Low-Complexity Scalable Distributed Multicamera Tracking of Humans

SEBASTIAN GRUENWEDEL, VEDRAN JELACA,
JORGE OSWALDO NINO-CASTANEDA, PETER VAN HESE,
DIMITRI VAN CAUWELAERT, DIRK VAN HAERENBORGH,
PETER VEELAERT, and WILFRIED PHILIPS, Ghent University TELIN-IPI-IBBT

Real-time tracking of people has many applications in computer vision, especially in the domain of surveillance. Typically, a network of cameras is used to solve this task. However, real-time tracking remains challenging due to frequent occlusions and environmental changes. Besides, multicamera applications often require a trade-off between accuracy and communication load within a camera network. In this article, we present a real-time distributed multicamera tracking system for the analysis of people in a meeting room. One contribution of the article is that we provide a *scalable* solution using smart cameras. The system is scalable because it requires a very small communication bandwidth and only light-weight processing on a "fusion center" which produces final tracking results. The fusion center can thus be cheap and can be duplicated to increase reliability.

In the proposed decentralized system all low level video processing is performed on smart cameras. The smart cameras transmit a compact high-level description of moving people to the fusion center, which fuses this data using a Bayesian approach. A second contribution in our system is that the camera-based processing takes feedback from the fusion center about the most recent locations and motion states of tracked people into account. Based on this feedback and background subtraction results, the smart cameras generate a best hypothesis for each person.

We evaluate the performance (in terms of precision and accuracy) of the tracker in indoor and meeting scenarios where individuals are often occluded by other people and/or furniture. Experimental results are presented based on the tracking of up to 4 people in a meeting room of 9 m by 5 m using 6 cameras. In about two hours of data, our method has only 0.3 losses per minute and can typically measure the position with an accuracy of 21 cm. We compare our approach to state-of-the-art methods and show that our system performs at least as good as other methods. However, our system is capable to run in real-time and therefore produces instantaneous results.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Network**]: Distributed Systems—*Distributed applications*; I.4.8 [**Image Processing and Computer Vision**]: *Sensor fusion*; I.5.4 [**Pattern Recognition**]: Applications—*Computer vision*

General Terms: Theory, Algorithms, Performance

Additional Key Words and Phrases: Distributed computer vision, sensor fusion, computer vision, people tracking, multicamera tracking

This research was funded in part by the IBBT iCOCOON and VAUE projects co-funded by IBBT (Interdisciplinary institute for Broadband Technology) a research institute founded by the Flemish Government. Companies and organizations involved in the iCOCOON project are Alcatel-Lucent Bell, VITO nv and Eyetronics, with project support of IWT. The work was also sponsored by the Flemish Fund for Scientific Research, through the project "Multi-camera human behavior monitoring and unusual event detection" (G.0.398.11.N.10).
Author's address: Ghent University TELIN-IPI-IBBT, Sint Pietersnieuwstraat 41, 9000 Gent, Belgium; email: sebastian.gruendwedel@telin.ugent.be.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2014 ACM 1550-4859/2014/01-ART24 $15.00
DOI: http://dx.doi.org/10.1145/2530282

ACM Reference Format:
Sebastian Gruenwedel, Vedran Jelaca, Jorge Oswaldo Nino-Castaneda, Peter Van Hese, Dimitri Van Cauwelaert, Dirk Van Haerenborgh, Peter Veelaert, and Wilfried Philips. 2014. Low-complexity scalable distributed multicamera tracking of humans. ACM Trans. Sensor Netw. 10, 2, Article 24 (January 2014), 32 pages.
DOI: http://dx.doi.org/10.1145/2530282

1. INTRODUCTION

Real-time tracking of people is an essential component of many computer vision applications, of which security and surveillance [Pflugfelder and Bischof 2010] of individuals for path-retracing is the best known application. However, other applications are emerging. For instance, in video-conferencing, positional data for each meeting attendant can be very valuable. It can be used to define regions of interest containing people, so as to limit more detailed processing to those areas. It can be helpful to focus pantiltzoom cameras (PTZ camera) on specific people [Aghajan and Cavallaro 2009], to determine when they enter and leave the room, to determine their identity even when they do not currently face a camera, and even to infer some activities [Fathi et al. 2011] such as getting a cup of coffee, . . .

In the above applications, which focus on individual tracks of individual people, avoiding tracking loss is essential, that is, tracks of individuals should not be lost due to occlusions and individuals should not be mixed up when they get close together. One way to avoid this problem is to rely on high-level feature analysis [Babenko et al. 2011], for instance, to periodically reidentify people. Such algorithms are computationally intensive and it is often better to restrict their usage. For instance, they can be activated when there is doubt about the current tracks or they can be run every few seconds only. Alternatively, high-level algorithms can be used to correct tracking losses when they are already executed for other purposes, as exemplified in our experimental setup. Here, we analyze people's faces when they are entering the room or when they are seated in front of a web-cam [Deboeverie et al. 2011]. This information can correct some tracking loss problems, but often with a large delay. In conclusion, while algorithms relying on high level analysis are certainly valuable, it is still very important for any tracking algorithm to minimize tracking loss in the first place.

In addition, tracking accuracy is a very important requirement in the above applications. On the one hand, a higher tracking accuracy helps to reduce tracking losses [Aghajan and Cavallaro 2009], so a reduced accuracy is a sensitive indicator of "near losses". Also, a high accuracy is needed for some types of detailed behavior analysis, for instance, to detect interactions between people (people who know each other well stand closer to each other) [Fathi et al. 2011]. For instance, in video-conferencing applications people sometimes move close to one another. In such an application, the tracking accuracy should not be better than the width of a person.

Reliable accurate tracking of multiple people in crowded scenes is still a very challenging task, mainly due to frequent occlusions and environmental changes. Even detecting and tracking a single nonoccluded person sometimes poses problems for state-of-the-art single-camera tracking algorithms, for instance, due to poor illumination or lighting changes. Tracking multiple people in the presence of furniture and other obstacles poses many additional problems. In this case, the problems are greatly reduced by using a top view rather than a side view camera.

While tracking may be possible with a single top view camera [Ozturk et al. 2009], joint analysis of multiple camera streams increases robustness in most applications [Aghajan and Cavallaro 2009] especially in highly cluttered indoor scenes. Moreover, most applications need side view cameras to enable more detailed analysis or visualization (e.g., of people's faces), so one might as well use them for people tracking.

More specifically, the principles of triangulation can help to estimate the positions of people with high accuracy from side view cameras. If enough cameras are available, problems due to occlusions can also be avoided and a top view camera may not even be needed.

While this article focuses on individual tracks of people, it should be noted that other applications rely on accurate statistics rather than on accurate individual tracks. For instance, in elderly care, behavior analysis based on track statistics such as walking speed, track smoothness and average activity levels can yield important information about physical and mental health degradation over long periods of time [Kröse et al. 2011]. Similarly, in a work environment, statistics about the time workers spend sitting, walking, and standing up can help to pinpoint productivity problems or potential health hazards, such as taking too few breaks. In marketing applications, they can provide information about the efficiency of billboards, etc. All these applications can tolerate some errors in individual tracks, which is important since reliable tracking over long periods of time in difficult environmental conditions is still a major problem. For this reason, the latter applications may actually be the easier ones to bring into practice. However, even in these applications we should strive for minimal tracking loss and maximal accuracy.

In this article, we also focus on real-time, low-latency and scalable tracking of multiple people, which adds another level of complexity compared to state-of-the-art papers such as Berclaz et al. [2011]. However, real-time and low-latency operation is needed in many indoor tracking applications, because they need to react quickly to changes in people's positions, for instance, to select appropriate high-resolution views for display and to run detailed analysis algorithms. Examples of such applications include surveillance, building occupancy monitoring, tele-classing (to focus one or more cameras on the moving presenter), etc. These applications typically involve long-term monitoring. Moreover, even in off-line applications (such as studying motion patterns) fast operation is essential so that the analysis can keep up with the data acquisition; faster than real-time operation may be required.

An often overlooked problem in multicamera research is that of scalability: centralized processing of multiple video streams creates not only a computing but also a communication bottleneck. From this point of view, multicamera tracking approaches can be categorized into centralized, decentralized and distributed tracking approaches [Taj and Cavallaro 2011]. Centralized approaches transmit all video streams to one or more servers (fusion centers) and process the video on these servers. The servers need to be very powerful computers and need to be able to sustain high communication bandwidths. Decentralized and distributed tracking approaches group cameras into clusters which communicate with a local fusion center (decentralized) or with each other (distributed tracking).

Combining real-time, low-latency, scalability, accuracy, and tracking loss requirements is highly non-trivial. In recent years it has become possible, with the deployment of smart networked cameras [Hengstler and Aghajan 2006; Hengstler et al. 2007] to shift the computation load towards the camera. Therefore, in this article, we consider a decentralized processing architecture, in which the most compute-intensive video processing is performed within smart cameras. In fact, since the requirements on the server are so low, it is even possible to run the server algorithms on each camera and end up with a distributed architecture. In this case, a camera essentially fuses its estimates with information received from neighboring cameras [Taj and Cavallaro 2011]. In our architecture no video transmission is needed for the purpose of tracking, not even for regions of interest within the camera views. When video transmission is needed for other purposes, the positional information provided by the tracking system can help to reduce the overall video bandwidth by restricting transmission to regions of interest.

However, many detailed image analysis algorithms, for instance, face recognition, can run on a single camera and do not require video transmission.

In our system, each camera first performs low-complexity foreground/background (FG/BG) segmentation, to segment the scene into moving blobs on a static background. The FG/BG algorithm is based in edge statistics and is more robust against light changes than other algorithms [Gruenwedel et al. 2011].

Next, each camera groups the blobs into bounding boxes ("cuboids") with respect to a world coordinate system, which most likely correspond to individual persons. Since we assume calibrated cameras, we are estimating the person's cuboid in world coordinates rather than in image coordinates. This allows a physical motion model to be expressed more easily than a model in the image domain where apparent speeds depend on the position of the person w.r.t. the camera. Furthermore, the estimates of a cuboid in world coordinates can be directly transmitted to a fusion center or even to other cameras without knowing the relationship between the image domain and the world coordinate system of this camera.

One approach to do so would be to track the motion of each blob using advanced Motion Estimation (ME) techniques; more specifically, using pixel based techniques such as optical flow as in [Grünwedel et al. 2012] or by tracking SIFT, SURF, FAST, etc. features within the blobs [Anjum and Cavallaro 2009]. Rather, in our article we aim to show that tracking is possible using simple FG/BG segmentation only, in combination with extremely simple blob analysis: For blob tracking, we rely on feedback from the fusion center on the most recent positions, speeds and geometries of individuals in the scene. Based on this feedback, we perform probabilistic occlusion reasoning in the camera to identify which blobs belong to which cuboid. The analysis also yields updated cuboid parameters (e.g., positions).

Specifically, our method does not involve sophisticated ME in each camera; However, if ME needs to be performed anyway to support more detailed video analysis for other purposes, our method can be modified to also use available ME information.

We will demonstrate that our approach works reasonably well, that is, that a simple analysis of changes in pictures, rather than ME results, is reliable and accurate for tracking in the multicamera case. Specifically, we can track up to four people using as little as six cameras without tracking losses in moderately complex environments, and less than two tracking losses per minute in sequences with abundant occlusion. The average accuracy is about 21 cm. These results are as good as state of the art algorithms as reported by Berclaz et al. [2011] and Fleuret et al. [2008]. However, we achieve those results in a real-time, low-latency, and scalable system, requiring low computational and network resources. In contrast, the methods in the cited papers are off-line methods. The method of Berclaz et al. [2011] moreover uses images of up to five seconds in the future and hence an on-line version of that method would incur a delay of about five seconds.

Our system has a very low communication overhead: alone a frequency of 10 FPS is sufficient for each camera to transmit the parameters (position, speed, width and height) together with a reliability measure of each tracked cuboid to the fusion center and nothing else. These *geometrical descriptors* are integrated on the fusion center, which sends fused descriptors for all individuals back. The resulting transmission bandwidths from camera to the fusion center and back are in the order of $\approx$900 Bytes/second per person. The low communication overhead results in a highly scalable system. Moreover, it is an asset in battery operated smart cameras, where battery lifetime is mostly limited by communication power [Taj and Cavallaro 2011]. It is also an asset in ad-hoc temporary setups, where wireless networks are preferred to avoid building works for installing cables.

The basic statistical estimation framework in our system, in the fusion center, is a Bayesian estimator. In each camera we use two approaches, namely a *hypothesis testing* and a *Kalman filter based* approach. Each approach in the camera obtains estimates for each person based on FG/BG segmentation and the feedback from the fusion center. Since those estimates are in world coordinates, uncertainties are introduced by the back-projection from the image domain to the world coordinate system. Nevertheless, the fusion center uses a linear Kalman Filter (KF) to obtain a joint decision of all cameras for each person. The final estimates are fed back to each cameras to minimize the uncertainties of the camera estimates.

The article is structured as follows: In Section 2, we discuss related work. Section 3 describes the overall decentralized system architecture containing a Section 3.2 describing the video processing and cuboid estimation within each camera. Furthermore, in Section 3.3 the information fusion at the server side is explained. Section 4 presents experiments to demonstrate the tracking performance (accuracy, precision) and computational and communication overhead. We compare our methods to Berclaz et al. [2011]. We designed an experimental setup and evaluate our method on test data from that setup. We also show results on a publicly available dataset of Berclaz et al. [2011], [Fleuret et al. 2008]. The results show that our system performs as accurately and robustly as the other methods, but has the advantages of being real-time and low-latency. More importantly, it is highly scalable and imposes little communication overhead. Section 5 concludes the article.

2. RELATED WORK

In this section, we provide an overview of state-of-the-art approaches in context with our proposed system architecture. We consider both single- and multicamera approaches as there are single-camera approaches with extension to multicamera ones, as well as different conceptional architectures. Reviews about both approaches and conceptional architectures exist. In Yilmaz et al. [2006] single-camera tracking approaches are discussed. Taj and Cavallaro [2011, 2010], Aghajan and Cavallaro [2009], Smith and Singh [2006], and Liu et al. [2007] focus on conceptional architectures and multicamera tracking approaches to which the reader is referred to for more detail.

