Ph.D. DISSERTATION

Simultaneous Association and Localization for Multi-Camera Multi-Target Tracking

다중 카메라에서 다중 물체 추적을 위한
동시적 데이터 연관 및 위치 추정

BY

BYEON MOONSUB

AUGUST 2017

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Simultaneous Association and Localization for Multi-Camera Multi-Target Tracking

다중 카메라에서 다중 물체 추적을 위한
동시적 데이터 연관 및 위치 추정

지도교수 최 진 영
이 논문을 공학박사 학위논문으로 제출함

2017년 8월

서울대학교 대학원

전기 컴퓨터 공학부

변 문 섭

변문섭의 공학박사 학위 논문을 인준함

2017년 8월

위 원 장:	조 남 익
부위원장:	최 진 영
위 원:	오 성 회
위 원:	곽 노 준
위 원:	이 민 식

Abstract

In this dissertation, we propose two approaches for three-dimensional (3D) localizing and tracking of multiple targets by using images from multiple cameras with overlapping views. The main challenge is to solve the 3D position estimation problem and the trajectory assignment problem simultaneously. However, most of the existing methods solve these problems independently. Unlike single camera multi-target tracking, it is much more complicated to solve both problems because the relationship between cameras is also taken into consideration in multi-camera. To tackle this challenge, we present two approaches: mixed multidimensional assignment approach and variational inference approach. In the mixed multidimensional assignment approach, we formulate the data association and 3D trajectory estimation problem as the mixed optimization problem with discrete and continuous variables and suggest an alternative optimization scheme which jointly solves the two coupled problems. To handle a large solution space, we develop an efficient optimization scheme that alternates between two coupled problems with a reasonable computational load. In this optimization formulation, we design a new cost function that describes 3D physical properties of each target. In the variational inference approach, we establish a maximum a posteriori (MAP) problem over trajectory assignments and 3D positions for given detections from multiple cameras. To find a solution, we develop an expectation-maximization scheme, where the probability distributions are designed by following the Boltzmann distribution of seven terms induced from multi-camera tracking settings.

keywords: 3D localization and tracking, multiple cameras, multiple target tracking, multidimensional assignment, variational inference, 3D trajectory estimation

student number: 2012-30211

Contents

List of Tables

List of Figures

Chapter 1

Introduction

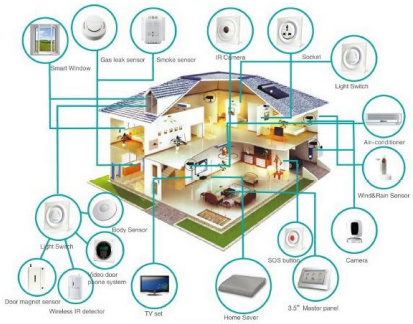
1.1 Background & Challenges

Background

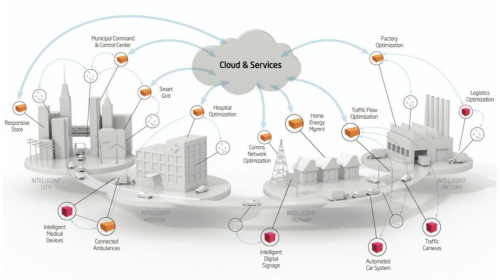
Ambient intelligence (AmI) refers to electronic environments that are sensitive and responsive to the presence of people. In Wikipedia, *in an ambient intelligence world, devices work in concert to support people in carrying out their everyday life activities, tasks and rituals in an easy, natural way using information and intelligence that is hidden in the network connecting these devices.* Examples of the application of AmI research are as follows:

- home automation systems [1, 2]
- distributed virtual communities with autonomous mobile agents [3, 4]
- intelligent city monitoring social threat in urban environment [5]
- interaction between remote patients and health care systems [6]

In order to achieve the AmI, it is necessary to establish a better ”intelligence” through sharing / cooperation among the various sensors.



(a) home automation systems



(b) intelligent city

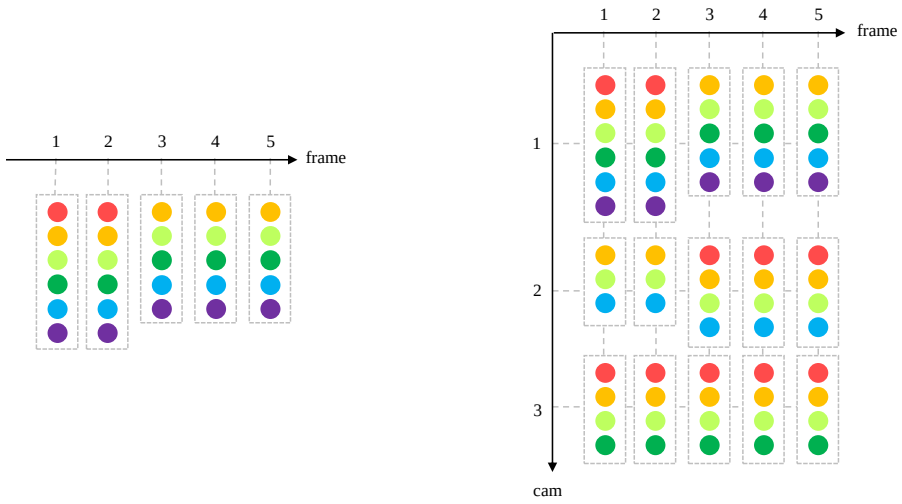
Figure 1.1: Two examples of the application of the ambient intelligence.

Multiple target tracking and localization in 3-dimensional (3D) space is one of the important issues that must be solved to achieve the ambient intelligence. Localization and tracking of multiple targets not only provides a higher level of service, but also enables a high level applications such as behavior understanding and action recognition. Various sensors can be used for the localization and tracking problem such as sonar [7], radar [8, 9, 10], and cameras. Among these various sensors, visual sensors plays an important role. It is because visual sensors have reasonable costs and easy to transfer and their video contains a wealth of information.

In this dissertation, we have investigated how to combine the data of multiple visual sensors to achieve better localization and tracking in a 3D space. The advantage of multiple overlapping cameras is that they can be seen by other cameras even if they are in one camera. The goal of this dissertation is to achieve robust localization and tracking performance even in occlusion situations by combining information from multiple overlapping cameras.

Challenges

Multi-target localization and tracking problem is well known problem in computer vision community, a task most often referred to multiple object tracking (MOT) or equivalently, multi-target tracking (MTT). The goal of general MTT or MOT is to



(a) data association in a single camera (b) data association in multiple cameras

Figure 1.2: Comparison of data association problem in a single camera and multiple cameras. (a) In the single-camera case, the number of possible associations is $6 \times 6 \times 5 \times 5 \times 5 = 4500$, (b) In the multi-camera case, $6 \times 6 \times 5 \times 5 \times 5 \times 3 \times 3 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 = 2,654,208,000$ which increases exponentially.

accurately estimate the position of multiple objects under the following conditions:

- the number of targets is unknown
- their number changes over time
- run automatically without manual initialization

The most challenging issue under these conditions is the automatic detection of the location of each object and the constant labeling of the unknown number of targets under missing detection and false detection caused by object occlusion.

In the case of a multi-camera, occlusion can be coped with the fact that the object can be seen by other cameras even if fully occluded in one camera, but there is still a more challenging part in some respects. First, the solution space is larger than the solution space of a single camera because the relationship between the different cameras

must be considered. As shown in Figure 1.2, if there are thousands of possible association numbers in a single camera, then the number of possible associations in a multi-camera will increase exponentially with billions Although several single camera-based approaches [11, 12] have proposed methods to obtain a globally optimum in a polynomial time, it is not straightforward to directly apply the existing algorithms to find solutions in the large problem space. This is because, in the case of multiple cameras, the relationship between different cameras must also be considered. Therefore, there is a need for efficient solution space exploration methods and appropriate assumptions to limit solution space.

Second, the problem of localization of multiple cameras is more challenging than the problem of localization of single cameras. In the case of a single camera, the localization problem is the problem of locating the object's center or bounding boxes on the image plane. Thus, both the observation and the hidden state to be estimated exist in the image plane. On the other hand, in a multi-camera situation, it is a problem of finding the exact 3D position of objects. It is necessary to find the point or volume of the 3D space from the observation on the image plane. Since 2D-3D ambiguity exists between the image plane and the 3D space, additional inference framework is required. In the next subsection, we will discuss how existing methods solved these multiple camera challenges.

1.2 Related Works

The problem of multiple target localization and tracking in a single camera has been studied for decades and has been actively studied recently [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. Most of single camera-based methods estimate the 3D position of people through ground plane assumption that a person stands on a 3D virtual plane. The ground plane assumption means trajectories on the plane rather than complete 3D trajectories. Nevertheless, several single camera-based methods have attempt to

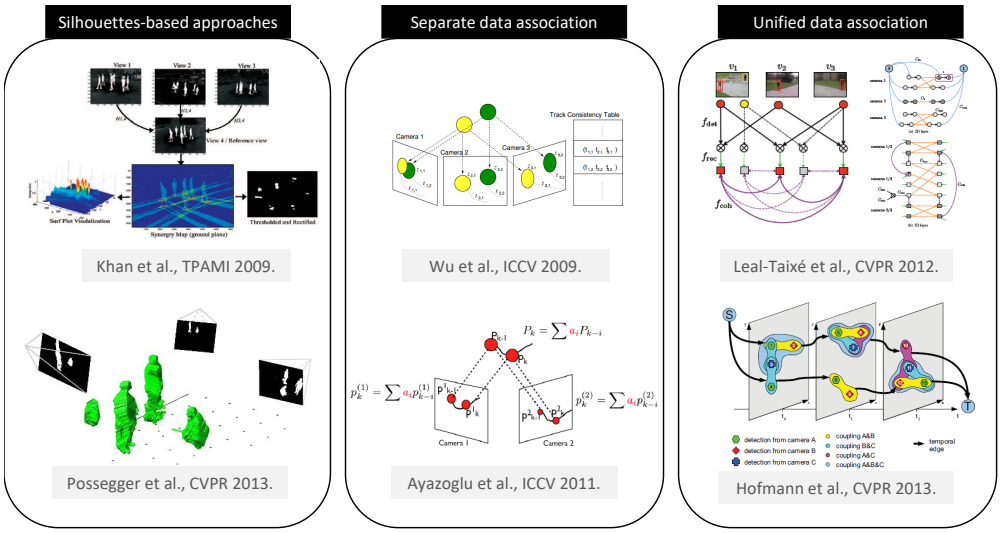


Figure 1.3: Three categories of existing approaches for multi-camera multi-target tracking. All figures are copied from the original papers.

solve *data association* and *trajectory estimation* simultaneously [18, 26]. Andriyenko *et al.* [18] successfully formulates *data association* as a discrete optimization problem and *trajectory estimation* as a continuous optimization problem. However, one of the main drawbacks of single camera-based approaches is that it suffers from occlusions because tracking targets may not be observed at all when the targets become severely occluded.

To overcome the problem of occlusions, multiple camera-based approaches have attempted to integrate the observations from multiple cameras and given promising results [27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]. In multiple camera-based approaches, one of main challenges is that the tracking and localization problem should be solved in two domains: time (i.e., temporally) and camera domain (i.e., spatially). As shown in Figure 1.3, the multiple camera-based approaches are roughly categorized into three groups: silhouette-based, *separate* and *unified* data association approaches.

Several silhouette-based approaches take into account observations from multi-

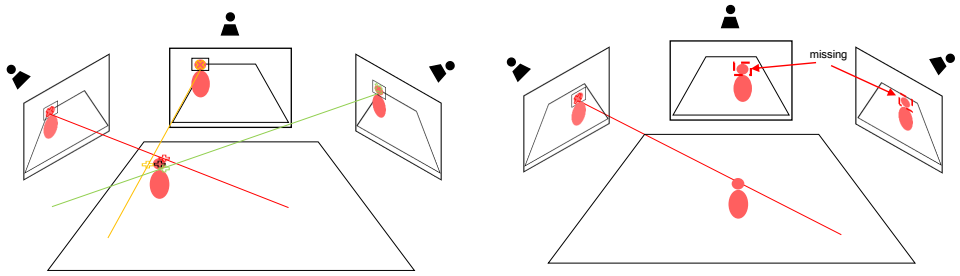
ple cameras at each time to generate global probability map called occupancy map [32, 39], synergy map [33], and volumetric density map [37, 40]. After reconstructing these global maps, they apply the single camera-based tracking approaches such as linear programming [12, 39], graph cut [33], and filtering-based single target tracker [37, 40]. Some *separate* data association approaches make 3D hypotheses by fusing object detections from all cameras and solve (temporal) data association problem of the 3D hypotheses [34, 35]. The main disadvantage of the silhouette-based and *separate* data association approaches is that although observations from multiple cameras are spatio-temporally correlated, they do not fully exploit the observations because they sequentially solve two subproblems in each camera and time domain respectively. In addition, the error in the first subproblem can be accumulated to the next subproblem by consecutively solving two subproblems.

In recent years, there has been an increasing interest in the *unified* data association approaches solving the multiple target tracking problem in both two domains simultaneously [36, 38, 41]. By adopting the tracking-by-detection framework, two combinatorial problems are considered at the same time: spatial data association through cameras and temporal data association between frames. Since the *spatio-temporal data association* problem is a well known NP-hard problem even in a small number of cameras or frames (more than 3) [42], it is difficult to make the problem tractable. In [38] and [36], the *spatio-temporal data association* problem is formulated as a min-cost network flow problem with generating a graph among detections and solved it by a binary integer programming (BIP) solver. Especially, Hofmann *et al.* [38] solves the data association problem in *spatio-temporal* domain, which gives a notable improvement in tracking performance. However, the algorithm complexity of BIP solver grows exponentially with respect to the number of cameras and the algorithms require a large memory budget [36, 38]. Yoo *et al.* [41] extended the multi-hypothesis tracking [43] to multiple cameras in an online manner.

On the other hand, most of multiple camera-based approaches do not consider se-

riously localization problem, i.e., estimating 3D locations of targets from observations of multiple cameras. 3D locations are determined by a frame-by-frame manner using observations at the specific frame rather than by exploiting all observations at entire frames. Wu *et al.* [34] reconstructs 3D trajectories of flying bats by minimizing the sum of the stereoscopic reconstruction errors computed for each view. Bredereck *et al.* [35] calculates 3D locations by averaging the points triangulated from every camera pairs. Recent *unified* data association approaches assume a flat ground plane where 3D locations are computed along every observation pairs [36] and combinations [38]. They calculate 3D locations of targets by simply triangulating the observations at each frame and assuming that people move on a flat ground plane. The *ground plane assumption* does not hold when targets are jumping or flying apart from the ground plane (e.g. tracking heads of people who stand and sit in a 3D space). More importantly, in their formulation [34, 35, 36, 38], 3D locations are not variables but constants. Finally, they solve a *spatio-temporal data association* problem with given 3D locations of targets.

Here, we present an *unified* data association approach to incorporate not only tracking problem but localization problem into a single optimization formulation. In the unified optimization framework, we focus on exploiting all observations from every camera and frame to find the optimal values of assignment variables for *spatio-temporal data association* and 3D location variable for *3D trajectory estimation*. Compared to the previous approaches, relatively accurate 3D trajectories are estimated by combining the two correlated problems without any assumption on the scene structure. Unfortunately, it is difficult to solve the coupled problem that has two types of optimization variables and a large solution space. The objective of this paper is to deal with the coupled optimization issue at the same time and to develop an efficient optimization scheme requiring reasonable memory budget.



(a) Triangulation using multiple calibrated cameras (b) 2D image-3D space ambiguity

Figure 1.4: The localization issues in overlapping cameras.

1.3 Problem Statements & Contributions

Problem Statements

Most of multi-target tracking methods have utilized the observations detected by object classifier or background subtraction methods, which is called *tracking-by-detection* framework [17, 18, 19, 26, 38, 36]. In the tracking-by-detection framework, tracking means linking observations from the same object, which is called *data association*, and localization indicates predicting states (location, velocity, etc.) of each object, which is called *trajectory estimation*. The benefit of the tracking-by-detection framework is that it is robust to drifting and easy to recover from tracking failure.

