

# A Bayesian Filter for Multi-view 3D Multi-object Tracking with Occlusion Handling

Jonah Ong, Ba-Tuong Vo, Ba-Ngu Vo, Du Yong Kim and Sven Nordholm

**Abstract**—This paper proposes an online multi-camera multi-object tracker that only requires monocular detector training, independent of the multi-camera configurations, allowing seamless extension/deletion of cameras without retraining effort. The proposed algorithm has a linear complexity in the total number of detections across the cameras, and hence scales gracefully with the number of cameras. It operates in the 3D world frame, and provides 3D trajectory estimates of the objects. The key innovation is a high fidelity yet tractable 3D occlusion model, amenable to optimal Bayesian multi-view multi-object filtering, which seamlessly integrates, into a single Bayesian recursion, the sub-tasks of track management, state estimation, clutter rejection, and occlusion/mis-detection handling. The proposed algorithm is evaluated on the latest WILDTRACKS dataset, and demonstrated to work in very crowded scenes on a new dataset.

**Index Terms**—Multi-view, Multi-sensor, Multi-object Visual Tracking, Occlusion Handling, Generalized Labeled Multi-Bernoulli

## 1 INTRODUCTION

THE interest of visual tracking is to jointly estimate an unknown time-varying number of object trajectories from a stream of images [1]. The challenges of visual tracking are the random appearance/disappearance of the objects, false positives/negatives, and data association uncertainty [2]. Multiple object tracking (MOT) algorithms can operate online to produce current estimates as data arrives, or in batch which delay the estimation until further data is available [3], [4]. In principle, batch algorithms are more accurate than online as they allow better data integration into the estimates [2], [5], [6], [7]. Online algorithms, however, tend to be faster and hence better suited for time-critical applications [4], [8], [9], [10], [11].

The common sub-tasks, traditionally performed by separate modules in a MOT system are track management, state estimation, clutter rejection, and occlusion/mis-detection handling. Track management involves the initiation, termination and identification of trajectories of individual objects, while state estimation is concerned with determining the state vectors of the trajectories. Problems such as track loss, track fragmentation and identity switching are caused by false negatives that can arise from occlusions when objects of interest are visually blocked from a sensor, or from mis-detections when the sensor/detector fails to register objects of interest. On the other hand, false positives can lead to false tracks and identity switching. Hence, occlusion/mis-detection handling and clutter rejection are critical for improving tracking performance.

While occlusion handling is just as challenging compared with the other sub-tasks, theoretical developments are

far and few [12]. This is due mainly to the complex object-to-object and object-to-background relationships, as well as computational tractability because, theoretically, all possible partitions of the set of objects need to be considered [4]. In a single-view setting, useful *a priori* information about the objects of interest are exploited to resolve occlusions [2], [6], [11], [13]. However, there are fundamental limitations on what can be achieved with single-view data. In contrast, a multi-view setting naturally allows exploiting complementary information from the data to resolve occlusions since an object occluded in one view may not be occluded in another [14]. Furthermore, from an information theoretic standpoint, data from diverse views will reduce the uncertainty on the set of objects of interest, thereby improving overall tracking performance. Given the proliferation of cameras in today's world, it is imperative to develop effective means for making the best of the information-rich multi-view data sources, not only for occlusion handling, but ultimately to achieve better visual tracking.

The perennial challenge in multi-view visual MOT is the high-dimensional data association problem between the detections and objects, across different views/cameras [12], [15]. Two common architectures for multi-view MOT are shown in Fig. 1. So far the best solutions are batch algorithms with the architecture in Fig. 1 (a). These solutions are based on: generative modeling and dynamic programming [15]; convolutional neural network (CNN) multi-camera detection (MCD), trained on multi-view datasets [16], followed by track management [17]; and MCD via multi-view CNN training combined with Conditional Random Fields (CRF) models to exploit multi-camera geometry (followed by track management) [18]. These MCD based MOT solutions, which produce trajectories on the ground plane, have been shown to outperform previous works [16], and demonstrated remarkable performance in crowded scenarios [18]. Note that such data-centric MCDs require retraining when the multi-camera system is extended/reconfigured, and that training/learning is expensive as the input space is very high-dimensional due to the large number of possible combina-

- 
- J. Ong, B.T. Vo, B.N. Vo, and S. Nordholm are with the Department of Electrical and Computer Engineering, Curtin University, Bentley, WA 6102, Australia.  
E-mail: {j.ong1, ba-tuong.vo, ba-ngu.vo, s.nordholm}@curtin.edu.au
  - D.Y. Kim is with the School of Engineering, RMIT University, Melbourne, Australia.  
E-mail: duyong.kim@rmit.edu.au

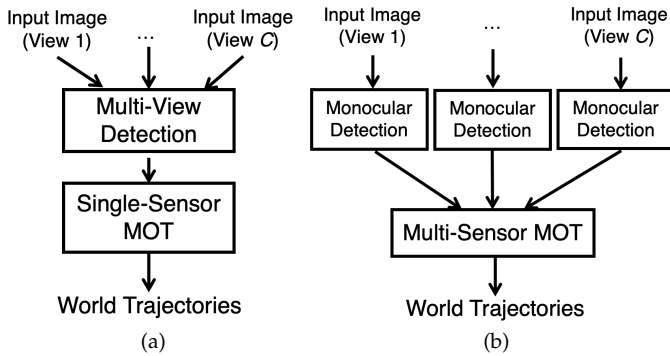


Fig. 1: Multi-view Architectures: (a) Multi-view Detection + Single-sensor Multi-object Tracking [17]; (b) Monocular Detection + Multi-sensor Multi-object Tracking.

tions across the cameras [19]. In practice, it is desirable for a multi-view MOT system to produce trajectories in 3D world frame, online, and requires no retraining for multi-camera extension/reconfiguration (including camera failures) so as to operate uninterrupted.

This paper proposes a model-centric, online multi-view visual MOT solution that only requires monocular detector training, independent of the multi-camera configurations, via the architecture of Fig. 1 (b). Hence, no retraining of the detectors is needed when the multi-camera system is extended/reconfigured. More importantly, our algorithm has a linear complexity in the total number of detections, thereby scales gracefully with the number of cameras. The algorithm intrinsically operates in the 3D world frame by exploiting multi-camera geometry, allowing it to track people jumping and falling, suitable for applications such as sports analytics, age care, school environment monitoring, etc. We validate the proposed method on the latest WILDTRACKS dataset on ground plane and show comparable results with Deep-Occlusion+KSP+ptrack [17]. To evaluate tracking performance in the 3D world frame, we develop a new dataset with varying degrees of difficulties on scenarios with very closely spaced people, with addition/deletion of cameras during operation, and with people jumping and falling.

The key innovation is a high fidelity yet tractable 3D occlusion model, amenable to Bayesian multi-sensor multi-object filtering [20], which seamlessly integrates, into a single Bayesian recursion, the sub-tasks of track management, state estimation, clutter rejection, and occlusion/misdetection handling. In the Bayesian paradigm, the multi-object filtering density captures all information on the set of trajectories in 3D, encapsulated in the observations, as well as dynamic and observation models. The novel occlusion model, incorporated in the multi-object measurement likelihood function, enables the MOT Bayesian filter to correctly maintain occluded tracks that would have otherwise been incorrectly terminated. The schematic in Fig. 2 shows the integration of the novel occlusion model into a near-optimal multi-sensor multi-object Bayes filter known as the Multi-Sensor Generalized Labeled Multi-Bernoulli (MS-GLMB) filter [20]. This configuration enables the proposed algorithm, herein referred to as Multi-View GLMB with Occlusion modeling (MV-GLMB-OC), to address occlusions, and inherits the numerical efficiency of the MS-GLMB filter. In short, our main technical contributions are:

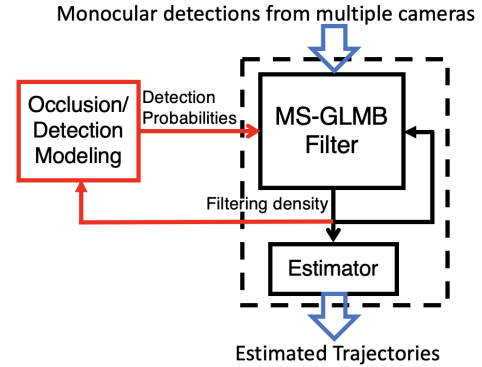


Fig. 2: MV-GLMB-OC filter Processing Chain. Monocular detections from multiple cameras are fed into the filter, which outputs the filtering density. This output is fed into: the estimator to generate track estimates; and back into the filter to process detections at the next time. The Occlusion Model (red) is an add-on that takes the filter output and compute the detection probabilities for the filter on-the-fly.

- A tractable and realistic detection model that accommodates 3D occlusion by taking into account the Lines of Sights (LoSs) of all objects in the scene with respect to the cameras. In contrast, conventional detection models either neglect the LoSs of the objects or are computationally intractable, leading to poor tracking performance in the presence of occlusions. Our new detection model can be regarded as a generalization of tractable conventional detection models;
- The first Bayesian multi-view MOT filter for such detection model, which resolves occlusion online and is scalable with the number of sensors. Experiments show better performance than the latest multi-camera tracking algorithm;
- A new dataset with full 3D annotations (not restricted to the ground plane), in terms of position and extent in all 3  $x$ ,  $y$ ,  $z$ -coordinates, including sequences that involve changes in the  $z$ -coordinate due to people jumping and falling. Instead of reporting performance for the entire scenario duration (as done traditionally), we also introduce live or online tracking performance evaluation over time, using the OSPA<sup>(2)</sup> metric [21], to characterize the behavior of the algorithm and demonstrate uninterrupted operation when the multi-camera system is extended/reconfigured.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 formulates the multi-view MOT problem, including the proposed occlusion/detection model, and the new tractable filter with occlusion handling capability via optimal Bayesian estimation. Section 4 presents the implementation of the algorithm. Section 5 shows experimental results and discussions. Finally, some conclusions are drawn in Section 6.

## 2 RELATED WORK

A Deep Convolutional Neural Network (CNN) trained on large-scale high-resolution image dataset, with efficient implementations such as Fast/Faster R-CNN [22], [23], has been shown to outperform all previous object detectors based on hand-engineered features, e.g. the Aggregated Channel Features (ACF) object detector [24]. Faster R-CNN

introduces the concept of Region Proposal Network (RPN) and exploits feature sharing together with efficient multi-scale solution to improve test-time speed and detection accuracy, achieving real-time detection at 5 frames-per-second (fps) [23]. Recently, the You Only Look Once (YOLO) real-time object detector, which attains 40fps at mAP of 76.8% (resolution of 544x544) on PASCAL VOC 2007, has gained immense popularity [25]. In contrast to the aforementioned techniques that rely on a sliding classifier for every image, YOLO's impressive speed is achieved by only scanning the image once. Additionally, spatial constraints, introduced to eliminate unlikely bounding boxes, allow trade-offs between speed and accuracy via a suitable score threshold [26]. The YOLO detector can also be extended to 3D [27]. The main drawback is the inability to detect small objects due to the imposed spatial constraints [26].

Progress in object detections facilitated the development of many tracking-by-detection approaches that typically join the detections together to form consistent trajectories [8], [28], [29]. Tracking-by-detection can be designed for batch or online operations. Online algorithms tend to be faster and better suited for time-critical applications, but may be prone to irrevocable errors if objects are undetected in several frames or if detections at different times are incorrectly joined [2]. Such errors can be reduced by global trajectory optimization over batches of frames [2], [3], [5], [6], [7]. However, track loss and fragmentation can still be caused by occlusion, which is an active area of research in itself [28]. In single-view/monocular settings, a popular approach to occlusion handling is to exploit *a priori* knowledge of the scene [2], [6], [7]. Deep neural network techniques that leverage spatio-temporal information in the images have shown to perform well in autonomous driving [30], [31].

In a multi-view setting, complementary information from the data can be exploited to resolve occlusions naturally, since an object occluded in one view may not be occluded in another view [14]. The hierarchical composition approach in [3] uses monocular information from multiple views to construct estimates in the ground plane. However, this approach is susceptible to reprojection errors and ignores occlusions [18]. In [32], the author formulates an occlusion model based on 2D silhouette-based visual angles from multiple views. Subsequently, a simple approach is to pre-process images from individual views (e.g. via background subtraction) from which occupancy (on the ground plane) can be estimated using Probability of Occupancy Map (POM) [15]. A more sophisticated approach was proposed in [12], which combines multi-view Bayesian network modeling of occlusion relationship and homography correspondence, across all views, with height-adaptive projection (HAP) to obtain final ground plane detections [12]. Stereo-based MOT approaches have also demonstrated improved 3D object estimation and tracking [33], [34], [35].

So far, the best multi-view tracking solution is based on a multi-camera detection (MCD) architecture that uses a CNN to train multi-view detectors from monocular and multi-view data [16], together with batch processing to compute global trajectories on the ground plane [17]. Combined with Conditional Random Field (CRF) modeling and Mean Field variational inference, this approach achieves remarkable performance in crowded scenarios [18]. This approach is

more data-centric than model-centric as the multi-camera detection relies mostly on training from data. Hence, large training sets are required, and the learning algorithm tends to be computationally expensive in exploring tight convergence levels, especially for high dimensional scenarios (e.g. large number of cameras) [19]. More examples of deeply learned multi-view approaches are found in [36], [37]. To the best of our knowledge, no online MOT algorithm has produced comparable tracking performance with these data-centric batch solutions.

In practice, it is desirable to have online algorithms whose complexity scale linearly with the number of cameras, and do not require multi-view training so that re-configuration (including addition and deletion) of cameras can be performed without interruption to the operation. Moreover, in a multi-view context, it is more prudent to have trajectories in the 3D world frame for applications such as sports analytics, age care, school environment monitoring, etc. While there are solutions to online 3D multi-view MOT with monocular data such as [38], [39], they do not scale gracefully with the number of cameras. Similar to the mentioned batch-processing methods, these solutions are more data-centric as they rely, respectively, on deep training for object depth information, and motion learning.

At the other end of the spectrum are the model-centric approaches that rely largely on physical models of the dynamics of the objects, the geometry and characteristics of the sensors/cameras. Such model-based solutions to 3D online MOT with monocular data, using 2D object detections, 3D object proposals, and 3D point cloud techniques were developed, respectively, in [33], [40], [41]. From a state-space modeling perspective, a natural choice for online MOT is the multi-object Bayes filter [42]. Since the inception of the Random Finite Sets (RFS) framework for multi-object state-space models, a number of multi-objects Bayesian filters have been developed [43], [44] and applied to visual MOT problems [4], [10], [45]. The latest is the Generalized labeled Multi-Bernoulli (GLMB) filter, an analytic solution to the multi-object Bayes filter that jointly estimates the number of objects and their trajectories online [46]. The salient feature of this approach is that it seamlessly integrates track management, state estimation, clutter rejection, occlusion/misdetection handling and multiple sensor data into a single recursion [4]. In this article, we use this framework to develop an online 3D multi-view MOT solution that only requires one-off monocular detector training (or off-the-shelf monocular detectors), yet is capable of producing comparable results with the aforementioned data-centric batch-processing approaches.

