Robust, Real-Time 3D Tracking of Multiple Objects with Similar Appearances

Taiki Sekii
Panasonic System Networks R&D Lab. Co., Ltd.
sekii.taiki@jp.panasonic.com

Abstract

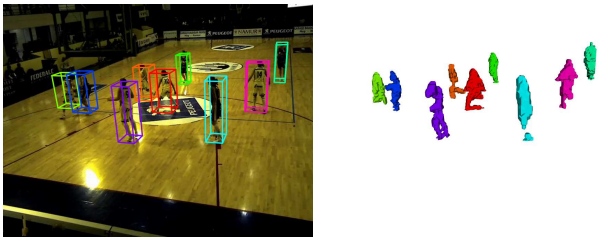
This paper proposes a novel method for tracking multiple moving objects and recovering their three-dimensional (3D) models separately using multiple calibrated cameras. For robustly tracking objects with similar appearances, the proposed method uses geometric information regarding 3D scene structure rather than appearance. A major limitation of previous techniques is foreground confusion*, in which the shapes of objects and/or ghosting artifacts are ignored and are hence not appropriately specified in foreground regions. To overcome this limitation, our method classifies foreground voxels into targets (objects and artifacts) in each frame using a novel, probabilistic two-stage framework. This is accomplished by step-wise application of a* track graph *describing how targets interact and the maximum a posteriori expectation-maximization algorithm for the estimation of target parameters. We introduce mixture models with semiparametric component distributions regarding 3D target shapes. In order to not confuse artifacts with objects of interest, we automatically detect and track artifacts based on a closed-world assumption. Experimental results show that our method outperforms state-of-the-art trackers on seven public sequences while achieving real-time performance.*

1. Introduction

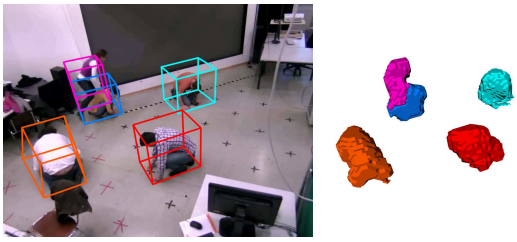
Multiple object tracking has long been an important task in computer vision research. It has broad applications such as surveillance, sports analysis, and human-computer interaction and is available in various types, depending on the final goal and the assumptions made:

- Track objects from one or more cameras.
- Track objects offline or online.

This paper focuses on online tracking of multiple objects and takes advantage of multiple cameras to deal with crowded scenes exhibiting objects and occlusions with significant density. In such a scenario, we pursue real-time performance for separately recovering three-dimensional (3D) models of multiple objects on a standard desktop PC, as shown in Fig. 1.



(a) APIDIS dataset.



(b) LEAF-2 sequence.

Figure 1: 3D models of objects separately recovered using the proposed method (*cf.* § 5).

1.1. Related works

We review not only existing multiple object trackers using multiple cameras but also previous works that involve separately recovering 3D models of multiple objects. In the initial tracking steps (TSs), most conventional works extract *foreground information* regarding foreground moving objects, *e.g.*, foreground voxels, foreground point crowds, and object presence likelihood maps at discretized locations. This information is typically obtained by projecting or accumulating detection responses from each camera in a common 3D space [6, 11, 15, 18, 19, 22, 24, 26] or two-dimensional (2D) plane [8, 10, 13, 16, 17, 21] in which objects move. These detection responses are generally acquired using either object detectors or standard background subtraction techniques [25]. Foreground information defined in a 2D space (2DFI) tends to generate more artifacts

than that defined in a 3D space (3DFI), owing to the absence of altitude; hence, such information induces low robustness in crowded scenes in most cases [18, 24].

Previous work using 3DFI can be divided into two approaches: one exploits appearances as primary cues for distinguishing objects [11, 18, 26] and the other is based on geometric information regarding 3D scene structure [6, 15, 19, 22, 24]. In the former methods, visual hulls are created by a volume-based shape-from-silhouette technique. Appearance models, which are trained based on their silhouettes for each object over time, are used to resolve occlusions in certain contexts, *e.g.*, voxel classification into objects [11] and tracking using Kalman filtering [18] or mean-shift [26]. These algorithms achieve very good results for objects whose appearances are discriminative but do not perform as well for those with the similar appearances.

In contrast, geometry-based methods handle occlusions using schemes that rely less on the appearances of objects and are hence not affected by the similarity of their appearances. Some of these separately recover 3D models of objects in each frame, for exsample, the level set method [15], human body model fitting [19], or the iterative closed point algorithm [22]. These methods can handle occlusions caused by two or three objects moving two-dimensionally on the ground-plane.

Some researchers have proposed geometry-based methods using particle filter frameworks in the context of multiple-object tracking and introduced the local mass density scores of voxel-based visual hulls for computing the posterior probabilities of particles [6, 24]. Most notably, Possegger et al. [24] achieved state-of-the-art performance by resolving occlusions using Voronoi partitioning of the hypothesis space.

Another approach to our scenario is to apply data-association methods that do not require appearance models [2, 5, 12, 27]. These formulate tracking as an optimization problem in the space of all possible families of trajectories and solve this problem using optimization algorithms, *e.g.*, the k-shortest path algorithm [2].

A major limitation of such previous techniques, illustrated in Fig. 2, is *foreground confusion*, in which the shapes of objects and/or ghosting artifacts are ignored and are hence not appropriately specified in foreground regions. This causes accumulated drifts and results in low robustness in crowded scenes.

1.2. Contributions and outline

In this paper, we propose a geometry-based method to track multiple objects of which the appearances are not discriminative. Similar to certain previous works, foreground regions, which represent connected components of foreground voxels created by a volume-based shape-from-silhouette technique, are used as inputs. We assume here that silhouettes of moving objects are detected using standard background subtraction techniques in multiple static cameras calibrated and distributed in scenes. Thus, appearance information other than silhouettes is not used at all.

