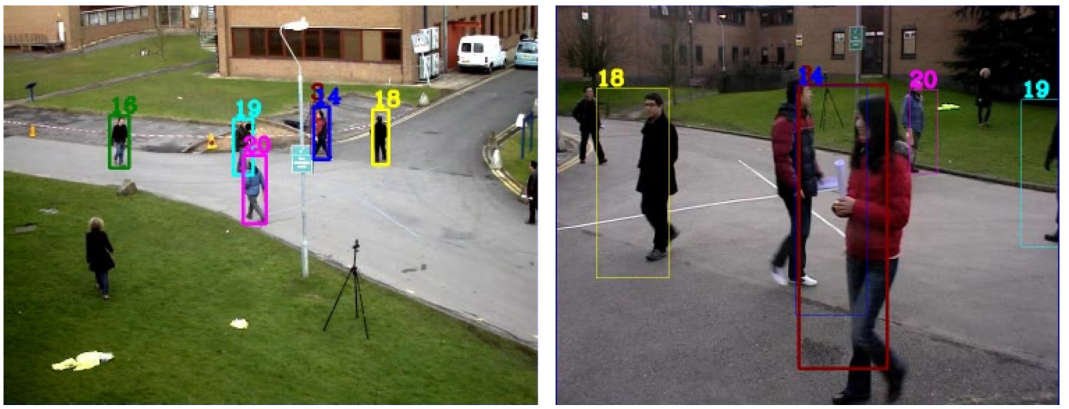


Fig. 3 Tracking task transfer of target 1 from (a) the main camera to (b) the assisting camera 3 which has a low calibration accuracy.

a camera whereas the projection error can cause complete bounding box misalignment (target is completely outside of the bounding box) for targets far away from the camera. The more appropriate accuracy measurement would be the percentage of the target's region covered by the projected bounding box. In our implementation, the local tracker[10] is mostly able to correct the bounding box misalignment if more than 50% of the target is covered by the projected bounding box.

The baseline tracker achieves relatively lower MOTA (78%) in campus 1 video sequence than the other two sequences due to its small camera pitch angle. A small camera pitch angle results in a less accurate 3-D position estimation and a more severe person–person occlusion (an occluded person can be completely covered by an occluding person). However, by deploying our occlusion handling framework on the baseline tracker, MOTA increases to 83%. When there is no occlusion, all persons are tracked only by the main camera of campus 1 video sequence, as shown in the first row of Fig. 4. A scenario in which a person is severely occluded by another person of the same video sequence is shown in the second row of Fig. 4. In the view of the main camera, target 3 is occluded by target 7 but not in the view of the assisting camera 1. Therefore, the tracking task of target 3 is transferred to the assisting camera 1. Similarly, the first row of Fig. 5 shows a scenario in which the main camera alone is sufficient enough to track all targets reliably in train station dataset. However, the tracking system demands assistance from assisting cameras when complex occlusion occurs as shown in a second row of Fig. 5. The detailed quantitative analysis of the tracking task distribution over the main camera and the assisting cameras is discussed in Sec. 5.3.

5.2 *Performance Comparison*

To validate the contribution of the proposed methods to the state-of-the-art, it is crucial to compare the performance as fairly as possible. An ideal way to achieve this is to evaluate all methods on the same dataset and ground truth, using the same evaluation criteria. However, in practice, only few works publicly provide an implementation of their methods. Reimplementing those methods on our own is a very time consuming and usually does not result in exactly the same algorithm. Even if working implementations are available, finding optimal parameters for a particular dataset is not trivial. Poor results for a particular tracker could be due to nonoptimal parameter setting. One way to avoid all these problems is to conduct a performance comparison to reported evaluation results of the related works that use the same dataset, ground truth, and evaluation criteria for their evaluation. In visual people tracking, PETS2009 S2.L1 video sequence has become a widely used benchmark.

Many state-of-the-art people trackers[5,14–21,42] use the PETS2009 S2.L1 video sequence to evaluate their work in terms of aforementioned MOTA and MOTP matrices. In the computation of MOTA and MOTP, all trackers use the same ground truth,[39] except in Refs. 15, 19, 21, and 42. The properties of each tracker, such as single or multiple view tracker, the use of calibrated cameras or not, etc. are listed in Table 2. The performance comparison in Fig. 6 shows that our proposed method outperforms all the state-of-the-art trackers.

Moreover, Liem and Gavrila[24] used the train station dataset to compare the performance of multiple state-of-the-art trackers KSP,[15] KSP-App,[43] RCTA,[44] and their proposed tracker (RCTA$^+$). Since MOTA of each tracker is reported in their work, the performance of our tracker on the train station dataset can be directly compared. In their comparison, KSP-App gives the best performance with MOTA of 92% while MOTA of KSP, RCTA$^+$, and RCTA are 91%, 89%, and 72%, respectively. Although both KSP-App and our method achieve the same MOTA on the train station dataset, KSP-App uses all three camera views for all frames while our method requires only one camera, i.e., only the main camera without any help from the assisting cameras), for almost 70% of the total frames. Another advantage of our method over the KSP-App is the ability to process frame by frame whereas KSP-App needs consecutive batches of frames.