By adopting the tracking-by-detection framework, tracking problem in overlapping cameras is still a data association problem like single camera. However, temporal data association to the time axis as well as spatial data association to the camera axis must be considered simultaneously. Therefore, the size of the combinatorial space is much larger than the single camera data association problem considering only the time axis. The localization problem in overlapping cameras is a problem of estimating three-dimensional positions from a bounding box detected from multiple cameras. Given calibration information, each camera can estimate the 3D position through triangulation (see Figure 1.4(a)). For example, the triangulation can be accomplished through the following process. One point of each image can be represented by a back-projection

line in three-dimensional space. If key-points are defined for the center points of each bounding box detected in multiple cameras, the key-points are represented by several back-projection lines in three-dimensional space. It is possible to estimate the position with the smallest distance from each back-projection line as a three-dimensional position. However, it is difficult to estimate the 3D position of a bounding box that is detected only in one camera (see Figure 1.4(b)). Therefore, it is necessary to determine the 3D position using the information of the adjacent time. The solution space that considers both the localization and tracking problem in overlapping cameras will exist in both discrete space of data association and continuous space of 3D localization so the solution space will be enormously large. In this dissertation, we propose two approaches that simultaneously solving the localization and tracking problems while efficiently resolves the solution space.

Contributions

We argue that solving localization and tracking problem simultaneously should lead to an accurate estimation of 3D trajectories by jointly considering two correlated problems. While exploiting all observations from each camera and frame, we incorporate localization and tracking problem into a single optimization framework combining *spatio-temporal data association* problem for tracking and *3D trajectory estimation* problem for localization. The main contribution of this dissertation will be to describe two optimization approach solving localizing and tracking problem simultaneously.

First, we present a mixed multidimensional assignment approach that incorporates both *spatio-temporal data association* and *3D trajectory estimation* in a single objective function (Chapter 2). To cope with significant increase of search space, we propose an approximation algorithm to solve the *spatio-temporal data association* problem with a reasonable computational load. To express an association corresponding to a target in the *spatio-temporal data association* problem, a new representation using a matrix to represent associations is introduced. We formulate the *spatio-temporal*

data association as a multidimensional assignment (MDA) problem of which optimization variables are called assignment matrices. Also, our optimization formulation handles localization problem for which 3D location variables are introduced to represent a solution of *3D trajectory estimation* problem. Furthermore, we also design a new cost function that describes the accuracy in 3D reconstruction, motion smoothness, missing detections from cameras, starting/ending zone, trajectory fragments, and false positives. In particular, the motion smoothness term is designed to model high-order motion of each object, which can reduce the possibility of ID switches.

Since two different types of optimization variables, i.e., discrete variables for assignment matrices and continuous variables for 3D location, are defined on the cost function, it is not straightforward to solve the joint optimization problem. To handle this mixed discrete-continuous optimization problem, we adopt an alternative optimization scheme where the assignment variables and the 3D location variables are alternatively optimized by fixing the other type of variables. For assignment matrices, the proposed approximation algorithm for the MDA problem iteratively improves a solution by a random splitting and optimal re-merging. While maintaining a feasible solution, the new solution is re-constructed by random splitting and optimal merging of the split assignment matrices. The new solution is evaluated by the proposed cost function and obtained so as to have a lower cost than the previous one. As the random splitting and re-merging is repeatedly performed, the new solution eventually converges to the local minimum. Hence, the proposed splitting/re-merging algorithm can be considered as a guided random search to find the global optimum through repeated random local searches. Given assignment matrices, the problem for 3D location variables becomes the traditional least squares problem and can be solved in a closed-form. To evaluate the performance of *3D trajectory estimation*, we present a new dataset containing the ground truth of 3D head trajectories of each person.

Second, we propose a variational inference approach to solve the localization and tracking problems simultaneously (Chapter 3). We formulate a maximum a posteriori

(MAP) problem on joint random variables of trajectory assignments and 3D positions for given detections from multi-camera 2D images. To tackle difficulties of inference the posterior distribution, we adopt a variational inference approximation to make the MAP problem tractable by marginalizing 3D position variable under the assumption of parametric variational distribution over the 3D position variable. By describing the variational distribution for the 3D position variable as Gaussian, we obtain a variational expectation-maximization (V-EM) formulation. In the V-EM, the mean and the variance of the 3D position variable are estimated in E-step and the assignment problem is solved as a a min-cost network problem in M-step. The remaining probability distributions used in the formulation are designed by following the Boltzmann distribution which are represented by seven terms induced from multi-camera tracking settings. By formulating and solving the joint localization and tracking problems, we achieve an accurate estimation of 3D trajectories, while outperforming the state-of-the-art methods.

Chapter 2

Mixed Multidimensional Assignment Approach

2.1 Problem Formulation

2.1.1 Problem Statements

Notation. Before explaining the details of the proposed method, Table 2.1 summarizes the notations used in the following sections. To easily notice the characteristic of the notations, bold capital letters $(\mathbf{A}, \mathbf{B}, ...)$ are used for discrete sets, while continuous ones are denoted by calligraphic capital letters $(\mathcal{A}, \mathcal{B}, ...)$. Standard capital letters $(A, B, ...)$ denote matrices, while standard small letters $(a, b, ...)$ represent variables, functions and indices. Standard bold letters $(\mathbf{a}, \mathbf{b}, ...)$ denote vectors and sans-serif font $(\mathsf{A}, \mathsf{B}, ..., \mathsf{a}, \mathsf{b}, ...)$ denotes constants.

Formulation Concepts. Extending the *tracking-by-detection* framework to multi-camera, multi-target tracking and localization problem consists of two sub-problems: *spatio-temporal data association* and *3D trajectory estimation* (see Figure 2.1). Although a data association problem in single camera aims to link detections from the same object temporally, data association in multiple cameras includes spatial data association in addition to (temporal) data association in a single camera. The data association problem in multiple cameras called *spatio-temporal data association* is a process of finding identity labels (IDs) of all detections. Given detections with the same label, a *3D tra-*

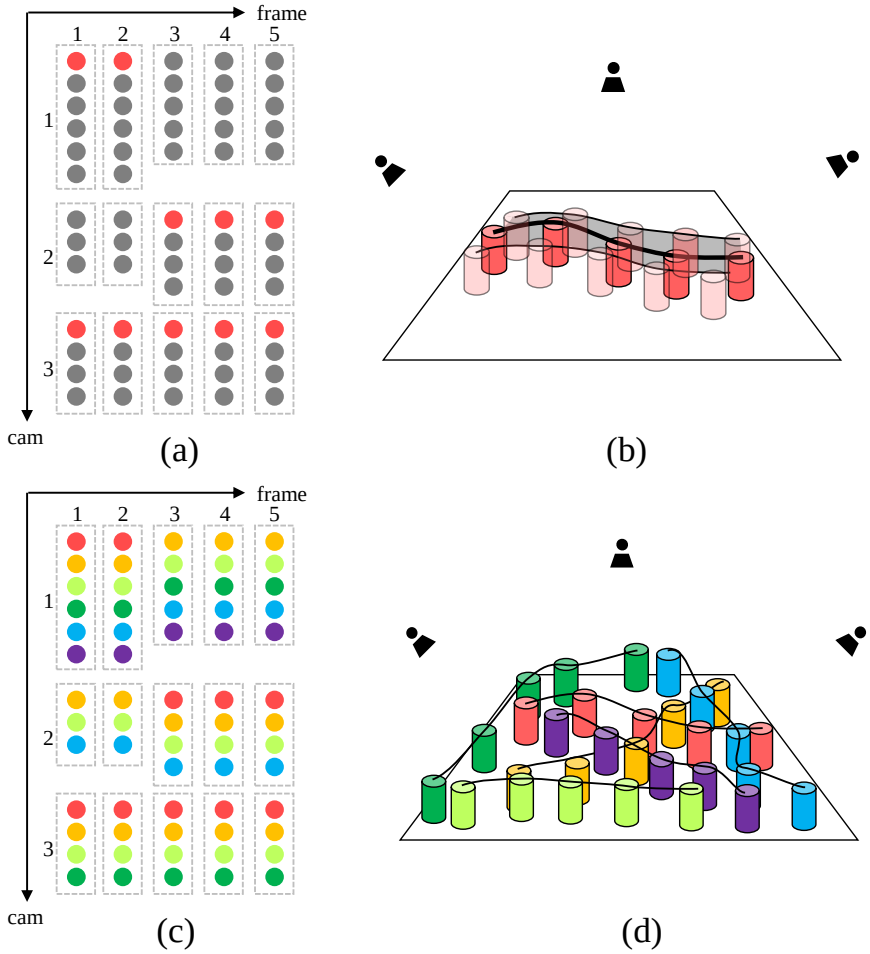


Figure 2.1: Example of data association problem and 3D trajectory estimation problem in multi-target tracking in multi-camera. In data association problem, (a) for every cameras and frames, each node denotes a detection and has their label (ID). (b) 3D trajectory estimation problem is to estimate 3D trajectories with the detections which have the same label. (c) The final labels of all detections and (d) their 3D trajectories are determined by the proposed optimization framework.

Table 2.1: Notations. Each block summarize the notation related to constants, detections, assignments, and trajectory hypotheses, respectively

Symbol	Description
F	length of sequence in frames
K	number of cameras
P	number of targets
N	number of trajectory hypotheses
M_{kt}	number of detections in camera k and frame t
$\mathbf{I}_{kt}$	set of detection indices in camera k and frame t
$\mathbf{D}$	set of all detections
$\mathbf{D}_{kt}$	set of detections in camera k and frame t
$\mathbf{d}_i^{k,t}$	the i-th detection in camera k and frame t
$\mathbf{A}$	set of assignment matrices
A_p	the p-th assignment matrix
A_p^t	the t-th column of the p-th assignment matrix
$\mathcal{X}$	set of all trajectory hypotheses
$\mathbf{x}_n$	the n-th trajectory hypothesis
$\mathbf{x}_n^t$	the n-th trajectory hypothesis at frame t
s_n,e_n	start and end frame indices of $\mathbf{x}_n$

jectory estimation problem is defined as the problem of estimating locations of objects in a 3D Euclidean space. In contrast to the previous literature [34, 35, 36, 38], the objective of our work is to solve the *spatio-temporal data association* and *3D trajectory estimation* problem at the same time.

Optimization variables. Mathematically, we introduce two types of variables: an assignment (matrix) A and a trajectory hypothesis (vector) $\mathbf{x}$. An assignment matrix is for the *spatio-temporal data association* problem and a trajectory hypothesis is for the *3D trajectory estimation* problem. First of all, we introduce details of each variable.

Assignments. An assignment A indicates detection indices corresponding a target,

which is represented as a matrix form,

$$A = \mathbb{R}^{\mathsf{K}\times\mathsf{F}},\ [A]_{k,t} = i,\ i \in \mathbf{I}_{kt}, \tag{2.1}$$

where each entry of a matrix A is determined by the following augmented index set,

$$\mathbf{I}_{kt} = \{0, 1, 2, ..., \mathsf{M}_{kt}\}. \tag{2.2}$$

Here, a dummy index 0 represents a missing or invisible detection and M_{kt} denotes the number of detections at the t-th frame of the k-th camera. We treat a trajectory with length 1 as a false positive. With the dummy index 0, we can express false positive, frame jump, and missing or invisible detections. In $\mathsf{K} = 3$, $\mathsf{F} = 5$ case, examples of assignments are as the following matrices:

$$\begin{pmatrix} 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 2 & 3 & 1 & 0 \end{pmatrix} : \text{Starts at frame 2 and ends at frame 4}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} : \text{False positive (trajectory length = 1)}$$

$$\begin{pmatrix} 1 & 2 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 3 & 3 & 2 \end{pmatrix} : \text{Missing detections at cam 2}$$

$$\begin{pmatrix} 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 & 2 \\ 1 & 2 & 0 & 3 & 2 \end{pmatrix} : \text{Missing detections at frame 3.}$$

Trajectory hypotheses. A trajectory hypothesis $\mathbf{x}$ represents a sequence of 3D points of a target. We denote 3D location $\mathbf{x}^t$ at the t-frame as (x^t, y^t, z^t). Then, the trajectory hypothesis $\mathbf{x}$ is denoted by a column vector concatenating the 3D points,

$$\mathbf{x} = (\mathbf{x}^s\ ...\ \mathbf{x}^e)^{\mathsf{T}} \in \mathbb{R}^{3(e-s+1)\times 1}, \tag{2.3}$$

where s and e are the start and the end frame indices, respectively.

Optimization Formulation. Following the *tracking-by-detection* framework, we construct a KF-partite hypergraph using detections from all cameras and frames as nodes.

A set of detections from the k-th camera and the t-th frame is denoted by $\mathbf{D}_{kt}$ whose element $\mathbf{d}_i^{k,t} \in \mathbf{D}_{kt}$ represents 2D bounding box for the i-th detection at the k-th camera and the t-th frame. A hyperedge in the KF-partite graph includes at most one detection in each partite set. The KF-partite graph is defined by

$$G = (\mathbf{V}, \mathbf{E}) = (\mathbf{D}_{11} \cup ... \cup \mathbf{D}_{\mathsf{KF}}, \mathbf{E}), \tag{2.4}$$

where $\mathbf{E}$ denotes a set of all possible hyperedges.

Since an assignment matrix represents a hyperedge, all possible assignment matrices is the same as the hyperedge set $\mathbf{E}$. Given a KF-partite graph, we define a trajectory hypothesis set $\mathcal{X}$ for the *3D trajectory estimation*. Letting N be the number of trajectory hypotheses, a trajectory hypothesis set is defined as

$$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{\mathsf{N}}\}. \tag{2.5}$$

Note that the number N equals to the number of hyperedges $|\mathbf{E}|$. Each trajectory hypothesis $\mathbf{x} \in \mathcal{X}$ represents 3D locations of the corresponding assignment in $\mathbf{E}$. Then, the *3D trajectory estimation* can be seen as the problem of updating trajectory hypothesis vectors in $\mathcal{X}$. On the other hand, the *spatio-temporal data association* can be formulated as finding a set of assignment matrices under satisfying two constraints: *non-overlap* and *union* constraints. Letting P be the unknown number of targets, the set of assignment matrices is defined by

$$\mathbf{A} = \{A_1, A_2, ..., A_{\mathsf{P}}\}, \tag{2.6}$$

where each assignment matrix $A_p \in \mathbf{A}$ does not share a detection with other assignment matrices and assignment matrices in $\mathbf{A}$ include all indices of detections. From the definition of $\mathbf{A}$ and $\mathcal{X}$, the final 3D trajectories are derived as,

$$\{\mathbf{x}_{n_1}, ..., \mathbf{x}_{n_p}, ..., \mathbf{x}_{n_{\mathsf{P}}}\} \subset \mathcal{X}, \tag{2.7}$$

where n_p denotes the trajectory hypothesis' index corresponding to the p-th assignment A_p.

The problem of finding $\mathbf{A}$ and $\mathcal{X}$ with a minimum cost can be formulated as the following mixed multidimensional assignment (MMDA) problem:

$$\min_{\mathbf{A},\mathcal{X}} \sum_{p=1}^{\mathsf{P}} c(A_p, \mathcal{X}) \tag{2.8}$$

subject to

$$[A_u]_{k,t} \neq [A_v]_{k,t},\ \forall u \neq v, \qquad \text{s.t. } [A_u]_{k,t}, [A_v]_{k,t} > 0, \tag{2.9}$$

$$\exists A_u \in \mathbf{A},\ \ \forall i \in \mathbf{I}_{kt} \backslash \{0\} \qquad \text{s.t. } [A_u]_{k,t} = i, \tag{2.10}$$

$$k = 1, ..., \mathsf{K},\ t = 1, ..., \mathsf{F},$$

where $c(A_p, \mathcal{X})$ denotes the cost function of the p-th object (the cost function will be defined in Section 2.1.2). The optimization problem in (2.8) is challenging because of the following three reasons. First, two different types of optimization variables are mixed; $\mathbf{A}$ in discrete domain, $\mathcal{X}$ in continuous domain. Next, the cost function is not even convex or submodular. Lastly, a feasible set satisfying the *non-overlap* and *union* constraints in (2.9) and (2.10) respectively, is defined in combinatorial solution space. To deal with this kind of problem, we apply an alternative optimization strategy where only one optimization variable is optimized by fixing the other variable and then the procedure is repeated by changing the optimizing variable and fixed variable to each other alternatively. In optimizing $\mathbf{A}$, to deal with the combinatorial solution space, we propose an iterative algorithm that efficiently finds local optimum in a random search manner. In optimizing $\mathcal{X}$ at continuous domain, the solution is given by a closed-form via least squared formulation. The details for the optimization will be addressed in Section 2.2.

