

Multi-Target Tracking and Occlusion Handling With Learned Variational Bayesian Clusters and a Social Force Model

Ata Ur-Rehman, Syed Mohsen Naqvi, *Senior Member, IEEE*, Lyudmila Mihaylova, *Senior Member, IEEE*, and Jonathon A. Chambers, *Fellow, IEEE*

Abstract—This paper considers the problem of multiple human target tracking in a sequence of video data. A solution is proposed which is able to deal with the challenges of a varying number of targets, interactions, and when every target gives rise to multiple measurements. The developed novel algorithm comprises variational Bayesian clustering combined with a social force model, integrated within a particle filter with an enhanced prediction step. It performs measurement-to-target association by automatically detecting the measurement relevance. The performance of the developed algorithm is evaluated over several sequences from publicly available data sets: AV16.3, CAVIAR, and PETS2006, which demonstrates that the proposed algorithm successfully initializes and tracks a variable number of targets in the presence of complex occlusions. A comparison with state-of-the-art techniques due to Khan *et al.*, Laet *et al.*, and Czyn *et al.* shows improved tracking performance.

Index Terms—Clustering, data association, multi-target tracking, occlusion.

I. INTRODUCTION

A. Motivation

THIS paper presents a robust multiple tracking algorithm which can be used to track a variable number of people moving in a room or enclosed environment. Hence, the estimation of the unknown number of targets and their states is considered in each video frame. Multi-target tracking has many applications, such as surveillance, intelligent transportation, behavior analysis and human computer interfaces [1]–[3]. It is a challenging problem and a wealth of research has been undertaken to provide more efficient solutions [1], [4], [5]. Two of

the major challenges associated with visual tracking of multiple targets are: 1) mitigating occlusions and 2) handling a variable number of targets. Therefore, the focus of this paper is to present a technique to robustly handle these challenges.

Overcoming occlusions is a difficult task because during an occlusion it is not possible to receive measurements originating from the occluded objects. Significant research effort has been made to address this problem, e.g., [1], [4], [6], [7].

Many 2-D tracking systems rely on appearance models of people. For instance [8] applies the kernel density approach while [9] is based on color histograms, gradients and texture models to track human objects. These techniques use template matching which is not robust to occlusions because the occluded parts of targets cannot be matched. They perform an exhaustive search for the desired target in the whole video frame which obviously requires much processing time. These techniques are also sensitive to illumination changes.

Occlusions can be handled with the help of efficient association of available data to the targets. Most of the existing multiple target tracking algorithms assume that the targets generate one measurement at a time [1], [10]. In the case of human tracking from video and many other tracking applications, targets may generate more than one measurement [4], [11]. To exploit multiple measurements, most algorithms add a preprocessing step which involves extracting features from the measurements [9]. This preprocessing step solves the problem to some extent but it results in information loss due to the dimensionality reduction and hence generally degrades the tracking results. Another problem with many of these algorithms is that they adopt the hard assignment technique [9], [12] wherein likelihood models are employed to calculate the probability of the measurements.

Recently, in [4] an approach relying on clustering and a joint probabilistic data association filter (JPDAF) was proposed to overcome occlusions. Rather than extracting features from the measurements, this approach groups the measurements into clusters and then assigns clusters to respective targets. This approach is attractive in the sense that it prevents information loss but it fails to provide a robust solution to mitigate the occlusion problem. This is because it only utilizes the location information of targets and hence the identity of individual targets is not maintained over the tracking period. As a result, the tracker may confuse two targets in close interaction, eventually causing a tracker switching problem. Other advances of such filtering approaches to multiple target tracking include multiple detection JPDAF (MD-JPDAF) [13] and interacting multiple model JPDAF (IMM-JPDAF) [14].

Manuscript received September 12, 2014; revised March 10, 2015, June 20, 2015, and November 04, 2015; accepted November 14, 2015. Date of publication November 30, 2015; date of current version February 05, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ruixin Niu. This work was supported in part by U.K. EPSRC under Grant EP/K021516/1 and in part by Bayesian Tracking and Reasoning over Time (BTaRoT).

A. Ur-Rehman and L. Mihaylova are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: a.ur-rehman@sheffield.ac.uk; L.S.Mihaylova@sheffield.ac.uk).

S. M. Naqvi and J. A. Chambers are with the School of Electrical and Electronic Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K. (e-mail: Mohsen.Naqvi@newcastle.ac.uk; Jonathon.Chambers@newcastle.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2015.2504340

Interaction models have been proposed in the literature to mitigate the occlusion problem [1], [15]. The interaction model presented in [1] exploits a Markov random field approach to penalize the predicted state of particles which may cause occlusions. This approach works well for tracking multiple targets where two or more of them do not occupy the same space, but it does not address tracking failures caused by inter-target occlusions.

B. Dynamic Motion Models

Dynamic models can be divided broadly into macroscopic and microscopic models [16]. Macroscopic models focus on the dynamics of a collection of targets. Microscopic models deal with the dynamics of individual targets by taking into account the behavior of every single target and how they react to the movement of other targets and static obstacles.

A representative of microscopic models is the social force model [17]. The social force model can be used to predict the motion of every individual target, i.e., the driving forces which guide the target towards a certain goal and the repulsive forces from other targets and static obstacles. A modified social force model is used in [18] and in [19] for modeling the movements of people in non-observed areas.

Many multi-target tracking algorithms in the literature only consider the case when the number of targets is known and fixed [3], [12], [15], [20]. In [1] a reversible jump Markov chain Monte Carlo (RJMCMC) sampling technique is described but in the experimental results a very strong assumption is made that the targets (ants) are restricted to enter or leave from a very small region (nest site) in a video frame. Random finite set (RFS) theory techniques have been proposed for tracking multiple targets, e.g., [21], [22], in particular the target states are considered as an RFS.

An appealing approach where the number of targets is estimated, is proposed in [4] and is based on the evaluated number of clusters. Non-rigid bodies such as humans can, however, produce multiple clusters per target. Therefore, the number of clusters does not always remain equal to the number of targets. Hence, calculating the number of targets on the basis of the number of clusters can be inaccurate in the case of human target tracking.

C. Main Contributions

The main contributions of this work are:

- 1) An algorithm that provides a solution to multiple target tracking and complex inter-target interactions and occlusions. Dealing with occlusions is based on social forces between targets.
- 2) An improved data association technique which clusters the measurements and then uses their locations and features for accurate target identification.
- 3) A new technique based on the estimated positions of targets, size and location of the clusters, geometry of the monitored area, along with a death and birth concept to handle robustly the variable number of targets.

The framework proposed in [4] uses the JPDAF together with the variational Bayesian clustering technique for data association and a simple dynamic model for state transition. Our proposed technique differs from that of [4] in three main aspects: 1) in the dynamic model which is based on social forces between

targets, 2) a new features based data association technique, and 3) a new technique for estimation of number of targets which is based on the information from clusters, position of existing targets and geometry of the monitored area.

The remainder of the paper is organized as follows: Section II describes the proposed algorithm, Section III presents the proposed data association algorithm, Section IV explains the clustering process and Section V describes the proposed technique for estimating the number of targets. Extensive experimental validation is given in Section VI. Section VII contains discussion and finally, conclusions are presented in Section VIII.

II. THE PROPOSED ALGORITHM

A. Bayesian Estimation

The goal of the multi-target tracking process is to track the state of N unknown targets. The state of each target is represented as \mathbf{x}_k^i , ($i = 1, \dots, N$), which is a column vector containing the position and velocity information of the i^{th} target at time k . The joint state of all targets is constructed as the concatenation of the individual target states $\mathbf{x}_k = [(\mathbf{x}_k^1)^T, \dots, (\mathbf{x}_k^i)^T, \dots, (\mathbf{x}_k^N)^T]^T$, where $(\cdot)^T$ denotes the transpose operator. Measurements at time k are represented as $\mathbf{y}_k = [(\mathbf{y}_k^1)^T, \dots, (\mathbf{y}_k^j)^T, \dots, (\mathbf{y}_k^L)^T]^T$, where \mathbf{y}_k^j is described in Section II-B.

Within the Bayesian framework, the tracking problem consists of estimating the belief of the state vector \mathbf{x}_k given the measurement vector $\mathbf{y}_{1:k}$. The objective is to sequentially estimate the posterior probability distribution $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ at every time step k . The posterior state distribution $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ can be estimated in two steps. First, the prediction step [23] is

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}, \quad (1)$$

where $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the state transition probability. After the arrival of the latest measurements, the update step becomes

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}, \quad (2)$$

where $p(\mathbf{y}_k | \mathbf{x}_k)$ is the measurement likelihood function.

The two step tracking process yields a closed form expression only for linear and Gaussian dynamic models [10]. Particle filtering [23] is a popular solution for suboptimal estimation of the posterior distribution $p(\mathbf{x}_k | \mathbf{y}_{1:k})$, especially in the nonlinear and non-Gaussian case. The posterior state distribution is estimated by a set of random samples \mathbf{x}_k^s , $s = 1, \dots, N_s$ and their associated weights w_k^s at time k

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx \sum_{s=1}^{N_s} w_k^s \delta(\mathbf{x}_k - \mathbf{x}_k^s), \quad (3)$$

where N_s is the total number of particles and $\delta(\cdot)$ denotes a multivariate Dirac delta function.

We apply particle filtering within the JPDAF framework to estimate the states of targets. The JPDAF recursively updates the marginal posterior distribution $p(\mathbf{x}_k^i | \mathbf{y}_{1:k})$ for each target. In our work, given the state vector \mathbf{x}_{k-1}^i of target i , the next set of particles at time step k is predicted by the social force dynamic model ((6)–(9)) which is described later in Section II-C. The state transition model is represented by (9). The N_s particles

and their corresponding weights can approximate the marginal posterior at time step k .

To assign weights to each sample we follow the likelihood model along with the data association framework which is explained in Section III.

B. Measurement Model

The algorithm considers L number of pixels in a single video frame captured by one camera as input measurements. From the sequentially arriving video frames, the silhouette of the moving targets (foreground) is obtained by background subtraction [24]. At time k , the measurement vector is: $\mathbf{y}_k = [(\mathbf{y}_k^1)^T, \dots, (\mathbf{y}_k^j)^T, \dots, (\mathbf{y}_k^L)^T]^T$. The state \mathbf{x}_k at time k is estimated by using the measurement vector

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{e}_{m,k}), \quad (4)$$

where $\mathbf{e}_{m,k}$ is the measurement noise vector. After the background subtraction [24], the measurements at time k are: $\tilde{\mathbf{y}}_k = [(\tilde{\mathbf{y}}_k^1)^T, \dots, (\tilde{\mathbf{y}}_k^j)^T, \dots, (\tilde{\mathbf{y}}_k^M)^T]^T$, which are assumed to correspond to the moving targets (please see Section VI for details). The measurement function $h_k(\cdot)$ is represented by some features of the video frames, e.g., the color histogram. The measurement vector $\tilde{\mathbf{y}}_k$ contains only a set of foreground pixels. After background subtraction the number of pixels with non-zero intensity values is reduced to M , where $M \ll L$. The number of foreground pixels M is variable over time.

The background subtracted silhouette regions are clustered with the variational Bayesian algorithm described in Section IV. During the clustering process each data point (pixel) $\tilde{\mathbf{y}}_k^j$ contains only the coordinates of the pixel. During the data association stage we use the red, green, blue (RGB) color information contained in each pixel to extract color features of a cluster.

C. The Social Force Model

The system dynamics model is defined by the following equation

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \boldsymbol{\xi}_{k-1}), \quad (5)$$

where $\boldsymbol{\xi}_{k-1}$ is the system noise vector. Under the Markovian assumption our state transition model for the i^{th} target is represented as $p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathcal{F}_i)$, where \mathcal{F}_i is the social force being applied on target i .

The motion of every target can be influenced by a number of factors, e.g., by the presence of other targets and static obstacles. The interactions between them are modeled here by means of a social force model [17], the key idea of which is to calculate attraction and repulsion forces which represent the social behavior of humans. The so-called ‘‘social forces’’ depend on the distances between the targets.

Following Newtonian dynamics, a change of states stems from the existence of exterior forces. Given a target i , the total number of targets that influence target i at time k , is N_i . The overall force \mathcal{F}_i applied on target i is the sum of the forces exerted by all the neighboring targets, $\mathcal{F}_i = \sum_{j \in N_i} \mathcal{F}_j$. Most of the existing force based models [17] consider only repulsive forces between targets to avoid collisions. However, in reality there can be many other types of social forces between targets due to different social behavior of targets, for instance attraction, spreading and following [25].

