Multiple People Tracking in Smart Camera Networks by Greedy Joint-Likelihood Maximization

Nyan Bo Bo, Francis Deboeverie, Peter Veelaert and Wilfried Philips
Ghent University/iMinds, TELIN, Image Processing and Interpretation
Sint-Pietersnieuwstraat 41, Ghent 9000, Belgium

Keywords: Multi-camera Tracking, Decentralized System, Joint-likelihood Maximization.

Abstract: This paper presents a new method to track multiple people reliably using a network of calibrated smart cameras. The task of tracking multiple persons is very difficult due to non-rigid nature of the human body, occlusions and environmental changes. Our proposed method recursively updates the positions of all persons based on the observed foreground images from all smart cameras and the previously known location of each person. The performance of our proposed method is evaluated on indoor video sequences containing person–person/object–person occlusions and sudden illumination changes. The results show that our method performs well with Multiple Object Tracking Accuracy as high as 100% and Multiple Object Tracking Precision as high as 86%. Performance comparison to a state of the art tracking system shows that our method outperforms.

1 INTRODUCTION

Nowadays, many computer vision-based applications, such as automatic surveillance, smart meeting rooms/homes and human behavior/activity analysis, require robust tracking of multiple people in indoor or outdoor environment. However, the task of reliably tacking multiple non-rigid targets such as humans is in fact very challenging because the appearance of a person easily changes by body movement, pose and orientation changes with respect to the camera view. Moreover, the appearance of the target may also be altered by changes in the environment, such as lighting, which makes the tracking even more difficult.

Since multiple people may be moving about in the scene, a person can be fully/partially occluded by another person(s) or other object(s) in the scene. This occlusion problem makes the tracking task even more challenging and is very difficult to address in single camera methods. Over the past fifteen years, many trackers for camera networks with overlapping views have been introduced. These trackers use information from different view points to handle the occlusion problem. Additionally, joint estimation of a person's position from multiple views is usually more precise than estimation from a single view point.

In this paper, we propose a new method to track multiple people in real time using a network of smart cameras. Our proposed method follows a decentralized architecture, i.e., all image processing tasks can be executed on smart cameras and only numbers are exchanged between nodes rather than images/frames.[1] Our tracker recursively estimates the current positions of all persons by maximizing the likelihood of the current observations of all cameras. The estimate is recursive because the aforementioned likelihood is determined by the previous positions.

In our method, the observations are binary images obtained by foreground/background estimation on all camera views. Moreover in our method these images remain in private memory within each smart camera and are never transmitted; this is one of the contributions of our method. In practice, observations clearly depend on the positions of all persons in the scene. Therefore in this paper observations are not tied to individual persons, but to camera views. Since the likelihood maximization takes into account the positions of all persons, our tracker does not need to perform an explicit occlusion detection and handling.

Another contribution of this paper is the real-time maximization of the likelihood of an observations from all cameras, given the positions of all persons, using a greedy search. The search space is defined by the known positions of all persons at previous time instance and the physical limitation that a person cannot move very far between two consecutive frames.

[1] In this paper, the smart cameras are simulated on a computer.

Bo, N., Deboeverie, F., Veelaert, P. and Philips, W.
Multiple People Tracking in Smart Camera Networks by Greedy Joint-Likelihood Maximization.
DOI: 10.5220/0005716206020609
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 3: VISAPP*, pages 604-611
ISBN: 978-989-758-175-5

These known positions at earlier time instance comes from a person detector or from the previous estimations of the tracker itself. We use the integral image in the algorithm's implementation to achieve high computational efficiency.

We evaluate our method on three multi-camera video sequences, each of which contains different scenarios such as people having meeting, walking a round under steady and varying lighting, captured in an indoor meeting room environment. The results confirm that our method achieves high accuracy and precision on video sequences containing frequent person–person and object–person occlusions. The performance comparison to a publicly available state of the art tracker shows that our method outperforms.

2 RELATED WORK

Over the last decade, many techniques have been proposed for robust tracking of multiple people using a single-camera as well as networks of cameras. Many trackers (Yang et al., 2009), (Henriques et al., 2011), (Yun et al., 2012) (single-camera trackers) (Bredereck et al., 2012), (Gruenwedel et al., 2014), (Bo Bo et al., 2014) (multi-camera trackers) recursively update the position of each person based on priors from previous frames and the observations in the current frame, such as appearance models based on color/texture, foreground blobs, person detector responses, and so on. These trackers are usually relying on extracting *observations of individual persons* from all the input images. Most of these methods then process the "person observations" independently, e.g. feeding them to a recursive state estimator (Kalman filter, Particle filter, ...) for that person. As such, persons are tracked independently after their observations have been separated.