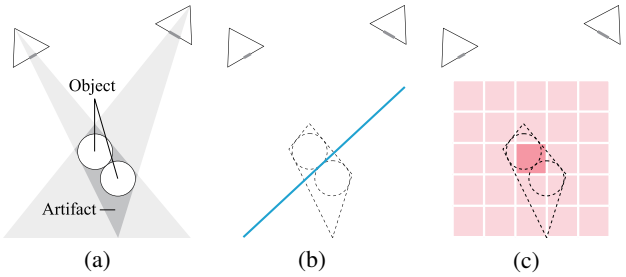


Figure 2: Examples of foreground confusion in two types of approaches (particle filter-based approaches using Voronoi partitioning (b) and typical data-association approaches using discretized maps (c)). Foreground regions, in which objects and ghosting artifacts can exist in scenes, are partially contained in a common region (*e.g.*, each side separated by a boundary of Voronoi cells (the blue line) in (b) and the central grid in (c)) and are handled as regions of a common target.

To overcome foreground confusion, we classify foreground voxels into objects and artifacts, which we call *targets*, using a novel, probabilistic two-stage framework. In the first stage, a candidate(s) for targets to which each foreground region can belong is extracted by constructing a *track graph* that describes events in which targets are isolated or interact with one another [9, 23]. In the second stage, each foreground voxel is classified into any one of the candidates, as specified in the first stage, by applying the maximum a posteriori expectation-maximization (MAP-EM) algorithm for the estimation of target parameters. Here, we introduce mixture models with semiparametric probabilistic density functions (PDFs) representing 3D target shapes. In order to not confuse artifacts with objects, we automatically detect and track artifacts using a *closed-world assumption*, in which an unknown object cannot suddenly appear at an arbitrary position [14, 24] (Fig. 3). In contrast to some previous studies [18, 27] that introduce graph structures to region-wisely model object interactions, our framework has the capabilities to voxel-wisely handle foreground confusion.

2. Overview of the method

Targets are tracked in each frame using the following TSs:

1. Obtain foreground segmentations from input images in each camera.

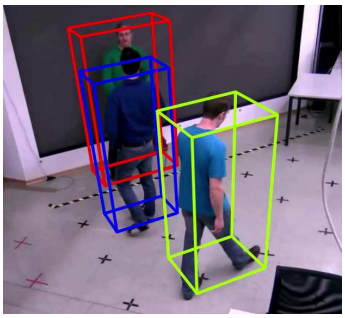


Figure 3: A ghosting artifact detected and tracked in our framework.

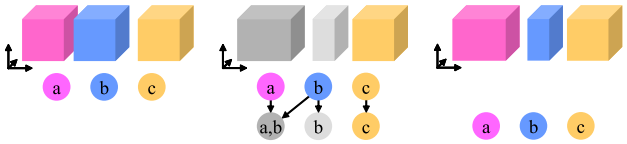


Figure 4: Construction and update of a track graph. A rectangular foreground region and the node corresponding to it are shown in a common color. Alphabets in nodes indicate labels to which their foreground regions can belong. The track graph updated at $(t-1)$ (a) is constructed (b) (§ 3.1) and updated (c) at t (§ 3.2).

2. Perform shape-from-silhouette using foreground segmentations obtained in TS 1 and extract foreground regions.
3. Detect foreground regions constructed from newly appearing objects either manually or using a detector.
4. Construct the track graph and obtain indices (labels) of candidate targets to which each foreground region might belong.
5. Classify foreground voxels into any one of the candidate(s), obtained in TS 4, using MAP-EM.
6. Update the track graph with the results in TS 5.

In these TSs, we assume the following:

- Moving distances of targets are sufficiently short that their foreground regions overlap with those in the previous frame.
- Shape changes of targets are sufficiently small to be ignorable in tracking.
- Foreground regions, which are not labeled in TS 3 and do not interact with those in the previous frame, can be regarded as ghosting artifacts based on the closed-world assumption.

The construction and updating of the track graph are described in §3. These correspond to TSs 4 and 6, respectively. §4 presents foreground voxel classification using MAP-EM, which corresponds to TS 5. In addition, the set of symbols is listed in the supplementary material.

3. Classification of foreground regions into targets

3.1. Construction of track graph

Given foreground regions at current time t, in order to obtain candidate(s) to which they (and their voxels) can belong, we label them by a set of labels $\mathcal{L} = \{1, \ldots, m\}$, which indicate indices of targets. m is the number of targets. This is achieved by constructing the track graph as follows.

The track graph is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ represent a set of foreground region nodes and a set of edges, respectively. Here, $\mathcal{V}$ is updated as follows:

$$\mathcal{V} = \mathcal{V}^t + \mathcal{V}^{t-1}, \tag{1}$$

where $\mathcal{V}^t$ and $\mathcal{V}^{t-1}$ represent foreground region nodes given at t and at $(t-1)$, respectively. Each random variable $v \in \mathcal{V}^t$ can take on one of the indices of foreground regions. $\mathcal{E}$ consists of temporal relations between foreground regions given at t and at $(t-1)$, *i.e.*,

$$\mathcal{E} = \{(i,j) | \text{Area}(i \cap j) > 0\}, \tag{2}$$

where $i \in \mathcal{V}^t$ and $j \in \mathcal{V}^{t-1}$. Then, the node information of i is represented as $(\mathcal{X}_i, \mathcal{L}_i)$. $\mathcal{X}_i$ and $\mathcal{L}_i$ are the set of 3D coordinates of the foreground voxels constructing i and the set of labels of the candidates to which i can belong, respectively. $\mathcal{L}_i$ is initialized as $\emptyset$ in the first frame at $t = 0$ and is computed at $t > 0$ as follows:

$$\mathcal{L}_i = \bigcup_{j' \in \mathcal{V}_i^{t-1}} \mathcal{L}_{j'}, \tag{3}$$

where $\mathcal{V}_i^{t-1}$ is the temporal neighboring nodes of i in the neighborhood system on the track graph, *i.e.*,

$$\mathcal{V}_i^{t-1} = \{j' | (i,j') \in \mathcal{E}\}. \tag{4}$$

Fig. 4(b) shows an example of track-graph construction.