TABLE I
PARAMETERS OF SOCIAL FORCE MODEL

Symbol	Definition	Value
b	Boundary of social force in metres	3m
r_i	Radius of target’s influence in metres	0.2m
m	Mass of target	80kg
f_a	Magnitude of attractive force	500N
f_r	Magnitude of repulsive force	500N
Δt	Time interval	1/25

Suppose a target i at time step k has N_i neighbors, therefore, it would have N_i links with other targets $l = 1 \dots N_i$. Only those targets are considered as neighbors of target i which are within a threshold distance \hat{d} from i . We consider that a social behavior over link l is given by $\varphi_l \in \{1 \dots \Gamma\}$ where Γ is the total number of social behaviors. An interaction mode vector $\boldsymbol{\varphi}_i = [\varphi_1, \varphi_2, \dots, \varphi_{N_i}]$ for target i is a vector comprising the different social behaviors over all the links. The total number of interaction modes is $S = \Gamma^{N_i}$.

A force due to one interaction mode is calculated as a sum of forces over all the social links

$$\mathcal{F}(\boldsymbol{\varphi}_i) = \sum_{l=1}^{N_i} \phi^l, \quad (6)$$

where $\mathcal{F}(\boldsymbol{\varphi}_i)$ is the force due to social mode $\boldsymbol{\varphi}_i$ and ϕ^l is the force over the link l . In our work we consider three social forces: repulsion, attraction and non-interaction. These broadly encompass all possible human motions.

People naturally tend to avoid collisions between each other. This is reflected by *repulsion* forces. *Attraction* accounts for the behaviors when people approach each other for the intention of meeting, this behavior is usually ignored in the existing force based models. The *non-interactions* mode represents the behavior of independent motion of every person.

The repulsive force applied by target j on target i over the link l is calculated as [25]

$$\phi_-^l = f_r \exp\left(\frac{r_{ij}^l - d_{ij}^l}{b}\right) \mathbf{u}_{ji}, \quad (7)$$

when d_{ij}^l is less than an empirically found threshold \hat{d} , defined as 3 m in Table I in Section VI-E; where b which is set equal to \hat{d} is the boundary in which a target j has its influence of force on target i . The Euclidean distance between targets i and j over link l is defined as d_{ij}^l , $r_{ij}^l = r_i^l + r_j^l$ is the sum of radii of influence of targets i and j over link l and f_r is the magnitude of the repulsive force. The unit vector from target j to target i is represented as \mathbf{u}_{ji} .

When targets approach each other, repulsive forces act, to avoid collisions. The repulsive force therefore increases as described by (7). Similarly, when targets move apart the repulsion decreases because it is less likely for a collision to occur, and therefore they tend to come closer to each other. Attraction is a phenomenon opposite to repulsion. When targets move apart, they tend to attract each other for the intension of meeting and when they move closer, the repulsion increases which means they are less attracted. This attraction phenomenon is described by (8). However, if targets are at a distance more than a threshold, it can be assumed they do not influence each other.

An attractive force applied by target j on target i over the link l is calculated as [25]

$$\phi_{+}^l = f_a \exp\left(-\frac{(r_{ij}^l - d_{ij}^l)}{b}\right) \mathbf{u}_{ij}, \quad (8)$$

when d_{ij}^l is less than an empirically found threshold \hat{d} , defined as 3 m in Table I in Section VI-E, where f_a is the magnitude of the attractive force and force due to non-interactions is considered to be zero.

To represent the state \mathbf{x}_k of targets at every time step k , a pixel coordinate system is used. Since, our state transition (9) below is based on the social force model, to predict the new particles \mathbf{x}_k^s , position and velocity values contained within the state vector \mathbf{x}_{k-1} and the particles at time step $k-1$ are converted to the image ground coordinate system (a process known as homography [26]). After predicting the new set of N_s particles for state \mathbf{x}_k these particles are converted back to pixel coordinates. For conversion between the two coordinate systems the four point homography technique is implemented, as used in [27]. The position and velocity at time step k in pixel coordinates are represented as \mathbf{p}_k and \mathbf{v}_k , respectively, whereas, the position and velocity in ground coordinates systems are represented as $\tilde{\mathbf{p}}_k$ and $\tilde{\mathbf{v}}_k$ respectively. At every time step a new state is predicted with respect to interaction mode φ_i according to the following model [25]

$$\begin{bmatrix} \tilde{\mathbf{p}}_k(\varphi_i) \\ \tilde{\mathbf{v}}_k(\varphi_i) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{p}}_{k-1}(\varphi_i) + \tilde{\mathbf{v}}_{k-1} \Delta t + \frac{1}{2} \frac{\mathcal{F}(\varphi_i)}{m} \Delta t^2 \\ \tilde{\mathbf{v}}_{k-1}(\varphi_i) + \frac{\mathcal{F}(\varphi_i)}{m} \Delta t \end{bmatrix} + \boldsymbol{\xi}_k, \quad (9)$$

where Δt is the time interval between two frames, m represents mass of the target and $\boldsymbol{\xi}_k$ is the system noise vector.

The social force model gives more accurate predictions than the commonly used constant velocity model [28] and helps tracking objects in the presence of close interaction. This cannot be done with a simple constant velocity motion model.

The implementation of the force model is further explained in **Algorithm 1** at the end of the next section.

III. DATA ASSOCIATION AND LIKELIHOOD

To deal with the data association problem we represent a measurement to target ($M \rightarrow T$) association hypothesis by the parameter ψ_k , whereas Ψ_k represents the set of all hypotheses. At this stage we assume that one target can generate one measurement. However, later in this section we relax this assumption and develop a data association technique for associating multiple measurements (clusters of measurements) to one target. If we assume that measurements are independent of each other then the likelihood model conditioned on the state \mathbf{x}_k of targets and measurement to target association hypothesis ψ_k can be represented as in [10] (11)

$$p(\mathbf{y}_k | \mathbf{x}_k, \psi_k) = \prod_{j' \in \mathbf{J}_o} p_c(\mathbf{y}_{k'}^{j'}) \prod_{j \in \mathbf{J}} p_t(\mathbf{y}_k^j | \mathbf{x}_k^{i,j}), \quad (10)$$

where \mathbf{J}_o and \mathbf{J} are, respectively, measurement indices sets, corresponding to the clutter measurements and measurements from the targets to be tracked, for convenience of notation the time dependence is not shown on these quantities. The terms $p_c(\cdot)$ and $p_t(\cdot)$ are characterizing, respectively, the clutter likelihood and the measurement likelihood association to each target i whose state is represented as $\mathbf{x}_k^{i,j}$. If we assume that the clutter likeli-

hood $p_c(\cdot)$ is uniform over the volume γ of the measurement space and f_k represents the total number of clutter measurements, then the probability $p(\mathbf{y}_k | \mathbf{x}_k, \psi_k)$ can be represented as

$$p(\mathbf{y}_k | \mathbf{x}_k, \psi_k) = \gamma^{f_k} \prod_{j \in \mathbf{J}} p_t(\mathbf{y}_k^j | \mathbf{x}_k^{i,j}). \quad (11)$$

The form of $p_t(\mathbf{y}_k^j | \mathbf{x}_k^{i,j})$ is given in Section III-C; for notational convenience, in (13) and the remainder of the paper we drop the subscript t on the target likelihood.

A. The JPDAF Framework

The standard JPDAF algorithm estimates the marginal distribution of each target by following the prediction and update process described in (1) and (2). In the development below, we exploit the framework from [10]. The prediction step for each target independently becomes

$$p(\mathbf{x}_k^i | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i) p(\mathbf{x}_{k-1}^i | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}^i, \quad (12)$$

where we assume that the prediction at time step k is based on the measurements only at $k-1$. The JPDAF defines the likelihood of measurements with respect to target i as (25) in [10]

$$p(\mathbf{y}_k | \mathbf{x}_k^i) = \mathcal{A}_k^{i,o} + \sum_{j=1}^L \mathcal{A}_k^{i,j} p(\mathbf{y}_k^j | \mathbf{x}_k^i), \quad (13)$$

where $\mathcal{A}_k^{i,j}$ is the association probability that the i^{th} target is associated with the j^{th} measurement and $\mathcal{A}_k^{i,o}$ is the association probability that the target i remains undetected. With prediction and likelihood functions defined by (12) and (13), respectively, the JPDAF estimates the posterior probability of the state of target i

$$p(\mathbf{x}_k^i | \mathbf{y}_{1:k}) \propto p(\mathbf{y}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{y}_{1:k-1}). \quad (14)$$

The association probability $\mathcal{A}_k^{i,j}$ can be defined as (27) in [10]

$$\mathcal{A}_k^{i,j} = \sum_{\psi_k \in \Psi_k^{i,j}} p(\psi_k | \mathbf{y}_{1:k}), \quad (15)$$

where $\Psi_k^{i,j}$ represents the set of all those hypotheses which associate the j^{th} measurement to the i^{th} target. The probability $p(\psi_k | \mathbf{y}_{1:k})$ can be represented as (28) in [10]

$$p(\psi_k | \mathbf{y}_{1:k}) \propto p(\psi_k | \mathbf{x}_k) \gamma^{f_k} \prod_{j \in \mathbf{J}} p(\mathbf{y}_k^j | \mathbf{y}_{1:k-1}), \quad (16)$$

where $p(\psi_k | \mathbf{x}_k)$ is the probability of assignment ψ_k given the current state of the objects. We assume that all assignments have equal prior probability and hence $p(\psi_k | \mathbf{x}_k)$ can be approximated by a constant. If we define a normalizing constant π ensuring that $p(\psi_k | \mathbf{y}_{1:k})$ integrates to one and define the predictive likelihood $p(\mathbf{y}_k^j | \mathbf{y}_{1:k-1})$ as

$$p(\mathbf{y}_k^j | \mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k^j | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{y}_{1:k-1}) d\mathbf{x}_k^i, \quad (17)$$

then (16) becomes

$$p(\psi_k | \mathbf{y}_{1:k}) = \pi \gamma^{f_k} \prod_{(j,i) \in \psi_k} \int p(\mathbf{y}_k^j | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{y}_{1:k-1}) d\mathbf{x}_k^i. \quad (18)$$

B. Particle Filtering Based Approach

The standard JPDAF assumes a Gaussian marginal filtering distribution of the individual targets [10], [28]. In this paper we adopt a particle filtering based approach which represents the state of every target with the help of samples. Therefore, we modify (18) as

$$p(\psi_k | \mathbf{y}_{1:k}) = \pi \gamma^{f_k} \prod_{(j,i) \in \psi_k} \sum_{s=1}^{N_s} \hat{w}_k^{i,s} p(\mathbf{y}_k^j | \mathbf{x}_k^{i,s}), \quad (19)$$

where N_s is the total number of particles which are required to estimate the filtering distribution over every individual target and $\hat{w}_k^{i,s}$ represents the associated predictive weights as described in [10]. We can modify (19) as

$$p(\psi_k | \mathbf{y}_{1:k}) = \pi \gamma^{f_k} \prod_{(j,i) \in \psi_k} \frac{1}{N_s} \sum_{s=1}^{N_s} p(\mathbf{y}_k^j | \mathbf{x}_k^{i,s}). \quad (20)$$

Once the probability $p(\psi_k | \mathbf{y}_{1:k})$ is calculated according to (20), we can substitute it in (15) to calculate the association probability. We can further use this association probability to calculate the measurement likelihood from (13).

After drawing particles from a suitably constructed proposal distribution [23], the weights associated with particles are calculated, $\hat{w}_k^{i,s} = p(\mathbf{y}_k^j | \mathbf{x}_k^{i,s})$. The particles and their weights can then be used to approximate the posterior distribution

$$p(\mathbf{x}_k^i | \mathbf{y}_{1:k}) \approx \sum_{s=1}^{N_s} w_k^{i,s} \delta(\mathbf{x}_k^i - \mathbf{x}_k^{i,s}), \quad (21)$$

for individual targets where $\mathbf{x}_k^{i,s}$ is the s^{th} particle for target i and $w_k^{i,s}$ is the associated weight. For estimation of a state we predict N_φ particles with respect to every interaction mode. There are S interaction modes, therefore, $N_s = S \times N_\varphi$ represents the total number of samples.

