# Occlusion Robust Symbol Level Fusion for Multiple People Tracking

Nyan Bo Bo, Peter Veelaert and Wilfried Philips
*imec-IPI-UGent, Sint-Pietersnieuwstraat 41, Ghent 9000, Belgium*
*{Nyan.BoBo, Peter.Veelaert, Wilfried.Philips}@ugent.be*

Keywords: Multi-camera Tracking, Data Fusion, Occlusion Handling, Uncertainty Assessment, Decentralized Computing.

Abstract: In single view visual target tracking, an occlusion is one of the most challenging problems since target's features are partially/fully covered by other targets as occlusion occurred. Instead of a limited single view, a target can be observed from multiple viewpoints using a network of cameras to mitigate the occlusion problem. However, information coming from different views must be fused by relying less on views with heavy occlusion and relying more on views with no/small occlusion. To address this need, we proposed a new fusion method which fuses the locally estimated positions of a person by the smart cameras observing from different viewpoints while taking into account the occlusion in each view. The genericity and scalability of the proposed fusion method is high since it needs only the position estimates from the smart cameras. Uncertainty for each local estimate is locally computed in a fusion center from the simulated occlusion assessment based on the camera's projective geometry. These uncertainties together with the local estimates are used to model the probabilistic distributions required for the Bayesian fusion of the local estimates. The performance evaluation on three challenging video sequences shows that our method achieves higher accuracy than the local estimates as well as the tracking results using a classical triangulation method. Our method outperforms two state-of-the-art trackers on a publicly available multi-camera video sequence.

## 1 INTRODUCTION

Despite many years of research, visual target tracking still remains a very challenging problem in computer vision. Among various targets, a human body is one of the most difficult targets to track due to its non-rigid nature, *i.e.*, the movement of body parts, such as arms and legs, alters its appearance. The appearance of a person can also change with the orientation of the body with respect to a camera view or with the variation in scene illumination. When the scene contains more than one moving person, the tracking task becomes even more difficult since a person may sometimes be occluded by another person(s) in a camera view.

Many monocular camera tracking methods (Khan and Shah, 2000), (Yang et al., 2009), (Henriques et al., 2011) have been proposed to track occluded people reliably from a single viewpoint by using various occlusion prediction and handling techniques. However, these methods rely on a robust segmentation of the non-occluded regions of a target when partial occlusion occurs. If a target is fully occluded, no observation is available and single view occlusion handling methods just interpolate an unavailable observation by using motion models and constraints on temporal continuity. The limitation to a single viewpoint in monocular camera tracking systems can be avoided by deploying a network of cameras with overlapping views, observing the same target from different viewpoints to handle the occlusion problem. The emergence of low-cost cameras and cheaper computing power makes the multi-camera tracking approach more feasible for practical applications.

However, when a target is observed/tracked from different viewpoints with multiple cameras, there is a need for an algorithm which systematically integrates those observations/tracking results to improve the overall tracking accuracy and precision. Many algorithms have been proposed to fuse information from multiple cameras at either the feature level or the symbol level. Feature level data fusion requires the transmission of extracted image features from the cameras to a fusion node which sometimes demands a large communication bandwidth. Moreover, the feature level fusion algorithms are usually coupled to a particular feature or to a set of features. Therefore, it is usually required to modify a fusion method or to replace it with a different fusion method, if another

Bo Bo N., Veelaert P. and Philips W.
Occlusion Robust Symbol Level Fusion for Multiple People Tracking.
DOI: 10.5220/0006127602160226
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 216-226
ISBN: 978-989-758-227-1

set of features is used. This limits the scalability and adaptability of the camera network.

In the symbol level data fusion, only symbols, *i.e.*, estimated positions possibly accompanied by corresponding uncertainties, are sent to a fusion node to generate fused estimates. Since positions are usually represented with few numbers, the inter-camera communication bandwidth requirement is relatively low. The lower communication bandwidth requirement results in a lower latency in the distributed multi-camera tracking. When there is a power consumption constraint on the wireless smart cameras (for example, battery-powered smart cameras which communicate with a fusion center over the wireless links), it is desirable to minimize the amount of data exchanges the since power consumption increases with the amount of data being sent and received. Moreover, the tracking algorithm deployed on the smart cameras is decoupled from the fusion algorithm in the symbol level fusion. It is possible to have different tracking algorithms deployed on different cameras within the network as long as they all estimate the same state of a target. Therefore, symbol level fusion provides higher genericity and scalability in multi-camera tracking.

In this paper, we propose a new symbol level data fusion method which takes into account the degree of occlusion in each camera view. Each smart camera locally estimates the position of all targets and sends them to a fusion center. Since only positions are sent to a fusion center, the latency of the whole tracking system is usually low. Using the projective geometry, a fusion center locally simulates the occlusions in each camera view to compute the uncertainty of each estimate by the corresponding camera. Fused estimates are then made by using a Bayesian approach based on the local estimates from all cameras and their corresponding uncertainties. Since our method requires only local estimates from smart cameras, it is feasible to deploy any black box single view tracker on any smart camera in the network.

The main contribution of this paper is the formulation of a fusion algorithm which fuses the local estimates of the same target from different camera views based on the corresponding uncertainties which are estimated from simulated occlusions. Our method allows high genericity and scalability while maintaining low latency. Another contribution is the performance evaluation of the proposed fusion method on three challenging multi-camera video sequences. The evaluation shows that our method drastically improves the accuracy in the video sequences containing frequent and severe person–person occlusions. We also demonstrate that a complete decentralized multi-camera tracking system which is the combination of our fusion method and previously implemented single view tracker (Bo Bo et al., 2015) outperforms state-of-the-art trackers in terms of multiple object tracking accuracy.

The rest of this paper is organized as follows. In Section 2, we gives a brief description of related work in the literature. Section 3 thoroughly discusses the details of our proposed fusion method. The performance evaluation of our method and the interpretation of the results are presented in 4. Finally, this paper is concluded in Section 5.