Some trackers (Zhang et al., 2008), (Bredereck et al., 2012), (Andriyenko and Schindler, 2010) (Berclaz et al., 2011) adopt a tracking-by-detection strategy, i.e., people are firstly detected in multiple video frames and then detections are linked across time. These systems usually need input of the entire video or a batch of frames. Additionally, to our knowledge the computation of occupancy maps or detection of humans takes a relatively large potion of allowed computation time for real-time tracking. These factors limit the feasibility for real-time tracking. However, these methods have the advantage of having information from future video frames, while estimating the person's positions. Therefore, by exploiting information from future video frames, these methods can potentially perform better in terms of accuracy.

Recently, several methods (Andriyenko and Schindler, 2011), (Milan et al., 2013) have been introduced to improve the trajectories produced by baseline trackers (Wojek et al., 2010), (Pirsiavash et al., 2011). The approach of Andriyenko *et al.* (Andriyenko and Schindler, 2011) formulates an energy function from human detector's detections, object dynamics, collision avoidance and object persistence together with a regularization term for each trajectory produced by a baseline tracker. Then the energy is minimized by growing/shrinking, splitting/merging or adding/removing the trajectories. The work of Milan *et al.* follows a similar approach but using a different energy formulation and optimization. The reported optimization time for these approaches is 1 to 2 second per frames excluding the time for generating baseline trajectories and person detections. Therefore, they are not feasible for real-time applications.

3 PROPOSED TRACKER

Our tracking system uses a decentralized architecture, i.e., the computational load is distributed over C calibrated cameras and a greedy likelihood maximization node. Each smart camera captures and processes its video on board. Thus, only *meta-data*, such as positions and likelihoods, are exchanged between nodes rather than images. This reduces network bandwidth requirements, thus increasing scalability, i.e., additional smart cameras may be added to the system without concerning too much about the communication bottleneck. Fig. 1 shows the building blocks of the proposed tracker and the data exchange between nodes.

One analysis cycle of processing a single video frame from all cameras is as follows. At time t, each smart camera c captures image and computes a foreground image from it. In our tracker, we use the same texture-based foreground detection method as in our previous work (Bo Bo et al., 2015). The likelihood maximizer node then requests the likelihood of observing foreground image in each camera view given the hypothesized positions of all persons. All smart cameras compute the requested likelihood and send back to the likelihood maximizer. This likelihood request and response process is repeated until the likelihood maximizer finds the positions of all persons which give the highest likelihood. Finally, the likelihood maximizer then outputs the jointly estimated positions of all persons and begins a new cycle to estimate the positions of all persons at next time instance $t+1$. The following subsections describe the details

of each component of our tracker.

3.1 The Proposed Likelihood Model

We extend the single camera likelihood model that we have previously proposed in (Bo Bo et al., 2015) to a multi-camera likelihood model. We model a person m at position $\mathbf{s}^m$ as a fixed size cuboid placed at $\mathbf{s}^m$. Thus, the observation model of a person m at $\mathbf{s}^m$ from a smart camera c is in fact the projection of the cuboid at $\mathbf{s}^m$. A more accurate model would be to represent persons by detailed 3D shapes, better approximating to the shape of the human body. However, projecting a cuboid on the image plane is computationally more efficient than projecting more complex 3D shapes and allows speedups using integral images. Also, complex 3D shapes with more parameters, require these parameters to be optimized as part of tracking. Finally, the foreground observations are imperfect anyway and the there is little benefit in employing an "perfect model." For this reason, we also further approximate the projected cuboid by its bounding box $\omega_c(\mathbf{s}^m)$.