Algorithm to Associate Multiple Measurements to a Target

In the case of tracking people in video, every person generates multiple measurements. To avoid any information loss we relax the assumption that every person can generate a single measurement. Here we present a data association technique for associating multiple measurements to every target. To achieve this, the algorithm groups the foreground pixels into clusters with the help of a variational Bayesian clustering technique, and then instead of associating one measurement to one target we associate clusters to every target. The clustering process returns κ clusters, where the number of clusters is not predefined or fixed. Clusters at discrete time k are regions represented as $\mathbf{Z}_k^1, \mathbf{Z}_k^2, \dots, \mathbf{Z}_k^\kappa$, where \mathbf{Z}_k^q is the q^{th} cluster which contains certain measurements, i.e., a number of vectors $\tilde{\mathbf{y}}_k^j$ of foreground pixels. The complete clustering process is described in Section IV. The clustering process aims to assign measurements to the respective clusters and gives as output a correspondence matrix $\mathbf{B}_k = [(\mathbf{b}_k^1)^T, \dots, (\mathbf{b}_k^j)^T, \dots, (\mathbf{b}_k^M)^T]^T$, where $\mathbf{b}_k^j = [b_k^{j,1}, b_k^{j,2}, \dots, b_k^{j,q}, \dots, b_k^{j,\kappa}]$ indicates, at discrete time k , to which cluster, the measurement vector $\tilde{\mathbf{y}}_k^j$ corresponds. All but one of the elements of \mathbf{b}_k^j is zero, if e.g., $\tilde{\mathbf{y}}_k^j$ belongs to the q^{th} cluster then the correspondence vector will be, $\mathbf{b}_k^j = [0, \dots, 0, 1, 0, \dots, 0]$, which shows only the q^{th} element of \mathbf{b}_k^j is nonzero. Please note that from this section onwards all equations are written with respect to the measurement vector $\tilde{\mathbf{y}}_k$ that contains foreground pixels.

We modify the measurement to target ($M \rightarrow T$) association probability $\mathcal{A}_k^{i,j}$ to calculate the cluster to target ($Z \rightarrow T$) association probability $\mathcal{A}_k^{i,q}$, where q represents the cluster index

$$\mathcal{A}_k^{i,q} = \sum_{\psi_k \in \Psi_k^{i,q}} p(\psi_k | \tilde{\mathbf{y}}_{1:k}), \quad (22)$$

where $\Psi_k^{i,q}$ represents the set of all those hypotheses which associate the q^{th} cluster to the i^{th} target. We modify (20) to define the probability $p(\psi_k | \tilde{\mathbf{y}}_{1:k})$ as

$$p(\psi_k | \tilde{\mathbf{y}}_{1:k}) = \pi \gamma^{f_k} \prod_{(q,i) \in \psi_k} \prod_{\tilde{\mathbf{y}}_k^j: b_k^{j,q}=1} \frac{1}{N_s} \sum_{s=1}^{N_s} p(\tilde{\mathbf{y}}_k^j | \mathbf{x}_k^{i,s}), \quad (23)$$

where $\tilde{\mathbf{y}}_k^j : b_k^{j,q} = 1$ represents only those measurements which are associated with cluster q . The measurement likelihood defined in (13) is modified as follows

$$p(\tilde{\mathbf{y}}_k | \mathbf{x}_k^{i,s}) = \mathcal{A}_k^{i,o} + \sum_{j=1}^M \mathcal{A}_k^{i,(q:b_k^{j,q}=1)} p(\tilde{\mathbf{y}}_k^j | \mathbf{x}_k^{i,s}), \quad (24)$$

where $q : b_k^{j,q} = 1$ represents that the j^{th} measurement is associated with the q^{th} cluster and $\mathcal{A}_k^{i,o}$ represents the probability that the cluster is not associated to the i^{th} target and $\mathcal{A}_k^{i,(q:b_k^{j,q}=1)}$ is the probability that the cluster is associated to the i^{th} target. To limit the number of hypotheses when the number of targets increases, we have adopted Murty's algorithm [29]. This algorithm returns the k -best hypotheses. The elements of the cost matrix for Murty's algorithm are calculated with the particles $\mathbf{x}_k^{i,s}$

$$cost_q^i = \prod_{\tilde{\mathbf{y}}_k^j: b_k^{j,q}=1} \frac{1}{N_s} \sum_{s=1}^{N_s} p(\tilde{\mathbf{y}}_k^j | \mathbf{x}_k^{i,s}), \quad (25)$$

where $cost_q^i$ represents the cost of assigning the q^{th} cluster to the i^{th} target.

The JPDAF [30] data association technique proposed in [4] performs well when two targets do not undergo a full occlusion. However, this data association technique sometimes fails to associate measurements correctly, especially when two or more targets partially occlude each other or when targets come out of full occlusion. This is due to the fact that this data association is performed on the basis of location information without any information about the features of targets. This can result in missed detections and wrong identifications.

In order to cope with these challenges under occlusions, a data association technique is proposed which assigns clusters to targets by exploiting both location and features of clusters. First, we define the term $\prod_{\tilde{\mathbf{y}}_k^j: b_k^{j,q}=1} \frac{1}{N_s} \sum_{s=1}^{N_s} p(\tilde{\mathbf{y}}_k^j | \mathbf{x}_k^{i,s})$ in equation (23) as follows

$$\prod_{\tilde{\mathbf{y}}_k^j: b_k^{j,q}=1} \frac{1}{N_s} \sum_{s=1}^{N_s} p(\tilde{\mathbf{y}}_k^j | \mathbf{x}_k^{i,s}) = p(\mathbf{Z}_k^q | \mathbf{x}_k^{i,s}) \prod_{\tilde{\mathbf{y}}_k^j: b_k^{j,q}=1} \frac{1}{N_s} \sum_{s=1}^{N_s} \mathcal{N}(\tilde{\mathbf{y}}_k^j | \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s), \quad (26)$$

where $\boldsymbol{\mu}_s = \mathbf{x}_k^{i,s}$ is the mean vector and $\boldsymbol{\Sigma}_s$ is the fixed and known diagonal covariance matrix. The association probability $p(\mathbf{Z}_k^q | \mathbf{x}_k^{i,s})$ in (26) is calculated by extracting features from the clusters at time step k and comparing them with a reference feature (template) of target i . The reference feature for target i is extracted when it first appears in the monitored area. To improve

the computational efficiency, the probability $p(\mathbf{Z}_k^q | \mathbf{x}_k^{i,s})$ is calculated only when targets approach each other. This association probability is defined as

$$p(\mathbf{Z}_k^q | \mathbf{x}_k^{i,s}) = \begin{cases} \exp\left(-\frac{d^i(H_{ref}^i, H_k^q)}{2\sigma^2}\right) & \text{if } P_{occlusion}^i > \vartheta \\ 1 & \text{otherwise} \end{cases} \quad (27)$$

where σ^2 is the measurement noise variance, H_{ref}^i and H_k^q are histograms of features of the i^{th} target and q^{th} cluster, respectively and $d^i(H_{ref}^i, H_k^q)$ is the distance between them. One possible option is to use the Bhattacharyya distance [31]. The parameter ϑ is a predefined threshold, and the probability of occlusion is defined as

$$P_{occlusion}^i = \exp\left(-\frac{d_{\min}^i}{\Delta_c}\right), \quad (28)$$

where d_{\min}^i is the minimum Euclidean distance among a set of distances which is calculated for target i from all other targets at time k ,

$$d_{\min}^i = \min_{i \neq n} \{d_{i,n}\}, \quad (29)$$

and Δ_c is the distance constant which limits the influence of target n on target i . A higher value of minimum distance d_{\min}^i results in a smaller value of probability of occlusions and when this probability drops below threshold ϑ the probability $p(\mathbf{Z}_k^q | \mathbf{x}_k^{i,s})$ becomes unity.

Algorithm 1: The social force model

Input: N_s particles and the state of targets at time step $k - 1$

Output: N_s particles for time step k

- 1: Convert the state at $k - 1$ from pixel coordinates system to real world ground coordinates.
 - 2: Use ground coordinates to calculate distances between targets.
 - 3: Based on distances create links between targets.
 - 4: **for** $i = 1, \dots, N$ (where N is the number of targets) **do**
 - 5: Create $S = \Gamma^{N_i}$ social modes
 - 6: Calculate forces due to every social mode by using (6), (7) and (8).
 - 7: **for** $s = 1 : \Gamma^{N_i} : N_s$ (where N_s is the number of particles) **do**
 - 8: Convert the particle $\mathbf{x}_{k-1}^{i,s}$ to the ground plane.
 - 9: Add random noise to get $\mathbf{x}_{k-1}^{i,s} + \boldsymbol{\xi}_{k-1}$.
 - 10: **for** $s_{\varphi_i} = 1, \dots, \Gamma^{N_i}$ **do**
 - 11: Predict particle $\mathbf{x}_{k-1}^{i,s+s_{\varphi_i}-1}$ by using particle $\mathbf{x}_{k-1}^{i,s} + \boldsymbol{\xi}_{k-1}$ w.r.t. the social mode φ_i in (9).
 - 12: Convert the particle $\mathbf{x}_{k-1}^{i,s+s_{\varphi_i}-1}$ to the pixel coordinates.
 - 13: **end for**
 - 14: **end for**
 - 15: **end for**
-

IV. VARIATIONAL BAYESIAN CLUSTERING—FROM MEASUREMENTS TO CLUSTERS

The clustering process aims to subdivide the foreground image into regions corresponding to the different targets. A

variational Bayesian clustering technique is developed to group the M foreground pixels into κ clusters. Each cluster at time index k is represented by its center $\boldsymbol{\mu}_k^q$, where $q = 1, \dots, \kappa$. A binary indicator variable $b_k^{j,q} \in \{0, 1\}$ represents to which of the q^{th} clusters data point $\tilde{\mathbf{y}}_k^j$ is assigned; for example, when the data point $\tilde{\mathbf{y}}_k^j$ is assigned to the q^{th} cluster then $b_k^{j,q} = 1$, and $b_k^{j,l} = 0$ for $l \neq q$.

We assume that the foreground pixel locations are modeled by a mixture of Gaussian distributions. The clustering task can be viewed as fitting mixtures of Gaussian distributions [32] to the foreground measurements. Every cluster \mathbf{Z}_k^q of foreground pixel locations is assumed to be modeled by a Gaussian distribution, $\mathcal{N}(\tilde{\mathbf{y}}_k^j | \boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q)$ which is one component of the mixture of Gaussian distributions. Each cluster has a mean vector $\boldsymbol{\mu}_k^q$ and a covariance matrix $\boldsymbol{\Sigma}_k^q$. Hence, the probability of a data point $\tilde{\mathbf{y}}_k^j$ can be represented as

$$p(\tilde{\mathbf{y}}_k^j | \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{q=1}^{\kappa} C_k^q \mathcal{N}(\tilde{\mathbf{y}}_k^j | \boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q), \quad (30)$$

where $\boldsymbol{\mu}_k = \{\boldsymbol{\mu}_k^q\}_{q=1}^{\kappa}$ and $\boldsymbol{\Sigma}_k = \{\boldsymbol{\Sigma}_k^q\}_{q=1}^{\kappa}$. The mixing coefficient vector is defined as $\mathbf{C}_k = [C_k^1, \dots, C_k^q, \dots, C_k^{\kappa}]^T$, where C_k^q represents the probability of selecting the q^{th} component of the mixture which is the probability of assigning the j^{th} measurement to the q^{th} cluster. If we assume that all the measurements are independent, then the log likelihood function becomes [32]

$$\ln p(\tilde{\mathbf{y}}_k | \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{j=1}^M \ln \left\{ \sum_{q=1}^{\kappa} C_k^q \mathcal{N}(\tilde{\mathbf{y}}_k^j | \boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q) \right\}. \quad (31)$$

The data point $\tilde{\mathbf{y}}_k^j$ belongs to only one cluster at a time and this association is characterized by a correspondence variable vector \mathbf{b}_k^j . Therefore, it can be represented as $C_k^q = p(b_k^{j,q} = 1)$ where the mixing coefficients must satisfy the following conditions: $0 < C_k^q \leq 1$ and $\sum_{q=1}^{\kappa} C_k^q = 1$. Since only one element of \mathbf{b}_k^j is equal to 1, the probability of the mixing coefficient can also be written as

$$p(\mathbf{b}_k^j | \mathbf{C}_k) = \prod_{q=1}^{\kappa} (C_k^q)^{b_k^{j,q}}. \quad (32)$$

Similarly, we assume that $p(\tilde{\mathbf{y}}_k^j | b_k^{j,q} = 1) = \mathcal{N}(\tilde{\mathbf{y}}_k^j | \boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q)$. Since only one element of \mathbf{b}_k^j is equal to 1, we can write again

$$p(\tilde{\mathbf{y}}_k^j | \mathbf{b}_k^j, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \prod_{q=1}^{\kappa} \mathcal{N}(\tilde{\mathbf{y}}_k^j | \boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q)^{b_k^{j,q}}. \quad (33)$$

As a result of the clustering process we will obtain estimates of the probabilistic weight vector \mathbf{C}_k , the mean vectors $\boldsymbol{\mu}_k$, the covariance matrices $\boldsymbol{\Sigma}_k$ and correspondence matrix \mathbf{B}_k .