## 2 RELATED WORK

According to a categorization by (Luo and Kay, 1990), data or information from multiple sensors can be fused at signal, pixel, feature and symbol levels of representation. The majority of multi-view trackers belongs to either the feature level or the symbol level data fusion scheme. In multi-view tracking, features or measurements to be fused can be foreground detected images, histograms, occupancy maps, object detector responses and so on. A central tracker uses the output of the feature level fusion to estimate the positions of the targets. For symbol level fusion in multi-camera tracking, symbols to be fused are the local position estimates of the smart cameras, which are sometimes accompanied by the corresponding uncertainties.

Some trackers (Mittal and Davis, 2003), (Fleuret et al., 2008), (Grünwedel et al., 2012) build probabilistic occupancy maps (POM) from foreground detected images of different cameras views using Bayesian or Dempster-Shafer theory. The trajectories of the targets are estimated from the resulting POM. The trackers of (Du and Piater, 2006), (Du and Piater, 2007) and (Mori et al., 2008) deploy particle filters in which the weight of each particle is calculated from measurements from multiple views using the Bayesian fusion approach. The approach of (Munoz-Salinas et al., 2009) is similar but image measurements are fused using Dempster-Shafer theory to computed the particles' weights. In the work of (Andriyenko and Schindler, 2010), person detector response scores from different views are fused into an observation model, which is one of the terms in their proposed energy function. This energy function is minimized to find the best trajectories. Feature level fusion is also used in our previous works (Bo Bo et al., 2014a) and (Bo Bo et al., 2016), in which likelihoods of people positions computed from foreground images of different cameras are fused in a fusion center based on Bayesian theory.

The aforementioned feature level fusion based trackers require the transmission of image features which are computed on each camera such as foreground detected images, histograms, occupancy maps, etc. to a central tracker or a fusion center. Therefore, the communication bandwidth requirements can be high and some of these trackers are not feasible for implementing in a distributed computing scheme. Moreover, feature level data fusion methods are usually coupled to specific feature/set of features as well as the tracking method. Adaptation of the fusion method to a new feature/set of features is usually not straightforward. Therefore, data fusion at the feature level sometimes results in lower genericity and scalability. However, symbol level fusion allows higher scalability since cameras send locally estimated positions, which is much more compact than image features, to the fusion center. In people tracking applications, symbols we consider are the locally estimated positions of persons in either image coordinates or world coordinates. Since data fusion is performed at the symbol level, different single view tracking algorithms can be deployed on cameras in the network.

Most widely used symbol level fusion methods in visual people tracking include triangulation, Bayesian estimation and so on. In the multi-view tracking method proposed by (Bredereck et al., 2012), each smart camera in the network locally estimates the position of a target in image coordinates. The fused estimate of a target position in world coordinates is the centroid of the pairwise triangulations of local estimates from all cameras. Similarly, locally estimated positions on the ground plane from each observing smart camera are fused by triangulation in our previous work of (Bo Bo et al., 2014b). A distributed tracking method proposed by (Gruenwedel et al., 2014) fuses positions estimated by each camera using a Bayesian estimating methods. However their fusion method does not take into account of occlusion.

Recently (Niño-Castañeda et al., 2016) proposed a Bayesian method to fuse trajectories produced by different tracking methods into more accurate trajectories. This method is used for a semi-automatic annotation of large visual target tracking datasets. In that work, probability distributions required for Bayesian fusion are learned from the data. An important requirement is that the training data must contain examples of all scenarios (occlusion, illumination variation, etc.), which usually cause performance degradation in each tracker. Since that method is intended for an automatic annotating of large datasets with some manual human interventions, it is not feasible to use for fully automatic people tracking.

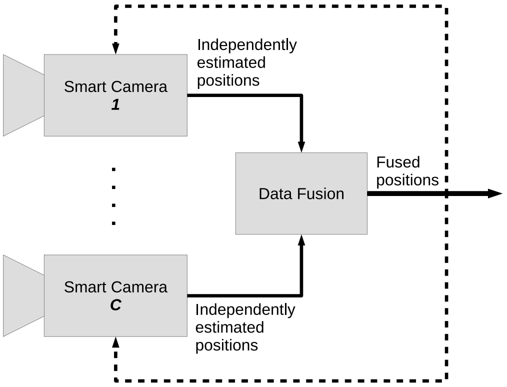

Smart Camera 1
Independently estimated positions
Data Fusion
Fused positions
Smart Camera C
Independently estimated positions

Figure 1: Building blocks of distributed multi-camera tracking system.

# 3 OCCLUSION ROBUST FUSION

In this paper, we consider a decentralized multi-camera tracking system as depicted in Fig. 1. Each smart camera $c$ independently estimates the ground plane position of a person $m$ in its view denoted as $\mathbf{s}_{m,c} = (x_{m,c}, y_{m,c})^T$. If $M$ persons are in the scene, a smart camera $c$ estimates the positions $\mathbf{s}_{1,c}, \dots \mathbf{s}_{M,c}$ of all $M$ persons. As depicted in Fig. 1, $C$ cameras are observing and tracking $M$ persons at the same time. Upon the completion of local estimation, each smart camera $c$ sends its local estimates to a fusion center. Therefore, $C$ different estimates for $M$ persons are received by the fusion center as $\mathbf{s}_{1,1}, \dots \mathbf{s}_{M,1}, \mathbf{s}_{1,2}, \dots \mathbf{s}_{M,C}$. The task of the fusion center is to integrate these local estimates systematically into more accurate and reliable global estimates $\mathbf{s}_1, \dots \mathbf{s}_M$ by taking into account of the occlusion in each camera view. The detailed description of the proposed fusion algorithm will be presented in the following subsections. The more accurate fused positions can be fed back to all smart cameras, as shown with dotted arrows in Fig. 1. Upon receiving of more accurate fused estimates, tracker on each smart camera can correct the current state of a target if its locally estimated position is far from the fused estimate.