5.3 *Utilization of Cameras and Communication Bandwidth*

In this section, we first analyze the utilization of cameras when tracking on the PETS2009 S2.L1 video sequence. Figure 7 shows how many cameras are actively involved in the position estimation of all targets in each frame of

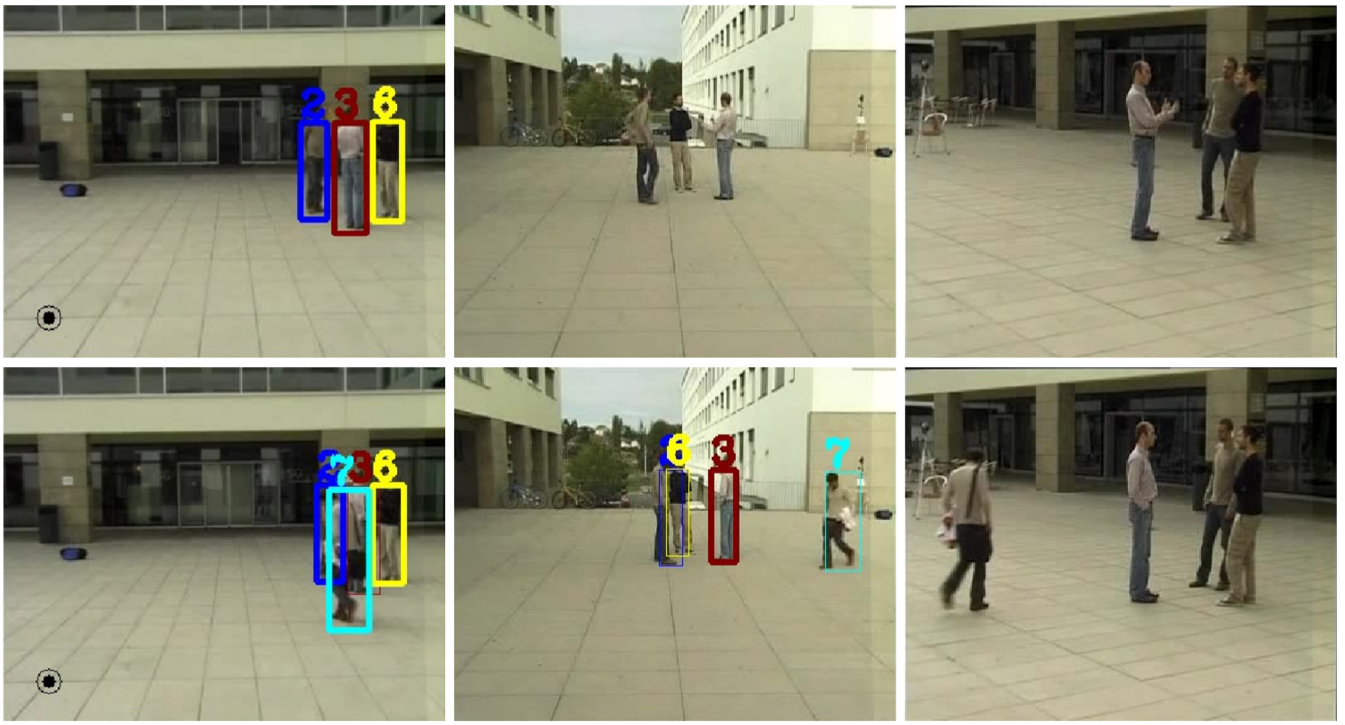


Fig. 4 Camera views of campus 1 video sequence: (a) main camera, (b) and (c) assisting cameras 1 and 2, respectively. The same target in different views is shown with bounding boxes in the same ID number above the bounding boxes. A thin bounding box implies that a target is tracked by another camera, while a thick bounding box indicates that the target is locally tracked.