2.1. Monocular Approaches

Monocular approaches have been an active research topic in the past two decades, ranging from detection to tracking algorithms for a single or multiple target(s) using only one camera view. Most algorithms use features which range from simple cues, such as color, shape or texture, to more complicated ones as classification with on-line adaptation. Blob-tracking is one of the most popular low-cost approaches for tracking objects [Collins 2003]. Usually blobs are detected on a frame by frame basis and are tracked by comparing their shape, location and appearance from one frame to another. This approach is followed, even in the case of occlusions.

The BraMBLe tracker [Isard and MacCormick 2001], for example, is a Bayesian multiblob tracker which computes the likelihood for each blob based on a known background model and appearance model of tracked people. It uses particle filtering to track an unknown number of people. Problems arise when objects merge into one blob with other close-by objects or with occluding objects, degrading the performance of this tracker. However, tracking can be improved by taking multiple cues into account.

In Giebel et al. [2004], Bayesian tracking based on particle filters is combined with a detector using learned spatio-temporal shapes to perform multicue 3D object tracking in a single camera. Their spatio-temporal object representation involves a set of distinct linear subspace models or Dynamic Point Distribution Models (DPDMs) and is learned fully automatically from training data. Furthermore, the representation

is enriched with texture information by means of intensity histograms and 3D measurement provided by a stereo system.

The results are tested on a small dataset and quite impressive but require shape, texture and image depth information to reliably track objects. This puts them in the category of more complex techniques which require significant computation time as well as the approach of Babenko et al. [2011]. The authors present a tracking technique based on the concept of "tracking by detection." In their approach, they use Multiple Instance Learning (MIL) to train a discriminative classifier in an online manner to separate the object from the background based on Histogram of Oriented Gradients (HOG) features. The algorithm is designed to track one object.

Smith et al. [2005] use particle filtering based on Markov chain Monte Carlo optimization to track people and handle entrances and departures using a fixed camera. Their framework uses a joint multiobject state-space formulation to recursively estimate the multiobject configuration and efficiently search the state-space by using particle filtering. As a global appearance model, binary images based on background subtraction together with foreground and background color statistics are used to discriminate between different objects in the scene.

However, there is an important difference between those papers and our approach: Our proposed system architecture utilizes fusion of multiple camera views simultaneously and a feedback channel for incorrect associations that combines the task of detection and tracking seamlessly.

2.2. Multicamera Approaches

Detection and tracking of multiple, possibly occluded, people in complex environments is a challenging task which makes multiple cameras indispensable: The different viewpoints offered by multiple cameras decrease the number and size of occluded regions. Also, multiple cameras simplify 3D analysis of the scene and provide redundant information which can help improve robustness.

A series of papers addressed multitarget tracking using the principle of associating objects in multiple views. For example, in [Nakazawa et al. 1998], human tracking is performed using template matching to track moving people. A state transition map together with action rules is used to coordinate between cameras. The state of the state transition map is described as three kinds of areas according to the camera coverage; areas visible to only one camera, areas visible to multiple cameras and areas visible to no camera. This state transition map stores the camera and view parameters of all cameras, while the action rules instruct each camera how to act.

Cai and Aggarwal [1998] rather adopted an optimal view selection approach: their tracker is basically a single-camera one, based on a Bayesian classification scheme. However, the tracker switches to another camera as soon as the current camera has no longer a good view of the tracking target. When to switch cameras is predicted by the tracking system for the position of an object along a spatial-temporal domain. As internal state, the tracker gathers sample pictures of upper human bodies as seen from various viewing angles. Nonhuman moving objects are excluded using Principal Component Analysis (PCA).

Bayesian networks are another popular approach to address the problem of multicamera tracking. Chang and Gong [2001] used a Bayesian network approach to combine geometry (epipolar geometry, homographies, and landmarks) and recognition (height and appearance) based features for matching objects between consecutive image frames and multiple camera views. Dockstader and Tekalp [2001] used also Bayesian networks to track objects and resolve occlusions in multiple calibrated cameras. Nillius et al. [2006] assume the existence of an isolated track graph and therefore focus more on the high-level tracking task. The goal of the article is to associate the identities of those

graph tracks. The problem is formulated as a Bayesian network inference which uses standard message propagation to find the most probable set of paths in an efficient way. Results of the multiobject tracking is applied on soccer players.

Stereo vision is used in Darrell et al. [2001], Krumm et al. [2000], and Mittal and Davis [2003]. For example, Krumm et al. [2000] use a stereo camera approach wherein depth information from multiple stereo cameras are combined in 3D space. Firstly, background subtraction is performed and then human-shaped blobs are detected in 3D space. Afterwards a distribution for each person based on color histograms are created to identify, and together with the blobs, are used to track multiple people.

Nevertheless, the underlying problem is that any type of features (like appearance, color, blob shapes, etc.) remains: they are easily corrupted due to occlusions or environment or lighting changes.

Khan and Shah [2006, 2009] use a homographic occupancy constraint to fuse foreground evidence retrieved by a background subtraction method from multiple cameras by geometrical constructs. The homographic occupancy constraint interprets foreground as scene occupancy by nonbackground objects and states that pixels corresponding to occupancies on a reference plane will consistently warp to foreground regions in every view. Their method resolves occlusions by localizing people on multiple reference planes and attempts to find image locations of scene points that are occupied by people.

Similar work was presented in Mittal and Davis [2003], Franco and Boyer [2005], Berclaz et al. [2006], and Fleuret et al. [2008]. Berclaz et al. [2011], and Fleuret et al. [2008], estimate probabilities of occupancy on the ground plane given binary images obtained by background subtraction. They use a generative model representing humans as simple rectangles to approximate the probabilities of occupancy at every location as the marginals of a product law minimizing the Kullback-Leibler divergence from the condition posterior distribution. Optimal tracks are computed from the raw observations by a greedy search strategy based on Dynamic Programming.

In a later extension [Berclaz et al. 2011], the trajectory estimation is treated as a constrained flow problem. This results in a convex optimization problem which is solved using the k-shortest paths algorithm. The results show a very good performance on difficult real-word applications.

Many of the aforementioned papers represent the accumulated knowledge on the location of people by occupancy maps, which emanated in research on robot navigation using range-sensor based sensors [Elfes 1989; Thrun 2003]. However, occupancy maps based methods usually perform poorly when humans are partially hidden (e.g., by furniture) or if the input data (result of background subtraction) is noisy or even not existent due to environmental changes in at least one of the cameras. The main reason is that they assume that pixels corresponding to occupancies will warp to foreground regions in every camera view. This assumption is not always valid and can therefore lead to errors in some of the occupancy maps, eventually resulting in tracking errors.

However, the major difference to our approach is that we do not use a discretization of the ground plane into grid cell. Instead, our approach obtains estimates of individuals using a Bayesian estimator in a continuous state space with respect to a world coordinate system. To do so, each camera makes its own estimates for each person and a final estimate is obtained by a fusion of all camera estimates on the server side. This has the advantage that cameras itself can make mistakes as long as the global estimates are correct. If this is not the case then the whole system will break down due to insufficient information, but this will happen in both approaches.

In Taj and Cavallaro [2009], a centralized tracking approach is presented where the input data from each camera view is projected on a top-view through the multilevel homographic transformation of Delannay et al. [2009], which projects foreground evidence to planes parallel to the ground plane. The projected planes are added up to

generate a detection volume. The method adopts a track-before-detect (TBD) approach to keep track of possible humans in the scene. In the TBD approach the entire image is considered as a measurement which is a highly nonlinear function of the target state. The target state consists of the position and speed of an object and the intensity of the image. In their approach it is solved by employing non-linear state estimation techniques such as particle filtering.

Anjum and Cavallaro [2009] used an unsupervised intercamera trajectory correspondence algorithm to link objects across a multicamera network. Association is implemented as a hybrid approach using local trajectory pairs estimated by multiple spatio-temporal features. Then image-plane reprojections of the matched trajectories are employed to resolve conflicting situations.

The latter approaches have high-data transfer rates due to the nature of centralized processing and therefore a lack of scalability and energy efficiency. However, our proposed approach intends to be scalable, efficient with respect to the communication load, and operates in real time due to the limited data exchange between cameras and the fusion center.

Other approaches for visual tracking range from Bayesian filtering algorithms to Probabilistic Graphical Models (PGMs). In Dore et al. [2010], a state-of-the-art review of Bayesian state estimation and PGMs, with respect to tracking applications is provided. In particular, in computer vision and video processing algorithms have been proposed based on different types of PGMs such as hidden Markov models (HMMs) or Kalman filter. In their review the authors describe PGMs as a statistical framework suitable for handling complex object representations. This framework enables consistent formalization and handling of uncertainties of visual observations. Moreover, this framework allows efficient solutions for complex problems meeting real-time requirements. One important step in these approaches is to model data association. Here, one commonly uses the Joint Probabilistic Data Association Filter (JPDAF) [Bar-Shalom 1987; Kirubarajan and Bar-Shalom 2004]. In Kirubarajan and Bar-Shalom [2004], an overview of the PDA technique and its application for different tracking scenarios is presented. This filter deals with multiple observations, assuming that an object emerges only by one true measurement and jointly estimates the solution for all objects.

Rasmussen and Hager [2001] use a constrained JPDAF filter for their randomized tracking algorithm which oversees correspondence choices between the tracker and image features. The algorithm is applied to three different kinds of tracking modalities, namely homogeneous regions, textured regions, and contours described as snakes. In addition, they consider depth ordering of tracked objects relative to the camera, resulting in the ability to predict occlusions between objects and allowing likelihoods coming from different cues.

Maggio et al. [2008] propose a filtering framework for multitarget tracking based on particle filtering and data association using graph matching. Their tracker is able to compensate for missing detections and remove noise and clutter produced by the detector. In their approach, a novel particle resampling strategy is proposed, and, moreover, the dynamic and observation models are adapted to cope with varying object scales.

There are some shortcomings of the JPDAF: the JPDAF does not consider situations in which multiple measurements can be assigned to one object or the same measurement is represented by two objects. Moreover, by using a JPDAF for data association, the processing time increases significantly with the number of objects since all possible hypotheses need to be calculated. It also cannot handle objects entering, or already tracked objects exiting the field of view.

Another approach for data association is Multiple Hypothesis Tracking (MHT) [Reid 1979]. In the MHT algorithm, several correspondence hypotheses for each object at each time are maintained and assessed. Using this approach, the correspondence decision is

deferred until the most likely set of observation correspondences is found. In MHT the probability of each potential track is calculated and typically only the most probable of all the tracks is reported. The algorithm has the ability to create or terminate tracks of objects, entering or exiting the field of view. Moreover, it can also handle occlusions in a way that the continuation of a track is found even if some of the measurements from the object are missing. MHT makes associations in a deterministic sense and exhaustively enumerates all possible associations.

A particle filtering approach that handles multiple measurements to track multiple objects has been proposed by Hue et al. [2002]. In their method, data association is handled in a similar way as in MHT, however, the state estimation is achieved through particle filters.

In comparison with the latter approaches we use a Bayesian filtering approach. These approaches provide efficient solutions [Dore et al. 2010], useful to achieve a scalable and efficient communication load due to the limited data exchange between cameras and the fusion center. However, in our proposed method we have a concrete data association strategy. In most of the cases, data association is straight forward since each camera sends a local estimate per person together with a reliability measure to the fusion center. The fusion center hereby fuses these local estimates for a specific person of all cameras to one global estimate. The data association strategy is described in a top-down manner, meaning that evidence for a specific person is gathered locally in a camera. In the special case of occlusions, where a concrete data association strategy is needed since it could introduce ambiguities, we use probabilistic foreground modeling (see Section 3.2.3) to perform occlusion reasoning. Furthermore, we use a hypothesis testing approach, like in MHT, to explore the posterior probability, taking all current measurements into account. However, we do not keep track of different hypothesis over time yet. This particular idea and the exchange of several possible hypothesis of a single object can lead to further improvements of our method.

3. THE PROPOSED SYSTEM

In this section, we describe our proposed decentralized tracking approach. Here, the server only operates on numbers (the states of people) and not on images. This allows the construction of huge smart camera networks without straining network and server resources. It also only requires simple processing in the smart cameras, leaving precious resources for other video processing algorithms if needed. We have built such a system with one fusion center and six (smart) cameras.

Figure 1 shows a block diagram of the system architecture. First, each smart camera computes FG/BG segmentation on the input video of the smart cameras to extract features. Simultaneously, each smart camera receives feedback from the server about the number of people and their positions, sizes and speed in the scene at time $t-1$. Of course, this estimate is slightly outdated as it is based on previous observations. However, especially at high frame rates this feedback enables an accurate estimate of the locations where people can be expected. The smart camera uses this feedback to generate a number of hypotheses about the location and size of a person at time t. It then tests these hypotheses using evidence from the FG/BG models. The result of this analysis in each smart camera is a set of zero or more cuboids describing the position, size and speed of people in world coordinates. These cuboids (if any) are sent together with a reliability measure to the fusion center.

The appearance of new people is handled by one single smart camera which observes a region of interest. If enough foreground is found in this region of interest then a new person is initialized in this smart camera at a fixed location on the ground plane. The current estimate of this smart camera is distributed to the fusion center which, on the other hand, communicates the new person to the remaining smart cameras.

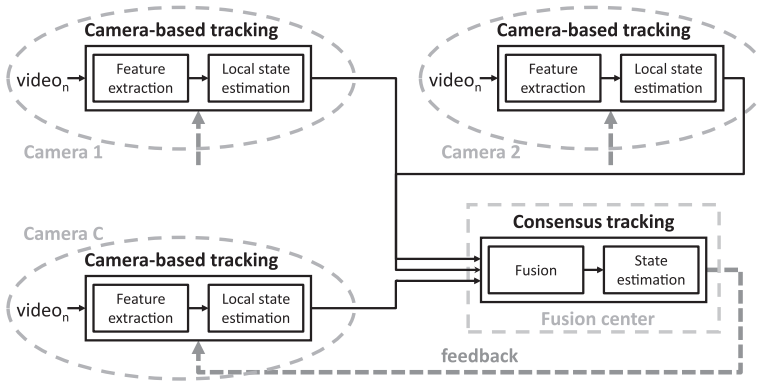

Fig. 1. *Decentralized system architecture*. In the camera-based tracking block, executed on a smart camera, features (foreground blobs) are extracted from the input video. Then the local states of all people (the position on the ground plane, speed, width and height of each person) are estimated. Afterwards a compact representation is sent to the fusion center, which fuses the individual estimates into a global estimate resulting in the best possible global state for each person. These states are fed back to each camera to correct possible mistakes.