## 3.1 Bayesian Fusion

As mentioned before, the main task of the fusion center is to estimate the fused positions $\mathbf{s}_1, \dots \mathbf{s}_M$ from local estimates $\mathbf{s}_{1,1}, \dots \mathbf{s}_{M,1}, \mathbf{s}_{1,2}, \dots \mathbf{s}_{M,C}$ sent by all $C$ smart cameras. In probabilistic terms, this estimation problem can be formulated as finding $\mathbf{s}_1, \dots \mathbf{s}_M$ that maximizes the posterior distribution

$$P(\mathbf{s}_1,\dots\mathbf{s}_M|\mathbf{s}_{1,1},\dots\mathbf{s}_{M,1},\mathbf{s}_{1,2},\dots\mathbf{s}_{M,C}). \quad (1)$$

However, searching for $\mathbf{s}_1,\dots\mathbf{s}_M$ that maximizes the posterior distribution in expression (1) is computational complex. This complex simultaneous maximization of all fused positions can be simplified as a maximization of individual fused positions if we assume that the fused positions $\mathbf{s}_1,\dots\mathbf{s}_M$ are conditionally independent. This assumption implies that the fused position of a person is independent of the fused positions of other persons. The complex maximization problem is now reduced to

$$P(\mathbf{s}_m|\mathbf{s}_{1,1},\dots\mathbf{s}_{M,1},\mathbf{s}_{1,2},\dots\mathbf{s}_{M,C}), \quad (2)$$

where $m \in \{1,\dots M\}$. Due to the assumption of independence between fused positions, it is possible that the same ground plane position can be occupied by multiple persons. However, this rarely occurs in practice.

The posterior distribution in expression (2) must take into account the possible occlusions over a person $m$ in the view of each smart camera. Given the local position estimates $\mathbf{s}_{1,c},\dots\mathbf{s}_{M,c}$ of a smart camera $c$, the possible occlusions over a person $m$ can be quantified as $w_{m,c}$. The detailed description on the computation of $w_{m,c}$ will be discussed in the following Subsection 3.2. Since $w_{m,c}$ summarizes the possible occlusions of other people over a person $m$, the condition of the posterior distribution in expression (2) can be rewritten as

$$P(\mathbf{s}_m|\mathbf{s}_{m,1},\dots\mathbf{s}_{m,C},w_{m,1},\dots w_{m,C}). \quad (3)$$

According to Bayes rule, maximization of the posterior distribution in expression (3) can be done by maximizing the product of a likelihood and a prior distribution:

$$P(\mathbf{s}_{m,1},\dots\mathbf{s}_{m,C},w_{m,1},\dots w_{m,C}|\mathbf{s}_m)P(\mathbf{s}_m). \quad (4)$$

Since we do not have a prior knowledge of which location is more likely to be the true position of a person $m$, the prior distribution $P(\mathbf{s}_m)$ is set to be a uniform distribution, *i.e*, all locations are equally likely to be the true position of a person $m$. Finally, a intractable maximum a posteriori estimation problem becomes a maximum likelihood estimation problem as:

$$\hat{\mathbf{s}}_m = \arg\max_{\mathbf{s}_m} P(\mathbf{s}_{m,1},\dots\mathbf{s}_{m,C},w_{m,1},\dots w_{m,C}|\mathbf{s}_m). \quad (5)$$

Here we make an additional assumption of conditional independence between the local estimations by the smart cameras. This assumption implies that conditioned on the fused position of a person, the local estimation of the person's position in a particular camera is independent of other cameras. It is a practically valid assumption since a single view tracker in each smart camera independently estimates the position of a person based only on image measurements from its own view. Therefore, the likelihood of the local estimates of all smart cameras being $\mathbf{s}_{m,1},\dots\mathbf{s}_{m,C},w_{m,1},\dots w_{m,C}$ given the fused position $\mathbf{s}_m$ is the product of the likelihood of the local estimates of each smart camera given the the fused position $\mathbf{s}_m$. Hence, the likelihood can be written as:

$$\begin{aligned}&P(\mathbf{s}_{m,1},\dots\mathbf{s}_{m,C},w_{m,1},\dots w_{m,C}|\mathbf{s}_m)\\ &\quad=\prod_{c=1}^{C}P(\mathbf{s}_{m,c},w_{m,c}|\mathbf{s}_m).\end{aligned} \quad (6)$$

What we need now is to compute the likelihood $P(\mathbf{s}_{m,c},w_{m,c}|\mathbf{s}_m)$ for each camera.

## 3.2 Likelihood from Occlusion

As discussed before, a fusion center receives locally estimated positions from smart cameras. Depending on the tracking algorithm deployed on the smart cameras, it is possible to send the uncertainty of each estimate to the fusion center. However in this work, we assume that smart cameras do not send any kind of uncertainty for each estimate. This assumption increases the genericity of our fusion methods since local estimates of different single view tracking methods can be fused without caring about how each method computes uncertainties of the estimates. Therefore, the likelihood $P(\mathbf{s}_{m,c},w_{m,c}|\mathbf{s}_m)$ for each camera must be computed from local estimates of cameras and other pre-acquired knowledge such as the geometric relationships between the cameras. Since a fusion center does not have any prior knowledge of the uncertainty of the local tracker on each camera, it assumes that the uncertainty of a local estimate of a target correlates to the severity of occlusion over the target.