In addition to algorithms, datasets for performance evaluation are an important aspect of 3D multi-view MOT research. Existing multi-view datasets include DukeMTMC [47], PETS 2009 S2.L1 [48], EPFL - Laboratory, Terrace and Passageway [15], SALSA [49], Campus [3] and EPFL-RLC [16]. However, in [17] the authors discussed a number of their shortcomings and introduced a seven-camera high-definition (HD) unscripted pedestrian dataset known as WILDTRACKS to provide a high quality, highly crowded and cluttered evaluation scenario. It comes with accurate joint (extrinsic and intrinsic) calibration, and 7 series of 400 annotated frames for detection at a rate of 2 frames per

TABLE 1: Basic Notation

Symbol	Description
$a^T$	Transpose of vector/matrix $a$
$\otimes$	Kronecker product (for matrices)
$I_n$	$n$ -dimensional identity matrix
$0_{n \times m}$	$n$ by $m$ zero matrix
$\text{diag}(\cdot)$	Converts a vector to a diagonal matrix
$X_{m:n}$	$X_m, X_{m+1}, \dots, X_n$
$\langle f, g \rangle$	$\int f(x)g(x)dx$
$h^X$	$\prod_{x \in X} h(x)$ where $h^0 = 1$
$\delta_Y[X]$	Kronecker delta function: 1 if $X = Y$ , 0 otherwise
$1_Y(x)$	Indicator function: 1 if $x \in Y$ , 0 otherwise
$\mathcal{N}(\cdot; \mu, P)$	Gaussian <i>pdf</i> with mean $\mu$ and covariance $P$

second (fps). The annotations of the tracks are given both as locations on the ground plane and 2D bounding boxes projected onto each view.

While WILDTRACKS is more extensive than earlier datasets, it is still not sufficient for comprehensive 3D MOT performance evaluation. Specifically, for actual 3D MOT applications where objects may also move vertically (e.g. sport analytics, age care, etc.), ground plane annotations are simply not adequate for evaluating tracking performance in full 3D, i.e. changes in all 3  $x, y, z$ -coordinates. To enrich the datasets and to enable performance evaluation in full 3D, we propose the Curtin Multi-Camera (CMC) dataset that comprises four calibrated cameras, on scenarios of varying difficulties in crowd density and occlusion, as well as scenarios with people jumping and falling, all with 3D centroid-with-extent annotations, along with camera locations and parameters. Note that in addition to extrinsic and intrinsic parameters, we also provide the absolute camera locations needed for testing and evaluation of model-centric solutions that exploit multi-camera geometry.

### 3 BAYESIAN FORMULATION

This section formulates the multi-view MOT problem (Sections 3.1-3.4), including the proposed occlusion/detection model (Section 3.5), and the new tractable filter with occlusion handling capability (Section 3.6). The notations used in this paper are tabulated in Table 1.

#### 3.1 Bayes Filter

We first recall the classical Bayesian filter where the state  $x$  of the object, in some finite dimensional state space  $\mathbb{X}$ , is modeled as a random vector. The dynamic of the state is described by a Markov chain with transition density  $f_+(x_+|x)$ , i.e. the probability density of a transition to the state  $x_+$  at the next time given the current state  $x$ . Note that for simplicity we omit the subscript for current time and use the subscript '+' denotes the next time step. Additionally, the current state  $x$  generates an observation  $z$  described by the likelihood function  $g(z|x)$ , i.e. the probability density of receiving the observation  $z$  given  $x$ . All information on the current the state is encapsulated in the filtering density<sup>1</sup>  $p$ , which can be propagated to the next time as  $p_+$ , via the celebrated Bayes recursion [50]

$$p_+(x_+) \propto g(z_+|x_+) \int f_+(x_+|x) p(x) dx. \quad (1)$$

1. The filtering densities are conditioned on the observations, which have been omitted for notational compactness.

The multi-view MOT Bayes filter used in this work is conceptually identical to the classical Bayes filter above by replacing:  $x$  and  $x_+$  with the sets  $\mathbf{X}$  and  $\mathbf{X}_+$ ;  $p$  and  $p_+$  with the multi-object filtering densities  $\pi$  and  $\pi_+$ ;  $f_+$  and  $g$  with the multi-object transition density  $\mathbf{f}_+$  and multi-object observation likelihood  $\mathbf{g}$ ;  $z_+$  with the observation set  $Z_+$ ; and the integral with the set integral [43], i.e.

$$\pi_+(\mathbf{X}_+) \propto \mathbf{g}(Z_+|\mathbf{X}_+) \int \mathbf{f}_+(\mathbf{X}_+|\mathbf{X}) \pi(\mathbf{X}) \delta \mathbf{X}. \quad (2)$$

The sets  $\mathbf{X}$  (and  $\mathbf{X}_+$ ) containing the object states at the current (and next) time, is called the current (and next) multi-object state. Each element of the multi-object state  $\mathbf{X}$  is an ordered pair  $\mathbf{x} = (x, \ell)$ , where  $x \in \mathbb{X}$  is a state vector, and  $\ell \triangleq (t, \alpha)$  is a unique label consisting of the object's time of birth  $t$ , and an index  $\alpha$  to distinguish those born at the same time [46]. The cardinality (number of elements) of  $\mathbf{X}$  and  $\mathbf{X}_+$  may differ due to the appearance and disappearance of objects from one frame to the next.

Under the Bayesian paradigm, the multi-object state is modeled as a random finite set, i.e. a finite-set-valued random variable, characterized by Mahler's multi-object density [43], [44] (equivalent to a probability density [51]). The multi-object transition density  $\mathbf{f}_+$  captures the motions as well as births and deaths of objects. The multi-object observation likelihood  $\mathbf{g}$  captures the detections, false alarms, occlusions, and misdetections.

#### 3.2 Motion and Birth/Death Models

An object at time  $k$ , represented by a state  $\mathbf{x} = (x, \ell)$ , either survives with probability  $P_S(\mathbf{x})$  and evolves to state  $\mathbf{x}_+ = (x_+, \ell_+)$  at the next time with transition density

$$\mathbf{f}_{S,+}(\mathbf{x}_+|\mathbf{x}) = f_{S,+}(x_+|x, \ell) \delta_{\ell}[\ell_+], \quad (3)$$

or dies with probability  $1 - P_S(\mathbf{x})$  [46]. At this next time, an object with label  $\ell$  is born with probability  $P_{B,+}(\ell)$ , and with feature-vector  $x$  distributed according to a probability density  $f_{B,+}(\cdot, \ell)$ . Note that the label of an object remains the same over time, and hence the *trajectory* of an object is a sequence of consecutive states with a common label [46].

Let  $\mathbb{B}_k$  denote the finite set of all possible labels for objects born at time  $k$ , then the label space for all objects up to time  $k$  is the disjoint union  $\mathbb{L}_k = \bigsqcup_{t=0}^k \mathbb{B}_t$ . For simplicity we omit the time subscript  $k$ , and let  $\mathcal{L}(\mathbf{x})$  denote the label of an  $\mathbf{x} \in \mathbb{X} \times \mathbb{L}$ . For any finite  $\mathbf{X} \subset \mathbb{X} \times \mathbb{L}$ , we define  $\mathcal{L}(\mathbf{X}) \triangleq \{\mathcal{L}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$ , and the *distinct label indicator*  $\Delta(\mathbf{X}) \triangleq \delta_{|\mathcal{X}|} [|\mathcal{L}(\mathbf{X})|]$ . At any time, the set  $\mathbf{X}$  of (states of) objects in the scene must have distinct labels, i.e.  $\Delta(\mathbf{X}) = 1$ . Conditional on the current set of objects, it is standard practice to assume that objects are born or displaced at the next time, independently of one another. The expression for the multi-object transition density  $\mathbf{f}_+$  is not needed in this work, interested readers are referred to [46].

#### 3.3 Multi-Sensor Observation Model

Suppose that at time  $k$ , there are  $C$  cameras (sensors), and a set  $\mathbf{X}$  of current objects. Each  $\mathbf{x} \in \mathbf{X}$  is either: detected by camera  $c \in \{1:C\}$ , with probability  $P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\})$  and generates an observation  $z^{(c)}$  in the measurement space  $\mathbb{Z}^{(c)}$  with likelihood  $g^{(c)}(z^{(c)}|\mathbf{x})$ ; or missed with probability

$1 - P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\})$ . Note that to account for occlusions (and uncertainty in the detection process), the probability of detecting an object  $\mathbf{x}$  also depends on the states of other current objects  $\mathbf{X} - \{\mathbf{x}\}$ . However, most MOT algorithms neglect this dependence for computational tractability.

The detection process also generates false positives at camera  $c$ , usually characterized by an intensity function  $\kappa^{(c)}$  on  $Z^{(c)}$ . The standard model is a Poisson distribution, with mean  $\langle \kappa^{(c)}, 1 \rangle$ , for the number of false positives, and the false positives themselves are i.i.d. according to the probability density  $\kappa^{(c)} / \langle \kappa^{(c)}, 1 \rangle$  [44], [52], [53]. Moreover, conditional on the set  $\mathbf{X}$  of objects, detections are assumed to be independent from false positives, and that the set  $Z^{(c)}$  of detections and false positives at sensor  $c$ , are independent from those at other sensors.

An association hypothesis (at time  $k$ ) associating labels with detections from camera  $c$  is a mapping  $\gamma^{(c)}: \mathbb{L} \rightarrow \{-1:|Z^{(c)}|\}$ , such that *no two distinct arguments are mapped to the same positive value* [46]. This property ensures each detection comes from at most one object. Given an association hypothesis  $\gamma^{(c)}: \gamma^{(c)}(\ell) = -1$  means object  $\ell$  does not exist;  $\gamma^{(c)}(\ell) = 0$  means object  $\ell$  is not detected by camera  $c$ ;  $\gamma^{(c)}(\ell) > 0$  means object  $\ell$  generates detection  $z_{\gamma^{(c)}(\ell)}$  at camera  $c$ ; and the set  $\mathcal{L}(\gamma^{(c)}) \triangleq \{\ell \in \mathbb{L} : \gamma^{(c)}(\ell) \geq 0\}$  are the *live labels* of  $\gamma^{(c)}$ . Under standard assumptions, the (multi-object) likelihood for camera  $c$  is given by the following sum over the space  $\Gamma^{(c)}$  of association hypotheses with domain  $\mathbb{L}$  and range  $\{-1:|Z^{(c)}|\}$  [46]:

$$\mathbf{g}^{(c)}(Z^{(c)}|\mathbf{X}) \propto \sum_{\gamma^{(c)} \in \Gamma^{(c)}} \delta_{\mathcal{L}(\gamma^{(c)})}[\mathcal{L}(\mathbf{X})] \left[ \psi_{\mathbf{X}-\{\cdot\}}^{(c, \gamma^{(c)})}(\cdot) \right]^{\mathbf{X}}, \quad (4)$$

where  $Z^{(c)} = \{z_{1:|Z^{(c)}|}\}$ , and

$$\psi_{\mathbf{X}-\{\mathbf{x}\}}^{(c, \gamma^{(c)})}(\mathbf{x}) = \begin{cases} 1 - P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\}), & \gamma^{(c)}(\mathcal{L}(\mathbf{x})) = 0 \\ \frac{P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\}) \mathbf{g}^{(c)}(z_j^{(c)}|\mathbf{x})}{\kappa^{(c)}(z_j^{(c)})}, & \gamma^{(c)}(\mathcal{L}(\mathbf{x})) = j > 0 \end{cases}, \quad (5)$$

Note that  $\psi_{\mathbf{X}-\{\mathbf{x}\}}^{(c, \gamma^{(c)})}(\mathbf{x})$  also depends on  $Z^{(c)}$ , but we omitted it for clarity. Interested readers are referred to the texts [43], [44] for the derivation/discussion.

A multi-sensor (association) hypothesis is an array  $\gamma \triangleq (\gamma^{(1)}, \dots, \gamma^{(C)})$  of association hypotheses with the same set of live labels, denoted as  $\mathcal{L}(\gamma)$ . The likelihood that  $\mathbf{X}$  generates the multi-sensor observation  $Z \triangleq (Z^{(1:C)})$  is the product  $\prod_{c=1}^C \mathbf{g}^{(c)}(Z^{(c)}|\mathbf{X})$ , which can be rewritten as [20]

$$\mathbf{g}(Z|\mathbf{X}) \propto \sum_{\gamma \in \Gamma} \delta_{\mathcal{L}(\gamma)}[\mathcal{L}(\mathbf{X})] \left[ \psi_{\mathbf{X}-\{\cdot\}}^{(\gamma)}(\cdot) \right]^{\mathbf{X}}, \quad (6)$$

where  $\Gamma$  is the set of all multi-sensor hypotheses,

$$\delta_{\mathcal{L}(\gamma)}[J] \triangleq \prod_{c=1}^C \delta_{\mathcal{L}(\gamma^{(c)})}[J], \quad (7)$$

$$\psi_{\mathbf{X}-\{\mathbf{x}\}}^{(\gamma)}(\mathbf{x}) \triangleq \prod_{c=1}^C \psi_{\mathbf{X}-\{\mathbf{x}\}}^{(c, \gamma^{(c)})}(\mathbf{x}). \quad (8)$$

Remark: The sets of objects, observations, and possibly the number of sensors and their parameters, may vary with time. However, for clarity we suppressed the time index.

### 3.4 Multi-Sensor GLMB Filter

Most of the literature on tracking assumes the probability of detection  $P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\}) = P_D^{(c)}(\mathbf{x})$ , i.e. independent of  $\mathbf{X} - \{\mathbf{x}\}$ . In this case, the Bayes recursion (2) admits an analytical solution based on Generalized Labeled Multi-Bernoulli (GLMB) models.

A GLMB is a multi-object density of the form [46]

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{I, \xi} w^{(I, \xi)} \delta_I[\mathcal{L}(\mathbf{X})] \left[ p^{(\xi)} \right]^{\mathbf{X}}, \quad (9)$$

where:  $I \in \mathcal{F}(\mathbb{L})$  the space of all finite subsets of  $\mathbb{L}$ ;  $\xi \in \Xi$  the space of all (multi-sensor) association hypotheses histories up to the current time, i.e.  $\xi \triangleq \gamma_{1:k}$ ; each  $w^{(I, \xi)}$  is a non-negative weight such that  $\sum_{I, \xi} w^{(I, \xi)} = 1$ ; and each  $p^{(\xi)}(\cdot, \ell)$  is a probability density on  $\mathbb{X}$ . For convenience, we represent a GLMB by its parameter-set

$$\pi \triangleq \left\{ \left( w^{(I, \xi)}, p^{(\xi)} \right) : (I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi \right\}. \quad (10)$$

Each GLMB *component*  $(I, \xi)$  can be interpreted as a hypothesis with probability  $w^{(I, \xi)}$ , and each individual object  $\ell \in I$  of this hypothesis has probability density  $p^{(\xi)}(\cdot, \ell)$ .

A simple multi-object state estimate can be obtained from a GLMB by first determining: the most probable cardinality  $n^*$  from the cardinality distribution [46]

$$\text{Prob}(|\mathbf{X}| = n) = \sum_{I, \xi} \delta_n[|I|] w^{(I, \xi)}; \quad (11)$$

and then the hypothesis  $(I^*, \xi^*)$  with highest weight such that  $|I^*| = n^*$ . The current state estimate for each object  $\ell \in I^*$  can be computed from  $p^{(\xi^*)}(\cdot, \ell)$ , e.g. the mode or mean. Alternatively, the entire trajectory of object  $\ell \in I^*$  can be estimated using the forward-backward algorithm, starting from its current filtering density  $p^{(\xi^*)}(\cdot, \ell)$  and propagating backward to its time of birth [20], [54].

