Online Visual Multi-Object Tracking via Labeled Random Finite Set Filtering
Du Yong Kim, Ba-Ngu Vo, Member, IEEE, and Ba-Tuong Vo, Member, IEEE

Abstract—This paper proposes an online visual multi-object tracking algorithm using a top-down Bayesian formulation that seamlessly integrates state estimation, track management, clutter rejection, occlusion and mis-detection handling into a single recursion. This is achieved by modeling the multi-object state as labeled random finite set and using the Bayes recursion to propagate the multi-object filtering density forward in time. The proposed filter updates tracks with detections but switches to image data when mis-detection occurs, thereby exploiting the efficiency of detection data and the accuracy of image data. Furthermore the labeled random finite set framework enables the incorporation of prior knowledge that mis-detections of long tracks which occur in the middle of the scene are likely to be due to occlusions. Such prior knowledge can be exploited to improve occlusion handling, especially long occlusions that can lead to premature track termination in on-line multi-object tracking. Tracking performance are compared to state-of-the-art algorithms on well-known benchmark video datasets.

Index Terms—online multi-object tracking, sequential Bayesian estimation, random finite set, video surveillance

1 INTRODUCTION
Tracking multiple objects is an indispensable task in many applications and is an important topic in computer vision research [cite]. In a multiple object setting, not only do the states of the objects vary with time, but the number of objects also changes due to objects appearing and disappearing. The aim of visual Multiple Object Tracking (MOT) is to jointly estimate the time-varying number of objects and their trajectories from a stream of noisy images. MOT can be formulated as a probabilistic [8], [11], [12], [15], [17], or deterministic [25], [28], [29] problem. In this work we are interested in MOT solutions that compute estimates at a given time using only data up to that time. These so-called online solutions are better suited for time-critical applications.

In most visual MOT approaches, each image in the data sequence is compressed into a set of detections before a filtering operation is applied to keep track of all objects (including undetected ones) [15], [17], [43]. Typically, in the filtering module, motion correspondence or data association is first determined followed by the application of standard filtering techniques such as Kalman or particle filtering [17], [43]. The main advantage of this approach is the computational efficiency in the compression of images into relevant detections. The main disadvantage is the loss of information, in addition to mis-detection and false alarms, especially in low signal to noise ratio (SNR) applications.

An alternative approach is to by-pass the detection module and exploit the spatio-temporal information directly from the image sequence. This so-called track-before-detect (TBD) methodology is often required in tracking from low SNR imagery applications [31], [40], [41], [42], [44]. Perhaps the most well-known visual MOT algorithm without detection is BraMBLe [6]. Other visual MOT algorithms that can be categorized as track-before-detect include [7], [9] which exploit color-based observation models, [8], [10], [15], which exploit multi-modality of distributions, and [45] which uses multi-Bernoulli random finite set models. While the TBD approach minimizes information loss, it is computationally more expensive. A balance between tractability and fidelity is important in the design of the measurement model.

A critical function of a visual multi-object tracker is track management, which concerns track initiation/termination and track labeling or identifying trajectories of individual objects. Track management is more challenging for online algorithms than for batch algorithms. Usually, track initiation/termination in online MOT algorithms is performed by examining consecutive detections in time [11], [15]. However, false positives can be generated by the background, and compounded by false negatives from object occlusions and mis-detections, which can result in false tracks and lost tracks, especially in online algorithms. False negatives also cause track fragmentation in batch algorithms as reported in [25], [28], [29]. Except for the recently proposed continuous-discrete optimization [30], and network flow [32] track labels are assigned on track initiation and are maintained over time until termination. An online multi-object Bayesian filter that provides systematic track labeling using the labeled random finite set (RFS) was proposed in [37].

In this paper, we present an efficient online visual MOT algorithm that exploits the advantages of both detection-based and TBD approaches in a principled manner. In particular, we adopt a top-down formulation that seamlessly integrates state estimation, track management, clutter rejection, mis-detection and occlusion handling, into one single Bayesian recursion. Conceptually, our solution is based on modeling the multi-object state as a labeled random finite set [37] and using the Bayes recursion to
propagate the multi-object filtering density forward in time. The proposed filter updates the multi-object filtering density with detections, but switches to image data in the presence of hypothesized false negatives. This is accomplished by a hybrid multi-object likelihood function that accommodates both detection and image observations, which generalizes the standard multi-object likelihood [4] and the separable likelihood for image in [44]. This strategy exploits the efficiency of the detection-based approach which avoids updating with the entire image, while at the same time exploiting relevant information at the image level by using only small regions of the image where mis-detected objects are expected.

Generally, an online MOT algorithm would terminate a track that has not been detected over several frames. In many visual MOT applications however, it is observed that away from designated exit regions such as scene edges, the longer an object is in the scene, the less likely it is to disappear. Intuitively, this observation can be used to delay the termination of tracks that have been occluded over an extended period, so as to improve occlusion handling. The use of labeled RFS in our proposed filter provides a principled and inexpensive means to exploit this observation for improved occlusion handling.

The remainder of the paper is structured as follows. In Section 2, the Bayesian filtering formulation of the MOT problem using labeled RFS is given. The proposed solution is detailed in Section 3 and Section 4 provides comparison studies with state-of-the-art trackers to evaluate the performance of the proposed algorithm. In Section 5, concluding remarks are made.

2 BAYESIAN MULTIPLE OBJECT TRACKING

This section outlines the RFS framework for MOT that accommodates uncertainty in the number of objects, the states of the objects and their trajectories. Moreover, this framework admits direct parallels between the traditional area of stochastic filtering and MOT.

We start by outlining key concepts from the well-known Bayesian filtering paradigm for state space models in Subsection 2.1. The modeling of the multi-object state as an RFS in Subsections 2.2 enables Bayesian filtering concepts in Subsection 2.3 to be directly translated to the multi-object case. Subsection 2.4 examines the MOT problem in the presence of occlusion.

2.1 Bayesian Filtering

In the classical dynamic state estimation problem [Kay, Barshalom], the state vector $x_k \in \mathbb{X}$ (e.g. an object’s 2-D position and velocity) is assumed to evolve in time according to a Markov transition density $f_{k|k-1}(x_k|x_{k-1})$, i.e. the probability density of a transition to the state $x_k$ at time $k$ given a state $x_{k-1}$ at time $k-1$. In addition, at time $k$, the state $x_k$ generates an observation $y_k$ according to the likelihood function $g_k(y_k|x_k)$, i.e. the probability density of receiving the observation $y_k$ given the state $x_k$.

Fig. 1: 1D multi-object trajectories with labeling

All information about the state history to time $k$ is encapsulated in the posterior density:

$$p_{0:k}(x_{0:k}|y_{1:k}) \propto \prod_{j=1}^{k} g_j(y_j|x_j)f_{j|j-1}(x_j|x_{j-1})p_0(x_0).$$

where the notation $y_{t:k}$ denotes the array $(y_1, \ldots, y_k)$, and $p_0$ is the initial state prior. The filtering density $p_k(\cdot|y_{1:k})$, is a marginal of the posterior density, which can be computed recursively using the Bayes recursion:

$$p_{k|k-1}(x_k|y_{1:k-1}) = \int f_{k|k-1}(x_k|x)p_{k-1}(x|y_{1:k-1})dx,$$

(1)

$$p_k(x_k|y_{1:k}) = \frac{g_k(y_k|x_k)p_{k|k-1}(x_k|y_{1:k-1})}{\int g_k(y_k|x)p_{k|k-1}(x|y_{1:k-1})dx},$$

(2)

where $p_{k|k-1}(\cdot|y_{1:k-1})$ is called the prediction density. The particle filter is a recursive Monte Carlo approximation of the Bayes recursion while the Kalman filter is a closed form solution in the case of linear Gaussian models.

An estimator of the state is a function $\hat{x}$ that assigns the observation history $y_{1:k}$ a value $\hat{x}(y_{1:k}) \in \mathbb{X}$. A cost $C(\hat{x}(y_{1:k}), x)$ is associated with using $\hat{x}(y_{1:k})$ to estimate $x$, and the Bayes risk $R(\hat{x})$ is the expected cost over all possible realizations of the observation and state. A Bayes optimal estimator is any estimator that minimizes the Bayes risk [1]. [2]. The most common estimators are the expected a posteriori (EAP) or conditional mean and maximum a posteriori (MAP) estimators [1], [2].

2.2 Multi-object State

The RFS approach to MOT retains the same Bayesian estimation methodology in subsection 2.1. The representation of the multi-object state as a finite set enables concepts such as state space models, the Bayes recursion, and Bayes optimality to be directly translated to the multi-object realm.