Given that a fusion center knows the calibration matrices of a smart camera $c$, person–person occlusion in the view of a camera $c$ can be simulated from the camera $c$'s locally estimated positions of people. First, a 3D model of a person (cuboid, cylinder, etc.) is placed at position of each person being tracked, that is, at $\mathbf{s}_{1,c},\dots\mathbf{s}_{M,c}$. These 3D models are projected on the image plane of the camera $c$ as $\omega(\mathbf{s}_{1,c}),\dots\omega(\mathbf{s}_{M,c})$. For simplicity we denote $\Omega_c = \{\omega(\mathbf{s}_{1,c}),\dots\omega(\mathbf{s}_{M,c})\}$ as a set of projections of 3D models of all $M$ persons. Person $m$ at position $\mathbf{s}_{m,c}$ is possibly occluded by one or more other persons if the projection $\omega(\mathbf{s}_{m,c})$ of person $m$ overlaps with the union of the projections of all other persons. The severity of a possible occlusion can be quantified as

$$\hat{\omega}_c(\mathbf{s}_{m,c}) = \omega_c(\mathbf{s}_{m,c}) \cap \bigcup_{\omega \in (\Omega_c \setminus \omega_c(\mathbf{s}_{m,c}))} \omega. \tag{7}$$

The area of $\hat{\omega}_c(\mathbf{s}_{m,c})$ increases as more body parts of a person $m$ are covered by other persons in the view of a camera $c$. However, the maximum possible size of $\hat{\omega}_c(\mathbf{s}_{m,c})$ is $\omega_c(\mathbf{s}_{m,c})$. Therefore occlusion severity can be normalized as

$$w_{m,c} = \frac{|\hat{\omega}_c(\mathbf{s}_{m,c})|}{|\omega_c(\mathbf{s}_{m,c})|}, \tag{8}$$

where the operator $|.|$ denotes the area of a geometric shape. When a person $m$ is completely occluded by one or more other people, $w_{m,c}$ is at its highest, *i.e.*, $w_{m,c} = 1$. Likewise, it is at its lowest, *i.e.*, $w_{m,c} = 0$ if no one is occluding a person $m$.

Occlusion usually degrades the performance of all types of visual trackers. Some trackers are designed to be more robust against occlusion. However, regardless of tracker robustness to occlusion, if a person $m$ is partially occluded in the view of a camera $c$ but not occluded in the view of another camera $c'$, it is more likely that the positional error on the estimate $\mathbf{s}_{m,c}$ made by the camera $c$ is larger than the positional error of $\mathbf{s}_{m,c'}$. Therefore the distribution $P(\mathbf{s}_{m,c}, w_{m,c}|\mathbf{s}_m)$ must model the uncertainty of the local position estimates based on the severity of the occlusion. If a person is not occluded the standard deviation $\sigma$ of $P(\mathbf{s}_{m,c}, w_{m,c}|\mathbf{s}_m)$ should be small and the $\sigma$ should increase with the severity of the occlusion.

As in the work of (Niño-Castañeda et al., 2016), the distribution $P(\mathbf{s}_{m,c}, w_{m,c}|\mathbf{s}_m)$ can be learned from the training data. Another approach is to select the probability density function and the parameters that best fit the training data. However, this method requires a large training data and it is probably necessary to retrain with a new training data if orientation and/or location of a camera is changed. Moreover, the shape of the distribution differs depending on many factors such as the position and orientation of a camera with respect to the scene, the deployed tracking algorithm, the calibration accuracy and so on. Therefore, to make our solution generic, we assume that the uncertainty of the local estimate due to occlusion can be modeled as a normal distribution: mean $\mu$ is at a locally estimated position and standard deviation $\sigma$ is directly proportional to $w_{m,c}$. Therefore, the likelihood of the locally estimated position and its uncertainty being $\mathbf{s}_{m,c}$ and $w_{m,c}$, given the fused position $\mathbf{s}_m$ is computed as

$$P(\mathbf{s}_{m,c}, w_{m,c}|\mathbf{s}_m) = \mathcal{N}(\mathbf{s}_{c,m}, \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}) \tag{9}$$

where

$$\sigma = e^{-(1-w_{m,c})}. \tag{10}$$

## 3.3 Implementation

To demonstrate how our fusion method handles occlusion problem well, we implement a complete decentralized multi-camera tracking system as depicted in Fig. 1 in which the proposed fusion method is deployed on a data fusion block. An existing single recursive tracker which was proposed in our previous work (Bo Bo et al., 2015) is used for the local tracking on smart camera blocks. The tracker on each camera tracks multiple persons 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 (Bo Bo et al., 2012) are used as observations $F_c$ in the likelihood computation. The whole system is implemented in C++ as a single executable in which local tracking of each camera and fusion is computed sequentially.

For the local tracking on each smart camera, we simply use the likelihood function $P(F_c|\mathbf{s}_{1,c}, \ldots \mathbf{s}_{M,c})$ together with the default parameters reported in (Bo Bo et al., 2015). The search space for maximizing $P(F_c|\mathbf{s}_{1,c}, \ldots \mathbf{s}_{M,c})$ is defined based on the known positions of all persons at the previous frame and the physical limitation 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. The fused estimates are fed back to all smart cameras so that local trackers can correct their local estimates, which are used as prior for the next cycle of position estimation. This prevents the local trackers from potential drifting due to error accumulation during the recursive state estimation.

# 4 PERFORMANCE EVALUATION

## 4.1 Videos for Evaluation

We evaluate the performance of the proposed fusion method on both indoor and outdoor multi-camera video sequences. For *Indoor* video, we capture a video sequence in a room of 8.8×9.2 m$^2$ using four calibrated cameras with overlapping views. The video is captured with the resolution of 780×580 pixels at 20 fps and it has a total duration of approximately six minutes. Up to four people are walking in the scene and they often occlude each other.

For the outdoor scenario, we use the publicly available *Campus 1* video sequence,[1] which is cap-

[1] http://cvlab.epfl.ch/data/pom

tured by Fleuret *et al.* for the performance evaluation of their tracker based on occupancy mapping (Fleuret et al., 2008). Three cameras with overlapping views are used to capture the video sequence with a resolution of 360×288 pixels at 25 fps while up to four people are walking in front of the cameras. Due to the low camera pitch angle, this sequence is very challenging with respect to accurate position estimation with local tracking on smart cameras.