Under the Bayes recursion (2), and the standard multi-object model (i.e. with no occlusions,  $P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\}) = P_D^{(c)}(\mathbf{x})$ ), the multi-object filtering density at any time is a GLMB [46]. Moreover, if (10) is the current GLMB filtering density, then the next GLMB filtering density

$$\pi_+ = \left\{ \left( w_+^{(I_+, \xi_+)}, p_+^{(\xi_+)} \right) : (I_+, \xi_+) \in \mathcal{F}(\mathbb{L}_+) \times \Xi_+ \right\}, \quad (12)$$

can be computed via the *MS-GLMB recursion* [20]

$$\pi_+ = \Omega(\pi; P_{D,+}), \quad (13)$$

where  $P_{D,+} \triangleq (P_{D,+}^{(1)}, \dots, P_{D,+}^{(C)})$ . The actual mathematical expressions for the recursion operator  $\Omega: \pi \mapsto \pi_+$  are not critical for our arguments, and hence omitted from this section. Nonetheless, for completeness the definition of  $\Omega$  is provided in Appendix 7.1. Note that  $\Omega$  also depends on the measurement  $Z_+$ , and model parameters for birth  $(P_{B,+}, f_{B,+})$ , death/survival  $P_S$ , motion  $f_{S,+}$ , false alarms  $\kappa_+ \triangleq (\kappa_+^{(1)}, \dots, \kappa_+^{(C)})$ , and detection  $g_+ \triangleq (g_+^{(1)}, \dots, g_+^{(C)})$  (described in Section 3.3). However, for our purpose it suffices to show the dependence on detection probabilities.

While the MS-GLMB filter can applied directly to multi-view MOT, a detection probability (of an object  $\mathbf{x}$ ) that does not depend on other objects, i.e.  $\mathbf{X} - \{\mathbf{x}\}$ , is unable to capture the effect of occlusions. On the other hand, accounting for occlusions with  $P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\})$  that actually depends

on  $\mathbf{X} - \{\mathbf{x}\}$ , results in filtering densities that are not GLMBs. One example is the merged-measurement model [55], which involves summing over all partitions of the set  $\mathbf{X}$ , making it intractable [55]. Although the resulting filtering density can be approximated by a GLMB, this solution is still computationally demanding and not suitable for large number of objects [55]. In what follows, we propose a new detection model that addresses occlusions and permits efficient multi-view MOT implementations.

### 3.5 Detection Model with Occlusion

For tracking in 3D, we consider the state  $\mathbf{x} = (x, \ell)$ , where:

$$\mathbf{x} = (x^{(p)}, \dot{x}^{(p)}, x^{(s)}); \quad (14)$$

$x^{(p)}$  is the object's position (centroid) in 3D Cartesian coordinates;  $\dot{x}^{(p)}$  is its velocity; and  $x^{(s)}$  is its shape parameter. The region in  $\mathbb{R}^3$  occupied by an object with labeled state  $\mathbf{x}$  is denoted by  $R(\mathbf{x})$ .

Consider camera  $c$  and the set  $\mathbf{X}$  of current objects. In this work, an object  $(x, \ell) \in \mathbf{X}$  is regarded as occluded from camera  $c$  when its position  $x^{(p)}$  is not in the line of sight (LoS) of the camera, i.e.  $x^{(p)}$  is in the *shadow regions* of the other objects in  $\mathbf{X}$ . Assuming straight LoSs, the shadow region of an object with labeled state  $\mathbf{x}'$ , relative to camera  $c$  (see Fig. 3), is given by

$$S^{(c)}(\mathbf{x}') = \left\{ y \in \mathbb{R}^3 : \overline{(u^{(c)}, y)} \cap R(\mathbf{x}') \neq \emptyset \right\}, \quad (15)$$

where  $\overline{(u^{(c)}, y)} \triangleq \{\lambda y + (1 - \lambda)u^{(c)} : \lambda \in [0, 1]\}$  is the line segment joining the position  $u^{(c)}$  of camera  $c$  and  $y$ . Note that for an ellipsoidal region  $R(\mathbf{x}')$ , the indicator function  $1_{S^{(c)}(\mathbf{x}')}$  of its shadow region can be computed in closed form (see Section 4.1).

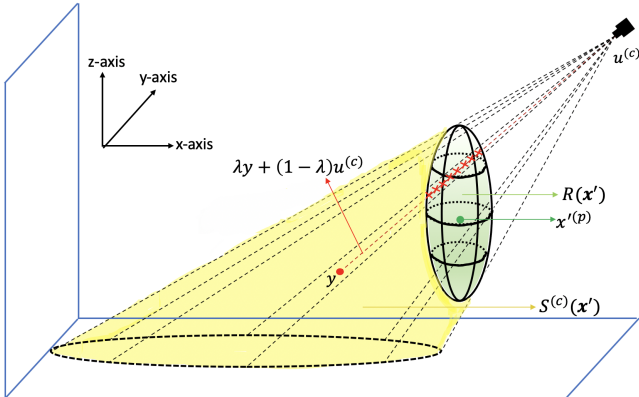


Fig. 3: The shadow region (in yellow) of object with labeled state  $\mathbf{x}'$ , relative to camera  $c$ .

To incorporate the effect of occlusions into the detection model, the probability that  $\mathbf{x} \in \mathbf{X}$  be detected by camera  $c$  should be close to zero when it is occluded from camera  $c$ . This can be accomplished by extending the standard detection probability so that: when  $\mathbf{x}$  is in the LoS of camera  $c$ , its detection probability is  $P_D^{(c)}(\mathbf{x})$ ; and when occluded by the other objects its detection probability scales down to  $\beta P_D^{(c)}(\mathbf{x})$ , where  $\beta$  is a small positive number. More explicitly,

$$\begin{aligned} P_D^{(c)}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\}) = \\ P_D^{(c)}(\mathbf{x}) \left( \mathcal{M}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\}) + \beta(1 - \mathcal{M}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\})) \right), \quad (16) \end{aligned}$$

where

$$\mathcal{M}(\mathbf{x}; \mathbf{X} - \{\mathbf{x}\}) = \prod_{\mathbf{x}' \in \mathbf{X} - \{\mathbf{x}\}} \left( 1 - 1_{S^{(c)}(\mathbf{x}')}(\mathbf{x}) \right) \quad (17)$$

Conditional on detection,  $\mathbf{x}$  is observed at camera  $c$  as a bounding box  $z^{(c)} \triangleq (z_p^{(c)}, z_e^{(c)})$ , where  $z_p^{(c)}$  is the center, and  $z_e^{(c)}$  is the extent, parameterized by the logarithms of the width (x-axis) and height (y-axis), in image coordinates. The observed  $z^{(c)}$  is a noisy version of the box  $\Phi^{(c)}(\mathbf{x})$  bounding the image of  $R(\mathbf{x})$  in the camera's image plane, under the projection of the camera matrix  $P_{3 \times 4}^{(c)}$ . This matrix projects homogeneous points in the world coordinate frame to homogeneous points in the image plane of camera  $c$ , and can be obtained by standard calibration techniques (see [56] for details). Note that for an ellipsoidal region  $R(\mathbf{x})$ , the axis-aligned  $\Phi^{(c)}(\mathbf{x})$  on the image plane can be computed analytically (see Section 4.1). This observation process can be modeled by the likelihood

$$\begin{aligned} g^{(c)}(z^{(c)} | \mathbf{x}) = \\ \mathcal{N} \left( z^{(c)}; \Phi^{(c)}(\mathbf{x}) + \begin{bmatrix} 0_{2 \times 1} \\ -v_e^{(c)}/2 \end{bmatrix}, \text{diag} \left( \begin{bmatrix} v_p^{(c)} \\ v_e^{(c)} \end{bmatrix} \right) \right), \quad (18) \end{aligned}$$

where  $v_p^{(c)}$  and  $v_e^{(c)}$  are respectively the vector of noise variances for the center and the extent (in logarithm) of the box. This Gaussian model of the logarithms of the width and height is equivalent to modeling the actual width and height as log-normals, which ensures that they are non-negative. Note that these log-normals have mean 1, and variances  $e^{v_{e,1}^{(c)}} - 1$  and  $e^{v_{e,2}^{(c)}} - 1$ , where  $v_{e,1}^{(c)}$  and  $v_{e,2}^{(c)}$  are the two components of  $v_e^{(c)}$ . This means the observed width and height are randomly scaled versions of their nominal values, with an expected scaling factor of 1.

### 3.6 Multi-view GLMB Filtering with Occlusions

This subsection presents a tractable GLMB approximation to the multi-view Bayes filter to address occlusions. The proposed filter (with the new detection model to account for occlusion) is referred to as Multi-View GLMB with occlusion modeling (MV-GLMB-OC).

Given the current GLMB filtering density (10), the predicted density  $\int \mathbf{f}_+(\mathbf{X}_+ | \mathbf{X}) \pi(\mathbf{X}) \delta \mathbf{X}$  in the Bayes recursion (2) is also a GLMB [46], which we denote by

$$\hat{\pi}_+(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{I_+, \xi} w_+^{(I_+, \xi)} \delta_{I_+}[\mathcal{L}(\mathbf{X}_+)] \left[ p_+^{(\xi)} \right]^{\mathbf{X}_+}, \quad (19)$$

where  $I_+ \in \mathcal{F}(\mathbb{L}_+)$ . Multiplying (19) by the likelihood (8) yields the next (unnormalized) multi-object density

$$\begin{aligned} \pi_+(\mathbf{X}_+) \propto \Delta(\mathbf{X}_+) \sum_{I_+, \xi, \gamma_+} \delta_{\mathcal{L}(\gamma_+)}[\mathcal{L}(\mathbf{X}_+)] w_+^{(I_+, \xi)} \\ \times \delta_{I_+}[\mathcal{L}(\mathbf{X}_+)] \left[ p_{\mathbf{X}_+ - \{\cdot\}}^{(\xi, \gamma_+)}(\cdot) \right]^{\mathbf{X}_+}, \quad (20) \end{aligned}$$

where

$$p_{\mathbf{X}_+ - \{\mathbf{x}_+\}}^{(\xi, \gamma_+)}(\mathbf{x}_+) = p_+^{(\xi)}(\mathbf{x}_+) \psi_{\mathbf{X}_+ - \{\mathbf{x}_+\}}^{(\gamma_+)}(\mathbf{x}_+). \quad (21)$$

As previously alluded to, the multi-object density (20) is not a GLMB because  $p_{\mathbf{X}_+ - \{\mathbf{x}_+\}}^{(\xi, \gamma_+)}$  depends on  $\mathbf{X}_+ - \{\mathbf{x}_+\}$ . Nonetheless, a good GLMB approximation of (20) can be

obtained by approximating  $p_{\mathbf{X}_+ - \{\mathbf{x}_+\}}^{(\xi, \gamma_+)}$  with a density that is independent of  $\mathbf{X}_+ - \{\mathbf{x}_+\}$ .

Note that  $\psi_{\mathbf{X}_+ - \{\mathbf{x}_+\}}^{(\gamma_+)}$  is the only factor of  $p_{\mathbf{X}_+ - \{\mathbf{x}_+\}}^{(\xi, \gamma_+)}$ , which depends on  $\mathbf{X}_+ - \{\mathbf{x}_+\}$  (see (21)). Further inspection of (5) and (8) reveals that the detection probability functions  $P_{D,+}^{(c)}(\cdot; \mathbf{X}_+ - \{\mathbf{x}_+\})$ ,  $c \in \{1:C\}$  are the only constituent terms that depend on  $\mathbf{X}_+ - \{\mathbf{x}_+\}$ . Moreover, it follows from (16) that  $P_{D,+}^{(c)}(\mathbf{x}_+; \mathbf{X}_+ - \{\mathbf{x}_+\})$  only takes on two values, depending on whether  $\mathbf{x}_+$  falls in the shadow region of  $\mathbf{X}_+ - \{\mathbf{x}_+\}$  w.r.t. camera  $c$ . Assuming the positions of the elements of  $\mathbf{X}_+ - \{\mathbf{x}_+\}$  are concentrated around their predicted values according to the prediction densities  $p_+^{(\xi)}(\cdot, \ell)$ ,  $\ell \in \mathcal{L}(\mathbf{X}_+ - \{\mathbf{x}_+\})$ , we can approximate  $P_{D,+}^{(c)}(\cdot; \mathbf{X}_+ - \{\mathbf{x}_+\})$  by replacing the set  $\mathbf{X}_+ - \{\mathbf{x}_+\}$  with its predicted value. Noting that the term  $\delta_{I_+}[\mathcal{L}(\mathbf{X}_+)]$  in (20) implies  $\mathcal{L}(\mathbf{X}_+) = I_+$ , the prediction of  $\mathbf{X}_+ - \{\mathbf{x}_+\}$  is

$$\mathbf{X}_+^{(\xi, I_+)} = \{(x_+^{(\xi, \ell)}, \ell) : \ell \in I_+ - \mathcal{L}(\mathbf{x}_+)\}, \quad (22)$$

where  $x_+^{(\xi, \ell)}$  denotes an estimate (e.g. mean, mode) from the density  $p_+^{(\xi)}(\cdot, \ell)$ , which is either the birth density  $f_{B,+}(\cdot, \ell)$  if  $\ell \in \mathbb{B}_+$  or  $\int f_{S,+}(\cdot | x, \ell) p^{(\xi)}(x, \ell) dx$  if  $\ell \notin \mathbb{B}_+$  [46].

The above approximation translates to

$$p_{\mathbf{X}_+ - \{\mathbf{x}_+\}}^{(\xi, \gamma_+)} \approx p_{\mathbf{X}_+^{(\xi, I_+)}}^{(\xi, \gamma_+)}, \quad (23)$$

which is independent of  $\mathbf{X}_+ - \{\mathbf{x}_+\}$ , thereby turning (20) into a GLMB. Moreover, the computation of this GLMB approximation to (20) only differs from the MS-GLMB recursion (13) in the detection probabilities

$$P_{D,+}^{(\xi, I_+)}(\ell) \triangleq \left( P_{D,+}^{(1)}((\hat{x}_+; \ell); \mathbf{X}_+^{(\xi, I_+)}) , \dots , P_{D,+}^{(C)}((\hat{x}_+; \ell); \mathbf{X}_+^{(\xi, I_+)}) \right), \quad (24)$$

where  $\ell = \mathcal{L}(\mathbf{x}_+)$ , and  $\hat{x}_+$  denotes an estimate (e.g. mean, mode) from the density  $p_+^{(\xi)}(\cdot, \ell)$ . Specifically, the GLMB approximation of the multi-object filtering density can be propagated by the MS-GLMB recursion

$$\pi_+ = \Omega\left(\pi; \{P_{D,+}^{(\xi, I_+)}(\ell) : \ell \in I_+, (\xi, I_+) \in \Xi \times \mathcal{F}(\mathbb{L}_+)\}\right). \quad (25)$$

The integration of the proposed occlusion model (via the detection probabilities) into the MS-GLMB filter is shown in Fig. 2. The implementation of this so-called MV-GLMB-OC filter is discussed in the next section.

## 4 IMPLEMENTATION

This section describe the implementation of the proposed filter for ellipsoidal objects. Section 4.1 provides mathematical representations for the objects and the multi-object model parameters. Propagation of the MV-GLMB-OC filtering density is then described in Section 4.2.