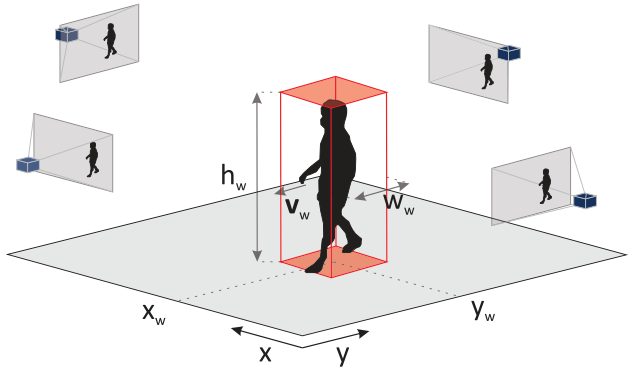

Fig. 2. *Person model*. A person is modeled as a "cuboid" with an attached speed vector. The cuboid model is described by its state, which is composed of: the location (x, y) on the ground plane, the speed $\mathbf{v} = (\dot{x}, \dot{y})$, the width and height (w, h). All of these numbers are expressed w.r.t. a world coordinate system.

The fusion center gathers data from each smart camera and calculates an overall estimate of the most likely global state of each person using an approach we call *consensus tracking*. It then distributes this information back to the cameras.

3.1. Problem Formulation

Our goal is to perform tracking of an a-priori unknown number of people. In this section, we formulate this problem as an estimation of the most probable *global state* of a hidden Markov process, given a set of *local estimates* of each camera obtained at each time t, denoted as *consensus tracking* (Figure 1).

As shown in Figure 2, we will model an individual as a cuboid of width w and height h at the location (x, y) on the ground plane, moving at instantaneous velocity $\mathbf{v} = (\dot{x}, \dot{y})$. All of these quantities are expressed w.r.t. a world coordinate system. They are different for each person $m = 1, \ldots, M$ and vary over time as the person moves. Together they constitute an unknown *state vector* $\mathbf{x}_t^m$:

$$\mathbf{x}_t^m = (x_\mathrm{w}, \dot{x}_\mathrm{w}, y_\mathrm{w}, \dot{y}_\mathrm{w}, w_\mathrm{w}, h_\mathrm{w})^T . \quad (1)$$

At first, we will focus on the mathematical description of the *consensus tracking*, that is, the tracking at the fusion center. Every camera c calculates an local estimate of each person m, denoted as $\widehat{\mathbf{x}}_t^{c,m}$, and sends the estimates together with a reliability measure, $P_t^{c,m}$, to the fusion center. We will explain in detail how these estimates in a camera are calculated in Section 3.2.2 and 3.2.4.

Given $\mathbf{Z}_t = \{\widehat{\mathbf{x}}_t^{1,1}, \ldots, \widehat{\mathbf{x}}_t^{C,M}\}$, that is, the set of local estimates of M people in C cameras at time t, our task is to find the most probable global states, $\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M$, that maximize the posterior joint probability

$$p\big(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M | \mathbf{Z}_{1:t}\big),$$

of all M individuals given all local camera estimates up to the current time t.

Using Bayes' theorem and the Markov assumption we obtain, as shown in Thrun et al. [2005], for all individuals that

$$p\big(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M | \mathbf{Z}_{1:t}\big) = \eta \cdot p\big(\mathbf{Z}_t | \mathbf{x}_t^1, \ldots, \mathbf{x}_t^M\big) p\big(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M | \mathbf{Z}_{1:t-1}\big), \tag{2}$$

where $\eta = p(\mathbf{Z}_t | \mathbf{Z}_{1:t-1})^{-1}$. The distribution $p(\mathbf{Z}_t | \mathbf{x}_t^1, \ldots, \mathbf{x}_t^M)$ is the likelihood of observing $\mathbf{Z}_t$ given all global state vectors at time instance t, whereas $p(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M | \mathbf{Z}_{1:t-1})$ is the predicted posterior probability from time $t-1$.

The likelihood $p(\mathbf{Z}_t | \mathbf{x}_t^1, \ldots, \mathbf{x}_t^M)$ specifies the probabilistic law according to which the estimates $\mathbf{Z}_t$ are generated from the global state vectors $\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M$ of all people. Since every camera calculates the estimates $\widehat{\mathbf{x}}_t^{c,m}$ of each person locally and sends a compact representation of each person m to the fusion center, it is appropriate to simplify Equation (2) and assume that all $\mathbf{x}_t^m$ are independent random variables, that is, that people's states are independent of those of other people. Moreover, this compact representation is essential for a real-time tracking system since the communication load is reduced. This assumption implies that for a particular global state $\mathbf{x}_t^m$ of person m, only part of the estimates $\mathbf{Z}_t$ of all people and all cameras are important, namely $\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m}$. This results in a direct data association, that is, the fusion center gets the local estimates of each camera c for a specific person m. In Section 3.2, we will explain why we can assume such a data association. Equation (2) can thus be decomposed into the estimation of its marginals independently as follows

$$p\big(\mathbf{x}_t^m | \mathbf{Z}_{1:t}\big) = \eta \cdot p\big(\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m} | \mathbf{x}_t^m\big) p\big(\mathbf{x}_t^m | \mathbf{Z}_{1:t-1}\big). \tag{3}$$

Each person is treated separately, that is, the fusion center uses a global Bayesian estimator for each person m which results in an efficient solution suited for real-time applications with limited data exchange.

The following two sections will explain in detail the estimation of $\widehat{\mathbf{x}}_t^{c,1}, \ldots, \widehat{\mathbf{x}}_t^{c,M}$, namely the *camera-based tracking* on the camera side (Section 3.2), and the estimation of $p(\mathbf{x}_t^m | \mathbf{Z}_{1:t})$, that is, the global Bayesian estimator per person m, denoted as *consensus tracking* on the fusion center (Section 3.3).

3.2. Video Processing and Tracking in Smart Cameras

In this section, we outline the calculation of the *local* estimates $\widehat{\mathbf{x}}_t^{c,1}, \ldots, \widehat{\mathbf{x}}_t^{c,M}$ in each camera c using two approaches: a *hypothesis testing* and a *Kalman filter based* approach. Each smart camera estimates the local state of people independently using either one of the two approaches. Note that both of our proposed approach only relies on FG/BG segmentation images as features.

Both approaches use not only the current camera's foreground mask, but also feedback from the fusion center which consists of the global posterior distributions of the global people's state $\mathbf{x}_{t-1}^m$ at earlier time instance $t-1$ calculated by a Bayesian estimator in the fusion center. The distribution for each person m is modeled as a Gaussian

with mean μ_{t-1}^{m} (the most likely global state of person m) and covariance matrix K_{t-1}^{m}. Section 3.3 explains in detail how these distributions are estimated.

The following subsections will explain the used FG/BG segmentation method, the *hypothesis testing* approach, how occlusions are handled using *probabilistic foreground modeling*, and the *Kalman filter based* approach.

Additionally, let us denote $\Omega_t^{c,m}$ as the image area obtained by the projection of the cuboid model associated with the local state $\mathbf{x}_t^{c,m}$ into the image of camera c. We define $\Omega_t^{c,m}(i)$ as a binary image where the pixel i describes being part of this projection, denoted as "1," or, if not, denoted as "0". Furthermore, let $FG_t^c(i)$ be a binary image which represents the result of FG/BG segmentation on camera c at time t.

3.2.1. Robust Foreground and Background Modeling. We use a method which we have previously developed [Gruenwedel et al. 2011], to which the reader is referred to for a detailed description. In this method, we propose to subtract foreground from background by detecting moving edges. Edges are detected by computing the edge strength, usually a first order derivative expression such as the gradient magnitude, and searching for local maxima. The FG/BG method detects moving edges via analysis of the image gradient and uses edge dependencies as statistical features of foreground and background regions. Foreground is defined as regions containing moving edges, and background as regions containing static edges of a scene. In particular, the FG/BG method is constructed to find edges on moving objects, using principles from change detection rather than more complicated motion estimation.

The background is described by a short- and long-term edge model using adaptive recursive smoothing for updating based on gradient estimates in x- and y-direction. The x and y components of both models are estimated independently. The first smoothing is applied with a very low learning factor and estimates the background of a scene, namely the long-term edge model. Due to the low learning factor, changes in the gradient estimates will be incorporated slowly into the background model. By comparing the background edge model to the recent gradient estimates, we might detect more edges than actually present because of the low learning factor. However, this situation is prevented by using a second smoothing approach, the short-term edge model, which is based on recursive smoothing with a higher learning factor. The two models are used jointly to obtain a foreground gradient estimate per direction, containing only regions where motion occur in the image.

In Figure 3, the block scheme of our FG/BG method is depicted. As input for the method, the gradient estimates in x- and y-directions, represented by two matrices $G_{x,t}$ and $G_{y,t}$, are calculated for the input image of frame t using the Sobel operator. The next step involves the comparison of the long- and short-term background edge models with the current gradient estimates resulting in the binary foreground masks $F_{x,t}^{l}$ and $F_{y,t}^{l}$ using hysteresis thresholding for the long-term and $F_{x,t}^{s}$ and $F_{y,t}^{s}$ using a fixed threshold for the short-term background edge models. The comparison per model is done using the differences between the background edge models and current gradient estimates instead of the absolute value of differences, which results in a better detection of moving edges. The resulting foreground gradient estimate, $G_{x,t}^{f}$, in x contains the gradient estimates, $G_{x,t}$, in foreground regions if, and only if, the binary masks $F_{x,t}^{s}$ and $F_{x,t}^{l}$ are one, otherwise zero values; and vice versa in y-direction. Finally, moving edges are extracted from the foreground gradient estimate using a nonmaximum suppression technique.

The foreground mask (silhouettes of moving people) is obtained by spatial clustering of the moving edges according to proximity, and computing the convex hull of each cluster. The clustering is needed to obtain silhouettes of moving people since our proposed algorithm works on FG/BG segmentation images. Future work will make use of

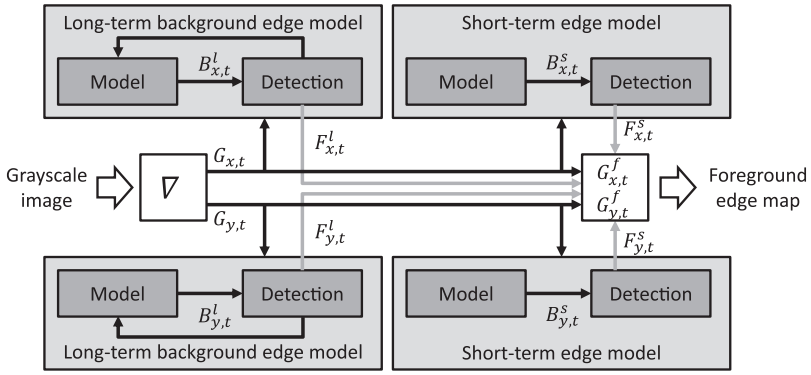

Fig. 3. *Robust Foreground and Background Modeling*. This scheme depicts the used FG/BG segmentation method, as presented in [Gruenwedel et al. 2011]. As input for the method, the gradient estimates in x- and y-directions are calculated for the input image of frame t using a discrete differentiation operator (e.g., Sobel operator). The gradient estimates are compared to the short- and long-term background edge models resulting in foreground gradient estimates. Finally, moving edges are extracted from the foreground gradient estimate using a non-maximum suppression technique.

the silhouettes for other purposes, such as appearance modeling based on color within the silhouettes. Furthermore, we will expand our proposed tracking approach so that foreground edges can be used directly. In Gruenwedel et al. [2011] we showed that the method is robust to sudden lighting changes and to spatially varying light distributions. This makes it more suitable for tracking applications than the other methods we have tested [Barnich and Van Droogenbroeck 2009; Zivkovic 2004].

3.2.2. Hypothesis Testing Approach. The local estimation in each camera c is described by maximizing the posterior joint probability of the local states $\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M}$, given the images of camera c acquired until time t

$$p\big(\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M} | I_{1:t}^c\big).$$

Applying Bayes' theorem and the Markov assumption we obtain, as shown in Thrun et al. [2005], for all local states that

$$p\big(\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M} | I_{1:t}^c\big) = \eta \cdot p\big(I_t^c | \mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M}\big) p\big(\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M} | I_{1:t-1}^c\big), \tag{4}$$

where $\eta = p(I_t^c | I_{1:t-1}^c)^{-1}$. The distribution $p(I_t^c | \mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M})$ is the likelihood of observing I_t^c given all local state vectors at time t, whereas $p(\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M} | I_{1:t-1}^c)$ is the predicted posterior probability from time $t-1$. The likelihood $p(I_t^c | \mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M})$ specifies the probabilistic law according to which the images I_t^c are generated from the local state vectors $\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,M}$ of all people in camera c. It is appropriate to think of I_t^c as noisy projections of the local state vectors. Unfortunately, maximizing the likelihood is a highly-complex and intractable optimization problem since it requires simultaneous optimization of all local state vectors and strongly depends on the features to represent the likelihood.

Therefore, we simplify Equation (4) and assume that all $\mathbf{x}_t^{c,m}$ are independent random variables, that is, that people's trajectories are independent of those of other people. Taking into account the assumed independence of the local state vectors, and moreover assuming that the projection of a person does not overlap with those of any other person, the posterior probability is well approximated as the estimation of its marginals

resulting in

$$p\left(\mathbf{x}_t^{c,m}|I_{1:t}^c\right) = \eta \cdot p\left(I_t^c|\mathbf{x}_t^{c,m}\right) p\left(\mathbf{x}_t^{c,m}|I_{1:t-1}^c\right). \tag{5}$$

While the assumption of independent random variables is not very restrictive in practice, the second assumption is clearly violated when one person occludes another. Note that a single person is rarely occluded in all cameras simultaneously. While the approximation of (5) maybe poor in some cameras, resulting in a bad local estimate $\widehat{\mathbf{x}}_t^{c,m}$ of person m in camera c, the tracking will still be good in at least some of the other smart cameras.

