Multi-Person Bayesian Tracking with Multiple Cameras

Jian Yao Jean-Marc Odobez

Idiap Research Institute
Centre du Parc, Rue Marconi 19, CH-1920 Martigny, Switzerland

Abstract

Object tracking is an important task within the field of computer vision, which is driven by the need to detect interesting moving objects in order to analyze and recognize their behaviours and activities. However, tracking multiple object is a complex task due to a large number of issues number ranging from the different types of sensing set-up to the complexity of the object appearance and behaviours. In this chapter, we analyze some of the important issues to solve for multiple object tracking, reviewing briefly how they are addressed in the literature. We then present a state-of-the-art algorithm for the tracking of a variable number of 3D persons in a multi-camera setting with partial field-of-view overlap. The algorithm illustrates how in a Bayesian framework the raised issues can be formulated and handled. More specifically, the tracking problem relies on a joint multi-object state space formulation with individual object states defined in the 3D world. It involves several key features for efficient and reliable tracking like the definition of appropriate multi-object dynamics and a global multi-camera observation model based on color and foreground measurements, the use of the Reversible-Jump Markov Chain Monte Carlo (RJ-MCMC) framework for efficient optimization, the exploitation of powerful human detector outputs in the MCMC proposals to automatically initialize/update object tracks. Experimental results on challenging real-world tracking sequences and situations demonstrate the efficiency of such an approach.

Key words: Tracking, multi-camera, 3D model, multi-objects, surveillance, color-histograms, Bayesian, MCMC, Reversible-Jump, uncertainties, human detector

1 Introduction

Multiple object tracking (MOT) in video is one of the fundamental research topics in dynamic scene analysis, as tracking is usually the first step before applying higher level scene analysis algorithms such as automated surveillance, video indexing, human-computer interaction, traffic monitoring, and vehicle

navigation. While fairly good solutions to the tracking of isolated objects or small number of objects having transient occlusion have been proposed in the past, MOT remains challenging with higher densities of people, mainly due to inter-person occlusion, bad observation viewpoints, small resolution images, entering/leaving of people, etc. These situations are often encountered in the visual surveillance domain. In the following we discuss some of the key factors which affect a tracking algorithm, and then introduce our algorithm.

1.1 Key Factors and Related Works

As stated above there are several issues which makes the tracking difficult: background clutter; small object size; complex object shape, appearance, and motion, and their changes over time or across camera views; inaccurate/rough scene calibration or inconsistent camera calibration between views for 3D tracking; large field-of-view (FOV) cameras with small overlap or without overlap, and real-time processing requirements. In what follows, we discuss some of the key components to take into account when designing a tracking algorithm, and relate them to the raised issues.

Set-ups and Scenarios. In the past decade, an abundance of approaches and techniques have been developed on multi-object tracking. They can be distinguished according to the physical environement considered, the set-up (where and how many sensors are used), and the scenario (under which conditions -e.g. crowdness level- the tracking is expected to perform). A first set of environment for tracking are the so-called *smart spaces* [1]. These are indoor environments -homes, offices, classrooms- equipped with multiple cameras, and also audio sensing systems and networked pervasive devices that can perceive ongoing human activities and respond to them. These settings usually involve the tracking of few people. They are usually equipped with multiple cameras, providing good image quality, and people sizes in the images are relatively high and of the same value accross camera views. In this context, robust and accurate tracking results have been demonstrated, e.g. [2,3], and current goals are to recover the pose of objects in addition to their localization, exploit other modalities such as audio [4], and characterize people activities.
Another set of environement are *open spaces*. as encountered in surveillance, for instance airport or metro indoor spaces, or outdoor areas like parkings or school campus [5–7]. In contrast to the previous case, the monitored space is much larger, and usually covered with only a few cameras. In general, robust and accurate tracking across large field-of-view cameras is difficult, as objects can be very small or have large image projection size variations within and across views due to depth effects, and object appearance is unclear and similar from one object to another due to the small scale. However, when the crowding level is not too high (e.g. when monitoring outdoor corporate parkings), good tracking can still be achieved.
In both the smart and open space cases, the view-point is an important vari-

able which affect the tracking difficulty. When seen from above, people in a group can still be distinguished from each other. When seen from the floor level or from a low view point, people will occlude each other. In this case, a tracking algorithm will have to explicitely account for this situation, in order to take measurements only on the un-occluded component of people, and predict the motion of occluded people.

Object state representation. The tracking problem depends on what kind of object state representation one wishes to recover. In its simplest form, object tracking can be defined as the problem of estimating the location of an object in the 2D image plane or in the 3D space as it moves around a scene, i.e. a tracker should assign consistent labels to the tracked objects in one or multiple video streams. Additionally, depending on the scenario and set-up, a tracker can also provide object-centric information, such as its size, orientation, or its pose. The selection of an adequate state space is a compromise between two goals: on one hand, the state space should be precise enough so as to model as well as possible the information in the image and provide the richest information to further higher level analysis modules. On the other hand, it has to remain simple enough and in adequation to the quality level of the data in order to obtain reliable estimates and keep the computation time low. One approach is to define objects in the 2D image plane, e.g. with their position, speed, scale, [8], and possibly the different object parts, like head-shoulder, torso, or legs as done in [9]. However, whenever possible, defining the object in the 3D space using a model-based approach is more appropriate and presents several advantages over a 2D approach. First, parameter setting, in most cases, will have a physical meaning (e.g. the standard height of a person, the average speed of a walking person [7]). Similarly, prior information about the state values will be more easy to specify, as they are somewhat 'built-in': for instance, according to the 3D position, we automatically know what should be the size of a person in the image plane. Finally, occlusion reasoning -when tracking multiple people- is simplified when using the 3D position. To represent people, generalized 3D cylinders or ellipsoids are often used when enough resolution is available [7, 10]. Alternatively, a simplified 2D version not corresponding to an explicit 3D model can be used. For instance, Zhao et al. [5, 6] parameterize a 3D human through ellipses caracterized by the head position, the person height, and 2D inclination. Note that the 2D inclination on the image plane is very important because the people vertically standing on the floor may not appear vertically on the image due to camera distorsions.

Object tracking problem formulation. Several approaches can be used to formulate the tracking problem. In a simple approach, tracking can be done by detecting position of objects at each frame and then matching these detections accross time using motion heuristics to constrain the correspondence problem [11]. For instance, when people are seen from a far distance with a static camera, background subtraction is usually applied. Blobs or connected components are extracted, possibly classified in different categories (person vs

vehicules, person vs group), and matched in time. However, since blobs do not always correspond to single objects, splitting of blobs into several tracks and the merge of tracks into one blob occur. To handle this issue, reasoning about the object counts and their appearance can be used to identify single tracks through Bayesian networks or graph analysis [12, 13]. For instance, Bose et al. [13] proposed the fragmentation and grouping scheme to deal with these situations. However, these approaches can not be applied when objects are closer to the camera, as the occlusions become too complex.

In past years, Bayesian state-space formulation have been shown to be very successful to address the multi-person tracking problem [5,7,8,10,14,15]. Some authors are using a single-object state-space model [16, 17], where the modes of the state are identified as individual objects. Fleuret et al. [18] use a greedy approach, by extracting the different tracks one by one from instantenous object detection features using a Hidden Markov model (HMM), and removing the detection features associated with the already extracted tracks. However, only a rigourous formulation of the MOT problem using a multi-object state space allows to formalize in a principled way the different components that one may wish for a tracker: uniquely identifying targets, modeling their interactions, handling the variability of the number of objects using track birth and death mechanisms. As a pionneer work, the BraMBLe system [10] was able to track up to 3 persons from a single camera and blob-likelihood based on a known background model and appearance models of the tracked people.

While the probabilistic tracking framework is appealing, it does not solve all the problems by itself. First, as highlighted in [10], one needs to have a global observation model with the same number of observations to deal with multi-object configurations varying in number, in order to obtain likelihoods of the same order of magnitude for configuration with different number of objects. This render somehow the usage of object oriented individual likelihood terms problematic, e.g. if one would defined the likelihood as the product of individual object likelihoods. Besides, due to the curse of dimension of the multi-object state space, solving the inference problem is not a straigthforward issue. The use of a plain particle filter [10] will quickly fails when more than 3 or 4 people need to be tracked. However, more recently, MCMC stochastic optimization, with reversible-jump [19] have been shown to be more effective at handling the large dimensional state. The algorithm presented in this paper belongs to this category of approaches.

Dynamics and interaction models. In tracking, it is often important to specify some prior knowledge about the temporal evolution of the object representation to discard wrong matches, or, in occlusion cases, predict the object location until the occluded object re-appears. Single-object dynamics usually assumes some continuity model (in position, speed, or acceleration) over the state variables, whose parameters can be learned from training data [20]. Linear models were commonly employed to comply with the Kalman filtering framework [21], but this happens not to be a restriction when using particle

filters [10]. In order to obtain more precise and meaningful dynamics, the use of auxiliary variables caracteristics of the dynamics can be used in switching models. For instance, in human tracking, a discrete variable indicating whether a person is walking or static could be added in the state, and would allow to consider different dynamics according to the person activity.
In the multi-object case, the state dynamics has to be defined for a varying number of objects. Indeed, specifying this dynamics allows to properly handle the birth and death of objects, as described later in the chapter. In addition, the modeling allows to introduce object interaction models [8,14,22], by defining priors over the joint state space. Such priors are usually based on object proximity, which prevents objects of occupying the same state space region or explaining twice the same piece of data. Technically, this can be achieved by defining a pairwise Markov Random Field (MRF) whose graph nodes are defined at each time step by proximity objects. Qualitatively, such models will be useful in crowded situations and to handle occlusion cases. More complex group dynamics can be defined. In [23], a model relying on the discrete choice theory was used to handle object interactions, by modeling and learning the behaviour of a given pedestrian given his assumed destination and the presence of other pedestrians in a nested grid in front of him. Due to the complexity of the model, however, the tracking task was solved independently for each person at each time step.