We also evaluate our method on the *PETS2009 S2.L1* video sequence,[2] which is widely used as a benchmark in the multi-person tracking literature. Up to 8 persons are present in the scene and the video lasts for about 1.5 minutes. Although it is a relatively short video, it contains various kinds of multi-people tracking challenges such as low frame rate (7 fps), frequent person–person occlusions and close proximity between persons. The *PETS2009 S2.L1* is captured with eight cameras but calibration accuracy of four cameras are quite low. Therefore, only four cameras with the higher calibration accuracy is used in our performance evaluation. accuracy of the unused cameras. Ground plane positions for each person have been manually annotated every 20 frames for both *Indoor* and *Campus 1* video sequences. However, annotated ground truth for *PETS2009 S2.L1* video sequence is publicly available[3].

## 4.2 Performance Metrics

For the performance evaluation measurement, we choose the CLEAR-MOT metrics (Bernardin and Stiefelhagen, 2008) since they are the most widely used systematic evaluation metrics in literature. Many state of the art trackers (Andriyenko and Schindler, 2011), (Yang et al., 2009), (Berclaz et al., 2011), (Bredereck et al., 2012) use these metrics to measure the performance of their methods. These metrics take into account all types of errors produced by multiple object tracking systems and summarize them into the Multiple Object Tracking Precision (MOTP) and the Multiple Object Tracking Accuracy (MOTA). MOTP measures the positional error between the ground truth and the tracker's estimate pairs over all frames. It is computed as

$$MOTP = (T_d - \frac{\sum_{m,t} d_{m,t}}{\sum_t c_t}) \cdot \frac{100}{T_d}, \tag{11}$$

where $d_{m,t}$ is the Euclidean distance between the tracker's estimate $\mathbf{s}_{m,t}$ and the corresponding ground truth. The total number of matches $c_t$ is the number of ground truth and tracker's estimate pairs, for which the Euclidean distance is less than the threshold $T_d$.

If the Euclidean distance between a tracker's estimate and its nearest ground truth exceeds the threshold $T_d$, it is counted as the number of object miss denoted as $miss_t$. Moreover, if a ground truth point has no matching tracker estimate, it is counted as the number of false positives denoted as $fp_t$. When multiple objects are getting close to each other, the tracker sometimes confuses the identity of the objects. This misidentification of two objects made by a tracker is counted as the number of identity mismatches denoted as $mme_t$. These error types are summarized into MOTA as

$$MOTA = \left(1 - \frac{\sum_t (miss_t + fp_t + mme_t)}{\sum_t g_t}\right) \cdot 100, \tag{12}$$

where $g_t$ is the total number of available ground truths at time $t$. For both MOTP and MOTA, a higher value indicates a better performance. Ideally trackers should have high MOTP and MOTA but sometimes one of the two metrics may be more important depending on higher level applications which use the trajectories of the tracker.

## 4.3 Quantitative Evaluation

We run our complete multi-camera tracker implementation on the aforementioned three video sequences. To observe how much our fusion method improves the performance, we also use the classical triangulation method to fuse the local estimates from the cameras. Both results are compared against the ground truths in terms of MOTA and MOTP, and listed in Table 1. The table shows that both triangulation and our method perform equally on the *Indoor* video sequence. The *Indoor* video sequence is captured with cameras installed at the height of approximately three meters with high camera pitch angle. Therefore, full person–person occlusion rarely occurs in this video sequence. If there is only small partial occlusion, the single view tracker (Bo Bo et al., 2015) can handle it well and the local estimates are quite accurate. Since the accuracy of all local estimates is high, the triangulation of those local estimates are also accurate. This makes the *Indoor* video sequence the least challenging of all three test video sequences. Therefore, both methods achieve the highest MOTA of 98% and MOTP of 82% on *Indoor* video sequence.

On the *Campus* video sequence, our method outperforms other methods with a MOTA of 80% while triangulation method only achieves a MOTA of 72%. However, the MOTP of the triangulation method is a bit higher than our method. Since MOTP is sensitive

[2] http://www.cvg.reading.ac.uk/PETS2009/

[3] http://www.milanton.de/data/

Table 1: Comparison of MOTA and MOTP for triangulation (TRI) and Bayesian (BAY) fusion.

| | MOTA | | MOTP | |
|---|---|---|---|---|
| Video | TRI | BAY | TRI | BAY |
| *Indoor* | 98% | 98% | 82% | 82% |
| *Campus* | 72% | 80% | 79% | 77% |
| *PETS2009* | 79% | 94% | 66% | 72% |

to annotation errors in the ground truth, a small difference in MOTP is not significant. The performance difference is the largest in terms of both MOTA and MOTP on *PETS2009* video sequence. The MOTA of our method is 15% higher and MOTP is 6% higher than the triangulation method. Unlike in the *Indoor* video sequence, cameras are installed at a height of approximately two meters with a relatively low camera pitch angle in the *Campus* and the *PETS2009* video sequences. Therefore, severe/full occlusions often occur in these video sequences. Since single view trackers usually can not handle full occlusion well, local estimates in a view with severe/full occlusion are sometimes far from the actual positions.

When the majority of the cameras provide local estimates with low accuracy, the position obtained by triangulation usually has a large positional error although the local estimates from the minority of the cameras with no or small occlusion are very accurate. However, our fusion method simulates occlusion in each camera view to assess the uncertainty of each local estimates. This uncertainty defines the importance of the corresponding local estimate in the computation of a fused position. Based on this uncertainty, our method assigns a higher weight to the local estimates from the views with no/small occlusion and a lower weight to the local estimates from the views with severe/full occlusion. By suppressing the influence of the local estimates with potentially low accuracy and relying more on the local estimates which are more likely to have a higher accuracy, our method can handle occlusion efficiently and improves the overall tracking accuracy.

To validate the contribution of the proposed method, we compare the performance of our tracker with two state-of-the-art multi-camera trackers proposed in the work of (Berclaz et al., 2011) and (Bredereck et al., 2012). As a recap, the tracker of (Berclaz et al., 2011) is based on the feature level fusion scheme whereas the tracker of (Bredereck et al., 2012) is implemented in symbol level fusion scheme. In their work, the *PETS2009 S2.L1* video sequence is used to evaluate the performance of their tracker and tracking performance is also reported in term of MOTA and MOTP. Since MOTA is more robust against bias and mistakes in manual ground truth annotation, only the MOTA of the trackers is compared. The reported MOTA of (Berclaz et al., 2011) and (Bredereck et al., 2012) is 76% and 80% respectively. Our method achieves a significantly better MOTA of 94%.