Consider a particular smart camera c, we have a foreground image $F_{t,c}$ as an observation at time t. Given the positions of M persons $\mathbf{s}_t^1,\dots\mathbf{s}_t^M$, we would like to compute the likelihood

$$p_c(F_{t,c}|\mathbf{s}_t^1,\dots\mathbf{s}_t^M) = l_c(\mathbf{s}_t^1,\dots\mathbf{s}_t^M). \tag{1}$$

of the observed image $F_{t,c}$. Since there are M persons in the scene, there will be M rectangles on the image plane, some of which may be intersecting each other. For simplicity, let $S_t \triangleq [\mathbf{s}_t^1,\dots\mathbf{s}_t^M]$ be the matrix whose columns are the M person positions and let $\Omega_c(S_t) \triangleq \bigcup_{m=1}^{M}\omega_c(\mathbf{s}_t^m)$ be the union of all rectangles in the image. Ideally, when S_t is equal to the true positions of all M persons, $\Omega_c(S_t)$ must contain all foreground pixels, and all background pixels must be outside of $\Omega_c(S_t)$. However, this ideal situation can never be reached since the cuboid model is far from the actual 3D shape of the human body. Moreover, a true foreground pixel can accidentally be detected as a background pixel and vice versa due to noise. Therefore, we formulate a likelihood function based on a noisy binary channel model.

Let us denote the probability that a true foreground pixel is wrongly detected as background as ε_f and the probability that a true background pixel is wrongly detected as foreground as ε_b. Obviously, the values of ε_f and ε_b highly depend on the performance of the foreground detection method. Here, we make an assumption that conditioned on S_t, all pixels of $F_{t,c}$ are statistically independent. Then the conditional probability $p_c(F_t^c|S_t)$ becomes a binomial distribution and the likelihood l_c of M person(s) at S_t, given the observation $F_{t,c}$ from a smart camera c can be computed as:

$$\begin{aligned} l_c(S_t) = &\prod_{\mathbf{r}\in\Omega_c(S_t)}(1-\varepsilon_f)^{F_{t,c}(\mathbf{r})}\varepsilon_f^{1-F_{t,c}(\mathbf{r})} \\ &\cdot\prod_{\mathbf{r}\notin\Omega_c(S_t)}(1-\varepsilon_b)^{1-F_{t,c}(\mathbf{r})}\varepsilon_b^{F_{t,c}(\mathbf{r})} \end{aligned} \tag{2}$$

The first factor of Eq. (2) evaluates how well the pixels *inside* $\Omega_c(S_t)$ agree with the hypothesis that S_t is correct, while the second factor evaluates how well the pixels *outside* of $\Omega_c(S_t)$ agree with the same hypothesis. In many heuristic methods for "template matching" in literature, the second factor is not taken into account, but it is required according to the proposed likelihood model and plays an important role. To simplify the computation, taking the logarithm of Eq. (2) gives

$$\begin{aligned} \mathrm{ll}_c(S_t) =& k+\lambda|\Omega_c(S_t)| + \lambda_f\sum_{\mathbf{r}\in\Omega_c(S_t)}F_{t,c}(\mathbf{r}) \\ &+\lambda_b\sum_{\mathbf{r}\notin\Omega_c(S_t)}(1-F_{t,c}(\mathbf{r}))\,, \end{aligned} \tag{3}$$

where $|R|$ is the area of an image region R, $\lambda_f \triangleq \ln\left(\frac{1-\varepsilon_f}{\varepsilon_f}\right)$, $\lambda_b \triangleq \ln\left(\frac{1-\varepsilon_b}{\varepsilon_b}\right)$, $\lambda \triangleq \ln\left(\frac{\varepsilon_f}{\varepsilon_b}\right)$, and $k \triangleq |I^c|\ln(\varepsilon_b)$ is a constant independent of S_t.

The interpretation of Eq. (3) is simplest when $\varepsilon_f = \varepsilon_b$, i.e., the chance of a true foreground pixel accidentally detected as background and the change of a true background pixel accidentally detected as foreground are equally likely. This simplification is reasonable in practice since parameters of foreground detection can be tuned to meet this condition. Thus $\lambda = 0$ and Eq. (4) becomes

$$\begin{aligned} \mathrm{ll}_c(S_t) =& k+\lambda_f\sum_{\mathbf{r}\in\Omega_c(S_t)}F_{t,c}(\mathbf{r}) \\ &+\lambda_b\sum_{\mathbf{r}\notin\Omega_c(S_t)}(1-F_{t,c}(\mathbf{r}))\,. \end{aligned} \tag{4}$$

This shows that log-likelihood increases when more foreground pixels are in $\Omega(S_t)$ and fewer foreground pixels are outside of $\Omega(S_t)$.