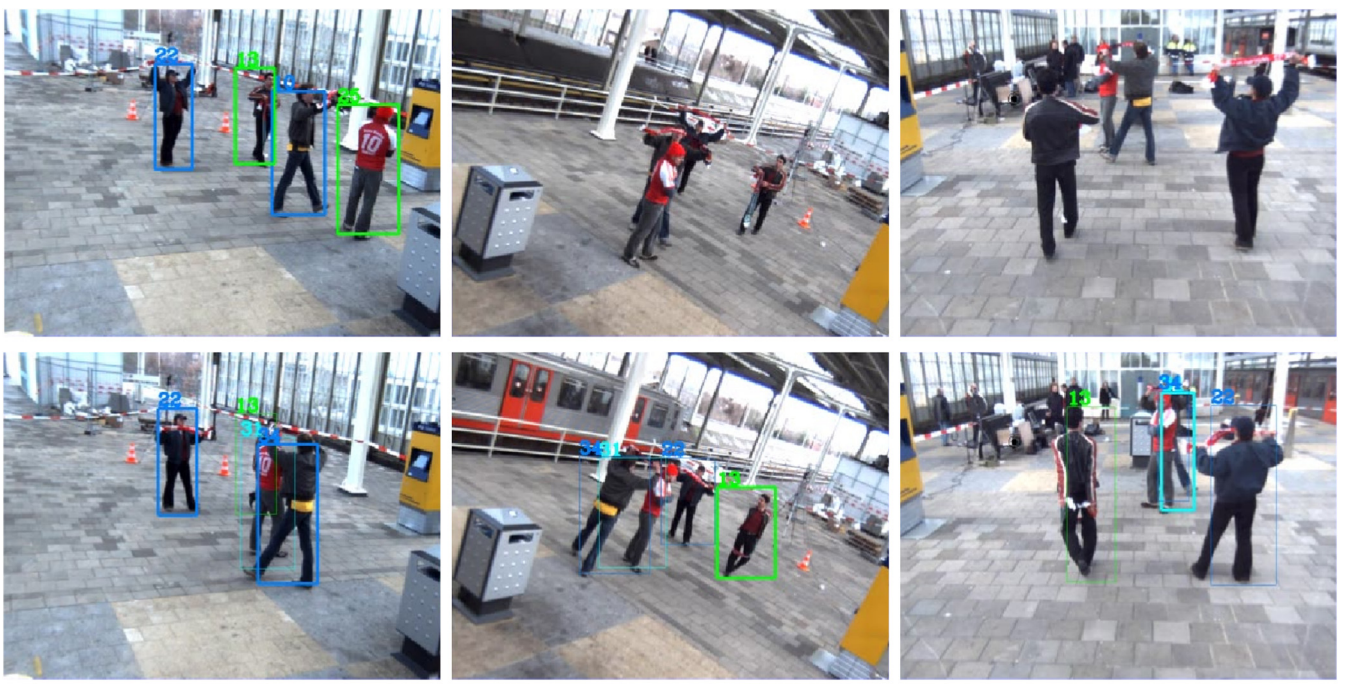


Fig. 5 Camera views of train station video sequence: (a) main camera, (b) and (c) assisting cameras 1 and 2, respectively. The same target in different views is shown with bounding boxes in the same ID number above the bounding boxes. A thin bounding box implies that a target is tracked by another camera, while a thick bounding box indicates that the target is locally tracked.

Table 2 Properties of the state-of-the-art trackers for performance comparison.

Name	Type	Same ground truth	Use calibration
Yang et al.[5]	Single view	No	Yes
Berclaz et al.[15]	Multiple views	No	Yes
Breitenstein et al.[42]	Single view	No	No
Andriyenko and Schindler[14]	Single view	Yes	Yes
Bredereck et al.[16]	Multiple views	Yes	Yes
Milan et al.[17]	Single view	Yes	Yes
Bae and Yoon[18]	Single view	Yes	No
Xiang et al.[19]	Single view	No	No
Bae and Yoon[20]	Single view	Yes	No
Naiel et al.[21]	Single view	No	Yes

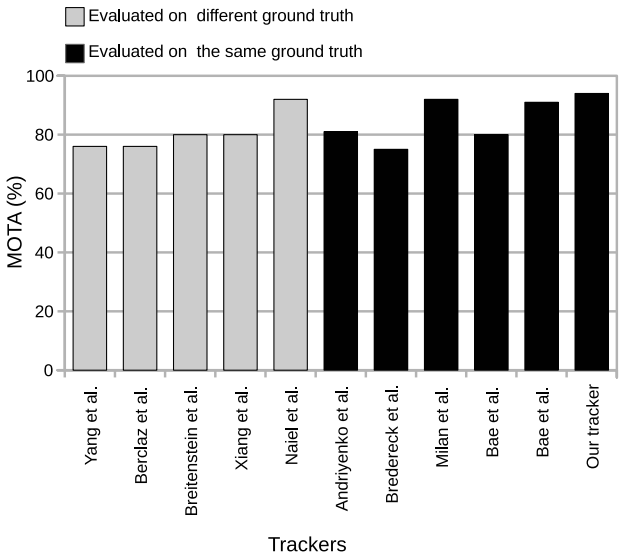


Fig. 6 MOTA comparison to state-of-the-art trackers for PETS2009 S2.L1 video.