Detection and tracking. Any tracking approach requires an object detection mechanism either in every frame or when the object first appears in the scene to create a track. A common approach is to use temporal information such as foreground detection, frame differencing or optical flow to highlight changing regions in consecutive frames, and propose to start tracks where such information is not yet accounted by already existing objects [5, 8, 10]. Indeed, when object detection can be done reliably enough at each frame, it is then possible to perform tracking-by-detection, i.e. only rely on the localization output to link detection over time. This is the case in blob based approaches cited above [12, 13], but also when multiple cameras are used [3]. When possible, this is a powerful approach which allows to integrate long term trajectory information in a lightweight manner since only state features are involved, and not images. For instance, Wu and Nevatia [9] used a learned detector to find human body parts, combine them, and then initialize the trajectory tracking from the detections. However, in many cases, obtaining detections at a majority of time steps for each object is difficult to achieve. Still, the use of powerful detectors, learned using training data using boosting or support-vector-machines, can be efficiently exploited for track initialization and better localization of objects during inference [5] as shown later in this chapter. However, it brings some difficulties to real-time tracking applications to some extent, as detectors can be time consuming.

Observations and multi-camera tracking. How we measure the evidence of the observed data with respect to a given multi-object state is one of the

most important point for a tracker. In multi-object tracking, color information is probably the most commonly used cue [5,7,8,24,25]. Color information is often represented using probability distribution functions represented by parametric models or histograms. They present the advantage of being relatively invariant under pose and view changes. In addition, to introduce some geometrical information and be more robust to the visual clutter, color information is usually computed for different body parts. While color is helpful to maintain the identity of tracked people, a key issue is to create and adapt over time the color model of the object to be tracked, as the use of a predefined color model is usually infeasible (except in some specific cases like sport games). This requires to identify which pixels in the image belong to a person to initialize a dedicated color histogram [26], and this is often done relying on foreground detection [5, 7, 8, 24, 25, 27]. Another commonly used observation for localization is foreground detection (probabilities or binary masks), which is usefull to assess the presence of objects in the image [3, 5, 7, 8]. For tracking objects with complex shapes, or to measure pose information, color is usually not sufficient and contour cues often needs to be extracted. For instance, Haritaoglu et al. [28] use silhouettes for object tracking in a surveillance application. Alternatively, one can represent people using sets of local templates or patches and geometric information, as proposed by Leibe et al. [29]. In addition, other modalities such as audio microphone arrays can be used for localization, especially in smart rooms [4].

Recently, growing interest has concentrated upon tracking objects using multiple cameras to extend the limited viewing angle of a single fixed camera. There are two main reasons to use multiple cameras for tracking [3, 24–26, 30]. The first one is the use of depth information for tracking and occlusion resolution. The second one is to increase the space under view for global tracking since it is not possible for a single camera to observe large spaces. Kim and Davis [24] proposed a multi-view multi-hypothesis approach, defined in a particle filtering framework, to segmenting and tracking multiple persons on a ground plane. Fleuret *et al.* [3] proposed an algorithm that can reliably track multiple persons in a complex environment and provide metrically accurate position estimates by combining a probabilistic occupancy map. Du and Piater [25] presented a novel approach to perform ground-plane single target tracking fusing multiple camera information using sequential belief propagation. This method performs very well and can handle the imprecise foot positions and calibration uncertainties, which is a key issue in multi-camera systems where it is not always possible to perform a precise and euclidian calibration of all the cameras. Most approaches use centralized systems in which informations from multiple cameras are jointly fused for tracking. These tracking systems need a very efficient method applied on a powerful computer for real-time applications. Other approaches are using distributed systems in which tracking is conducted independently at each camera, and then results from the different cameras are fused and combined at a higher level. For instance, [15] presents a

probabilistic decentralized approach which allows to more efficiently achieved using a group of computers.

1.2 Our Approach and Chapter Outline

In this chapter, we present our approach to automatically detect and track a variable number of people in a multi-camera environment with partial field of view overlap. We believe it includes most state-of-the-art components of MOT tracker in Bayesian tracking. More precisely, we adopt a multi-object state space Bayesian formulation, solved through RJ-MCMC sampling for efficiency reasons [8,14]. The proposal (i.e function sampling new state configurations to be tested) used in this scheme takes advantage of a powerful machine learning human detector allowing to efficiently update tracks or initialize new tracks. We adopt a 3D approach where object states are defined in a common 3D space allowing to represent people with a body model, and to facilitate occlusion reasoning. The multi-camera fusion is solved by using global likelihood models over foreground and color observations. Our algorithm combines and integrates efficient algorithmic components in our framework which have been shown in the past (often separately) to be essential for accurate and efficient tracking, and to prsents additional techniques to solve specific issues as detailed below.

To efficiently handle the interaction between multiple objects for avoiding multiple objects to occupy a same state space region, we propose to refine priors over the joint state space by exploiting both the body orientation in the definition of proximity, and by using the prediction of the future object state to model that moving people tend to avoid colliding each other.

Multi-camera tracking in surveillance scenarios is usually quite different than tracking in indoor rooms. Larger field-of-view (FOV) cameras are used to cover more physical space, the overlaps between the FOV are smaller, and people appear with dramatically different image resolutions due to their placements and points of views. As a consequence, a small and seemingly not significative 2D position change (e.g. one pixel) in one view can correspond to a large position change in the other view, as illustrated in Fig. 2. This is particularly problematic at *transitions* between FOV cameras, when a person enters a new view which has much higher resolution than the current one. As due to this uncertainty, the projection of the current estimate does not match at all the person in the new view. As a result, the tracker will assume that the person remains only in the first view, and will initialize a new track in the new view. To solve this issue, the proposed algorithm integrates in the 3D object state prior a component which models the effects of the image estimation uncertainties according to the views in which the object is visible, and to use a proposal taking into account the human detection output per view to draw samples at well localized places in the new view.

In addition, one contribution of this chapter is an image rectification step

allowing to reduce people geometric appearance variability (esp. remove people slant) in images due to the use of of large FOV cameras.

This section introduced the object tracking problem, and described the key issues and applications of object tracking. It then presented a brief overview on related works and summarized our approach. The rest of this chapter is organized as follows. Section 2 describes the multi-camera multi-person Bayesian tracking framework with the state space and model representation. The main features of our proposed tracking framework are then described in Sections 3-5: Sections 3 and 4 introduce the dynamical model and observation model for multi-object tracking, respectively. Section 5 describes the Reversible-Jump Markov Chain Monte Carlo sampling approach for optimization. 6 shows how we can rectify images to remove people slant, and shows tracking results on real data. 7 concludes the chapter.

2 Bayesian Tracking Problem Formulation

The goal is to track a variable number of persons from multiple overlapped camera views. To successfully achieve this objective, we use a Bayesian approach. In the Bayesian tracking framework, the goal is to estimate the conditional probability $p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t})$ of the joint multi-person configuration $\tilde{\mathbf{X}}_t$ at time t given the sequence of observations $\mathbf{Z}_{1:t} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_t)$. This posterior probability $p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t})$, known as the filtering distribution, can be expressed recursively using the Bayes filter equation:

$$p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t}) = \frac{1}{C} p(\mathbf{Z}_t|\tilde{\mathbf{X}}_t) \times \int_{\tilde{\mathbf{X}}_{t-1}} p(\tilde{\mathbf{X}}_t|\tilde{\mathbf{X}}_{t-1}) p(\tilde{\mathbf{X}}_{t-1}|\mathbf{Z}_{1:t-1}) d\tilde{\mathbf{X}}_{t-1}, \tag{1}$$

where the dynamical model $p(\tilde{\mathbf{X}}_t|\tilde{\mathbf{X}}_{t-1})$ governs the temporal evolution of the joint state $\tilde{\mathbf{X}}_t$ given the previous state $\tilde{\mathbf{X}}_{t-1}$. and the observation likelihood model $p(\mathbf{Z}_t|\tilde{\mathbf{X}}_t)$ measures the fitting accuracy of the observation data $\mathbf{Z}_t$ given the joint state $\tilde{\mathbf{X}}_t$. C is a normalization constant. In non-Gaussian and non-linear cases, the filter equation can be approximated using Monte Carlo methods, in which the posterior $p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t})$ is represented by a set of N samples $\{\tilde{\mathbf{X}}_t^{(r)}\}_{r=1}^N$. For efficiency, in this work we use the Markov Chain Monte Carlo (MCMC) method, where the set of samples have equal weights and form a so-called Markov chain. Consequently, we obtain the following Monte Carlo approximation:

$$p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t}) \approx \frac{1}{C} p(\mathbf{Z}_t|\tilde{\mathbf{X}}_t) \times \sum_{r=1}^{N} p(\tilde{\mathbf{X}}_t|\tilde{\mathbf{X}}_{t-1}^{(r)}). \tag{2}$$

Thus to define our filter, the main elements to be specified are: our multi-object state space; the dynamics; the likelihood model, and an efficient sampling scheme in order to effectively place the particles at good location during

optimization. We first introduce our state model in the next subsections.

2.1 Single Object 3D State and Model Representation

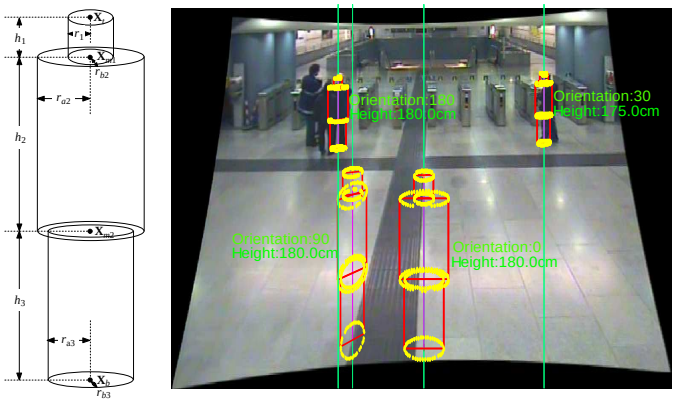
As stated in the introduction, the selection of the state space is a compromise between what can be expected to be reliably estimated from the observations, and the richness level of the information we want to extract. In the current situation, we decided to use a state space defined in the 3D space, comprising the person location and speed on the ground plane, as well as his height and orientation.