To compute the likelihood of jointly observing $F_{t,1},F_{t,2},\dots F_{t,C}$ in camera $1,2,\dots C$ respectively, given the positions of all persons S_t, we assume that the observations $F_{t,c}$ of different cameras are statistically independent, when conditioned on S_t. the joint likelihood l of all camera observations is then

$$p(F_{t,1},F_{t,2},\dots F_{t,C}|S_t) = \prod_{c=1}^{C} l_c(S_t) \tag{5}$$

and

$$\mathrm{ll}(S_t) = \sum_{c=1}^{C}\mathrm{ll}_c(S_t). \tag{6}$$

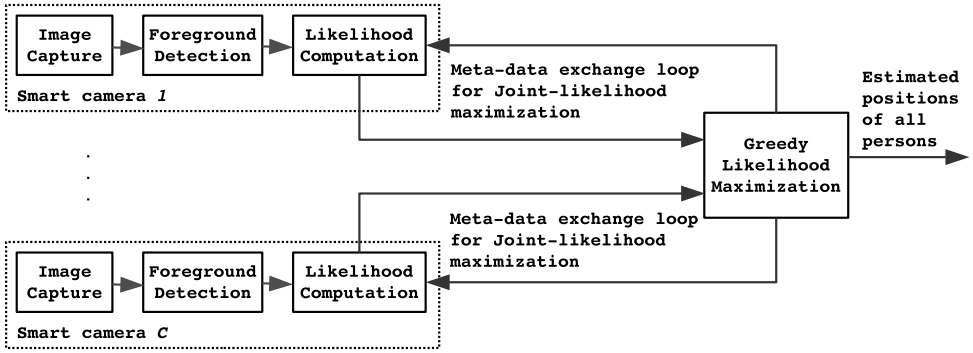

Figure 1: Block diagram showing the essential components and information flow of proposed decentralized multi-camera tracker.

3.2 Real-time Likelihood Maximization

Our tracker maximizes the product of the likelihood $p(F_{t,1}, F_{t,2}, \ldots F_{t,C}|S_t)$ and the motion model $p(S_t|S_{t-1})$. The motion model expresses the probability of persons moving from one position to the another over the course of one time unit. We assume that all persons move independently of each other. This assumption is reasonable when people are far apart, but not when they get closer together. In the latter case, our assumption e.g., allows that people move to exactly the same position, which is physically impossible. However, this assumption makes the likelihood computation computationally tractable and some of the possible inconsistencies can be detected and corrected using post-processing.

Because of the assumption, the motion model becomes:

$$p(S_t|S_{t-1}) = p(\mathbf{s}_t^1|\mathbf{s}_{t-1}^1), \ldots p(\mathbf{s}_t^M|\mathbf{s}_{t-1}^M). \quad (7)$$

Due to physical limitation, a person can move up to a particular distance limit d_{max} over the course of one time unit. For our tracker, we assume that all motion within a distance less than d_{max} is equally likely, i.e., we adopt a uniform motion model. We believe that this model is more robust to sudden motion changes than more traditional Gaussian models. The latter favors slow motion over fast motion, but the uniform model allows sudden motion changes. In the motion model, we also do not take into account people's velocities and motion directions at time $t-1$, as would be done in popular Kalman models. Many Kalman models in literature employ unrealistic assumptions, i.e., they favor continuous or even constant velocity or acceleration, which increases the probability of tracking loss. On the other hand, modeling discontinuous motion is quite difficult. Therefore, we prefer the "non-informative" uniform motion model.

A brute force search for $S_t = \{\mathbf{s}_t^1, \ldots, \mathbf{s}_t^M\}$ that gives the highest likelihood $\mathrm{ll}(S_t)$ in a discretized search space is a very time consuming task because it requires the computation of likelihood using Eq. (6) for all possible combinations of discretized people's positions on the ground plane. Fortunately, because of the constrained uniform motion model, the search space is already greatly reduced: each person position can only be in a circular region with radius d_{max} around the last known position of each person. Even in this constrained search space, the computation time for brute force likelihood maximization increases exponentially with the number of persons M and with the discretization accuracy.

However, by using the greedy likelihood maximization as shown in the pseudo code in Algorithm 1, we can make the computation time increases linearly with the number of persons M. The inputs of the algorithm are the positions of all persons at time $t-1$ sorted in descending order of $\sum_{c=1}^{C} \Omega_c(\mathbf{s}_{t-1})$, i.e., the sum of the area of projected cuboids in all views for a person at position $\mathbf{s}_{t-1}$. The reason behind this sorting is that a larger projected cuboid contributes more in likelihood computation. Thus it is desirable to start the greedy maximization from a person's position which contributes the most to the joint likelihood computation.