The estimation problem can be simplified by introducing hidden (latent) variables. We consider the correspondence variables \mathbf{B}_k as latent variables and suppose that they are known. Then $\{\tilde{\mathbf{y}}_k, \mathbf{B}_k\}$ represents the complete dataset. The log likelihood function for this complete data set becomes $\ln p(\tilde{\mathbf{y}}_k, \mathbf{B}_k | \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Now the new set of unknown parameters contains $\mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ which can be estimated by the proposed adaptive variational Bayesian clustering algorithm. If, for simplicity, these parameters are represented as

a single parameter $\Theta_k = (\mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, then the desired distribution can be represented as $p(\Theta_k|\tilde{\mathbf{y}}_k)$. The log marginal distribution $p(\tilde{\mathbf{y}}_k)$ can be decomposed as described in [32] (Section 9.4 and equations (10.2)–(10.4))

$$\ln p(\tilde{\mathbf{y}}_k) = L(q) + KL(q||p). \quad (34)$$

We define the distribution $\mathcal{Q}(\Theta_k)$ as an approximation of the desired distribution. Then the objective of the variational Bayesian method is to optimize this distribution, by minimizing the Kullback Leibler (KL) divergence

$$KL(q||p) = - \int \mathcal{Q}(\Theta_k) \ln \left\{ \frac{p(\Theta_k|\tilde{\mathbf{y}}_k)}{\mathcal{Q}(\Theta_k)} \right\} d\Theta_k. \quad (35)$$

The symbol $\|\cdot\|$ represents the KL divergence, and

$$L(q) = \int \mathcal{Q}(\Theta_k) \ln \left\{ \frac{p(\Theta_k, \tilde{\mathbf{y}}_k)}{\mathcal{Q}(\Theta_k)} \right\} d\Theta_k. \quad (36)$$

Minimizing the KL divergence is equivalent to maximizing the lower bound $L(q)$. The maximum of this lower bound can be achieved when the approximate distribution $\mathcal{Q}(\Theta_k)$ is exactly equal to the desired posterior distribution $p(\Theta_k|\tilde{\mathbf{y}}_k)$.

The joint distribution $p(\Theta_k, \tilde{\mathbf{y}}_k)$ can be decomposed as [32]

$$\begin{aligned} p(\Theta_k, \tilde{\mathbf{y}}_k) &= p(\tilde{\mathbf{y}}_k, \mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= p(\tilde{\mathbf{y}}_k|\mathbf{B}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)p(\mathbf{B}_k|\mathbf{C}_k)p(\mathbf{C}_k|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k). \end{aligned} \quad (37)$$

We assume that the variational distribution $\mathcal{Q}(\Theta_k) = \mathcal{Q}(\mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ can be factorized between latent variable \mathbf{B}_k and parameters $\mathbf{C}_k, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$.

$$\mathcal{Q}(\mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathcal{Q}(\mathbf{B}_k)\mathcal{Q}(\mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (38)$$

A similar assumption is made in [32], please see equation (10.42). Optimization of the variational distribution can therefore be represented as

$$\mathcal{Q}^*(\mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathcal{Q}^*(\mathbf{B}_k)\mathcal{Q}^*(\mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (39)$$

By considering equation (10.54) from [32] the distribution $\mathcal{Q}^*(\mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ can further be decomposed into $\mathcal{Q}^*(\mathbf{C}_k)$, $\mathcal{Q}^*(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $(\cdot)^*$ represents the optimum distribution. Therefore, the optimum distribution can be written as

$$\mathcal{Q}^*(\mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \mathcal{Q}^*(\mathbf{B}_k)\mathcal{Q}^*(\mathbf{C}_k)\mathcal{Q}^*(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (40)$$

This shows that the optimum over the joint distribution is equivalent to obtaining $\mathcal{Q}^*(\mathbf{B}_k)$, $\mathcal{Q}^*(\mathbf{C}_k)$ and $\mathcal{Q}^*(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Therefore, the optimum distributions over Θ_k can be evaluated by optimizing $L(q)$ with respect to all parameters one-by-one. A general form of optimization can be written as in equation (10.9) of [32]

$$\ln \mathcal{Q}^*(\Theta_k^c) = E_{d \neq c}[\ln p(\Theta_k, \tilde{\mathbf{y}}_k)] + \text{constant}, \quad (41)$$

where $E_{d \neq c}[\cdot]$ represents the mathematical expectation with respect to the distributions over all the elements in Θ_k for $d \neq c$, where d is the d^{th} component of Θ_k . $\mathcal{Q}^*(\Theta_k^c)$ represents the optimum approximate distribution over the c^{th} component of

Θ_k . We next evaluate the optimum distributions $\mathcal{Q}^*(\mathbf{B}_k|\mathbf{C}_k)$, $\mathcal{Q}^*(\mathbf{C}_k)$ and $\mathcal{Q}^*(\boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q)$ by using (41).

1) *Optimum Distribution Over \mathbf{B}_k* : According to (41), the optimum distribution over \mathbf{B}_k can be written as

$$\ln \mathcal{Q}^*(\mathbf{B}_k) = E_{\mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}[\ln p(\mathbf{B}_k, \tilde{\mathbf{y}}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] + \text{constant}. \quad (42)$$

Probabilities $p(\mathbf{B}_k|\mathbf{C}_k)$ and $p(\tilde{\mathbf{y}}_k|\mathbf{B}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ can be defined by using (32) and (33) respectively

$$p(\mathbf{B}_k|\mathbf{C}_k) = \prod_{j=1}^M \prod_{q=1}^{\kappa} (\mathbf{C}_k^q)^{b_k^{j,q}}, \quad (43)$$

and

$$p(\tilde{\mathbf{y}}_k|\mathbf{B}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \prod_{j=1}^M \prod_{q=1}^{\kappa} \mathcal{N}(\tilde{\mathbf{y}}_k^j|\boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q)^{b_k^{j,q}}. \quad (44)$$

By using (42), (43) and (44), the optimum distribution over the correspondence variable \mathbf{B}_k becomes

$$\mathcal{Q}^*(\mathbf{B}_k) = \prod_{j=1}^M \prod_{q=1}^{\kappa} (r_k^{j,q})^{b_k^{j,q}}, \quad (45)$$

where $r_k^{j,q}$ is the responsibility that component q takes to explain the measurement $\tilde{\mathbf{y}}_k^j$. The derivation of (45) and $r_k^{j,q}$ can be found in the Appendix (and in Appendix-A of the long version of the paper available online at *arXiv*).

2) *Optimum Distribution Over \mathbf{C}_k* : Before evaluating the optimum distributions $\mathcal{Q}^*(\mathbf{C}_k)$ and $\mathcal{Q}^*(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ we need to first define their priors. The Dirichlet distribution is chosen as a prior over the mixing coefficient \mathbf{C}_k

$$p(\mathbf{C}_k) = \text{Dir}(\mathbf{C}_k|\alpha_o), \quad (46)$$

where $\text{Dir}(\cdot)$ denotes the Dirichlet distribution, and α_o is an effective prior number of observations associated with each component of the mixture. Using (41) we can write

$$\ln \mathcal{Q}^*(\mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = E_{\mathbf{B}}[\ln p(\tilde{\mathbf{y}}_k, \mathbf{B}_k, \mathbf{C}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] + \text{const}, \quad (47)$$

where const. represents a constant. The optimum distributions $\mathcal{Q}^*(\mathbf{C}_k)$ over \mathbf{C}_k can be calculated by using (44), (46) and (47), which becomes

$$\mathcal{Q}^*(\mathbf{C}_k) = \text{Dir}(\mathbf{C}_k|\boldsymbol{\alpha}_k), \quad (48)$$

where $\boldsymbol{\alpha}_k = [\alpha_k^1, \dots, \alpha_k^{\kappa}]$ and one of its components α_k^q can be defined as

$$\alpha_k^q = \alpha_o + N_k^q, \quad (49)$$

$$N_k^q = \sum_{j=1}^M r_k^{j,q}, \quad (50)$$

and $r_k^{j,q}$ is the responsibility that component q takes to explain the measurement $\tilde{\mathbf{y}}_k^j$. The derivation of (48) can be found in Appendix-B of the long version of the paper available online at *arXiv*.

3) *Optimum Distribution Over $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$* : The prior over the mean $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$ is defined by the independent Gaussian-Wishart distribution

$$p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)p(\boldsymbol{\Sigma}_k) = \prod_{q=1}^{\kappa} \mathcal{N}(\boldsymbol{\mu}_k^q|\mathbf{m}_o, \beta_o^{-1}\boldsymbol{\Sigma}_k^q)\mathcal{W}(\boldsymbol{\Sigma}_k^q|\mathbf{W}_o, v_o), \quad (51)$$

where $\mathbf{m}_o, \beta_o, \mathbf{W}_o$ and v_o are the prior parameters. By using (44), (46) and (47), decomposition of (37) and following the steps explained in Appendix-C of the long version of the paper available online at *arXiv*, the distribution becomes

$$\mathcal{Q}^*(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)\mathcal{Q}^*(\boldsymbol{\Sigma}_k) = \prod_{q=1}^{\kappa} \mathcal{N}(\boldsymbol{\mu}_k^q|\mathbf{m}_q, \beta_q^{-1}\boldsymbol{\Sigma}_k^q)\mathcal{W}(\boldsymbol{\Sigma}_k^q|\mathbf{W}_q, v_q), \quad (52)$$

where $\mathbf{m}_q, \beta_q, \mathbf{W}_q$ and v_q are defined from

$$\beta_q = \beta_o + N_k^q, \quad (53)$$

$$\mathbf{m}_q = \frac{1}{\beta_q}(\beta_o + N_k^q\bar{\mathbf{y}}_k^q), \quad (54)$$

where,

$$\bar{\mathbf{y}}_k^q = \frac{1}{N_k^q} \sum_{j=1}^M r_k^{j,q} \tilde{\mathbf{y}}_k^j, \quad (55)$$

$$\mathbf{W}_q^{-1} = \mathbf{W}_o^{-1} + N_k^q \mathbf{S}_k^q + \frac{\beta_o N_k^q}{\beta_o + N_k^q} (\bar{\mathbf{y}}_k^q - \mathbf{m}_o)(\bar{\mathbf{y}}_k^q - \mathbf{m}_o)^T, \quad (56)$$

$$v_q = v_o + N_k^q, \quad (57)$$

and

$$\mathbf{S}_k^q = \frac{1}{N_k^q} \sum_{j=1}^M r_k^{j,q} (\tilde{\mathbf{y}}_k^j - \bar{\mathbf{y}}_k^q)(\tilde{\mathbf{y}}_k^j - \bar{\mathbf{y}}_k^q)^T. \quad (58)$$

The variational Bayesian technique operates in two steps to optimize the posterior distributions of unknown variables and parameters. In the first step it calculates the responsibilities $r_k^{j,q}$ using (81) and in the second step it uses these responsibilities to optimize the distributions by using (45), (48) and (52). These steps are repeated until some convergence criterion is met. In our work we monitor the lower bound $L(q)$ after every iteration to test the convergence. When the algorithm converges, the value of the lower bound does not change more than a small amount. The clustering algorithm is further summarized in **Algorithm 2** given in Section V.

One of the important advantages of variational Bayesian clustering is that it can automatically determine the number of clusters by using the measurements. This can be achieved if we set the parameter α_o less than 1. This helps to obtain a solution which contains a minimum number of clusters to represent the data [4].

The position and shape of clusters are defined using the parameters \mathbf{m}_o and \mathbf{W}_o , respectively. A possible choice for selecting these priors is given in Section VI. This stage returns the minimum possible number of clusters and their associated measurements which are defined by the correspondence matrix \mathbf{B}_k . In the next stage of the algorithm these clusters are assigned to the targets by using a data association technique as described after Section III-C.

V. VARIABLE NUMBER OF TARGETS

Many multi-target tracking algorithms assume that the number of targets is fixed and known [3], [12], [15], [20]. In [33] the JPDAF is extended to a variable number of targets. In other works, including [4] the posterior probability of the number of targets N is estimated given the number of clusters κ at each discrete time step k

$$p(N_k|\kappa_{1:k}) \propto p(\kappa_k|N_k) \sum_n \{p(N_k|N_{k-1}=n)p(N_{k-1}=n|\kappa_{1:k-1})\}, \quad (59)$$

where $p(\kappa_k|N_k)$ is the probability of the κ_k clusters given N_k targets. Here we deal with changeable shapes and calculate the number of targets in a different way. The number of targets is estimated based on the: 1) location $\boldsymbol{\mu}_k^q$ of clusters at time step k , 2) size of clusters at time step k , and 3) position vector \mathbf{x}_{k-1} of the targets at the previous time step $k-1$.