Given these parameters, we modeled people using general cylinders, as illustrated in Fig. 1. Given the resolution of the images, we decided to use one cylinder for the head, one for the torso, and one for the legs. To account for the 'flatness' of people (or in other words, the width of people is usually larger than their thickness), we decided to use elliptic cylinders (i.e. the section of the cylinder is an ellipse). Utilizing this 3D human body model, one person standing on the ground plane with different orientation should produce different projected models in which the main difference is the width of the projected human bodies. There are fixed physical aspect-ratios of these three body parts, e.g. height ratios 2:7:6 for head, torso and legs. Thus, in summary, the state space is represented by a 6-dimensional column vector:

$$\mathbf{X}_{i,t} = (x_{i,t}, y_{i,t}, \dot{x}_{i,t}, \dot{y}_{i,t}, h_{i,t}, \alpha_{i,t})^\top , \tag{3}$$

where $\mathbf{u}_{i,t} = (x_{i,t}, y_{i,t})^\top$ denotes the person ground plane 2D position in the 3D physical space. Variables $\dot{\mathbf{u}}_{i,t} = (\dot{x}_{i,t}, \dot{y}_{i,t})^\top$, h and $\alpha_{i,t}$ denote the velocity, the height of the object (in cm), and the orientation w.r.t. the X-direction on the ground plane, respectively. Fig. 1 shows the body model along with the projection of the body model, for different state values, on one image.

However, to reduce computation, we indeed projected each of the body part into one 2D bounding box, which will be used to compute observation likelihood as described in Section 4. To get such a bounding box for a given part, we first find the 3D coordinates of the 4 tangent points to the top and bottom elliptical sections of the body part. These points are then projected into the 2D image plane, and the minimum bounding-box containing these four points is used to represent the projection of each cylinder. Examples can be seen in the result Section. Most of the time, such a projection is a good approximation to the full projection. However, when people are close to the camera, intersections among three projected bounding boxes can occur (see for instance Fig. 9).



(a) (b)

Fig. 1. (a) the 3D human body model, which consists of three elliptic cylinders representing head, torso and legs, respectively; (b) projections of the body model in the rectified image for different state values. Notice the change of width due to variation of the body orientation.

2.2 The Multi-Object State Space

To track a variable number of people, we defined the joint state space of multiple objects as:

$$\tilde{\mathbf{X}}_t = (\mathbf{X}_t, \mathbf{k}_t), \tag{4}$$

where $\mathbf{X}_t = \{\mathbf{X}_{i,t}\}_{i=1\ldots M}$, M is the maximum number of objects appearing in the scene at any given time instant, and $\mathbf{k}_t = \{k_{i,t}\}_{i=1\ldots M}$ is a M-dimensional binary vector. The boolean value $k_{i,t}$ signals whether the object i is valid/exists in the scene at time t ($k_{i,t} = 1$), or not ($k_{i,t} = 0$). The identifier set of existing objects is thus represented as $\mathcal{K}_t = \{i \in [1, M] | k_{i,t} = 1\}$, and $\bar{\mathcal{K}}_t = \{1, 2, 3, \ldots, M\} \setminus \mathcal{K}_t$ where the symbol $\setminus$ denotes the set subtraction. In this way, the "full" state vector has the same dimension, whether objects are present or not.

3 Dynamical Model

The dynamical model governs the evolution of the state between time steps. It is responsible for predicting the motion of people as well as modeling interpersonal interactions between the various people.

3.1 Joint Dynamical Model

The joint dynamical model for a variable number of people is defined as follows:

$$p(\tilde{\mathbf{X}}_t|\tilde{\mathbf{X}}_{t-1}) = p(\mathbf{X}_t, \mathbf{k}_t|\mathbf{X}_{t-1}, \mathbf{k}_{t-1}) = p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{k}_t, \mathbf{k}_{t-1}) p(\mathbf{k}_t|\mathbf{k}_{t-1}, \mathbf{X}_{t-1}) \tag{5}$$

$$\propto p_0(\mathbf{X}_t|\mathbf{k}_t) \left(\prod_{i=1}^{M} p(\mathbf{X}_{i,t}|\mathbf{X}_{t-1}, \mathbf{k}_t, \mathbf{k}_{t-1}) \right) p(\mathbf{k}_t|\mathbf{k}_{t-1}, \mathbf{X}_{t-1}) \tag{6}$$

with

$$p(\mathbf{X}_{i,t}|\mathbf{X}_{t-1},\mathbf{k}_t,\mathbf{k}_{t-1}) = \begin{cases} p(\mathbf{X}_{i,t}|\mathbf{X}_{i,t-1}) & \text{if } i \in \mathcal{K}_t \text{ and } i \in \mathcal{K}_{t-1}, \\ p_{birth}(\mathbf{X}_{i,t}) & \text{if } i \in \mathcal{K}_t \text{ and } i \notin \mathcal{K}_{t-1} \text{ (birth)}, \\ p_{death}(\mathbf{X}_{i,t}) & \text{if } i \notin \mathcal{K}_t \text{ and } i \in \mathcal{K}_{t-1} \text{ (death)}. \end{cases}$$

where we have assumed that targets which were born, died, or stayed behave independly of each other, and all the components are described below.

In the above, the term $p_0(\mathbf{X}_t|\mathbf{k}_t)$ models the interaction prior of multiple objects given current joint states, and the term $p(\mathbf{X}_{i,t}|\mathbf{X}_{i,t-1})$ denotes a single person dynamics, as discussed below. The term $p_{birth}(\mathbf{X}_{i,t})$ denotes a prior distribution over the state space of object values for a new born object i at time t. while $p_{death}(\mathbf{X}_{i,t})$ denotes a uniform probability over the state space of objects, for a dead object i at time t. Interestingly enough, these distributions are state-dependent, which allows to specify regions where the probability of creating or deleting objects is higher, typically near entrance and exit points [8]. In our current implementation, we used a uniform probability. The last term $p(\mathbf{k}_t|\mathbf{k}_{t-1},\mathbf{X}_{t-1})$ in Eq. (6) allows to define a prior over the number of objects which die and are born at a given time step, thus disfavoring for instance the deletion of an object and its replacement by a newly created object. It is defined as:

$$p(\mathbf{k}_t|\mathbf{k}_{t-1},\mathbf{X}_{t-1}) = p(\mathbf{k}_t|\mathbf{k}_{t-1}) = p(\mathcal{K}_t|\mathcal{K}_{t-1}) \propto (p_a)^{|\mathcal{K}_t \setminus \mathcal{K}_{t-1}|}(p_d)^{|\mathcal{K}_{t-1} \setminus \mathcal{K}_t|}, \quad (7)$$

where $|\mathcal{A}|$ denotes the size of the set $\mathcal{A}$. Here we assumed that the probabilities of new indices are independent of the past state values. p_a and p_d are priors which prevent from changing the set of valid indices. It acts not only as a prior on the change of number of objects, but also on the index values: a low value will avoid the deletion of an object and its replacement by a newly created object.

3.1.1 Shape Oriented and Person Avoidance Interactions Prior

Person interactions are modeled by the the term p_0 in Eq. (6) and defined by pairwise prior over the *joint* state space:

$$p_0(\mathbf{X}_t|\mathbf{k}_t) = \prod_{i,j \in \mathcal{K}_t, i \neq j} \phi(\mathbf{X}_{i,t},\mathbf{X}_{j,t}) \propto \exp\left\{-\lambda_g \sum_{i,j \in \mathcal{K}_t, i \neq j} g(\mathbf{X}_{i,t},\mathbf{X}_{j,t})\right\}, \quad (8)$$

where $g(\mathbf{X}_{i,t},\mathbf{X}_{j,t})$ is a penalty function. In papers [8, 14] which used such a prior, authors defined this penalty function based on the current 2D overlap between the object projections, or on the Euclidean distance between the two object centers, for instance, $g(\mathbf{X}_{i,t},\mathbf{X}_{j,t}) = \psi(\|\mathbf{u}_{i,t} - \mathbf{u}_{j,t}\|)$ where $\psi(x)$ denotes some function, e.g. $\psi(x) = |x|$ or $\psi(x) = |x|^2$. In our case, we can propose

two improvements: first, as people are not 'circular', we replaced the above Euclidean distance by a Mahalanobis distance:

$$g_p(\mathbf{X}_{i,t}, \mathbf{X}_{j,t}) = \psi\left(d_{m,i}(\mathbf{u}_{i,t} - \mathbf{u}_{j,t})\right) + \psi\left(d_{m,j}(\mathbf{u}_{i,t} - \mathbf{u}_{j,t})\right), \tag{9}$$

where $d_{m,i}$ (resp. $d_{m,j}$) is the Mahalanobis distance defined by the ellipsoid shape of the person i (resp. j). Qualitatively, this term favors the alignment of people orientation (of close by people) in contrast to people with perpendicular orientations. People following each other is a typical situation where this term could be useful.

Secondly, when people move, they usually look forward to *avoid collision* with other people. We thus introduced a prior as well on the state $\mathbf{X}^{pr}_{i,t+1}$ predicted from the current state value $\mathbf{X}_{i,t}$, by defining the penalty function as $g(\mathbf{X}_{i,t}, \mathbf{X}_{j,t}) = g_p(\mathbf{X}_{i,t}, \mathbf{X}_{j,t}) + g_p(\mathbf{X}^{pr}_{i,t+1}, \mathbf{X}^{pr}_{j,t+1})$. This term will thus prevent collision, not only when people are coming close, but also when people are moving together in the same direction.