To distinguish different object trajectories in a multi-object setting, each object is assigned a unique label $\ell_i$ that consists of an ordered pair $(t, i)$, where $t$ is the time of birth and $i$ is the index of individual objects born at the same time [37]. For example, if two objects appear in the scene at time 1, one is assigned label (1,1) while the other is assigned label (1,2), see Figure 1. The trajectory or track of an object is given by the sequence of states with the
same label.

Formally, the state of an object at time $k$ is a vector $x_k = (x_k, \ell_k) \in \mathbb{X} \times \mathbb{L}_k$, where $\mathbb{L}_k$ denotes the label space for objects at time $k$ (including those born prior to $k$). Note that $\mathbb{L}_k$ is given by $\mathbb{B}_k \cup \mathbb{L}_{k-1}$, where $\mathbb{B}_k$ denotes the label space for objects born at time $k$ (and is disjoint from $\mathbb{L}_{k-1}$). Suppose that there are $N_k$ objects at time $k$, with states $x_{k,1}, \ldots, x_{k,N_k}$. In the context of MOT, the collection of states, referred to as the multi-object state, is naturally represented as a finite set

$$X_k = \{x_{k,1}, \ldots, x_{k,N_k}\} \in \mathcal{F}(\mathbb{X} \times \mathbb{L}_k),$$

where $\mathcal{F}(\mathbb{X} \times \mathbb{L}_k)$ denotes the space of finite subsets of $\mathbb{X} \times \mathbb{L}_k$. We denote cardinality (number of elements) of $X$ by $|X|$ and the set of labels of $X$, $\{\ell : (x, \ell) \in X\}$, by $\mathcal{L}(X)$. Note that since the label is unique, no two objects have the same label, i.e. $\delta_X(|\mathcal{L}(X)|) = 1$. Hence $\Delta(X) \triangleq \delta_{|\mathcal{L}(X)|}$ is called the distinct label indicator.

For the rest of the paper, we follow the convention that single-object states are represented by lower-case letters (e.g. $x$, $x$), while multi-object states are represented by upper-case letters (e.g. $X$, $X$), symbols for labeled states and their distributions are bold-faced to distinguish them from unlabeled ones (e.g. $\mathbb{X}$, $\pi$, etc.), spaces are represented by blackboard bold (e.g. $\mathbb{Z}$, $\mathbb{L}$, $\mathbb{N}$, etc.). The inner product $\int f(x)g(x)dx$ is denoted by $\langle f, g \rangle$. The list of variables $X_m, X_{m+1}, \ldots, X_n$ is abbreviated as $X_{m:n}$. For a finite set $X$, its cardinality (number of elements) is denoted by $|X|$, in addition we use the multi-object exponential notation $f^X$ for the product $\prod_{x \in X} f(x)$, with $f^\emptyset = 1$. We denote a generalization of the Kroneker delta that takes arbitrary arguments such as sets, vectors, integers etc., by

$$\delta_Y[X] \triangleq \left\{ \begin{array}{ll} 1, & \text{if } X = Y \\ 0, & \text{otherwise} \end{array} \right.$$

For a given set $S$, $1_S(\cdot)$ denotes the indicator function of $S$, and $\mathcal{F}(S)$ denotes the class of finite subsets of $S$.

2.3 Multi-object Bayes filter

From a Bayesian estimation viewpoint the multi-object state is naturally modeled as an RFS or a simple-finite point process [20]. However, the space $\mathcal{F}(\mathbb{X} \times \mathbb{L}_k)$ does not inherit the Euclidean notion of probability density. In the MOT literature, Mahler’s Finite Set Statistics (FISST) notion of integration/density is used to characterize RFSs [4, 5]. This approach is mathematically consistent with measure theoretic integration/density but circumvents measure theoretic constructs [24].

At time $k$, the multi-object state $X_k$ is observed as an image $y_k$. All information on the set of object trajectories conditioned on the observation history $y_{1:k}$, is captured in the multi-object posterior density

$$\pi_{0:k}(X_{0:k}|y_{1:k}) \propto \prod_{j=1}^k g_j(y_j|X_j)f_{j|j-1}(X_j|X_{j-1})\pi_0(X_0)$$

where $\pi_0$ is the initial prior, $g_j(\cdot|\cdot)$ is the multi-object likelihood function at time $j$, $f_{j|j-1}(\cdot|\cdot)$ is the multi-object transition density to time $j$. The multi-object likelihood function encapsulates the underlying observation model while the multi-object transition density encapsulates the underlying models for motions, births and deaths of objects. Note that track management is incorporated into the Bayes recursion via the multi-object state with distinct labels.

MCMC approximations of the posterior density have been proposed in [46, 47] for detection measurements and image measurements respectively. Results on satellite imaging applications reported in [47] are very impressive. However, these techniques are still expensive and not suitable for on-line application.

For real-time tracking, a more tractable alternative is the multi-object filtering density, a marginal of the multi-object posterior. For notational compactness, herein we omit the dependence on data in the multi-object densities. The multi-object filtering density can be recursively propagated by the multi-object Bayes filter [20, 21] according to the following prediction and update

$$\pi_{k+1|k}(X_{k+1}) = \int f_{k+1|k}(X_{k+1}|X_k)\pi_k(X_k)\delta X_k,$$

$$\pi_{k+1}(X_{k+1}) = \int g_{k+1}(y_{k+1}|X_{k+1})\pi_{k+1|k}(X_{k+1})\delta X_k,$$

where the integral is a set integral defined for any function $f : \mathcal{F}(\mathbb{X} \times \mathbb{L}_k) \to \mathbb{R}$ by

$$\int f(X)\delta X = \sum_{i=0}^{\infty} \frac{1}{i!} \int f([x_1, \ldots, x_i])d(x_1, \ldots, x_i).$$

Bayes optimal multi-object estimators can be formulated by minimizing the Bayes risk as in Subsection 2.1 with ordinary integrals replaced by set integrals. One such estimator is the marginal multi-object estimator [4].

A generic particle implementation of the Bayes multi-object filter [3, 4] was proposed in [24] and applied to labeled multi-object states in [41]. The Generalized labeled Multi-Bernoulli (GLMB) filter is an analytic solution to the Bayes multi-object filter, under the standard multi-object dynamic and observation models [37].

2.3.1 Standard multi-object dynamic model

Given the multi-object state $X_k$ (at time $k$), each state $(x_k, \ell_k) \in \mathbb{X}_k$ either survives with probability $P_{S,k}(x_k, \ell_k)$ and evolves to a new state $(x_{k+1}, \ell_{k+1})$ (at time $k + 1$) with probability density $f_{k+1|k}(x_{k+1}|x_k, \ell_k)\delta \ell_k(\ell_{k+1})$ or dies with probability $1 - P_{S,k}(x_k, \ell_k)$. The set $B_{k+1}$ of new objects born at time $k + 1$ is distributed according to the labeled multi-Bernoulli (LMB)

$$\Delta(B_{k+1})\omega_{B,k+1}(\mathcal{L}(B_{k+1}))p_{B,k+1},$$

where $\omega_{B,k+1}(L) = \left[1 - r_{B,k+1}P_{B,k+1}(L) - \sum_{\ell=1}^{L-1} r_{B,k+1}(\ell)\right]^{B_{k+1}-L}$, $r_{B,k+1}(\ell)$ is the probability that a new object with label $\ell$ is born, and $p_{B,k+1}(\cdot, \ell)$ is the distribution of its kinematic state [37]. The multi-object state $X_{k+1}$ (at time $k + 1$) is the superposition of surviving objects and new born objects. It
is assumed that, conditional on $X_k$, objects move, appear and die independently of each other. The expression for the multi-object transition density $f_{k+1|k}$ can be found in [37, 38]. The standard multi-object dynamic model enables the Bayes multi-object filter to address motion, births and deaths of objects.

### 2.3.2 Standard multi-object observation model

In most applications a designated detection operation $D$ is applied to $y_k$ to produce a set of points

$$Z_k = D(y_k),$$

in some space $Z$. Since the detection process is not perfect, false positives and false negatives are inevitable. Hence some of the points of $Z_k$ correspond to some objects in the scene (not all objects are detected) while some are false positives. The most popular detection-based observation model is described in the following.