The number of targets in the first video frame is determined based on the variational Bayesian clustering. At every following video frame the number N_k of targets is evaluated, by the following stochastic relationship, similarly to [34]

$$N_k = N_{k-1} + \varrho_k. \quad (60)$$

The variable $\varrho_k \in \{-1, 0, 1\}$ is defined as

$$\varrho_k = \begin{cases} -1 & \text{if } p(\text{death}) > p(\text{birth}) \& p(\text{death}) > Thr \\ 1 & \text{if } p(\text{birth}) > p(\text{death}) \& p(\text{birth}) > Thr \\ 0 & \text{otherwise} \end{cases} \quad (61)$$

where $p(\text{death})$ corresponds to the probability of decrementing the number of targets, similarly, $p(\text{birth})$ represents the probability of incrementing number of targets and Thr is an appropriately chosen threshold.

We assume that at a given time only one target can enter or leave. In the monitored area people can only enter or leave through a specifically known region in a video frame. This known region is called the *red region*. This red region is modeled by a Gaussian distribution with its center $\boldsymbol{\mu}^r$ and covariance matrix $\boldsymbol{\Sigma}^r$. Multiple Gaussian distributions can be used if the red region was disjoint.

A. Probability of Death

The probability of death of a target depends on two things: 1) the probability $p(\mathbf{Z}_k^r)$ of finding a cluster \mathbf{Z}_k^r in the red region and 2) the probability $p(\text{Existing Target}|\mathbf{Z}_k^r, \mathbf{x}_{k-1})$ that the cluster found in the red region is due to an existing target (denoted by the variable *Existing Target*). This means that when a target is leaving the monitored area, a cluster is found in the red region, i.e., at the entrance or exit point, and this cluster will be due to an existing target. Therefore, probability of death is modeled as

$$p(\text{death}) = p(\mathbf{Z}_k^r)p(\text{Existing Target}|\mathbf{Z}_k^r, \mathbf{x}_{k-1}). \quad (62)$$

In order to estimate $p(\mathbf{Z}_k^r)$, we calculate the probability of the measurements (foreground pixels) from

$$p(\tilde{\mathbf{y}}_k^j|\boldsymbol{\mu}^r, \boldsymbol{\Sigma}^r) = \mathcal{N}(\tilde{\mathbf{y}}_k^j|\boldsymbol{\mu}^r, \boldsymbol{\Sigma}^r). \quad (63)$$

Note that we are not considering measurements of one cluster only, because a target in the red region may generate more than one cluster. Therefore, measurements which have probability $p(\tilde{\mathbf{y}}_k^j | \boldsymbol{\mu}^r, \boldsymbol{\Sigma}^r)$ greater than a set threshold value are considered as measurements found in the red region and all these measurements are therefore grouped to form cluster \mathbf{Z}_k^r . Since we assume that maximum one target can be in the red region, then, all the measurements of \mathbf{Z}_k^r are considered to be generated by one target. For an N_p number of measurements in cluster \mathbf{Z}_k^r , the probability of finding a cluster in the red region depends on the number of pixels in \mathbf{Z}_k^r and is calculated as

$$p(\mathbf{Z}_k^r) = 1 - \exp\left(-\frac{N_p}{\Delta_p}\right), \quad (64)$$

where Δ_p is a pixel constant which limits the range of pixels to be considered for calculating $p(\mathbf{Z}_k^r)$. The second probability in (62) is the probability that cluster \mathbf{Z}_k^r is generated by an existing target. This probability depends on the location of existing targets and the location of cluster \mathbf{Z}_k^r . The probability $p(\text{Existing Target} | \mathbf{Z}_k^r, \mathbf{x}_{k-1})$ is calculated as

$$p(\text{Existing Target} | \mathbf{Z}_k^r, \mathbf{x}_{k-1}) = \exp\left(-\frac{\tilde{d}_{\min}}{\Delta_d}\right), \quad (65)$$

where Δ_d is a constant which can be chosen experimentally. To calculate the distance \tilde{d}_{\min} , we chose the minimum distance among the distances which are found from the centroid of \mathbf{Z}_k^r , $\text{centroid}(\mathbf{Z}_k^r)$ to each of the existing targets at time step $k-1$. Note that this \tilde{d}_{\min} is different from the d_{\min} calculated in (28).

$$\tilde{d}_{\min} = \min_{i=1 \dots N_{k-1}} \|\text{centroid}(\mathbf{Z}_k^r) - \mathbf{x}_{k-1}^i\|, \quad (66)$$

where $\|\cdot\|$ denotes the Euclidean distance operator. Note that the probability \mathbf{Z}_k^r depends on the distance between the centroid of \mathbf{Z}_k^r and the closest existing target. The probability that cluster \mathbf{Z}_k^r is generated by an existing target will be high if \tilde{d}_{\min} is small and vice versa.

B. Probability of Birth

The probability of birth of a target depends on two different probabilities: 1) the probability of finding a cluster \mathbf{Z}_k^r in the red region $p(\mathbf{Z}_k^r)$ and 2) the probability that the cluster found in the red region is not due to an existing target. According to our assumption that at a given time step the cluster in the red region can only be either from an existing target or a new target, the probability that the cluster in the red region is due to a new target is $1 - p(\text{Existing Target} | \mathbf{Z}_k^r, \mathbf{x}_k)$. The probability of birth can be calculated as

$$p(\text{birth}) = p(\mathbf{Z}_k^r)(1 - p(\text{Existing Target} | \mathbf{Z}_k^r, \mathbf{x}_k)). \quad (67)$$

Finally, (62) and (67) are applied in (60) and the number of targets is updated by using (66). This completes the full description of our algorithm. A summary of the overall proposed algorithm is described in **Algorithm 2**.

Algorithm 2: Summary of Proposed Algorithm

Input: Video frame at time step k

Output: 2-D position of all the targets in each frame k

- 1: Perform the background subtraction to extract M foreground pixels.
 - 2: Save the coordinates of foreground pixels which represent data points $\tilde{\mathbf{y}}_k$.
 - Clustering**
 - 3: Initialize the parameters $\alpha_o, \beta_o, \mathbf{m}_o \mathbf{W}_o$ and v_o .
 - 4: **for** $j = 1, \dots, M$ **do**
 - 5: **for** $q = 1, \dots, \kappa$ **do**
 - 6: Evaluate initial $\rho_o^{j,q}$ by using initialized parameters $\alpha_o, \beta_o, \mathbf{m}_o \mathbf{W}_o$ and v_o in (79) and (80).
 - 7: **end for**
 - 8: Calculate the normalization factor $\tilde{\eta} = \sum_{q=1}^{\kappa} \rho_o^{j,q}$
 - 9: **for** $q = 1, \dots, \kappa$ **do**
 - 10: Evaluate normalized responsibilities: $r_o^{j,q} = \frac{\rho_o^{j,q}}{\tilde{\eta}}$.
 - 11: **end for**
 - 12: **end for**
 - 13: **while** Convergence criterion is not satisfied **do**
 - 14: **for** $q = 1, \dots, \kappa$ **do**
 - 15: Evaluate new $N_k^q, \tilde{\mathbf{y}}_k^q, \alpha_k^q, \beta_k^q, \mathbf{m}_q, \mathbf{W}_q^{-1}, v_q$ and \mathbf{S}_k^q with (50), (55), (49), (53), (54), (56), (57) and (58) respectively. For first iteration use initial responsibilities $r_o^{j,q}$.
 - 16: **end for**
 - 17: Evaluate new responsibilities $r_k^{j,q}$ for all j and q by using new $N_k^q, \tilde{\mathbf{y}}_k^q, \alpha_k^q, \beta_k^q, \mathbf{m}_q, \mathbf{W}_q^{-1}, v_q$ and \mathbf{S}_k^q and repeating steps (4) through (12).
 - 18: **end while**
 - 19: Assign the l^{th} cluster to measurement $\tilde{\mathbf{y}}_k^j$, when $r_k^{j,l} = \max_{q=1:\kappa} r_k^{j,q}$ and repeat it for all the measurements.
 - 20: Delete the small clusters.
 - Identify Number of Targets**
 - 21: Evaluate probability of death and birth by using (62) and (67) respectively.
 - 22: Identify number of targets by using (60).
 - Data Association and Tracking**
 - 23: Evaluate the cost matrix by using (25).
 - 24: Evaluate k-best hypotheses by using Murty's algorithm [29].
 - 25: **for** $i = 1, \dots, N$ (where N is the total number of targets) **do**
 - 26: Draw N_s samples by using the state transition (9) as explained in **Algorithm 1**.
 - 27: **for** $q = 1, \dots, \kappa$ **do**
 - 28: Evaluate the set of hypotheses $\Psi_k^{i,q}$ which assign the q^{th} cluster to the i^{th} target.
 - 29: For every hypothesis $\psi_k \in \Psi_k^{i,q}$ evaluate $\ln p(\psi_k | \tilde{\mathbf{y}}_{1:k})$ by using (23) and (26).
 - 30: Evaluate $\mathcal{A}_k^{i,q}$ with (22).
 - 31: **end for**
 - 32: Weight all the particles using (24).
 - 33: Update states of all the targets according to (21).
 - 34: **end for**
-



Fig. 1. Background subtraction results for certain frames of sequence “seq45-3p-1111_cam3_divx_audio” of the AV16.3 dataset: codebook background subtraction is implemented to separate the foreground pixels from the background. (a) Frame 255 (b) Frame 278 (c) Frame 288 (d) Frame 313.

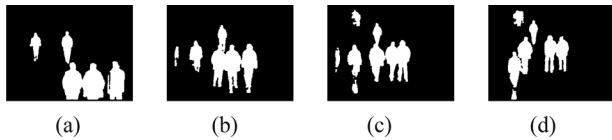


Fig. 2. Background subtraction results for certain frames of sequence “ThreePastShop2cor” of the CAVIAR dataset: codebook background subtraction is implemented to separate the foreground pixels from the background. (a) Frame 334 (b) Frame 440 (c) Frame 476 (d) Frame 524.

VI. EXPERIMENTAL RESULTS

To examine the robustness of the algorithm to close interactions, occlusions and varying number of targets, the algorithm is evaluated by tracking a variable number of people in three different publicly available video datasets: CAVIAR, PETS2006 and AV16.3. The test sequences are recorded at a resolution of 288×360 pixels at 25 frames/sec and in total there are 45 sequences. The performance of the proposed algorithm is compared with recently proposed techniques [4], [1] and [35]. All the parameters are discussed in the following subsections. The algorithm automatically detects and initializes the targets when they enter the monitored area. For evaluation of the tracking results we convert the final tracked locations of targets from pixel coordinates to the ground coordinates by using four point homography.

A. Background Subtraction Results

The codebook background subtraction method [24] is implemented which is one of the best background subtraction methods since it is resistant to illumination changes and can capture the structure of background motion. Adaptation to the background changes and noise reduction is additionally achieved with the blob technique [36]. Figs. 1–3 show the results obtained from the codebook background subtraction method [24] for a few selected video frames from the AV16.3, CAVIAR and PETS2006 datasets respectively. In the experiment, we set the shadow bound $\alpha = 0.5$, highlight bound $\beta = 2$ and the color detection threshold $\varepsilon = 20$ (see [24] for further details about these parameters). These parameters are the same for all the sequences of all three datasets. The background subtraction results provide the coordinates of the foreground pixels (the moving objects) which represent data points $\tilde{\mathbf{y}}_k$. Frames 440 and 476 in Fig. 2 show that we get a few extra foreground pixels due to reflections on the floor and in the glass which can be eliminated at the clustering stage.

B. Clustering Results

For the clustering process at each time step we assume that there are three types of clusters: 1) clusters for the existing targets, 2) clusters in the neighbors of the existing targets and 3)



Fig. 3. Background subtraction results for certain frames of sequence “S1-T1-C” of the PETS2006 dataset: codebook background subtraction is implemented to separate the foreground pixels from the background. (a) Frame 85 (b) Frame 292 (c) Frame 327 (d) Frame 1038.

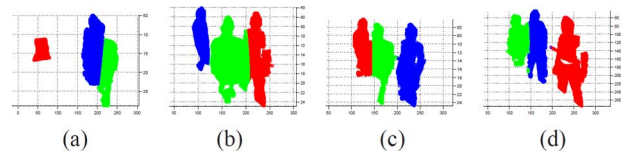


Fig. 4. Clustering results for certain frames of sequence “seq45-3p-1111_cam3_divx_audio” of the AV16.3 dataset: First, second and third clusters are represented by blue, red and green colors respectively (a) target 2 starts occluding target 1 (b) target 2 appearing again after occlusion (c) target 2 is approaching target 3 (d) target 3 comes out of occlusion. (a) Frame 255 (b) Frame 278 (c) Frame 288 (d) Frame 313.