In our hypothesis testing approach, we approximate the predicted posterior probability $p(\mathbf{x}_t^{c,m}|I_{1:t-1}^c)$ from time $t-1$ by the feedback of the fusion center

$$p\left(\mathbf{x}_t^{c,m}|I_{1:t-1}^c\right) \approx p\left(\mathbf{x}_{t-1}^m|\mathbf{Z}_{1:t-1}\right),$$

described as a normal distribution $N_{\mathbf{x}_{t-1}^m}(\mu_{t-1}^m, K_{t-1}^m)$.

Using the uncertainties described in the covariance matrix K_{t-1}^m of the feedback, we can create an uncertainty area W_t^m around the last-known state $\mathbf{x}_{t-1}^m$ of the person m at time instance $t-1$. We consider for this uncertainty area W_t^m only possible locations of person m on the ground plane and treat the remaining state variables of $\mathbf{x}_{t-1}^m$ as constant. The assumption is that person m cannot have moved outside of W_t^m by time t and, furthermore, that all movements within this area are equally likely. Within this area we distribute N possible hypotheses, denoted as $\widehat{\mathbf{x}}_n^{c,m} \in W_t^m$ with $n = 1, \ldots, N$, and calculate for each hypothesis $\widehat{\mathbf{x}}_n^{c,m}$ the likelihood $p(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m})$. The results is a good approximation of the posterior distribution $p(\mathbf{x}_t^{c,m}|I_{1:t}^c)$ of each person m in camera c around the last-known state $\mathbf{x}_{t-1}^m$, as stated in Thrun et al. [2005]. The calculation of the likelihood $p(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m})$ for a particular hypothesis $\widehat{\mathbf{x}}_n^{c,m} \in W_t^m$ for the local state $\mathbf{x}_t^{c,m}$ is explained in detail in Section 3.2.3 using *probabilistic foreground modeling*.

For a scalable and efficient communication load, limited data exchange between cameras and the fusion center is essential. Therefore, we approximate the destitution $p(\mathbf{x}_t^{c,m}|I_{1:t}^c)$ by a Gaussian, $N_{\mathbf{x}_t^{c,m}}(\widehat{\mathbf{x}}_t^{c,m}, P_t^{c,m})$, with the mean $\widehat{\mathbf{x}}_t^{c,m}$, and a corresponding covariance matrix $P_t^{c,m}$. The mean $\widehat{\mathbf{x}}_t^{c,m}$ of the Gaussian is chosen either as the mean or the best hypothesis of the posterior distribution $p(\mathbf{x}_t^{c,m}|I_{1:t}^c)$, that is:

$$\widehat{\mathbf{x}}_t^{c,m} \propto \eta \cdot \sum_{\widehat{\mathbf{x}}_n^{c,m} \in W_t^m} p\left(\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m}|I_{1:t}^c\right) \cdot \widehat{\mathbf{x}}_n^{c,m},$$

where $\eta = (\sum_{\widehat{\mathbf{x}}_n^m \in W_t^m} p(\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m}|I_{1:t}^c))^{-1}$, or

$$\widehat{\mathbf{x}}_t^{c,m} \propto \underset{\widehat{\mathbf{x}}_n^m \in W_t^m}{\arg\max}\, p\left(\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m}|I_{1:t}^c\right),$$

respectively. The covariance matrix $P_t^{c,m}$ is estimated from all hypotheses or even learned over time since it will only be an approximation of the real distribution. For simplicity, we choose the covariance matrix $P_t^{c,m}$ as constant for this approach.

3.2.3. Occlusion Reasoning using Probabilistic Foreground Modeling. Even if we assume that images of a person are not affected by the presence of other persons, it is not fully valid since the results of the FG/BG segmentation method do not reflect this assumption (see Equation (5)). Especially if people create a common foreground blob in case of occlusions, the results will not be independent (Figure 4). The probability that a pixel is part of the individual's foreground blob depends on the foreground mask FG_t^c, but also on other individuals since the pixel in the foreground mask FG_t^c can also belong to others than the current individual. Whether or not a particular part of a persons silhouette is detected as foreground in the camera depends, of course, not only on geometrical considerations,

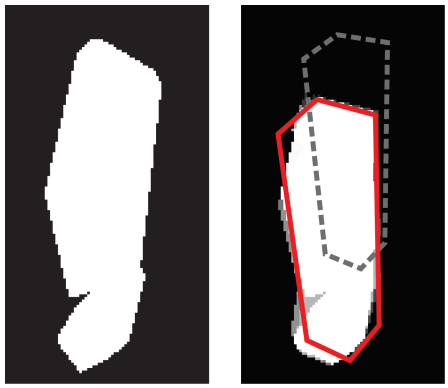

Fig. 4. *Occlusion Reasoning*. The principle of occlusion reasoning is to use only the visible part of the foreground mask FG_t^c (a) of camera c which is not affected by any occlusion (b) for person m. (b) shows this visible part for person m. The projected cuboid of person m is visualized as a solid line. In case of occlusion the foreground of all other person (dashed line) is not used to calculate the likelihood $p(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m})$.

but also on the speed of the person (immobile persons are invisible in FG/BG segmentation), their appearance (persons with the same color as the background are invisible), etc. Therefore a concrete data association strategy is needed. In case of no occlusion, the foreground blob of a person is not affected by the presence of other persons and the data association is straight forward. On the other hand, in the case of occlusions, this is not fully valid and the FG/BG segmentation results depend on other people. Hence, we need to treat the occlusion problem separately. In this case, we use *probabilistic foreground modeling* to perform occlusion reasoning, that is, we describe the foreground produced by a person w.r.t. the presence of other people. The principle of the probabilistic foreground modeling is to take only the person's visible part of the foreground mask from the camera's point of view, which is not affected by any occlusion, into account to calculate the likelihood $p(I_t^c|\mathbf{x}_t^{c,m})$ (Equation (5)). The likelihood calculation is essential for the hypothesis testing approach and specifies our data association strategy. In the following, we will explain how we are calculating the likelihood $p(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m})$ for a specific hypothesis $\widehat{\mathbf{x}}_n^{c,m}$ using probabilistic foreground modeling.

To understand the likelihood estimation, let us first consider the simple case in which we have *no occlusions* in the cameras. In this simple case and to calculate the likelihood $p(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m})$ for a particular realization of $\mathbf{x}_t^{c,m}$, denoted as hypothesis $\widehat{\mathbf{x}}_n^{c,m}$, we need to take the projection, $\Omega_n^{c,m}$, for this hypothesis within the image of camera c into account. Assuming all pixels of I_t^c are independent, the likelihood over the parameter $\mathbf{x}_t^m$ can be expressed as

$$p\left(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m}\right) = \prod_{i \in \Omega_n^{c,m}} \alpha^{FG_t^c(i)} (1-\alpha)^{1-FG_t^c(i)} \prod_{i \notin \Omega_n^{c,m}} \beta^{1-FG_t^c(i)} (1-\beta)^{FG_t^c(i)},$$

where α is the probability that the pixel $FG_t^c(i)$ is foreground given that the same pixel is part of the projection $\Omega_n^{c,m}$. The probability β, on the other hand, specifies that the pixel $FG_t^c(i)$ is background given that it lies outside the projection $\Omega_n^{c,m}$. The model is reasonable since it measures the agreement between our assumed person model (Figure 2) and the observed foreground mask.

However, there are rarely no occlusions in practice. For the purpose of the following discussion, consider that at each time a person may be fully or only partially visible in the camera of interest due to occlusions by other people. To calculate the likelihood

for a particular hypothesis $\widehat{\mathbf{x}}_n^{c,m}$ of person m, we fix the remaining local state vectors $\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,m-1}, \mathbf{x}_t^{c,m+1}, \ldots, \mathbf{x}_t^{c,M}$ and use the feedback from the fusion center at time $t-1$. Let $\omega_t^{c,m}$ be the visible part of the projection $\Omega_t^{c,m}$ for the local state vector $\mathbf{x}_t^{c,m}$ at time t according to the model described in Figure 2. A visible part of the projection hereby refers to the part of the projected silhouette which can be seen from the camera's point of view, taking all other projections of state vectors into account. This implies that, if person 1 is occluded by another person from this camera's point of view, only the visible part of this occlusion will be included in the projections $\omega_t^{c,1}$ in camera c. We define $\omega_t^{c,m}(i)$ as a binary image where the pixel i describes being part of the projected visible silhouette of local state $\mathbf{x}_t^{c,m}$, denoted as "1," or, if not, denoted as "0". Furthermore, we define an occlusion map, denoted as $OM_t^{c,m}$, for an individual m, which combines all visual projections of the remaining individuals $\mathbf{x}_t^{c,1}, \ldots, \mathbf{x}_t^{c,m-1}, \mathbf{x}_t^{c,m+1}, \ldots, \mathbf{x}_t^{c,M}$ as

$$OM_t^{c,m}(i) = f\left(\bigvee_{\substack{n=1 \\ n \neq m}}^{M} \omega_t^{c,n}(i)\right). \tag{6}$$

Here, the function $f(\cdot)$ models the uncertainty for a visible projection $\omega_t^{c,n}$ introduced by the fixation of the state with the last-known estimate at time $t-1$. The occlusion map $OM_t^{c,m}$ for an individual m characterizes the visual projections and their corresponding uncertainties for all individuals, except for individual m. Note, that the occlusion map $OM_t^{c,m}$ is still a binary image; uncertainty areas are modeled as being part of the projection and therefore are denoted as "1."

Therefore, the likelihood $p(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m})$ for a particular hypothesis $\widehat{\mathbf{x}}_n^m$, assuming that all pixels of I_t^c are independent, can be calculated as

$$p\left(I_t^c|\mathbf{x}_t^{c,m} = \widehat{\mathbf{x}}_n^{c,m}\right) = \prod_{\substack{i \in \Omega_n^{c,m} \\ i \notin OM_t^{c,m}}} \alpha^{FG_t^c(i)}(1-\alpha)^{1-FG_t^c(i)} \prod_{\substack{i \notin \Omega_n^{c,m} \\ i \notin OM_t^{c,m}}} \beta^{1-FG_t^c(i)}(1-\beta)^{FG_t^c(i)} \tag{7}$$

In this likelihood function we only take the pixels of the projection $\Omega_n^{c,m}$ for the current hypothesis $\widehat{\mathbf{x}}_n^{c,m}$ into account and exclude any pixel which is part of the occlusion map $OM_t^{c,m}$. In our experiments a value of $\alpha = 0.95$ and $\beta = 0.9$ produced the best results.

In summary, by using Equation (7) it is possible to find an approximation of the posterior $p(\mathbf{x}_t^{c,m}|I_{1:t}^c)$ by combining the likelihood of all hypotheses $\widehat{\mathbf{x}}_n^{c,m} \in W_t^m$ for the local state $\mathbf{x}_t^{c,m}$.

3.2.4. Kalman Filtering Approach. We perform local Kalman filtering [Kalman 1960], because we want to keep the local estimates $\widehat{\mathbf{x}}_t^{c,m}$ of each person in camera c on a frame-by-frame basis. This has the advantage that a smart camera can track people for a certain number of frames independently, even in cases of occlusion with a certain probability of failure. Furthermore, the local Kalman filtering operates on the frame rate of the smart camera itself, which provides information for every frame. This is an advantage compared with hypothesis testing that depends on the feedback frequency from the fusion center. The feedback from the fusion center makes sure that possible failures in the local Kalman filtering are corrected and we incorporate the feedback as another input into the local Kalman filtering. The decentralized system architecture could take advantage of this design and further reduce the communication load. For instance, a smart camera could be only corrected when it is really wrong which saves bandwidth and hence energy.

By using a local Kalman filter in the smart camera as a technique for filtering and prediction in linear Gaussian systems, the filter represents the posterior probability by

the moments, the mean $\widehat{\mathbf{x}}_t^{c,m}$ and the covariance $P_t^{c,m}$ of the local state vector, denoted as $\mathbf{x}_t^{c,m}$ (similar to Equation (1)). In other words, the local Kalman filter estimates the parameters of the following normal distribution

$$N_{\mathbf{x}_t^m}\left(\widehat{\mathbf{x}}_t^{c,m}, P_t^{c,m}\right).$$

A linear Kalman filter assumes that the evolution of a persons state over time is described by the following state transition equation:

$$\mathbf{x}_t^{c,m} = A_t \mathbf{x}_{t-1}^{c,m} + B_t \mathbf{u}_t + \epsilon_t \tag{8}$$

in which ϵ_t is a multivariate Gaussian random variable. Here, $\mathbf{x}_t^{c,m}$ and $\mathbf{x}_{t-1}^{c,m}$ are state vectors, and $\mathbf{u}_t$ is the control vector at time t. Since we do not have control data in our system, we can omit the term $B_t\mathbf{u}_t$. A_t is thereby the state-transition matrix and ϵ_t the process noise. We use the constant velocity model in the state-transition modeling A_t, which is defined as

$$A_t = \begin{bmatrix} F & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & F & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & diag\,(1,1) \end{bmatrix}, F = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \tag{9}$$

The vector ϵ_t is modeled as a multivariate Gaussian random variable with zero mean and the covariance Q_t. The covariance Q_t can be expressed as

$$Q_t = \begin{bmatrix} D(\sigma_{\dot{x}}) & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & D(\sigma_{\dot{y}}) & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & diag\left(\sigma_w^2, \sigma_h^2\right) \end{bmatrix}, D(\sigma) = \begin{bmatrix} \frac{\sigma^2}{3}dt^3 & \frac{\sigma^2}{2}dt^2 \\ \frac{\sigma^2}{2}dt^2 & \sigma^2 dt \end{bmatrix} \tag{10}$$

where $(\sigma_{\dot{x}}^2, \sigma_{\dot{y}}^2)$ are variances for velocity noise, and (σ_w^2, σ_h^2) the variances of the noise for width and height of a person. dt refers to the time difference between two time instances.