2.1.2 Cost Design

In this subsection, we present a cost design for our formulation, which considers 3D reconstruction accuracy, motion smoothness, and penalty terms (see Figure 2.2). The penalty terms enforce the trajectories to start and end at the entrance/exit zone, and

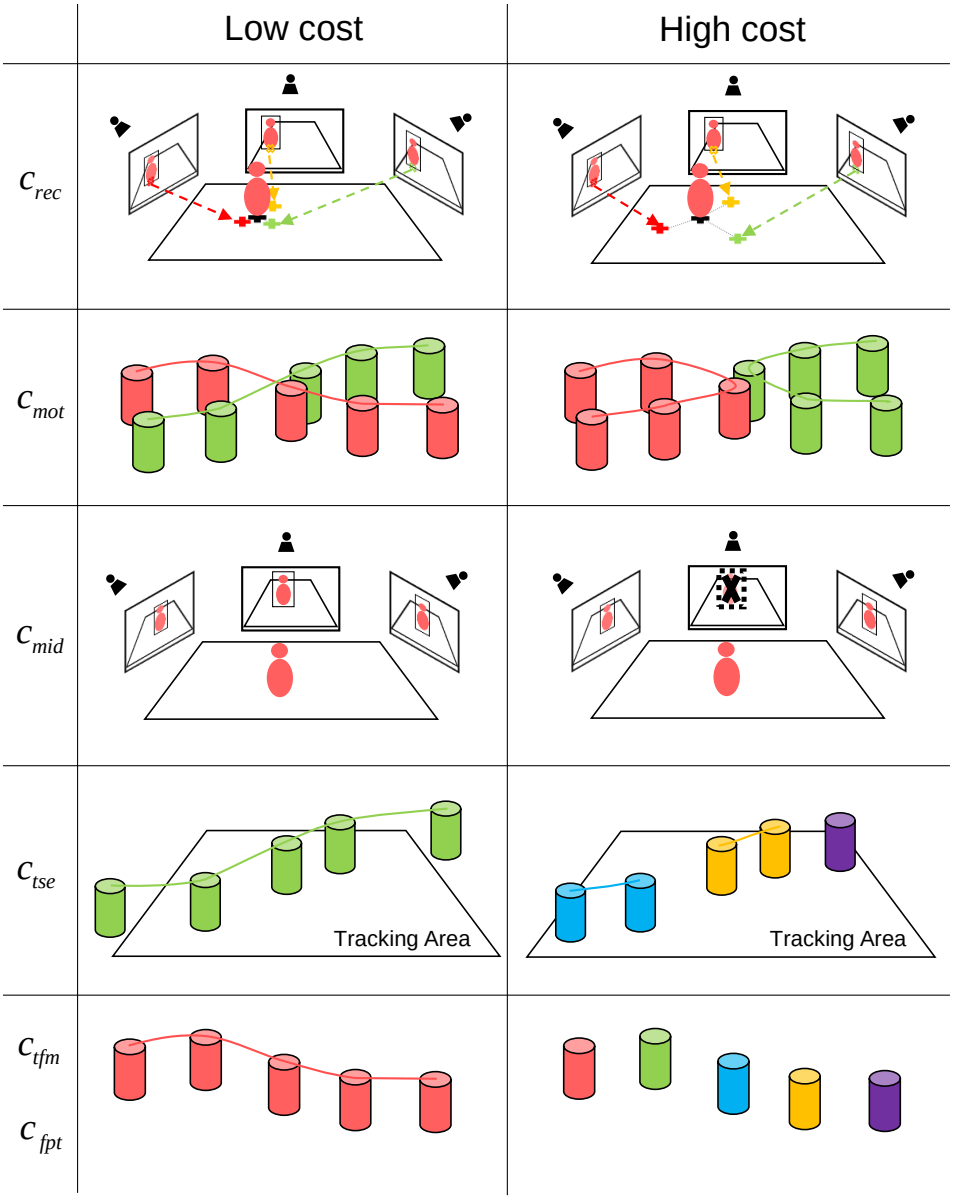


Figure 2.2: The physical properties of different terms of the cost function. The left column shows an example with a lower cost value, the right column with a higher cost value for each individual term.

prevent the trajectories from having too many missing detections or a short length. A cost function $c(A_p, \mathcal{X})$ is a function that maps an A_p and $\mathcal{X}$ to a real value. The cost function can be rewritten as $\widetilde{c}(\cdot)$ by introducing the reconstruction function $r(A_p)$ that maps an assignment matrix A_p to a trajectory hypothesis $\mathbf{x}_n$ as follows:

$$c(A_p, \mathcal{X}) = \widetilde{c}(A_p, r(A_p)) = \widetilde{c}(A_p, \mathbf{x}_{n_p}), \tag{2.11}$$

where n_p is an index corresponding to the p-th assignment matrix among the indices in $\mathcal{X}$. For simplicity of notations, we omit the index n_p and p in the following.

Our cost function is a summation of six individual terms: cost for 3D reconstruction accuracy (c_{rec}), cost for motion smoothness (c_{mot}), cost for missing detections (c_{mid}), cost for starting/ending zone of a trajectory (c_{tse}), cost for trajectory fragments (c_{tfm}) and cost for false positives (c_{fpt}) defined by,

$$\begin{aligned}\widetilde{c}(A, \mathbf{x}) = \lambda_{rec} \cdot c_{rec} + \lambda_{mot} \cdot c_{mot} + \lambda_{mid} \cdot c_{mid} \\ + \lambda_{tse} \cdot c_{tse} + \lambda_{tfm} \cdot c_{tfm} + \lambda_{fpt} \cdot c_{fpt},\end{aligned} \tag{2.12}$$

where λ indicates the weighting parameter of each cost term.

Cost for 3D Reconstruction Accuracy. At each frame t, the first term c_{rec} measures 3D reconstruction error obtained from a 3D point $\mathbf{x}^t$ and 3D back-projection lines of assigned detections at each camera. The cost value increases proportionally to the Euclidean distance between a 3D point and a 3D back-projection line of a detection in each camera. The c_{rec} is defined as a summation of the average of 3D reconstruction errors ε_{rec} over the entire frames. At each frame, 3D reconstruction error ε_{rec} is defined in the sense of mean squared error (MSE). The error $\varepsilon_{rec}(A, \mathbf{x}, k, t)$ at frame t and camera k is obtained by the distance between 3D point $\mathbf{x}^t$ and 3D back-projection line from the detection $\mathbf{d}_i^{k,t}$ where $i = [A]_{k,t}$. To prevent that a missed detection has a zero error, we set a default error term r when a person is visible but not detected. Letting $\mathbf{N}(\mathbf{x}^t)$ be the index set of visible cameras at a 3D location $\mathbf{x}^t$, the cost for the 3D

reconstruction accuracy is given by

$$c_{rec}(A, \mathbf{x}) = \sum_{t=s}^{e} \sum_{k \in \mathbf{N}(\mathbf{x}^t)} \frac{\varepsilon_{rec}(A^t, \mathbf{x}^t, k)^2}{|\mathbf{N}(\mathbf{x}^t)|}, \tag{2.13}$$

$$\varepsilon_{rec}(A^t, \mathbf{x}^t, k) = \tag{2.14}$$

$$\begin{cases} dist(\Phi^k(\mathbf{d}_i^{k,t}), \mathbf{x}^t), & \text{if } [A^t]_k = i,\ i > 0, \\ \mathsf{r}, & \text{if } [A^t]_k = 0,\ k \in \mathbf{N}(\mathbf{x}^t), \\ 0, & otherwise, \end{cases}$$

where $\Phi^k(\mathbf{d})$ indicates the back-projection line of a detection $\mathbf{d}$ at the k-th camera. The distance function is modeled as a form of linear equation and thus can be solved in a closed-form.

Remark 1 *Let a back-projection line* $\Phi^k(\mathbf{d})$ *be* $\frac{x-\mathsf{c}}{\mathsf{a}} = \frac{y-\mathsf{d}}{\mathsf{b}} = z$*, where the constants* $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}$ *are constant parameters to determine the back-projection line* $\Phi^k(\mathbf{d})$*. The distance function is defined by Euclidean distance between the 3D line and a 3D point* $\mathbf{x}^t = (x^t, y^t, z^t)$ *at the* $z = z^t$*. The point at* $z = z^t$ *in the back-projection line is* $(\mathsf{a}z^t + \mathsf{c},\ \mathsf{b}z^t + \mathsf{d},\ z^t)$*. Finally, the distance function* $dist(\Phi^k(\mathbf{d}), \mathbf{x}^t)$ *in (2.14) can be rewritten by* $\left\|P\mathbf{x}^t - \mathbf{q}\right\|$ *where* $P = \begin{pmatrix} -1 & 0 & \mathsf{a} \\ 0 & -1 & \mathsf{b} \\ 0 & 0 & 0 \end{pmatrix}$, $\mathbf{q} = (\mathsf{c}, \mathsf{d}, 0)^\mathsf{T}$.

Cost for Motion Smoothness. The cost for motion smoothness measures how much a trajectory is inconsistent with the natural motion of a person. It is assumed that a person moves through the shortest path, and in most cases also smooth path. For this purpose, we adopt a spline-based cost function for a motion model considering motion curvature term ε_c as well as average distance ε_d:

$$c_{mot}(A, \mathbf{x}) = \alpha_m \cdot \varepsilon_d + (1 - \alpha_m) \cdot \varepsilon_c, \tag{2.15}$$

$$\varepsilon_d = \sum_{t=s+1}^{e} w_d(t) \cdot \left\|\mathbf{x}^t - \mathbf{x}^{t-1}\right\|^2, \tag{2.16}$$

$$\varepsilon_c = \sum_{t=s+1}^{e-1} w_c(t) \cdot \left\|\mathbf{x}^{t+1} - 2 \cdot \mathbf{x}^t + \mathbf{x}^{t-1}\right\|^2. \tag{2.17}$$

There exists an adjusting weight α_m that controls the trade-off between ε_c and ε_d. Note that at each frame t, each term is weighted by the number of average detections over the consecutive frames. Denoting $d(A^t)$ for the number of detections for the assignment A at the t-th frame, their weights are $w_d(t) = \frac{d(A^t)+d(A^{t-1})}{2}$ and $w_c(t) = \frac{d(A^{t+1})+d(A^t)+d(A^{t-1})}{3}$ respectively. The spline-based motion model has been used in single camera approaches [16, 18], but we have extended it to a multiple camera case by weighting the average numbers of detections.

Cost for Missing Detections. The cost for missing detections is designed to penalize the case that a person is visible at the established cameras. The cost for missing detections is proportional to the number of missed detections among the number of visible cameras. The penalty is given as

$$c_{mid}(A, \mathbf{x}) = \sum_{t=s}^{e} (|\mathbf{N}(\mathbf{x}^t)| - d(A^t)), \tag{2.18}$$

where the number of visible cameras, $|\mathbf{N}(\mathbf{x}^t)|$, is always greater than or equal to the number of detections $d(A^t)$.

Cost for Starting/Ending Zone Violation. A trajectory is enforced to start and end at entrance/exit zone, respectively. If a trajectory starts or ends at out of the entrance/exit zone, we give a penalty in the cost function proportional to the number of detection, that is,

$$c_{tse}(A, \mathbf{x}) = d(A^s) \cdot e(\mathbf{x}^s) + d(A^e) \cdot e(\mathbf{x}^e), \tag{2.19}$$

where a function $e(\mathbf{x}^t)$ is a indicator function that has 1 when $\mathbf{x}^t$ is out of the entrance/exit zone.

Cost for Trajectory Fragments. This cost is to prevent a trivial solution that has many trajectory fragments rather than connects them. Similar to [13], we give a penalty to the trajectory whenever it starts or ends, which means that trajectory fragments increases the cost. The cost is proportional to the number of detection at start/end time,

$$c_{tfm}(A) = d(A^s) + d(A^e). \tag{2.20}$$

Cost for False Positive Trajectory. The cost for false positive trajectories prevents a trivial solution that all detection are considered as false positives. We penalize a false positive trajectory which consists of detections at the frame where a trajectory starts and ends at the same time (i.e., trajectory length is 1). The cost can be defined as

$$c_{fpt}(A) = \begin{cases} d(A^s), & \text{if } s = e, \\ 0, & \text{otherwise.} \end{cases} \tag{2.21}$$

2.2 Optimization

It is clear that the cost terms described in Section 2.1.2 are defined on both discrete and continuous domain where an assignment variable A is in discrete space and trajectory hypothesis $\mathbf{x}$ is in continuous space. Unfortunately, it is not trivial to simultaneously optimize two types of variables $\mathbf{A}$ and $\mathcal{X}$ in (2.8). In this paper, we adopt an alternative optimization framework, which optimizes one set of variables under fixing the other set of variables. Starting from initial solution $\mathbf{A}^0$ and $\mathcal{X}^0$, we first find a locally optimal assignment set $\mathbf{A}$ (i.e., *spatio-temporal data association* problem). Next, fixing the locally optimal assignment set $\mathbf{A}$, a locally optimal trajectory hypotheses set $\mathcal{X}$ is found (i.e., trajectory estimation problem). This alternative procedure is repeated. To summarize, starting from the previous solution $\mathbf{A}^{(iter-1)}$ and $\mathcal{X}^{(iter-1)}$, the proposed optimization framework alternates solving the following two objective functions:

$$\begin{cases} \mathbf{A}^{(iter)} = \arg\min_{\mathbf{A}} \sum_p c(A_p, \mathcal{X}^{(iter-1)}), & \text{(2.22a)} \\ \mathcal{X}^{(iter)} = \arg\min_{\mathcal{X}} \sum_p c(A_p^{(iter)}, \mathcal{X}). & \text{(2.22b)} \end{cases}$$

First, given trajectory hypotheses set $\mathcal{X}^{(iter-1)}$, the problem of finding an assignment set with the minimum cost (2.22a) is the multidimensional assignment problem (MDA), which is an NP-hard problem when $\mathsf{K} \geq 3$ or $\mathsf{F} \geq 3$ [42]. The MDA problem for data association has been widely studied in multi-target tracking [16] and multi-sensor fusion problem [42, 44]. It can be solved by using an approximate method such

as greedy, branch and bound techniques, the Lagrangian relaxation methods [42, 44], and an iterative algorithm [16]. Unlike the previous works [42, 44, 16], we consider both multi-target tracking and multi-sensor fusion problem simultaneously. The MDA problem is formulated for *sptio-temporal data association* problem that links detections across both multiple cameras and frames. Furthermore, we present an iterative algorithm that efficiently solves the *sptio-temporal data association* problem by a random search manner (Details are given in Section 2.2.1). In (2.22b), fixing assignment set $\mathbf{A}$, the problem for finding 3D trajectories that minimizes the designed cost terms, can be formulated as minimizing least squared errors of each trajectory hypothesis $\mathbf{x} \in \mathcal{X}$ and their detections (Details in Section 2.2.2). The complete algorithm is summarized in Algorithm 1. The following subsections describe the detail of each parts.

Algorithm 1 An Overall Optimization Framework

Input: $\mathbf{A}^0$, $\mathcal{X}^0$, max_iter, max_loop

Output: $\mathbf{A}^*$, $\mathbf{x}^*_{n_1}, ..., \mathbf{x}^*_{n_p}, ..., \mathbf{x}^*_{n_{\mathsf{P}^*}}$

1: **for** $iter \leftarrow 1, \ldots, max\_iter$ **do**

2: $\mathbf{A}^{(iter)} = ISR(\mathbf{A}^{(iter-1)}, \mathcal{X}^{(iter-1)}, max\_loop)$

3: $\mathcal{X}^{(iter)} = THU(\mathbf{A}^{(iter)}, \mathcal{X}^{(iter-1)})$
end for

4: $\mathbf{A}^* = \mathbf{A}^{(max\_iter)}$

5: $\mathbf{x}^*_{n_p} \leftarrow r(A^*_p), \ \ \forall A^*_p \in \mathbf{A}^*$

2.2.1 Spatio-temporal Data Association

Splitting/re-merging. Starting from the assignment set $\mathbf{A}^{(iter-1)}$, a new assignment set $\mathbf{A}^{(iter)}$ is calculated by the proposed iterative splitting/re-merging algorithm to be described in this section. The initial solution of splitting/re-merging $\widetilde{\mathbf{A}}^{(0)}$ is set to

$\mathbf{A}^{(iter-1)}$.[1] The key idea of the splitting/re-merging algorithm is that we randomly split the given assignment set $\widetilde{\mathbf{A}}^{(l-1)}$ into two split assignment sets $\widetilde{\mathbf{A}}^{\mathsf{I}}$ and $\widetilde{\mathbf{A}}^{\mathsf{J}}$, then optimally re-merge the two split assignment sets to obtain the new solution $\widetilde{\mathbf{A}}^{(l)}$ as a result. Our splitting/re-merging strategy is designed to find the solution better than or equal to the previous one. The final assignment set $\widetilde{\mathbf{A}}^{(max\_loop)}$ is returned after max_loop times of iterations.