### 4.1 Object Representation and Model Parameters

Each object is represented by an axis-aligned ellipsoid. For an object with labeled state  $\mathbf{x} = (x, \ell)$ , the position  $x^{(p)}$  is the centroid, and the shape parameter  $x^{(s)}$  is a vector containing the logarithms of the half-lengths of the ellipsoid's principal axes. Further, the time-evolution of the state vector  $x$  is modeled by a linear Gaussian transition density:

$$f_{S,+}(x_+ | x, \ell) = \mathcal{N}\left(x_+; \mathbf{F}x + \begin{bmatrix} 0_{6 \times 1} \\ -v^{(s)}/2 \end{bmatrix}, \mathbf{Q}\right), \quad (26)$$

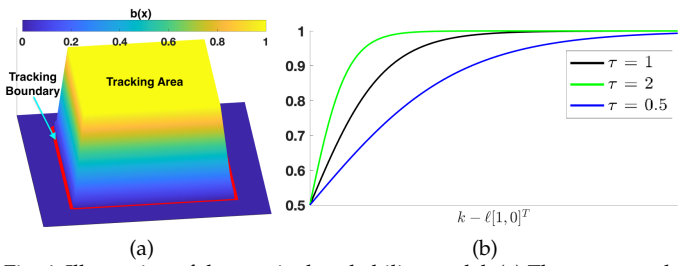


Fig. 4: Illustration of the survival probability model: (a) The scene mask  $b(x)$ ; (b) The control parameter  $\tau$  of the sigmoid function.

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} & 0_{6 \times 3} \\ 0_{3 \times 6} & \mathbf{I}_3 \end{bmatrix}, \quad (27)$$

$$\mathbf{Q} = \begin{bmatrix} \text{diag}(v^{(p)}) \otimes \begin{bmatrix} T/2 \\ T \end{bmatrix} & 0_{6 \times 3} \\ 0_{3 \times 6} & \text{diag}(v^{(s)}) \end{bmatrix}, \quad (28)$$

$T$  is the sampling period,  $v^{(p)}$  and  $v^{(s)}$  are, respectively, 3D vectors of noise variances for the components of the centroid and shape parameter (in logarithm) of the ellipsoid. This transition density describes a nearly constant velocity model for the centroid and a Gaussian random-walk for the shape parameter. Gaussianity of the logarithms of the half-lengths is equivalent to modeling the half-lengths as log-normals, which ensure that they are non-negative. Note that these log-normals have mean 1, and variances  $e^{v_i^{(s)}} - 1$ ,  $i = 1, 2, 3$ , where  $v_i^{(s)}$  is the  $i^{\text{th}}$  components of  $v^{(s)}$ . Hence, the observed half-lengths are randomly scaled versions of their nominal values, with an expected scaling factor of 1.

Empirically, objects that are in the scene for a long time, are more likely to remain in the scene, unless they are close to the borders (exit regions). This can be modeled via the following object survival probability [4]:

$$P_S(x, \ell) = \frac{b(x)}{1 + \exp(-\tau(k - \ell[1, 0]^T))}, \quad (29)$$

where  $b(x)$  is the the scene mask (chosen to be close to one in the middle of the scene, and close to zero in the designated exit regions and beyond) as depicted in Fig. 4 (a), and  $\tau$  is the control parameter of the sigmoid function that is dependent on the duration (age) of the track  $k - \ell[1, 0]^T$  as depicted in Fig. 4 (b).

The detection probability (16)-(17) can be computed in closed form when the objects extents are ellipsoids. As alluded to in Section 3.5, the shadow region indicator function  $1_{S^{(c)}}(\mathbf{y})(\cdot)$  used for checking whether an object is in the shadow region of the object  $\mathbf{y}$ , can be determined analytically. Suppose that  $R(\mathbf{y})$  in (15) is a quadric, then it intersects the line  $(u^{(c)}, x^{(p)})$  (between  $u^{(c)}$  and  $x^{(p)}$ ) if the roots of a certain quadratic equation are real [57]. Consequently, for an axis-aligned ellipsoidal object representation, the shadow region indicator function is given by

$$1_{S^{(c)}}(\mathbf{y})(\mathbf{x}) = \begin{cases} 1, & (\mathcal{B}_{\mathbf{x}, \mathbf{y}}^{(c)})^2 - 4\mathcal{A}_{\mathbf{x}, \mathbf{y}}^{(c)}\mathcal{C}_{\mathbf{y}}^{(c)} \geq 0 \\ 0, & \text{otherwise} \end{cases}, \quad (30)$$

where

$$\mathcal{A}_{\mathbf{x}, \mathbf{y}}^{(c)} = (x^{(p)} - u^{(c)})^T \left( \text{diag}(y^{(s)}) \right)^{-2} (x^{(p)} - u^{(c)}), \quad (31)$$

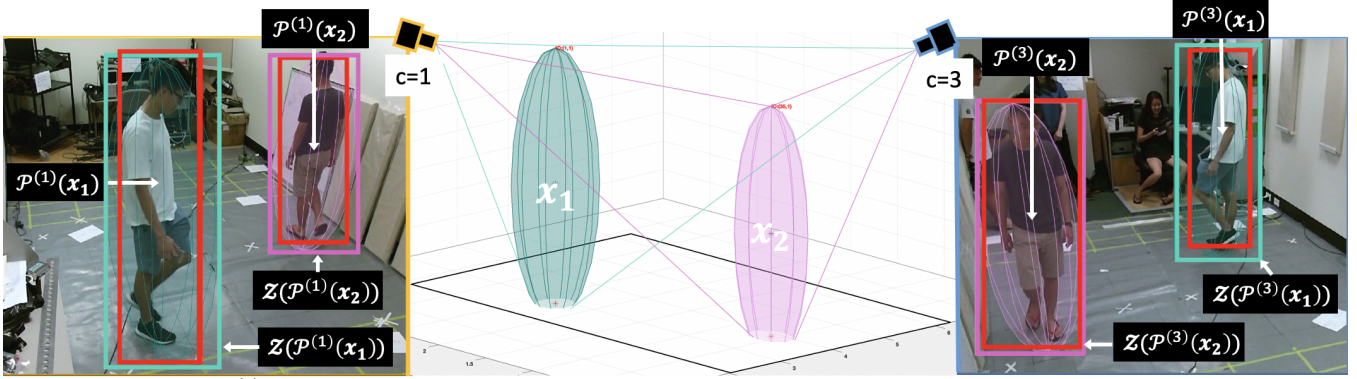


Fig. 5: The projections  $\mathcal{P}^{(c)}$  of two quadrics (in cyan and pink) onto two image views ( $c = 1, 3$ ) result in 2D conics. The transformation  $\mathcal{Z}$  yields the corresponding estimated bounding boxes (in cyan and pink). The estimated bounding box and the measured bounding box (in red) from monocular detector formulate the measurement likelihood (18).

$$\mathcal{B}_{x,y}^{(c)} = (x^{(p)} - u^{(c)})^T \left[ 2 \left( \text{diag}(y^{(s)}) \right)^{-2} u^{(c)} + d_y \right], \quad (32)$$

$$\mathcal{C}_y^{(c)} = (u^{(c)})^T \left[ \left( \text{diag}(y^{(s)}) \right)^{-2} u^{(c)} + d_y \right] + \mathcal{E}_y, \quad (33)$$

$$d_y = -2 \frac{y^{(p)}}{(y^{(s)} \cdot y^{(s)})}, \quad \mathcal{E}_y = \left\| y^{(p)} / y^{(s)} \right\|_2^2 - 1, \quad (34)$$

and  $u^{(c)}$  is the position of camera  $c$ , with multiplication/division of two vectors of the same dimension to be understood as point-wise multiplication/division.

In addition, using quadric projection [58, pp. 201], the relationship between the estimated bounding box  $\Phi^{(c)}(x)$  and measured bounding box  $z^{(c)}$  captured in the measurement likelihood (18), has the following closed form

$$\Phi^{(c)}(x) \triangleq \mathcal{Z}(\mathcal{P}^{(c)}(x)), \quad (35)$$

where

$$\mathcal{P}^{(c)}(x) = \left( P_{3 \times 4}^{(c)} \left[ \begin{array}{c|c} (\text{diag}(x^{(s)}))^{-2} & d_x/2 \\ \hline d_x^T/2 & \mathcal{E}_x \end{array} \right]^{-1} (P_{3 \times 4}^{(c)})^T \right)^{-1}, \quad (36)$$

$$\mathcal{Z} \left( \left[ \begin{array}{c|c} A & r \\ \hline r^T & q \end{array} \right] \right) = \left[ \begin{array}{c} -\mathcal{Q}D^{-1}\mathcal{Q}^T r \\ 2\nu \left\| \begin{bmatrix} 1, 0 \\ 0, 1 \end{bmatrix} \mathcal{Q}D^{-0.5} \right\|_2 \\ 2\nu \left\| \begin{bmatrix} 0, 1 \\ 0, 1 \end{bmatrix} \mathcal{Q}D^{-0.5} \right\|_2 \end{array} \right], \quad (37)$$

$$\nu = (r^T \mathcal{Q}D^{-1} \mathcal{Q}^T r - q)^{0.5}, \quad (38)$$

$\mathcal{Q}$  is a matrix containing the eigenvectors of  $A$ , and  $D$  is a diagonal matrix of the eigenvalues of  $A$ . Given the camera matrices  $P_{3 \times 4}^{(1)}, \dots, P_{3 \times 4}^{(C)}$ ,  $\mathcal{P}^{(c)}(\cdot)$  is a matrix-to-matrix projection that transforms the quadric into a conic on each image of camera  $c$  [58, pp. 201].  $\mathcal{Z}(\cdot)$  is a matrix-to-vector transformation that transforms the conic into a 4D bounding box (in the same format as  $z^{(c)}$ ). The illustration of the overall transformation (35) is depicted in Fig. 5.

The Poisson false alarms intensity for camera  $c$  is  $\kappa^{(c)} \triangleq \lambda_c \mathcal{U}(\cdot)$ , where  $\lambda_c$  is the false-positive (clutter) rate, and  $\mathcal{U}(\cdot)$  is a uniform distribution on the measurement space  $\mathbb{Z}^{(c)}$ . In many visual tracking cases, this value can either be estimated offline or manually tuned. The false alarm intensity can be estimated by the Cardinalized Probability Hypothesis Density (CPHD) clutter estimator [59]. In this work, we bootstrap the CPHD clutter intensity estimator output to the tracker [60].

## 4.2 MV-GLMB-OC Filter Implementation

The number of components of the GLMB filtering density grows super-exponentially over time. To maintain tractability in GLMB filter implementations, truncating insignificant components has been proven to minimize the  $L_1$  approximation error [20]. This truncation strategy can be formulated as an NP-hard multi-dimensional assignment problem [20]. Nonetheless, it can be solved by exploiting certain structural properties, and suitable adaptation of 2D assignment solutions such as Murty's or Auction [20].

The MV-GLMB-OC recursion described in Section 3.6, can be directly implemented with separate prediction and update, i.e. by computing a truncated version of the prediction (19) and the corresponding detection probabilities  $\{P_{D,+}^{(\xi, I_+)}(\ell) : \ell \in I_+, (\xi, I_+) \in \Xi \times \mathcal{F}(\mathbb{L}_+)\}$ , then using this to compute a truncated version of the update (25). This strategy requires keeping a significant portion of the predicted components that would end up as updated components with negligible weights, thereby wasting computations in solving a large number of 2D assignment problems. Thus, this approach is inefficient and becomes infeasible for systems with many sensors [20].

In this work, we exploit an efficient GLMB truncation strategy that has a linear complexity in the sum of the measurements across all sensors [20]. This approach bypasses the prediction truncation, and returns the significant components of the next GLMB filtering density (25) by sampling from a discrete probability distribution proportional to the weights of the components [20]. This means GLMB components with higher weights are more likely to be selected than those with lower weights. For the MV-GLMB-OC recursion, this discrete probability distribution  $s(\cdot; P_{D,+})$  of the GLMB components, is determined by the detection probabilities  $P_{D,+} \triangleq \{P_{D,+}^{(\xi, I_+)}(\ell) : \ell \in I_+, (\xi, I_+) \in \Xi \times \mathcal{F}(\mathbb{L}_+)\}$  (and other multi-object system parameters, which are suppressed for clarity) [20]. However, since truncation of the prediction (19) has been bypassed, the predicted components  $\{(\xi, I_+) \in \Xi \times \mathcal{F}(\mathbb{L}_+)\}$  and their corresponding detection probabilities are not available. Nonetheless, importance sampling can be used to generate weighted samples of  $s(\cdot; P_{D,+})$  by sampling from  $s(\cdot; \hat{P}_{D,+})$ , where  $\hat{P}_{D,+} \triangleq \{P_{D,+}^{(\xi, I \uplus \mathbb{B}_+)}(\ell) : \ell \in I \uplus \mathbb{B}_+, (\xi, I) \in \Xi \times \mathcal{F}(\mathbb{L})\}$ , and then re-weight the resulting samples accordingly [50]. Note that the detection probabilities  $\hat{P}_{D,+}$  can be readily computed



from the components of the (truncated) current GLMB filtering density  $\{(w^{(I,\xi)}, p^{(\xi)}) : (I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi\}$ . Moreover,  $P_{D,+}^{(\xi, I \uplus \mathbb{B}_+)} \succeq P_{D,+}^{(\xi, I_+)}$ , for any  $I_+ \subseteq I \uplus \mathbb{B}_+$ , it follows from [61] that  $s(\cdot; \hat{P}_{D,+})$  is more diffused than  $s(\cdot; P_{D,+})$ , i.e. the support of  $s(\cdot; \hat{P}_{D,+})$  contains the support of  $s(\cdot; P_{D,+})$ .

The MS-GLMB and MV-GLMB-OC recursions are presented in Algorithm 1 and 2 respectively. Observe that the main difference is the additional computation of the detection probabilities prior to and re-weighting after the Gibbs sampling step in the MV-GLMB-OC filter.

In this work, the object's birth density  $f_{B,+}(\cdot, \ell)$ , single-object transition (26) and likelihood (18) are all Gaussians. Standard Kalman prediction and Unscented Kalman update are used to evaluate the single-object filtering density  $p_+^{(\xi_+)}$ , which results in a Gaussian.