For a given multi-object state $X_k$, each $(x, \ell) \in X_k$ is either detected with probability $P_{D,k}(x, \ell)$ and generates a detection $z \in Z_k$ with likelihood $g_{D,k}(z|x, \ell)$ or missed with probability $1 - P_{D,k}(x, \ell)$. The ratio

$$\sigma_{D,k}(z|x, \ell) \triangleq \frac{g_{D,k}(z|x, \ell)}{\kappa_k(z)}$$

can be interpreted as the detection signal to noise ratio (SNR). The multi-object observation $Z_k$ is the superposition of the observations from detected objects and Poisson clutter with intensity $\kappa_k$.

Assuming that, conditional on $X_k$, detections are independent of each other and clutter, the multi-object likelihood function is given by [4, 37, 38]

$$g_k(y_k|X_k) \propto \sum_{\theta \in \Theta_k(I)} \prod_{(x,\ell) \in X_k} \psi_{D(y_k)}^{(\theta(\ell))}(x,\ell)$$

where: $\Theta_k(I)$ is the set of positive 1-1 maps $\theta : I \rightarrow \{0:Z_k\}$, i.e. maps such that no two distinct arguments are mapped to the same positive value; and

$$\psi_{D(y_k)}^{(\theta(\ell))}(x,\ell) = \begin{cases} P_{D,k}(x,\ell) \sigma_{D,k}(z|x,\ell), & \text{if } j = 1:M \\ 1 - P_{D,k}(x,\ell), & \text{if } j = 0 \end{cases}$$

The map $\theta$ specifies which objects generated which detections, i.e. object $\ell$ generates detection $z_{\theta(\ell)} \in Z_k$, with undetected objects assigned to 0. The positive 1-1 property means that $\theta$ is 1-1 on $\{\ell : \theta(\ell) > 0\}$, the set of labels that are assigned positive values, and ensures that any detection in $Z_k$ is assigned to at most one object.

The standard multi-object observation model enables the Bayes multi-object filter to address mis-detection and false detection, but not occlusion. It assumes that each object is detected independently from each other, and that a detection cannot be assigned to more than one object. This is clearly not valid in occlusions.

### 2.4 Bayes Optimal Occlusion Handling

By relaxing the assumption that each object is independently detected, a multi-object observation model that explicitly addresses occlusion (as well as mis-detections and false positives) was proposed in [51]. The main difference between this so-called merged-measurement model and the standard model is the idea that each group of objects (instead of each object) in the multi-object state generates at most one detection [51]. Fig. 2 shows various partitions or groupings of a multi-object state with five objects.

A partition $U_X$ of a finite set $X$ is a collection of mutually exclusive subsets of $X$, whose union is $X$. The collection of all partitions of $X$ is denoted by $P(X)$. It is assumed that given a partition $U_X$, each group $Y \in U_X$ generates at most one detection with probability $P_{D,k}(Y)$, independent of other groups, and that conditional on detection generates $z$ with likelihood $g_{D,k}(z|Y)$.

Let $\mathcal{L}(U_X)$ denote the collection of labels of the partition $U_X$, i.e. $\mathcal{L}(U_X) \triangleq \{\mathcal{L}(Y) : Y \in U_X\}$ (note that $\mathcal{L}(U_X)$ forms a partition of $\mathcal{L}(X)$). Let $\Xi_k(\mathcal{L}(U_X))$ denote the class of all positive 1-1 mappings $\vartheta : \mathcal{L}(U_X) \rightarrow \{0:Z_k\}$. Then, the likelihood that a given partition $U_X$ of a multi-object state $X$, generates the detection set $Z_k$ is

$$\psi(\vartheta(\mathcal{L}(U_X)))(Y) = \sum_{\vartheta \in \Xi_k(\mathcal{L}(U_X))} \prod_{Y \in U_X} \psi_{Z_k}^{(\vartheta(\mathcal{L}(U_X)))(Y)}(\mathcal{L}(U_X))$$

where

$$\psi_{Z_k}^{(\vartheta(\mathcal{L}(U_X)))(Y)}(\mathcal{L}(U_X)) = \begin{cases} P_{D,k}(Y) \sigma_{D,k}(z_j|Y), & \text{if } j = 1:M \\ 1 - P_{D,k}(Y), & \text{if } j = 0 \end{cases}$$

with $\sigma_{D,k}(z_j|Y) = g_{D,k}(z_j|Y)/\kappa_k(z_j)$ denoting the detection SNR for group $Y$. The merged-measurement likelihood function is obtained by summing the group likelihoods (9) over all partitions of $X$ [51]:

$$g_k(y_k|X) = \sum_{U_X \in P(X)} \sum_{\vartheta \in \Xi_k(\mathcal{L}(U_X))} \prod_{Y \in U_X} \psi_{Z_k}^{(\vartheta(\mathcal{L}(U_X)))(Y)}(\mathcal{L}(U_X))$$

The multi-object filter [3, 4] with merged-measurement likelihood is Bayes optimal in the sense that the filtering density contains all information on the current multi-object state in the presence of false positives, mis-detections and occlusions. Unfortunately, this filter is numerically intractable due to the sum over all partitions of the multi-object state in the merged-measurement likelihood. At present, there is no polynomial time technique for truncating sums over partitions. Moreover, given a partition, computations involving the joint detection probability $P_{D,k}(Y)$, joint likelihood $g_{D,k}(z|Y)$ and associated joint densities are intractable except for scenarios with a few objects.

A GLMB approximation that reduces the number of
3.1 Death model

In most visual MOT applications, if an object stays in the scene for a long time, then it is more likely to continue to do so, provided it is not close to the designated exit regions. Such prior empirical knowledge can be used to improve the handling of occlusions, especially long occlusions that can lead to premature track termination in on-line MOT algorithms. In general, the GLMB filter would terminate an object that has not been detected over several frames. However, if this object has been in the scene for some time and is not in the proximity of designated exit regions, then it is highly likely to be occluded and track termination should be delayed. The labeled RFS formulation enables such prior information to be incorporated into track termination in a principled manner, via the survival probability.

The labeled RFS formulation accommodates survival probabilities that depend on track lengths, since a labeled state contains the time of birth in its label, and the track length is simply the difference between the current time and the time of birth. In practice, it is unlikely for an object to disappear in the middle of the visual scene (even if mis-detected or occluded) whereas it is more likely to disappear near designated exit regions due to the scene structure (e.g. the borders of the scene). Hence, we require the survival probability to be large (close to one) in the middle of the scene and small (close to zero) on the edges or designated death regions. Furthermore, since objects staying in the scene for a long time are more certain to continue existing, we require the survival probability to increase to one as its track length increases.

An explicit form of the survival probability that satisfies these requirements is given by

$$P_{S,k}(x, \ell) = \frac{b(x)}{1 + \exp(-\gamma(k - \ell[1,0]^T))} \tag{10}$$

where $b(x)$ is a scene mask that represents the scene structure, e.g., entrance or exit as illustrated in Fig. 3, and $\gamma$ is a control parameter of the sigmoid function. The scene mask $b(x)$ can be learned from a set of training data or designed from the known scene structure.

3.2 Hybrid Multi-Object Measurement Likelihood

While the detection set $Z_k$ is an efficient compression of the image observation $y_k$, mis-detected (including occluded) objects will not be updated by the filter. On the other hand the uncompressed observation $y_k$ contains relevant information about all objects, but the update step is computationally expensive. Conceptually, we can have the best of both worlds by updating detected objects with the associated detections and mis-detected objects with the visual observations localised to regions where these objects are expected. More importantly, this strategy exploits the
fact that occluded objects share measurements with the objects occluding them as illustrated in Fig. 4.

A hybrid tractable multi-object likelihood function that accommodates both detection and image observations can be obtained as follows. For tractability, it is assumed that each object generates observation independently from each other (similar to the standard observation model).

Given an object with state \((x,\ell)\) the likelihood of observing the local image \(T(y_k)\) (some transformation of the image \(y_k\)) is \(g_{T,k}(T(y_k)|x,\ell)\). On the other hand, given that there are no objects, the likelihood of observing \(T(y_k)\) is \(g_{T,k}(T(y_k)|\emptyset)\). The ratio

\[
\sigma_{T,k}(T(y_k)|x,\ell) \equiv \frac{g_{T,k}(T(y_k)|x,\ell)}{g_{T,k}(T(y_k)|\emptyset)}
\]  

(11)

can be interpreted as the visual SNR (c.f. detection SNR \((7)\)).