3.2 Single Object Dynamical Model

The dynamical model of a single person is defined as:

$$p(\mathbf{X}_{i,t}|\mathbf{X}_{i,t-1}) = p(\mathbf{u}_{i,t}, \dot{\mathbf{u}}_{i,t}|\mathbf{u}_{i,t-1}, \dot{\mathbf{u}}_{i,t-1})p(h_{i,t}|h_{i,t-1})p(\alpha_{i,t}|\alpha_{i,t-1}, \dot{\mathbf{u}}_{i,t}), \tag{10}$$

where we have assumed that the evolution of state parameters is independent given the previous state values. In this equation, the height prior $p(h_{i,t}|h_{i,t-1})$ assumes a constant height model with a steady-state value, to avoid large deviations towards too high or small values. The body orientation dynamics $p(\alpha_{i,t}|\alpha_{i,t-1}, \dot{\mathbf{u}}_{i,t})$ is composed of two terms which favor temporal smoothness and orientation alignment with the walking speed (which depends on the speed magnitude) as we have described in the single person tracking algorithm [7].

In addition to prior terms which prevent invalid floor positions for people and reduce the likelihood of the state when the walking speed exceeds some predefined limit, the position/speed dynamics is defined by

$$\dot{\mathbf{u}}_t = \mathbf{A}\dot{\mathbf{u}}_{t-1} + \mathbf{B}\mathbf{w}_{1,t} \quad \text{and} \quad \mathbf{u}_t = \mathbf{u}_{t-1} + \tau\dot{\mathbf{u}}_t + \mathbf{C}(\mathbf{u}_{t-1})\mathbf{w}_{2,t} \tag{11}$$

where $\mathbf{w}_{q,t} = (w^{(x)}_{q,t}, w^{(y)}_{q,t})^\top$ is a Gaussian white noise random variable ($q = 1, 2$), and τ is the time step between two frames. First assume that $\mathbf{C}(\mathbf{u}_{t-1}) = 0$. In this case, Eq. (11) represents a typical auto-regressive model, i.e. a Langevin motion, with $\mathbf{A} = a\mathbf{I}$ and $\mathbf{B} = b\mathbf{I}$ ($\mathbf{I}$ denotes the 2×2 identity matrix) and where β accounts for the speed damping and $\bar{v}$ is the steady-state root-mean square speed.

Fig. 2. *Left images.* Due to depth effects, very similar positions in the first camera view corresponds to dramatically different image locations on the other view. *Right graph.* for the same scene, floor map of localization uncertainties, propagated from image localization uncertainties. In green, floor locations visible in both cameras. In blue/red, locations visible by only one camera.

3.2.1 Introducing 2D-to-3D Localization Uncertainties

In multi-view environments with small overlapping regions between views, and important depth scene effects with large image projection size variations of people within and across views (see Fig. 2), the Langevin motion is not enough to represent the state dynamics uncertainty. Fig. 2 illustrates a typical problem at view transitions: a person appearing at a small scale in a given view enters a second view. Observations from the first view are insufficient to accurately localize the person on the 3D ground plane. Thus, when the person enters the second view, the image projections obtained from the state prediction of the MCMC samples will often results in a mismatch with the actual localization of the person in the second view. This mismatch might be too high to be covered (in one time step) by the regular noise of the dynamical model. As a result, the algorithm may keep (for some time) the person track so that it is only visible in the first view, and create a second track to account for the person's presence in the second view. To solve this issue, we added the noise term $\mathbf{C}(\mathbf{u}_{t-1})\mathbf{w}_{2,t}$ on the location dynamics (see Eq. (11)), whose covariance magnitude and shape depend on the person location. The covariance of this noise, which models 2D-to-3D localization uncertainties, is obtained as follows. The assumed 2D Gaussian noises on the image localization of a person's feet from the different views are propagated to the 3D floor position using an Unscented Transform [31], and potentially merged for people positions visible from several cameras, leading to the pre-computed noise model illustrated in the right image of Fig.2. Qualitatively, this term guarantees that in the MCMC process, state samples drawn from the dynamics will actually spread the known uncertainty 3D regions, and those samples drawn by exploiting the human detectors will not be disregarded as being too unlikely according to the dynamics.

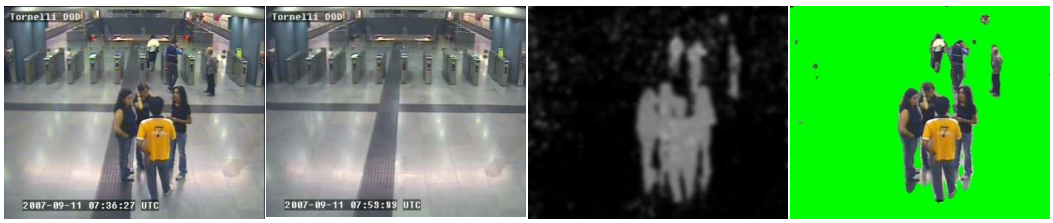


Fig. 3. An example of foreground detection on a real metro image with strong reflection on the ground floor. The second image shows the background learned from the first layer of the background subtraction algorithm.

4 Observation Model

When modeling $p(\mathbf{Z}_t|\tilde{\mathbf{X}}_t)$, which measures the likelihood of the observation $\mathbf{Z}_t$ for a given multi-object state configuration $\tilde{\mathbf{X}}_t$, it is crucial to be able to compare likelihoods when the number of objects is changing. Thus, we paid great care to propose a formulation that provides likelihoods of similar orders of magnitudes for different number of objects. For simplicity, we dropped the subscript t in this section. Our observations are defined as $\mathbf{Z} = (\mathbf{I}_v, \mathbf{D}_v)_{v=1..N_v}$, where $\mathbf{I}_v$ and $\mathbf{D}_v$ denotes the color and the background subtraction observations for each of the N_v camera views. More precisely, $\mathbf{D}_v$ is a background distance map obtained from the background subtraction of [32], with values between 0 and 1 where 0 means a perfect match with the background. Assuming the conditional independence of the camera views, we have:

$$p(\mathbf{Z}|\tilde{\mathbf{X}}) = \prod_{v=1}^{N_v} p(\mathbf{I}_v|\mathbf{D}_v, \tilde{\mathbf{X}}) p(\mathbf{D}_v|\tilde{\mathbf{X}}). \tag{12}$$

These two terms are described below (where we dropped the subscript v for simplicity).

4.1 Foreground Likelihood

The robust background subtraction technique described in [32] was used in this paper. In short, its main characteristics are the use of an approach similar to the Mixture of Gaussian (MoG) [33], the use of Local Binary Pattern features as well as a perceptual distance in the color space to avoid the detection of shadows, and the use of hysteresis values to model the temporal dynamics of the mixture weights. An example is shown in Fig. 3.

The foreground likelihood of one camera is modeled as:

$$p(\mathbf{D}|\tilde{\mathbf{X}}) = \prod_{\mathbf{x}\in S} \exp\left(-\lambda_{fg}(1 - \mathbf{D}(\mathbf{x}))\right) \prod_{\mathbf{x}\in\bar{S}} \exp\left(-\bar{\lambda}_{fg}\mathbf{D}(\mathbf{x})\right) \tag{13}$$

$$\propto \prod_{\mathbf{x}\in S} \exp\left(c_1(\mathbf{D}(\mathbf{x}) - c_2)\right) \tag{14}$$

where $\mathbf{x}$ denotes an image pixel, S denotes the object regions of the image,

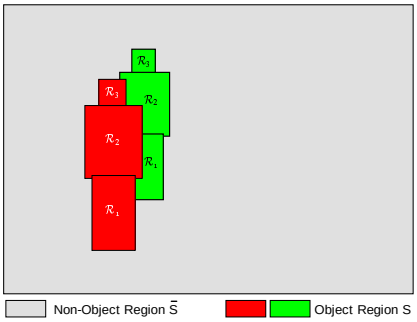


Fig. 4. Object and non-object regions used to compute observation likelihoods.

$\bar{S}$ denotes its complement, as illustrated in Fig. 4, and $c_1 = (\lambda_{fg} + \bar{\lambda}_{fg})$ and $c_2 = \lambda_{fg}/c_1$. In Eq. (13), we can clearly notice that the number of terms is independent of the number of objects. Equation (14), which was obtained by factoring out the constant term $\prod_{\mathbf{x}\in\bar{S}\cup S} \exp\left(-\bar{\lambda}_{fg}\mathbf{D}(\mathbf{x})\right)$, indicates that the placement (for track or birth) of objects will be encouraged in regions where $\mathbf{D}(\mathbf{x}) > c_2$.

4.2 The Color Likelihood

The color likelihood in one camera is modeled as:

$$\begin{aligned} p(\mathbf{I}|\mathbf{D},\tilde{\mathbf{X}}) &= \prod_{i\in\mathcal{K}} \prod_{b=1}^{3} \exp\left(-\lambda_{im}|\mathcal{R}_{i,b}|D_c(\mathbf{I},\mathbf{D},\mathcal{R}_{i,b})\right) \prod_{\mathbf{x}\in\bar{S}} \exp\left(-\lambda_{im}D_{min}\right) \\ &\propto \prod_{i\in\mathcal{K}} \prod_{b=1}^{3} \exp\left(-\lambda_{im}|\mathcal{R}_{i,b}|(D_c(\mathbf{I},\mathbf{D},\mathcal{R}_{i,b}) - D_{min})\right) \end{aligned} \tag{15}$$

where $\mathcal{R}_{i,b}$ denotes, for an existing object i visible in the camera view, the image part of its body region b which are not covered by other objects (see Fig. 4), and $|\mathcal{R}_{i,b}|$ denotes the area of $\mathcal{R}_{i,b}$. The above expression provides a comparable likelihood for different number of objects, and will favor the placement of tracked objects at positions for which the body region color distance $D_c(\mathbf{I},\mathbf{D},\mathcal{R}_{i,b})$ is high, and favor the object existence if this distance is (on average) higher than the expected minimum distance D_{min}.