Splitting. To split an assignment set into two random assignment sets, we introduce a pair of binary matrices called a pair of Random Split Masks (RSMs) satisfying

$$\begin{aligned} &M^{\mathsf{I}} + M^{\mathsf{J}} = \mathbb{1}^{\mathsf{K}\times\mathsf{F}}, \\ &[M^{\mathsf{I}}]_{i,j}, [M^{\mathsf{J}}]_{i,j} \in \{0,1\},\ i = 1,...,\mathsf{K},\ j = 1,...,\mathsf{F}, \end{aligned} \tag{2.23}$$

where all entries of the matrix $\mathbb{1}^{\mathsf{K}\times\mathsf{F}}$ have 1. By choosing a pair of RSMs M^{I} and M^{J}, we split the previous assignment set $\widetilde{\mathbf{A}}^{(l-1)}$ into two groups $\widetilde{\mathbf{A}}^{\mathsf{I}}$ and $\widetilde{\mathbf{A}}^{\mathsf{J}}$.

Remark 2 *A pair of RSMs M^{I} and M^{J} are determined by three components in our experiments: splitting frame* t*, a set of splitting cameras* $\mathbf{K}$ *and RSM type* c*. At each splitting frame* t*, we determine a pair of RSMs by randomly choosing a set of splitting cameras* $\mathbf{K}$ *and RSM type* c*. Thus, a pair of RSMs are functions of* t*,* $\mathbf{K}$*, and* c*. The details of each component is described below.*

- *The splitting frame* t *is a reference frame where the* t*-th columns of assignment matrices are mainly changed. During the iteration l, we sequentially choose the splitting frame, i.e., from frame 1 to* F.

- *After choosing the splitting frame* t*, a set of splitting cameras* $\mathbf{K}$ *is randomly chosen in powerset of* $\{1,2,...,\mathsf{K}\}$ *excluding empty set and universal set, that is,*

$$\mathbf{K} \in 2^{\{1,2,...,\mathsf{K}\}} \setminus \{\{\emptyset\}, \{1,2,...,\mathsf{K}\}\}. \tag{2.24}$$

[1] To distinguish the notation of inner iterations from that of outer iterations, $\widetilde{\mathbf{A}}^{(l)}$ denotes the assignment set at the l-th inner iteration.

- *Given splitting frame* t *and set of splitting cameras* $\mathbf{K}$*, two RSM types are defined: reconstruction and track type. RSM type determines the remaining columns of* M^{I} *and* M^{J} *except the* t*-th column. The RSM type is illustrated as follows. In* $\mathsf{K}=3$*,* $\mathsf{F}=5$ *case, examples of a pair of RSMs* M^{I} *and* M^{J} *are as the following matrices:*

$$\mathsf{t}=3,\ \mathbf{K}=\{1,2\},\ \mathsf{c}=\textit{reconstruction type},$$
$$M^{\mathsf{I}}=\begin{pmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&0&1&1\end{pmatrix},\ M^{\mathsf{J}}=\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&1&0&0\end{pmatrix},$$
$$\mathsf{t}=3,\ \mathbf{K}=\{1,2\},\ \mathsf{c}=\textit{track type},$$
$$M^{\mathsf{I}}=\begin{pmatrix}1&1&1&0&0\\1&1&1&0&0\\1&1&0&0&0\end{pmatrix},\ M^{\mathsf{J}}=\begin{pmatrix}0&0&0&1&1\\0&0&0&1&1\\0&0&1&1&1\end{pmatrix}.$$

At each splitting frame $\mathbf{t}$*, we select a pair of RSMs in a random permutation of all pairs of RSMs with different* $\mathbf{K}$ *and RSM type . Thus, the number of all pairs of RSMs is given by* $\mathsf{F}\times(2^{\mathsf{K}}-2)\times 2$*.*

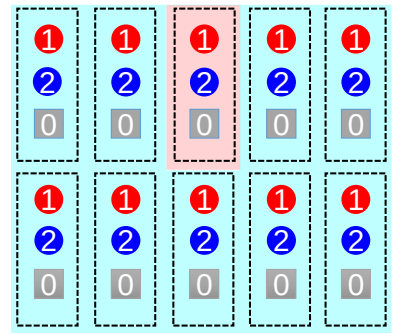
Each assignment matrix $\tilde{A}_p \in \widetilde{\mathbf{A}}^{(l-1)}$ is split into two assignment matrices $\tilde{A}_p^{\mathsf{I}}$ and $\tilde{A}_p^{\mathsf{J}}$:

$$\tilde{A}_p^{\mathsf{I}} = M^{\mathsf{I}} \otimes \tilde{A}_p,\ \ \tilde{A}_p^{\mathsf{J}} = M^{\mathsf{J}} \otimes \tilde{A}_p, \tag{2.25}$$

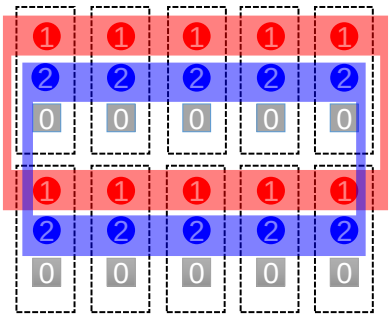
where $\otimes$ is the Hadamard (or entrywise) product. Multiplying M^{I} and M^{J} to all assignment matrices in $\widetilde{\mathbf{A}}^{(l-1)}$, we finally obtain two assignment sets as follows:

$$\widetilde{\mathbf{A}}^{(l-1)} = \{\tilde{A}_p\},\quad p=1,...,\mathsf{P},$$
$$\widetilde{\mathbf{A}}^{\mathsf{I}} = \{M^{\mathsf{I}} \otimes \tilde{A}_p\}\backslash\{O_{\mathsf{K}\times\mathsf{F}}\},$$
$$\widetilde{\mathbf{A}}^{\mathsf{J}} = \{M^{\mathsf{J}} \otimes \tilde{A}_p\}\backslash\{O_{\mathsf{K}\times\mathsf{F}}\},$$
$$|\widetilde{\mathbf{A}}^{\mathsf{I}}| = \mathsf{P}^{\mathsf{I}},\ |\widetilde{\mathbf{A}}^{\mathsf{J}}| = \mathsf{P}^{\mathsf{J}},$$

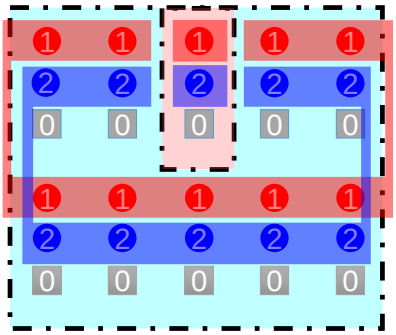
where P^{I}, P^{J} denote the number of elements of each split assignment set respectively and t denotes the *splitting frame* of given pair of RSMs. Note that both P^{I} and P^{J}



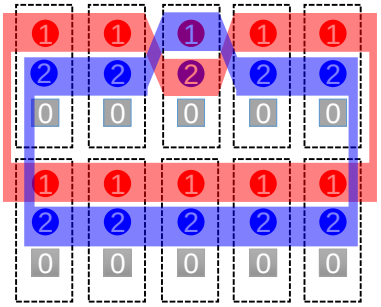
(a) $M^{\mathsf{I}}, M^{\mathsf{J}}$



(b) $\widetilde{\mathbf{A}}^{(l-1)}$



(c) $\widetilde{\mathbf{A}}^{\mathsf{I}}$,$\widetilde{\mathbf{A}}^{\mathsf{J}}$



(d) $\widetilde{\mathbf{A}}^{(l)}$

Figure 2.3: An example of a splitting/re-merging in 5 frames and 2 cameras. (a) Random split masks (M^{I}: cyan, M^{J}: light red). (b) Previous assignment set $\widetilde{\mathbf{A}}^{(l-1)} = \{A_1, A_2\}$ (A_1: red, A_2: blue). (c) Split assignment sets $\widetilde{\mathbf{A}}^{\mathsf{I}} = \{\tilde{A}_1^{\mathsf{I}}, \tilde{A}_2^{\mathsf{I}}\}$, $\widetilde{\mathbf{A}}^{\mathsf{J}} = \{\tilde{A}_1^{\mathsf{J}}, \tilde{A}_2^{\mathsf{J}}\}$ ($\tilde{A}_1^{\mathsf{I}}, \tilde{A}_1^{\mathsf{J}}$: red, $\tilde{A}_2^{\mathsf{I}}, \tilde{A}_2^{\mathsf{J}}$: blue). (d) New assignment set $\widetilde{\mathbf{A}}^{(l)} = \{\tilde{A}_1^{\mathsf{I}} + \tilde{A}_2^{\mathsf{J}}, \tilde{A}_2^{\mathsf{I}} + \tilde{A}_1^{\mathsf{J}}\}$ ($\tilde{A}_1^{\mathsf{I}} + \tilde{A}_2^{\mathsf{J}}$: red, $\tilde{A}_2^{\mathsf{I}} + \tilde{A}_1^{\mathsf{J}}$: blue).

are less than or equal to P because split assignments might be all-zero column at the *splitting frame* t.

Re-merging. After the random splitting operation, the two split assignment sets are re-merged so that the re-merged assignment set has a cost smaller than or equal to the previous cost. For each $\tilde{A}_i^{\mathsf{I}} \in \widetilde{\mathbf{A}}^{\mathsf{I}}, \tilde{A}_j^{\mathsf{J}} \in \widetilde{\mathbf{A}}^{\mathsf{J}}$, the merged assignment set is obtained by the summation of two matrices, i.e., $A_{<i,j>} = A_i^{\mathsf{I}} + A_j^{\mathsf{J}}$. The new merging pairs $\tilde{A}_i^{\mathsf{I}}, \tilde{A}_j^{\mathsf{J}}$ are determined so that they have the minimal cost among all possible merging combinations between $\widetilde{\mathbf{A}}^{\mathsf{I}}$ and $\widetilde{\mathbf{A}}^{\mathsf{J}}$. Let ψ_{ij} indicate a binary variable for a merging pair, which is set to 1 if $\tilde{A}_i^{\mathsf{I}}$ and $\tilde{A}_j^{\mathsf{J}}$ are selected for re-merging; otherwise it is assigned to 0. With binary variables ψ, the re-merging process can be formulated as the following equation:

$$\min_{\psi} \sum_{i=0}^{\mathsf{P}^{\mathsf{I}}} \sum_{j=0}^{\mathsf{P}^{\mathsf{J}}} c_{ij}\psi_{ij} \tag{2.26}$$

subject to

$$\sum_{j=0}^{\mathsf{P}^{\mathsf{J}}} \psi_{ij} = 1; \quad i = 1, ..., \mathsf{P}^{\mathsf{I}},$$
$$\sum_{i=0}^{\mathsf{P}^{\mathsf{I}}} \psi_{ij} = 1; \quad j = 1, ..., \mathsf{P}^{\mathsf{J}},$$

where $c_{ij} = c(\tilde{A}_i^{\mathsf{I}} + \tilde{A}_j^{\mathsf{J}}, \mathcal{X})$ means a cost of re-merged two assignments $\tilde{A}_i^{\mathsf{I}}$ and $\tilde{A}_j^{\mathsf{J}}$. The problem can be solved exactly in polynomial time by the Kuhn-Munkres Hungarian algorithm [45]. The re-merged assignment set $\widetilde{\mathbf{A}}^{(l)}$ is obtained by the optimal ${\psi_{ij}}^*$ of the two-dimensional assignment problem as follows:

$$\widetilde{\mathbf{A}}^{(l)} = \{A_{<i,j>}\}, \forall i, j \text{ satisfy } {\psi_{ij}}^* = 1, \tag{2.27}$$

$$A_{<i,j>} = \begin{cases} \tilde{A}_i^{\mathsf{I}} + \tilde{A}_j^{\mathsf{J}} & i, j \geq 1, \\ \tilde{A}_i^{\mathsf{I}} & i \geq 1, j = 0, \\ \tilde{A}_j^{\mathsf{J}} & i = 0, j \geq 1. \end{cases} \tag{2.28}$$

Finally, the new assignment set $\widetilde{\mathbf{A}}^{(l)}$ at the l-th iteration is a set of re-merged assignment between $\widetilde{\mathbf{A}}^{\mathsf{I}}$ and $\widetilde{\mathbf{A}}^{\mathsf{J}}$. Next, we repeat this splitting/re-merging operation until the predefined maximum number of iterations.

Example of a splitting/re-merging. Consider two people tracked through five frames and two cameras. Assume that we have a feasible assignment set at the $(l-1)$-th iteration $\widetilde{\mathbf{A}}^{(l-1)} = \{A_1, A_2\}$ with cost $\mathsf{C}^{(l-1)} = c(A_1, \mathcal{X}) + c(A_2, \mathcal{X})$ (see Figure 2.3b). First, let us assume that a pair of RSMs M^{I} and M^{J} are generated as the following matrices (see Figure 2.3a),

$$M^{\mathsf{I}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \; M^{\mathsf{J}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.29}$$

By the pair of RSM, the given assignment set $\widetilde{\mathbf{A}}^{(l-1)}$ is split into two groups $\widetilde{\mathbf{A}}^{\mathsf{I}} = \{\tilde{A}_1^{\mathsf{I}}, \tilde{A}_2^{\mathsf{I}}\}$, $\widetilde{\mathbf{A}}^{\mathsf{J}} = \{\tilde{A}_1^{\mathsf{J}}, \tilde{A}_2^{\mathsf{J}}\}$ (see Figure 2.3c). And their assignment matrices are

$$\tilde{A}_1^{\mathsf{I}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad \tilde{A}_2^{\mathsf{I}} = \begin{pmatrix} 2 & 2 & 0 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{pmatrix}, \tag{2.30}$$

$$\tilde{A}_1^{\mathsf{J}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{A}_2^{\mathsf{J}} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Next, the problem to find optimal merging pairs between the two assignment sets $\widetilde{\mathbf{A}}^{\mathsf{I}}$ and $\widetilde{\mathbf{A}}^{\mathsf{J}}$ can be formulated as the classical two-dimensional assignment problem [46] to minimize the sum of costs by utilizing the following cost matrix,

$$C = \begin{pmatrix} C_{\mathsf{IJ}} & C_{\mathsf{I0}} \\ C_{\mathsf{0J}} & C_{\mathsf{00}} \end{pmatrix}, \tag{2.31}$$

where the matrix C_{IJ} represent the costs of the re-merging assignments and C_{I0}, C_{0J}, and C_{00} are used for the case that assigns to nothing. Specifically, the $[C_{\mathsf{IJ}}]_{i,j}$ means the cost of the re-merging assignment $\tilde{A}_i^{\mathsf{I}} + \tilde{A}_j^{\mathsf{J}}$ given by

$$C_{\mathsf{IJ}} = \begin{pmatrix} c(\tilde{A}_1^{\mathsf{I}} + \tilde{A}_1^{\mathsf{J}}, \mathcal{X}) & c(\tilde{A}_1^{\mathsf{I}} + \tilde{A}_2^{\mathsf{J}}, \mathcal{X}) \\ c(\tilde{A}_2^{\mathsf{I}} + \tilde{A}_1^{\mathsf{J}}, \mathcal{X}) & c(\tilde{A}_2^{\mathsf{I}} + \tilde{A}_2^{\mathsf{J}}, \mathcal{X}) \end{pmatrix}, \tag{2.32}$$

The diagonal terms of matrix C_{I0} and C_{0J} is $c(\tilde{A}_i^{\mathsf{I}}, \mathcal{X})$ and $c(\tilde{A}_j^{\mathsf{J}}, \mathcal{X})$ respectively. C_{00} is an all-zero matrix and thus it makes the cost matrix C into a square matrix.