PETS2009 S2.L1. There are some periods in which people are moving in the view of the main camera without occluding each other for a while. In these periods, covering 67.7% of the total number of frames, only the main camera actively tracks all the targets. These periods can be clearly seen in Fig. 7 especially between frame 218 to 276 and 370 to 465. Since PETS2009 S2.L1 contains very frequent occlusions, at least two cameras (main camera + one of the assisting cameras) are actively tracking in about 26.2% of all frames. Only 0.1%, i.e., a single frame, of the total number of frames requires all available cameras to be active in order to estimate the position of every person in the scene. Active

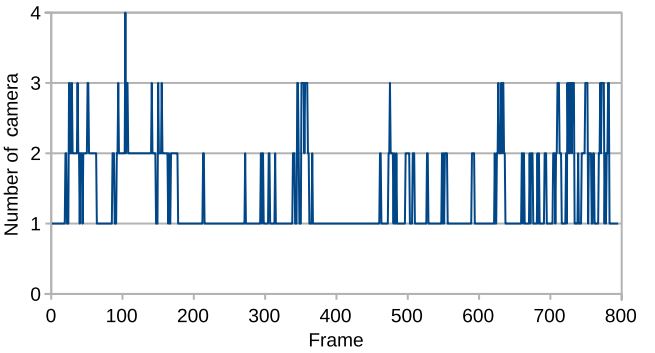


Fig. 7 Number of active camera involved in estimation of persons' positions in each frame of the PETS2009 S2.L1 video.

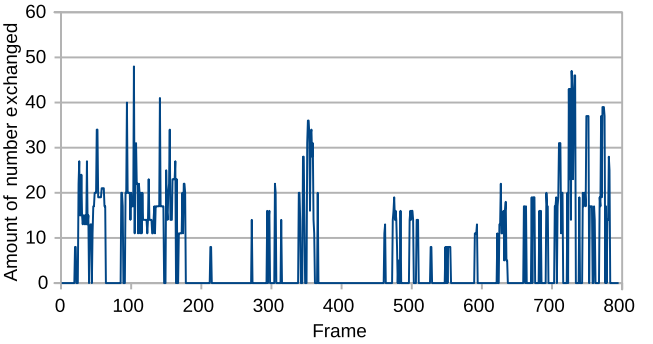


Fig. 8 Amount of numbers (positions and IDs of persons) exchanged between cameras in estimation of persons' positions in each frame of the PETS2009 S2.L1 video.

tracking of three cameras are needed to estimate persons' position in only 6% of the total number of frames.

Moreover, we examine the utilization of intercamera communication bandwidth between the main camera and assisting cameras. As discussed in Sec. 3.4, the main camera sends previously known positions of all persons being tracked together with IDs of persons whose positions are requested to be estimated by a selected assisting camera, i.e., three numbers per person (x, y, ID). The assisting camera responds with locally estimated positions of the requested persons, i.e., two numbers (x, y). Here we do not use number of byte as a measurement for the data exchange since the number of byte required to represent x, y, and ID depends on implementation details. Figure 8 shows the total amount of numbers exchanged (transmit + receive) between the main camera and assisting cameras to handle occlusion in the view of the main camera for each frame. From observing both Figs. 7 and 8, the number of active camera and amount of numbers exchanged is directly proportional. When the main camera does not need assistance from any of assisting cameras, there is no intercamera data exchange. Once there is at least one active assisting camera, intercamera data exchange takes place. The amount of data being exchanged depends on the number of active assisting camera and the number of people being tracked.

5.4 *Contribution of Cameras*

It is also interesting to examine how position estimation tasks are distributed over all cameras as well as how each assisting

camera contributes to the improvement of local tracking in the main camera. Therefore, we run our tracker with different assistance camera combinations and analyze the results. The distribution of position estimation tasks over cameras and corresponding performance matrices for different assisting camera combinations for PETS2009 S2.L1 is shown in Tables 3 and 4. When all assisting cameras are allowed to be used, the main camera processes 91% of position estimation tasks and assisting camera 1, 2, and 3 process 4%, 2%, and 3% of the tasks, respectively, as shown in the last row of Table 3. When only one of three assisting cameras are allowed to be used, estimation tasks processed by the main camera slightly increases. Allowing only assisting camera 1 in occlusion handling does not increase MOTA although 4% of the position estimation tasks are carried out by the assisting camera 1. Since both cameras are observing the scene with the similar view angle, a person being occluded in the view of main camera is often occluded in the view of assisting camera 1 as well. Although Eq. (8) selects assisting camera 1 over main camera, the quality of observation from its view may be insignificantly better than the main camera. Therefore, assistance of assisting camera 1 in occlusion handling is redundant. However, allowing only assisting camera 2 or 3 improves MOTA substantially.