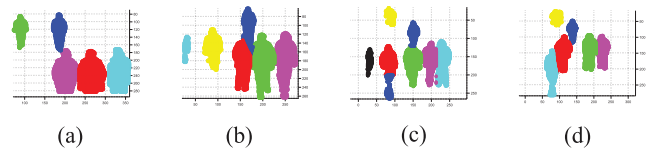


Fig. 5. Clustering results for certain frames of sequence “ThreePastShop2cor” of the CAVIAR dataset: clusters 1, 2, 3, 4, 5, 6 and 7 are represented by blue, red, green, magenta, cyan, yellow and black colors respectively. In frame 334, we have 8 clusters and hence the 8th cluster is again represented by blue. (a) Frame 334 (b) Frame 440 (c) Frame 476 (d) Frame 524.

clusters near boundaries of the field of view. Due to the Dirichlet prior over the mixing coefficient and by setting $\alpha_o = 0.6$ the clustering algorithm converges automatically to the minimum possible number of clusters. The means \mathbf{m}_o of clusters are initialized with the tracked location of existing targets and hypothesized location of new targets. Other prior parameters are defined as: $v_o = 3$ and $\beta_o = 1$. However, a full study about sensitivity of the system to the choice of different parameters is beyond the scope of the paper.

The prior parameter \mathbf{W}_o used in (56) determines the human shape which is modeled as an ellipse. It is defined with the help of the following equation

$$\mathbf{W}_o = (\mathbf{U})^T * \begin{bmatrix} l1 & 0 \\ 0 & l2 \end{bmatrix} * \mathbf{U}, \quad (68)$$

where $l1$ and $l2$ are equatorial radii of the ellipse which models the human shape. The equatorial radii $l1$ and $l2$ are set to 500 and 300 respectively while U is defined as

$$\mathbf{U} = \begin{bmatrix} \cos(\pi/2) & -\sin(\pi/2) \\ -\sin(\pi/2) & \cos(\pi/2) \end{bmatrix}. \quad (69)$$

The clustering results for a few of the video frames are shown in Figs. 4–6. Blue, red, green, magenta, cyan, yellow and black represent first, second, third, fourth, fifth, sixth and seventh clusters respectively. If there are more than 7 clusters we repeat the color scheme.

These figures show that the clustering performs well and clusters do not contain regions of multiple targets even when targets

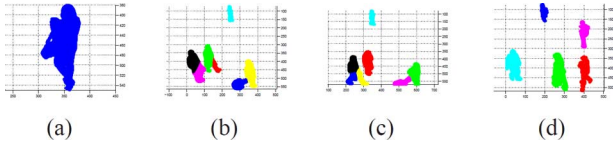


Fig. 6. Clustering results for certain frames of sequence “S1-T1-C” of the PETS2006 dataset: clusters 1, 2, 3, 4, 5, 6 and 7 are represented by blue, red, green, magenta, cyan, yellow and black colors respectively. (a) Frame 85 (b) Frame 292 (c) Frame 327 (d) Frame 1038.

are in close interaction or partially occlude each other. To eliminate the extra foreground pixels due to the reflections, the small clusters consisting of less than 100 pixels are deleted.

C. Data Association and Occlusion Handling

For data association the ten best hypotheses of joint association event ψ_k are considered which are obtained with the help of Murty's algorithm [29]. The Bhattacharyya distance between the color histograms of cluster q and target i is used to calculate the distance $d^i(H_{ref}^i, H_k^q)$

$$d^i(H_{ref}^i, H_k^q) = \sqrt{1 - \rho(H_{ref}^i, H_k^q)}, \quad (70)$$

where H_{ref}^i is the reference histogram which is created by using the cluster associated with target i at the time step when the target i first appears in the video. H_k^q is the histogram created for cluster q at the current time step k and $\rho(H_{ref}^i, H_k^q)$ is the Bhattacharyya coefficient

$$\rho(H_{ref}^i, H_k^q) = \sum_{g=1}^G \sqrt{H_{ref}^{i,g} H_k^{q,g}}, \quad (71)$$

where G represents the number of histogram bins and we have used $16 \times 16 \times 16$ color histograms bins.

A tracking algorithm with only a JPDAF, without the variational Bayesian clustering and without a social force model fails to identify an object during close interactions. In the proposed algorithm these tracking failures are overcome with the help of the proposed data association technique. Sometimes, due to the varying shape of the targets, the clustering stage may produce more than one cluster per target. Therefore, the proposed data association technique assigns multiple clusters to every target with some association probability.

D. Variable Number of Targets Results

The robustness of the algorithm for estimating the correct number of targets in a video sequence is compared with the framework developed in [4]. In [4] the number of targets is calculated on the basis of only the number of clusters. Figs. 7 and 8 present a comparison of number of targets and number of clusters in video frames.

It is apparent from Figs. 7 and 8 that the number of targets does not always match the number of clusters and hence it is difficult to train the system to estimate the number of targets on the basis of the number of clusters. In the proposed algorithm, instead of using the number of clusters we have exploited the size and location of clusters to estimate the number of targets on all the sequences. This leads to accurate estimation of the

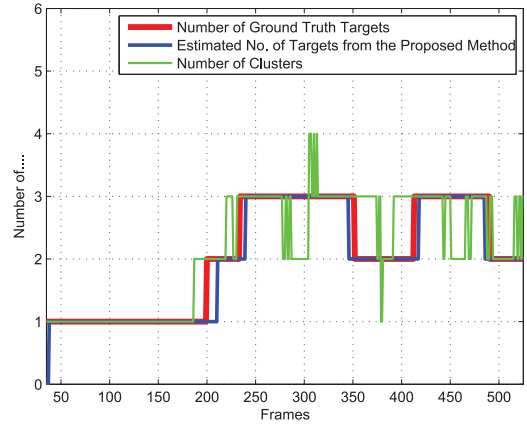


Fig. 7. The graph shows the actual number of targets, estimated number of targets by using the proposed method and the number of clusters as a function of video frames for sequence “seq45-3p-1111_cam3_divx_audio” of the AV16.3 dataset.

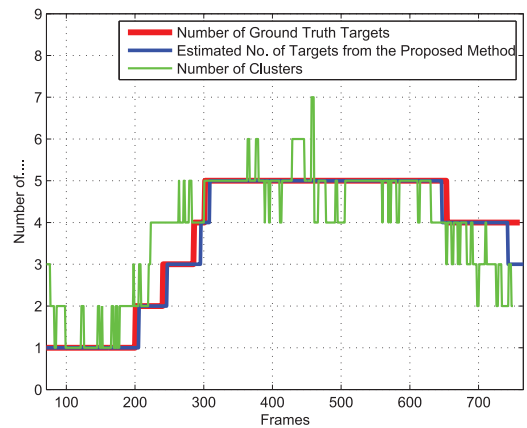


Fig. 8. The graph shows the actual number of targets, estimated number of targets by using the proposed method and the number of clusters as a function of video frames for sequence “ThreePastShop2cor” of the CAVIAR dataset.

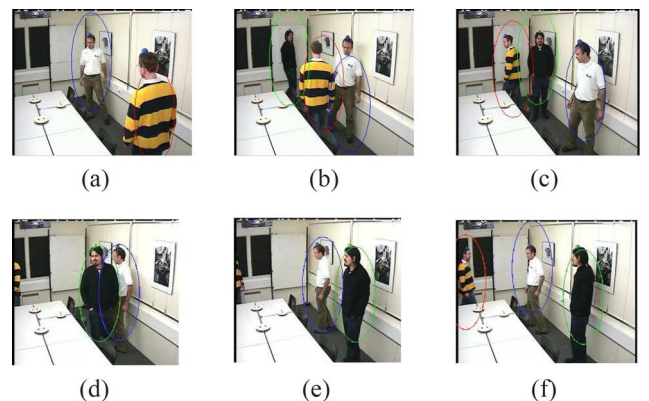


Fig. 9. Tracking results for certain frames of sequence “seq45-3p-1111_cam3_divx_audio” of the AV16.3 dataset with a variable number of targets while handling complex occlusions. (a) tracking of two targets, (b) and (f) show that the algorithm successfully initializes the new tracker, (c) shows that the tracker is deleted when the target leaves the room, (b)-(d) show successful occlusion handling. (a) Frame 225 (b) Frame 278 (c) Frame 320 (d) Frame 375 (e) Frame 405 (f) Frame 420.

number of targets and is demonstrated on Figs. 7 and 8 for two different data sets and in Figs. 9–11.

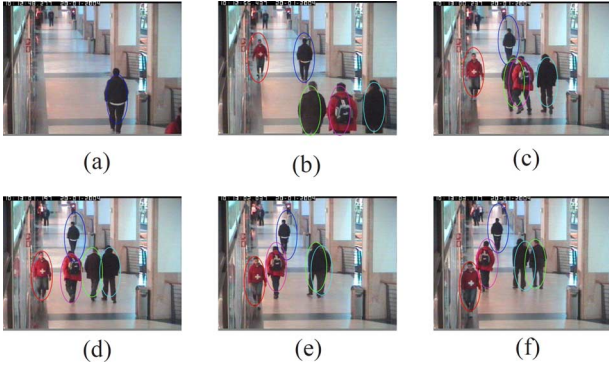


Fig. 10. Tracking results for certain frames of sequence “ThreePastShop2cor” of the CAVIAR dataset : the proposed tracking algorithm can successfully track a variable number of targets while handling complex occlusions. (a) Frame 152 (b) Frame 334 (c) Frame 452 (d) Frame 476 (e) Frame 497 (f) Frame 524.

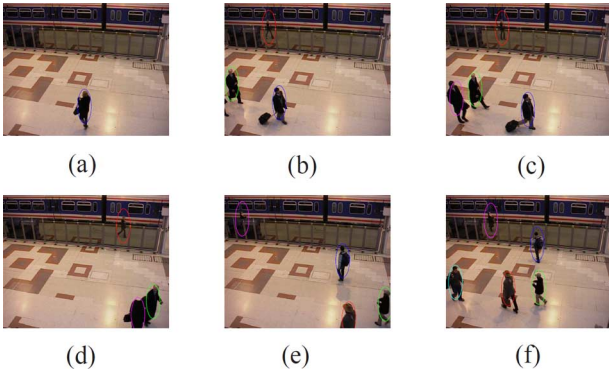


Fig. 11. Tracking results for certain frames of sequence “S1-T1-C” of the PETS2006 dataset: the proposed tracking algorithm can successfully track a variable number of targets while handling complex occlusions. (a) Frame 85 (b) Frame 273 (c) Frame 292 (d) Frame 379 (e) Frame 996 (f) Frame 1038.

E. Tracking Results

A minimum mean square error (MMSE) particle filter is used to estimate the states of the multiple targets. When a new target enters the room, the algorithm automatically initializes it with the help of the data association results by using the mean value of the cluster assigned to that target. Similarly, the algorithm removes the target which leaves the room. The particle size N_s is chosen to be equal to 60 and the number S of social links are updated at every time step. Table I shows the values of the other parameters used in the social force model. The prediction step is performed by the social force model, described in Section II-A.

The tracking results for a few selected frames from AV16.3, CAVIAR and PETS2006 datasets are given in Figs. 9–11 respectively. Blue, red, green, magenta and cyan ellipses represent first, second, third, fourth and fifth targets, respectively. Fig. 9 shows that the algorithm has successfully initialized new targets in frames 278 and 420. In frame 278 it can be seen that the algorithm can cope with the occlusions between target 1 and target 2. Frame 320 show that the algorithm has successfully handled the occlusion between targets 2 and 3. In frame 375 we can see that target 2 has left the room and the algorithm has automatically removed its tracker, which is started again when the target has returned back in frame 420. Success in dealing with occlusions between targets 3 and 1 can be seen in frames 375 and 405.

Results in Fig. 10 demonstrate that new targets appearing in frame 334 are initialized. In frame 476 it can be seen that the tracker has solved the occlusion between target three and four. Similarly, frame 524 shows that the tracker copes with occlusion between target three and five. A similar behavior is also observed in Fig. 11.

1) *CLEAR MOT Matrices*: The tracking performance of the proposed algorithm is measured based on two different performance measures: CLEAR multiple object tracking (MOT) matrices [37] and OSPAMT [38]. The performance is compared with the methods proposed in [4], [1] and [35].

The CLEAR MOT measure includes the multiple object tracking precision (MOTP) matrix, miss detection rate, false positive rate and mismatch rate.