In Algorithm 1, we initialize S_t with all previously known positions in S_{t-1}. Then, each position in S_t is optimized one at a time. The following steps are repeated until all positions in S_t are updated. At each iteration, a previously known location at the m^{th} column is selected as $\mathbf{s}_{t-1}^m$. We define H as a set of all discretized positions that fall within the circle with radius d_{max}, which center is at $\mathbf{s}_{t-1}^m$. Then, an exhaustive search is performed for each position in H as $\mathbf{s}_t^m$ which gives the highest likelihood while keeping the positions of the remaining person(s) fixed. The previously known position at the m^{th} column in S_t is replaced by $\mathbf{s}_t^m$. The iteration in Algorithm 1 can be repeated until a stable solution is reached. However, we experimentally found that terminating the iteration when all the positions in S_t are updated gives the op-

Algorithm 1: Greedy likelihood maximization.

1: Initialize $S_t \triangleq [\mathbf{s}_t^1, \ldots \mathbf{s}_t^M]$ with $[\mathbf{s}_{t-1}^1, \ldots \mathbf{s}_{t-1}^M]$
2: **for** $m = 1$ **to** M **step** 1 **do**
3: $\quad \mathbf{s}_t^m$ = the m^{th} column of S_t
4: $\quad H^m$ = the set of points $\mathbf{s}$ with $\|\mathbf{s} - \mathbf{s}_t^m\| \leq d_{\max}$.
5: $\quad \hat{\mathbf{s}}_t^m = \max_{\mathbf{s} \in H^m} \text{ll}([\mathbf{s}_t^1, \ldots \mathbf{s}_t^m, \mathbf{s}, \mathbf{s}_t^{m+1}, \ldots \mathbf{s}_t^M])$
6: $\quad$ Update S_t by replacing its m^{th} column by $\hat{\mathbf{s}}_t^m$.
7: **end for**

timal trade-off between the accuracy and the computational time.

3.3 Decentralized Processing

This subsection describes how decentralized processing is done in Algorithm 1 as well as the meta-data exchanges between nodes. At each iteration of the loop in the algorithm, all smart cameras compute $\text{ll}_c([\mathbf{s}_t^1, \ldots \mathbf{s}_t^m, \mathbf{s}, \mathbf{s}_t^{m+1}, \ldots \mathbf{s}_t^M]) : \forall s \in H^m$ using Eq. 4 locally and send computed likelihoods (a set of numbers) to the likelihood maximizer. This meta-data exchange is the most significant amount of data exchange between smart cameras and the likelihood maximizer. The numbers of likelihoods to be sent depends on the grid size of the discretized ground plane and d_{max}. The detailed analysis on communication bandwidth between nodes will be discussed in Section 5. Once required likelihoods are received from all cameras, the likelihood maximizer computes $\text{ll}([\mathbf{s}_t^1, \ldots \mathbf{s}_t^m, \mathbf{s}, \mathbf{s}_t^{m+1}, \ldots \mathbf{s}_t^M])$ using Eq. 6 and proceeds to the remaining steps in the algorithm.

4 EVALUATION

4.1 Test Videos

To evaluate the performance of our tracker, we use video sequences captured in a room of 8.8×9.2 m^2. Each sequence has a total duration of approximately six minutes and is captured using four calibrated cameras with overlapping views. There are a table and chairs at the center of the meeting room. All videos are captured at 20 fps with a resolution of 780×580 pixels. Up to four people are walking in the scene and they are often occluded by other persons or furniture. During the video capturing of the last sequences, lights are switched on and off a couple of times to create a scenario of rapid illumination changes. Ground plane positions for each person are manually annotated every 20 frames.

Walking Sequence: The purpose of this sequence is to measure the performance of the tracker when people are just walking around in the scene. In this sequence, four people come into the meeting room one after another, greet each other, walk around in the room and finally leave the room. Most of the existing publicly available datasets focus on tracking people over a short period of time. For example, tracking a person passing through the scene which may last for a minute or two. In contrast, the same people walk around in the room for about five minutes in this sequence.