For a given association map \(\theta\) in the likelihood function \((7)\), an object with state \((x,\ell)\) is mis-detected if \(\theta(\ell) = 0\), in which case the value of \(\psi_{y_k}^{(\theta(\ell))}(x,\ell)\) is \(1 - P_{D,k}(x,\ell)\), the probability of a miss. Consequently, after the Bayes update, track \(\ell\) has no dependence on the observation \(y_k\). In order for track \(\ell\) to be updated with the local image \(T(y_k)\), the value of \(\psi_{y_k}^{(\theta(\ell))}(x,\ell)\) should be scaled by the visual SNR \(\sigma_{T,k}(T(y_k)|x,\ell)\). Note that the value of \(\psi_{y_k}^{(\theta(\ell))}(x,\ell)\) should remain unchanged for \(\theta(\ell) > 0\). Formally, this can be achieved by defining an extension of \((8)\) as follows

\[
\varphi_{y_k}^{(j)}(x,\ell) \equiv \psi_{y_k}^{(j)}(x,\ell) [\sigma_{T,k}(T(y_k)|x,\ell)]^{\delta_0[j]},
\]  

(12)

In other words, for \(j = 0\), \(\varphi_{y_k}^{(j)}(x,\ell)\) is equal to the visual SNR \((11)\) scaled by the mis-detection probability, otherwise it is equal to the detection SNR \((6)\) scaled by the detection probability.

Given a state \((x,\ell)\), \(\varphi_{y_k}^{(\theta(\ell))}(x,\ell)\) plays the same role as \(\psi_{y_k}^{(\theta(\ell))}(x,\ell)\), but accommodates both detection measurements and image measurements. Moreover, since each object generates observation independently from each other, the hybrid multi-object likelihood function has the same form as \((7)\), but with \(\psi_{y_k}^{(\theta(\ell))}(x,\ell)\) replaced by \(\varphi_{y_k}^{(\theta(\ell))}(x,\ell)\), i.e.

\[
g_k(y_k|X_k) \propto \sum_{\theta \in \Theta_k(L(X_k))} \prod_{(x,\ell) \in X_k} \varphi_{y_k}^{(\theta(\ell))}(x,\ell).
\]  

(13)

In visual occlusions, the hybrid likelihood allows occluded objects to share the image observations of the objects that occlude them. Moreover, when integrated into the Bayes recursion \((3)\)-\((4)\), consideration for a track-length-dependent survival probability in combination with information from the image observation, reduces uncertainties in the states of occluded objects and maintains their existence probabilities to keep the tracks alive. Hence, hi-jacking and premature track termination in longer occlusions will be avoided.

Remark: The hybrid multi-object likelihood function \((13)\) is a generalization of both the standard multi-object likelihood and the separable likelihood in \((44)\). When \(P_{D,k}(x,\ell) = 1\) for each \((x,\ell) \in X_k\), i.e. there is no misdetection, the hybrid likelihood \((13)\) is the same as the standard likelihood \((7)\). On the other hand, if \(P_{D,k}(x,\ell) = 0\) for each \((x,\ell) \in X_k\), i.e. there is no detection, then the only non-zero term in the hybrid likelihood \((13)\) is one with \(\theta(\ell) = 0\) for all \(\ell \in L(X_k)\). In this case, the hybrid likelihood \((13)\) reduces to the separable likelihood in \((44)\). For a general detection profile \(P_{D,k}\), the hybrid likelihood \((13)\) reduces to the standard likelihood \((7)\) when \(\sigma_{T,k}(T(y_k)|x,\ell) = 1\) for each \((x,\ell) \in X_k\).

Note that a hybrid likelihood function can be also developed for the merged-measurement model. However, the resulting multi-object filter still suffers from the same intractability as the merged-measurement filter.

### 3.3 Visual GLMB Recursion

A GLMB density can be written in the following form

\[
\pi(X) = \Delta(X) \sum_{\xi \in \Xi} \sum_{I \subseteq \Xi} \omega_{(I,\xi)} \delta_{I}(L(X)) \left[ p(\xi) \right]^{X},
\]  

(14)

where each \(\xi \in \Xi\) represents a history of association maps \(\xi = (\xi_1,\ldots,\xi_n)\) each \(p(\xi)\) is a probability density on \(X\), and each \(\omega_{(1,\xi)}\) is non-negative with \(\sum_{\xi \in \Xi} \sum_{I \subseteq \Xi} \omega_{(I,\xi)} = 1\).

The cardinality distribution of a GLMB is given by

\[
Pr(|X| = n) = \sum_{\xi \in \Xi} \sum_{I \subseteq \Xi} \delta_n(|I|) \omega_{(I,\xi)},
\]  

(15)

while, the existence probability and probability density of track \(\ell \in \mathbb{L}\) are respectively

\[
r(\ell) = \sum_{\xi \in \Xi} \sum_{I \subseteq \Xi} 1_I(\ell) \omega_{(I,\xi)},
\]  

(16)

\[
p(x,\ell) = \frac{1}{r(\ell)} \sum_{\xi \in \Xi} \sum_{I \subseteq \Xi} 1_I(\ell) \omega_{(I,\xi)} p(\xi)(x,\ell).
\]  

(17)

Given the GLMB density \((14)\), an intuitive multi-object estimator is the multi-Bernoulli estimator, which first determines the set of labels \(L \subseteq \mathbb{L}\) with existence probabilities above a prescribed threshold, and second the maximum a posteriori (MAP) or the mean estimates from the densities \(p(\cdot,\ell), \ell \in L\), for the states of the objects. A popular estimator is a suboptimal version of the Marginal Multi-object Estimator \((4)\), which first determines the pair \((L,\xi)\) with the highest weight \(\omega_{(L,\xi)}\) such that \(|L|\) coincides with the MAP cardinality estimate, and second the MAP/mean estimates from \(p(\cdot,\ell), \ell \in L\), for the states of the objects.

The GLMB family also enjoys a number of nice analytical properties. The void probability functional—a necessary and sufficient statistic—of a GLMB, the Cauchy-Schwarz divergence between two GLMBs, the \(L_1\)-distance between a GLMB and its truncation, can all be computed in closed form \((38)\). The GLMB is flexible enough to approximate any labeled RFS density with matching intensity function and cardinality distribution \((42)\).

More importantly, the GLMB family is closed under the prediction equation \((3)\) and conjugate with respect to the standard observation likelihood \((37)\) as well as the hybrid observation likelihood (since the these likelihoods have the
same form). Hence, starting from an initial GLMB prior, all multi-object predicted and updated densities propagated by the Bayes recursion (17)-(18) are GLMBs.

In this work we adopt the joint prediction and update strategy [39] for the proposed visual MOT GLMB filter. For notational compactness, we subscript the $k$ for the current time, the next time is indicated by the subscript `$+$`. Given the filtering density (14) at time $k$, the filtering density at time $k+1$ is given by

$$
\pi_k(X) \propto \Delta(X) \sum_{I,\xi,\theta,\alpha} \omega(I,\xi) \omega(y,\theta) \delta_{I,\alpha} L(X)[p(\xi,\theta)]^X
$$

where $I \in \mathcal{F}(\mathbb{L})$, $\xi \in \Xi$, $\alpha \in \Theta$, and $p(\xi,\theta) = \sum_{\theta} \sum_{\xi} \omega(I,\xi,\theta) \delta_{I,\alpha} L(X)[p(\xi,\theta)]^X$

(18)

Hence at the next iteration we only propagate forward the survivals together with associations of new measurements to their component sets (19). Hence, starting from an initial GLMB prior, $\omega(I,\xi,\theta)$ is given by

$$
\omega(I,\xi,\theta) = \frac{1}{N} \sum_{h,t} \omega(I,\xi,\theta) \delta_{I,\alpha} L(X)[p(\xi,\theta)]^X
$$

(20)

implementing the GLMB filter amounts to propagating the component set $\{((I,h),\omega(h),p(h))\}_{h=1}^H$ (there is no need to store $\xi(h)$) forward in time using (18)-(24). The procedure for computing the component set $\{(I,h),\omega(h),p(h)\}_{h=1}^H$ at the next time is summarized in Algorithm 1. Note that to be consistent with the indexing by $h$ instead of $(I,\xi)$, we also abbreviate

$$
P^h_S(\xi,\theta) \propto \sum_{\xi(h)} \pi(\xi,\theta) (I + 1,\xi,\theta)
$$

(21)

where $I(h) = I + 1$. Hence, given a component $I(h)$, the weights, the smaller the $\omega(h)$, the less likely $I(h)$ is to be extended. Note that a component $I(h)$ is extended only when $\omega(h)$ is above a threshold.