Ideally, using all assisting cameras should increase the tracking accuracy. However, in practice using an all assisting camera does not always improve the tracker's performance. As aforementioned, MOTA may drop due to misalignment of projected bounding boxes while estimation tasks are transferred between cameras with low calibration accuracy. Therefore, in some cases, the tracking accuracy may stay the same or may even drop from adding an assisting camera with low calibration accuracy. Thus, it would be beneficial if the calibration accuracy is known (preferably accuracies for each discretized ground positions to image coordinates) and considered in extension of the proposed energy function described in Eq. (7). In this work, we implement the estimation task transfer between cameras only relying on projective geometry to demonstrate the application of the proposed framework. However, additional computer vision techniques, such as locating the corresponding target in other views using a person detector, a target association across camera views using an appearance model, etc., can be used for better alignment of a projected bounding box over the corresponding target in the view of an assisting camera or the main camera.

Furthermore, we analyze the MOTA improvement by different assisting camera combinations for indoor video sequence as shown in Table 5. Similar to the scenario of main camera and assisting camera 1 of PETS2009 S2.L1, there is no significant MOTA improvement when the main camera gets assistance only from the assisting camera 3 in indoor video sequence. Since calibration accuracy of cameras in indoor video sequence is high (average projection error of one pixel for 3-D points used for calibration), it cannot be due to the misalignment during task transfer resulting from low calibration accuracy. The main reason is that both main camera and assisting camera 3 are observing the scene from a similar viewpoint, and observation quality over both viewpoints is similar. Therefore, getting assistance from the assisting camera 3 does not help much in occlusion handling, compared to the assistance from assisting camera 1 and 2, which are observing the targets from completely different viewpoints.

Table 3 Target's position estimation task distribution for different camera combinations in PETS2009 S2.L1 video.

Main camera	Assisting camera 1	Assisting camera 2	Assisting camera 3
96%	4%	Excluded	Excluded
94%	Excluded	6%	Excluded
93%	Excluded	Excluded	7%
91%	4%	2%	3%

Table 4 MOTA achieves by different camera combinations in PETS2009 S2.L1 video.

Included assisting cameras	MOTA	MOTP
All assisting cameras	94%	76%
Assisting camera 1	92%	77%
Assisting camera 2	94%	77%
Assisting camera 3	93%	77%

5.5 *Influence of Static Occlusion Mask Accuracy*

According to Eq. (7), the optimality of the camera selection is directly proportional to the accuracy of the automatic occlusion mask estimation. An under estimated occlusion mask may result in lower E_{obj} and over estimation may give higher E_{obj} than the actual energy, causing the method to select suboptimal assisting camera. However, using the occlusion map allows the main camera to compute the energies E_{obj} of all assisting cameras locally without needing the actual observed images from the assisting cameras. To observe how our method performs when occlusion mask O is under/overestimated, our method is evaluated with the occlusion mask in Fig. 2 dilated or eroded using circular structuring element with different diameter (2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 pixels). Examples of dilated and eroded occlusion mask are shown in Figs. 9(a) and 9(b). The results show that MOTA varies between 93% and 95% (still better than the tracking results without any assistance) as the

Table 5 MOTA achieves by different camera combinations in indoor video.

Included assisting cameras	MOTA	MOTP
All assisting cameras	98%	71%
Assisting camera 1	96%	72%
Assisting camera 2	97%	71%
Assisting camera 3	95%	71%

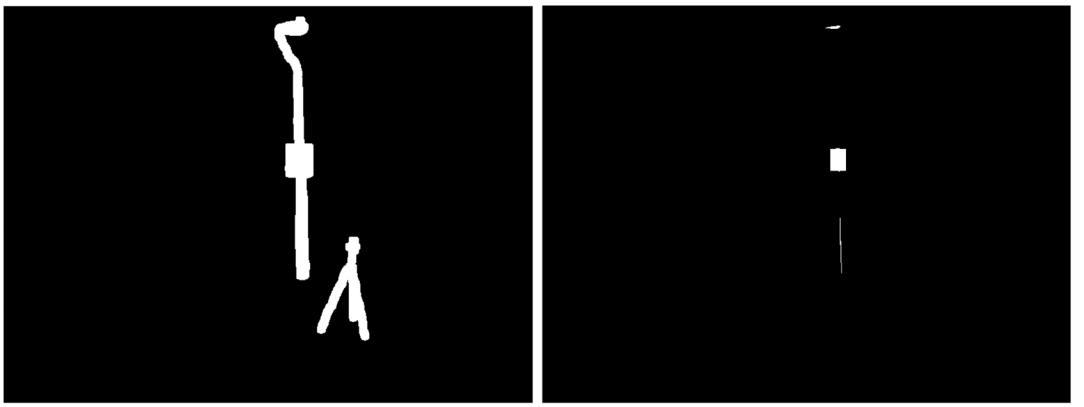


Fig. 9 Static occlusion mask O for the main camera of PETS2009 S2.L1 video (a) dilated and (b) eroded by circular structuring element with diameter of 10 pixels. Original static occlusion mask is shown in Fig. 2(b).

diameter of the circular structuring element changes as shown in Fig. 10. This indicates that our method is still able to select the most appropriate assisting camera for occlusion handling relying on the other two energy terms: E_{per} and E_{FOV}, although E_{obj} does not reflect the actual static occlusion due to inaccurate occlusion mask.