Meeting Sequence: The majority of publicly available dataset for people tracking are made in a scenario in which people are moving around in the scene. We are interested in evaluating our tracker in a scenario in which people walk, sit down and walk again. Thus, we capture this video sequence in a meeting room scenario. In this video, four people come into the room, greet each other and sit around the table. During the meeting, a person sometimes leave the chair to give a presentation.

Unsteady Lighting Sequence: To evaluate the performance of our tracker when there are rapid illumination changes, we capture the video sequence of people walking while the illumination of the scene is deliberately changed suddenly several times. The scenario of this sequence is the same as *Walking* sequence except for the lighting in the room which is deliberately reduced to half and then completely turned off several times. Note that although the lights are switched off in the room, there is still dimmed light coming from outside of the room.

4.2 Evaluation and Comparison

The performance of our tracker is evaluated on three aforementioned video sequences. The same set of parameters is used for processing all sequences. We experimentally select the parameters, i.e., brute force search for a parameters combination, which gives the optimal performance on all video sequences. The parameters of foreground extraction are set as follows: the sliding window size $k = 9$, the correlation coefficient threshold $\rho_{min} = 0.98$. For likelihood maximization, the ground plane is discretized with 10 cm $\times$ 10 cm grid. The other parameters are set $\varepsilon_f = \varepsilon_b = 0.001$ and the maximum distance that a person can move between two consecutive frames $d_{max} = 50$ cm. Using aforementioned ground truth, we compute the most widely used systematic evaluation metrics (Bernardin and Stiefelhagen, 2008); the Multiple Object Tracking Accuracy (MOTA) and the Multiple Object Tracking Precision (MOTP) of the resulting trajectories.

We use the same hit/miss threshold 100 cm as in (Bredereck et al., 2012), (Andriyenko and Schindler, 2011), (Milan et al., 2013). For both metrics, higher value indicates better performance.

Moreover, we compare the performance of our tracker to the publicly available state of the art tracker of Berclaz *et al.* (Berclaz et al., 2011) in terms of both MOTA and MOTP. Their approach computes the Probabilistic Occupancy Map (POM) on the ground plane using the results of foreground detection from all camera views. Their POM computation only considers information from the current frames of all cameras, but not the positions estimated in the previous frame. However, in the data association step, *K*-Shortest Path (KSP) optimization is used to find the optimal trajectory of a person by enforcing temporal continuity constraints over POMs computed from both past and future frames. In this paper, we will refer to their tracker as POM-KSP.

For fair comparison, we use the same foreground detection method as well as the same set of parameters for the POM-KSP tracker. As in our tracker, we also set 10 cm $\times$10 cm grids for the POM-KSP tracker. Since we set $d_{max} = 50$ cm for our tracker, we also set maximum distance traveled between consecutive frames for POM-KSP to be 5 grid locations.

5 RESULTS AND DISCUSSION

Our tracker achieves a MOTA value of 100% for *Walking* sequence with a MOTP value as high as 86%. In the *Meeting* sequence, the MOTA value drops slightly to 98% with MOTP of 77%, which is still showing quite good performance for such challenging sequence. Performance comparison to publicly available state of the art tracker shows that our tracker outperforms.

In our tracker, we just approximate the volumetric model of an *upright standing* human as a cuboid with fixed width and height. The model fits well to the observation as long as a person is standing. However, when a person sits down, the height of a person is just a bit over an half of the standing height and the actual observed foreground shape is far from the shape of the projected model. Moreover, lower body parts are occluded by both table and chair when a person sits down and foreground detection fails to detect occluded body parts. These are the main reasons for small drop in MOTA and relatively high decrement in MOTP. The example tracking results from our tracker for both the *Walking* and the *Meeting* sequences are shown in Fig. 2. The bounding box around each person is in fact the $\Omega_c(\mathbf{s}_t^m)$ and a different box color shows different identity. The tailing dots of the bound box, which has the same color as the box, is the projection of the positions estimates from the past 80 frames.

The evaluation of the *Unsteady Lighting* sequence shows that our tracker is robust to sudden illumination changes. Even in such difficult scenario, our tracker achieves a MOTA value of 97% and a MOTP value of 84%. The drop in MOTA is mainly caused by false positive trajectories due to the presence of false positive foreground blobs. These false positive foreground blobs appear every time the light of the room is completely off, i.e., the captured image tends to be very noisy when illumination in the scene is very low. The example frames for all four views of the *Unsteady Lighting* sequence is shown in Fig. 3.