The two-dimensional assignment problem can be solved in a polynomial time by using the Hungarian algorithm [45]. If $c_{12} + c_{21} < c_{11} + c_{22}$, the merged assignment set $\widetilde{\mathbf{A}}^{(l)}$ with the minimal cost is determined by $\{A_{<1,2>}, A_{<2,1>}\}$ where $A_{<1,2>} = \tilde{A}_1^{\mathsf{I}} + \tilde{A}_2^{\mathsf{J}}$, $A_{<2,1>} = \tilde{A}_2^{\mathsf{I}} + \tilde{A}_1^{\mathsf{J}}$ (see Figure 2.3d). Note that the cost of $\widetilde{\mathbf{A}}^{(l)}$ is less than or equal to that of $\widetilde{\mathbf{A}}^{(l-1)}$, i.e., $\mathsf{C}^l = c_{12} + c_{21} \leq \mathsf{C}^{(l-1)} = c_{11} + c_{22}$. Therefore, the previous assignment set is maintained if $c_{12} + c_{21} > c_{11} + c_{22}$. Owing to the descent search strategy and the randomness of $M^{\mathsf{I}}, M^{\mathsf{J}}$ selected in each iteration, the resulting assignment set must converge to a local optimum as the iteration goes to infinity.

Perturbations. A splitting/re-merging operation is a random search guided by a pair of RSMs $M^{\mathsf{I}}, M^{\mathsf{J}}$, which finally leads to a lower cost. However, this descent only strategy might get stuck in a local basin far from the optimum. To escape from a local basin, we adopt a perturbation technique similar to most of random search method.

Our perturbation is a guided perturbation rather than a random perturbation because the proposed perturbation is performed under a specific condition. Among three components to determine a pair of RSMs $M^{\mathsf{I}}, M^{\mathsf{J}}$, the *splitting frame* $\mathtt{t}$ is a crucial parameter determining the condition for perturbation (see details of RSMs in Remark 2). Choosing a pair of RSMs with *splitting frame* $\mathtt{t}$ means that the $\mathtt{t}$-th column of assignment matrices is mainly changed. If all detections are missing at the $\mathtt{t}$-th frame, there is no information changed by a pair of RSM with *splitting frame* $\mathtt{t}$. Hence, the proposed splitting/re-merging with *splitting frame* $\mathtt{t}$ usually returns the original assignment matrix with the $\mathtt{t}$-th all-zero column. It implies that the solution is trapped in a local minimum. In this case, we perform a perturbation that the re-merging procedure is skipped after splitting the assignment matrix. This perturbation may increase the cost value but increase the possibility to escape from a local minimum as usual guided random search schemes

For this purpose, we modify re-merging cost c_{ij} in (2.26). Let a re-merging as-

signment matrix $A_{<i,j>}$ be a summation of split assignment matrix $\tilde{A}_i^{\mathsf{I}} + \tilde{A}_j^{\mathsf{J}}$. In the re-merging step with *splitting frame* $\mathtt{t}$, we set the re-merging cost to infinity if the $\mathtt{t}$-th column of re-merging assignment matrix $A_{<i,j>}$ has all-zero values as follows:

$$c_{ij} = \begin{cases} \text{Inf}, & \text{if } A^t_{<i,j>} = O_{\mathsf{K}\times 1}, \\ c(\tilde{A}_i^{\mathsf{I}} + \tilde{A}_j^{\mathsf{J}},\ \mathcal{X}), & otherwise. \end{cases} \tag{2.33}$$

Proposed iterative splitting/re-merging with perturbation is summarized in Algorithm 2. In Section 4.3.1, we show that the iterative splitting/re-merging with perturbation finally achieves a better solution than without perturbation.

Algorithm 2 An Iterative Splitting/Re-merging (ISR) with Perturbation for MDA

Input: $\mathbf{A}^{(iter-1)}$, $\mathcal{X}^{(iter-1)}$, max_loop

Output: $\mathbf{A}^{(iter)}$

Initialize $\widetilde{\mathbf{A}}^{(0)} \leftarrow \mathbf{A}^{(iter-1)}$
for $l \leftarrow 1, \ldots, max\_loop$ **do**
    Select Random Split Mask $M^{\mathsf{I}}, M^{\mathsf{J}}$
    for all $\tilde{A}_p \in \widetilde{\mathbf{A}}^{(l-1)}$ **do**
        if $M^{\mathsf{I}} \otimes \tilde{A}_p \neq O_{\mathsf{K}\times\mathsf{F}}$ **then**
            $\widetilde{\mathbf{A}}^{\mathsf{I}} \leftarrow \widetilde{\mathbf{A}}^{\mathsf{I}} \cup \{\tilde{A}_p^{\mathsf{I}}\},\ \tilde{A}_p^{\mathsf{I}} = M^{\mathsf{I}} \otimes \tilde{A}_p$
        end if
        if $M^{\mathsf{J}} \otimes A_p \neq O_{\mathsf{K}\times\mathsf{F}}$ **then**
            $\widetilde{\mathbf{A}}^{\mathsf{J}} \leftarrow \widetilde{\mathbf{A}}^{\mathsf{J}} \cup \{\tilde{A}_p^{\mathsf{J}}\},\ \tilde{A}_p^{\mathsf{J}} = M^{\mathsf{J}} \otimes \tilde{A}_p$
        end if
    end for
    Find optimal assignments ψ^*_{ij} by solving (2.26) with perturbation in (2.33)
    for all i, j satisfying $\psi^*_{ij} = 1$ **do**
        $\widetilde{\mathbf{A}}^{(l)} \leftarrow A_{<i,j>}$ calculated from (2.28)
    end for
end for
$\mathbf{A}^{(iter)} = \widetilde{\mathbf{A}}^{(max\_loop)}$

2.2.2 3D Trajectory Estimation

Recalling the original objective function in (2.8), the goal of 3D trajectory estimation is to find 3D locations of each target properly describing the six cost terms defined in Section 2.1.2. Given the assignment set $\mathbf{A}^{(iter)}$, the original objective function can be simplified to the one in (2.22b), which is the problem of finding trajectory hypotheses set $\mathcal{X}^{(iter)}$ with a minimum cost. Using (2.11) and (2.22b), we can rewrite the objective function for 3D trajectory estimation,

$$\min_{\mathbf{x}_{n_1},..,\mathbf{x}_{n_p},..,\mathbf{x}_{n_\mathsf{P}}} \sum_{p} \widetilde{c}(A_p^{(iter)}, \mathbf{x}_{n_p}), \tag{2.34}$$

where n_p denotes the index of trajectory hypothesis matched to the assignment matrix A_p. Each trajectory hypothesis is calculated by minimizing the new cost function $\widetilde{c}$ consists of six terms in (2.12). Except c_{tfm} and c_{fpt}, the cost terms c_{rec}, c_{mot}, c_{mid} and c_{tse} are affected by the trajectory hypothesis $\mathbf{x}$. However, it is difficult to directly minimize c_{mid} and c_{tse} w.r.t $\mathbf{x}$ since each derivative w.r.t $\mathbf{x}$ in not tractable because of the discrete valued functions in c_{mid} and c_{tse}. Instead, we adopt an alternative method that finds the new trajectory hypothesis $\mathbf{x}_{n_p}^*$ while fixing the values of the discrete functions and then updates the values of the discrete functions based on $\mathbf{x}_{n_p}^*$. The problem of finding $\mathbf{x}_{n_p}^*$ is given by solving the following objective function,

$$\min_{\mathbf{x_{n_p}}} \lambda_{rec} \cdot c_{rec}(A_p^{(iter)}, \mathbf{x}_{n_p}) + \lambda_{mot} \cdot c_{mot}(A_p^{(iter)}, \mathbf{x}_{n_p}), \tag{2.35}$$

which can be expressed by a weighted least-squares minimization problem and solved in a closed-form (see details in Section 2.4.1).

The final trajectories denoted $\{\mathbf{x}_{n_p}^* | p = 1, ..., \mathsf{P}\}$ are optimally estimated by solving (2.35). The trajectory hypothesis set $\mathcal{X}^{(iter)}$ should adequately reflect these final trajectories so that when we find $\mathbf{A}^{(iter+1)}$ at the next iteration, the selected trajectory hypotheses by splitting and re-merging are influenced by the optimally estimated trajectories $\{\mathbf{x}_{n_p}^*\}$. To do that, we perform an update for every trajectory hypotheses related to $\{\mathbf{x}_{n_p}^*\}$. This update does not change the sum of costs itself, but it helps to find

better assignment matrices $\mathbf{A}^{(iter+1)}$ at the next iteration. For each person p and frame t, we find a trajectory hypothesis set $\mathcal{X}_{[p,t]}$ for the update, which is a set of related trajectory hypotheses to $\mathbf{x}^*_{n_p}$ at frame t. To find $\mathcal{X}_{[p,t]}$, we define an assignment set $\mathbf{A}_{[p,t]}$, where the t-th column of each assignment matrix is the same as that of $A_p \in \mathbf{A}^{(iter)}$, that is,

$$\mathbf{A}_{[p,t]} = \{A |\ A^t = A^t_p\}. \tag{2.36}$$

$\mathcal{X}_{[p,t]}$ are a set of trajectory hypothesis corresponding to each assignment matrix in $\mathbf{A}_{[p,t]}$, which is given by

$$\mathcal{X}_{[p,t]} = \{\mathbf{x} |\ \mathbf{x} = r(A),\ \forall A \in \mathbf{A}_{[p,t]}\}, \tag{2.37}$$

where r is the reconstruction function which maps an assignment matrix to its trajectory hypothesis.

Since a target might be standing or sitting, 3D location of the target has an ambiguity when the target is detected by only one camera. To handle this ambiguity, we adopt two different kinds of update rules with respect to the number of detection $d(A^t)$ at frame t. The t-th frame of a trajectory hypothesis in $\mathcal{X}_{[p,t]}$ is updated with that of the optimally estimated trajectory $\mathbf{x}^*_{n_p}$ or a newly calculated point, that is,

$$\mathbf{x}^t = \begin{cases} \mathbf{x}^{*t}_{n_p}, & \text{if } d(A^t) > 1, \\ \mathbf{x}^{*t}_{n_p} + \Delta\mathbf{x}, & \text{if } d(A^t) = 1. \end{cases} \tag{2.38}$$

$\Delta\mathbf{x}$ denotes a displacement vector for making z-value of $\mathbf{x}^{*t}_{n_p} = (x^t_{n_p}, y^t_{n_p}, z^t_{n_p})$ into $z^t_{n_p} = \hat{z}$, where $\hat{z}$ is a linearly interpolated z-value from the neighboring frames where the target is detected at more than two cameras. The displacement vector $\Delta\mathbf{x}$ is obtained by a back-projection line $\Phi^k(\mathbf{d})$ where $\mathbf{d}$ is an associated detection of $\mathbf{x}^{*t}_{n_p}$. Letting $\Psi^k(\mathbf{d}, z')$ be a 3D point at $z = z'$ on the back-projection line $\Phi^k(\mathbf{d})$,

$$\Delta\mathbf{x} = \Psi^k(\mathbf{d}, \hat{z}) - \Psi^k(\mathbf{d}, z^t_{n_p}). \tag{2.39}$$

In conclusion, we reflect $\{\mathbf{x}^*_{n_p}\}$ to all trajectory hypotheses related with $\mathcal{X}_{[p,t]}$ in $\mathcal{X}^{(iter)}$. For two different assignment matrices A_{p_i} and A_{p_j}, the intuition for this ad-

ditional update is that if the t-th column of A_{p_i} is equals to that of A_{p_j}, the t-th frame of trajectory hypotheses $\mathbf{x}_{n_i}^t$ and $\mathbf{x}_{n_j}^t$ have the same 3D location. Finally, the newly updated $\mathcal{X}^{(iter)}$ from $\{\mathbf{x}_{n_p}^*\}$ is helpful to find better assignment matrices $\mathbf{A}^{(iter+1)}$ at the next iteration. The full update procedure is summarized in Algorithm 3.

Algorithm 3 Trajectory Hypotheses Update (THU)

Input: $\mathbf{A}^{(iter)} = \{A_1, A_2, ..., A_{\mathsf{P}}\}$, $\mathcal{X}^{(iter-1)}$

Output: $\mathcal{X}^{(iter)}$

for $p \leftarrow 1, 2, ..., \mathsf{P}$ **do**
  i) Update the n_p-th trajectory hypotheses:
  $\mathbf{x}_{n_p} \leftarrow \arg\min_{\mathbf{x}} \widetilde{c}(A_p, \mathbf{x})$ solved by (2.54)
  ii) Update trajectory hypotheses related to $\mathbf{x}_{n_p}$:
  for $t \leftarrow s_p, ..., e_p$ **do**
    Find $\mathcal{X}_{[p,t]}$ using (2.36,2.37)
    for all $\mathbf{x} \in \mathcal{X}_{[p,t]}$ **do**

$$\mathbf{x}^t \leftarrow \begin{cases} \mathbf{x}_{n_p}^t, & \text{if } d(A^t) > 1, \\ \mathbf{x}_{n_p}^t + \Delta\mathbf{x}, & \text{if } d(A^t) = 1. \end{cases}$$

    end for
  end for
end for

2.2.3 Initialization

Recalling our overall optimization framework, our method requires an initial feasible solution $\mathbf{A}^0$ and $\mathcal{X}^0$ to alternately optimize each variable. We introduce an initialization method to find an initial point with light computation. Before finding initial trajectory hypothesis set $\mathcal{X}^0$, we find all possible assignment matrix set $\mathbf{E}$ described in (2.4). For each assignment matrix $A_{p_n} \in \mathbf{E}$, the corresponding trajectory hypothesis $\mathbf{x}_n \in \mathcal{X}^0$ is calculated by minimizing only the 3D reconstruction accuracy, that is,

$$\min_{\mathbf{x}_n} c_{rec}(A_{p_n}, \mathbf{x}_n), \tag{2.40}$$

where $c_{rec}(\cdot)$ has been described in Sec. 2.1.2. The objective function in (2.40) can be rewritten as a problem minimizing a sum of square errors $\sum_t \sum_k \varepsilon_{rec}(A^t_{p_n}, \mathbf{x}^t_n, k)^2$ as shown in (2.13). Since the sum of square errors is independently calculated at each frame t, we find each 3D point $\mathbf{x}^t_n$ by solving a least squares problem,

$$\min_{\mathbf{x}^t_n} \sum_k \varepsilon_{rec}(A^t_{p_n}, \mathbf{x}^t_n, k)^2, \tag{2.41}$$

which can be solved in a closed-form as in (2.35) (Refer Section 2.4.1). Note that if a target is detected by only one camera, $\mathbf{x}^t_n$ cannot be determined since it has many solutions. In this case, $\mathbf{x}^t_n$ is decided by a given z-value, which is set to 0 for fullbody detection case or average height (1.7 meter in our experiment) for head detection case.

Next, we find an initial assignment set $\mathbf{A}^0$ that selects the P people's trajectories among the initial trajectory hypotheses in $\mathcal{X}^0$. We propose a greedy approach that finds initial assignment set $\mathbf{A}^0$ in a two-stage. In the first stage, for each frame t, we find *spatial assignment* set $\widetilde{\mathbf{A}}^t$ where each assignment matrix in $\widetilde{\mathbf{A}}^t$ has all-zero values except the t-th column. Using the $\mathbf{x}_{n_p} \in \mathcal{X}^0$ obtained from (2.40), the *spatial assignment* set $\widetilde{\mathbf{A}}^t = \{\widetilde{A}^t_p\}$ is obtained by solving the following equation,

$$\begin{aligned} \min_{\widetilde{\mathbf{A}}^t} \sum_p (\lambda_{rec} \quad & \cdot c_{rec}(\widetilde{A}^t_p, \mathbf{x}_{n_p}) + \lambda_{mid} \cdot c_{mid}(\widetilde{A}^t_p, \mathbf{x}_{n_p})) \\ & + \lambda_{fpt} \cdot (\sum_k |\mathbf{D}_{kt}| - \sum_p d(\widetilde{A}^t_p)), \end{aligned} \tag{2.42}$$

which is equivalent to minimize 3D reconstruction accuracy, missing detection terms defined in Sec. 2.1.2 and a penalty term that prevents a trivial solution that none of detections are selected. The penalty term is proportional to the number of detections not included in $\widetilde{\mathbf{A}}^t$ and $|\mathbf{D}_{kt}|$ denotes the number of detections in camera k and frame t. Here, we greedily find $\widetilde{\mathbf{A}}^t$ starting from $\widetilde{\mathbf{A}}^t = \varnothing$ by adding an assignment matrix $\widetilde{A}^t_p$ that has a minimum cost in (2.42). The greedy algorithm stops if all detections at frame t are included in $\widetilde{\mathbf{A}}^t$. In the second stage, for every frame t and $t+1$, we temporally associate $\widetilde{\mathbf{A}}^t$ and $\widetilde{\mathbf{A}}^{t+1}$ by solving a bipartite matching problem. A bipartite graph is generated where each node denotes a *spatial assignment* matrix in $\widetilde{\mathbf{A}}^t$ and

$\widetilde{\mathbf{A}}^{t+1}$ and a weight of each edge is defined by 3D Euclidean distance between two nodes. The bipartite matching problem is solved by the Hungarian algorithm [45]. Note that for each node in $\widetilde{\mathbf{A}}^{t}$, we calculate a ratio of the second minimal weight to the first minimal weight. If a node has a dominant edge that has small distance, the ratio would be large enough. To find only a reliable edge, each node that has the ratio smaller than a threshold τ is excluded from the matching problem. To summarize, the initial assignment set $\mathbf{A}^0$ is calculated by finding *spatial assignment* sets at all frames $\widetilde{\mathbf{A}}^1, ..., \widetilde{\mathbf{A}}^{\mathsf{F}}$ and solving bipartite matching problems between $\widetilde{\mathbf{A}}^{t}$ and $\widetilde{\mathbf{A}}^{t+1}$ at every frame t and $t+1$.