Let $\mathbf{G}_k = [(\mathbf{g}_k^1)^T, \dots, (\mathbf{g}_k^i)^T, \dots, (\mathbf{g}_k^N)^T]^T$ be the ground truth at time $k \in [k_s, k_e]$, where k_s and k_e are respectively the starting and ending points of the observation interval. Each ground truth vector component $\mathbf{g}_k^i = (\mathbf{p}_k^i, I_i)$ contains the actual position and identity of the target i . Similarly, $\mathbf{O}_k = [(\mathbf{o}_k^1)^T, \dots, (\mathbf{o}_k^i)^T, \dots, (\mathbf{o}_k^N)^T]^T$ represents the output of the tracking algorithm at time k , where each $\mathbf{o}_k^i = (\hat{\mathbf{p}}_k^i, \hat{I}_i)$ represents the estimated location vector and identity variable of target i . At every time k the error is defined as:

- Missed detections corresponding to the number of missed targets, calculated based on the difference between the ground truth and the estimates from the developed technique.
- Mismatch corresponding to the number of targets which have given a wrong identity.
- False positives corresponding to the estimated target locations which are not associated with any of the targets.

The threshold is set at 45 cm, which is the distance between the estimated position of a target and its ground truth beyond which it is considered as a missed detection.

If we assume that m_k , mme_k , fp_k and gt_k are respectively the total number of missed detections, mismatches, false positives and ground truths at time k , the errors are calculated as [37]

$$\bar{m} = \frac{\sum_k m_k}{\sum_k gt_k}, \overline{mme} = \frac{\sum_k mme_k}{\sum_k gt_k}, \bar{fp} = \frac{\sum_k fp_k}{\sum_k gt_k}. \quad (72)$$

The precision is calculated as [37]

$$MOTP = \frac{\sum_{k,i} d_{i,k}}{\sum_{k,i} c_k}, \quad (73)$$

where $d_{i,k}$ is the distance between the estimated location and the ground truth location of target i and c_k is the total number of matches found at time step k . Table II presents the performance result of the proposed algorithm and the tracking algorithms proposed in [4], [1] and [35]. The performance results are obtained by using 45 sequences from AV16.3, CAVIAR and PETS2006 datasets. All three datasets have different environments and backgrounds. The results show that the proposed algorithm has significantly improved performance compared with other techniques [4], [1] and [35]. Table II shows that there is a significant reduction in missed detections, mismatches and false positives. The reduction in the missed detections is mainly due to the proposed clustering based approach along with the new

TABLE II
COMPARISON OF PERFORMANCE OF THE PROPOSED TRACKING ALGORITHM
WITH THE APPROACH OF [4], [1] AND [35]

Method	$\bar{m}(\%)$	$\overline{mm\bar{e}}(\%)$	$\bar{f}p(\%)$
Tracking algorithm [1]	6.41	9.32	3.92
Tracking algorithm [35]	9.38	5.15	5.08
Tracking algorithm [4]	5.41	4.57	1.95
Proposed algorithm	5.04	1.31	0.33

TABLE III
COMPARISON OF PRECISION OF THE PROPOSED TRACKING ALGORITHM
WITH THE APPROACH OF [4], [1] AND [35]

Method	Dataset	MOTP(cm)
Tracking algorithm [1]	AV16.3	9.07
	CAVIAR	11.15
	PETS2006	11.02
Tracking algorithm [35]	AV16.3	7.12
	CAVIAR	9.45
	PETS2006	8.99
Tracking algorithm [4]	AV16.3	5.85
	CAVIAR	7.13
	PETS2006	7.01
Proposed algorithm	AV16.3	3.97
	CAVIAR	5.52
	PETS2006	5.13

social force model based estimation technique. A simple particle filter based algorithm without any interaction model proposed by [35] shows the worst results mainly due to the highest number of mismatches. The algorithm proposed in [1] shows better results because of a better interaction model. However, the number of missed detections, mismatches and false positives are higher than that of [4]. This is because [1] does not propose a solution when two targets partially occlude each other or appear again after full occlusion. The technique proposed in [4] gives improved results due to the clustering and the JPDAF however fails when targets reappear after occlusion. This is due to the fact that only location information is used for data association and a simple constant velocity model is used for state prediction. Our proposed algorithm improves the tracking accuracy by reducing the number of missed detections, mismatches and false positives. Reduction in the number of missed detections is mainly due to the proposed particle filter based force model for particles predictions, while the proposed data association technique reduces the chances of mismatches. This improvement is thanks to the utilization of both feature information of targets along with their locations together with the location of clusters.

There is a significant 3.26% reduction in the wrong identifications which has improved the overall accuracy of the tracker. Average precision results for the video sequences from AV16.3, CAVIAR and PETS2006 data sets are shown in Table III.

Precision plots for two video sequences against the video frame are shown in Fig. 12. Results show that the MOTP remains less than 4cm for most of the frames. It increases to 12cm for the frames where occlusions occur but it again drops when targets emerge from occlusion.

2) *OSPAMT*: A new performance metric, optimal subpattern assignment metric for multiple tracks (OSPAMT) [38] is also used to evaluate the performance of the proposed technique and approach of [4]. The OSPAMT metric calculates the localization

distance $\mathcal{D}_k^{Loc}(\omega, \omega')$ between a set of true tracks ω and a set of estimated tracks ω' at time step k as follows

$$\mathcal{D}_k^{Loc}(\omega, \omega') = \left[\frac{1}{n_t} \sum_{i=1}^{|\omega|} d_k(\tau_i^\omega, \omega') \right]^{1/p}, \quad (74)$$

where $n_t = \max\{n_t^\omega, n_t^{\omega'}\}$ and n_t^ω and $n_t^{\omega'}$ are the number of targets in true and estimated set of tracks, respectively, at time step k . The distance $d_t(\tau_i^\omega, \omega')$ is between the i^{th} true track and the set of estimated tracks ω' , complete explanation of which can be found in Section IV of [38]. The distance $\mathcal{D}_k^{Card}(\omega, \omega')$ at time index k is calculated as

$$\mathcal{D}_k^{Card}(\omega, \omega') = \left[\frac{1}{n_t} \mathcal{S}_k \right]^{1/p}, \quad (75)$$

where \mathcal{S}_k is defined as follows

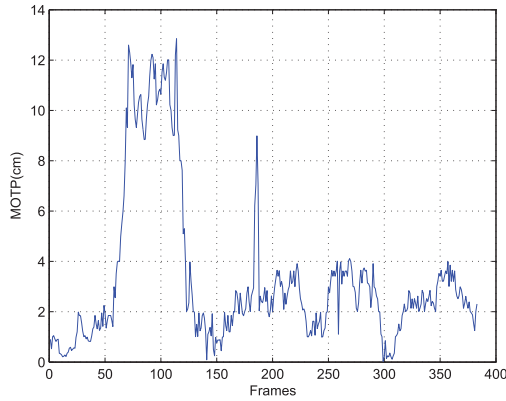
$$\mathcal{S}_k = \sum_{i=1}^{|\omega|} |\tau_i^\omega(k)| \max\{\bar{n}_{k,i}^\lambda - 1, 0\} (\Delta^p + c^p) + c^p (n_k - \sum_{i=1}^{|\omega|} \bar{n}_{k,i}^\lambda |\tau_i^\omega(k)|), \quad (76)$$

where $\bar{n}_{k,i}^\lambda$ is the number of targets at time k in ω' assigned to target i in ω , λ represents an assignment between tracks in ω' and the tracks in ω , Δ is the assignment parameter, c is the cutoff parameter and $p = 2$. Details about all these parameters and the OSPAMT metric can be found in [38]. For results in Figs. 13 and 14 we have used $c = 80$ and $\Delta = 10$. Note that $\mathcal{D}_k^{Card}(\omega, \omega')$ is basically a distance between the objects. Figs. 13 and 14 show the OSPAMT distances, (75) and (76), respectively, plotted against the video frame index. Fig. 13 shows low localization errors compared to the technique proposed in [4]. Fig. 14 indicates a peak dynamic error when a new target enters the field of view which is between frame numbers 63 and 65.

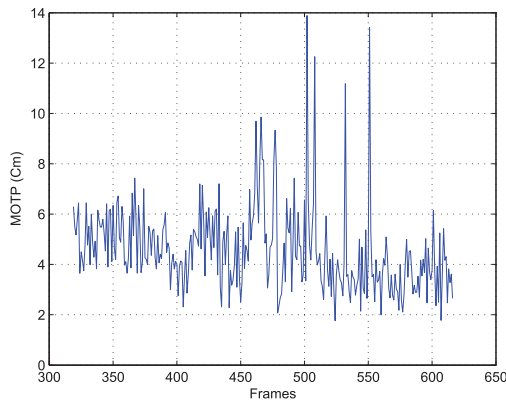
F. Computational Complexity

A variable measurement size technique reduces significantly the computational complexity. The measurement size is increased or decreased on the basis of the distance between the targets. Downsampling of foreground pixels is performed when the distance between targets becomes 80 cm or less. The decrease in the measurement size reduces the time and number of iterations for reaching clustering convergence which results in improved computational complexity. This is shown in Table IV with the help of a few selected frames from sequence “seq45-3p-1111_cam3_divx_audio” of the AV16.3 dataset.

Table IV also demonstrates that the proposed technique reduces the number of iterations needed for achieving convergence. The reduction in the number of iterations for convergence improves the run-time of the tracking algorithm. The average run-time (calculated using 45 video sequences from three datasets) of the proposed algorithm due to the reduction in number of iterations is 0.587 seconds per frame, as compared to the run-time without measurement reduction which is 2.611 seconds per frame. The run-time of the approach of [4] is 1.417 seconds per frame. This run-time comparison is



(a)



(b)

Fig. 12. Precision plot against the video frames of sequences: (a) “seq45–3p–1111_cam3_divx_audio” of the AV16.3 dataset (b) “ThreePastShop2cor” of the CAVIAR dataset.

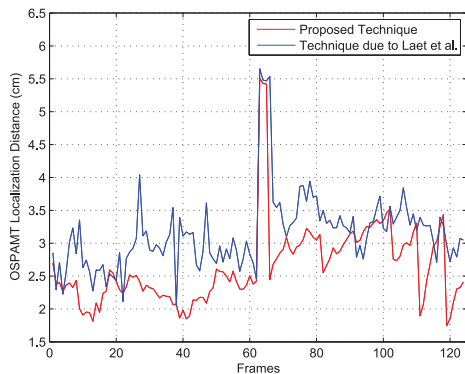


Fig. 13. OSPAMT localization distance plot against the video frames of sequence “ThreePastShop2cor” of the CAVIAR dataset.

made by implementing the algorithms on MATLAB (version R2012a) with a 3.2GHz I5 processor.

G. Summary

Successful background subtraction is achieved with the help of the codebook background subtraction technique and the results are shown in Figs. 1–3. Grouping foreground pixels into clusters with the help of the proposed variational Bayesian technique is presented in Figs. 4–6. Tracking results are shown in Figs. 9–11. The variational Bayesian clustering improves

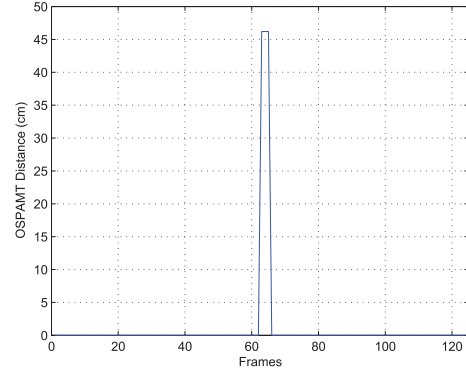


Fig. 14. OSPAMT distance plot against the video frames of sequence “ThreePastShop2cor” of the CAVIAR dataset.

TABLE IV
COMPARISON OF CONVERGENCE ITERATIONS FOR
DIFFERENT MEASUREMENT SIZES

Frame No.	Minimum Distance	Measurement Size		Convergence Iterations	
		Original	Reduced	Original	Reduced
210	227.33	11439	1255	193	15
218	201.01	14496	1614	389	20
221	186.85	15887	1780	282	25
225	159.64	17792	1966	279	29
228	137.60	17819	1964	334	40

the overall tracking results especially during close inter-target occlusions. The JPDAF based technique proposed in [4] does not always assign the appropriate clusters to the targets and hence results in tracking failures while solving the complex occlusions.

The graphical results shown in Figs. 7 and 8 indicate that estimating the number of targets on the basis of the number of clusters does not produce accurate results because the number of clusters does not always vary with the number of targets. Figs. 9–11 show that accurate target number estimation results are achieved by the proposed technique thanks to exploiting the sizes and locations of clusters, and the estimated state of the targets at the previous time step.

The results in Tables II and III confirm that the overall performance of the proposed tracking algorithm results in an improvement as compared with the recently proposed techniques [4], [1] and [35].