5.6 *Processing Time*

Since our proposed optimal camera selection is purely based on the projective geometry of cameras, the average computation time (measured on all three test video sequences) required for each frame is about 0.2 ms on an Intel Core2Quad at 2.66 GHz. The average processing time of the prototype baseline tracker for a video frame with resolution of 768×576 is about 31 ms on the same computer. This allows our proposed method to track multiple people in real time at about 30 fps. If the processing time of the baseline tracking can be reduced, for example, by precomputing projected bounding boxes, the overall system can track with higher frame rate.

5.7 *Adaptability and Scalability*

In theory, if there is no constraint on computation time and each person is not severely occluded in at least one camera view, our method should be able to track all persons reliably for any number of persons in the scene. Regardless of the number of persons in the scene, if there is no occlusion (best case) in the view of main camera, the main camera alone is enough to take all persons reliably. The theoretical worst case occurs when M persons are either occluding each other or occluded by static objects such that each person must be observed from different viewpoints to have no/less occluded observation for reliable tracking. In this case, number of camera required is the number of person being tracked. Fortunately, such a worst case hardly occurs in practice. Moreover, it is not feasibility/common to install many cameras to observe the same scene of interest in surveillance application. If the number of available cameras is fixed for practical reason, the performance of the system is more likely to drop as the number of person increases, i.e., higher chance that a person may be severely occluded in all camera views. For real-time application, the maximum number of persons being tracked is limited by the speed of a local tracker. In our current implementation, tracking time for each frame increases with the number of persons being tracked. Therefore, the limit on maximum number of persons is reached, when tracking time for all persons exceeds the processing time allowed for real-time application.

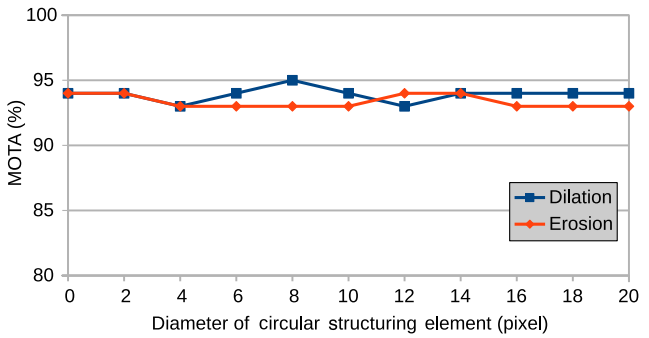


Fig. 10 Influence of occlusion map accuracy over the performance of our proposed method when evaluated on the PETS2009 S2.L1 video sequence.

Furthermore, there is a possibility to adapt our occlusion handling framework for the network of cameras with pan-tilt-zoom functionality or even for the network of moving cameras. However, this adaptation faces many challenges. The first challenge is that cameras must be automatically recalibrated once they pan/tilt/zoom/move. The action of pan/tilt/zoom/move must be constrained so that their overlapping views cover all persons being tracked. The static occlusion mask must be recomputed and local tracker on a camera must be able to cope with moving/changing background due to pan/tilt/zoom/move action. The moving/changing background problem is even more difficult to solve in a foreground/background segmentation-based tracker since most foreground/background segmentation methods are based on the loose assumption that the background is static or only part of the background changes very slowly. Assuming that all these problems are solved, the performance of our method in a network of cameras with pan-tilt-zoom/move functionality should still be the same. If there is an algorithm to automatically adjust the cameras in a configuration such that occlusion in all camera

views are minimum using pan/tilt/zoom/move action, tracking accuracy may even be improved. However, automatic recalibration of cameras, pan/tilt/zoom/move action planning, and static occlusion mask recomputation impose additional computation load to the framework.

6 Conclusion

In this paper, we presented an occlusion handling framework for person tracking by a smart camera in a multicamera network assisted by other cameras only when needed. Moreover, an extensible energy function was proposed for selecting the optimal assisting camera for tracking an occluded target in the main camera's view. The proposed framework was integrated in our previously proposed baseline tracker with a simple projective geometry-based task transfer strategy to demonstrate how well it can handle occlusions. The complete tracker was evaluated on both indoor and outdoor video sequences with frequent person–person and object–person occlusions. The results show that a substantial performance improvement is achieved by deploying our proposed method over the baseline tracker. When compared to a selection of the state-of-the-art trackers, performed on the same datasets, and using the same evaluation criteria, our tracker outperforms. The low computational requirements of our projective geometry-based camera selection and the low bandwidth requirement allow real-time tracking. Despite the good performance on the test videos, there are still many possibilities to further improve the performance as we discussed throughout this paper. These possibilities will be explored in our future work.