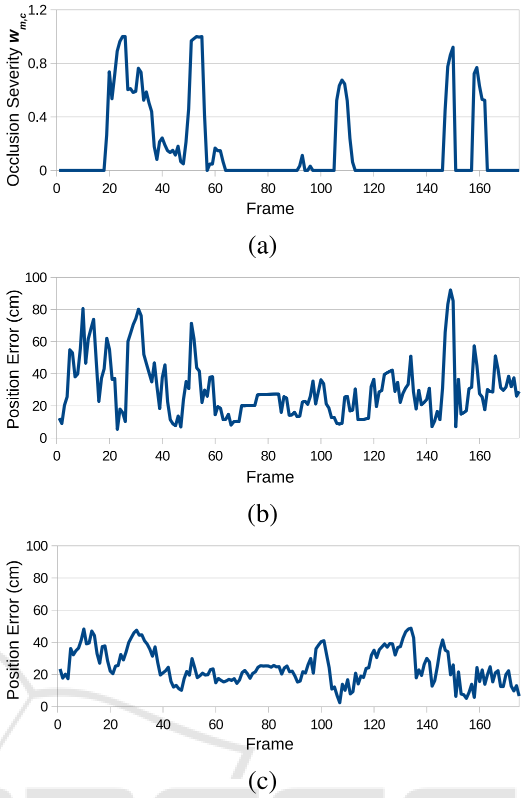
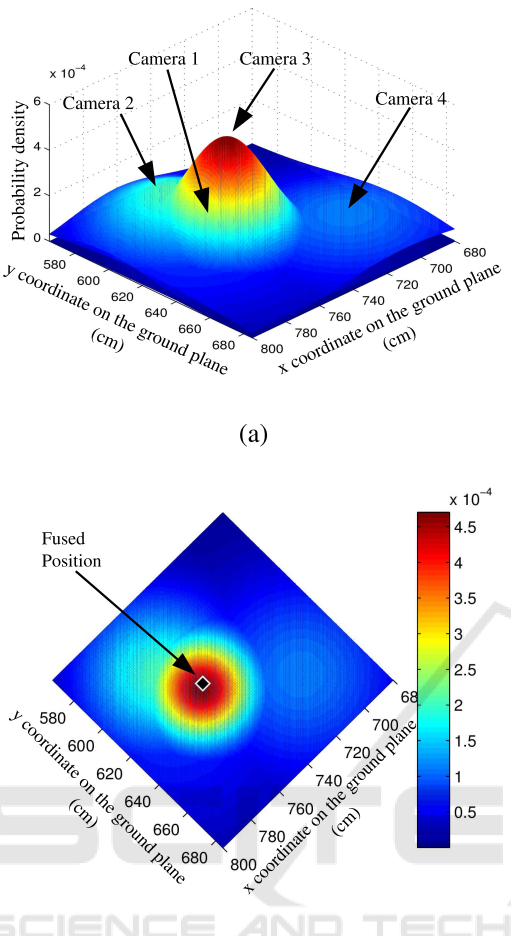


Figure 2: Time series plots of: (a) occlusion severity $w_{2,4}$ of *Person 2* in the view of the camera 4, (b) positional error of *Person 2*'s position locally estimated by the camera 4 in centimeter and (c) positional error of person 2's fused estimate in centimeter.

## 4.4 Analysis and Discussion

We further analyze the results of the local estimations and the proposed fusion method by comparing it to the ground truth. Figure 2 (b) shows the positional error made by a local tracker on the camera 4 when estimating the position of a person with ID 2 (denoted as *Person 2*) between frame 0 to 180 of the *PETS2009* video sequence. If the positional error is compared to the occlusion severity $w_{m,c}$, we found that high error peaks in Fig. 2 (b) usually correspond to occlusion ratio $w_{m,c}$ peaks in Fig. 2 (a). However in some cases, the local trackers are still able to make local estimates despite the presence of heavy occlusions. An example of this scenario can be seen between frame 100 and 120 of plots in Fig. 2 (a) and (b). Moreover, the positional error can be increased by other factors such



(a)

(b)

Figure 3: Distributions of $P(\mathbf{s}_{2,1}, w_{2,1}|\mathbf{s}_2)$, $P(\mathbf{s}_{2,2}, w_{2,2}|\mathbf{s}_2)$, $P(\mathbf{s}_{2,3}, w_{2,3}|\mathbf{s}_2)$ and $P(\mathbf{s}_{2,4}, w_{2,4}|\mathbf{s}_2)$: (a) tilted profile view, and (b) top view.

as occlusion by objects (table, lamp post, car, etc.) in the scene, variation in lighting, or when a target is partially outside of the field of view.