---

#### Algorithm 1 MS-GLMB Filter [20]

---

**Global Input:**  $\{(P_{B,+}(\ell), f_{B,+}(\cdot, \ell))\}_{\ell \in \mathbb{B}_+}, \mathbf{f}_{S,+}(\cdot|\cdot), P_S(\cdot)$   
**Global Input:**  $\kappa, P_D, g$   
**Input:**  $\pi \triangleq \{(w^{(I,\xi)}, p^{(\xi)}) : (I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi\}$   
**Output:**  $\pi_+ \triangleq \{(w_+^{(I_+,\xi_+)}, p_+^{(\xi_+)}) : (I_+, \xi_+) \in \mathcal{F}(\mathbb{L}_+) \times \Xi_+\}$

---

**for**  $(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi$   
     Construct stationary distribution from inputs  
     Run Gibbs sampler to obtain samples  $\gamma_+$  [20, Algorithm 3]  
     Use samples  $\gamma_+$  to compute  $\pi_+$   
**end for**  
 Extract labeled state estimates

---



---

#### Algorithm 2 MV-GLMB-OC Filter

---

**Global Input:**  $\{(P_{B,+}(\ell), f_{B,+}(\cdot, \ell))\}_{\ell \in \mathbb{B}_+}, \mathbf{f}_{S,+}(\cdot|\cdot), P_S(\cdot)$   
**Global Input:**  $\kappa, P_D, g$   
**Input:**  $\pi \triangleq \{(w^{(I,\xi)}, p^{(\xi)}) : (I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi\}$   
**Output:**  $\pi_+ \triangleq \{(w_+^{(I_+,\xi_+)}, p_+^{(\xi_+)}) : (I_+, \xi_+) \in \mathcal{F}(\mathbb{L}_+) \times \Xi_+\}$

---

**for**  $(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi$   
     Compute occlusion-based probability of detection  
      $\{P_{D,+}^{(\xi, I \uplus \mathbb{B}_+)}(\ell) : \ell \in I \uplus \mathbb{B}_+\}$  via (24)  
     Construct stationary distribution from inputs and  
      $\{P_{D,+}^{(\xi, I \uplus \mathbb{B}_+)}(\ell) : \ell \in I \uplus \mathbb{B}_+\}$   
     Run Gibbs sampler to obtain samples  $\gamma_+$  [20, Algorithm 3]  
     Update occlusion-based probability of detection  
      $\{P_{D,+}^{(\xi, \mathcal{L}(\gamma_+))}(\ell) : \ell \in \mathcal{L}(\gamma_+)\}$ , via (24)  
     Use samples  $\gamma_+$ ,  $\{P_{D,+}^{(\xi, \mathcal{L}(\gamma_+))}(\ell) : \ell \in \mathcal{L}(\gamma_+)\}$  to compute  $\pi_+$   
**end for**  
 Extract labeled state estimates

---

## 5 EXPERIMENTS

This section demonstrates the three main advantages of the proposed MV-GLMB-OC approach. The first is the capability to produce 3D object trajectories using independent monocular detections from multiple views, where each object is represented as a 3D ellipsoid of unknown location and extent (Section 5.2). The second is the amenability for uninterrupted/seamless operation in the event that cameras are added, removed or repositioned on the fly (Section 5.3). The third is the flexibility of not confining objects to the

ground plane, which is demonstrated by tracking people jumping and falling (Section 5.4). The effectiveness of the proposed occlusion model is also studied, by comparing the tracking performance of the MV-GLMB-OC against that of the standard MS-GLMB filter.

We first focus our demonstrations on the latest WILDTRACKS dataset<sup>2</sup>, which involves seven-cameras at 1920×1080 resolution with overlapping views. The WILDTRACKS dataset is also supplied with calibrated intrinsic and extrinsic camera parameters, along with 3D ground plane annotations although these are restricted to the ground plane. WILDTRACKS was initially introduced to address various perceived shortcomings in older multi-view datasets, the arguments for which were originally presented in [17] and are summarized as follows. The DukeMTMC dataset [47] is essentially non-overlapping in views and is now no longer available. The PETS 2009 S2.L1 dataset [48] has supposed inconsistencies when projecting 3D points across the views. The EPFL, SALSA and Campus datasets [3], [15], [49] involve a relatively small number of people, and are relatively sparse in terms of person density, but do not provide 3D annotations. In addition, the EPFL-RLC dataset [16] only provides annotations for a small subset of the last 300 of 8000 frames. For the same reasons that the authors of WILDTRACKS were motivated to introduce their new dataset, the older multi-view datasets superseded by WILDTRACKS are not suitable for evaluating the MV-GLMB-OC filter in the 3D world frame.

In the context of demonstrating the MV-GLMB-OC approach however, the WILDTRACKS dataset is not suitable for evaluating tracking performance in full 3D, i.e. changes in all 3 x, y, z-coordinates. While WILDTRACKS provides 3D annotations, these are restricted to the ground plane. Moreover the annotations are for centroids only, and do not capture the extent (in terms of length, width and height) of objects in the world coordinates. In our performance comparisons, the outputs of the proposed MV-GLMB-OC filter on WILDTRACKS are limited to the estimated centroids projected onto the ground plane. To demonstrate the full capabilities of MV-GLMB-OC, it is critical to have annotations of the 3D centroids and their 3D extent, along with the ground truths for each of the camera locations. Consequently we introduce a new Curtin Multi-Camera (CMC) dataset which meets these requirements.

The new CMC dataset is a four-camera 1920x1024 resolution dataset recorded at 4fps in a room with dimensions 7.67m x 3.41m x 2.7m. The CMC dataset has 5 different sequences with varying levels of person density and occlusion: CMC1 has a maximum of 3 people and virtually no occlusion; CMC2 has a maximum of 10 people with some occlusion; CMC3 has a maximum of 15 people with significant occlusion; while CMC4 and CMC5 involve people jumping and falling with a maximum of 3 and 7 people respectively. CMC1 and CMC4 have low person density and are intended for basic testing, while CMC2, CMC3 and CMC5 have higher person density and significant visual occlusions across multiple overlapping cameras, and are intended to highlight performance differences. The convention for the world coordinate frame is illustrated in Fig. 6. The origin is

2. <https://www.epfl.ch/labs/cvlab/data/data-wildtrack/>

at the lower corner and the ground plane corresponds to the x-y plane i.e.  $z = 0$ . In every sequence, each person enters the tracking area at (2.03m, 0.71m) with an average height of 1.7m. The dataset is also supplied with camera locations and parameters, along with annotations for 3D centroid and extent. The 2D monocular annotation for bounding boxes is carried out with the MATLAB Image Labeler Tool, and the world coordinates are obtained by averaging the homographic projection of the feet coordinates from each view. The actual height and width of each person is used for the annotation.

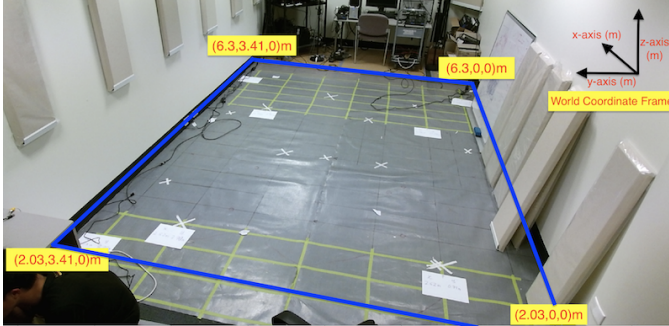


Fig. 6: Layout for CMC dataset: The blue line denotes the boundary of the tracking area. The yellow boxes denote the coordinates of the boundary in (x,y,z) axes. The 4 cameras are positioned (in sequence) at the top 4 corners of the room.

A common setting for object survival and detection model parameters is used in both evaluations on the WILDTRACKS and CMC datasets. Specifically: the survival probability  $P_S(x)$  given by (29), is parameterized by the control parameter  $\tau = 0.5$  and the scene mask  $b(\cdot)$  with a margin of 0.3m inside the border of the tracking area; the detection probability, given in Section 3.5 is parameterized by  $P_D^{(c)}(x) = 0.9$  and  $\beta = 0.1$ . For all cameras, the observed bounding box model is described in (18), with position noise parameterized by  $v_p^{(c)} = [400, 400]^T$ , and the extent noise parameterized by  $v_e^{(c)} = [0.01, 0.0025]^T$  (on the logarithms of the half-lengths of the principal axes).

## 5.1 Performance Evaluation Criteria

### 5.1.1 Standard Evaluation on 3D Position Estimates

The performance of various combinations of detectors and trackers are evaluated using the CLEAR MOT devkit provided in [62]. For computing CLEAR MOT, we adhere to the convention of using the Euclidean distance ( $L_2$ -norm) on the estimated 3D centroid with a threshold of 1m.

For MOT, the following performance indicators are reported: Multiple Object Tracking Accuracy (MOTA) which penalizes normalized false negatives (FNs), false positives (FPs) and identity switches (IDs) between consecutive frames; Multiple Object Tracking Precision (MOTP) which accounts for the overall dissimilarity between all true positives and the corresponding ground truth objects [63]; Mostly Tracked (MT), Partially Tracked (PT), Mostly Lost (ML) which indicate how much of the trajectory is retained or lost by the tracker; Fragmentations (FM) which account for interrupted tracks based on ground truth trajectories; Identity Precision (IDP), Identity Recall (IDR) and  $F_1$  score (IDF1) which are analogous to the standard *precision*, standard *recall* and  $F_1$  score with identifications (tracks) [47].

For reference, we also provide performance indicators on the bounding box detections, where we set the threshold at 0.5 and report: Multiple Object Detection Accuracy (MODA) which accounts for misdetections and false alarms; Multiple Object Detection Precision (MODP) which accounts for the spatial overlap information between the bounding boxes; *precision* which is the measure of exactness; and *recall* which is the measure of quality.

We note that CLEAR MOT is traditionally calculated over the entire scenario window, and thus the tracking performance is reported after the entire data stream has been processed. To evaluate the live or online tracking performance over time, we employ the Optimal Sub-Pattern Assignment (OSPA<sup>(2)</sup>) distance between two sets of tracks [21]. This distance is based on the OSPA metric that captures both localization and cardinality errors between two finite sets of a metric space with a suitable base-distance between objects (e.g. the Euclidean distance) [64]. The OSPA<sup>(2)</sup> metric is defined as the OSPA distance between two sets of tracks over a time window. Details for OSPA and OSPA<sup>(2)</sup> metrics are given in Appendix 7.2. By design, OSPA<sup>(2)</sup> captures both localization and cardinality errors between the set of true and estimated tracks, and penalizes switched tracks or label changes [21]. The resultant metric carries the interpretation of a time-averaged per-track error. In our evaluation of the position estimate in real world coordinates, we use a 3D Euclidean base-distance for OSPA<sup>(2)</sup> with order parameter 1 and cutoff parameter 1m. Performance evaluation for live or online tracking is given by plotting the error over a sliding window of length  $L_w = 10$  frames, while overall performance is captured in a single number by calculating the error over the entire scenario window.

### 5.1.2 GloU Based Evaluation on 3D Position with Extent

As the proposed MV-GLMB-OC filter outputs 3D estimates of the object centroid and extent, we extend the performance evaluations to capture the joint error in the centroid and extent. This is achieved by employing an alternative base-distance between two objects, in this case a 3D generalized intersection over union (GloU), which extends the commonly used IoU to non-overlapping bounding boxes [65]. The details for the IoU and GloU metrics are given in Appendix 7.3. It is important to note that if there is no overlap between the ground truth and estimated shape, the IoU distance is zero regardless of their separation, whereas the GloU distance captures the extent of the error while retaining the metric property [65]. We present evaluations of the estimated centroid with extent for CLEAR MOT (using a GloU base-distance with a threshold of 0.5) and OSPA<sup>(2)</sup> metric with GloU base-distance (and with unit order and cut-off parameters). We refer the reader to [66] for the rationale and discussions on the use of OSPA<sup>(2)</sup>-GloU for performance evaluation.

## 5.2 WILDTRACKS Dataset

We test MV-GLMB-OC against the latest multi-camera detector (Deep-Occlusion) [18] coupled with the  $k$ -shortest-path (KSP) algorithm [5] and *ptrack* as shown in [17] (Deep-Occlusion+KSP+ptack). KSP is an optimization algorithm

TABLE 2: WILDTRACKS Performance Benchmarks for 3D Position Estimates (restricted to the ground plane)

Detector and Tracker	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$	MT $\uparrow$	PT $\downarrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$	FM $\downarrow$	MOTA $\uparrow$	MOTP $\uparrow$	OSPA <sup>(2)</sup> $\downarrow$
YOLOv3+MV-GLMB-OC	74.3%	<b>85.0%</b>	75.9%	<b>136</b>	111	37	<b>424</b>	1333	104	86	69.7%	<b>73.2%</b>	<b>0.69m</b>
YOLOv3+MS-GLMB	74.2%	79.0%	69.9%	116	85	83	841	1951	139	105	61.9%	68.3%	0.81m
Faster-RCNN(VGG16)+MV-GLMB-OC	76.5%	84.5%	70.0%	119	118	47	545	1621	104	81	65.3%	71.9%	0.72m
Faster-RCNN(VGG16)+MS-GLMB	75.5%	76.8%	74.3%	98	104	82	1114	1716	179	116	61.5%	65.8%	0.88m
Deep-Occlusion+GLMB	72.5%	82.7%	72.2%	160	86	39	960	<b>990</b>	107	<b>64</b>	70.1%	63.1%	0.73m
Deep-Occlusion+KSP+ptrack	<b>78.4%</b>	84.4%	73.1%	72	<b>74</b>	<b>25</b>	2007	5830	<b>103</b>	95	<b>72.2%</b>	60.3%	0.75m

CLEAR MOT scores and OSPA<sup>(2)</sup> distance are calculated on standard position estimates ( $\uparrow$  means higher is better while  $\downarrow$  means lower is better). Three different detectors are considered -Deep-Occlusion (multiocular), Faster-RCNN(VGG16) (monocular) and YOLOv3 (monocular). Three types of trackers are considered -KSP+ptrack or GLMB (single-sensor), MV-GLMB-OC (multi-view with occlusion model) and MS-GLMB (multi-sensor without occlusion model).

that finds the most likely sequence of ground plane occupancies (trajectories) given by the multi-camera detector, and *ptrack* described in [67] improves and smooths over tracks by learning motion patterns. As a baseline comparison, we employ the Deep-Occlusion multi-camera detector combined with single-view GLMB (Deep-Occlusion+GLMB). Since WILDTRACKS provides annotations in real-world coordinates but restricted to the ground plane, tracking is performed in real-world coordinates but also restricted to the ground plane. To further explore the performance of MV-GLMB-OC, we also run experiments using monocular detections from each of the cameras. For the detectors, we use the monocular backbone of the Deep-Occlusion detector i.e. VGG16-net trained using Faster-RCNN [23], and separately with the newer YOLOv3 [68], to produce separate monocular detections for input to MV-GLMB-OC. Since WILDTRACKS does not supply the camera positions required for our proposed occlusion model, we reconstruct the camera positions from the given camera parameters. We note that KSP and/or ptrack is an offline or batch method, while GLMB is online or recursive, and provides estimates on the fly.

### 5.2.1 Model Parameters

The birth density is adaptive/measurement-driven (see Section F in [69]) with  $P_{B,+}(\ell) = 0.001$  and  $f_{B,+}(x, \ell) = \mathcal{N}(x; \mu_{B,+}^{(\ell)}, 0.1^2 I_9)$  where  $\mu_{B,+}^{(\ell)}$  is obtained via clustering (e.g. *k*-means). The single-object transition is as described in (26) with position noise and extent (in logarithm) noise parameterized by:

$$v^{(p)} = [0.0016, 0.0016, 0.0016]^T,$$

$$v^{(s)} = [0.0036, 0.0036, 0.0004]^T.$$

### 5.2.2 Discussion

Table 2 shows the CLEAR MOT and OSPA<sup>(2)</sup> benchmarks for MV-GLMB-OC (with occlusion modeling) and MS-GLMB (without occlusion modeling) with two different detectors YOLOv3 and Faster-RCNN(VGG16). Results for Deep-Occlusion+KSP+ptrack being the reference, are reproduced directly from the original paper [17]. The results indicate that the two trackers based on multi-camera detections, i.e. Deep-Occlusion+KSP+ptrack and Deep-Occlusion+GLMB, have very similar tracking performance in terms of MOTA/MOTP and OSPA<sup>(2)</sup>. Importantly, closer examination of the tracking results based on multiple monocular detections indicates that performance is significantly improved with the addition of the occlusion model. This can be seen from the relative changes in the MOTA/MOTP and OSPA<sup>(2)</sup>. Several observations can also be drawn from comparing the multi-camera detector with batch processing method (Deep-Occlusion+KSP+ptrack), and the related monocular detector

with online processing (Faster-RCNN(VGG16)+MV-GLMB-OC). While the MOTP improves due to the use of multiple monocular detectors, the MOTA degrades due to the use of an online method which is unable to correct past estimates. This is corroborated by the overall OSPA<sup>(2)</sup> value which improves slightly from Deep-Occlusion+KSP+ptrack to Faster-RCNN(VGG16)+MV-GLMB-OC. Surprisingly, the results based on YOLOv3 are better across the board than that for Faster-RCNN(VGG16), even though YOLOv3 is more efficient than Faster-RCNN(VGG16). For reference, the CLEAR evaluations for the detectors used in the experiment are presented in Appendix 7.4, from which it is noted that the monocular detections are generally much poorer than the multi-camera detections due to severe occlusions.