The projected cuboid to all views $\Omega_c(\mathbf{s}_t^m)$ and the projected points of 80 previously estimated positions are shown as in Fig. 2. The first row of Fig.3 shows the scene under under usual illumination. The second row shows the frame after five seconds (100 frames), where the lighting in the room is completely off. Within a five seconds interval, the lighting in the room is firstly reduced to half and then fully off. Despite the presence of sudden illumination changes, we can see that our tracker is still able to track all persons correctly.

As aforementioned, we compare the performance of our tracker to the state of the art tracker of Berclaz *et al.* (POM-KSP) as shown in Table 1. The comparison shows that our tracker outperforms in all three sequences. The POM-KSP performs well in sequences containing only upright walking persons, i.e., the *Walking* and the *Meeting* sequences. However, the POM-KSP performs poorly in the *Meeting* sequence. The reason could be that their tracker is not designed to track sitting persons as there is reported results in their paper (Berclaz et al., 2011) are only for upright moving people. Moreover, Berclaz *et al.* (Berclaz et al., 2011) reported that their POM-KSP tracker is sensitive to *false detections and missing ones*. In the *Unsteady Lighting* sequence, the sudden illumination changes cause foreground detection to produce false positive blobs which leads to false detection in POM thus lower accuracy in tracking. However, for our tracker, these false positive blobs cause only 3% drop in MOTA. This shows that our tracker is less sensitive to false positive detections.

In each iteration of greedy likelihood algorithm (Algorithm 1), each smart camera sends 81 likelihood values (numbers) to the likelihood maximizer when using 10 cm $\times$10 cm grids and $d_{max} = 50$ cm. Since the number of iteration is the number of person being tracked, each smart camera sends $81 \times m$ numbers per

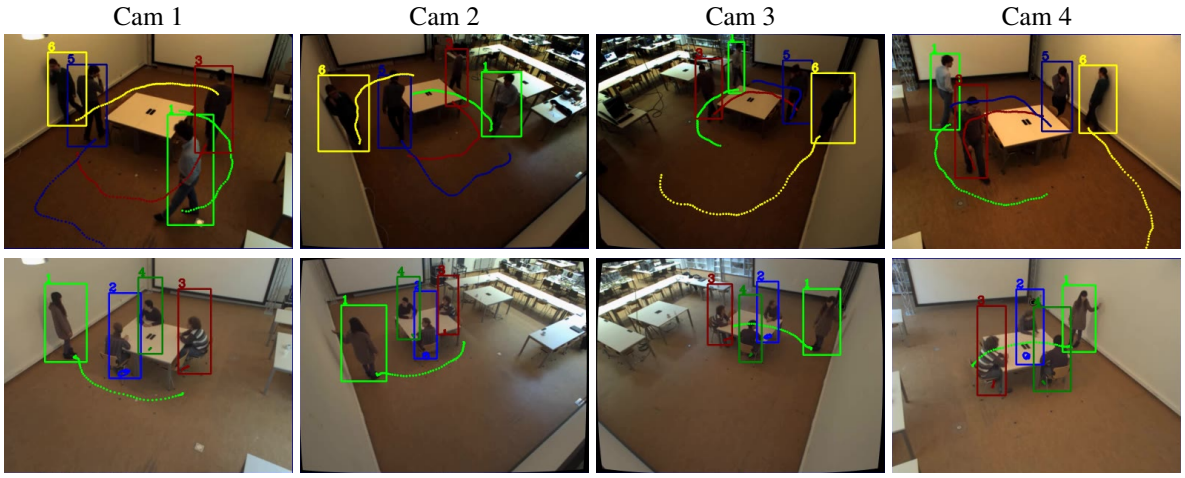

Figure 2: Example tracking results obtained by our tracker. The first row shows the example trajectories in four views of the *Walking Sequence* and the second row shows the *Metting Sequence*. Different bounding box colors indicates different identities.