In our experiment, we use this greedy method not only for the initialization but also the baseline of our experiments, denoted as "Baseline". For the initialization, threshold τ is empirically set to 1.5. On the other hand, for the "Baseline", long trajectories are preferred rather than short and fragmented trajectories. Thus, we do not exclude any node in the bipartite matching, which means that τ is infinite.

2.3 Application: Real-time 3D localizing and tracking system

For real surveillance scenario, it needs to achieve satisfactory localizing and tracking performance in real-time and online manner. However, it is a challenging problem because it requires much time to perform object detection tasks simultaneously on several cameras, and the search space of data association for tracking increases exponentially in proportion to the number of cameras. The objective of this section is to develop a real-time scheme to achieve satisfactory performance of 3D-LTP using multiple cameras. For real-time processing, we parallelize the detection modules that produce bounding boxes detecting people in each camera. Robust detection performance is achieved through validation gating method using region of interest (ROI) setting and camera calibration. The online multi-camera data association problem is

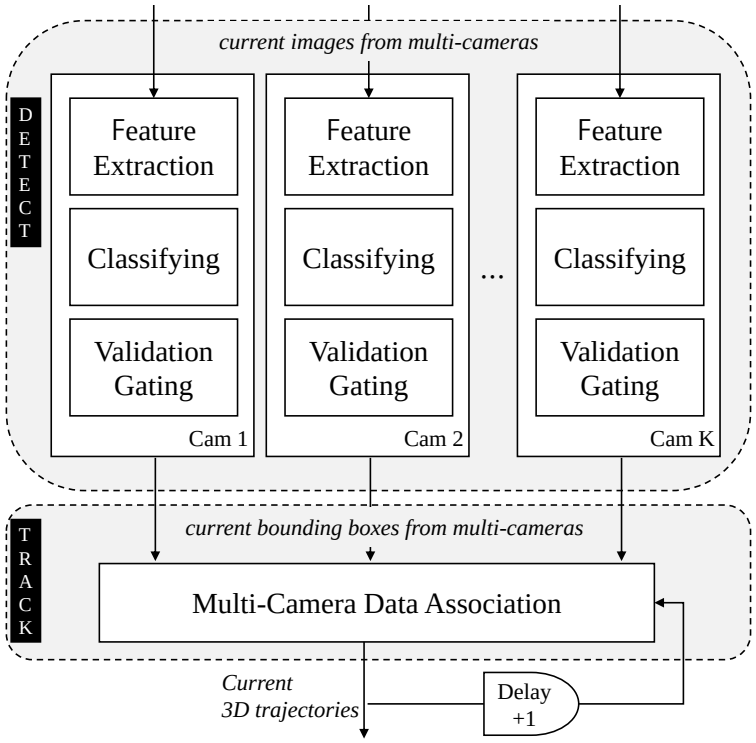


Figure 2.4: System overview.

formulated as the multidimensional assignment (MDA) problem between the previous trajectory and detections of people at each camera. To resolve the huge solution space, the proposed splitting/re-merging algorithm in Section 2.2 is modified to operate in an online manner.

2.3.1 System overview

The proposed system is composed of detection module and tracking module as shown in Figure 2.4. Each frame enters the detection module from the installed multiple cameras and people detection is performed. After the detection procedure, detected bounding boxes of all cameras are transferred to the tracking module. The tracking module maps the received detections into 3D space and performs multiple people tracking in 3D space. To operate the system in an actual environment, the system is designed so

as to run in real-time and online using only past and current detections.

2.3.2 Detection

In the proposed system, the goal of detection module is to localize people in the input frames from the employed cameras in the surveillance scene. As shown in Figure 2.4, we first perform people detection independently on each frame and provide the detection results to the tracking module in the form of bounding boxes. Since our system has K cameras, to operate the whole system in real-time (5 fps), the detection speed for each camera is at least faster than $5 \times K$ fps. In addition, the detection results should have a small number of false positives because the false positives will degrade the tracking accuracy and increase computational complexity in tracking module. To reduce false positives, we can utilize high-performance detector such as deep network-based detectors [47, 48, 49]. However, such kind of high-end detectors are hard to be applied to real-time multi-camera systems due to heavy computational complexity.

In this paper, we adopt a low computational detector based on Aggregated Channel Feature (ACF) [50], which uses simple hand-crafted features. To compensate the accuracy of the low computational detector, we propose a validation gating method by setting region of interest and using camera calibration. The ACF detector extracts 10 channel features from the input image including LUV channels and image gradients. To cope with various sizes of people, the input image is resized with 10 scales from 0.5 times to 2.0 times and construct image pyramid [50] by extracting channel features from the resized images. The ACF detects people by performing a binary classifier on the entire image pyramid using the trained people model.

The original ACF detects people using a full-body model, which is trained using INRIA pedestrian dataset [51]. The full-body model shows satisfactory performance in non-crowded scene, but, it has difficulties in detecting people in crowded scenes. In this case, the body parts of people tend to be severely occluded, while the head parts are relatively less occluded. Therefore, the head detector might show more robust

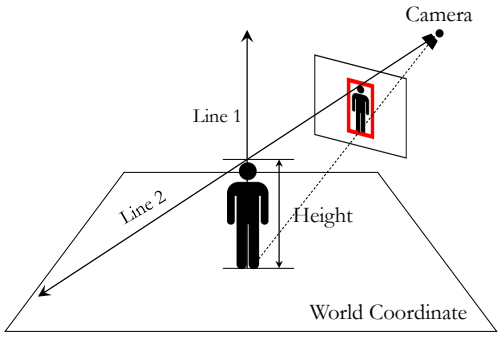
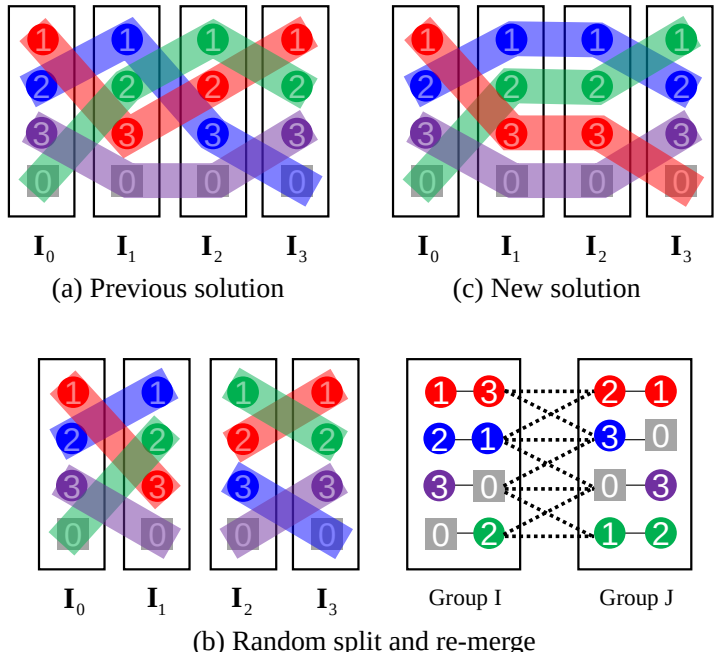


Figure 2.5: Example of estimating 3D height of the detected person in world coordinates.

performance than the full body model in a crowded scene. We train the ACF model for head detection using NLPR Head dataset [52]. The proposed system can select a proper detection type, head or full-body, to localize people in the input frame according to the situation of the target scene.

Since the raw detection results usually include a lot of false positives, post-processing is required to reject falsely detected people. The heights of the detected people in 3D space can be estimated using the camera calibration information. If the estimated height is outside the pre-determined upper or lower threshold, it is considered as a false positive. Figure 2.5 presents the process of estimating 3D heights of the detected boxes in the case of a full-body detector. First, the bottom center and top center coordinates in the image are converted to 3D world coordinates using camera calibration [53]. At this time, it is assumed that the z-axis value of the 3D bottom center is 0mm for the full body detection and 1700mm for the head detection. To estimate the hight of people or head, we use two lines. 'Line 1' is obtained by the line perpendicular to the ground at the bottom center point. 'Line 2' is obtained by the line back-projected from the 2D top center point in 3D space. The 3D top center coordinate can be estimated as the point at 'Line 1' where the distance between 'Line 1' and 'Line 2' becomes minimum. Using the z coordinate of the 3D top center point, we can estimate the height of a person in



(a) Previous solution (c) New solution

(b) Random split and re-merge

Figure 2.6: An example of an iteration of random split and re-merge.

3D space using the 3D top center and bottom center coordinates. The upper and lower threshold of 3D height are set to 1600 mm and 2200 mm for the full-body model, respectively. For the case of head detection model, they are set to 130 mm and 400 mm, respectively. If the estimated 3D height is outside of the threshold, it is regarded as a false positive and rejected from the detections.

In addition, our system can reject false positives by setting the region-of-interest in the surveillance scene. Detection results outside the area of interest are removed at this stage because they are often false positives or hard to be used for 3D localization and tracking.

2.3.3 Tracking

The proposed tracking module estimates 3D trajectories of the current frame from the detection bounding boxes of the current frame obtained from the detection module.

In order for the entire system to operate in an online manner, the tracking result of the current frame must be derived from that of the previous frame and the bounding boxes of the current frame. In this case, the problem of tracking corresponds to the problem of assigning the bounding box of each camera to the trajectory generated up to the previous frame. Since each bounding box is derived from one object, it must be assigned to only one trajectory. This assignment problem is known to be an NP-hard problem and so we propose an approximate algorithm to solve the problem with real-time speed as well as satisfactory performance.

When the number of cameras is K, one assignment is denoted as $(K+1)$-dimensional tuple $(i_0, i_1, ..., i_K)$. i_0 is the index of the trajectory generated up to the previous frame. The remaining i_k is the index of a detection in the k-th camera, which is assigned to the i_0 trajectory. For example, if K = 3, $(1, 3, 2, 1)$ means that the object having index 3 at 1-st camera, index 2 at 2-nd camera, and index 1 at 3-rd camera, is assigned to the previous trajectory with index 1. Since a newly appearing object has no previous trajectory, the index i_0 is assigned by 0. That is, $(0, 2, 1, 2)$ means that a new object is detected by the 3 cameras with indexes 2, 1, 2, respectively. The index of missing detection is assigned by 0. That is, $(2, 1, 3, 0)$ means that the object to be connected to 2-nd trajectory is missing and not detected at 3-rd camera.

As shown in Figure 2.6 (a), all possible combinations of assignments can be expressed through $(K+1)$-dimensional partite hypergraph. The first partite set is a set of indexes of the previous trajectories and the remaining partite set is a set of indexes of bounding boxes of each camera. The aforementioned assignments become hyperedges of the hypergraph. Formally, the set of hyperedges is defined by

$$\mathbf{E} = \{(i_0, i_1, ..., i_K) |\ i_k \in \mathbf{I}_k,\ k = 0, ..., K\}, \tag{2.43}$$

where $\mathbf{I}_k$ means an index set of bounding boxes detected by the k-th camera or trajectories assigned until the previous frame, that is,

$$\mathbf{I}_k = \{0, ..., N_k\}; \quad k = 0, 1, ..., K. \tag{2.44}$$

N_k denotes the number of detections in the k-th camera and the index 0 refers to the dummy index meaning a missing detection or there is no previous trajectory to be connected to the detections at the current frame. Adopting the cost function defined in [54], the cost of each assignment is imposed by considering the five physical characteristics: 3D reconstruction accuracy, motion smoothness, visibility from camera, starting/ending zone violation, false positive trajectory. The cost function is defined as $c : \mathbf{E} \rightarrow \mathbb{R}$. We simply write the cost of assignment $(i_0, i_1, ..., i_K)$ as $c_{(i_0,i_1,...,i_K)}$.

By introducing binary decision variable x, the problem of finding the best assignments among $\mathbf{E}$ with disjoint constraints can be formulated as the following multidimensional assignment problem (MDA), i.e.,

$$\min \sum_{(i_0,i_1,...,i_K)\in\mathbf{E}} c_{(i_0,i_1,...,i_K)} x_{(i_0,i_1,...,i_K)} \tag{2.45}$$

subject to

$$\sum_{(i_0,i_1,...,i_K)\in\mathbf{E}, i_k:fixed} x_{(i_0,i_1,...,i_K)} = 1,$$

$$i_k = 1, 2, ..., N_k, \qquad k = 1, 2, ..., K,$$

where $x_{(i_0,i_1,...,i_K)}$ has 1 if the assignment $(i_0, i_1, ..., i_K)$ is selected, otherwise has 0. The MDA problem in (2.45) is a two-dimensional assignment problem for a single camera, and a $(K + 1)$ - dimensional assignment problem for K cameras. Then total number of constraints becomes $\sum_k N_k$. In the case of two-dimensional assignment problem, there exists an algorithm that can solve in a polynomial-time with Hungarian algorithm [45]. But the MDA problem over three dimensions is NP-hard problem [42]. Recently, Byeon et al. [54] proposed the iterative split and re-merging algorithm to solve multi-camera data association problem in a batch setting. Here, we modify their iterative split and re-merging algorithm to fit an online setting. Figure 2.6 (a)-(c) shows an example of an iteration of split and re-merging algorithm. Given the previous solution, we first randomly split into two groups. Next, we re-merge the two groups by solving the assignment problem between the two groups. The number of valid cases

for splitting two groups of (K + 1)-partite sets is $2^K - 1$. Note that the batch approach in [54] requires $T * (2^K - 2)$ number of valid cases for splitting two groups, where T denotes the number of frames. Among the valid cases, we randomly select one at every iteration. The problem of finding optimal re-merge pairs are two-dimensional assignment problem, which can be solved in a polynomial time [45]. In our experiments, the proposed algorithm iterates this process for the given maximum iteration.