## 5.3 CMC1, CMC2 and CMC3

This subsection focuses on scenarios with people walking in order of increasing difficulty, i.e. CMC1-CMC3. Similar to the WILDTRACKS evaluation, we evaluate our method based on 2 monocular detectors, namely Faster-RCNN(VGG16) and YOLOv3. For each sequence, the effect of the occlusion model is studied by comparing the proposed MV-GLMB-OC with the standard MS-GLMB filter.

### 5.3.1 Model Parameters

Unlike WILDTRACKS where objects enter the scene from anywhere at the boundary, in CMC we know the location of objects entering the scene. Hence, we specify the birth parameters as  $P_{B,+}(\ell) = 0.001$  and  $f_{B,+}(x, \ell) = \mathcal{N}(x; \mu_{B,+}, 0.1^2 I_9)$  where

$$\mu_{B,+} = [2.03 \ 0 \ 0.71 \ 0 \ 0.825 \ 0 \ -1.2 \ -1.2 \ -0.18]^T.$$

We use the single-object transition density (26) with position noise and extent (in logarithm) noise parameterized by:

$$v^{(p)} = [0.0012, 0.0012, 0.0012]^T,$$

$$v^{(s)} = [0.0036, 0.0036, 0.0004]^T.$$

### 5.3.2 Effectiveness of Occlusion Model

Table 3 shows the CLEAR MOT and OSPA<sup>(2)</sup> benchmarks with a Euclidean base-distance, for the estimated 3D centroids only. Table 4 shows the CLEAR MOT and OSPA<sup>(2)</sup> benchmarks with a 3D GIoU base-distance, for the estimated 3D centroids and extent. Both tables compare the tracking performance with and without an occlusion model, i.e. MV-GLMB-OC and MS-GLMB respectively. The asterisk entry denotes the multi-camera reconfiguration case which is discussed later on. All results are presented for two different detectors YOLOv3 and Faster-RCNN(VGG16).

We focus our initial examination on the non-asterisked entries in Tables 3 and 4. This corresponds to the case where

TABLE 3: CMC1,2,3 Performance Benchmarks for 3D Position Estimates

CMC1 (Maximum/Average 3 people)													
Detector and Tracker	IDF1 ↑	IDP ↑	IDR ↑	MT ↑	PT ↓	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	OSPA <sup>(2)</sup> ↓
YOLOv3+MV-GLMB-OC	99.7%	99.4%	100%	3	0	0	4	0	0	0	99.4%	91.8%	0.13m
YOLOv3+MV-GLMB-OC*	98.9%	97.9%	99.8%	3	0	0	14	1	0	0	97.7%	90.5%	0.16m
YOLOv3+MS-GLMB	95.9%	92.3%	99.8%	3	0	0	55	1	1	0	91.3%	91.4%	0.34m
Faster-RCNN(VGG16)+MV-GLMB-OC	99.5%	99.1%	100%	3	0	0	6	0	0	0	99.1%	91.8%	0.13m
Faster-RCNN(VGG16)+MV-GLMB-OC*	95.5%	91.4%	100%	3	0	0	62	0	1	0	90.4%	90.5%	0.14m
Faster-RCNN(VGG16)+MS-GLMB	99.6%	99.2%	100%	3	0	0	5	0	0	0	99.2%	91.4%	0.36m
CMC2 (Maximum/Average 10 people)													
Detector and Tracker	IDF1 ↑	IDP ↑	IDR ↑	MT ↑	PT ↓	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	OSPA <sup>(2)</sup> ↓
YOLOv3+MV-GLMB-OC	91.0%	91.1%	91.3%	10	0	0	16	11	9	2	98.3%	81.7%	0.30m
YOLOv3+MV-GLMB-OC*	90.1%	90.2%	90.0%	10	0	0	38	29	11	7	96.2%	78.9%	0.34m
YOLOv3+MS-GLMB	67.7%	79.9%	58.9%	4	6	0	8	550	34	30	71.5%	74.4%	0.70m
Faster-RCNN(VGG16)+MV-GLMB-OC	90.6%	90.5%	90.9%	10	0	0	50	37	9	5	95.4%	83.7%	0.35m
Faster-RCNN(VGG16)+MV-GLMB-OC*	86.2%	85.5%	87.5%	10	0	0	120	60	25	13	90.1%	79.8%	0.48m
Faster-RCNN(VGG16)+MS-GLMB	75.3%	81.9%	69.7%	7	3	0	7	316	23	19	83.3%	80.4%	0.58m
CMC3 (Maximum/Average 15 people)													
Detector and Tracker	IDF1 ↑	IDP ↑	IDR ↑	MT ↑	PT ↓	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	OSPA <sup>(2)</sup> ↓
YOLOv3+MV-GLMB-OC	77.9%	79.7%	76.1%	13	2	0	63	191	44	33	89.5%	76.4%	0.51m
YOLOv3+MV-GLMB-OC*	72.1%	77.9%	67.2%	11	4	0	47	437	51	37	81.1%	72.3%	0.61m
YOLOv3+MS-GLMB	50.5%	69.9%	39.5%	0	15	0	5	1234	54	51	54.2%	67.8%	0.83m
Faster-RCNN(VGG16)+MV-GLMB-OC	71.7%	74.9%	68.8%	12	3	0	71	303	44	32	85.2%	73.5%	0.61m
Faster-RCNN(VGG16)+MV-GLMB-OC*	67.7%	72.1%	63.8%	10	5	0	92	419	59	44	79.8%	68.0%	0.70m
Faster-RCNN(VGG16)+MS-GLMB	54.3%	73.2%	43.1%	0	15	0	3	1165	53	55	56.8%	65.9%	0.81m

CLEAR MOT scores and OSPA<sup>(2)</sup> distance are calculated on standard position estimates (↑ means higher is better while ↓ means lower is better). Two different detectors are considered - Faster-RCNN(VGG16) (monocular) and YOLOv3 (monocular). Two types of trackers are considered - MV-GLMB-OC (multi-view with occlusion model) and MS-GLMB (multi-sensor without occlusion model). The asterisk (\*) indicates the multi-camera reconfiguration experiment.

TABLE 4: CMC1,2,3 Performance Benchmarks for 3D Centroid with Extent Estimates

CMC1 (Maximum/Average 3 people)													
Detector and Tracker	IDF1 ↑	IDP ↑	IDR ↑	MT ↑	PT ↓	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	OSPA <sup>(2)</sup> ↓
YOLOv3+MV-GLMB-OC	99.7%	99.4%	100%	3	0	0	4	0	0	0	99.4%	67.8%	0.20
YOLOv3+MV-GLMB-OC*	98.9%	97.9%	99.8%	3	0	0	14	1	0	0	97.7%	66.7%	0.20
YOLOv3+MS-GLMB	95.9%	92.3%	99.8%	3	0	0	55	1	1	0	91.3%	67.5%	0.40
Faster-RCNN(VGG16)+MV-GLMB-OC	99.5%	99.1%	100%	3	0	0	6	0	0	0	99.1%	67.5%	0.20
Faster-RCNN(VGG16)+MV-GLMB-OC*	95.5%	91.4%	100%	3	0	0	62	0	1	0	90.4%	67.2%	0.20
Faster-RCNN(VGG16)+MS-GLMB	99.6%	99.2%	100%	3	0	0	5	0	0	0	99.2%	66.9%	0.40
CMC2 (Maximum/Average 10 people)													
Detector and Tracker	IDF1 ↑	IDP ↑	IDR ↑	MT ↑	PT ↓	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	OSPA <sup>(2)</sup> ↓
YOLOv3+MV-GLMB-OC	87.9%	87.5%	87.7%	10	0	0	19	14	8	2	98.0%	62.3%	0.32
YOLOv3+MV-GLMB-OC*	87.3%	87.1%	87.5%	10	0	0	53	44	14	12	94.7%	57.0%	0.38
YOLOv3+MS-GLMB	59.4%	69.9%	51.7%	4	6	0	21	563	30	31	70.4%	55.7%	0.62
Faster-RCNN(VGG16)+MV-GLMB-OC	86.7%	86.5%	87.0%	10	0	0	68	55	10	8	93.6%	60.9%	0.34
Faster-RCNN(VGG16)+MV-GLMB-OC*	81.3%	80.2%	82.5%	10	0	0	127	67	33	15	89.1%	55.0%	0.45
Faster-RCNN(VGG16)+MS-GLMB	68.6%	74.6%	63.5%	7	3	0	23	332	23	21	81.8%	57.1%	0.52
CMC3 (Maximum/Average 15 people)													
Detector and Tracker	IDF1 ↑	IDP ↑	IDR ↑	MT ↑	PT ↓	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	OSPA <sup>(2)</sup> ↓
YOLOv3+MV-GLMB-OC	70.7%	72.3%	69.1%	14	1	0	94	222	45	37	87.2%	52.8%	0.53
YOLOv3+MV-GLMB-OC*	60.8%	65.7%	56.6%	9	6	0	91	481	66	56	77.4%	46.4%	0.60
YOLOv3+MS-GLMB	41.4%	57.3%	32.4%	0	15	0	10	1239	64	60	53.5%	46.7%	0.76
Faster-RCNN(VGG16)+MV-GLMB-OC	63.7%	66.6%	61.1%	12	3	0	97	329	63	41	82.7%	52.8%	0.58
Faster-RCNN(VGG16)+MV-GLMB-OC*	57.3%	61.0%	54.0%	10	5	0	133	460	78	60	76.3%	47.9%	0.66
Faster-RCNN(VGG16)+MS-GLMB	45.7%	61.7%	36.3%	0	15	0	13	1175	61	67	55.8%	46.6%	0.75

CLEAR MOT scores and OSPA<sup>(2)</sup> distance are calculated with a 3D GloU base-distance for estimates of 3D centroid with extent (↑ means higher is better while ↓ means lower is better). Two different detectors are considered - Faster-RCNN(VGG16) (monocular) and YOLOv3 (monocular). Two types of trackers are considered - MV-GLMB-OC (multi-view with occlusion model) and MS-GLMB (multi-sensor without occlusion model). The asterisk (\*) indicates the multi-camera reconfiguration experiment.

all cameras are operational. For the sparse scenario CMC1, both MV-GLMB-OC and MS-GLMB on either detectors achieved a close to perfect CLEAR MOT scores in MOTA and MOTP. Some of the flagged FPs are caused by track initiation/termination mismatches with the ground truths (annotations). The OSPA<sup>(2)</sup> values are relatively low due to the sparsity of the scenario.

For the medium scenario CMC2, Fig. 7 shows a screenshot of the detections and the MV-GLMB-OC estimates. In this case, MV-GLMB-OC on both detectors managed to maintain consistent tracks and accurate estimates overall. The CLEAR MOT benchmarks for CMC2 show high MOTA and MOTP but with some FNs and FPs. We observe an

improvement in performance for MV-GLMB-OC over MS-GLMB, and on both detectors due to the inclusion of occlusion modeling. The improvement in performance due to occlusion modeling is also reflected in the OSPA<sup>(2)</sup>.

For the dense scenario CMC3, MV-GLMB-OC on both detectors managed to achieve acceptable MOTA/MOTP scores, but is penalized with high FPs, FNs, IDs and FMs. This outcome occurs even with the proposed occlusion model, as the algorithm fails when a person is totally occluded in all views. An example of this occurrence is illustrated in Fig. 8, where the red bounding boxes denote detections, while the yellow bounding boxes indicate people who are undetected in all views. Such an event could



Fig. 7: CMC2 Camera 1 to 4 (left to right): YOLOv3 detections (top row) and MV-GLMB-OC estimates (bottom row).

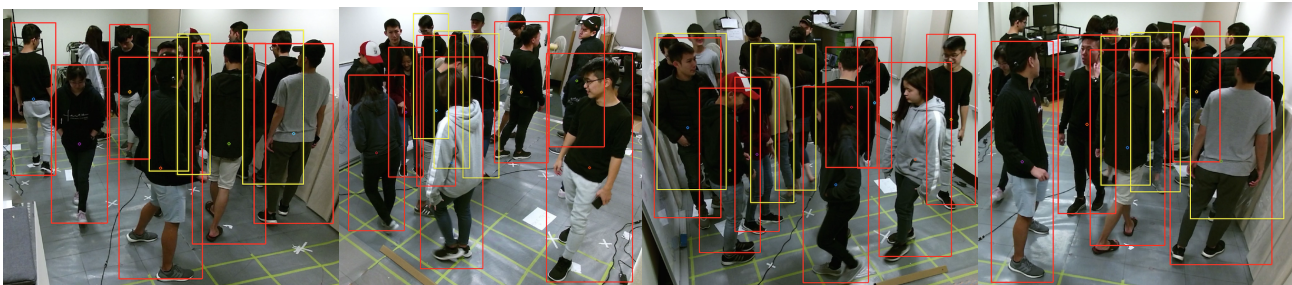


Fig. 8: CMC3 Camera 1 to 4 (left to right): YOLOv3 detections (red bounding boxes) and people that are occluded in all four cameras (yellow bounding boxes).

cause track termination/switching and is reflected in the performance evaluation. It is evident from Tables 3 and 4 that the tracking performance improves considerably with the occlusion model. Examination of the OSPA<sup>(2)</sup> error leads to a similar conclusion.

Overall, YOLOv3+MV-GLMB-OC performs slightly better than Faster-RCNN(VGG16)+MV-GLMB-OC due to better detections. The tracking performance of the proposed MV-GLMB-OC filter generally degrades as the number of people in the scene increases, since the visual occlusions become more frequent and more difficult to resolve. The results of this study on the proposed occlusion model suggest that without proper modeling of the probability of detection, the algorithm fails to maintain tracks, resulting in poorer tracking results. The CLEAR evaluation for the monocular detectors used are given in Appendix 7.4.

### 5.3.3 Multi-Camera Reconfiguration

The MV-GLMB-OC approach requires only a one-off training on each monocular detector, and hence can operate without retraining and without interruption, in the event that cameras are added, removed or repositioned on the fly. To demonstrate this capability, we design a multi-camera reconfiguration experiment. At the start of the sequence, all four cameras are operational. Later, one camera is taken offline to mimic a camera failure. Subsequently, two cameras are taken offline to mimic a more severe camera failure. After this, the two previously offline cameras are made operational, while the previously operational cameras are taken offline, which mimics the event that the two operational cameras are moved to different locations. We benchmark the multi-camera reconfiguration experiment against the ideal case when all cameras are operational.

Results for the experiments on multi-camera reconfiguration are denoted with an asterisk in Tables 3 and 4. The reported CLEAR MOT scores and OSPA<sup>(2)</sup> errors show

similar trends in respect of inclusion of the occlusion model, increasing scenario density, and relative performance on the two detectors. The tracking performance in the multi-camera reconfiguration case is generally worse than the case when all cameras are active. This relative observation is in line with expectations, as there is less sensor data to resolve occlusions and perform estimation.