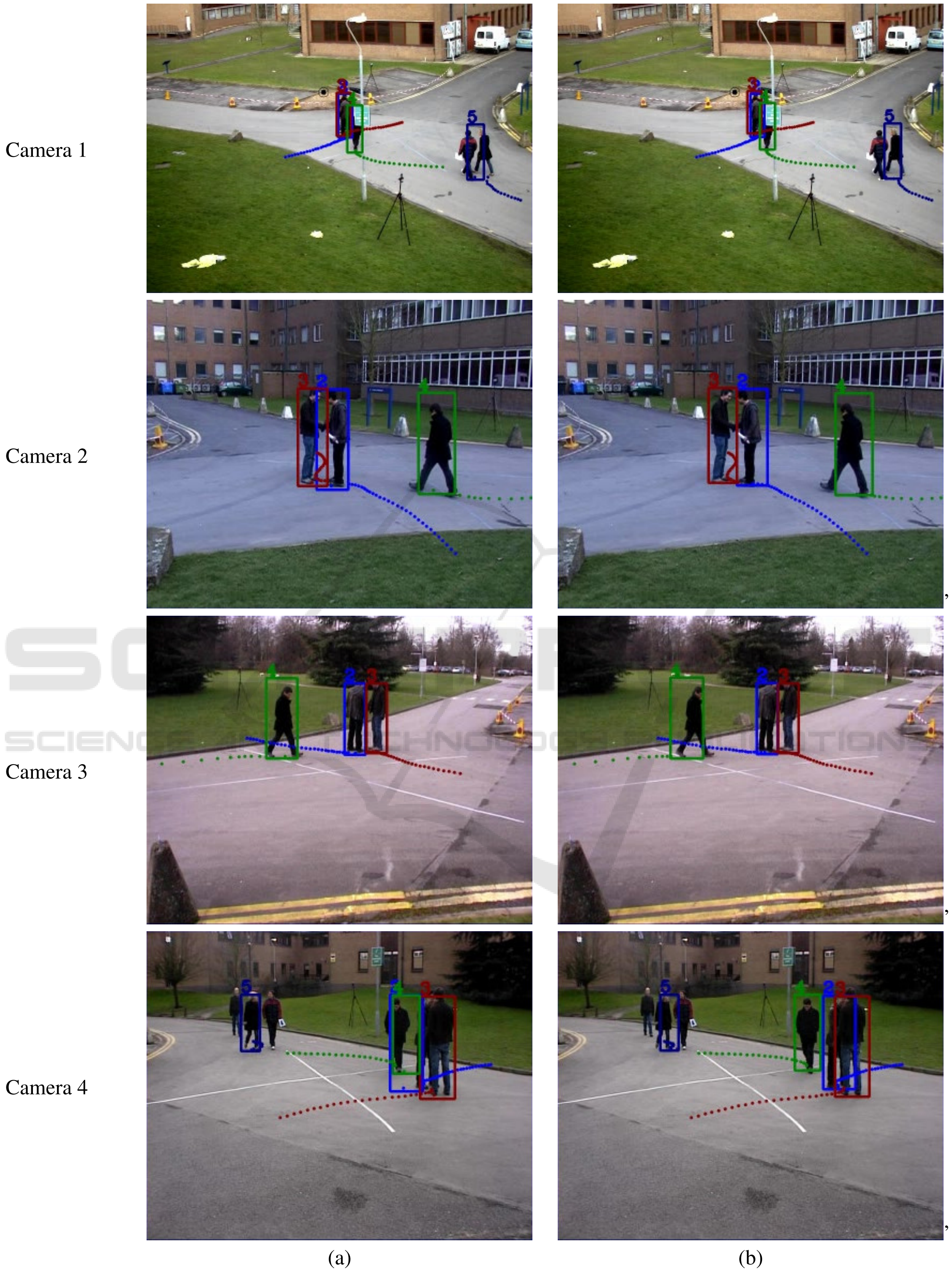
Suppose that a local estimation of a person's position made by a tracker on the camera A is accurate although the person is heavily occluded by other persons and the camera B makes a huge positional error in its local estimation although the person is not occluded in its view. In this situation, our method gives lower uncertainty to the local estimation of the camera B and higher uncertainty to the local estimation of the camera A. Therefore, the fused position will be closer to the position estimate of the camera B and will have a large positional error. A large positional error sometimes causes a tracker to drift away from a target which can lead to the tracking loss and the identity switching problems. Fortunately in practice, only a few cameras in the network usually make such mistakes and accurate estimations from the remaining cameras usually pull the fused positions closer to the true positions. Therefore our method achieves a better MOTA than the triangulation methods in videos that contain heavy/full person–person occlusion.

An example of a large positional error in the local estimation caused by severe occlusion can be seen in the view of camera 4 in Fig. 4 (a). Since *Person 2* is almost fully occluded by a person with ID 3 (denoted as *Person 3*), the local tracker is not able to estimate the position of *Person 2*. Due to this error, the projected bounding box of *Person 2* is severely misaligned with the actual person in the view of camera 4. Although the person–person occlusion in the view of camera 1 in Fig. 4 (a) is severe, a local tracker is able to make accurate local estimates since the projected bounding boxes are well-aligned with their corresponding targets. However, the fusion center quantifies the uncertainty of each local estimate based on how much a person is involved in the occlusion with other persons in its local simulation.

The fusion method assigns higher uncertainty to local estimates of *Person 2* from the camera 1 and 4. The local estimate of *Person 2* from the camera 2 has lower uncertainty and the camera 3 gets the lowest uncertainty. Therefore probability distributions of the camera 2 and 3 are having smaller standard deviations than the distribution of the camera 1 and 4 as shown in Fig. 3 (a). Moreover, it is visually difficult to locate the peak of the distribution for the camera 4 in Fig. 3 (a) since its standard deviation is very large. Since our method suppresses the influence of local estimates from views with heavy occlusion and relies on views with less occlusion, the fused position is closer to peak of the distribution of the camera 2 and 3 as shown in Fig. 3 (b). The projected bounding boxes of the fused estimated are depicted in Fig. 4 (b). Well-aligned projected bounding boxes over all targets indicates that the fused positions are accurate.

## 5 CONCLUSION

We presented a symbol level data fusion method for efficiently fusing the locally estimated positions from trackers deployed on smart cameras. Our method assigns the uncertainty of each local estimate by assessing how much a person is covered by other persons in the view of the camera. A key contribution of our method is its genericity since any tracking method can be deployed on camera nodes as long as they produce position estimated in a common coordinate system. Moreover, the proposed method allows highly scalable multi-camera tracking since a new camera can be added without worrying about communication and



(a) (b)

Figure 4: Projection of (a) locally estimated and (b) fused positions on the image plane of each camera. The same target in different views is shown with bounding boxes in the same ID number above the bounding boxes.

computation bottlenecks.

We demonstrated the performance of our method by an evaluation on three multi-camera video sequences, confirming the accuracy improvement over the classical triangulation method when there are frequent and severe occlusions. Performance comparison with state-of-the-art trackers on the widely used *PETS2009* video sequence shows that our tracker outperforms other methods. Furthermore, the analysis of the local estimations as well as the fused result reveals that huge positional errors in local estimation often correspond to occlusion and that our fusion method is able to minimize these errors.