Let i_s denote a foreground region detected as a newly tracked object in TS 3 and let i_u denote one that is still not labeled (*i.e.*, artifact regions that appear without being identified and thus satisfy $\mathcal{L}_{i_u} = \emptyset$). Here, s and u are a new object index and a new artifact, respectively, and are one-by-one created as $(m+1)$, following which m is increased. We label their nodes by s and u, initializing $\mathcal{L}_{i_s}$ and $\mathcal{L}_{i_u}$

with $\{s\}$ and $\{u\}$, respectively. Then, s and u are added to $\mathcal{L}$.

We group $\mathcal{L}$ into subsets of labels of targets interacting with each other at t, and we call them *merging groups* because their foreground regions merge when they interact. Let the k^{th} merging group be denoted by $\mathcal{T}_k \subseteq \mathcal{L}$ and let $\mathcal{T} = \{\mathcal{T}_k\}_{k\in\mathcal{K}}$, where $\mathcal{K}$ is a set of indices of merging groups. $\mathcal{T}_k$ is defined as follows:

$$\mathcal{T}_k = \bigcup_{i'\in\mathcal{V}'} \mathcal{L}_{i'}, \tag{5}$$

where

$$\mathcal{V}' \subseteq \mathcal{V}^t : \bigcap_{i'\in\mathcal{V}'} \mathcal{L}_{i'} \neq \emptyset. \tag{6}$$

Finally, if one object and one or more artifacts with small volume interact with each other or if two or more artifacts interact with each other, we merge them. This leads to a reduction in the interactions of targets, which need to be resolved in MAP-EM, and results in a significant reduction in computational complexity of MAP-EM. This is achieved by removing merging groups based on several constraints. Incrementally, if the number of object indices in $\mathcal{T}_k$ is one and the volume of an artifact in $\mathcal{T}_k$ is less than a threshold volume v, its artifact index is removed from $\{\mathcal{L}_i\}_{i\in\mathcal{V}^t}$. Then, $\mathcal{T}_k$ is removed from $\mathcal{T}$ if $|\mathcal{T}_k| = 1$. On the other hand, if all labels in $\mathcal{T}_k$ are artifact indices, the labels (excluding the oldest artifact index) and $\mathcal{T}_k$ are removed from $\{\mathcal{L}_i\}_{i\in\mathcal{V}^t}$ and $\mathcal{T}$, respectively.

Based on the track graph, a function for obtaining candidates of targets to which a foreground voxel $\mathbf{x}$ can belong is defined as follows:

$$C(\mathbf{x}) = \begin{cases} \mathcal{L}_i & \text{if } \exists i : \mathbf{x} \in \mathcal{X}_i, \\ \emptyset & \text{otherwise.} \end{cases} \tag{7}$$

3.2. Update of track graph

We assume that each foreground voxel is labeled by any one of the labels $\hat{l} \in \mathcal{L}$ in TS 5. First, all of the nodes and edges in $\mathcal{G}$ are removed, *i.e.*, $\mathcal{V} = \emptyset$, $\mathcal{E} = \emptyset$. In connected components of foreground voxels labeled by a common label $\hat{l}$, we regard the component with maximum volume as the foreground region created by $\hat{l}$ and denote the set of nodes of such foreground regions by $\hat{\mathcal{V}}^t$. Then, the set of nodes $\hat{\mathcal{V}}^t$ is added to $\mathcal{V}$ and their node information is computed. Fig. 4(c) shows an example of an update of the track graph.

4. Classification of foreground voxels into targets

In this section, PDFs used in MAP-EM are described first. Subsequently, an algorithm to classify foreground voxels into targets using MAP-EM is described.

4.1. PDFs

We consider the classification of foreground voxels into targets as the inference problem of determining the joint distribution $p(\mathbf{x}, l)$. Here, $\mathbf{x}$ represents the 3D coordinates of a foreground voxel and $l \in \mathcal{L}$. This distribution can be expressed in the form

$$p(\mathbf{x}, l) = p(l)p(\mathbf{x}|l), \tag{8}$$

where $p(l)$ is the prior. This prior is used as the mixing coefficient in the EM framework and is given by

$$p(l) = \pi_l, \tag{9}$$

where π_l must satisfy

$$\sum_{l\in\mathcal{L}} \pi_l = 1 \ \wedge \ 0 \leq \pi_l \leq 1, \forall l \in \mathcal{L}. \tag{10}$$

Let $\boldsymbol{\mu}_l^t$ denote the 3D position of l at t. The posterior distribution on the right-hand side of (8) is defined as the class-conditional likelihood function that is the probability of $\mathbf{x}$ given $\boldsymbol{\mu}_l^t$ and is given by

$$p(\mathbf{x}|l) \sim p(\mathbf{x}|\boldsymbol{\mu}_l^t). \tag{11}$$

Here, we represent $p(\mathbf{x}|\boldsymbol{\mu}_l^t)$ as a semiparametric PDF representing the 3D shape of a target. If the shape of l does not change (or can be ignored in tracking) from a particular previous time $\hat{t} < t$ to t and $\mathbf{x}$ belongs to l at t, then $\mathbf{x}$ corresponds to any of the dense foreground voxels belonging to l at $\hat{t}$. Thus, we first represent a likelihood function, where $\mathbf{x}$ belongs to any of a set of voxels $\mathcal{Y}$, using the local mass density [24] as follows:

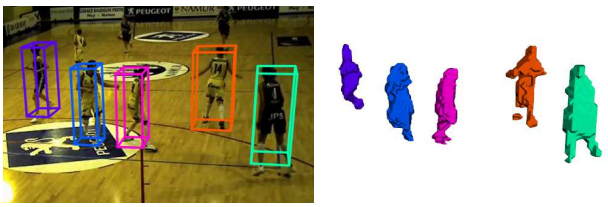
$$f(\mathbf{x}|\mathcal{Y}) = \frac{S}{M}|\mathcal{R}_{\mathbf{x},\mathcal{Y}}|, \tag{12}$$

where $\mathcal{R}_{\mathbf{x},\mathcal{Y}}$ is a subset of $\mathcal{Y}$, which are contained in an $r_x \times r_y \times r_z$ cuboid R centered at $\mathbf{x}$ and is given by

$$\begin{aligned} \mathcal{R}_{\mathbf{x},\mathcal{Y}} = \{(x', y', z')^{\mathrm{T}} | \ & |x' - x| \leq r_x/2 \\ & \wedge |y' - y| \leq r_y/2 \\ & \wedge |z' - z| \leq r_z/2 \\ & \wedge (x', y', z')^{\mathrm{T}} \in \mathcal{Y}\}, \end{aligned} \tag{13}$$

where $\mathbf{x} = (x, y, z)^{\mathrm{T}}$. S and M are the normalizing constant and the volume of R, respectively. Then, we define $p(\mathbf{x}|\boldsymbol{\mu}_l^t)$ using f as a likelihood function, where a foreground voxel $\mathbf{x} - \boldsymbol{\mu}_l^t + \boldsymbol{\mu}_l^{\hat{t}}$, to which $\mathbf{x}$ is shifted along the moving direction of l from t to $\hat{t}$, belongs to any of the foreground voxels constructed from l at $\hat{t}$:

$$p(\mathbf{x}|\boldsymbol{\mu}_l^t) = f(\mathbf{x} - \boldsymbol{\mu}_l^t + \boldsymbol{\mu}_l^{\hat{t}}|\mathcal{X}_{j_l}), \tag{14}$$



(a) 3D models of tracked targets.

(b) Projected semiparametric PDFs.

Figure 5: As a class-conditional likelihood function for each target, we use a semiparametric PDF (b) computed from a 3D model of a target (a). In (b), each class colors every target, and a shade of a color displays a likelihood value projected.

where $\mathcal{X}_{j_l}$ is the set of 3D coordinates of the voxels in the foreground region that are constructed from l at $\hat{t}$. Here, $\boldsymbol{\mu}_l^{\hat{t}}$ and $\mathcal{X}_{j_l}$ on the right-hand side of (14) are determined at $\hat{t}$ and are constant at t. In addition, there is no assumption regarding the form of the distribution of $p(\mathbf{x}|\boldsymbol{\mu}_l^t)$, which depends on the arbitrary shape of l. Thus, we see that $p(\mathbf{x}|\boldsymbol{\mu}_l^t)$ is the semiparametric PDF of which the parameters are nothing but $\boldsymbol{\mu}_l^t$. Fig. 5 shows an example of this PDF. In our implementation, $p(\mathbf{x}|\boldsymbol{\mu}_l^t)$ is preserved as a lookup table for every target and is updated when l is isolated, since the shape of a target recovered is unstable while interacting with the others.

For the MAP estimate of $\boldsymbol{\mu}_l^t$ in MAP-EM, assuming that positions of targets at t are near those at $(t-1)$, we define the prior for $\boldsymbol{\mu}_l^t$ as the multivariate Gaussian distributions

$$p(\boldsymbol{\mu}_l^t) \sim \mathcal{N}(\boldsymbol{\mu}_l^t|\boldsymbol{\mu}_l^{t-1}, \boldsymbol{\Sigma}), \tag{15}$$

where $\boldsymbol{\Sigma}$ denotes the covariance.

4.2. MAP-EM

In the following, we first adapt MAP-EM to our problem and then classify foreground voxels into targets using the track graph and MAP-EM. Here, we explain the EM algorithm based on its description in [4, 7, 20].

4.2.1 Overview of MAP-EM

Let $\mathcal{X}$ denote the set of 3D coordinates of all foreground voxels. Our goal in using MAP-EM is to find the maximum a posteriori solution for our mixture models with the above-mentioned PDFs when the *incomplete*-data set $\mathcal{X}$ is given. This is achieved using the iterative framework with $p(\mathbf{x}, l)$ (§4.1).

We first initialize the set of all model parameters

$$\Theta = \{\pi_l, \boldsymbol{\mu}_l^t\}_{l\in\mathcal{L}} \tag{16}$$

and then iteratively compute the revised estimate Θ^{new} from the current estimate Θ^{old} through the E and M steps. In the E step, we use Θ^{old} to find the posterior distribution $p(l|\mathbf{x}, \Theta^{\text{old}})$ $(\forall \mathbf{x} \in \mathcal{X})$ of the latent variables (labels in our case). In the M step, we determine Θ^{new} by maximizing the sum of the expectation of the *complete*-data log likelihood $Q(\Theta, \Theta^{\text{old}})$ and the logarithm of the prior $p(\Theta)$ with respect to Θ as follows:

$$\Theta^{\text{new}} = \arg\max_{\Theta} S(\Theta, \Theta^{\text{old}}), \tag{17}$$

where

$$S(\Theta, \Theta^{\text{old}}) = Q(\Theta, \Theta^{\text{old}}) + \ln p(\Theta). \tag{18}$$