2.4 Appendix

2.4.1 Derivation of equation (2.35)

Given assignment matrix $A_p^{(iter)}$, the problem in (2.35) minimizes the cost terms c_{rec} and c_{mot} with repect to $\mathbf{x}$. The cost term $\lambda_{rec} \cdot c_{rec}$ in (2.13) can be rewritten by a matrix-form,

$$\sum_{t=s}^{e} \left\| (\widehat{W}_r^t)^{1/2} (\widehat{P}_t \mathbf{x}^t - \widehat{\mathbf{q}}^t) \right\|^2 + \mathsf{c}', \tag{2.46}$$

where for each frame t, the matrices $\widehat{P}_t \in \mathbb{R}^{3\mathsf{d}\times 3}$, $\widehat{\mathbf{q}}_t \in \mathbb{R}^{3\mathsf{d}\times 1}$, and $\widehat{W}_t \in \mathbb{R}^{3\mathsf{d}\times 3\mathsf{d}}$ are calculated from the 3D back-projection lines of the number of d detections,

$$\widehat{P}^t = \begin{bmatrix} P_1^t \\ \vdots \\ P_{d(A^t)}^t \end{bmatrix}, \widehat{\mathbf{q}}_t = \begin{bmatrix} \mathbf{q}_1^t \\ \vdots \\ \mathbf{q}_{d(A^t}^t \end{bmatrix}, \widehat{W}_r^t = \frac{\lambda_{rec}}{|\mathbf{N}(\mathbf{x}^t)|}\mathbb{I}. \tag{2.47}$$

The motion smoothness term $\lambda_{mot} \cdot c_{mot}$ in (2.15) also can be represented by the following matrix-form,

$$\sum_{t=s}^{e} \left\| (\widehat{W}_d^t)^{1/2} \widehat{R}^t \begin{bmatrix} \mathbf{x}^{t-1} \\ \mathbf{x}^t \end{bmatrix} \right\|^2 + \left\| (\widehat{W}_c^t)^{1/2} \widehat{S}^t \begin{bmatrix} \mathbf{x}^{t-1} \\ \mathbf{x}^t \\ \mathbf{x}^{t+1} \end{bmatrix} \right\|^2 \tag{2.48}$$

where

$$\widehat{R}^t = (-\mathbb{I}_{3\times 3}\ \mathbb{I}_{3\times 3}) \in \mathbb{R}^{3\times 6}, \tag{2.49}$$

$$\widehat{S}^t = (\mathbb{I}_{3\times 3}\ -2\mathbb{I}_{3\times 3}\ \mathbb{I}_{3\times 3}) \in \mathbb{R}^{3\times 9}, \tag{2.50}$$

$$\widehat{W}_d^t = \lambda_{mot} w_d(t) \cdot \mathbb{I}_{3\times 3}, \tag{2.51}$$

$$\widehat{W}_c^t = \lambda_{mot} w_c(t) \cdot \mathbb{I}_{3\times 3}. \tag{2.52}$$

By taking in account all frames, concatenated matrices are given by $\widehat{P} = \begin{bmatrix} \widehat{P}^s & & \\ & \ddots & \\ & & \widehat{P}^e \end{bmatrix}$, $\widehat{\mathbf{q}} = \begin{bmatrix} \widehat{\mathbf{q}}^s \\ \vdots \\ \widehat{\mathbf{q}}^e \end{bmatrix}$, $\widehat{R} = \begin{bmatrix} \widehat{R}^s & & \\ & \ddots & \\ & & \widehat{R}^e \end{bmatrix}$, $\widehat{S} = \begin{bmatrix} \widehat{S}^s & & \\ & \ddots & \\ & & \widehat{S}^e \end{bmatrix}$, $\widehat{W}_r = \begin{bmatrix} \widehat{W}_r^s & & \\ & \ddots & \\ & & \widehat{W}_r^e \end{bmatrix}$, $\widehat{W}_d = \begin{bmatrix} \widehat{W}_d^s & & \\ & \ddots & \\ & & \widehat{W}_d^e \end{bmatrix}$, $\widehat{W}_c = \begin{bmatrix} \widehat{W}_c^s & & \\ & \ddots & \\ & & \widehat{W}_c^e \end{bmatrix}$. The problem in (2.35) can be rewritten by the following weighted least-squares minimization problem,

$$\min_{\mathbf{x}} \left\| \widehat{W}_r^{1/2}(\widehat{P}\mathbf{x} - \widehat{\mathbf{q}}) \right\|^2 + \left\| \widehat{W}_d^{1/2}(\widehat{R}\mathbf{x}) \right\|^2 + \left\| \widehat{W}_c^{1/2}(\widehat{S}\mathbf{x}) \right\|^2, \tag{2.53}$$

which can be solved in a closed-form,

$$\mathbf{x}^* = (\widehat{P}^\mathsf{T}\widehat{W}_r\widehat{P} + \widehat{R}^\mathsf{T}\widehat{W}_d\widehat{R} + \widehat{S}^\mathsf{T}\widehat{W}_c\widehat{S})^{-1}\widehat{P}^\mathsf{T}\widehat{W}_r\widehat{\mathbf{q}}. \tag{2.54}$$

Chapter 3

Variational Inference Approach

3.1 Problem Formulation

In this section, we formulate a MAP problem to solve the trajectory assignment problem and 3D position estimation problem (See Figure 3.1). In section 3.1.1, the notations are defined for explaining our formulation. In section 3.1.2, the variational inference framework is derived to get a tractable MAP formulation.

3.1.1 Notations

Detections. $\mathbf{D}$ denotes the set of all detections in all frames and cameras. Each detection $d_i \in \mathbf{D}$ is defined by a vector $d_i = (x_i, y_i, w_i, h_i, t_i, c_i)$ where x_i, y_i, w_i, h_i represent the position and size of a bounding box and t_i, c_i denote the index of time (i.e., frame) and camera, respectively.

Generalized index I. A generalized index $I \in \Omega$ denotes an index set of detections guessed to come from an object in a frame, where I must include at most one detection from each camera. All possible combinations are collected as

$$\Omega = \{I | t_i = t_j \wedge c_i \neq c_j, \forall i, j \in I,\ i \neq j\} \tag{3.1}$$

In the following, we use I to denote a detection hypothesis for an object.

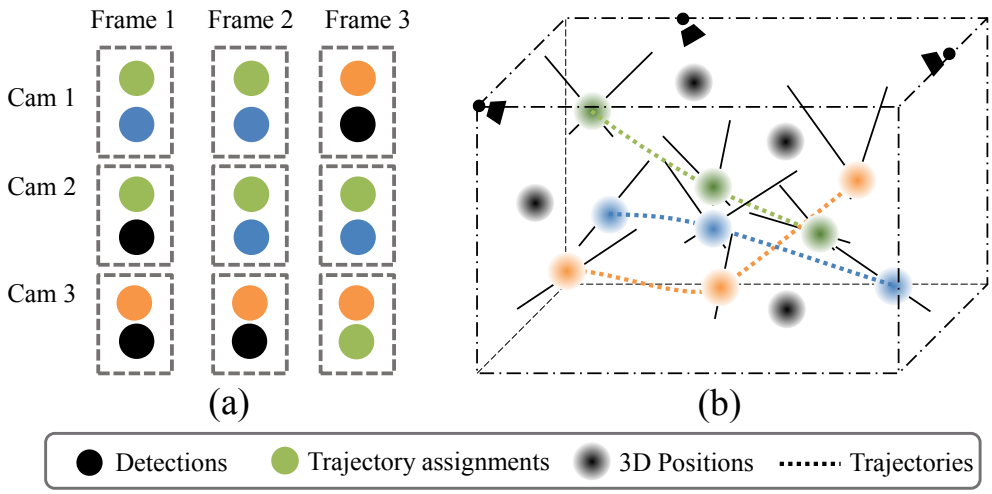


Figure 3.1: The problem of data association and 3D localization in multiple cameras. (a) For the data association problem, a spatial association across cameras and a temporal association between frames should be calculated. The color of solid circle depicts the identity of a trajectory assignment. (b) In general, the 3D position is reconstructed by back-project on from the detection in each camera. Since the altitude (height) of a target is unknown, the 3D position should be determined by exploiting multiple detections from multiple cameras and neighbor frames. We argue that the two problems of data association and 3D localization (object position estimation) are highly correlated. The problem to assign detections to a trajectory needs to be solved with the problem of finding the 3D positions of the trajectory assignments.

Detection hypothesis. The detection hypothesis for an object indexed by I, D_I is defined by,

$$D_I \triangleq \{d_i | i \in I\}. \tag{3.2}$$

3D Position variables. 3D position of a target is defined by a random variable $\mathbf{x}_I \in \mathbb{R}^3$. $\mathbf{x}_I$ is associated with D_I and the set of all possible $\mathbf{x}_I$ is denoted by $\mathbf{X}$.

Trajectory assignment variables. A trajectory of a target is indexed by an ordered set of generalized indices Is associated with the target. The trajectory assignment variable τ for a target indexed by Is is defined by

$$\tau = (I^{\mathsf{s}}, I^{\mathsf{s}+1}, ..., I^{\mathsf{e}}), \tag{3.3}$$

where s and e denote the start and end frames. A set of trajectory assignment variables is defined by $\mathbf{T} = \{\tau_0, \tau_1, ..., \tau_K\}$. τ_0 refers to a set of Is for a fake (phantom etc.) and τ_k refers to a set of Is for the k-th target. The $\mathbf{T}$ satisfies two constraints: *union* and *non-overlap* constraints.

Union constraint. Each detection hypothesis is involved in one trajectory assignment $\tau \in \mathbf{T}$.

$$\bigcup_{\tau \in \mathbf{T}} \bigcup_{I \in \tau} D_I = \mathbf{D} \tag{3.4}$$

Non-overlap constraint. Each trajectory assignment variable $\tau_u \in \mathbf{T}$ does not share the same detection with any other τ_v for $u \neq v$, i.e.,

$$\left(\bigcup_{I \in \tau_u} D_I \right) \cap \left(\bigcup_{I \in \tau_v} D_I \right) = \emptyset. \tag{3.5}$$

3.1.2 MAP formulation

Given detections $\mathbf{D}$, we formulate the problem of finding optimal trajectory assignments $\mathbf{T}$ and 3D positions $\mathbf{X}$ that maximize a posterior, which is given by

$$\mathbf{T}^*, \mathbf{X}^* = \arg\max_{\mathbf{T}, \mathbf{X}} p(\mathbf{T}, \mathbf{X} | \mathbf{D}). \tag{3.6}$$

However, solving the MAP problem is difficult. Previous multi-camera approaches fixed $\mathbf{X}$ to the pre-computed one [54, 38, 36, 41] and optimized $\mathbf{T}$ with respect to the given 3D locations. Instead, we consider $\mathbf{X}$ as random variables. Then a tractable optimization problem is formulated via variational approximation of MAP problem.

Adopting variational inference framework [55, Ch.9] that regards $\mathbf{X}$ as hidden variables in $p(\mathbf{T}, \mathbf{X}|\mathbf{D})$, we introduce a variational distribution $q(\mathbf{X})$ defined over the 3D position variables $\mathbf{X}$. And for any choice of $q(\mathbf{X})$, the following decomposition holds:

$$\ln p(\mathbf{T}|\mathbf{D}) = \mathcal{L}(q(\mathbf{X}), \mathbf{T}) + KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{T}, \mathbf{D})), \tag{3.7}$$

$$\mathcal{L}(q(\mathbf{X}), \mathbf{T}) = \int_{\mathbf{X}} q(\mathbf{X}) \ln \frac{p(\mathbf{T}, \mathbf{X}|\mathbf{D})}{q(\mathbf{X})} \tag{3.8}$$

$$KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{T}, \mathbf{D})) = -\int_{\mathbf{X}} q(\mathbf{X}) \ln \frac{p(\mathbf{X}|\mathbf{T}, \mathbf{D})}{q(\mathbf{X})} \tag{3.9}$$

where $KL(\cdot)$ denotes Kullback-Leibler (KL) divergence and $\mathcal{L}$ is a lower bound of which maximum occurs when the KL divergence vanishes.

Variational distribution $q(\mathbf{X})$. We choose a fully factorized form of $q(\mathbf{X})$, i.e.,

$$q(\mathbf{X}) = \prod_{I \in \Omega} q_I(\mathbf{x}_I), \tag{3.10}$$

where each factorized distribution q_i contributes to approximate $p(\mathbf{X}|\mathbf{T}, \mathbf{D})$. We assume that the factorized distribution of each 3D hypothesis $q_I(\mathbf{x}_I)$ is parameterized by Gaussian mean μ_I and covariance matrix Σ_I,

$$q_I(\mathbf{x_I}) \sim N(\mu_I, \Sigma_I). \tag{3.11}$$

A set of means and covariance matrices are denoted by $\widehat{\mu} = \{\mu_I | I \in \Omega\}$ and $\widehat{\Sigma} = \{\Sigma_I | I \in \Omega\}$ respectively.

Variational Expectation-Maximization (V-EM). To maximize the left-hand side in (3.7), we have to estimate $q(\mathbf{X})$ as well as maximizing the objective with respect to $\mathbf{T}$, but it is intractable. Instead, we adopt an expectation-maximization (EM) framework

that alternately optimize $q(\mathbf{X})$ and $\mathbf{T}$ while fixing the other. In the E-step of the k-th iteration, $q(\mathbf{X})$ is found such that it minimizes the KL-divergence in (3.9) fixing $\mathbf{T} = \mathbf{T}^{*(k-1)}$. Using (3.7), the E-step is equivalent to the problem of finding Gaussian means and covariance matrices that minimize $-\mathcal{L}$ given $\mathbf{T}^{*(k-1)}$, i.e.,

$$\widehat{\mu}^{*(k)}, \widehat{\Sigma}^{*(k)} = \underset{\widehat{\mu}^{(k)}, \widehat{\Sigma}^{(k)}}{\arg\min} -\mathcal{L}(\widehat{\mu}^{(k)}, \widehat{\Sigma}^{(k)}, \mathbf{T}^{*(k-1)}). \tag{3.12}$$

In the M-step, the lower bound $\mathcal{L}$ is maximized with respect to $\mathbf{T}$. By taking the minus, the M-step is formulated by

$$\mathbf{T}^{*(k)} = \underset{\mathbf{T}^{(k)}}{\arg\min} -\mathcal{L}(\widehat{\mu}^{*(k)}, \widehat{\Sigma}^{*(k)}, \mathbf{T}^{(k)}). \tag{3.13}$$

3.2 Optimization

In this section, optimization procedure via V-EM is presented. In section 3.2.1, the posterior distribution $p(\mathbf{T}, \mathbf{X}|\mathbf{D})$ is derived to obtain $\mathcal{L}(\cdot)$ in (3.8). In section 3.2.2, Variational EM procedure is described to find the solution iteratively.

3.2.1 Posterior distribution

According to the Bayes rule, the posterior distribution on $\mathbf{T}$ and $\mathbf{X}$ given $\mathbf{D}$, is proportional to the product of likelihood and prior as

$$p(\mathbf{T}, \mathbf{X}|\mathbf{D}) \propto p(\mathbf{D}|\mathbf{T}, \mathbf{X}) \cdot p(\mathbf{T}, \mathbf{X}). \tag{3.14}$$

Likelihood: The likelihood $p(\mathbf{D}|\mathbf{T}, \mathbf{X})$ evaluates how good $\mathbf{T}$ and $\mathbf{X}$ describe the observed detections $\mathbf{D}$. For instance, if a detection hypothesis D_I is a fake actually, the likelihood is maximized when it is assigned to τ_0. If D_I is observed from a true target, it should be assigned to the best track τ^* with $\mathbf{x}_I$ having the minimal observation error. Following this concept, the likelihood is defined by

$$p(\mathbf{D}|\mathbf{T}, \mathbf{X}) \propto \prod_{\tau \in \mathbf{T} \setminus \{\tau_0\}} \prod_{I \in \tau} \psi_{tar}(D_I, \mathbf{x}_I) \prod_{I \in \tau_0} \psi_{fak}(D_I) \tag{3.15}$$

where $\psi_{tar}(\cdot)$ denotes a probability related with 3D observation error between detections D_I and a 3D position $\mathbf{x}_I$, whereas ψ_{fak} means a probability that the detection D_I be a fake.

Prior: The prior on $\mathbf{X}$ and $\mathbf{T}$ is obtained by the product of individual priors by assuming that every trajectory moves independently to others:

$$p(\mathbf{T}, \mathbf{X}) = \prod_{\tau \in \mathbf{T}} p(\tau, \mathbf{X}). \tag{3.16}$$

Each prior is modeled to be proportional to linking probability following Markov model as

$$p(\tau, \mathbf{X}) \propto \psi_s(\mathbf{x}_{I^s}) \psi_e(\mathbf{x}_{I^e}) \prod_{I,J \in adj(\tau)} \psi_{link}(\mathbf{x}_I, \mathbf{x}_J), \tag{3.17}$$

where ψ_{link} encodes a linking probability of temporally adjacent I and J within a trajectory and $\psi_s(\mathbf{x})$, $\psi_e(\mathbf{x})$ model the probability of a trajectory starting/ending at the 3D location $\mathbf{x}$.

Probability modeling

We design a probability model using the seven cost terms designed for multi-camera tracking problem: cost for 3D observation error c_{obs}, missing detection c_{mid}, fake c_{fak}, frequent starting or ending c_{fse}, starting or ending at non-entrance/exit zone c_{eez}, non-smoothing motion c_{mot} and jumped frame c_{jmp}, which are defined in the following. Every probability model ψ_{tar}, ψ_{fak}, ψ_{link}, ψ_s, and ψ_e follows Boltzmann distribution, which are defined by

$$-\ln \psi_{tar} = \lambda_{obs} \cdot c_{obs} + \lambda_{mid} \cdot c_{mid}, \tag{3.18}$$

$$-\ln \psi_{fak} = \lambda_{fak} \cdot c_{fak}, \tag{3.19}$$

$$-\ln \psi_{link} = \lambda_{mot} \cdot c_{mot} + \lambda_{jmp} \cdot c_{jmp}, \tag{3.20}$$

$$-\ln \psi_s = -\ln \psi_e = \lambda_{fse} \cdot c_{fse} + \lambda_{eez} \cdot c_{eez}, \tag{3.21}$$

of which costs are explained in the following.