The Kalman theory assumes that states cannot be directly observed. Available inputs $\mathbf{z}_t^{c,m}$ that are a linear function of the unknown local state $\mathbf{x}_t^{c,m}$ are incorporated into the local Kalman filter as follows

$$\mathbf{z}_t^{c,m} = C_t \mathbf{x}_t^{c,m} + \delta_t. \tag{11}$$

Here, C_t corresponds to the measurement update matrix. The distribution of δ_t is a multivariate Gaussian with zero mean and covariance R_t.

In our case, $\mathbf{z}_t^{c,m} = (\mathbf{z}_t^F, \mathbf{z}_t^H, h_t)^T$, where $\mathbf{z}_t^F = \mu_{t-1}^m$ describes the global state estimates fed back from the fusion center from time $t-1$, $\mathbf{z}_t^H = \widehat{\mathbf{x}}_n^{c,m}$ is the best hypothesis within an area W_t^m, estimated in the same way as in Section 3.2.2 with the difference that, instead of the feedback from $t-1$, the predicted local state of the local Kalman filter is used (Equation (5)).

The height h_t of a person m with respect to a world coordinate system is estimated by comparing the foreground evidence given the local state $\mathbf{x}_t^{c,m}$ to two different height models, namely the *sitting* and *standing* model. Both models have two different heights assigned to each other which can be set ad-hoc or learned over time by modeling the height of each person individually.

The final result of the local Kalman filter is the mean $\widehat{\mathbf{x}}_t^{c,m}$ and the covariance $P_t^{c,m}$ which together describe the best possible estimate over the local state $\mathbf{x}_t^{c,m}$ by a normal distribution. Those two parameters are sent as an estimate for every person m to the fusion center.

3.3. Consensus Tracking Approach

The goal is to fuse the likelihood distributions $p(\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m}|\mathbf{x}_t^m)$ of each camera c to a final decision $p(\mathbf{x}_t^m|\mathbf{Z}_{1:t})$ for each person m (Equation (3)). To do so, the consensus tracking uses a *global Bayesian estimator* per person (Figure 1). In any case, the local estimates of each person m in the camera-based tracking for each smart camera c are approximated by Gaussian distribution $N_{\mathbf{x}_t^m}(\widehat{\mathbf{x}}_t^{c,m}, P_t^{c,m})$.

To estimate the global state of each person m, we use a Bayesian filter that calculates the posterior probability based on a motion model and acquired local estimates $\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m}$ from every smart cameras. The Bayes filter algorithm splits Equation (3) up into a *prediction* and a *correction* step. The prediction step is defined as follows

$$p\left(\mathbf{x}_t^m|\mathbf{Z}_{1:t-1}\right) = \int p\left(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m\right) p\left(\mathbf{x}_{t-1}^m|\mathbf{Z}_{1:t-1}\right) d\mathbf{x}_{t-1}^m. \tag{12}$$

Here, the state transition probability $p(\mathbf{x}_t^m|\mathbf{x}_{t-1}^m)$ is the motion model for person m and the $p(\mathbf{x}_t^m|\mathbf{Z}_{1:t-1})$ is the predicted posterior probability based on a motion model. The correction step takes the local estimates $\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m}$ from each smart cameras into account and is defined as

$$p\left(\mathbf{x}_t^m|\mathbf{Z}_{1:t}\right) = \eta p\left(\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m}|\mathbf{x}_t^m\right) p\left(\mathbf{x}_t^m|\mathbf{Z}_{1:t-1}\right), \tag{13}$$

where $\eta = p(\mathbf{I}_t|\mathbf{I}_{1:t-1})^{-1}$. The likelihood distribution $p(\mathbf{I}_t|\mathbf{x}_t^m)$ is the fusion of all local estimates $\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m}$ from every smart camera.

As a particular implementation of the Bayes filter in our proposed tracker, we use a linear Kalman filter (already introduced in Section 3.2.4), denoted as *global Kalman filter*, with mean μ_t^m (the most likely state of person m) and covariance matrix K_t^m.

The prediction step (Equation (8)) of the global Kalman filter is a constant velocity model describing the motion of a person (see Section 3.2.4).

In the correction step (Equation (11)) the local estimates $\mathbf{z}_t^m = (\widehat{\mathbf{x}}_t^{1,m}, \ldots, \widehat{\mathbf{x}}_t^{C,m})^T$ of each smart camera are incorporated, which are obtained from the camera-based tracking by one of the two approaches (Section 3.2.2 or Section 3.2.4).

The measurement update matrix C_t and the measurement noise R_t are described as follows:

$$C_t = \begin{bmatrix} C_t^{1,m} \\ \vdots \\ C_t^{C,m} \end{bmatrix}, R_t = \begin{bmatrix} P_t^{1,m} & \cdots & 0_{6\times6} \\ \vdots & \ddots & \vdots \\ 0_{6\times6} & \cdots & P_t^{C,m} \end{bmatrix}. \tag{14}$$

Here, the measurement update matrix $C_t^{c,m}$ for each person m and a specific camera c are given by an identity matrix and the covariance matrices $P_t^{c,m}$ are the covariances of the local estimates from the smart cameras.

Equation (14) states that all local estimates for each person m from every camera are taken into account. Note that it is possible that a camera does not have any information about a specific person due to the fact that the person is not seen by this camera or is occluded by other persons or furniture. In this case, the measurement update matrix $C_t^{c,m}$ of this camera c is set to a zero matrix, so that this local estimate is not taken into account for the joint decision. It is also worth mentioning that we use the covariance matrix of each person m of a camera c to model the measurement noise. Since the covariance matrix is the best model for the uncertainties of the state, this matrix includes a qualitative measure of the reliability of each state variable.

Finally, the global estimates $\mathbf{x}_t^m$ of each person m are fed back to every camera to correct for possible mistakes. This feedback is essential since the tracking of each individual camera does not need to be perfect. This is even not possible, since there are

situation where a camera cannot contribute or gather any information. For example, if a person is completely occluded, this camera does not contribute any information, but also cannot estimate any further local state of this person. In this particular case the only way for this camera to keep track of this person is by using the feedback of the fusion center, which is based on the input of all cameras. That's why the feedback is very important for the overall performance of the system.

4. RESULTS

In order to evaluate our approach, we conducted several experiments using the video data we collected for two different scenarios: an indoor scenario and a meeting scenario. Those scenarios fall into the domain of surveillance and behavior analysis of people in meetings. Here, the overall performance of our proposed system architecture is evaluated in terms of accuracy up to a certain degree (up to twice the width of a person) and the precision (in terms of number of object losses).

In the *indoor* scenario people were observed while walking in a room without furniture. In the *meeting* scenario the room was equipped with furniture (tables and chairs) and people were observed while having a meeting: entering the room, shaking hands with each other, walking around the table to find a place to sit, sitting, moving chairs to sit at another position, standing (to give a presentation), and leaving the room. All videos were recorded using a six-camera setup, consisting of four side-view and two top-view cameras, operating on a frame rate of 20 FPS. The cameras were mounted at ceiling height ($3m$ approximately), and extrinsically calibrated and synchronized up to frame accuracy.

Our proposed framework was implemented in C++, in a client-server fashion. In the experiments we performed, each camera was connected to a PC (a client), with a single-core 2.8 GHz processor to simulate a "smart camera." All smart cameras were connected to another single PC, with a single-core 2.8 GHz processor which functions as the fusion center. The purpose of the experiments was to test several important attributes of the framework: the *calibration accuracy*, the *real-time performance* and *scalability* and the *performance of the proposed system architecture*, in terms of accuracy and precision. Furthermore, we compared our results with the *state-of-the-art method* of [Berclaz et al. 2011]. In this section we present experimental results for each of these attributes separately.

4.1. Calibration Accuracy

We calibrated the cameras using the calibration method of Bouguet [1999] for the side cameras, and the method of Kannala and Brandt [2006] for the top cameras. To do so, we used a checkerboard pattern for intrinsic calibration and manually measured reference points in the scene for extrinsic calibration. We used the above mentioned methods to obtain the intrinsic and extrinsic calibration of each camera. As shown in Figure 5(a), the calibration of each camera is very accurate with an average error per camera below one pixel. To measure the overall accuracy of all cameras, we used the obtained calibration results to compute the 3D coordinate of each reference point. Furthermore, we compared the results to our manual-measured reference points and obtained an overall accuracy of 0.56 cm (Figure 5(b)), which is the mean of the square-root distance between the obtained and measured reference points.

The results show that our calibration procedure is quite precise and that 3D coordinates of objects in the scene can be obtained from 2D image points with sufficient accuracy.

4.2. Real-Time Performance and Scalability

We tested two important aspects: the real-time performance of the whole system which is limited by the tracking time on the camera side and the scalability which is limited

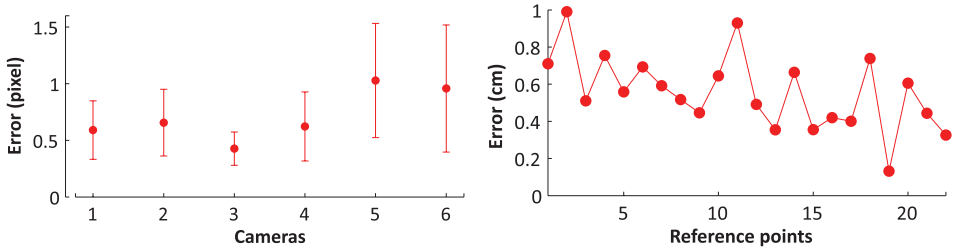

Fig. 5. *Calibration accuracy*. In (a) calibration error of each camera is very small and therefore the calibration of an individual camera very accurate. To evaluate the overall accuracy of the calibration, we compared estimated 3D coordinates to manually measure reference points. In (b), the error to each reference point is shown with a average accuracy of 0.56 cm.

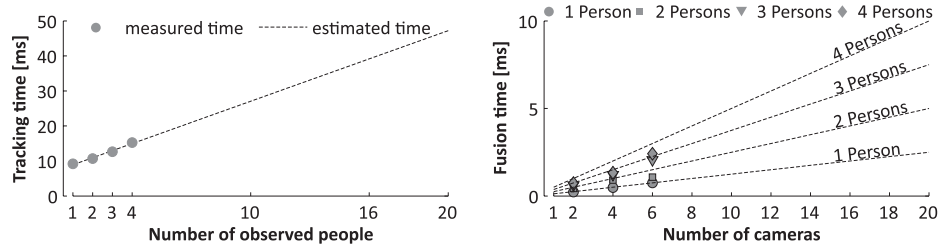

Fig. 6. *Real-time performance and scalability*. In (a), the dots show the measured data and the dashed line the estimated timing depending on the number of people (up to 20 people). Due to the nature of the local linear Kalman filter at the camera side the computation time increases linearly with the number of people. In (b), the points show the measured data and the dashed line the estimated timing up to 20 cameras for up to four persons. The estimated lines assume that a person is always seen by all cameras which is, in practice, not the case. That's why there is a difference between the measured points and the estimated lines.

by the tracking time on the fusion center. We conducted all experiments with the Kalman filter approach at the camera side (see Section 3.2.4) since the hypothesis testing (Section 3.2.2) is also part of the Kalman filter approach.

To test the real-time performance we measured the tracking execution time on each camera for a different number of observed people (one to four people).

To get an average execution time for one camera, we used all test sequences and averaged execution times over the number of sequences and all cameras. The results are shown in Figure 6(a).

We see that tracking time on the camera is less than 15 ms per frame which is faster than real time (40 ms - 25 FPS). The camera-based tracking time depends linearly on the number of viewed people. This is due to a local Kalman filter on the camera side assigned to each person visible in the camera view. This means that an increase in the number of viewed people increases just the number of used filters, like the global Kalman filters in the fusion center. Accordingly, the line in Figure 6(a) represents the camera tracking time estimated for more than four people. We see that up to 16 people can be tracked on a camera side at 25 FPS. Such a performance is suitable for most applications since only in very crowded environments one camera will have more than 16 people in the field of view.

Furthermore, if we relate this result to the scalability of the proposed system, we see that the description from 20 cameras can be fused at 25 FPS and fed back to the cameras. This refers to the slowest case in which each camera tracks 16 people.

Therefore, even in very crowded environments, the use of more cameras that observe less people per camera (e.g., with a narrower field of view) can be a solution to cope with the real-time limitations of the camera-based tracking, still keeping the real-time performance of the whole system.

To test the scalability, we varied both the number of tracked people and the number of cameras connected to the fusion center. The obtained results are shown in Figure 6(b). Since each person is usually not visible in all cameras connected to the fusion center, the fusion time is shorter than the one given by a linear function, because some cameras do not contribute with a measurement for each person. Therefore, the lines in the graph of Figure 6(b) represent the maximal fusion time as a function of the number of cameras connected to the fusion center and the number of tracked people.

We see that it is possible to fuse information from many cameras in realtime at 25 FPS (e.g., from 21 cameras for 15 tracked people). Also, the fusion time and the number of cameras in which the person is observed correlate in a linear fashion due to the addition of observations from n cameras which increases only the number of measurements by n for a given person whilst the dimension of the measurement vector remains the same. Such an efficiency enables highly scalable tracking systems that could deploy sufficient number of cameras for any area and any amount of people that need to be observed. Note that the estimated scalability could be further increased by using multicore processing units or by optimizing the implementation for hardware accelerated processing (e.g., GPU processing).

4.3. Performance of the Proposed System Architecture

We express the performance of our proposed tracker in two ways: as *precision*, that is, the total number of losses of tracked people (the number of object losses (NoOL)), and as *accuracy*, that is, the Euclidean distance between the ground truth positions of people and positions estimated by the tracker (the total average tracking error (TATE)). We created ground truth by manually annotating the ground positions of people each second in all video sequences. Using the camera calibration parameters we then calculated the ground truth positions of people in real world coordinates (x and y coordinates, $z = 0$). The Euclidean distance between these positions and the real world positions computed by the tracker is used to express the accuracy. For the precision, the number of people losses, we consider that people are lost by the tracker if the Euclidean distance between their estimated position and the ground truth position is bigger than 100 cm (twice the assumed width of a person).