To facilitate an examination of the relative performance in further detail, Fig. 9 plots the OSPA<sup>(2)</sup> error with 3D Giou base-distance over a sliding window with time. As a reference point for the performance comparison, the YOLOv3+MV-GLMB-OC with all cameras operational case is also shown. The spikes in the error curve at the beginning and the end of the scenario are due to mismatches in track initiation and termination with the ground truths. For CMC1, we observe that the error curves are relatively close to the reference case. This would be expected for a sparse scenario as there are virtually no occlusions even when some cameras are offline. For CMC2 and CMC3, the error curves for both YOLOv3+MV-GLMB-OC\* and Faster-RCNN(VGG16)+MV-GLMB-OC\* begin to deviate midway into sequence from the all cameras operational reference. The errors become more pronounced entering the 2-camera only segment, as the more crowded scenarios exacerbate the effect of occlusions and misdetections. Nonetheless, the results show that the MV-GLMB-OC filter is able to accommodate on-the-fly changes to the camera configurations.

## 5.4 CMC4 and CMC5 (3D Multi-Modal Tracking)

Here we present the first multi-camera dataset with people jumping and falling, which is more challenging for MOT than scenarios with only normal walking. We demonstrate the versatility of the proposed MOT framework by using a Jump Markov System (JMS), to cater for potential switching between upright and fallen modes [70].

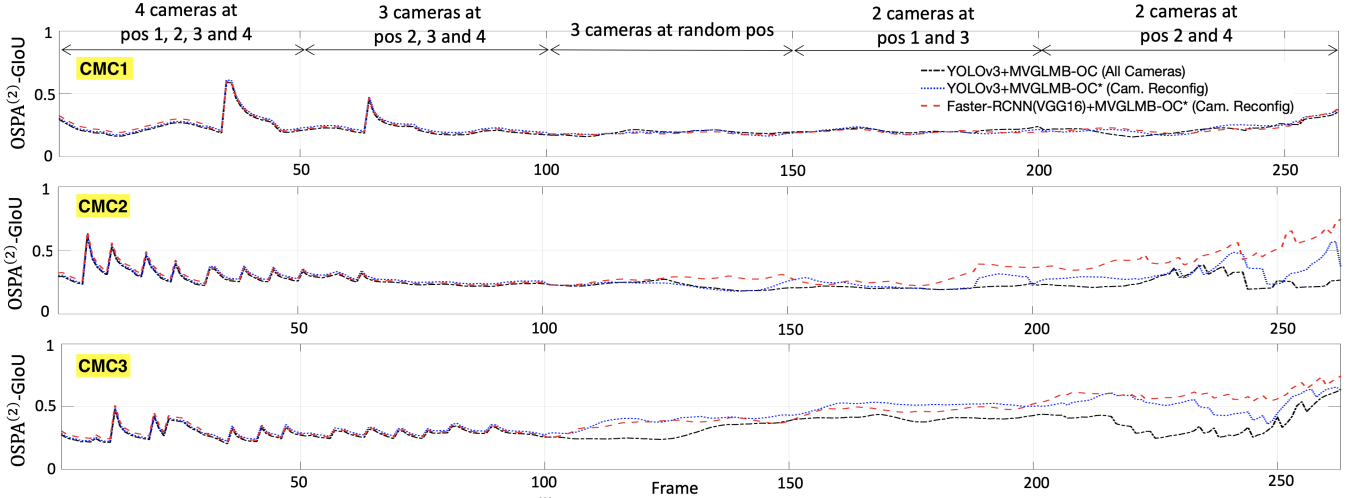


Fig. 9: Multi-Camera Reconfiguration Experiment: OSPA<sup>(2)</sup> plots with 3D GIoU base-distance for estimates of 3D centroid with extent. Three trackers are considered: YOLOv3+MV-GLMB-OC\* (multi-camera reconfiguration) and Faster-RCNN+MV-GLMB-OC\* (multi-camera reconfiguration) and with YOLOv3+MV-GLMB-OC (all cameras operational).

#### 5.4.1 Model Parameters

Each state is augmented  $\mathbf{x}$  with a discrete mode or class  $m \in \{0, 1\}$ , where  $m = 0$  corresponds to a standing state and  $m = 1$  corresponds to a fallen state. We consider the single-object state as  $(\mathbf{x}, m)$ , with single-object density  $p^{(\xi)}(\mathbf{x}, m) = p^{(\xi)}(\mathbf{x}|m)\mu^{(\xi)}(m)$ . The following single-object transition density and observation likelihood are used

$$f_{S,+}(\mathbf{x}_+m_+|\mathbf{x}, m) = f_{S,+}^{(m_+)}(x_+|x, \ell, m)\delta_\ell[l_+]\mu_+(m_+|m),$$

$$g^{(c)}(z^{(c)}|\mathbf{x}, m) \propto g_e^{(c)}(z_e^{(c)}|m) \times \mathcal{N}\left(z^{(c)}; \Phi^{(c)}(\mathbf{x}) + \begin{bmatrix} 0_{2 \times 1} \\ -v_e^{(c,m)}/2 \end{bmatrix}, \text{diag}\left(\begin{bmatrix} v_p^{(c)} \\ v_e^{(c,m)} \end{bmatrix}\right)\right).$$

The mode transition probabilities are  $\mu_+(0|0) = 0.6$ ,  $\mu_+(1|0) = 0.4$ ,  $\mu_+(0|1) = 0.6$  and  $\mu_+(1|1) = 0.4$ .

For a standing object, i.e.  $m = 0$ , we have  $v_e^{(c,0)} = v_e^{(c)} = [0.01, 0.0025]^T$  in the above observation likelihood. Further, standing objects typically have a bounding box size ratio (y-axis/x-axis) greater than one, thus the mode dependent likelihood component is chosen as  $g_e^{(c)}(z_e^{(c)}|0) = e^{-\rho\left(\frac{((1,1)z_e^{(c)})}{(1,0)z_e^{(c)}} - 1\right)}$  for all cameras, where  $\rho = 2$  is a control parameter. The transition density to another standing state  $f_{S,+}^{(0)}(x_+|x, \ell, 0)$ , is the same as per the previous subsection.

For a fallen object, i.e.  $m = 1$ , we have  $v_e^{(c,1)} = [0.0025, 0.01]^T$  in the above observation likelihood, and the mode dependent likelihood component is chosen as  $g_e^{(c)}(z_e^{(c)}|1) = e^{-\rho\left(\frac{((1,1)z_e^{(c)})}{(1,0)z_e^{(c)}} - 1\right)}$  for all cameras because fallen objects typically have a bounding box size ratio (y-axis/x-axis) less than one. The transition density to another fallen state  $f_{S,+}^{(1)}(x_+|x, \ell, 1)$  is the same as that for standing-to-standing except for the large variance  $v^{(s)} = [0.15, 0.15, 0.04]^T$  to capture all possible orientations during the fall.

For a state transition involving a mode switch i.e. standing-to-fallen or fallen-to-standing, the transition density  $f_+^{(1)}(x_+|x, \ell, 0)$  or  $f_+^{(0)}(x_+|x, \ell, 1)$  takes the form (26),

with position noise and extent (in logarithm) noise parameterized by:

$$v^{(p)} = [0.0049, 0.0049, 0.0049]^T, \\ v^{(s)} = [0.01, 0.01, 0.01]^T.$$

Notice that the position noise is increased in the case of a mode switch compared to the case of no switching, in order to capture the abrupt change in the size of the object during mode switching.

The birth density is an LMB with parameters  $P_{B,+}(\ell) = 0.001$  and

$$f_{B,+}(x, \ell, 0) = 0.9\mathcal{N}(x; \mu_{B,+0}, \Sigma_{B,+0}), \\ f_{B,+}^{(\ell)}(x, \ell, 1) = 0.1\mathcal{N}(x; \mu_{B,+1}, \Sigma_{B,+1}), \\ \mu_{B,+0} = [2.03 \ 0 \ 0.71 \ 0 \ 0.825 \ 0 \ -1.2 \ -1.2 \ -0.18]^T, \\ \mu_{B,+1} = [2.03 \ 0 \ 0.71 \ 0 \ 0.413 \ 0 \ -0.18 \ -0.18 \ -1.2]^T, \\ \Sigma_{B,+0} = \Sigma_{B,+1} = 0.1^2 \mathbf{I}_9.$$

#### 5.4.2 Effectiveness of Occlusion Model

Tables 5 and 6 show the CLEAR MOT and OSPA<sup>(2)</sup> benchmarks for MV-GLMB-OC and MS-GLMB on both detectors YOLOv3 and Faster-RCNN(VGG16). The CLEAR evaluations for the monocular detections are given in Appendix 7.4.

For CMC4 which has a maximum of 3 people, both MV-GLMB-OC and MS-GLMB on either detectors achieved high CLEAR MOT scores in MOTA/MOTP, and low OSPA<sup>(2)</sup> errors. The incidence of FPs and FNs is caused by track initiation/termination mismatches with the ground truths. Nonetheless, we observe that on MOTA/MOTP and OSPA<sup>(2)</sup>, MV-GLMB-OC outperforms MS-GLMB.

For CMC5 which has a maximum of 7 people, both MV-GLMB-OC and MS-GLMB on either detectors were still capable of producing reasonable MOTA/MOTP scores and OSPA<sup>(2)</sup> errors. Fig. 10 shows a snapshot of detections and estimates on a single view. However, due to poor detections and more occlusions in CMC5, we observe many IDs and FNs. Again on MOTA/MOTP and OSPA<sup>(2)</sup>, MV-GLMB-OC outperforms MS-GLMB.

TABLE 5: CMC4,5 Performance Benchmarks for 3D Position Estimates

CMC4 (Jumping and Falling, Maximum/Average 3 people)													
Detector and Tracker	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$	MT $\uparrow$	PT $\downarrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$	FM $\downarrow$	MOTA $\uparrow$	MOTP $\uparrow$	OSPA <sup>(2)</sup> $\downarrow$
YOLOv3+MV-GLMB-OC	<b>99.3%</b>	<b>99.0%</b>	<b>99.5%</b>	<b>3</b>	<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>	<b>0</b>	<b>0</b>	<b>98.5%</b>	<b>89.5%</b>	<b>0.16m</b>
YOLOv3+MV-GLMB-OC*	95.0%	93.5%	96.5%	<b>3</b>	<b>0</b>	<b>0</b>	17	4	5	1	93.6%	87.7%	0.18m
YOLOv3+MS-GLMB	95.9%	94.0%	97.8%	<b>3</b>	<b>0</b>	<b>0</b>	21	5	4	1	92.6%	86.4%	0.21m
Faster-RCNN(VGG16)+MV-GLMB-OC	98.0%	98.5%	97.5%	<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>	<b>7</b>	<b>2</b>	<b>1</b>	<b>97.0%</b>	<b>87.1%</b>	<b>0.18m</b>
Faster-RCNN(VGG16)+MV-GLMB-OC*	85.1%	82.2%	88.1%	<b>3</b>	<b>0</b>	<b>0</b>	29	<b>0</b>	6	<b>0</b>	91.3%	86.6%	0.19m
Faster-RCNN(VGG16)+MS-GLMB	89.3%	85.8%	93.1%	<b>3</b>	<b>0</b>	<b>0</b>	34	<b>0</b>	12	<b>0</b>	88.6%	87.0%	0.22m

CMC5 (Jumping and Falling, Maximum/Average 7 people)													
Detector and Tracker	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$	MT $\uparrow$	PT $\downarrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$	FM $\downarrow$	MOTA $\uparrow$	MOTP $\uparrow$	OSPA <sup>(2)</sup> $\downarrow$
YOLOv3+MV-GLMB-OC	<b>60.5%</b>	<b>63.5%</b>	<b>61.3%</b>	<b>3</b>	<b>4</b>	<b>0</b>	<b>388</b>	<b>933</b>	<b>55</b>	<b>47</b>	<b>61.1%</b>	<b>69.3%</b>	<b>0.63m</b>
YOLOv3+MV-GLMB-OC*	59.3%	58.1%	60.1%	<b>3</b>	<b>3</b>	<b>1</b>	418	1172	69	60	56.7%	63.9%	0.69m
YOLOv3+MS-GLMB	50.9%	51.1%	47.6%	<b>3</b>	<b>2</b>	<b>2</b>	735	1699	85	69	50.7%	59.5%	0.79m
Faster-RCNN(VGG16)+MV-GLMB-OC	60.1%	62.5%	60.1%	<b>3</b>	<b>4</b>	<b>0</b>	410	1185	61	49	60.3%	64.1%	0.66m
Faster-RCNN(VGG16)+MV-GLMB-OC*	56.2%	55.6%	59.2%	<b>3</b>	<b>3</b>	<b>1</b>	534	1493	63	61	55.7%	63.6%	0.70m
Faster-RCNN(VGG16)+MS-GLMB	49.1%	49.7%	46.1%	<b>3</b>	<b>3</b>	<b>1</b>	781	1337	92	69	49.6%	61.6%	0.80m

CLEAR MOT scores and OSPA<sup>(2)</sup> distance are calculated on standard position estimates ( $\uparrow$  means higher is better while  $\downarrow$  means lower is better). Two different detectors are considered - Faster-RCNN(VGG16) (monocular) and YOLOv3 (monocular). Two types of trackers are considered - MV-GLMB-OC (multi-view with occlusion model) and MS-GLMB (multi-sensor without occlusion model). The asterisk (\*) indicates the multi-camera reconfiguration experiment.