4.2.1 Object Color Representation and Distance

From the visible part of the body region $\mathcal{R}_b$ of an object, we extract two color histograms: $\mathbf{h}_b$, which uses only foreground pixels (i.e. for which $\mathbf{D}(\mathbf{x}) > c_2$), and $\mathbf{H}_b$, which uses all pixels in $\mathcal{R}_b$. While the former should be more accurate by avoiding pooling pixels from the background, the latter one guarantees that we will have enough observations. To efficiently account for appearance variability due to pose, lighting, resolution and camera view changes, we propose to represent each object body region using a set of B automatically learned reference histograms, $\mathcal{H} = \{\bar{\mathbf{H}}_k\}_{k=1}^{B}$, learned as detailed below in Subsection

Algorithm 1 Multi-Modal Reference Histogram Modeling of a Human Body Part

Initialization: For a new born object selected from the camera view v in time t_0, we initialize as: $K_{t_0} = B_{t_0} = 1$, $\bar{\mathbf{H}}_{1,t_0} = (\mathbf{H}_{t_0} + \mathbf{h}_{t_0})/2$, and $w_{1,t_0} = w_0$ where w_0 denotes a low initial weight. $\mathbf{H}_{t_0}$ and $\mathbf{h}_{t_0}$ are the normalized histograms computed from all the pixels and all foreground pixels in the unoccluded region for the human body part observed at the current view, respectively.

Update: If the person object exists in the scene, we repeat:

For the camera view $v = 1$ to N_v: if the person object is fully visible in the current view (i.e. without occlusion) based on the mean state, we update the model (otherwise, we do not update it):

- Compute the normalized histograms $\mathbf{H}_t$ (for all the pixels) and $\mathbf{h}_t$ (only for foreground pixels) of the human body part observed at the current view.
- For $\mathbf{H} = \mathbf{H}_t$ and $\mathbf{H} = \mathbf{h}_t$, we repeat the following steps:
  - Compute the Bhattachayya distances $\{D_{bh}(\bar{\mathbf{H}}_{k,t}, \mathbf{H})\}_{k=1}^{K_t}$ and find the best matched mode $\tilde{k}$ with the smallest distance.
  - If the best matched mode is close enough to the data, i.e. $D_{bh}(\bar{\mathbf{H}}_{\tilde{k},t}, \mathbf{H}) < \theta_b$, we update with $K_t = K_{t-1}$:
    - *The best matched mode:* $\bar{\mathbf{H}}_{\tilde{k},t} = (1 - \alpha_h)\bar{\mathbf{H}}_{\tilde{k},t-1} + \alpha_h \mathbf{H}_t$ and $w_{\tilde{k},t} = (1 - \alpha_w)w_{\tilde{k},t-1} + \alpha_w$ where α_h and α_w are the learning rates.
    - *Other modes:* $w_{k,t} = (1 - \alpha_w)w_{k,t-1}$.

    Otherwise, we create a new mode $\{\mathbf{H}, w_0\}$, and add it (if $K_{t-1} < K_{\max}$) resulting in $K_t = K_{t-1} + 1$ or replace the existing mode with the smallest weight using it with $K_t = K_{t-1}$ (if $K_{t-1} = K_{\max}$).
  - Sort all the modes in decreasing order according to their corresponding weights.
  - Select the first B_t modes as the potential reference histogram modes, satisfying $\sum_{k=1}^{B_t} w_{k,t} / \sum_{k=1}^{K_t} w_{k,t} \geq T_h, T_h \in [0, 1]$, typically $T_h = 0.6$.

4.2.2. The color distance is then defined as:

$$D_c(\mathbf{I}, \mathbf{D}, \mathcal{R}_b) = (1 - \lambda_f) D_h^2(\mathbf{H}_b, \mathcal{H}) + \lambda_f D_h^2(\mathbf{h}_b, \mathcal{H}) \tag{16}$$

with

$$D_h(\mathbf{H}, \mathcal{H}) = \min_k D_{bh}\left(\mathbf{H}, \bar{\mathbf{H}}_k\right) \tag{17}$$

where D_{bh} denotes the standard Bhattacharyya distance between two histograms, and λ_f weights the contribution of each extracted histogram to the overall distance.

For a new born object, we do not have reference histograms compute the color distance defined in Eq. (16). Still, when creating an object, we need to be able to evaluate the color likelihood for those new born objects. Thus, at creation time, the initial reference histogram for each body part of the new born object is computed as the average of two currently extracted histograms $\mathbf{H}_b$ and $\mathbf{h}_b$, as described in Algorithm 1.

4.2.2 Multi-Modal Reference Histogram Modeling

Due to pose changes, lighting changes, non-rigid motions or multiple resolutions in multi-views, the histogram for each human body part may vary over time. To deal with this variance, we propose to utilize a multi-modal learning method to learn the statistical information about reference histogram for each human body part, which is similar to the used background modeling method [32] for foreground detection. Let $\mathcal{H}_t = \left\{K_t, \left\{\bar{\mathbf{H}}_{k,t}, w_{k,t}\right\}_{k=1}^{K_t}, B_t\right\}$ represent the learned statistical reference histogram model at time t for some human body part, which consists of a list of K_t reference histogram modes $\{\bar{\mathbf{H}}_{k,t}\}_{k=1}^{K_t}$ with weights $\{w_{k,t}\}_{k=1}^{K_t}$, of which the first $B_t(\leq K_t)$ modes have been identified as representing reference histogram observations, used in Eq. (17). To keep the complexity bounded, we set a maximal mode list size $K_{\max}$. The observed histograms (extracted from the object mean state at the end of each time step) are matched against the reference histograms and used to update the best matched histogram, or create a new reference histogram if the best match is not close enough. The whole flowchart for multi-modal reference histogram modeling algorithm is described in Algorithm 1.

5 Reversible-Jump MCMC

Given the high and variable dimensionality of our state space, the inference of the filtering distribution $p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t})$ is conducted using a Reversible-Jump MCMC (RJ-MCMC) sampling scheme which has been shown to be the very efficient in such cases [5,8,14]. In RJ-MCMC, a Markov Chain is defined such that its stationary distribution is equal to the target distribution, Eq. (2) in our case. The Markov Chain is sampled using the Metropolis-Hastings (MH) algorithm. Starting from an arbitrary configuration, the algorithm proceeds by repetitively selecting a move type m from a set of moves Υ with prior probability p_m and sampling a new configuration $\tilde{\mathbf{X}}'_t$ from a proposal distribution $q_m(\tilde{\mathbf{X}}'_t|\tilde{\mathbf{X}}_t)$. The move can either change the dimensionality of the state (as in birth or death) or keep it fixed. Then, either the proposed configuration is added with probability (known as *acceptance ratio*)

$$a = \min\left(1, \frac{p(\tilde{\mathbf{X}}'_t|\mathbf{Z}_{1:t})}{p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t})} \times \frac{p_{m'}}{p_m} \times \frac{q_{m'}(\tilde{\mathbf{X}}_t; \tilde{\mathbf{X}}'_t)}{q_m(\tilde{\mathbf{X}}'_t; \tilde{\mathbf{X}}_t)}\right) \tag{18}$$

to the Markov Chain, where m' is the reverse move of m, or the current configuration is added otherwise. In the following, we describe the moves and proposals we used and highlight the key points.

Fig. 5. Detection results at the same time instant in the 3 views.

5.1 Human Detection

Good and accurate automatic track initialization is crucial for multi-object tracking, in particular since it is the phase where the initial object model (color histograms) is extracted. In addition, being able to propose accurate positions to update current tracks is important. To this end, we have developed a human person detector [34] which builds on the approach of Tuzel *et al.* [35], and takes full advantage of the correlation existing between the shapes of humans in foreground detection maps and their appearance in the RGB images In multi-view calibrated environment, the detector was applied on each view separately, on windows i) which correspond to plausible people sizes; ii) for which the corresponding windows in the other camera views (obtained thanks to the calibration) all contained enough (20%) foreground pixels. Note that apart from this latter constraint, we did not try to merge the detection output in the different views. The main reason is that such fusion could reduce the number of detection (e.g. as the object might be too small, occluded or noisy in a given image). Also it appeared to be better to keep the best localizations in each of the camera views when initializing or updating track states in the MCMC tracking framework. Fig. 5 provides an example of obtained detections.

5.2 Move Proposals

In total, we define six move types: *add*, *delete*, *stay*, *leave*, *switch*, and *update*. The proposal of each move type is defined as follows. The first four corresponds to the typical moves and corresponding acceptance ratios which can be found in [14].

1) *add*: The human detector described above is used to find a set of persons $\mathcal{K}_t^{det} = \{\mathcal{K}_{v,t}^{det}\}_{v=1}^{N_v}$ in the scene at time t where the set $\mathcal{K}_{v,t}^{det}$ consists of all detected persons from the camera view v at time t. If a detected person i^* from one of camera views is not yet in the current existing object set $\mathcal{K}_t$, we propose adding it. The proposal of the *add* move type is defined as:

$$q_{add}(\tilde{\mathbf{X}}'_t; \tilde{\mathbf{X}}_t) = 1 \Big/ |\mathcal{K}_t^{det} \setminus \mathcal{K}_t|, \tag{19}$$

if $\tilde{\mathbf{X}}'_t$ contains the same objects as in $\tilde{\mathbf{X}}_t$, plus a detected object, and 0 otherwise

[1] . The symbol '\' denotes the set subtraction and $|\cdot|$ denotes the set size. Here we define $\mathcal{K}_t^{det} \setminus \mathcal{K}_t = \{\mathcal{K}_{v,t}^{det} \setminus \mathcal{K}_t\}_{v=1}^{N_v}$. One of the following two conditions must be satisfied to assess that a detected object i from the camera view v is not yet in the current existing object $\mathcal{K}_t$. The first one is that the minimal distance between the object i and all objects in $\mathcal{K}_t$ on the ground plane is larger than some threshold value (i.e. the detected object can not be associated with any existing object) The second one is that the percentage of occlusion in the camera view v by the existing objects is lower than some threshold value. If all detected objects have already been added, we set the probability p_{add} of the *add* proposal to zero.