The sum at (18) can be interpreted as an enumeration of all possible combinations of births, deaths and survivals together with associations of new measurements to hypothesized tracks. Observe that (18) does indeed take the same form as (14) when rewritten as a sum over $I,\xi,\theta$ with weights

$$
\omega(I,\xi,\theta) \propto \sum_{I,\xi,\theta} \omega(I,\xi) \omega(I,\xi,\theta).
$$

(24)

Hence at the next iteration we only propagate forward the components $(I,\xi,\theta)$ with weights $\omega(I,\xi,\theta)$.

### 3.4 GLMB Filter Implementation

The number of terms in the GLMB filtering density grows super-exponentially, and it is necessary to truncate these terms without exhaustive enumeration. The first implementation of the GLMB filter truncates the prediction density using the K-shortest path algorithm, and then truncates filtering density by solving using the ranked assignment algorithm [38]. In [39] the joint prediction and update (18)-(24) was designed to improve the truncation efficiency of the two-stage implementation. Further, the GLMB truncation can be performed via Gibbs sampling with linear complexity in the number of detections (the reader is referred to [39] for derivations and analysis). Fortuitously, this implementation can be readily adapted for the visual MOT GLMB filter.

A GLMB filtering density (14) at time $k$ is completely characterized by the parameters $(\omega(I,\xi),p(\xi))$, $(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi$, which can be enumerated as $\{(I(h),\xi(h),\omega(h),p(h))\}_{h=1}^H$ where

$$
\omega \propto \omega(I(h),\xi(h)), \quad p(h) \propto p(\xi(h)).
$$

(25)

Since the GLMB (14) can be rewritten as

$$
\pi(X) = \Delta(X) \sum_{h=1}^H \omega(h) \delta_{I(h)} L(X)[p(h)]^X,
$$

(26)

implementing the GLMB filter amounts to propagating the component set $\{(I(h),\omega(h),p(h))\}_{h=1}^H$ (there is no need to store $\xi(h)$) forward in time using (18)-(24). The procedure for computing the component set $\{(I(h),\omega(h),p(h))\}_{h=1}^H$ at the next time is summarized in Algorithm 1. Note that to be consistent with the indexing by $h$ instead of $(I,\xi)$, we also abbreviate

$$
P^h_S(\xi,\theta) \propto \sum_{\xi(h)} \pi(\xi,\theta) (I + 1,\xi,\theta)
$$

(27)

where $I(h) = I + 1$. Hence, given a component $I(h)$, the weights, the smaller the $\omega(h)$, the less likely $I(h)$ is to be extended. Note that a component $I(h)$ is extended only when $\omega(h)$ is above a threshold.

There are three main steps in one iteration of the GLMB filter. First, the Gibbs sampler is used to generate (without exhaustive enumeration) the auxiliary vectors $\gamma(h,t)$, $h = 1:H$, $t = 1:T(h)$, with the most significant weights $\omega(h)$. Note that $\gamma(h,t)$ is an equivalent representation of $(I(h,t),\theta(h,t))$, with components $\gamma_i(h,t)$, $i = 1:|I(h) \cup \Theta|$, defined as $\theta_i(h,t)$ when $\tilde{e}_i \in I(h)$, and $-1$ otherwise. The Gibbs sampler (Algorithm 1a) has an exponential convergence rate [39]. More importantly, it is not necessary to discard burn-ins and wait for samples from the stationary distribution. All distinct samples can be used, the larger the weights, the smaller the $L_1$ error from the true GLMB filtering density [39].

Second, the auxiliary vectors are used to generate an intermediate set of parameters with the most significant weights $(I(h),\omega(h),p(h))$, $h = 1:H$, $t = 1:T(h)$, via (18). Note that given a component $h$ and $\gamma(h,t)$, it can be shown that

$$
I(h) = \{\tilde{e}_i \in I(h) \cup \Theta : \gamma_i(h,t) \geq 0\},
$$

(28)

$$
\omega(h) \propto \omega(h) \prod_{i=1}^{|I(h) \cup \Theta|} \eta_i(h)(\gamma_i(h,t)),
$$

(29)

$$
p(h)(\gamma_i(h,t)) \propto \frac{\omega(h)}{\omega(h) \prod_{i=1}^{|I(h) \cup \Theta|} \eta_i(h)(\gamma_i(h,t))} \gamma_i(h,t).
$$

(30)

Note also that $\theta_i(h,t)(\tilde{e}_i) = \gamma_i(h,t)$ when $\gamma_i(h,t) \geq 0$, for $\tilde{e}_i \in I(h)$. Third, the intermediate parameters are marginalized via (24) to give the new parameter set $\{(I(h),\omega(h),p(h))\}_{h=1}^H$. Note that $U_{h,t}$ gives the
index of the GLMB component at time \( k + 1 \) that 
\((I^{(h)}, I^{(h, t)}, p^{(h, t)}) \) contributes to.

**Algorithm 1. Joint Prediction and Update**
- **input:** \( \{(I^{(h)}, \omega^{(h)}, p^{(h)})\}_{h=1}^{H_{max}}, y_{+}, Z_{+}, H_{max}^{+} \)
- **input:** \( \{(I^{(t)}_{B, \omega^{(t)}_{B}}, p^{(t)}_{B})\}_{t \in \mathbb{B}_{+}}, P_{S}, f_{+}(\cdot), \sigma_{D,+}(\cdot) \)
- **output:** \( \{(I^{(h, +)}, \omega^{(h, +)}, p^{(h, +)})\}_{h=1}^{H_{+}} \)

sample counts \( [T^{(h)}_{+} H_{+} = 1] \) from multinomial distribution with parameters \( H_{+} \) trials and weights \( [\omega^{(h)}^{+}]_{h=1}^{H_{+}} \)

for \( h = 1 : H \)

compute \( \eta^{(h)} := \{h^{(j)}\}_{j=1}^{(H_{+} B_{+} + |Z_{+}^{+})} \) using (25)

initialize \( \gamma^{(h, 1)} \)

\( \{\gamma^{(h, t)}\}_{t=1}^{T^{(h)}} := \text{Unique}(\text{Gibbs}(\gamma^{(h, 1)}, T^{(h)}), \eta^{(h)}) \)

for \( t = 1 : T^{(h)} \)

compute \( I_{+}^{(h, t)} \) from \( I^{(h)} \) and \( \gamma^{(h, t)} \) using (26)

compute \( \omega^{(h, t)} \) from \( \omega^{(h)} \) and \( \gamma^{(h, t)} \) using (27)

compute \( p_{+}^{(h, t)} \) from \( p^{(h)} \) and \( \gamma^{(h, t)} \) using (28)

end

\( \{(I^{(h, +)}, p^{(h, +)})\}_{h=1}^{H_{+}} \sim \text{Unique}(\{(I^{(h, t)}, p^{(h, t)}_{+})\}_{(h, t)=(1,1)}^{(H, T^{(h)})}) \)

for \( h_{+} = 1 : H_{+} \)

\( \omega_{+}^{(h_{+})} := \sum_{h, t; U_{h, t}=h_{+}} \omega^{(h, t)} \)

end

normalize weights \( [\omega^{(h, +)}]_{h=1}^{H_{+}} \)

**Algorithm 1a. Gibbs**
- **input:** \( \gamma^{(1)} \), \( T, \eta = [\eta_{n}(j)] \)
- **output:** \( \gamma^{(t)} \), \( n_{t} \)

\( P := \text{size}(\eta, 1); \ M := \text{size}(\eta, 2) - 2; \ c := [-1 : M]; \)

for \( t = 2 : T \)

\( \gamma^{(t)} := [\cdot] \)

for \( n = 1 : P \)

\( p_{n} := [\eta_{n}(-1), \ldots, \eta_{n}(M)] \)

for \( j = 1 : M \)

\( p_{n}(j) := p_{n}(j)(1 - \gamma^{(t)}_{\gamma_{n+1,\gamma_{n+1},\gamma_{n+1}}(j)}) \)

end

\( \gamma^{(t)}_{n} \sim \text{Categorical}(c, p_{n}); \gamma^{(t)} := [\gamma^{(t)}, \gamma^{(t)}_{n}] \)

end

4.1 Scenario and Implementation Settings

**Dynamic motion model:** Individual object kinematics are described by a 4D state vector \( x_{k} = [p_{x, k}, p_{y, k}, \dot{p}_{x, k}, \dot{p}_{y, k}]^{T} \) of position and velocity, which follows a constant velocity model with sampling period \( T_{s} \) equal to the sampling rate of the video, and process noise \( v_{k} \sim \mathcal{N}(v_{k}; 0, Q) \) where \( Q = \sigma_{v}^{2} I_{2} \), \( I_{2} \) is the \( 2 \times 2 \) identity matrix, and \( \sigma_{v} = 3 \) pixels/frame is the noise standard deviation (set by considering the maximum speed of the object with regard to the frame rate). Hence, the transition density

\[ f_{k|k-1}(x_{k}|x_{k-1}) = \mathcal{N}(x_{k}; F x_{k-1}, Q), \]

where \( F = I_{2} \otimes \begin{bmatrix} 1 & T_{s} \\ 0 & 1 \end{bmatrix} \).