We conducted several experiments under different circumstances: several indoor scenarios of people walking around in a room with and without equipped furniture and multiple meeting scenarios. Both scenarios include up to four people and changing environmental conditions (esp. with and without lighting changes). In total, we have collected more than 120 minutes of data. In the following section we focus on different aspects of our system.

4.3.1. Overall Performance of the Proposed Tracker. The average accuracy and precision over 120 minutes of data of our proposed tracker are 21 cm total average tracking error and 0.3 object losses per minute. The results are very promising and show the robustness of our system. The evaluation also include some very difficult cases, that is, we tested our tracking system under severe occlusions.

4.3.2. Performance for Different Number of Cameras. Firstly, we show the influence of the number of cameras on the proposed tracker. For this experiment we used an indoor sequence in which four people are walking around. In Figure 7 the comparison between different number of cameras is shown. For this indoor sequence we compare four side cameras, two top-view cameras, and the complete setup, consisting of six cameras.

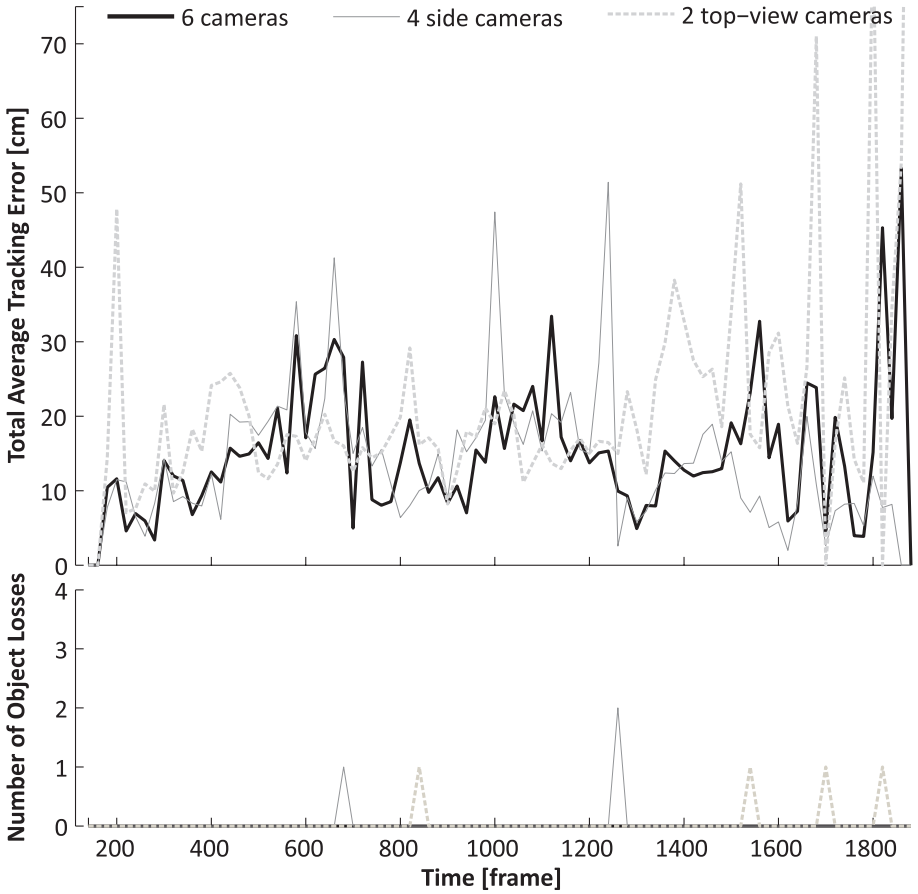

Fig. 7. *Performance for different number of cameras*. The comparison was done between four side cameras (light gray line), two top-view cameras (dashed line) and the whole setup based on top and side views (black line). The best results are achieved by using top and side cameras together. Especially the use of only top-view cameras alone is not suitable enough to maintain accurate tracking of people.

The results show that using only top-view cameras is not enough for accurate people tracking. This is mainly because it is difficult to accurately locate people's ground position in top-views without additional detection of body parts. These inaccuracies lead to losses of the tracked people. On the other hand, side views give a better estimation of people's ground position, but their observations are more prone to occlusions, which also causes some losses of people. This is especially noticeable in other sequences where people are often occluded by tables and chairs. Therefore, the highest accuracy is achieved by combining top and side views, that is, by using all six views together.

4.3.3. Hypothesis Testing vs. Kalman Filtering. In this section, the comparison between the hypothesis testing (Section 3.2.2) and the Kalman filtering (Section 3.2.4) approach used in the camera-based tracking (Section 3.2) is discussed. Here, we used several different sequences to observed the performance for both methods. In Figure 8, the results for a meeting scenario with four attendees is exemplarily shown. We see that the accuracy of the tracker performs similar for both methods. The main difference is the number of object losses which is higher for the the hypothesis testing approach. This is caused by the fact that the Kalman filtering approach does not depend on

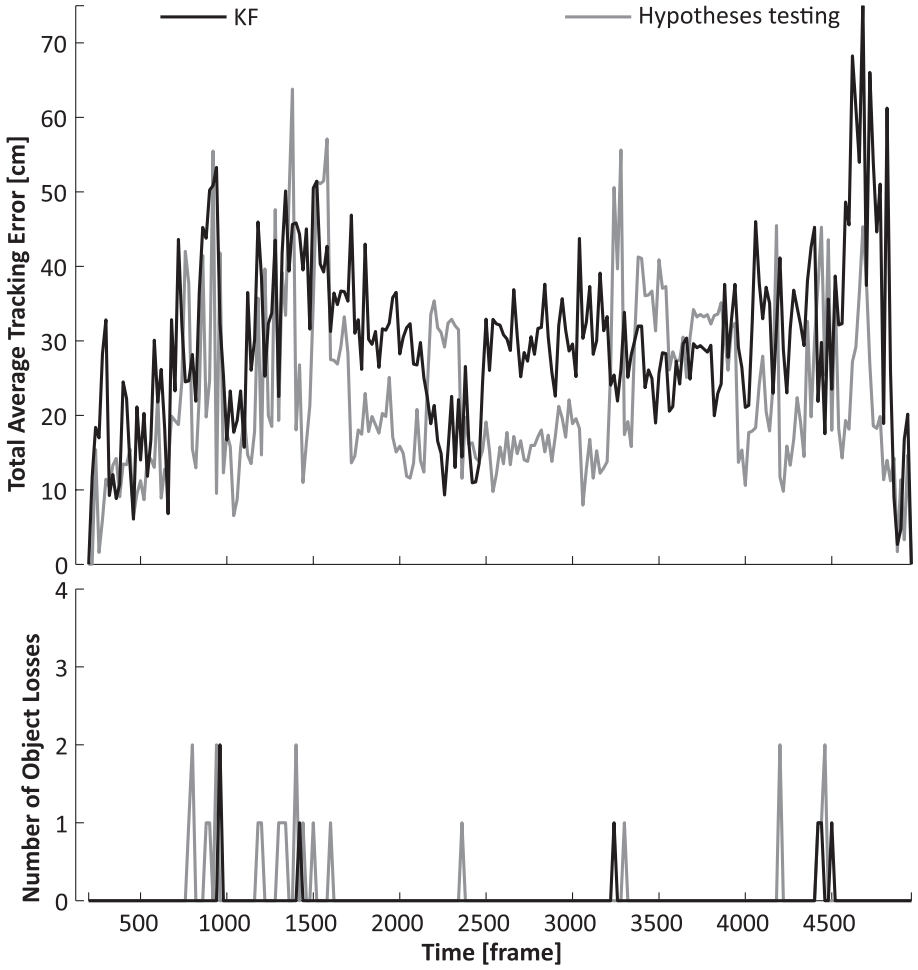

Fig. 8. *Hypothesis testing vs. Kalman filtering*. This comparison shows the performance between the hypothesis testing (Section 3.2.2) and the Kalman filtering (Section 3.2.4) approach used in the camera-based tracking (Section 3.2). It can be seen that the accuracy is nearly the same for both methods. But the hypothesis testing approach produces some object losses. This is caused by the fact that the Kalman filtering approach does not depend on the feedback frequency and operates on the camera frame rate.

the feedback frequency of the fusion center and operates on the camera frame rate. Therefore, it has more information available than the hypothesis testing approach, which depends on the feedback of the fusion center.

In general, it is an advantage to keep a local estimate of each person on the camera side. The Kalman filtering approach is only an example of possible filtering approaches. It shows very promising results and can be further explored. Other approaches, especially non-parametric filter approach like particle filters, can be taken into account to improve the robustness of the system.

4.3.4. Influence of Feedback and Feedback Frequency. Feedback is essential in the proposed system architecture. To demonstrate the use of feedback we performed two experiments. At first, we tested the influence between feedback in the system and no feedback at all (Figure 9). Secondly, we evaluated the influence of different feedback

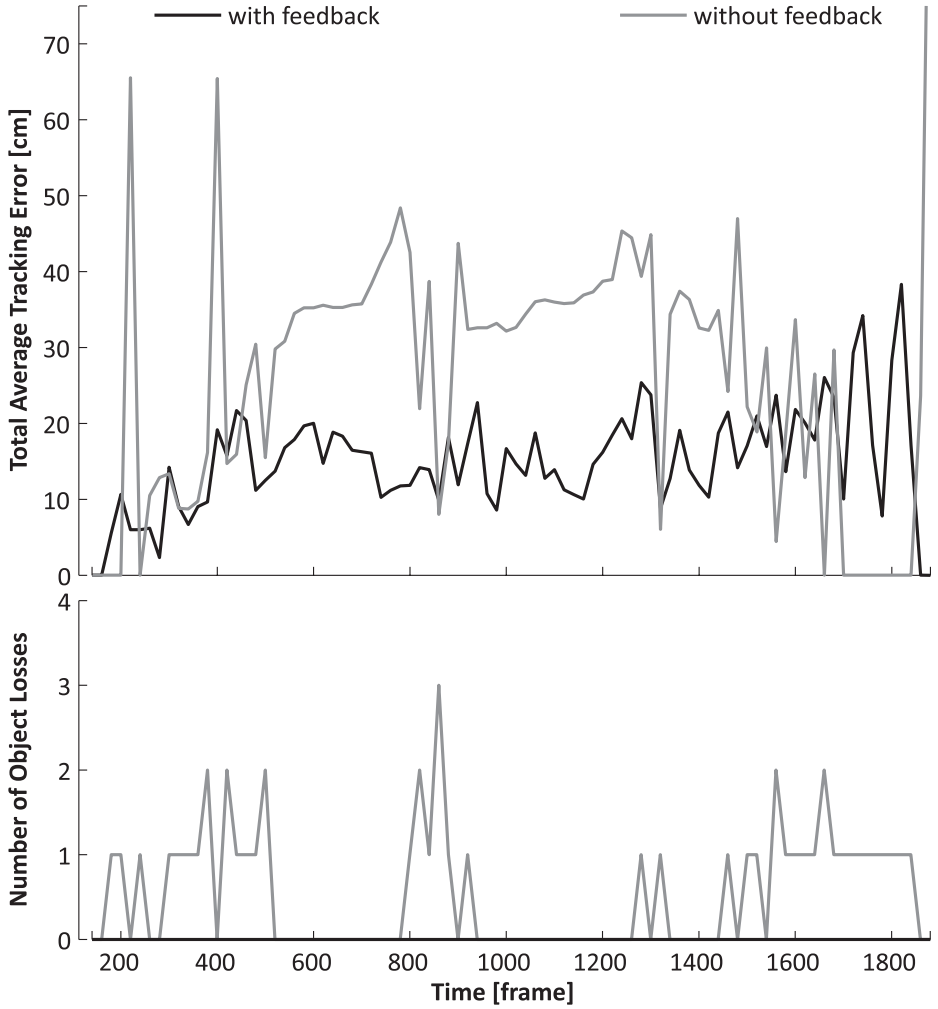

Fig. 9. *Influence of feedback*. To demonstrate the use of feedback in the whole system architecture we show the difference between feedback and no feedback. The results clearly show that accuracy and precision is much better with the use of feedback in the system.

frequencies between the cameras and the fusion center (Figure 9). For both experiments we used an indoor scenario with up to four people walking around.

In Figure 9 the results demonstrate clearly the use of feedback. There is a big difference in accuracy and precision. This is mainly because a single camera cannot recover from its own mistakes and cannot even know that it made one. In the case of mistakes, the camera keeps sending wrong information to the fusion center which are still taken into account and lead to wrong estimates at the fusion center. The system starts to fail and cannot recover from this. This indicates that feedback is essential for a robust system. Possible improvements could be: the fusion center could find out if a camera is wrong and send only feedback to the cameras which are starting to fail. This could further improve the communication load and therefore be more energy efficient.

In Table I the influence of different feedback frequencies is shown. We evaluated the influence for both methods, the hypothesis testing and the Kalman filtering approach.

Table I. Comparison between Different Feedback Frequencies

	Kalman filter approach		Hypothesis testing	
Frequency (ms)	TATE (cm)	NoOL	TATE (cm)	NoOL
50	15.0	0	25.3	2
100	15.0	0	25.5	2
200	16.5	1	26.1	3
500	17.8	3	23.6	6
1000	18.9	4	23.0	16

We evaluated the influence of feedback for both methods, the hypothesis testing and the Kalman filtering approach. The results indicate that even with a feedback of 200ms the accuracy and precision is acceptable for most applications.