2) *delete*: As required by the reversible-jump MCMC algorithm, the *add* proposal above needs to have a corresponding reverse jump defined, in order to potentially move the chain back to a previous hypothesis. We define the proposal of the *delete* move type by removing a randomly selected person i^* from the identifier set $\mathcal{K}_t^{det} \cap \mathcal{K}_t$, which is the set of detected objects that have already been added to $\mathcal{K}_t$, as:

$$q_{delete}(\tilde{\mathbf{X}}_t'; \tilde{\mathbf{X}}_t) = 1 \Big/ |\mathcal{K}_t^{det} \cap \mathcal{K}_t|, \tag{20}$$

where the operation $\cap$ denotes the intersection of two sets. If no detected objects were added yet (i.e. $\mathcal{K}_t = \phi$) or the intersection is empty, we set the probability p_{delete} of the *delete* proposal to zero.

3) *stay*: The *add*/*delete* proposals defined above enable new objects to enter or be removed from the field of view at each time step, driven by a human detector. Additionally, we need a mechanism for deciding on the fate of objects that were already represented in the previous sample set $\{\tilde{\mathbf{X}}_{t-1}^{(r)}\}_{r=1}^{N}$ at time $t-1$. We define a valid identifier set existing in the previous sample set as $\mathcal{K}_{t-1}^* \triangleq \{i \in [1, M] | \sum_{r=1}^{N} k_{i,t-1}^{(r)} > 0\}$, i.e. an object identifier is valid if it appeared in enough samples. If a given object i^* is no longer valid in the current sample state $\tilde{\mathbf{X}}_t$, i.e. $k_{i^*,t} = 0$ but exists in $\mathcal{K}_{t-1}^*$, we propose to re-add it with uniform probability $1 \Big/ |\mathcal{K}_{t-1}^* \setminus \mathcal{K}_t|$ and sample a new state from

$$q(\mathbf{X}_{i^*}; i^*) = \sum\nolimits_{r=1, s \in \mathcal{K}_{t-1}^{(r)}}^{N} p(\mathbf{X}_{i^*,t} | \mathbf{X}_{i^*,t-1}^{(r)}).$$

In this way, the *stay* proposal can be defined as:

$$q_{stay}(\tilde{\mathbf{X}}_t'; \tilde{\mathbf{X}}_t) = \Big(1 \Big/ |\mathcal{K}_{t-1}^* \setminus \mathcal{K}_t|\Big) q(\mathbf{X}_{i^*}; i^*). \tag{21}$$

If the set $\mathcal{K}_{t-1}^* \setminus \mathcal{K}_t$ is empty, we set the probability p_{stay} of the *stay* proposal to zero.

4) *leave*: The corresponding reverse jump of the move *stay* randomly selects

[1] Note that the cases where the probability is 0 are implicit in the way the move is defined. For other moves, we will not mention it

an identifier i^* from the set $\mathcal{K}_t \setminus \mathcal{K}_t^{det}$ and removes it from the current sample state. Thus, the *leave* proposal can be defined as:

$$q_{leave}(\tilde{\mathbf{X}}'_t; \tilde{\mathbf{X}}_t) = 1 \Big/ |\mathcal{K}_t \setminus \mathcal{K}_t^{det}|, \tag{22}$$

If the set $\mathcal{K}_t \setminus \mathcal{K}_t^{det}$ is empty, we set the probability p_{leave} of the *leave* proposal to zero.

5) *switch*: This move allows to randomly select a pair of close-by objects $(i^*, j^*) \in \mathcal{K}_t$ (i.e. $k_{i^*,t} = k_{j^*,t} = 1$) and exchange their states. In practice it allows to check whether the exchange of color models better fits the data. According to the Mahalanobis distance between two objects used in our interactions prior (see Section 3.1.1), we define the *switch* proposal as:

$$q_{switch}(\tilde{\mathbf{X}}'_t; \tilde{\mathbf{X}}_t) = 1/g_p(\mathbf{X}_{i^*,t}, \mathbf{X}_{j^*,t}) \tag{23}$$

Thus, there are a larger probability to select a pair of objects with smaller distance. If the Mahalanobis distance of the pair of objects (i^*, j^*) is large enough, i.e., $g_p(\mathbf{X}_{i^*,t}, \mathbf{X}_{j^*,t}) > \theta_{md}$, we set the probability p_{switch} of the *switch* proposal to zero.

6) *update*: This is an important move which allows to find good estimates for the object states without changing dimension. It works by first randomly selecting a valid object i^* from the current joint configuration (i.e. for which $k_{i^*,t} = 1$), and then propose a new state for update. This new state is drawn in two ways (i.e the proposal is a mixture). In the first case, the object position, height and orientation are locally perturbed according to a Gaussian kernel [14]. Importantly, in order to propose interesting state values that may have a visual impact, the noise covariance in position is defined as $\Sigma(\mathbf{u}_{i^*}) = C(\mathbf{u}_{i^*})C^\top(\mathbf{u}_{i^*})$, where $C(\mathbf{u}_{i^*})$ is the noise matrix in Eq. (11) which is used to define the noise covariance in Fig. 2. The proposal in this case is mathematically defined as (for the selected object, the other objects remain unchanged):

$$q_{u_1}(\mathbf{X}'_{i^*,t}; \mathbf{X}_{i^*,t}) = \mathcal{N}(\mathbf{X}'_{i^*,t}; \mathbf{X}'_{i^*,t}, \Sigma_u) \tag{24}$$

where $\Sigma_u = diag(\Sigma(\mathbf{u}_{i^*}), \sigma_h^2, \sigma_\alpha^2)$ where σ_h^2 and σ_α^2 are noise variances in height and orientation, respectively. The second way is to update the object location by sampling the new location around one of the positions provided by the human detector which are close enough from the selected object i^*. Here again, closeness is defined by exploiting $\Sigma(\mathbf{u}_{i^*})$, and the perturbation covariance around the selected detection is given by $\Sigma(\mathbf{u}_{i^*})$. Thus, the proposal in this case can be defined as:

$$q_{u_2}(\mathbf{X}'_{i^*,t}; \mathbf{X}_{i^*,t}) = \frac{1}{C_d} \sum_{k \in \mathcal{K}_{i^*,t}} \frac{\mathcal{N}(\mathbf{u}_{i^*,t}; \mathbf{u}_{k,t}, \Sigma(\mathbf{u}_{i^*}))}{\psi(d_{m,i}(\mathbf{u}_{i^*,t}, \mathbf{u}_{k,t}))} \tag{25}$$

where C_d is a normalization factor, i.e. $C_d = \sum_{k \in \mathcal{K}_{i^*,t}} 1 \Big/ \psi(d_{m,i}(\mathbf{u}_{i^*,t}, \mathbf{u}_{k,t}))$,

and $d_{m,i}(\mathbf{u}_{i^*,t}, \mathbf{u}_{k,t})$ denotes the the Mahalanobis distance used in Eq. (9) for interactions prior. The set $\mathcal{K}_{i^*,t}$ consists of the human detected objects which are close enough to the object i^*, i.e. $\mathcal{K}_{i^*,t} = \{k \in \mathcal{K}_t^{det} | d_{m,i}(\mathbf{u}_{i^*,t}, \mathbf{u}_{k,t}) \neq \theta_{md}\}$ where θ_{md} is a predefined distance threshold. So the final update proposal will be defined as:

$$q_{update}(\tilde{\mathbf{X}}'_t; \tilde{\mathbf{X}}_t) = \frac{1}{|\mathcal{K}_t|}\Big(\lambda_u q_{u_1}(\mathbf{X}'_{i^*,t}; \mathbf{X}_{i^*,t}) + (1-\lambda_u) q_{u_2}(\mathbf{X}'_{i^*,t}; \mathbf{X}_{i^*,t})\Big) \quad (26)$$

where λ_u a real value, $\lambda_u \in [0, 1]$.

5.3 Summary

The steps of the proposed MCMC-based tracking algorithm for a variable number of objects existing in the scene are summarized in the Algorithm 2. Importantly, note that while the overall expression of the filtering distribution is quite complex, the expression of the acceptance ratio is usually simpler. Indeed, many of the terms at the numerator and the denominator cancel each other, since the likelihood terms as well as the interaction terms only involve local computation, and most of the objects don't change at each move.

6 Experiments

6.1 Calibration and slant removal

Before processing the videos, cameras were first calibrated, and a rectification homography was precomputed in order to remove people slant in images at run-time.

Camera calibration. Cameras were calibrated using the available information and exploiting geometrical constraints [36], like 3D lines should appear as undistorted, or vertical direction Z is obtained from the image coordinates of the vertical vanishing point $\mathbf{v}_\perp$, computed as the intersection of the image projections of a set of 3D world parallel vertical lines. The image-to-ground homography $\mathbf{H}$ was estimated using a set of manually marked points in the image plane and their 3D correspondences in the 3D ground plane.