Remark: While the object’s extent such as its bounding box \([11], [13], [19], [29], [30]\) can be included in the object state, effective modeling of extent dynamics is application dependent. In experiments we estimate an object’s extent via the median values of the x, y scale of the detections associated with existing tracks in a given time window.

Remark: Similar to single-object visual tracking filtering in \([11]\), the predicted covariance for each track is capped to a prescribed value to prevent it from exploding over time.

The RFS framework accommodates a time-varying birth model. In this experiment, we use a birth model that consists of both static and dynamic components. The static component is an LMB that describes expected locations where objects are highly likely to appear e.g., the image border/footpaths near the image border. The dynamic component is a time-varying LMB that exploits associations with weak associations (to existing tracks) to describe highly likely object births at the next time frame \([50]\).

**Measurement model:** The detection \( z \in D(y_{k}) \) of an object is obtained by a discriminative part-based HOG model (DPM) \([33]\) detector, which can be modelled by the likelihood function

\[ g_{D,k}(z|x, \ell) = \mathcal{N}(z; H x, \Sigma) \]

where \( H = [1 \ 0 \ 1 \ 0]; \Sigma = \text{diag}(5^{2}, 5^{2}) \). The probability of detection \( P_{D} \) is 0.98 and the clutter rate is 5, i.e., an average of 5 clutter measurements per frame. These parameters can be obtained from training data or learned on-the-fly in the RFS framework as proposed in \([22]\).

Various types of image features can be exploited to construct alternative observations for objects missed by the DPM detector. In this experiment we use the superpixel segmentation proposed in \([36]\) since it admits tractable object appearance models. For a raw image \( y_{t} \), the superpixel image \( T(y_{k}) \) is a set \( \{T_{1}, \ldots, T_{m_{s}}\} \) of superpixels of \( y_{t} \). The appearance model of an individual object consists of colour intensity clusters with centers \( \bar{f}_{i} \) and radii \( r_{i}, i = 1, \ldots, n_{c} \), learned from the superpixels of the training samples collected in time as described in \([36]\). The superpixel-based image observation likelihood is given by

4 Experimental results

In this section, we test the proposed MOT filter on the S2Li sequence of the PET2009 dataset \([33]\), the BAHNHOF and SUNNYDAY sequences of the ETH dataset \([18]\), and the TUD-Stadtmitte sequence from \([19]\). To benchmark its tracking performance against a number of recent algorithms, we use publicly accessible detection results and the evaluation tool from \([33], [34]\).
indicating how much the 350 DPM detections are used to update changes, superpixels inside object regions from confident and the background, respectively. To adapt to appearance changes, superpixels inside object regions from confident DPM detections are used to update \( \bar{f}_i \) and \( r_i \). Empirically, this strategy is more robust than the update scheme in [26] and computationally much more efficient because the cluster model is updated by confident DPM responses so that accumulated learning errors are avoided. Note that the Unscented Transform is used for the measurement update when the superpixel-based image observations is considered.

**State Estimation:** In the experiments, the maximum number of track hypotheses \( H_{\text{max}} \) is set to 200 and track estimates are obtained from the GLMB filtering density via the LMB estimator described in Subsection 3.3. Note that when the LMB estimator terminates a track, the GLMB filtering density still contains its existence probability and state density (hence state estimate). This information is completely deleted only when its existence probability is so negligible that all relevant GLMB components are truncated. If not completely deleted, it is possible that due to new evidence in the data at later time, a track’s existence probability becomes significant enough to be selected by multi-object estimator, leading to track fragmentation. While this problem can be addressed in a principled manner via multi-object smoothing, the GM-PHD smoother [23] is not applicable and an implementation of the forward-backward GLMB smoother [52] is not yet available. Nonetheless, we can exploit the available information on the terminated track from the GLMB density in previous frames to recover missing state estimates.

\[
g_{T,k}(y_k|x, \ell) \propto \sum_{i=1}^{n_c} \sum_{j=1}^{m^*_i} C_i(x) e^{-\frac{1}{\sigma_i} \|f(T_j) - f_i\|},
\]

\[
C_i(x) = \frac{S_i^+(A(x)) - S_i^-(A(x))}{S_i^+(A(x)) + S_i^-(A(x))},
\]

where \( m^*_i \) is the number of superpixels in the region occupied by an object with state \( x \); \( f(T_j) \) is the colour intensity of the \( j \)th superpixel; \( A(x) \) is the region occupied by a given state \( x \); \( S_i^+(A(x)) \) and \( S_i^-(A(x)) \) are scores indicating how much the \( i \)th cluster overlap with \( A(x) \) and the background, respectively. To adapt to appearance changes, superpixels inside object regions from confident DPM detections are used to update \( \bar{f}_i \) and \( r_i \).

**BAHNHOFF sequence:** This sequence is more challenging than the previous two for online MOT due to fluctuating camera motion and frequent total occlusions resulting from the low camera view. Bearing in mind that relatively high track losses and ID switches are expected from online algorithms in such scenarios, the GLMB filter achieved a relatively low level of track fragmentation. In addition, observe from the 1st row of Fig. 7 that new tracks are correctly initialized despite distractions from closely spaced or overlapping objects. The 2nd row demonstrates that once initialized, most track labels are correctly maintained in the busy scene. The 3rd row illustrates that potential track loss from the mis-detections of target #180, due to occlusions from #175, is avoided by the GLMB filter via the track-length dependent survival probability and the superpixel observation updates.

**Cardinality estimation:** Unlike other tracking approaches, the RFS approach also provides the probability distribution of the current number of objects, i.e., cardinality distribution (see Section 3.3). In Fig. 8, the cardinality distribution for each frame is displayed for the three sequences.

### 4.2 Qualitative performance analysis

**S2LI sequence:** Fig. 5 displays the output of the proposed GLMB tracker for frames 37 to 110, 144 to 163, and 350 to 365, in which each tracked object is represented a bounding box labeled with an integer, and those detected are marked with yellow stars. Interactions between the three people near the light pole results in mutual occlusion and one person not detected. Observe from the yellow stars, in the 1st and 2nd rows that the mis-detection jumps between different people over a short time. This causes frequent ID switches and track fragmentation in tracking methods such as [29], [49]. In the 3rd row, two of the three interacting people are simultaneously mis-detected, a scenario that invariably leads to track loss in online tracking methods. However, for the GLMB filter, mis-detected tracks are still maintained by the track-length dependent survival probability and updated with the superpixel observations thereby minimizing ID switches.

**TUD-Stadtmitte sequence:** In this sequence, people are frequently overlapping due to the low camera view, and track initiation is more difficult than the S2LI sequence because new people in the scene are often occluded by others. This problem is addressed by the combination of static and dynamic measurement-driven birth models as described in Section 4.1. As can be seen in Fig. 6, mutual occlusions among people are effectively handled by the GLMB filter. It is worth noting that in this experiment, the GLMB filter mainly exploits position measurements and appearance information is only used when the position measurements are not associated. Further reduction of ID switches from occlusions can be achieved by the GLMB filter by incorporating appearance information into the likelihood function.

**Evaluation metrics and software:** The GLMB filter tracking performance is benchmarked against offline-based methods such as: StruckMOT [27], PRIMPT [26]. Centracker [29], and KSP [28]; and recent online trackers: [16], [48]. Also note that the online tracker [48] cannot be applied to the second and third sequences because ground plane information is not available. We use well-known MOT performance indices [34] such as Recall (correctly tracked objects over total ground truth), Precision (correctly tracked objects over total tracking results), and false positives per frame (FPF). We also report the number of identity switches (IDS) and the number of fragmentations (Frag). Table 1 shows the ratio of tracks with successfully tracked parts for more than 80% (mostly tracked (MT)), less than
20% (mostly lost (ML)), or less than 80% and more than 20% (partially tracked (PT)). The up (down) arrows in Table 1 mean that higher (lower) the values indicate better performance.