This expectation $Q(\Theta, \Theta^{\text{old}})$, which we describe later, is sometimes called the *Q-function*. Assuming that the $\boldsymbol{\mu}_l^t$'s are independent of each other, we approximate $\ln p(\Theta)$ in (18) using the prior $p(\boldsymbol{\mu}_l^t)$ given by (15) as follows:

$$\ln p(\Theta) \sim \sum_{l\in\mathcal{L}} \ln p(\boldsymbol{\mu}_l^t). \tag{19}$$

Q-function. As mentioned above, the Q-function is the expectation of the complete-data log likelihood and is expressed in the form

$$Q(\Theta, \Theta^{\text{old}}) = \sum_{l\in\mathcal{L}} p(l|\mathcal{X}, \Theta^{\text{old}}) \ln p(\mathcal{X}, l|\Theta). \tag{20}$$

Here, if the $\mathbf{x}$'s are independent, then $p(l|\mathcal{X}, \Theta)$ and $\ln p(\mathcal{X}, l|\Theta)$ can be respectively written using (8), (11), (15), (16), and Bayes' theorem as follows:

$$p(l|\mathcal{X}, \Theta) = \frac{p(l)p(\mathcal{X}|l, \Theta)}{p(\mathcal{X})} = \frac{p(l)\prod_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}|l)}{\prod_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x})}. \tag{21}$$

$$\begin{aligned} \ln p(\mathcal{X}, l|\Theta) &= \ln \prod_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}, l) \\ &= \sum_{\mathbf{x}\in\mathcal{X}} \{\ln p(l) + \ln p(\mathbf{x}|l)\}. \end{aligned} \tag{22}$$

Then, the Q-function can be consequently rewritten by making use of (8), (9), (11), (20) − (22), and Bayes' theorem as follows:

$$Q(\Theta, \Theta^{\text{old}}) = \sum_{l\in\mathcal{L}} \sum_{\mathbf{x}\in\mathcal{X}} \gamma_{\mathbf{x},l} \{\ln \pi_l + \ln p(\mathbf{x}|\boldsymbol{\mu}_l^t)\}, \tag{23}$$

where $\gamma_{\mathbf{x},l}$ is called the *responsibility* that component l takes for "explaining" the observation $\mathbf{x}$ and is given by

$$\gamma_{\mathbf{x},l} = p(l|\mathbf{x}, \Theta^{\text{old}}). \tag{24}$$

Here, $p(l|\mathbf{x}, \Theta)$ is derived by making use of (8), (9), (11), (16), and Bayes' theorem as follows:

$$p(l|\mathbf{x}, \Theta) = \frac{p(l)p(\mathbf{x}|l)}{p(\mathbf{x})} = \frac{\pi_l p(\mathbf{x}|\boldsymbol{\mu}_l^t)}{\sum_{l' \in \mathcal{L}} \pi_{l'} p(\mathbf{x}|\boldsymbol{\mu}_{l'}^t)}, \quad (25)$$

where $p(\mathbf{x}|\boldsymbol{\mu}_l^t)$ is given by (14).

MAP estimates of parameter Θ. In the M step, we estimate Θ^{new} to maximize $S(\Theta, \Theta^{\text{old}})$. We obtain $\pi_l^{\text{new}} \in \Theta^{\text{new}}$ by solving the constrained extremal problem with (10) using Lagrange's multiplier method as follows:

$$\pi_l^{\text{new}} = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} \gamma_{\mathbf{x},l}. \quad (26)$$

In addition, the steepest descent method is used to estimate $\boldsymbol{\mu}_l^{t\,\text{new}} \in \Theta^{\text{new}}$. Let $\hat{\boldsymbol{\mu}}_l^t$ be the current estimate of $\boldsymbol{\mu}_l^t$ and let $\hat{\Theta} = \{\pi_l^{\text{old}}, \hat{\boldsymbol{\mu}}_l^t\}_{l \in \mathcal{L}}$, where $\pi_l^{\text{old}} \in \Theta^{\text{old}}$. $\boldsymbol{\mu}_l^t$ is initialized by $\boldsymbol{\mu}_l^{t\,\text{old}} \in \Theta^{\text{old}}$ and is updated as follows:

$$\boldsymbol{\mu}_l^t = \hat{\boldsymbol{\mu}}_l^t + \epsilon \frac{\partial S(\hat{\Theta}, \Theta^{\text{old}})}{\partial \boldsymbol{\mu}_l^t}, \quad (27)$$

where ϵ is the value of the step size.

4.2.2 MAP-EM based on track graph

If straightforwardly adapting only MAP-EM to our problem, we are faced with loss of efficiency and redundant calculations to a certain extent. For example, when two foreground regions are sufficiently distant, it is impossible for their voxels to be labeled by a common label. Thus, such foreground voxels must be separately handled. In our EM framework, this can be naturally achieved by using prior knowledge about the relationships between foreground regions and targets, which are obtained from the track graph.

Before applying MAP-EM, we first specify targets, which are isolated and do not interact with the others, using the track graph. Let $\check{l}$ denote the index of such a target. $l = \check{l}$, if $l \notin \mathcal{T}_k, \forall k \in \mathcal{K}$. Then, each foreground voxel $\mathbf{x}$ is labeled by $\check{l}$ if $C(\mathbf{x}) = \{\check{l}\}$.