Removing Slant by Mapping the Vertical Vanishing Point to Infinity. In Fig. 6, we observe that standing people appear with different slants in the image. This introduces variability in the feature extraction process when using rectangular regions. To handle this issue, we compute an appropriate projective transformation $\mathbf{H}_\perp$ of the image plane in order to map its vertical finite vanishing point to a point at infinity, as described in [37]. As a result, the 3D vertical direction of persons standing on the ground plane will always

Algorithm 2 Multi-Person Tracking with Reversible-Jump MCMC

At each time step t, the posterior over joint object states $\tilde{\mathbf{X}}_{t-1}$ at time $t-1$ is represented by a set of unweighted samples $\{\tilde{\mathbf{X}}_{t-1}^{(r)}\}_{r=1}^{N}$. The approximation of the current distribution $p(\tilde{\mathbf{X}}_t|\mathbf{Z}_t)$ is constructed by RJ-MCMC sampling as follows:

1) **Initialization**: Initialize the Markov Chain by randomly selecting a sample $\tilde{\mathbf{X}}_{t-1}^{(r)}$ and apply the motion model to each object, and accept it as the first sample.

2) **RJ-MCMC Sampling**: Draw $(B+N)$ samples according to the following schedule, where B is the length of the burn-in period:
   · Randomly select a move type from the set of moves $\Upsilon = \{add, delete, stay, leave, switch, update\}$ (for details, see 5.2).
   · Select an object i^* (or two objects i^* and j^* for *switch*).
   · Propose a new state $\tilde{\mathbf{X}}_t'$ depending on the randomly selected proposal type
   add: add a new object i^* to the current state.
   delete: delete an existing object i^* from the current state.
   stay: re-add an existing object i^* to the current state.
   leave: remove an existing object i^* from the current state.
   switch: exchange the states of two close-by objects (i^*, j^*).
   update: update the parameters of object i^*.
   · Compute the acceptance ratio a defined in Eq. (18) for the chosen move type.
   · If $a \geq 1$ then accept the proposed state $\tilde{\mathbf{X}}_t \leftarrow \tilde{\mathbf{X}}_t'$ as a new sample. Otherwise, we accept it with probability a, and reject it otherwise.

3) **Approximation**: As an approximation to the current posterior $p(\tilde{\mathbf{X}}_t|\mathbf{Z}_t)$, return the new sample set $\{\tilde{\mathbf{X}}_t^{(r)}\}_{r=1}^{N}$ obtained after discarding the initial B burn-in samples.

4) **Update Reference Histograms**: Compute the mean states of all existing objects from the new sample set and update their corresponding reference histograms (for details, see 4.2.2).

map to 2D vertical lines in the new image, as illustrated in Fig. 6. This transformation should thus help in obtaining better detection results or extracting more accurate features while still keeping the computation efficiency, e.g. by using integral images. At runtime, this does not generate an extra-cost since this mapping can be directly integrated with the distorsion removal step.

6.2 Results

Two datasets captured from two different scenes were used to evaluate our proposed multi-person tracking system. The first one consists of three 2h30 minutes video footage captured by three wide-baseline cameras in the Torino metro station scene as shown in Fig. 5. These sequences are very challenging, due to the camera view points (small average people size and large people size variations in a given view, occlusion, partial field of view overlap), crowded scenes in front of the gates, and the presence of many specular reflections

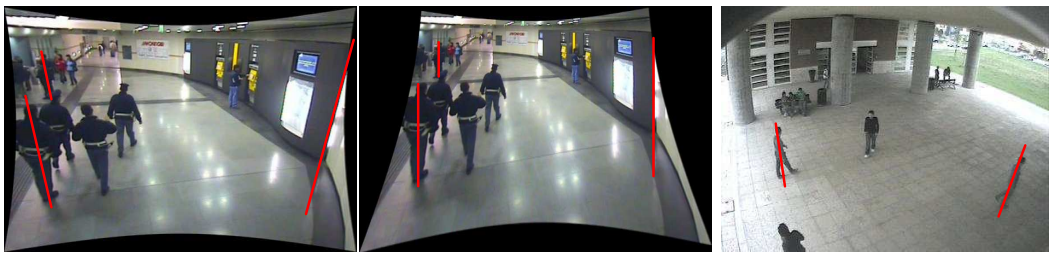


Fig. 6. Vertical vanishing point mapping. *Left:* after distorsion removal and before the mapping. We can observe people slant according to their position. *Central:* after the mapping to the infinity. Bounding-boxes will fit more closely the silhouette of people. *Right:* another example.

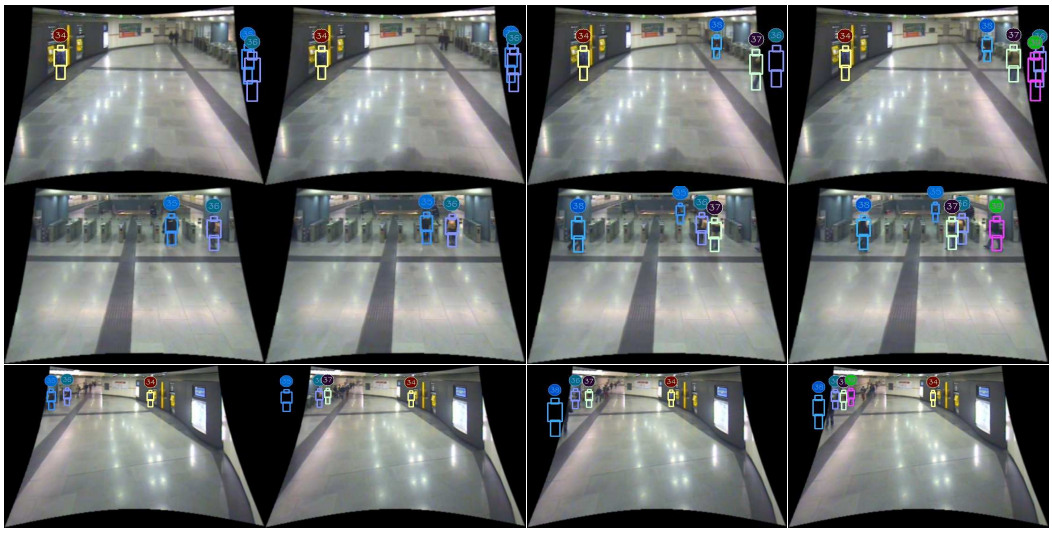


Fig. 7. Tracking results on the metro scene.

on the ground which in combination with cast shadows generate many background subtraction false alarms. In addition, most people are dressed with similar colors. The second dataset comprises 10 minutes of video footage also captured by three wide-baseline cameras in an outdoor scene. In this scene, people often appear slanted in the left or/and right borders of an image (see Fig. 6). The camera view point issues mentioned above for the metro scene also exist in this outdoor scene. The following experiments were obtained using a total of 1500 moves in the RJ-MCMC sampling with 500 in the burn-in phase.

Fig. 7 shows some tracking results on the first dataset. In this example,our tracking system performed very well, successfully adding people using the human detector mediated birth move, and efficiently handling inter-person occlusion and partial visibility between camera views.

The benefit of using the 2D-to-3D ground plane noise in our algorithm, and especially in the dynamics, is illustrated in a simple example, Fig. 8. In the first two rows, this component was not used, i.e. $\mathbf{C}(\mathbf{u}) = 0$ in Eq. (11). As can be seen, the estimated state from the first view lags a little bit behind,

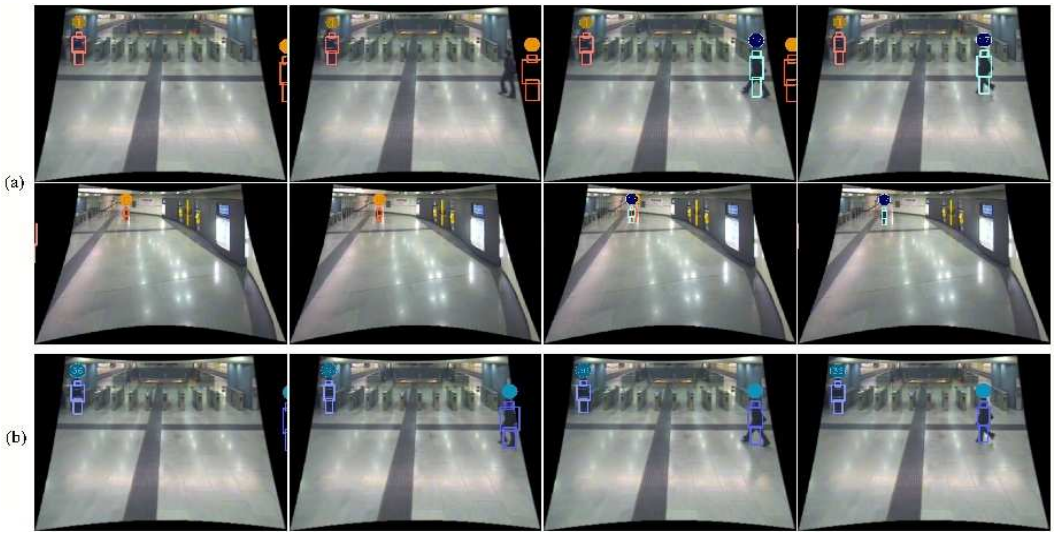


Fig. 8. Tracking results on the metro scene: (a) without integrating the ground plane noise model in dynamical model; (b) with integration.