**Discussion:** In the ETH sequences, the RMOT [16] shows slightly better results because the proposed relative motion network model in RMOT is especially tailored for handling of full occlusions in tracking scenarios of group of people walking in the same directions. As mentioned in Section 4.2, more fragmentation is observed in the EHT sequences due to re-initialization of objects from the measurement-driven birth model when they emerge from very long full occlusions. Due to the generality of the framework, more sophisticated motion models and other types of detections and appearance features can be incorporated for further improvements.

The tracking experiments with the proposed GLMB filter are implemented in MATLAB using single core (Intel 2.4GHz 5500) CPU laptop. A comparison of tracking speed with other trackers (excluding the detection process) is summarized in Table 2, which shows an average (over all three experiments) of 20 fps for the GLMB filter (without code optimization). Hence, the GLMB filter is very well-suited for online applications considering further speed up can be achieved using C++ and code optimization. Further, the salient feature of the proposed GLMB filter is its linear complexity with respect to the number of detections [39].

It is important to note that the computation speeds in Table 2 can be only used for a rough indication because all implementations are dependent on the hardware platform, programming language, code structure, test sequence scenarios, etc.

5 **Conclusion**

An efficient online visual MOT algorithm that seamlessly integrates state estimation, track management, clutter rejection, mis-detection and occlusion handling into one single Bayesian recursion has been proposed. Further, the proposed MOT algorithm exploits the advantages of both detection-based and TBD approaches. In particular, it has the efficiency of detection-based approach that avoids updating with the entire image, while at the same time...
Fig. 7: Selected frames of BAHNHOF tracking results for case study

Fig. 8: Cardinality distributions for three sequences left:S2L1, center:TUD, right:BAHNHOF

| Dataset          | Method             | Recall ↑ | Precision ↑ | FPF ↓ | GT | MT ↑ | PT ↓ | ML ↓ | Frag ↓ | IDS ↓ |
|------------------|--------------------|----------|-------------|-------|----|-----|------|------|--------|-------|
| PETS09-S2L1      | GLMB               | 95.6 %   | 92.2 %      | 0.03  | 19 | 97 % | 5 %  | 0.0 %| 23     | 3     |
|                  | RMOT [16]          | 95.6 %   | 95.4 %      | 0.05  | 19 | 94.7 %| 5.3 %| 0.0 %| 23     | 1     |
|                  | StruckMOT [27]     | 97.2 %   | 93.7 %      | 0.38  | 19 | 94.7 %| 5.3 %| 0.0 %| 19     | 4     |
|                  | PRIMPT [26]        | 89.5 %   | 99.6 %      | 0.02  | 19 | 78.9 %| 21.1 %| 0.0 %| 23     | 1     |
|                  | CemTracker [29]    | -        | -           | -     | 19 | 94.7 %| 5.3 %| 0.0 %| 15     | 22    |
|                  | KSP [28]           | -        | -           | -     | 23 | 73.9 %| 5.3 %| 0.08 %| 22     | 13    |
|                  | GeodesicTracker [35]| -       | -           | -     | 23 | 100 % | 0 %  | 0 %  | 16     | 9     |
| TUD-Stadtmitte   | GLMB               | 89.4 %   | 87.6 %      | 0.11  | 10 | 80.0 %| 20.0 %| 0.0 %| 10     | 1     |
|                  | RMOT [16]          | 87.9 %   | 96.6 %      | 0.19  | 10 | 80.0 %| 20.0 %| 0.0 %| 7      | 6     |
|                  | StruckMOT [27]     | 87.3 %   | 95.4 %      | 0.25  | 10 | 80.0 %| 20.0 %| 0.0 %| 11     | 0     |
|                  | PRIMPT [26]        | 81.0 %   | 99.5 %      | 0.028 | 10 | 60.0 %| 30.0 %| 10.0 %| 0      | 1     |
|                  | OnlineCRF [33]     | 87.0 %   | 96.7 %      | 0.18  | 10 | 70.0 %| 30.0 %| 0.0 %| 1      | 0     |
|                  | CemTracker [29]    | -        | -           | -     | 10 | 40.0 %| 60.0 %| 0.0 %| 13     | 15    |
|                  | KSP [28]           | -        | -           | -     | 9  | 11 %  | 5.3 %| 0.08 %| 15     | 5     |
| ETH              | GLMB               | 80.1 %   | 85.6 %      | 0.98  | 124| 62.4 %| 32.6 %| 5.0 %| 70     | 20    |
|                  | RMOT [16]          | 81.5 %   | 86.3 %      | 0.98  | 124| 67.7 %| 27.4 %| 4.8 %| 38     | 40    |
|                  | StruckMOT [27]     | 78.4 %   | 84.1 %      | 0.98  | 124| 62.7 %| 29.6 %| 7.7 %| 72     | 5     |
|                  | MOT-TBD [31]       | 78.7 %   | 85.5 %      | -     | 125| 62.4 %| 29.6 %| 8.0 %| 69     | 45    |
|                  | PRIMPT [26]        | 76.8 %   | 86.6 %      | 0.89  | 125| 58.4 %| 33.6 %| 8.0 %| 23     | 11    |
|                  | OnlineCRF [33]     | 79.0 %   | 85.0 %      | 0.64  | 125| 68.0 %| 24.8 %| 7.2 %| 19     | 11    |
|                  | CemTracker [29]    | 77.3 %   | 87.2 %      | -     | 124| 66.4 %| 25.4 %| 8.2 %| 69     | 57    |
|                  | MT-TBD [31]        | -        | -           | -     | 125| 62.4 %| 29.60 %| 8 %  | 69     | 45    |

TABLE 1: Comparison of tracking performance with the state-of-the-art trackers
making use of information at the image level by using only small regions of the image where mis-detected objects are expected. The proposed algorithm has a linear complexity in the number of detections and quadratic in the number of hypothesized tracks, making it suitable for real-time computation. Moreover, experimental results on well-known datasets indicated that proposed algorithm is competitive in tracking accuracy compared to data association based batch algorithms and recent online algorithms.

ACKNOWLEDGMENTS

REFERENCES

[1] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice-Hall, New Jersey, 1993.
[2] Y. B.-Sharlom, X. R. Li, and T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software, Wiley & Sons, 2001.
[3] C. Robert, The Bayesian Choice, Prentice-Hall, Springer, 2007.
[4] R. Mahler, Statistical Multisource-Multitarget Information Fusion, Norwood, MA: Artech House, 2007.
[5] R. Mahler, Advances in Statistical Multisource-Multitarget Information Fusion, Norwood, MA: Artech House, 2014.
[6] M. Isard, and J. MacCormick, “BraMBLe: A Bayesian Multiple-Blob Tracker,” Proc. IEEE Int. Conf. Computer Vision, 2001.
[7] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based Probabilistic Tracking,” Proc. European Conf. Computer Vision, 2002.
[8] J. Vermaak, A. Doucet, and P. Perez, “Maintaining Multi-Modality through Mixture Tracking,” Proc. IEEE Int. Conf. Computer Vision, 2003.
[9] K. Nummiaro, E. Koller-Meier, and L. V. Gool, “An Adaptive Color-based Particle Filter,” Image and Vision Computing, 21 (10): 99-110, 2003.
[10] J. Czyz, B. Ristic, and B. Macq, “A Particle Filter for Joint Detection and Tracking of Color Objects,” Image and Vision Computing, 25 (18): 1271-1281, 2007.
[11] M. D. Breitenstein, F. Reichlin, B. Leibe, E. K.-Meier, and L.V. Gool, “Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera,” IEEE Trans. Pattern Analysis and Machine Intelligence, 33 (9): 1820-1833, 2011.
[12] T. D. Laet, H. Bruyninckx, and J.D. Schutter, “Shape-Based Online Multitarget Tracking and Detection for Targets Causing Multiple Measurements: Variational Bayesian Clustering and Lossless Data Association,” IEEE Trans. Pattern Analysis and Machine Intelligence, 33 (12): 2477-2491, 2011.
[13] W. Choi, C. Pantofaru, and S. Savarase, “A General Framework for Tracking Multiple People from a Moving Camera,” IEEE Trans. Pattern Analysis and Machine Intelligence, 35 (7): 1577-1591, 2013.
[14] Z. Khan, T. Balch, and F. Drellar, “MCMB-C-Based Particle Filtering for Tracking a Variable Number of Interacting Targets,” IEEE Trans. Pattern Analysis and Machine Intelligence, 27 (11): 1805-1819, 2005.
[15] K. Okuma, A. Taleghani, N. de Freitas, and D. G. Lowe, “A Boosted Particle Filter: Multitarget Detection and Tracking,” Proc. European Conf. Computer Vision, 2004.
[16] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, “Bayesian Multi-Object Tracking using Motion Context from Multiple Objects,” Proc. IEEE Winter Conference on Applications of Computer Vision (WACV), 2015.