We subsequently apply MAP-EM to the remaining targets, except for those mentioned above. Let $\bar{\mathcal{L}} \subseteq \mathcal{L}$ be the set of labels $\bar{l} \neq \check{l}$ and let $i_{\bar{l}} \in \mathcal{V}^t$ be a foreground region node that satisfies $\bar{l} \in \mathcal{L}_{i_{\bar{l}}}$. The target parameter $\bar{\Theta} = \{\pi_{\bar{l}}, \boldsymbol{\mu}_{\bar{l}}^t\}_{\bar{l} \in \bar{\mathcal{L}}}$ is first initialized using $\{\hat{\pi}_{\bar{l}}, \boldsymbol{\mu}_{\bar{l}}^{t-1}\}_{\bar{l} \in \bar{\mathcal{L}}}$. $\hat{\pi}_{\bar{l}}$ indicates the ratio of the volume of $\bar{l}$ to the total volume of $\bar{\mathcal{L}}$ at $(t-1)$. In the E step, revising (25) using the track graph, we compute $\gamma_{\mathbf{x},\bar{l}}$ as follows:

$$\gamma_{\mathbf{x},\bar{l}} = p(\bar{l}|\mathbf{x}, \bar{\Theta}^{\text{old}}) \sim \begin{cases} \frac{\pi_{\bar{l}} p(\mathbf{x}|\boldsymbol{\mu}_{\bar{l}}^t)}{\sum_{\bar{l}' \in C(\mathbf{x})} \pi_{\bar{l}'} p(\mathbf{x}|\boldsymbol{\mu}_{\bar{l}'}^t)} & \text{if } \bar{l} \in C(\mathbf{x}), \\ 0 & \text{otherwise.} \end{cases} \quad (28)$$

In the M step, we revise (23) by substituting $\Theta = \bar{\Theta}$, $\mathcal{L} = \bar{\mathcal{L}}$, and $\mathcal{X} = \mathcal{X}_{i_{\bar{l}}}$ and revise (19) by substituting $\Theta = \bar{\Theta}$ and $\mathcal{L} = \bar{\mathcal{L}}$. Then, we estimate $\bar{\Theta}^{\text{new}}$, which maximizes $S(\bar{\Theta}, \bar{\Theta}^{\text{old}})$, as mentioned in §4.2.1. In addition, each foreground voxel $\mathbf{x}$ is labeled by $\arg\max_l \gamma_{\mathbf{x},l}$. Finally, the 3D position of target $\check{l}$ and that of $\bar{l}$ are computed as the averages of the 3D coordinates of foreground voxels labeled by $\check{l}$ and $\bar{l}$, respectively.

5. Experiments

5.1. Datasets

We evaluate our approach on seven challenging video sequences contained in two publicly available datasets (APIDIS[1] and ICG-Lab-6 [24]), for which the performance of some baselines has been provided by their original authors. In these datasets, 2D ground truth object positions in the ground-plane are distributed. Thus, it must be noted that our method is designed for 3D tracking, but is validated with respect to only 2D coordinates of objects without the altitude (height). The technical characteristics of the datasets are summarized in Table 1. In the following, we explain the datasets based on the descriptions in [24].

APIDIS. This dataset provides a one-minute sequence of a basketball game. This sequence contains various challenges, *e.g.*, the similar appearance of all players in a common team, many occlusions, and the degradations of foreground segmentations caused by strong shadows and reflections on the floor. Similar to [24], we evaluate the performance on the left half of the court (15×15 m^2) and use the imageries from cameras covering that side, *i.e.*, cameras 1, 2, 4, 5, and 7.

ICG-Lab-6. This dataset provides six sequences that show various situations. These correspond to CHAP,

[1] http://sites.uclouvain.be/ispgroup/index.php/Softwares/APIDIS

Table 1: Sequence characteristics indicating the number of cameras N_{C}, the maximum number of simultaneously visible objects N_{O}, the total number of frames, the frame rate (FPS), and the resolution of the video streams.

Sequence	N_{C}	N_{O}	Frames	FPS	Resolution
APIDIS	7	12	1500	25	800×600
CHAP	4	5	3760	20	1024×768
LEAF-1	4	4	1800	20	1024×768
LEAF-2	4	5	2400	20	1024×768
MUCH	4	5	2400	20	1024×768
POSE	4	6	1820	20	1024×768
TABLE	4	5	1760	20	1024×768

LEAF-1, LEAF-2, MUCH, POSE, and TABLE in Table 1. In this dataset, the appearance of people is discriminative since people wear clothes of different colors. Thus, the proposed method is compared with only the geometry-based methods.

The **CHAP** sequence shows a standard surveillance scenario in which five people move close to each other two-dimensionally on the floor and imposes additional challenges to the appearance-based tracking approaches, because people change their appearance throughout the scene by putting on jackets with significantly different colors than their sweaters.

The **LEAF-1** and **LEAF-2** sequences show leapfrog games in which players move three-dimensionally by leaping over each other's stooped backs. These scenarios impose several challenges, such as difficult poses, out-of-plane motion, and frequent collisions between players.

The **MUCH** sequence shows *musical chairs* (also known as *Going to Jerusalem*), in which players race to sit down in one of the chairs. This sequence exhibits specific challenges owing to the nature of this game. Players move quickly, and there are dynamic background items, *viz.*, a chair is removed after each round.

In the **POSE** sequence, six people show various poses, such as kneeling, crawling, and sitting. In addition to these poses, *e.g.*, the upright standing pose and movement on the common ground-plane, which violate the assumptions for pedestrians, the background illumination changes, thereby causing further challenges to robust foreground segmentation.

The **TABLE** sequence shows five people walking and jumping over a table. While people are moving, dense crowding creates additional challenges.