Acknowledgments

The work was financially supported by FWO through the project G.0.398.11.N.10 "Multi-camera human behavior monitoring and unusual event detection" and BOF–GOA through the project 01GA2111W "Distributed Smart Camera Systems."

References

1. I. Haritaoglu, D. Harwood, and L. Davis, "W4: Who? When? Where? What? A real time system for detecting and tracking people," in *Proc. Third IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 222–227 (1998).
2. J. F. Henriques, R. Caseiro, and J. Batista, "Globally optimal solution to multi-object tracking with merged measurements," in *IEEE Int. Conf. on Computer Vision (ICCV '11)*, Barcelona, Spain, November 6–13, 2011, pp. 2470–2477 (2011).
3. S. Khan and M. Shah, "Tracking people in presence of occlusion," in *Asian Conf. on Computer Vision*, pp. 1132–1137 (2000).
4. Y. Huang and I. Essa, "Tracking multiple objects through occlusions," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR '05)*, Vol. 2, p. 1182 (2005).
5. J. Yang et al., "Probabilistic multiple people tracking through complex situations," in *11th IEEE Int. Workshop on PETS*, pp. 79–86 (2009).
6. V. Papadourakis and A. Argyros, "Multiple objects tracking in the presence of long-term occlusions," *Comput. Vision Image Understanding* **114**, 835–846 (2010).
7. N. Papadakis and A. Bugeau, "Tracking with occlusions via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 144–157 (2011).
8. S. Gruenwedel et al., "Low-complexity scalable distributed multicamera tracking of humans," *ACM Trans. Sens. Networks* **10**, 24 (2014).
9. N. Bo Bo et al., "Human mobility monitoring in very low resolution visual sensor network," *Sensors* **14**(11), 20800–20824 (2014).
10. N. Bo Bo et al., "Real-time multi-people tracking by greedy likelihood maximization," in *Proc. of the 9th Int. Conf. on Distributed Smart Cameras (ICDSC '15)*, pp. 32–37, ACM, New York (2015).
11. J. Berclaz et al., "EPFL dataset: multi-camera pedestrian videos," 2011, http://cvlab.epfl.ch/data/pom (30 June 2017).
12. J. Ferryman and A. Shahrokni, "PETS2009 dataset," 2009, http://www.cvg.reading.ac.uk/PETS2009/ (30 June 2017).
13. M. C. Liem, M. Hofmann, and D. M. Gavrila, "UvA Person tracking benchmarks: train station dataset," 2009, http://www.lookingatpeople.com/download-uva-icvs13-liem-gavrila/ (30 June 2017).
14. A. Andriyenko and K. Schindler, "Globally optimal multi-target tracking on a hexagonal lattice," in *Proc. of the 11th European Conf. on Computer Vision: Part I (ECCV '10)*, pp. 466–479 (2010).
15. J. Berclaz et al., "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(9), 1806–1819 (2011).
16. M. Bredereck et al., "Data association for multi-object tracking-by-detection in multi-camera networks," in *Sixth Int. Conf. on Distributed Smart Cameras (ICDSC '12)*, pp. 1–6 (2012).
17. A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3682–3689 (2013).
18. S. H. Bae and K. J. Yoon, "Robust online multiobject tracking with data association and track management," *IEEE Trans. Image Process.* **23**, 2820–2833 (2014).
19. J. Xiang et al., "Data association based multi-target tracking using a joint formulation," *Lect. Notes Comput. Sci.* **10114**, 240–255 (2016).
20. S. H. Bae and K. J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* (2017).
21. M. A. Naiel et al., "Online multi-object tracking via robust collaborative model and sample selection," *Comput. Vision Image Understanding* **154**, 94–107 (2017).
22. R. Baillargeon, E. S. Spelke, and S. Wasserman, "Object permanence in five-month-old infants," *Cognition* **20**(3), 191–208 (1985).
23. Y. Yun, I.-H. Gu, and H. Aghajan, "Maximum-likelihood object tracking from multi-view video by combining homography and epipolar constraints," in *Sixth Int. Conf. on Distributed Smart Cameras (ICDSC '12)* (2012).
24. M. C. Liem and D. M. Gavrila, *A Comparative Study on Multi-person Tracking Using Overlapping Cameras*, pp. 203–212, Springer, Berlin, Heidelberg (2013).
25. L. Tessens et al., "Principal view determination for camera selection in distributed smart camera networks," in *Second ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC '08)*, pp. 1–10 (2008).
26. L. Tessens et al., "Camera selection for tracking in distributed smart camera networks," *ACM Trans. Sens. Networks* **10**, 23 (2014).
27. A. P. Dempster, "A generalization of Bayesian inference," *J. R. Stat. Soc. Ser. B (Methodol.)* **30**(2), 205–247 (1968).
28. G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton (1976).
29. R. Munoz-Salinas et al., "Multi-camera people tracking using evidential filters," *Int. J. Approximate Reasoning* **50**(5), 732–749 (2009).
30. G. Brostow and I. Essa, "Motion based decompositing of video," in *Proc. of the Seventh IEEE Int. Conf. on Computer Vision, 1999.*, Vol. 1, pp. 8–13 (1999).
31. P. Favaro et al., "On exploiting occlusions in multiple-view geometry," in *Proc. Ninth IEEE Int. Conf. on Computer Vision, 2003*, Vol. 1, pp. 479–486 (2003).
32. N. Apostoloff and A. Fitzgibbon, "Learning spatiotemporal t-junctions for occlusion detection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR '05)*, Vol. 2, pp. 553–559 (2005).
33. L. Guan et al., "Visual hull construction in the presence of partial occlusion," in *Third Int. Symp. on 3D Data Processing, Visualization, and Transmission*, pp. 413–420 (2006).
34. M. Slembrouck et al., "Self-learning voxel-based multi-camera occlusion maps for 3D reconstruction," in *Int. Conf. on Computer Vision Theory and Applications (VISAPP '14)*, Vol. 2, pp. 502–509 (2014).
35. K. Kawamoto, T. Yonekawa, and K. Okamoto, "Visual vehicle tracking based on an appearance generative model," in *6th Int. Conf. on Soft Computing and Intelligent Systems, and the 13th Int. Symp. on Advanced Intelligence Systems*, pp. 711–714 (2012).
36. J. Arróspide and L. Salgado, "On-road visual vehicle tracking using Markov chain Monte Carlo particle filtering with metropolis sampling," *Int. J. Automot. Technol.* **13**(6), 955–961 (2012).
37. N. Bo Bo et al., "PhD forum: illumination-robust foreground detection for multi-camera occupancy mapping," in *Proc. of the Sixth Int. Conf. on Distributed Smart Cameras (ICDSC '12)* (2012).
38. F. Fleuret et al., "Multicamera people tracking with a probabilistic occupancy map," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**, 267–282 (2008).
39. A. Bernardin and K. Schindler, "Manually annotated ground truth for PETS2009 S2.L1 video sequence," 2011, http://www.milanton.de/data/ (30 June 2017).
40. K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP J. Image Video Process.* **2008**, 1–10 (2008).
41. A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1265–1272 (2011).