### Table 2: Comparison of averaged speed

| Method          | Average speed | Implementation |
|-----------------|---------------|----------------|
| GLMB            | 20 fps        | MATLAB         |
| RMOT            | 3.5 fps       | MATLAB         |
| GeodesicTracker | 11.2 fps      | MATLAB         |
| StruckMOT       | 15 fps        | MATLAB         |
| CemTracker      | 0.55 fps      | MATLAB         |
| OnlineCRF       | 8 fps         | C++            |

[17] N. Chenouard, I. Bloch, and J.-C. O.-Marin, “Multiple Hypothesis Tracking for Cluttered Biological Image Sequences,” IEEE Trans. Pattern Analysis and Machine Intelligence, 35 (11): 2736-2750, 2013.
[18] A. Ess, B. Leibe, K. Schindler, and L.V. Gool “A Mobile Vision System for Robust Multi-Person Tracking,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008.
[19] M. Andriluka, S. Roth, and B. Schiele “Monocular 3D Pose Estimation and Tracking by Detection,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2010.
[20] R. Mahler, “Multi-target Bayes Filtering via First-order Multi-target Moments,” IEEE Trans. Aerosp. Electron. Syst, 39 (4): 1152-1178, 2003.
[21] R. Mahler, “PHD Filters of Higher Order in Target Number,” IEEE Trans. Aerosp. Electron. Syst, 43(3): 1523-1543, 2007.
[22] R. Mahler, B.-T. Vo, and B.-N. Vo, “CPHD Filtering with Unknown Clutter Rate and Detection Profile,” IEEE Trans. Signal Processing, 59 (8): 3497-3513, 2011.
[23] R. Mahler, B.-T. Vo, and B.-N. Vo “Forward-Backward Probability Hypothesis Density Smoothing,” IEEE Trans. Aerosp. Electron. Syst, 48 (1): 707-728, 2012.
[24] B.-N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo methods for Multitarget Filtering with Random Finite Sets,” IEEE Trans. Aerosp. Electron. Syst, 46 (5): 1224-1245, 2010.
[25] L. Zhang, Y. Li and R. Nevatia, “Global Data Association for Multi-Object Tracking using Network Flows,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008.
[26] C.-H. Kuo and R. Nevatia, “How Does Person Identity Recognition Help Multi-Person Tracking?,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2011.
[27] S. Kim, S. Kwak, J. Feyereisl, and B. Han, “Online Multi-target Tracking by Large Margin Structured Learning,” Proc. Asian Conf. Computer Vision, 2012.
[28] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, “Multiple Object Tracking Using K-Shortest Paths Optimization,” IEEE Trans. Pattern Analysis and Machine Intelligence, 33 (9): 1808-1819, 2011.
[29] A. Milan, S. Roth, and K. Schindler, “Continuous Energy Minimization for Multitarget Tracking,” IEEE Trans. Pattern Analysis and Machine Intelligence, 36 (1): 58-72, 2014.
[30] A. Milan, K. Schindler, and S. Roth, “Multi-Target Tracking by Discrete-Continuous Energy Minimization,” IEEE Trans. Pattern Analysis and Machine Intelligence, 38 (10): 2054-2068, 2016.
[31] F. Poiesi, R. Mazzon, and A. Cavallaro, “Multi-target Tracking on Confidence Maps: An Application to People Tracking,” Computer Vision and Image Understanding, 117(10): 1257-1272, 2013.
[32] A. Dehghan, Y. Tian and R. Nevatia, “Target Identity-aware Network Flow for Online Multiple Target Tracking,” Proc. IEEE Conf. Computer Vision and Pattern Recognition 2015.
[33] B. Yang, and R. Nevatia, “An Online Learned CRF Model for Multi-Target Tracking,” Proc. IEEE Conf. Computer Vision and Pattern Recognition 2012.
[34] Y. Li, C. Huang, and R. Nevatia, “Learning to Associate: Hybrid Boosted Multi-target Tracker for Crowded Scene,” Proc. IEEE Conf. Computer Vision and Pattern Recognition 2009.
[35] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” IEEE Trans. Pattern Analysis and Machine Intelligence, 32 (9): 1627-1645, 2010.
[36] F. Yang, H. Lu, and M.-H. Yang, “Robust Superpixel Tracking,” IEEE Trans. Image Processing, 23 (4): 1639-1651, 2014.
[37] B.-T. Vo, and B.-N. Vo, “Labeled Random Finite Sets and Multi-Object Conjugate Priors,” IEEE Trans. Signal Processing, 61 (13): 3460-3475, 2013.
[38] B.-N. Vo, B.-T. Vo, and D. Phung, “Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter,” IEEE Trans. Signal Processing 62 (24): 6554-6567, 2014.
[39] B.-N. Vo, B.-T. Vo, and H. Hoang, “An Efficient Implementation of the Generalized Labeled Multi-Bernoulli Filter,” Available: https://arxiv.org/abs/1606.08350.
[40] S. Davey, M. Rutter, N. Gordon, M. Mahillick, V. Krishnamurty and B.-N. Vo, Eds., Integrated Tracking, Classification, and Sensor Management: Theory and Applications. New York, NY, USA: Wiley/IEEE, 2012, ch. 8, pp. 311-361.
[41] F. Papi, and D. Y. Kim, “A Particle Multi-target Tracker for Super-positional Measurements using Labeled Random Finite Sets,” IEEE Trans. Signal Processing 63 (16): 4348-4358, 2015.
[42] F. Papi, B.-N. Vo, B.-T. Vo, C. Fantacci, M. Beard, “Generalized Labeled Multi-Bernoulli Approximation of Multi-Object Densities,” IEEE Trans. Signal Processing 63 (20): 5487-5497, 2015.

[43] E. Maggio, M. Taj, and A. Cavallaro, “Efficient Multi-target Visual Tracking using Random Finite Sets,” IEEE Trans. Circuits Syst. Video Techn., 18 (8): 1016-1027, 2008.

[44] B.-N. Vo, B.-T. Vo, N.-T. Pham and D. Suter, “Joint Detection and Estimation of Multiple Objects from Image Observations,” IEEE Trans. Signal Processing, 58 (10): 5129-5241, 2010.

[45] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter, “Visual Tracking of Numerous Targets via Multi-Bernoulli Filtering of Image Data,” Pattern Recognition, 45 (10): 3625-3635, 2012.

[46] T. Vu, B.-N. Vo, and R. Evans, “A Particle Marginal Metropolis-Hastings Multi-Target Tracker,” IEEE Trans. Signal Processing, 62 (15): 3953-3964, 2014.

[47] P. Craciun, M. Ortner, and I. Zervakis, “Joint Detection and Tracking of Moving Objects using Spatio-Temporal Marked Point Processes,” Proc. IEEE Winter Conference on Applications of Computer Vision (WACV), 2015.

[48] H. Possegger, T. Maauthner, P. N. Roth, and H. Bischof, “Occlusion Geodesics for Online Multi-Object Tracking,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2014.

[49] M. Yang, Y. Liu, L. Wen, Z. You, and S. Z. Li, “A Probabilistic Framework for Multitarget Tracking with Mutual Occlusions,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2014.

[50] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, “The Labeled Multi-Bernoulli Filter,” IEEE Trans. Signal Processing, 62 (12): 3246–3260, 2014.

[51] M. Beard, B.-T. Vo, and B.-N. Vo, "Bayesian Multi-Target Tracking with Merged Measurements using Labelled Random Finite Sets", IEEE Signal Processing, 63 (6): 1433-1447, 2015.

[52] M. Beard, B.-T. Vo, and B.-N. Vo, “Generalized Labelled Multi-Bernoulli Forward-Backward Smoothing,” Proc. Int. Conf. Inf. Fusion, July 2016.

[53] J. Ferryman, IEEE Workshop Performance Evaluation of Tracking and Surveillance, 2009.

[54] R. Stiefelhagen, K. Bernardin, R. Bowers, J.S. Garofolo, D. Mostefa, and P. Soundararajan, “The CLEAR 2006 Evaluation,” Proc. First Int’l Evaluation Conf. Classification of Events, Activities and Relationships, 2006.