TABLE 6: CMC4,5 Performance Benchmarks for 3D Centroid with Extent Estimates

CMC4 (Jumping and Falling, Maximum/Average 3 people)													
Detector and Tracker	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$	MT $\uparrow$	PT $\downarrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$	FM $\downarrow$	MOTA $\uparrow$	MOTP $\uparrow$	OSPA <sup>(2)</sup> $\downarrow$
YOLOv3+MV-GLMB-OC	<b>99.3%</b>	<b>99.0%</b>	<b>99.5%</b>	<b>3</b>	<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>	<b>0</b>	<b>0</b>	<b>98.5%</b>	<b>60.1%</b>	<b>0.18</b>
YOLOv3+MV-GLMB-OC*	95.0%	93.5%	96.5%	<b>3</b>	<b>0</b>	<b>0</b>	17	4	5	1	93.6%	58.9%	0.20
YOLOv3+MS-GLMB	95.9%	94.0%	97.8%	<b>3</b>	<b>0</b>	<b>0</b>	21	5	4	1	92.6%	57.0%	0.26
Faster-RCNN(VGG16)+MV-GLMB-OC	98.0%	98.5%	97.5%	<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>	<b>7</b>	<b>2</b>	<b>1</b>	<b>97.0%</b>	<b>59.3%</b>	<b>0.20</b>
Faster-RCNN(VGG16)+MV-GLMB-OC*	85.1%	82.2%	88.1%	<b>3</b>	<b>0</b>	<b>0</b>	29	<b>0</b>	6	<b>0</b>	91.3%	56.2%	0.24
Faster-RCNN(VGG16)+MS-GLMB	89.3%	85.8%	93.1%	<b>3</b>	<b>0</b>	<b>0</b>	34	<b>0</b>	12	<b>0</b>	88.6%	55.3%	0.28

CMC5 (Jumping and Falling, Maximum/Average 7 people)													
Detector and Tracker	IDF1 $\uparrow$	IDP $\uparrow$	IDR $\uparrow$	MT $\uparrow$	PT $\downarrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$	FM $\downarrow$	MOTA $\uparrow$	MOTP $\uparrow$	OSPA <sup>(2)</sup> $\downarrow$
YOLOv3+MV-GLMB-OC	<b>59.8%</b>	<b>61.0%</b>	<b>60.8%</b>	<b>3</b>	<b>4</b>	<b>0</b>	<b>404</b>	<b>951</b>	<b>67</b>	<b>54</b>	<b>60.6%</b>	<b>45.0%</b>	<b>0.65</b>
YOLOv3+MV-GLMB-OC*	55.9%	54.9%	57.1%	<b>3</b>	<b>3</b>	<b>1</b>	689	1125	80	85	55.3%	43.4%	0.71
YOLOv3+MS-GLMB	49.5%	50.1%	45.0%	<b>3</b>	<b>2</b>	<b>2</b>	715	1750	94	91	49.3%	42.6%	0.78
Faster-RCNN(VGG16)+MV-GLMB-OC	58.1%	60.8%	59.4%	<b>3</b>	<b>4</b>	<b>0</b>	451	1008	72	57	59.9%	43.1%	0.66
Faster-RCNN(VGG16)+MV-GLMB-OC*	55.9%	53.6%	51.6%	<b>3</b>	<b>3</b>	<b>1</b>	569	1519	81	88	51.4%	42.7%	0.75
Faster-RCNN(VGG16)+MS-GLMB	48.8%	45.3%	41.7%	<b>3</b>	<b>3</b>	<b>1</b>	734	1493	96	98	43.3%	43.9%	0.81

CLEAR MOT scores and OSPA<sup>(2)</sup> distance are calculated with a 3D GIoU base-distance for estimates of 3D centroid with extent ( $\uparrow$  means higher is better while  $\downarrow$  means lower is better). Two different detectors are considered - Faster-RCNN(VGG16) (monocular) and YOLOv3 (monocular). Two types of trackers are considered - MV-GLMB-OC (multi-view with occlusion model) and MS-GLMB (multi-sensor without occlusion model). The asterisk (\*) indicates the multi-camera reconfiguration experiment.

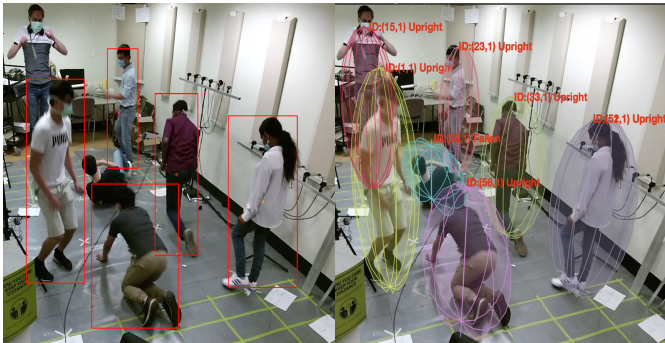


Fig. 10: CMC5 Camera 1: YOLOv3 detections (left) and MV-GLMB-OC estimates (right).

### 5.4.3 Multi-Camera Reconfiguration

The multi-camera reconfiguration experiment described in Section 5.3.3 is repeated for the multi-modal datasets CMC4 and CMC5. The results for the multi-camera reconfiguration are denoted with asterisks in Tables 5 and 6. The plot for OSPA<sup>(2)</sup> with 3D GIoU base-distance over a sliding window with time is given in Fig. 11. While similar observations can be made from the experiments without jumping and falling (CMC1-CMC3), the results for CMC4-CMC5 exhibit different behavior for people in the fallen state. The estimated extent is warped out of its ordinary shape when the person is on the ground, and more data is required to infer the

corresponding state of the fallen person. In CMC4-CMC5, the effect of occlusions or misdetections is exacerbated by having fewer cameras when the person is on the ground, which would likely lead to track termination or switching. Nonetheless, the results confirm that the JMS variant of the MV-GLMB-OC algorithm can automatically accommodate multi-camera reconfiguration.

TABLE 7: MV-GLMB-OC Runtime on WILDTRACKS and CMC

Dataset (Cams)	Frames	No. Obj (avg)	Exec. Time (s/frame)
W.T. (7)	400	20	18.0
CMC1 (4)	261	3	0.1
CMC2 (4)	263	10	3.2
CMC3 (4)	263	15	7.9
CMC4 (4)	147	3	0.4
CMC5 (4)	560	7	5.5

## 5.5 Runtimes

The runtimes for the MV-GLMB-OC filter on the WILDTRACKS and CMC datasets are summarized in Table 7. The current implementation is via unoptimized MATLAB code. The reported runtimes appear to be consistent with the computational complexity of the MV-GLMB-OC algorithm: quadratic in the number of objects and linear in the sum of the number of detections across all cameras.

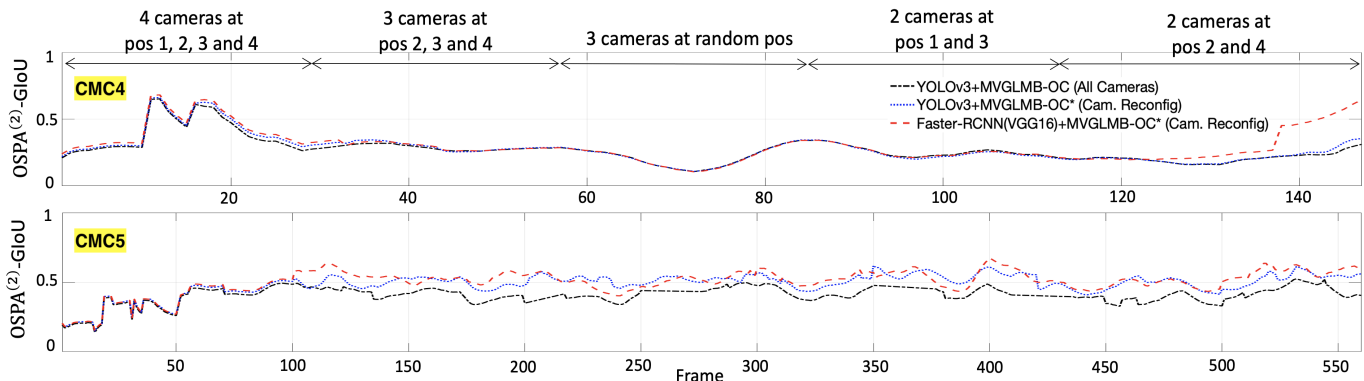


Fig. 11: Multi-Camera Reconfiguration Experiment: OSPA<sup>(2)</sup> plots with 3D GloU base-distance for estimates of 3D centroid with extent. Three trackers are considered: YOLOv3+MV-GLMB-OC\* (multi-camera reconfiguration) and Faster-RCNN+MV-GLMB-OC\* (multi-camera reconfiguration) and with YOLOv3+MV-GLMB-OC (all cameras operational).

## 6 CONCLUSIONS

By developing a tractable 3D occlusion model, we have derived an online Bayesian multi-view multi-object filtering algorithm that only requires monocular detector training, independent of the multi-camera configurations. This enables the multi-camera system to operate uninterrupted in the event of extension/reconfiguration (including camera failures), obviating the need for multi-view retraining. Moreover, it addresses the multi-camera data association problem in a way that is scalable in the total number of detections. Experiments on existing 3D multi-camera datasets have demonstrated similar performance to the state-of-the-art batch method. We also demonstrated the ability of the proposed algorithm to track in densely populated scenarios with high occlusions, and with people jumping/falling in the 3D world frame.

## REFERENCES

- [1] F. Poiesi, R. Mazzon, and A. Cavallaro, "Multi-target tracking on confidence maps: An application to people tracking," *COMPUT VIS IMAGE UND*, vol. 117, no. 10, pp. 1257–1272, 2013.
- [2] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1614–1627, 2014.
- [3] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, "Multi-view people tracking via hierarchical trajectory composition," in *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4256–4265, 2016.
- [4] D. Y. Kim, B.-N. Vo, B.-T. Vo, and M. Jeon, "A labeled random finite set online multi-object tracker for video data," *Pattern Recognition*, vol. 90, pp. 377–389, 2019.
- [5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, 2011.
- [6] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, 2014.
- [7] X. Wang, E. Turetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2312–2326, 2016.
- [8] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, 2010.
- [9] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2010.
- [10] 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*, vol. 45, no. 10, pp. 3625–3635, 2012.
- [11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2014.
- [12] P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang, "Robust multiple cameras pedestrian detection with multi-view Bayesian network," *Pattern Recognition*, vol. 48, no. 5, pp. 1760–1772, 2015.
- [13] A. Andriyenko, S. Roth, and K. Schindler, "An analytical formulation of global occlusion reasoning for multi-target tracking," in *2011 IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, pp. 1839–1846, IEEE, 2011.
- [14] S. L. Dockstader and A. M. Tekalp, "Multiple camera fusion for multi-object tracking," in *Proc. 2001 IEEE Workshop on Multi-Object Tracking*, pp. 95–102, IEEE, 2001.
- [15] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 267–282, 2008.
- [16] T. Chavdarova and F. Fleuret, "Deep multi-camera people detection," in *2017 16th IEEE Int. Conf. Mac. Learning and Applicat. (ICMLA)*, pp. 848–853, IEEE, 2017.
- [17] T. Chavdarova et al., "WILDTRACK: A multi-camera HD dataset for dense unscripted pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5030–5039, 2018.
- [18] P. Baqué, F. Fleuret, and P. Fua, "Deep occlusion reasoning for multi-camera multi-target detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 271–279, 2017.
- [19] J. Domke, "Learning graphical model parameters with approximate marginal inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2454–2467, 2013.
- [20] B.-N. Vo, B.-T. Vo, and M. Beard, "Multi-sensor multi-object tracking with the generalized labeled multi-bernoulli filter," *IEEE Trans. Signal Process.*, vol. 67, no. 23, pp. 5952–5967, 2019.
- [21] M. Beard, B. T. Vo, and B.-N. Vo, "A solution for large-scale multi-object tracking," *IEEE Trans. on Signal Process.*, vol. 68, pp. 2754–2769, 2020.
- [22] R. Girshick, "Fast R-CNN," in *Proc. of the IEEE Int. Conf. on Comput. Vis.*, pp. 1440–1448, 2015.
- [23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Proc. Systems*, pp. 91–99, 2015.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Process. Systems*, pp. 1097–1105, 2012.
- [25] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 7263–7271, 2017.
- [26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 779–788, 2016.
- [27] X. Zhao, H. Jia, and Y. Ni, "A novel three-dimensional object detection with the modified you only look once method," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 2, 2018.
- [28] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *2008 IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 1–8, IEEE, 2008.
- [29] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *Trans. Pat. Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, 2013.



- [30] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning," in *Proc. of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6499–6508, 2020.
- [31] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun, "PnPNet: End-to-end perception and prediction with tracking in the loop," in *Proc. of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 11553–11562, 2020.
- [32] K. Otsuka and N. Mukawa, "Multiview occlusion analysis for tracking densely populated objects based on 2-d visual angles," in *Proc. of the 2004 IEEE Comput. Soc. Conf. on Comput. Vis. and Pattern Recognit.*, vol. 1, pp. I–I, IEEE, 2004.
- [33] A. Osep, W. Mehner, M. Mathias, and B. Leibe, "Combined image- and world-space tracking in traffic scenes," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1988–1995, IEEE, 2017.
- [34] P. Li, J. Shi, and S. Shen, "Joint spatial-temporal optimization for stereo 3D object tracking," in *Proc. of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6877–6886, 2020.
- [35] M. Pedersen, J. B. Haurum, S. H. Bengtson, and T. B. Moeslund, "3D-ZeF: A 3D zebrafish tracking benchmark dataset," in *Proc. of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 2426–2436, 2020.
- [36] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3D tracking by detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 635–642, IEEE, 2018.
- [37] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proceedings of the IEEE International Conf. on Comput. Vis.*, pp. 2365–2374, 2019.
- [38] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 433–440, IEEE, 2018.
- [39] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu, "Joint monocular 3D vehicle detection and tracking," in *Proceedings of the IEEE International Conf. on Comput. Vis.*, pp. 5390–5399, 2019.
- [40] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1683–1698, 2008.
- [41] X. Weng and K. Kitani, "A baseline for 3D multi-object tracking," *arXiv preprint arXiv:1907.03961*, 2019.
- [42] R. P. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [43] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, 2014.
- [44] R. P. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., 2007.
- [45] E. Maggio, M. Taj, and A. Cavallaro, "Efficient multitarget visual tracking using random finite sets," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 18, no. 8, pp. 1016–1027, 2008.
- [46] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3460–3475, 2013.
- [47] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conf. Comput. Vis.*, pp. 17–35, Springer, 2016.
- [48] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 1–6, IEEE, 2009.
- [49] X. Alameda-Pineda, J. Staiano, R. Subramanian, L. Batrinca, E. Ricci, B. Lepri, O. Lanz, and N. Sebe, "Salsa: A novel dataset for multimodal group behavior analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1707–1720, 2015.
- [50] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2003.
- [51] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.
- [52] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable, "Tracking and data association," 1990.
- [53] S. Blackman and R. Popoli, "Design and analysis of modern tracking systems (artech house radar library)," *Artech house*, 1999.
- [54] T. T. D. Nguyen and D. Y. Kim, "GLMB tracker with partial smoothing," *Sensors*, vol. 19, no. 20, p. 4419, 2019.
- [55] M. Beard, B.-T. Vo, and B.-N. Vo, "Bayesian multi-target tracking with merged measurements using labelled random finite sets," *IEEE Trans. Signal Processing*, vol. 63, no. 6, pp. 1433–1447, 2015.
- [56] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pat. Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [57] P. Schneider and D. H. Eberly, *Geometric tools for computer graphics*. Elsevier, 2002.
- [58] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [59] R. P. Mahler, B.-T. Vo, and B.-N. Vo, "CPHD filtering with unknown clutter rate and detection profile," *IEEE Trans. on Signal Process.*, vol. 59, no. 8, pp. 3497–3513, 2011.
- [60] C.-T. Do and T. T. D. Nguyen, "Multiple marine ships tracking from multistatic Doppler data with unknown clutter rate," in *Int. Conf. on Control, Autom. and Inf. Sci. (ICCAIS)*, pp. 1–6, IEEE, 2019.
- [61] B.-N. Vo, B.-T. Vo, and H. G. Hoang, "An efficient implementation of the generalized labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1975–1987, 2017.
- [62] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.
- [63] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 319–336, 2008.
- [64] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, 2008.
- [65] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 658–666, 2019.
- [66] H. Rezatofighi, T. T. D. Nguyen, B.-N. Vo, B.-T. Vo, S. Savarese, and I. Reid, "How trustworthy are the existing performance evaluations for basic vision tasks?," *arXiv preprint arXiv:2008.03533*, 2020.
- [67] A. Maksai, X. Wang, F. Fleuret, and P. Fua, "Non-Markovian globally consistent multi-object tracking," in *Proceedings of the IEEE International Conf. on Comput. Vis.*, pp. 2544–2554, 2017.
- [68] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [69] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3246–3260, 2014.
- [70] Y. G. Punchihewa, B.-T. Vo, B.-N. Vo, and D. Y. Kim, "Multiple object tracking in unknown backgrounds with labeled random finite sets," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 3040–3055, 2018.