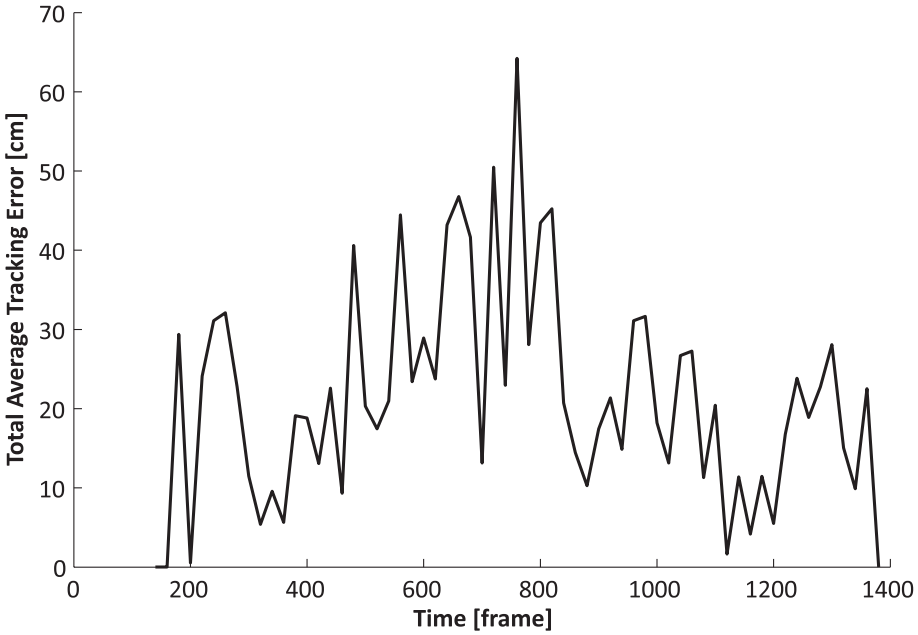

Fig. 10. *Influence of changing light conditions*. To demonstrate the robustness against lighting changes for the whole system architecture we conducted an experiment with sudden and continuous light changes as well as changes only in parts of the scene. The results are promising and show the robustness of our system. Note that we do not have any object loss for this sequence.

The results show that even with a feedback of 200ms the Kalman filtering approach produced very good results. As already mentioned, the feedback mechanism could be further explored to reduce the communication load and save bandwidth and therefore energy. In general, the Kalman filtering approach produced better results than the hypothesis testing.

4.3.5. Challenging Test Cases. At first, we tested our proposed tracker on a sequence containing changing light conditions. We tried several setups and evaluated one of them, in which the light changes in different ways: sudden and continuous light changes as well as changes only in parts of the scene. In Figure 10 we show the results of our experiment. In this particular sequence we do not have any object losses. The total average tracking error is around 21.3 cm. The results are really impressive since some cameras do not see anything for several seconds (Figure 11). In those cases the camera-based tracking does not send any information to the fusion center and the

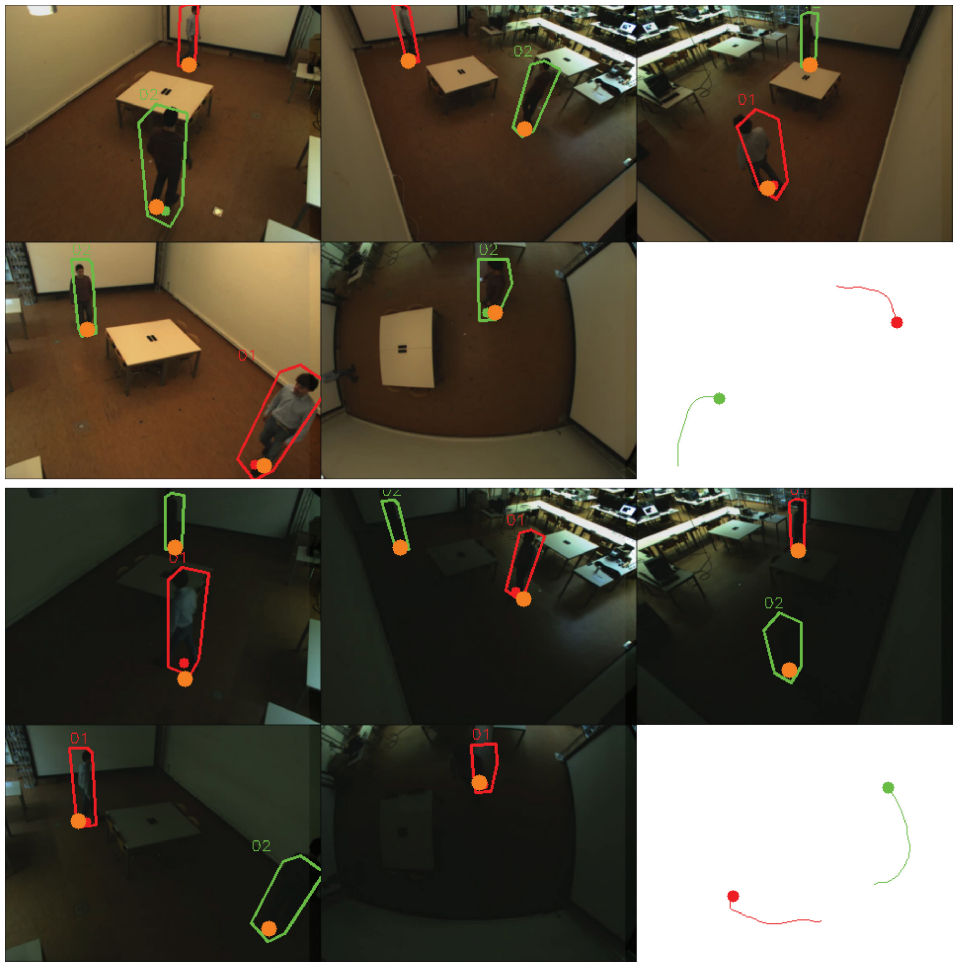

Fig. 11. *Influence of changing light conditions*. Two sample frames (frame 400 and 620) from a sequence containing changing light conditions are shown. Note that parts of the scene are really dark, but our tracker is able to keep track of the people in the scene.

fused estimates of each person are only calculated from the cameras which contribute information to the whole system.

In summary, we are able to track people even under challenging light conditions like sudden and/or continuous light changes. The results benefit from the robust FG/BG segmentation, especially against lighting changes (Section 3.2.1). Improvements can be made by including a light map which models the light intensities to improve the foreground masks of the FG/BG segmentation.

In our second type of experiments we tested our method under severe occlusions for over 5 min of data. The experiment includes four people walking around in an area of 5 m by 8.8 m. This sequence contains tables and chairs resulting in partly occluded people. The results are shown in Figure 12. The accuracy for this five-minute sequence is 29.3 cm which is still below the width of a person and therefore accurate enough to localize a person. Of course, sometimes the TATE is above 50 cm (approx. the width of a person) which is mainly due to switches of identities. As show in Figure 12 we sometimes have object losses. This is expected. If people are too close to each other,

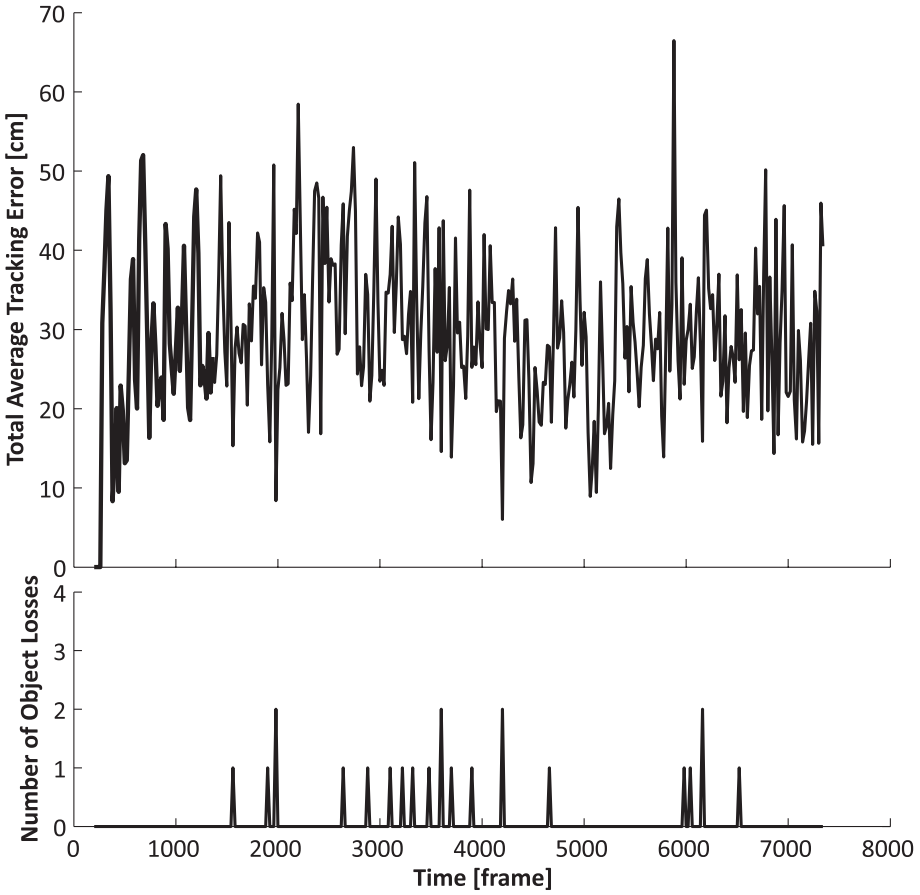

Fig. 12. *Influence of severe occlusions*: We tested our tracker under severe occlusions for over 5 min. The average accuracy for this five-minute sequence is 29.3 cm which is still below the width of a person and therefore accurate enough to localize a person. There are some object losses which are due to the proximity of people which results in switching of identities.

object switching might happen due to the lack of an appearance model for a person. Appearance modeling of people remains challenging since their appearance can change rapidly in a camera view. To model the appearance taking the aforementioned problems into account is one of our future research goals.

4.4. Comparison with a State-of-the-Art Method

We compared our proposed tracker with the state-of-the-art multicamera tracking approach of Berclaz et al. [2011]. At first, we made a comparison with two of our sequences, an indoor sequence and a meeting sequence with up to four people. The results we obtained are similar to the state-of-the-art tracker of Berclaz et al. [2011]. The tracker of Berclaz et al. was configured with a grid cell size of 10 by 10 cm and with the results of our FG/BG segmentation method as input. As differently described in their paper, however, their tracker did not use any color information to perform tracking. We tested different grid cell sizes (10, 20 and 30 cm), but the grid cell size of 10 by 10 cm achieved the best results.

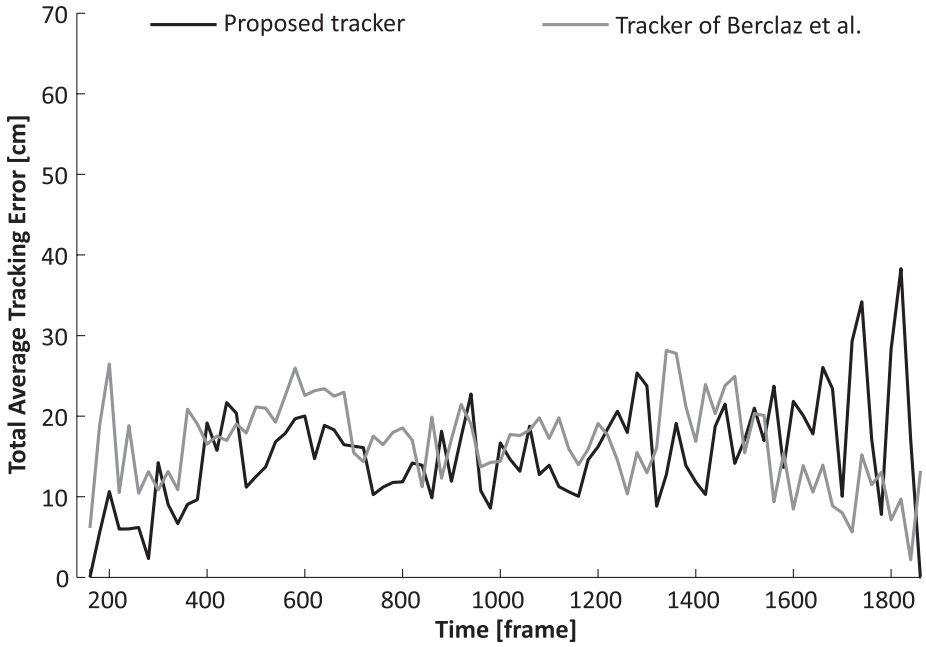

Fig. 13. *Comparison of an indoor sequence*. We compare our proposed tracker with the state-of-the-art tracker of [Berclaz et al. 2011]. The total average tracking error for both trackers is similar (proposed tracker: 15 cm, tracker of Berclaz et al.: 16.6 cm). There are no object losses for this sequence.

4.4.1. Comparison on our Test Datasets. First, we show the performance of an indoor sequence with up to four people (Figure 13) where it can be seen that the performance of both trackers is similar. With our proposed tracker we achieve an accuracy of 15 cm and with the tracker of Berclaz et al. 16.6cm. There are no object losses in this sequence.

In our second experiment, we chose a more challenging meeting sequence for comparison. Our tracking results for this meeting sequence are already shown in Figure 8. The tracker of Berclaz et al. performed poorly on this sequence, so we are not providing a numerical comparison. This is mainly due to the fact that, after a person sits down, the FG/BG results vanishes as their tracker is not suited for these circumstances. In such situations their tracker calculates the shortest path to exit the room which would lead to huge accuracy differences and tracking losses.

4.4.2. Comparison with Public Datasets. To compare our proposed tracker with a public dataset, we chose a dataset provided by the CVLAB at EPFL [Berclaz et al. 2011]. They used three cameras with a small overlapping area in an outdoor scenario. Up to five people can be seen in this sequence. We conducted ground truth at a one-second interval for one of their sequences and chose it for comparison. The results are depicted in Figure 14. The overall accuracy of our tracker for this sequence is 36.4 cm which is still very good, considering the use of only three cameras (Figure 15). Berclaz et al. [2011] did not provide numerical results on this public dataset. Therefore, we run their method on the dataset to obtain numerical results for comparison. The tracker of Berclaz et al. [2011] shows an overall accuracy of 25.4cm on this sequence.