5.2. Evaluation metrics

For quantitative evaluation, we rely on the standard CLEAR multiple-object tracking (MOT) performance metric [3], *viz.*, MOT accuracy (MOTA). The MOTA score is computed from three error ratios, false negatives (FNs), false positives (FPs), and identity switches (IDSs). Higher MOTA values indicate better robustness, with its maximum value being one, representing a perfect tracking result. To compute the MOTA score, similar to [24], we set the distance threshold between the ground truth and tracking results to 50 cm in all the sequences.

5.3. Implementation details

We use a standard background subtraction method using Gaussian mixture models [28] in the foreground segmentation step. It is implemented using OpenCV [1] and its parameters are changed to not chip silhouettes for each camera. Foreground regions extracted in TS 2 are represented as a 3D binary image. This image is smoothed and binarized before being used for the track graph, because the visual hull reconstruction is sensitive to noise, *i.e.*, missing foreground segmentations cause holes in the volume and its separations. In addition, foreground regions, which appear at the defined entry areas and are bigger than half the size of an average human, are detected as regions constructed from newly appearing objects in TS 3.

The reference parameters used in this paper are presented as follows. The volume threshold v for merging objects and artifacts (§3.1) is set to half the volume of an average human. Voxel sizes are set to $6 \times 6 \times 6$ cm^3 in the APIDIS dataset and to $4 \times 4 \times 4$ cm^3 in ICG-Lab-6. The sizes of the cuboids used to compute semiparametric PDFs representing 3D shapes of targets (§4.1) are set to $5 \times 5 \times 5$ voxel^3 in the APIDIS dataset and to $7 \times 7 \times 7$ voxel^3 in the ICG-Lab-6 dataset. The kernel sizes used to smooth foreground regions in TS 2 are set to $3 \times 3 \times 5$ voxel^3 in the APIDIS dataset and to $5 \times 5 \times 9$ voxel^3 in the ICG-Lab-6 dataset. We consider a prior for a target position $p(\boldsymbol{\mu}_l^t)$ in (15) as an isotropic Gaussian governed by a single precision parameter α (*i.e.*, $\boldsymbol{\Sigma} = \alpha^2 \boldsymbol{I}$), and α is set to 30 cm for objects and to voxel sizes for artifacts.

5.4. Results and discussion

Illustrative results of our method are shown in Fig. 1, while Table 2 lists the performance metrics on the individual sequences[2]. The baselines [24, 2] correspond to two methods whose foreground confusion is explained in Fig. 2(b) and Fig. 2(c), respectively; the results of [24] are taken from its original paper and those of [2] are taken from [24]. We found that our method is more reliable than other state-of-the-art trackers based on the MOTA scores and the IDS in both standard visual surveillance scenarios (*e.g.*, CHAP and LEAF-1) and more complex ones (*e.g.*, APIDIS, LEAF-2, MUCH, POSE, and TABLE). Considering the high number of FP scores, baselines suffer from missed detections of objects and drifts to artifacts over all the sequences, in contrast to our method specifying artifacts. This is one of the sources of their lower MOTA scores.

On the other hand, for the LEAF-1 and LEAF-2 sequences, in which players three-dimensionally contact each other, our IDS scores are clearly better than those of the baselines, even when considering our higher FN score for LEAF-2 (actually several contacts by two objects occurred during our FNs). In addition, our method outperforms the appearance-based method ([24] w/ color) for the APIDIS dataset, in which players wearing similar jerseys interact with each other. These show the utility of using geometric information regarding 3D target shapes in such complex situations, *e.g.*, the contact by two players in Fig. 1(b). However, in several sequences, our method sometimes could not

[2]For the supplementary material and videos, please visit: http://taikisekii.com

resolve insignificant occlusions that previous trackers could by using appearance information such as a discriminative visual classifier [24].

Table 3 lists the performance metrics of some different versions of the proposed method. First, when artifacts are ignored in our approach (reported as *w/o artifact*), *i.e.*, when suddenly appearing foreground regions are not tracked, the performance deteriorates from that of the full implementation (reported as *Full.*). This shows that tracking of artifacts in our approach is effective in improving foreground confusion between objects and artifacts. Second, when we use the multivariate Gaussian distribution as a PDF representing 3D target shapes (reported as *w/ Gaus.*), the performance similarly deteriorates. This shows that our proposed semiparametric PDF expresses more realistic target shapes. Third, when the track graph is not used (reported as *w/o track graph*), *i.e.*, all targets can be subject to candidates to which foreground voxels can belong in MAP-EM, the performance similarly deteriorates. This shows that the track graph works effectively as prior knowledge for classification of foreground voxels into objects in MAP-EM.

Table 2: Quantitative comparison results of the proposed method with other state-of-the-art trackers. Tracking in [24] is conducted with and without the appearance models in the APIDIS dataset (reported as *w/ color* and *w/o color*, respectively). For each evaluated dataset, we report the accuracy metric MOTA (higher is better), as well as the total number of TPs, FPs, FNs, and IDSs. The best values for each evaluation and each criterion are highlighted.

Sequence	Method	MOTA	TP	FP	FN	IDS
APIDIS	[2]	0.490	607	156	220	46
	[24] w/ color	0.675	656	88	172	9
	[24] w/o color	0.597	625	121	202	10
	Ours	**0.855**	**738**	**18**	**95**	**8**
CHAP	[2]	0.952	**1607**	50	21	7
	[24] w/o color	0.719	1316	193	241	**4**
	Ours	**0.989**	1582	**9**	**5**	**4**
LEAF-1	[2]	0.976	**495**	6	**1**	5
	[24] w/o color	0.721	436	83	44	7
	Ours	**0.991**	465	**1**	**1**	**2**
LEAF-2	[2]	0.819	**913**	87	**66**	24
	[24] w/o color	0.727	856	115	117	34
	Ours	**0.842**	832	**5**	144	**5**
MUCH	[2]	0.754	**770**	139	**32**	26
	[24] w/o color	0.736	694	99	99	11
	Ours	**0.808**	672	**23**	119	**10**
POSE	[2]	0.555	427	156	31	17
	[24] w/o color	0.822	456	42	44	**3**
	Ours	**0.910**	**494**	**37**	**5**	**3**
TABLE	[2]	0.719	573	105	58	14
	[24] w/o color	0.818	577	56	**51**	7
	Ours	**0.894**	**738**	**18**	95	**6**