As future work, we will explore the possibility of integrating other view specific attributes, which can potentially correlate to the accuracy of the local position estimations, into the proposed fusion method. These attributes include calibration accuracy at the target's position, distance between the target and the camera, and so on. We will also conduct experiments to show the genericity of our fusion method by deploying different single view tracking algorithms on different camera views and observing the accuracy improvement in the fused results.

## ACKNOWLEDGEMENTS

The work was financially supported by FWO through the BOF–GOA project the project 01GA2111W "Distributed Smart Camera Systems".

## REFERENCES

Andriyenko, A. and Schindler, K. (2010). Globally optimal multi-target tracking on a hexagonal lattice. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 466–479.

Andriyenko, A. and Schindler, K. (2011). Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1265–1272.

Berclaz, J., Fleuret, F., Turetken, E., and Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819.

Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: The clear mot metrics. *J. Image Video Process.*, 2008:1–10.

Bo Bo, N., Deboeverie, F., Eldib, M., Guan, J., Xie, X., Nio, J., Van Haerenborgh, D., Slembrouck, M., Van de Velde, S., Steendam, H., Veelaert, P., Kleihorst, R., Aghajan, H., and Philips, W. (2014a). Human mobility monitoring in very low resolution visual sensor network. *Sensors*, 14(11):20800–20824.

Bo Bo, N., Deboeverie, F., Veelaert, P., and Philips, W. (2015). Real-time multi-people tracking by greedy likelihood maximization. In *Proceedings of the 9th International Conference on Distributed Smart Cameras*, ICDSC '15, pages 32–37, New York, NY, USA. ACM. [doi:10.1145/2789116.2789125].

Bo Bo, N., Deboeverie, F., Veelaert, P., and Philips, W. (2016). Multiple people tracking in smart camera networks by greedy joint-likelihood maximization. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 602–609.

Bo Bo, N., Gruenwedel, S., Van Hese, P., Niño Castañeda, J., Van Haerenborgh, D., Van Cauwelaert, D., Veelaert, P., and Philips, W. (2012). Phd forum: Illumination-robust foreground detection for multi-camera occupancy mapping. In *Proceedings of the Sixth International Conference on Distributed Smart Cameras (ICDSC)*.

Bo Bo, N., Grünwedel, S., Van Hese, P., Guan, J., Nio-Castaeda, J., Van Haerenborgh, D., Van Cauwelaert, D., Veelaert, P., and Philips, W. (2014b). Illumination-robust people tracking using a smart camera network. In *PROCEEDINGS OF SPIE, Intelligent Robots and Computer Vision XXXI: Algorithms and Techniques*, volume 9025, pages 90250G–90250G–10.

Bredereck, M., Jiang, X., Korner, M., and Denzler, J. (2012). Data association for multi-object tracking-by-detection in multi-camera networks. In *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6.

Du, W. and Piater, J. (2006). Data fusion by belief propagation for multi-camera tracking. In *2006 9th International Conference on Information Fusion*, pages 1–8.

Du, W. and Piater, J. (2007). Multi-camera people tracking by collaborative particle filters and principal axis-based integration. In *Proceedings of the 8th Asian Conference on Computer Vision - Volume Part I*, ACCV'07, pages 365–374, Berlin, Heidelberg. Springer-Verlag.

Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2008). Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):267–282. [doi:10.1109/TPAMI.2007.1174].

Gruenwedel, S., Jelača, V., Niño Castañeda, J., Van Hese, P., Van Cauwelaert, D., Van Haerenborgh, D., Veelaert, P., and Philips, W. (2014). Low-complexity scalable distributed multi-camera tracking of humans. *ACM Transactions on Sensor Networks*, 10(2).

Grünwedel, S., Jelaa, V., Niño Castañeda, J., Van Hese, P., Van Cauwelaert, D., Veelaert, P., and Philips, W. (2012). Decentralized tracking of humans using a camera network. In Roning, J. and Casasent, D., editors, *PROCEEDINGS OF SPIE, Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, volume 8301. SPIE.

Henriques, J. F., Caseiro, R., and Batista, J. (2011). Globally optimal solution to multi-object tracking with merged measurements. In *IEEE International Con-*

*ference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2470–2477.

Khan, S. and Shah, M. (2000). Tracking people in presence of occlusion. In *In Asian Conference on Computer Vision*, pages 1132–1137.

Luo, R. C. and Kay, M. G. (1990). A tutorial on multisensor integration and fusion. In *Industrial Electronics Society, 1990. IECON '90., 16th Annual Conference of IEEE*, pages 707–722 vol.1.

Mittal, A. and Davis, L. S. (2003). M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Int. J. Comput. Vision*, 51(3):189–203.

Mori, T., Matsumoto, T., Shimosaka, M., Noguchi, H., and Sato, T. (2008). Multiple Persons Tracking with Data Fusion of Multiple Cameras and Floor Sensors Using Particle Filters. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2 2008*, Marseille, France. Andrea Cavallaro and Hamid Aghajan.

Munoz-Salinas, R., Medina-Carnicer, R., Madrid-Cuevas, F., and Carmona-Poyato, A. (2009). Multi-camera people tracking using evidential filters. *International Journal of Approximate Reasoning*, 50(5):732 – 749. [doi:10.1016/j.ijar.2009.02.001].

Niño-Castañeda, J., Frías-Velázquez, A., Bo, N. B., Slembrouck, M., Guan, J., Debard, G., Vanrumste, B., Tuytelaars, T., and Philips, W. (2016). Scalable semi-automatic annotation for multi-camera person tracking. *IEEE Transactions on Image Processing*, 25(5):2259–2274.

Yang, J., Vela, P. A., Shi, Z., and Teizer, J. (2009). Probabilistic multiple people tracking through complex situations. In *11th IEEE International Workshop on PETS*, pages 79–86.