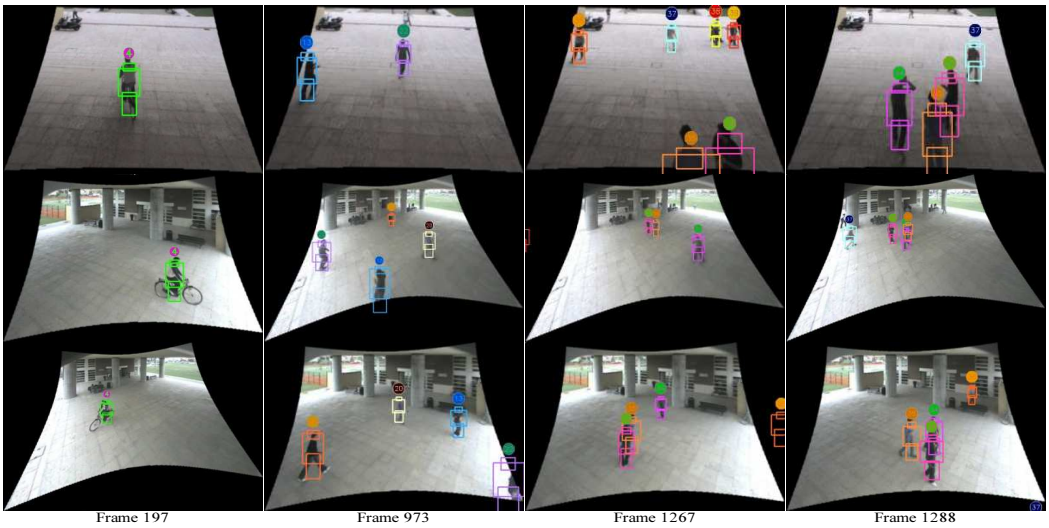


Fig. 9. Example of multi-person tracking on the outdoor sequence.

resulting in a mismatch when the tracked person enters the second view. As a consequence, a new object is created. The first track stays for some time, and is then removed, resulting altogether in a track break. On the other hand, when using the proposed term, the transition between cameras is successfully handled by the algorithm, as shown in Fig. 8(b).

On the second dataset, our approach performed very well, with almost no tracking errors in the 10-minute sequences. Results on four frames are shown in Fig. 9. Anectodically, our human detector was able to successfully detect a person on a bicycle and our tracking system was able to track him/it robustly.

7 Conclusions

In this chapter, we have discussed the general multi-person tracking issues, and presented a state-of-the-art multi-camera 3D tracking algorithm. The strength of the algorithm relies on several key factors: the joint multi-state Bayesian formulation, appropriate interaction models using state-prediction to model collision avoidance, the RJ-MCMC inference sampling scheme, and well balanced observation models. The use of a fast and powerful human detector proved to be essential for good track initialization and state update. In the same way, the use of predefined 2D to 3D geometric uncertainty measures on the state dynamics did improve the results, and removing the slant of people in the image through a simple rectification scheme allowed the use of efficient human detector and feature extraction based on integral images.
There are several ways to improve the current algorithm. The first one is to use longer term constraints on the dynamics. One standard approach is to do post-processing of the extracted trajectory, to remove transient tracks or resolve identity switches and merge trajectory fragments by observing the data on a longer time window. A second avenue of research for all tracking algorithms is to define more accurate likelihoods, especially in the presence of occlusions. There are two aspects to this issue. The first one is to use more sophisticated descriptions of the object, allowing to better explain the image content and resulting in better localization information. This can be done by including shape cues in the model, or representing objects through their parts. The second related aspect is to find appropriate measurements and likelihood models to infer the presence of the objects given its model. This should be robust enough to the noise inherent in the data. Given the large variety of scenes, people appearances or poses, illumination conditions, camera set-ups, image resolutions, there will not be a unique solution to this problem.

8 Acknowledgement

This work was supported by the European Union 6th FWP Information Society Technologies CARETAKER project (Content Analysis and Retrieval Technologies Applied to Knowledge Extraction of Massive Recordings, FP6-027231).

References

[1] R. Singh, P. Bhargava, S. Kain, State of the art smart spaces: application models and software infrastructure, Ubiquity 37 (7) (2006) 2–9.

[2] K. Bernardin, T. Gehrig, R. Stiefelhagen, Multi- and single view multiperson tracking for smart room environments, in: Proc. Workshop on Classification of Events, Actions and Relations (CLEAR), 2006.

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

[4] K. Bernardin, R. Stiefelhagen, Audio-visual multi-person tracking and identification for smart environments, in: MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia, 2007.

[5] T. Zhao, R. Nevatia, Tracking multiple humans in crowded environment, in: Proc. IEEE CVPR, Washington DC, 2004.

[6] T. Zhao, R. Nevatia, B. Wu, Segmentation and tracking of multiple humans in crowded environments, IEEE Trans. Pattern Anal. Machine Intell. 30 (7) (2008) 1198–1211.

[7] J. Yao, J.-M. Odobez, Multi-camera 3d person tracking with particle filter in a surveillance environment, in: The 16-th European Signal Processing Conference (EUSIPCO-2008), 2008.

[8] K. Smith, D. Gatica-Perez, J.-M. Odobez, Using particles to track varying numbers of interacting people, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), San Diego, 2005.

[9] B. Wu, R. Nevatia, Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors, Int. Journal of Comp. Vision 72 (2) (2007) 247–266.

[10] M. Isard, J. MacCormick, BRAMBLE: A Bayesian multi-blob tracker, in: Proc. IEEE ICCV, Vancouver, 2001.

[11] C. J. Veenman, M. J. T. Reinders, E. Backer, Resolving motion correspondence for densely moving points, IEEE Trans. Pattern Anal. Machine Intell. 23 (1) (2001) 54–72.

[12] J. Sullivan, S. Carlsson, Tracking and labelling of interacting multiple targets, in: Europe Conf. Comp. Vision (ECCV), Vol. 3953 of Lecture Notes in Computer Science, 2006, pp. 619–632.

[13] B. Bose, X. Wang, E. Grimson, Multi-class object tracking algorithm that handles fragmentation and grouping, in: IEEE Conf. Comp. Vision & Pattern Recognition (CVPR), 2007.

[14] Z. Khan, T. Balch, F. Dellaert, Mcmc-based particle filtering for tracking a variable number of interacting targets, IEEE Trans. Pattern Anal. Machine Intell. 27 (2005) 1805–1819.

[15] W. Qu, D. Schonfeld, M. Mohamed, Distributed bayesian multiple target tracking in crowded environments using multiple collaborative cameras, EURASIP Journal on Applied Signal Processing, Special Issue on Tracking in Video Sequences of Crowded Scenes.

[16] D. Tweed, A. Calway, Tracking Many Objects using Subordinate Condensation, in: Proc. BMVC, Cardiff, 2002.

[17] K. Okuma, A. Taleghani, N. Freitas, J. Little, D. Lowe, A boosted particle filter: multi-target detection and tracking, in: Proc. European Conference on Computer Vision (ECCV), Prague, 2004.

[18] J. Berclaz, F. Fleuret, P. Fua, Robust people tracking with global trajectory optimization, in: IEEE Conf. Comp. Vision & Pattern Recognition (CVPR), Vol. 1, 2006, pp. 744–750.

[19] P. J. Green, Trans-dimensional Markov chain Monte Carlo, in: P. J. Green, N. L. Hjort, S. Richardson (Eds.), Highly Structured Stochastic Systems, Oxford Univ. Press, 2003.

[20] M. Isard, A. Blake, Condensationconditional density propagation for visual tracking, Int. Journal of Comp. Vision 29 (1) (1998) 5–28.

[21] D. Beymer, K. Konolige, Real-time tracking of multiple people using continuous detection, in: IEEE International Conference on Computer Vision (ICCV) Frame-Rate Workshop, 1999.

[22] K. Smith, S. O. Ba, J.-M. Odobez, D. Gatica-Perez, Tracking the visual focus of attention for a varying number of wandering people, IEEE Trans. Pattern Anal. Machine Intell. 30 (7) (2008) 1212–1229.

[23] G. Antonini, S. V. Martinez, M. Bierlaire, J. P. Thiran, Behavioral priors for detection and tracking of pedestrians in video sequences, Int. Journal of Comp. Vision 69 (2) (2006) 159–180.

[24] K. Kim, L. S. Davis, Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering, in: Europe Conf. Comp. Vision (ECCV), Vol. 3953 of Lecture Notes in Computer Science, 2006, pp. 98–109.

[25] W. Du, J. Piater, Multi-camera people tracking by collaborative particle filters and principal axis-based integration, in: Asian Conf on Comp. Vision (ACCV), 2007.

[26] A. Mittal, L. S. Davis, M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo, in: Europe Conf. Comp. Vision (ECCV), 2002.

[27] P. Perez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: Proc. European Conference on Computer Vision (ECCV), Copenhagen, 2002.

[28] I. Haritaoglu, D. Harwood, L. Davis, W4: real-time surveillance of people and their activities, IEEE Trans. Pattern Anal. Machine Intell. 22 (8) (2000) 809–830.

[29] B. Leibe, K. Schindler, L. V. Gool, Coupled detection and trajectory estimation for multi-object tracking, in: International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brasil, 2007.

[30] N. T. Pham, W. Huang, S. H. Ong, Probability hypothesis density approach for multi-camera multi-object tracking, in: Asian Conf on Comp. Vision (ACCV), 2007.

[31] S. Julier, J. Uhlmann, Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations, in: Proceedings of the 2002 American Control Conference, Vol. 2, 2002, p. 887892.

[32] J. Yao, J.-M. Odobez, Multi-layer background subtraction based on color and texture, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Workshop on Visual Surveillance (CVPR-VS), 2007, pp. 1–8.

[33] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE Conf. Comp. Vision & Pattern Recognition (CVPR), Vol. 2, 1999, pp. 246–252.

[34] J. Yao, J.-M. Odobez, Fast human detection from videos using covariance features, in: ECCV 2008 Workshop on Visual Surveillance (VS2008), 2008.

[35] O. Tuzel, F. Porikli, P. Meer, Human detection via classification on riemannian manifolds, in: IEEE Conf. Comp. Vision & Pattern Recognition (CVPR), 2007.

[36] G. Wang, Z. Hu, F. Wu, H.-T. Tsui, Single view metrology from scene constraints, Image & Vision Computing Journal 23 (2005) 831–840.

[37] J. Yao, J.-M. Odobez, Multi-camera multi-person 3d space tracking with mcmc in surveillance scenarios, in: ECCV 2008 Workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2), 2008.