In summary, the tracking approaches perform similar and the results of both trackers are very good in terms of accuracy and precision. The tracker of Berclaz et al. [2011] is slightly more accurate on this dataset, although the number of losses is the same. It is worth mentioning that the implementation of the tracker of Berclaz et al. optimizes

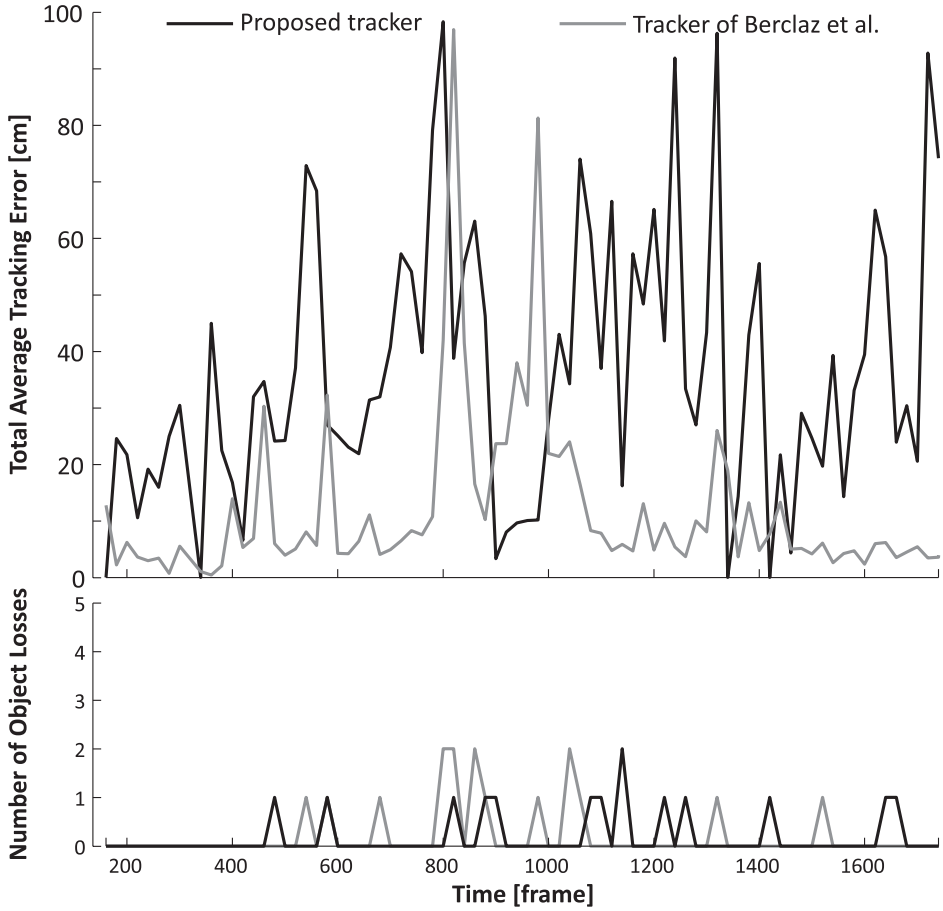

Fig. 14. *Performance on an outdoor dataset [Berclaz et al. 2011].* For a comparison with a public dataset, we chose an outdoor dataset provided by the lab of CVLAB at EPFL [Berclaz et al. 2011]. It is worth mentioning that the dataset uses only three cameras which do not have a huge overlapping area. The results of both methods are still very good despite having some object losses which are due to lack of a appearance model and the poor coverage of the scene.

trajectories over the whole sequence and can therefore cope better with object losses, however, it cannot work online. The method described in Berclaz et al. [2011] uses images of up to five seconds in the future and hence an on-line version of this method would incur a delay of about five seconds. Our proposed tracker works on a frame-by-frame basis and calculates the best estimates of the tracked people instantaneously.

5. CONCLUSIONS

In this article, we presented a novel decentralized system architecture for real-time video processing applications to track humans within a network of cameras. This is achieved by distributing tasks between cameras and a fusion center to obtain a common fusion result from multiple views. On the fusion center, only high-level compact information, sent by each individual smart camera, are taken into account. Hereby, each smart camera estimates the likelihood by a camera-based tracking method. We evaluated our proposed system architecture on multiple sequences, even in cases of severe occlusion and with/without furniture. Experimental results show a reasonable

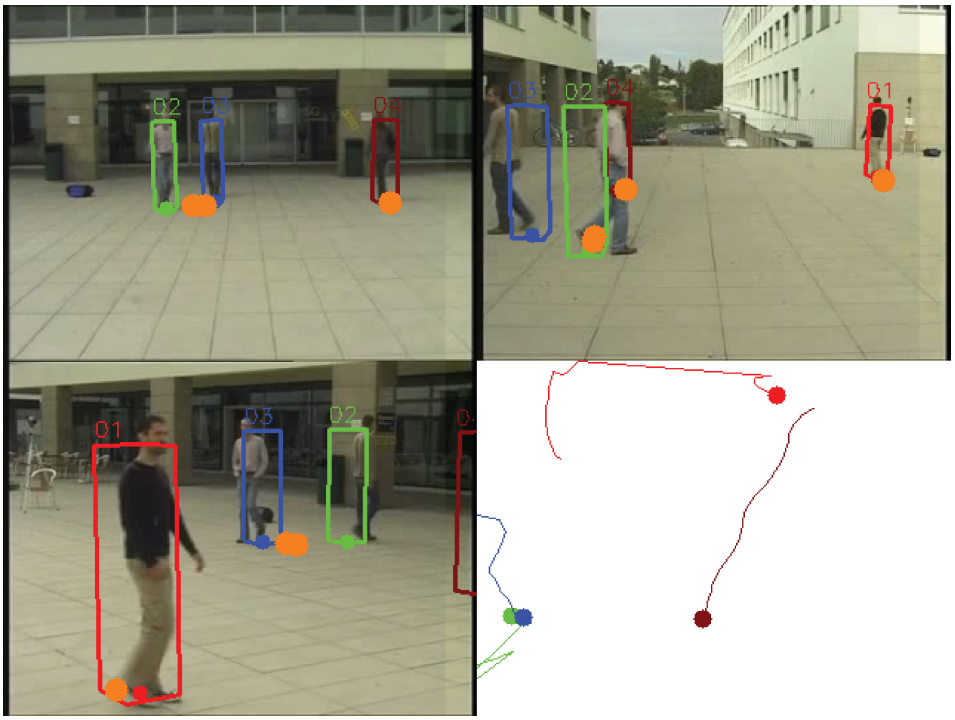

Fig. 15. *Performance on an outdoor dataset [Berclaz et al. 2011].* The sample frame (frame 550) from the public dataset provided by the lab of CVLAB at EPFL [Berclaz et al. 2011] is shown. The overall accuracy of our tracker for this sequence is 36.4 cm and is still very good, considering the use of only three cameras.

accuracy and precision, sufficient for many applications such as surveillance or behavior analysis of people in meetings, even in cases of occlusions.

There are many possible extensions to this work. One possibility is to incorporate more advanced methods to model the appearance of a person. Another extension could be a more detailed study of the feedback loop.

REFERENCES

H. K. Aghajan and A. Cavallaro. 2009. *Multi-Camera Networks: Principles and Applications*. Academic Press.

N. Anjum and A. Cavallaro. 2009. Trajectory association and fusion across partially overlapping cameras. In *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance*. 201–206.

B. Babenko, M.-H. Yang, and S. Belongie. 2011. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 1619–1632.

Y. Bar-Shalom. 1987. *Tracking and Data Association*. Academic Press.

O. Barnich and M. Van Droogenbroeck. 2009. ViBe: A powerful random technique to estimate the background in video sequences. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 945–948.

J. Berclaz, F. Fleuret, and P. Fua. 2006. Robust people tracking with global trajectory optimization. In *Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 1. 744–750.

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

Jean-Yves Bouguet. 1999. Visual methods for three-dimensional modeling. Ph.D. dissertation. California Institute of Technology, Pasadena, CA.

Q. Cai and J. K. Aggarwal. 1998. Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. In *Proceedings of the 6th IEEE European Conference on Computer Vision*. 356–362.

Ting-Hsun Chang and Shaogang Gong. 2001. Tracking multiple people with a multi-camera system. In *Proceedings of the IEEE Workshop on Multi-Object Tracking*. 19–26.

R. T. Collins. 2003. Mean-shift blob tracking through scale space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2. 234–240.

T. Darrell, D. Demirdjian, N. Checka, and P. Felzenszwalb. 2001. Plan-view trajectory estimation with dense stereo background models. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, Vol. 2. 628–635.

F. Deboeverie, P. Veelaert, and W. Philips. 2011. Face analysis using curve edge maps. In *Proceedings of the 16th International Conference on Image Analysis and Processing*. Lecture Notes in Computer Science, vol. 6979, 109–118.

D. Delannay, N. Danhier, and C. De Vleeschouwer. 2009. Detection and recognition of sports (wo)men from multiple views. In *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras*. 1–7.

S. L. Dockstader and A. M. Tekalp. 2001. Multiple camera fusion for multi-object tracking. In *Proceedings of the IEEE Workshop on Multi-Object Tracking*. 95–102.

A. Dore, M. Soto, and C. S. Regazzoni. 2010. Bayesian tracking for video analytics. *IEEE Signal Process Mag.* 27, 5, 46–55.

Alberto Elfes. 1989. Occupancy grids: A probabilistic framework for robot perception and navigation. Ph.D. dissertation. Carnegie Mellon University, Pittsburgh, PA.

A. Fathi, X. Ren, and J. M. Rehg. 2011. Learning to recognize objects in egocentric activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3281–3288.

F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. 2008. Multicamera people tracking with a probabilistic occupancy map. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 267–282.

J.-S. Franco and E. Boyer. 2005. Fusion of multiview silhouette cues using a space occupancy grid. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, Vol. 2. 1747–1753.

J. Giebel, D. Gavrila, and C. Schnörr. 2004. A bayesian framework for multi-cue 3d object tracking. In *Proceedings of the 8th IEEE European Conference on Computer Vision*. Lecture Notes in Computer Science, vol. 3024, 241–252.

S. Gruenwedel, P. Van Hese, and W. Philips. 2011. An edge-based approach for robust foreground detection. In *Advances Concepts for Intelligent Vision Systems*, Lecture Notes in Computer Science, vol. 6915, 554–565.

S. Grünwedel, V. Jelaca, J. Niño Castañeda, P. Van Hese, D. Van Cauwelaert, P. Veelaert, and W. Philips. 2012. Decentralized tracking of humans using a camera network. *Proc. SPIE*, vol. 8301, 9.

S. Hengstler and H. Aghajan. 2006. A smart camera mote architecture for distributed intelligent surveillance. In *Proceedings of the ASME Dynamic Systems and Control Conference*.

S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan. 2007. MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In *Proceedings of the ACM/IEEE Conference on Information Processing in Sensor Networks*. 360–369.

C. Hue, J. P. Le Cadre, and P. Perez. 2002. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Trans. Signal Process.* 50, 2, 309–325.

M. Isard and J. MacCormick. 2001. BraMBLe: A Bayesian multiple-blob tracker. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, Vol. 2. 34–41.

R. E. Kalman. 1960. A new approach to linear filtering and prediction problems. *J. Basic Engin.* 82, 1, 35–45.

J. Kannala and S. S. Brandt. 2006. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 8, 1335–1340.

S. M. Khan and M. Shah. 2006. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceedings of the 9th IEEE European Conference on Computer Vision.* Lecture Notes in Computer Science, vol. 3954, 133–146.

S. M. Khan and M. Shah. 2009. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 505–519.

T. Kirubarajan and Y. Bar-Shalom. 2004. Probabilistic data association techniques for target tracking in clutter. *Proc. IEEE* 92, 3, 536–557.

B. Kröse, T. Oosterhout, and T. Kasteren. 2011. Activity monitoring systems in health care. In *Computer Analysis of Human Behavior*, Springer, 325–346.

J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. 2000. Multi-camera multi-person tracking for easy living. In *Proceedings of the IEEE Workshop on Visual Surveillance*. 3–10.

J. Liu, M. Chu, and J. E. Reich. 2007. Multitarget tracking in distributed sensor networks. *IEEE Signal Process. Mag.* 24, 3, 36–46.

E. Maggio, M. Taj, and A. Cavallaro. 2008. Efficient multitarget visual tracking using random finite sets. *IEEE Trans. Circuits Syst. Video Technol.* 18, 8, 1016–1027.

A. Mittal and L. S. Davis. 2003. M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Int. J. Comput. Vision* 51, 3, 189–203.

A. Nakazawa, H. Kato, and S. Inokuchi. 1998. Human tracking using distributed vision systems. In *Proceedings of the 14th International Conference on Pattern Recognition*. Vol. 1, 593–596.

P. Nillius, J. Sullivan, and S. Carlsson. 2006. Multi-target tracking-linking identities using bayesian network inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2. 2187–2194.

O. Ozturk, T. Yamasaki, and K. Aizawa. 2009. Tracking of humans and estimation of body/head orientation from top-view single camera for visual focus of attention analysis. In *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops*. 1020–1027.

R. Pflugfelder and H. Bischof. 2010. Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 4, 709–721.

C. Rasmussen and G. D. Hager. 2001. Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 6, 560–576.

D. Reid. 1979. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control* 24, 6, 843–854.

D. Smith and S. Singh. 2006. Approaches to multisensor data fusion in target tracking: A survey. *IEEE Trans. Knowl. Data Eng.* 18, 12, 1696–1710.

K. Smith, D. Gatica-Perez, and J. M. Odobez. 2005. Using particles to track varying numbers of interacting people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1, 962–969.

M. Taj and A. Cavallaro. 2009. Multi-camera track-before-detect. In *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras*. 1–6.

M. Taj and A. Cavallaro. 2010. Multi-view multi-object detection and tracking. *Int. J. Comput. Vision,* 263–280.

M. Taj and A. Cavallaro. 2011. Distributed and decentralized multicamera tracking. *IEEE Signal Process Mag.* 28, 3, 46–58.

S. Thrun. 2003. Learning occupancy grid maps with forward sensor models. *Autonomous Robots* 15, 2, 111–127.

S. Thrun, W. Burgard, and D. Fox. 2005. *Probabilistic Robotics*. MIT Press.

A. Yilmaz, O. Javed, and M. Shah. 2006. Object tracking: A survey. *ACM Comput. Surv.* 38, 4.

Z. Zivkovic. 2004. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the IEEE 17th International Conference on Pattern Recognition*. Vol. 2, 28–31.

Received April 2012; revised September 2012; accepted January 2013