42. M. Breitenstein et al., "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 1820–1833 (2011).
43. H. B. Shitrit et al., "Tracking multiple people under global appearance constraints," in *IEEE Int. Conf. on Computer Vision (ICCV '11)*, pp. 137–144 (2011).
44. M. Liem and D. M. Gavrila, *Multi-Person Localization and Track Assignment in Overlapping Camera Views*, pp. 173–183, Springer, Berlin, Heidelberg (2011).

Nyan Bo Bo received his MSc degree in information technology from SIIT, Thammasat University, Thailand, in 2008, after which he worked as a lecturer at Prince of Songkla University, Thailand, until 2012. Currently, he is pursuing his PhD at Image Processing and Interpretation Research Group, Ghent University Belgium. His research interests are visual tracking, data fusion/distributed computing in camera networks, and human activity analysis.

Francis Deboeverie received his master's of science in electronics and ICT engineering technology from the University of Ghent in Belgium in 2007. After his studies, he joined the Department of Telecommunications and Information Processing of Ghent University from where he received his PhD. He worked for the image processing and interpretation group (IPI-TELIN-iMinds) at Ghent University with research focus on image processing problems. His research interests include image interpretation with polynomial feature models for real-time vision systems.

Peter Veelaert received his degree in electronic engineering from the University of Ghent in 1981. After his studies, he started working as an engineer at the Digitized Information Systems Corporation, Belgium, where he developed computer graphics software. In 1986, he joined the Laboratory for Electronics at the University of Ghent from where he received his PhD in 1992. He currently teaches and does research at the Department of Applied Engineering Sciences, Ghent University. His current research interests include real-time systems for low-level vision, image interpretation of road scenes, and geometric uncertainty models.

Wilfried Philips received his diploma degree in electrical engineering in 1989 and his PhD in applied sciences in 1993, both from Ghent University, Belgium. Since November 1997, he is with Ghent University as full-time professor and is heading the research group Image Processing and Interpretation. Some of the recent research activities in the group include image and video restoration and analysis, image and video quality assessment and image analysis and computer vision.