5.5. Runtime performance

Our approach is implemented in C++ without any code optimization and is conducted on a standard desktop PC with a 3.4 GHz Intel CPU, 32 GB RAM, and a GeForce GTX970. The average speeds are 26 FPS and 25 FPS on the APIDIS sequence and the ICG-Lab-6 dataset, respectively. We achieved frame rates greater than those at which each sequence is actually recorded, although only background subtraction in the foreground segmentation step is processed on the GPU, exploiting inherent parallelism. On the other hand, when the track graph is not used in our method (w/o track graph in Table 3), its average speeds decrease to 13 FPS and 11 FPS on the APIDIS dataset and ICG-Lab-6, respectively. This shows that the track graph can accelerate MAP-EM and increases its range of real-world applicability.

6. Conclusion

We proposed a method for tracking multiple objects and separately recovering their 3D models using multiple calibrated cameras. Our principal innovations for robustly tracking objects with similar appearances are to incorporate geometric information regarding 3D scene structure and a track graph describing how objects and artifacts interact in the MAP-EM framework and to probabilistically classify foreground voxels into any of them. In the experiments, we confirmed that our method outperforms state-of-the-art trackers on seven public sequences while achieving real-time performance. In future work, to improve performance in cases where objects are discriminative, we plan to explore a tracking framework in which appearance information of targets is incorporated into our proposed approach as additional tracking cues.

Table 3: Quantitative comparison results of different versions of the proposed method, averaged over all seven sequences. Three error ratios for computing the MOTA scores are indicated by FPR, FNR, and IDSR (FP, FN, and IDS ratios, respectively).

Method	MOTA	FPR	FNR	IDSR
w/o artifact	0.858	0.024	0.108	0.009
w/ Gaus.	0.878	0.060	**0.052**	0.010
w/o track graph	0.864	0.033	0.094	0.009
Full.	**0.898**	**0.021**	0.073	**0.007**

References

[1] OpenCV: Open Source Computer Vision. `http://opencv.org`. 7

[2] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 33(9):1806–1819, 2011. 2, 7, 8

[3] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. In *EURASIP JIVP*, 2008. 7

[4] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006. 5

[5] M. Byeon, S. Oh, K. Kim, H. Yoo, and J. Y. Choi. Efficient spatio-temporal data association using multidimensional assignment for multi-camera multi-target tracking. In *BMVC*, 2015. 2

[6] C. Canton-Ferrer, J. R. Casas, M. Pardàs, and E. Monte. Multi-camera multi-object voxel-based monte carlo 3D tracking strategies. *EURASIP JASP*, 2011(114):1–15, 2011. 1, 2

[7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977. 5

[8] R. Eshel and Y. Moses. Tracking in a dense crowd using multiple cameras. *IJCV*, 88(1):129–143, 2010. 1

[9] P. Figueroa, N. Leite, R. Barros, I. Cohen, and G. Medioni. Tracking soccer players using the graph representation. In *ICPR*, 2004. 2

[10] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *PAMI*, 30(2):267–282, 2008. 1

[11] L. Guan, J. S. Franco, and M. Pollefeys. Probabilistic multi-view dynamic scene reconstruction and occlusion reasoning from silhouette cues. *IJCV*, 90(3):283–303, 2010. 1, 2

[12] M. Hofmann, D. Wolf, and G. Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *CVPR*, 2013. 2

[13] M. Hu, J. Lou, W. Hu, and T. Tan. Multicamera correspondence based on principal axis of human body. In *ICIP*, 2004. 1

[14] S. S. Intille and A. F. Bobick. Visual tracking using closed-worlds. In *ICCV*, 1995. 2

[15] Y. Iwashita, R. Kurazume, K. Hara, and T. Hasegawa. Robust motion capture system against target occlusion using fast level set method. In *ICRA*, 2006. 1, 2

[16] S. M. Khan and M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *PAMI*, 31(3):505–519, 2009. 1

[17] K. Kim and L. S. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *ECCV*, 2006. 1

[18] M. C. Liem and D. M. Gavrila. Joint multi-person detection and tracking from overlapping cameras. *CVIU*, 128:36–50, 2014. 1, 2

[19] X. Luo, B. Berendsen, R. T. Tan, and R. C. Veltkamp. Human pose estimation for multiple persons based on volume reconstruction. In *ICPR*, 2010. 1, 2

[20] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley, second edition, 2008. 5

[21] A. Mittal and L. S. Davis. M_2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV*, 51(3):189–203, 2003. 1

[22] D. Mitzel and B. Leibe. Taking mobile multi-object tracking to the next level: People, unknown objects, and carried items. In *ECCV*, 2012. 1, 2

[23] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking - linking identities using bayesian network inference. In *CVPR*, 2006. 2

[24] H. Possegger, S. Sternig, T. Mauthner, P. M. Roth, and H. Bischof. Robust real-time tracking of multiple objects by volumetric mass densities. In *CVPR*, 2013. 1, 2, 4, 6, 7, 8

[25] A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *CVIU*, 122:4–21, 2014. 1

[26] A. Tyagi, M. Keck, J. Davis, and G. Potamianos. Kernel-based 3D tracking. In *IEEE International Workshop on Visual Surveillance*, 2007. 1, 2

[27] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *CVPR*, 2012. 2

[28] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, 2004. 7