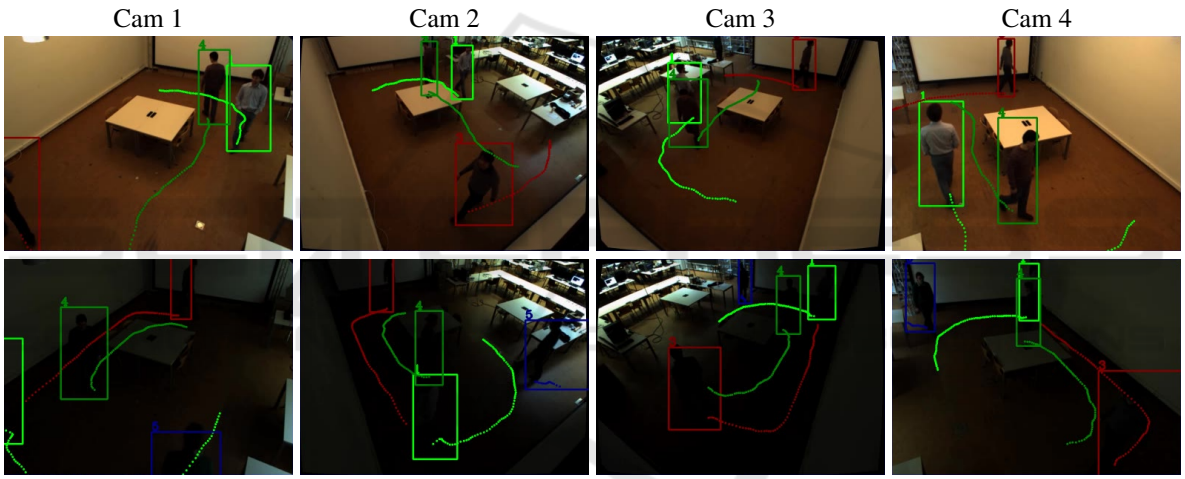

Figure 3: Example tracking results obtained by our proposed tracker in the *Unsteady Lighting* sequence. Different bounding box colors indicates different identities.

frame, i.e., the amount of data exchange increases linearly with the number of person being tracked. This shows that our tracker is more communication efficient than those follows centralized architecture.

6 CONCLUSION

In this paper, we present a novel method for tracking multiple people simultaneously within a smart camera network by maximizing the likelihood of observed foreground image in all camera views given the positions of all persons. The evaluation results show that our tracker achieves MOTA value as high as 100% and MOTP value as high as 86%. We also show that our tracker is able to track people reliably even when they are sitting down. This is a desirable advantage in several applications, such as smart meeting room, since people are sitting in the scene most of the time. Moreover, our tracker is robust to sudden illumination changes.

When we compare to publicly available state of the art tracker (POM-KSP), our tracker outperforms. Currently, our prototype tracker is implemented on an ordinary computer as a single thread C++ program, which processes all four camera views sequentially. Without code optimization, the average processing time of our tracker is 200 milliseconds per frame, approximately 50 milliseconds per frame per camera view, on a single core of Intel Core2Quad @ 2.66 GHz with 8 GB of memory. Therefore, we believe that our tracker can reach the processing speed

Table 1: MOTA and MOTP comparison.

	MOTA		MOTP	
Sequence	Proposed	POM-KSP	Proposed	POM-KSP
Walking	100%	93%	86%	83%
Meeting	98%	54%	77%	75%
Unsteady Lighting	97%	83%	86%	84%

of 20 frames per second if the processing is parallelized in a distributed architecture, i.e., processing for each camera is implemented as a separate thread or on a smart camera.

ACKNOWLEDGEMENT

The work was financially supported by FWO through the project G.0.398.11.N.10 “Multi-camera human behavior monitoring and unusual event detection”.

REFERENCES

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

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

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

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

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

Bo Bo, N., Deboeverie, F., Veelaert, P., and Philips, W. (2015). Real-time multi-people tracking by greedy likelihood maximization. In *International Conference on Distributed Smart Cameras (ICDSC2015)*, Seville, Spain.

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

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

Henriques, J. F., Caseiro, R., and Batista, J. (2011). Globally optimal solution to multi-object tracking with merged measurements. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2470–2477.

Milan, A., Schindler, K., and Roth, S. (2013). Detection- and trajectory-level exclusion in multiple object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3682–3689.

Pirsiavash, H., Ramanan, D., and Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR ’11, pages 1201–1208.

Wojek, C., Roth, S., Schindler, K., and Schiele, B. (2010). Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV’10, pages 467–481.

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

Yun, Y., Gu, I.-H., and Aghajan, H. (2012). Maximum-likelihood object tracking from multi-view video by combining homography and epipolar constraints. In *The 2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*.

Zhang, L., Li, Y., and Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*.