VII. DISCUSSION

The quality of the clustering results influences the proposed data association technique. We initialize the clusters on the basis of the estimated locations of targets and by taking into account merging and splitting of targets. The motion of people is predicted with the social force model. This yields accurate clustering results in scenarios where one cluster contains regions of multiple targets. Therefore, the proposed data association technique performs well even during close interactions of targets. It associates multiple clusters to targets with the correct proportion of probability. These association results help in achieving highly accurate tracking of multiple targets even during their close interactions and partial occlusions. The extensive evaluation of the proposed technique on 45 sequences from three very different datasets validates the tracking accuracy of the proposed algorithm in such scenarios.

A further improvement in accuracy of cluster-to-target association can be achieved by calculating joint associations between multiple clusters. A several-to-one or several-to-several correspondence strategy can be adopted. However, for these strategies the association problem is combinatorial and complexity may become intractable because it will grow geometrically. The cluster-to-target association can also be improved by considering the regions of multiple targets in a cluster as unique identities and associating those regions to respective targets [27].

VIII. CONCLUSIONS

A learned variational Bayesian clustering and social force model algorithm has been presented for multi-target tracking by using multiple measurements originating from a target. An improved data association technique is developed that exploits the clustering information to solve complex inter-target occlusions. The algorithm accurately and robustly handles a variable number of targets. Its is compared extensively with recent techniques [1], [4] and [35].

APPENDIX

All the variables in the following derivation are for time index k . Therefore, for simplicity, we omit the index k from the equations. According to (42) we can write

$$\ln \mathcal{Q}^*(\mathbf{B}) = E_{\mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\Sigma}}[\ln p(\tilde{\mathbf{y}}, \mathbf{B}, \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + Const \quad (77)$$

and using the decomposition of (37) we have

$$\ln \mathcal{Q}^*(\mathbf{B}) = E_{\mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\Sigma}}[\ln p(\tilde{\mathbf{y}}|\mathbf{B}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \ln p(\mathbf{B}|\mathbf{C}) + \ln p(\mathbf{C}) + \ln p(\boldsymbol{\mu}|\boldsymbol{\Sigma}) + \ln p(\boldsymbol{\Sigma})] + Const. \quad (78)$$

By keeping only those terms which are depending on \mathbf{B} we get $\ln \mathcal{Q}^*(\mathbf{B}) = E_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}[\ln p(\tilde{\mathbf{y}}|\mathbf{B}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] + E_{\mathbf{C}}[\ln p(\mathbf{B}|\mathbf{C})] + Const.$ By using (32) and (33) we can write

$$\ln \mathcal{Q}^*(\mathbf{B}) = \sum_{j=1}^M \sum_{q=1}^{\kappa} b^{j,q} \left[-\frac{D}{2} \ln(2\pi) - \frac{1}{2} E_{\boldsymbol{\Sigma}}[\ln |\boldsymbol{\Sigma}^q|] - \frac{1}{2} E_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}[(\tilde{\mathbf{y}}^j - \boldsymbol{\mu}^q)^T (\boldsymbol{\Sigma}^q)^{-1} (\tilde{\mathbf{y}}^j - \boldsymbol{\mu}^q)] + E_{\mathbf{C}} \ln(C^q) \right] + Const.$$

Considering that

$$\ln \rho^{j,q} = -\frac{D}{2} \ln(2\pi) - \frac{1}{2} E_{\boldsymbol{\Sigma}}[\ln |\boldsymbol{\Sigma}^q|] - \frac{1}{2} E_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}[(\tilde{\mathbf{y}}^j - \boldsymbol{\mu}^q)^T (\boldsymbol{\Sigma}^q)^{-1} (\tilde{\mathbf{y}}^j - \boldsymbol{\mu}^q)] + E_{\mathbf{C}} \ln(C^q) \quad (79)$$

we get $\ln \mathcal{Q}^*(\mathbf{B}) = \prod_{j=1}^M \prod_{q=1}^{\kappa} (\rho^{j,q})^{b^{j,q}} + Const$, where according to [32]

$$\begin{aligned} E_{\boldsymbol{\Sigma}}[\ln |\boldsymbol{\Sigma}^q|] &= \sum_{\tilde{i}}^D \tilde{\psi} \left(\frac{v+1-\tilde{i}}{2} \right) + D \ln 2 + \ln |\mathbf{W}_q|, \\ E_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}[(\tilde{\mathbf{y}}^j - \boldsymbol{\mu}^q)^T (\boldsymbol{\Sigma}^q)^{-1} (\tilde{\mathbf{y}}^j - \boldsymbol{\mu}^q)] &= D\beta_q^{-1} + v_q (\tilde{\mathbf{y}}^j - \mathbf{m}_q)^T \mathbf{W}_q (\tilde{\mathbf{y}}^j - \mathbf{m}_q), \\ E_{\mathbf{C}} \ln(C^q) &= \tilde{\psi}(\alpha^q) - \tilde{\psi}(\tilde{\alpha}), \end{aligned} \quad (80)$$

where $\tilde{\alpha} = \sum_{q=1}^{\kappa} \alpha^q$ and $\tilde{\psi}(\cdot)$ is the digamma function. We can write: $\mathcal{Q}^*(\mathbf{B}) \propto \prod_{j=1}^M \prod_{q=1}^{\kappa} (\rho^{j,q})^{b^{j,q}}$. Requiring that this distribution be normalized

$$r^{j,q} = \frac{\rho^{j,q}}{\sum_{q=1}^{\kappa} \rho^{j,q}}. \quad (81)$$

For a detailed proof please refer to Appendix of the long version of the paper available online at *arXiv*.

ACKNOWLEDGMENT

The authors greatly appreciate the constructive comments of the Associate Editor and anonymous reviewers to help improve this paper.

REFERENCES

- [1] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1819, Nov. 2005.
- [2] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filter for Tracking Applications*. Boston, MA, USA: Artech House, 2004.
- [3] S. M. Naqvi, M. Yu, and J. A. Chambers, "A multimodal approach to blind source separation of moving sources," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 5, pp. 895–910, Oct. 2010.
- [4] T. D. Laet, H. Bruyninckx, and J. D. Schutter, "Shape-based online multi-target tracking and detection for targets causing multiple measurements: Variational Bayesian clustering and lossless data association," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2477–2491, Dec. 2011.
- [5] M. Montemerlo, W. Whittaker, and S. Thrun, "Particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 695–701.
- [6] S. Velipasalar and W. Wolf, "Multiple object tracking and occlusion handling by information exchange between uncalibrated cameras," in *Proc. IEEE Int. Conf. Image Process.*, 2005, vol. 2, pp. 418–421.
- [7] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, 2004, vol. 2, pp. 406–413.
- [8] A. M. Elgammal, R. Duraiswami, and L. S. Davis, "Efficient non-parametric adaptive color modeling using fast Gauss transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2001, vol. 2, pp. 563–570.
- [9] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential Monte Carlo tracking by fusing multiple cues in video sequences," *Image Vis. Comput.*, vol. 25, no. 8, pp. 1217–1227, 2007.
- [10] J. Vermaak, S. J. Godsill, and P. Perez, "Monte Carlo filtering for multi-target tracking and data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 309–332, Jan. 2005.
- [11] S. M. Naqvi, W. Wang, M. S. Khan, M. Barnard, and J. A. Chambers, "Multimodal (audio-visual) source separation exploiting multi-speaker tracking, robust beamforming, and time-frequency masking," *IET Signal Process.*, vol. 6, pp. 466–477, 2012.
- [12] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, 2003.
- [13] B. Habtmariam, R. Tharmarasa, T. Thayaparan, M. Mallick, and T. Kirubarajan, "A multiple-detection joint probabilistic data association filter," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 461–471, Jun. 2013.
- [14] H. Benoudnine, M. Keche, A. Ouamri, and M. S. Woolfson, "New efficient schemes for adaptive selection of the update time in the IMM-PDAF," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 197–214, Jan. 2012.
- [15] S. K. Pang, J. Li, and S. J. Godsill, "Detection and tracking of coordinated groups," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 1, pp. 472–502, Jan. 2011.
- [16] S. Ali, K. Nishino, D. Manocha, and M. Shah, *Modeling, Simulation and Visual Analysis of Crowds*. New York, USA: Springer, 2013.
- [17] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, 1995.
- [18] R. Mazzon and A. Cavallaro, "Multi-camera tracking using a multi-goal social force model," *Neurocomput.*, vol. 100, pp. 41–50, 2013.

- [19] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 261–268.
- [20] F. Septier, S. K. Pang, A. Carmi, and S. Godsill, "On MCMC based particle methods for Bayesian filtering: Application to multi-target tracking," in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2009, pp. 360–363.
- [21] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Boston, MA, USA: Artech House, 2007.
- [22] B. Ristic, J. Sherrah, and A. F. García-Fernández, "Performance evaluation of random set based pedestrian tracking algorithms," in *Proc. Intell. Sens., Sens. Netw. Inf. Process.*, 2013, pp. 300–305.
- [23] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [24] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Int. J. Real-Time Imag.*, vol. 11, pp. 172–185, 2005.
- [25] X. Yan, I. Kakadiaris, and S. Shah, "Modeling local behavior for predicting social interactions towards human tracking," *Pattern Recogn.*, vol. 47, no. 4, pp. 1626–1641, 2014.
- [26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge University Press, 2001.
- [27] Y. Ma, Q. Yu, and I. Cohen, "Target tracking with incomplete detection," *Comput. Vis. Image Understanding*, vol. 113, no. 4, pp. 580–587, 2009.
- [28] Y. Bar-Shalom and T. Fortman, *Tracking and Data Association*. London, U.K.: Academic Press, 1988.
- [29] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Oper. Res.*, vol. 16, no. 3, pp. 682–687.
- [30] I. J. Cox, "A review of statistical data association techniques for motion correspondence," *Int. J. Comput. Vis.*, vol. 10, pp. 53–66, 1993.
- [31] A. Bhattacharayya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–110, 1943.
- [32] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer, 2006.
- [33] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *Int. J. Robot. Res.*, vol. 22, no. 2, pp. 99–116, 2003.
- [34] W. Ng, J. Li, S. Godsill, and S. K. Pang, "Multi-target initiation, tracking and termination using Bayesian Monte Carlo methods," *Comput. J.*, vol. 50, no. 6, pp. 674–693, Nov. 2007.
- [35] J. Czyz, B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of color objects," *Image Vis. Comput.*, vol. 25, no. 8, pp. 1271–1281, 2007.
- [36] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. A. Chambers, "Posture recognition based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1274–1286, Nov. 2012.
- [37] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.
- [38] T. Vu and R. Evans, "A new performance metric for multiple target tracking based on optimal subpattern assignment," in *Proc. Int. Conf. Inf. Fusion*, Jul. 2014, pp. 1–8.



Ata Ur-Rehman received the B.Eng. degree in electronic engineering from Air University, Islamabad, Pakistan, in 2006, and the M.Sc. degree (Hons.) in digital communication systems and the Ph.D. degree in signal processing from Loughborough University, Loughborough, U.K., in 2010 and 2014, respectively.

He then joined the University of Engineering and Technology, Lahore, Pakistan, as a Lecturer/Lab Engineer in 2007. He moved to Loughborough University in 2009. Since 2013, he has been a Postdoctoral Research Associate in the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, U.K. His main area of research is multi-target tracking.



Syed Mohsen Naqvi (S'07–M'09–SM'14) received the Ph.D. degree in signal processing from Loughborough University, Loughborough, U.K., in 2009.

He was a Postdoctoral Research Associate on the EPSRC U.K.-funded projects and REF Lecturer from 2009 to 2015. He is a Lecturer in Signal and Information Processing at the School of Electrical and Electronic Engineering, Newcastle University, Newcastle, U.K. and a Fellow of the Higher Education Academy. His research interests include multimodal processing for human behavior analysis,

multi-target tracking, and source separation—all for machine learning.



Lyudmila Mihaylova (M'98–SM'08) is Associate Professor (Reader in Advanced Signal Processing and Control) in the Department of Automatic Control and Systems Engineering at the University of Sheffield, Sheffield, U.K. She is an Associate Editor of the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS and *Signal Processing*. Her research is in the areas of machine learning and autonomous systems with various applications, such as navigation, surveillance, and sensor network systems.



Jonathon A. Chambers (S'83–M'90–SM'98–F'11) received the Ph.D. and D.Sc. degrees in signal processing from the Imperial College of Science, Technology and Medicine (Imperial College London), London, U.K., in 1990 and 2014, respectively.

Currently, he is a Professor of Signal and Information Processing within the School of Electrical and Electronic Engineering, Newcastle University, Newcastle, U.K., and a Fellow of the Royal Academy of Engineering, London, U.K. He also served as an Associate Editor for the IEEE TRANSACTIONS ON

SIGNAL PROCESSING over the periods 1997 to 1999 and 2004 to 2007, and as a Senior Area Editor from 2011 to 2015.