Target ψ_{tar}**.** In modeling ψ_{tar}, the c_{obs} is the cost for observation error which is given by the mean of distances between each detection $d_i \in D_I$ and a 3D position $\mathbf{x}_I$. The cost increases whenever the 3D position $\mathbf{x}_I$ is far from any detections in D_I. The distance between detection and 3D position is modeled in 3D space. To calculate the distance in 3D space, we back-project a 2D point of d_i (e.g. foot or head point) to 3D space which is called a back-projection line. The distance between back-projection line and 3D position is modeled as a form of linear equation.

$$c_{obs}(D_I, \mathbf{x}_I) = \frac{1}{|I|} \cdot \sum_{i \in I} \|A_i \mathbf{x}_I - \mathbf{b}_i\|^2, \tag{3.22}$$

where A_i and $\mathbf{b}_i$ are determined by detection d_i and calculated by using camera projection matrix. See Section 3.3.1 for the details of calculations of A_i and $\mathbf{b}_i$. The c_{mid} measures how well the number of visible cameras matches the number of actually detected detections. If the number of detections D_I is less than the number of visible cameras of $\mathbf{x}_I$, c_{mid} increases, i.e.,

$$c_{mid}(D_I, \mathbf{x}_I) = \|v(\mathbf{x}_I) - |I|\|\,, \tag{3.23}$$

where $v(\mathbf{x}_I)$ denotes the number of camera where $\mathbf{x}_I$ is visible.

Fake ψ_{fak}**.** The cost $c_{fak}(D_I)$ for a fake D_I is proportional to the number of false positives in D_I, i.e., $c_{fak}(D_I) = |I|$

Link ψ_{link}**.** The c_{mot} models the tendency that a target tends to move through a shortest path. $c_{mot}(\mathbf{x}_I, \mathbf{x}_J)$ is proportional to Euclidean distance between $\mathbf{x}_I$ and $\mathbf{x}_J$, i.e.,

$$c_{mot}(\mathbf{x}_I, \mathbf{x}_J) = (\frac{|I| + |J|}{2}) \cdot \|\mathbf{x}_I - \mathbf{x}_J\|^2, \tag{3.24}$$

In addition to c_{mot}, the cost for linking increases if I and J are temporally apart (jumped), which is given by

$$c_{jmp} = (\frac{|I| + |J|}{2}) \cdot \Delta t_{IJ}, \tag{3.25}$$

where Δt_{IJ} refers to the number of jumped frames between frames of I and J.

Starting/ending ψ_s,ψ_e. The cost to the starting or ending of a trajectory is defined by two aspects; First, each trajectories has a cost whenever it starts or ends, i.e., $c_{fse} = \frac{|I|}{2}$, which implies that a cost is given to a short trajectory, i.e., the frequently starting or ending trajectory. Second, every trajectory is enforced to start and end at entrance/exit zone, e.g., locations near doorways. Hence, the cost is given to a starting or ending out of entrance/exit zone as

$$c_{eez}(\mathbf{x}_I) = \frac{|I|}{2} \cdot u(\mathbf{x}_I), \tag{3.26}$$

where $u(\mathbf{x}_I)$ is an indicator function that has 0 when $\mathbf{x}_I$ is in entrance/exit zone and 1 otherwise.

3.2.2 V-EM algorithm

The proposed variational EM algorithm alternately optimize (3.12) and (3.13) using the lower bound derived from (3.8). Since $q(\mathbf{X})$ follows Gaussian distribution and is fully factorized, the lower bound is decomposed to a posterior term $\mathcal{G}$ on $p(\mathbf{T}, \mathbf{X}|\mathbf{D})$ and an entropy term $\mathcal{H}$ on $q(\mathbf{X})$ from (3.8) as

$$-\mathcal{L}(\widehat{\mu}^{(k)}, \widehat{\Sigma}^{(k)}, \mathbf{T}^{(k)}) = \mathcal{G} - \mathcal{H}, \tag{3.27}$$

such that

$$\mathcal{G} = \sum_{\tau \in \mathbf{T}^{(k)} \setminus \{\tau_0\}} g(\widehat{\mu}_\tau^{(k)}, \widehat{\Sigma}_\tau^{(k)}), \;\; \mathcal{H} = \sum_{I \in \Omega} h(\Sigma_I^{(k)}). \tag{3.28}$$

The $\widehat{\mu}_\tau^{(k)}$ and $\widehat{\Sigma}_\tau^{(k)}$ are a set of all means and covariance matrices of 3D positions in τ at the k-th iteration. And the entropy term $\mathcal{H}$ for Gaussian distribution can be easily derived as $h(\Sigma_I^{(k)}) = \frac{1}{2} \ln \left((2\pi e)^3 \cdot \left| \Sigma_I^{(k)} \right| \right)$.

Using (3.14-3.21), the posterior term g for τ is derived by integrating log-posterior model (details are given in Section 3.3.2). In derivation, it is intractable to integrate the functions $v(\mathbf{x}_I)$ in (3.23) and $u(\mathbf{x}_I)$ in (3.26). Instead, we use the zeroth-order approximation of each function at the recently obtained mean of $\mathbf{x}_I$, i.e., $\mathbf{x}_I = \mu_I^{*(k-1)}$ in the E-step or $\mathbf{x}_I = \mu^{*(k)}$ in the M-step. As shown in Section 3.3.2, the posterior

term g for τ is obtained by

$$g(\widehat{\mu}_\tau^{(k)}, \widehat{\Sigma}_\tau^{(k)}) = g_I^s(\mu_{I^s}^{(k)}, \Sigma_{I^s}^{(k)}) + g_I^e(\mu_{I^e}^{(k)}, \Sigma_{I^e}^{(k)}) + \sum_{I\in\tau} g_I(\mu_I^{(k)}, \Sigma_I^{(k)}) + \sum_{I,J\in adj(\tau)} g_{IJ}(\mu_I^{(k)}, \mu_J^{(k)}, \Sigma_I^{(k)}, \Sigma_J^{(k)}), \tag{3.29}$$

where

$$g_I^s = g_I^e = \frac{|I|}{2} \cdot \left(\lambda_{fse} + \lambda_{eez} \cdot u(\mu_I^{*(k-1)})\right), \tag{3.30}$$

$$g_I = \lambda_{obs} \cdot \left(\frac{1}{|I|}\sum_{i\in I} tr(A_i\Sigma_I^{(k)}A_i^T) + \left\|A_i\mu_I^{(k)} - \mathbf{b}_i\right\|^2\right) + \lambda_{mid} \cdot \left(v(\mu_I^{*(k-1)}) - |I|\right), \tag{3.31}$$

$$g_{IJ} = \lambda_{mot} \cdot \left(\frac{|I|+|J|}{2}\right) \cdot \left(tr(\Sigma_I^{(k)} + \Sigma_J^{(k)}) + \left\|\mu_I^{(k)} - \mu_J^{(k)}\right\|^2\right). \tag{3.32}$$

Note that $u(\mu_I^{*(k-1)})$ and $v(\mu_I^{*(k-1)})$ are valid in E-step and they are replaced by $u(\mu_I^{*(k)})$ and $v(\mu_I^{*(k)})$ in M-step.

E-step

In the k-th iteration, for the fixed $\mathbf{T}^{(k)} = T^{*(k-1)}$ in (3.27), the optimal mean and covariance matrix of each $\mathbf{x}_I$ is found by minimizing the minus of lower bound in (3.27). Each $\tau \in \mathbf{T}^{*(k-1)}\backslash\{\tau_0\}$ is independent to the others and the optimal mean for each τ is derived by,

$$\begin{aligned}\widehat{\mu}_\tau^{*(k)} &= \arg\min_{\widehat{\mu}_\tau^{(k)}} g(\widehat{\mu}_\tau^{(k)}, \widehat{\Sigma}_\tau^{(k)}), \\ &= \arg\min_{\widehat{\mu}_\tau^{(k)}} \lambda_{obs} \cdot \sum_{I\in\tau}\frac{1}{|I|}\sum_{i\in I}\left\|A_i\mu_I^{(k)} - \mathbf{b}_i\right\|^2 + \\ &\quad \lambda_{mot} \cdot \sum_{I,J\in adj(\tau)}\left(\frac{|I|+|J|}{2}\right) \cdot \left\|\mu_I^{(k)} - \mu_J^{(k)}\right\|^2\end{aligned} \tag{3.33}$$

which is the weighted least square problem that has a closed-form solution. For covariance matrix $\Sigma_I^{(k)}$, taking the gradient of $\mathcal{L}$ w.r.t. $\Sigma_I^{(k)}$ equal to zero, leads to

$$\Sigma_I^{*(k)} = \left(2 \cdot \frac{\lambda_{obs}}{|I|} \sum_{i \in I} A_i^\mathsf{T} A_i + \lambda_{mot} \cdot w_{mot} \cdot \mathbb{I}_{3\times 3} \right)^{-1} \tag{3.34}$$

where

$$w_{mot} = \begin{cases} |I_{prev}| + |I|\,, \text{ if } I \text{ is a start index,} \\ |I_{next}| + |I|\,, \text{ if } I \text{ is an end index,} \\ |I_{prev}| + 2 \cdot |I| + |I_{next}|\,, \text{ otherwise,} \end{cases} \tag{3.35}$$

whereas I_{prev} and I_{next} are the previous index and the next index, respectively. See Section 3.3.3 for the details of derivations of (3.33 - 3.35).

For the case that the length of trajectory equals to 1, the optimal mean and covariance matrix of $\mathbf{x}_I$ in (3.33) and (3.34) is obtained as follows:

$$\begin{cases} \mu_I^{*(k)} = (\sum_{i \in I} A_i^\mathsf{T} A_i)^{-1} (\sum_{i \in I} A_i^\mathsf{T} \mathbf{b}_i) \\ \Sigma_I^{*(k)} = \left(2 \cdot \frac{\lambda_{obs}}{|I|} \sum_{i \in I} A_i^T A_i \right)^{-1} \end{cases} \tag{3.36}$$

Note that the inverse term is not available when $\sum_{i \in I} A_i^\mathsf{T} A_i$ is a singular matrix in the case of $|I| = 1$. This means that 3D position of a target cannot be determined when the target is detected by only one camera. Our key idea to resolve this problem is to introduce a virtual back-projection line making $\sum_{i \in I} A_i^\mathsf{T} A_i$ non-singular. The virtual back-projection line is induced from the nearest trajectory to the back-projection line of detection d_i at the corresponding frame. To find the nearest trajectory, for each $\tau \in \mathbf{T}^{*(k-1)} \backslash \{\tau_0\}$, the 3D point of τ at frame t_i where d_i is detected, is defined by $\mu_{I(\tau,t_i)}^{*(k-1)}$. If the point does not exist due to missing detection, a prediction is performed to find the unobserved 3D point. For the prediction, we use autoregressive model [56] which is one of linear regression model using Hankel matrix. The coefficients of virtual back-projection line, $A_{I,\tau}$ and $b_{I,\tau}$, are obtained by the tangent line at the point $\mu_{I(\tau,t_i)}^{*(k-1)}$. Using $A_{I,\tau}$ and $\mathbf{b}_{I,\tau}$, the distance from a point $\mathbf{x}$ to the virtual line is given

by $\|A_{I,\tau}\mathbf{x} - \mathbf{b}_{I,\tau}\|$ similar to (3.22). The nearest trajectory τ_* is obtained by

$$\tau^* = \arg\min_{\tau} \min_{\mathbf{x}} \|A_i\mathbf{x} - \mathbf{b}_i\|^2 + \|A_{I,\tau}\mathbf{x} - \mathbf{b}_{I,\tau}\|^2. \tag{3.37}$$

Using τ^*, the formula of μ_I^* and Σ_I^* in (3.36) for $|I| = 1$ are changed by

$$\begin{cases} \mu_I^{*(k)} = (A_i^\mathsf{T} A_i + A_{I,\tau^*}^\mathsf{T} A_{I,\tau^*})^{-1}(A_i^\mathsf{T}\mathbf{b}_i + A_{I,\tau^*}^\mathsf{T}\mathbf{b}_{I,\tau^*}), \\ \Sigma_I^{*(k)} = \left(2 \cdot \frac{\lambda_{obs}}{|I|}(A_i^T A_i + A_{I,\tau^*}^\mathsf{T} A_{I,\tau^*})\right)^{-1}. \end{cases} \tag{3.38}$$

Although $\mu_I^{(k)}, \Sigma_I^{(k)}$ for $I \in \tau_0$ are not influenced by minimizing (3.27), the optimal $\mu_I^{*(k)}$ and $\Sigma_I^{*(k)}$ are calculated by using (3.38). These updates do not change the posterior probability, but it helps to find better $\mathbf{T}^{*(k)}$ in the M-step.

M-step

In the M-step, $\mathbf{T}^{(k)}$ is optimized by fixing every $\mu_I^{*(k)}$ and $\Sigma_I^{*(k)}$. Thus, The entropy term $H(\Sigma_I^{*(k)})$ in (3.27) is a constant. Using this fact, the M-step in (3.13) is equivalent to the problem of

$$\mathbf{T}^{*(k)} = \arg\min_{\mathbf{T}^{(k)}} \sum_{\tau \in \mathbf{T}^{(k)} \setminus \{\tau_0\}} g(\widehat{\mu}_\tau^{*(k)}, \widehat{\Sigma}_\tau^{*(k)}), \tag{3.39}$$

where $g(\cdot)$ is given in (3.29). By introducing binary variables f, the problem in (3.39) can be re-written as an integer linear program:

$$\min_{\mathbf{f}} \sum_{I \in \Omega} c_I^s f_I^s + \sum_{I \in \Omega} c_I^e f_I^e + \sum_{I \in \Omega} c_I f_I + \sum_{I,J} c_{IJ} f_{IJ} \tag{3.40}$$

subject to

$$f_I^s, f_I^e, f_I, f_{IJ} \in \{0, 1\}, \tag{3.41}$$

$$f_I^s + \sum_{J \in \Omega} f_{JI} = f_I = f_I^e + \sum_{J \in \Omega} f_{IJ}, \tag{3.42}$$

$$\sum_{I \in \Omega_i} f_I \leq 1, \; i = 1, ..., |\mathbf{D}|, \tag{3.43}$$

where f_I^s and f_I^e have value 1 if I is at the start and end frames in $\mathbf{T}^{(k)}$ respectively. $f_I = 1$ encodes the fact whether I is in $\mathbf{T}^{(k)}$ and $f_{IJ} = 1$ if I and J are adjacent in any trajectory association variable. By using (3.32), the terms in (3.40) are changed by the following equations,

$$
\begin{aligned}
c_I^s &= g_I^s(\mu_I^{*(k)}, \Sigma_I^{*(k)}), \\
c_I^e &= g_I^e(\mu_I^{*(k)}, \Sigma_I^{*(k)}), \\
c_I &= g_I(\mu_I^{*(k)}, \Sigma_I^{*(k)}), \\
c_{IJ} &= g_{IJ}(\mu_I^{*(k)}, \mu_J^{*(k)}, \Sigma_I^{*(k)}, \Sigma_J^{*(k)}).
\end{aligned} \tag{3.44}
$$

As shown in (3.41-3.43), there exist three constraints: *unit capacity*, *flow conservation*, *non-overlap detection* constraints. The *unit capacity* in (3.41) means that the maximum amount of flow at every edge is 1. The *flow conservation* constraint in (3.42) means that the amount of flow incoming to I is the same as the amount of flow outgoing from I. Lastly, *non-overlap detection* constraint in (3.43) is based on the fact that each trajectory assignment variable does not share the same detection, which is defined in (3.5). The conflict index set Ω_i of detection d_i is defined by a set of all Is that have the same i as an element, i.e. $\{I \in \Omega |\ s.t.\ \exists i\ \in I\}$. *non-overlap detection* constraint in (3.43) is that the amount of flows passing into every conflict index $I \in \Omega_i$ is at most 1, which arises specially in multi-camera setting.

If the *non-overlap detection* constraint is satisfied for all f (e.g. a single camera case), the integer linear program in (3.40) is equivalent to the minimum cost flow problem [13]. The solution of the minimum cost flow problem is found in a polynomial time by using push relabeling algorithm [13] or successive shortest path algorithm [12, 11]. With *non-overlap detection* constraint, the integer linear program can be solved by the branch-and-bound algorithm implemented with Gurobi Optimization Library [57].

3.3 Appendix

3.3.1 Derivation of equation (3.12)

Given the $(k-1)$-th trajectory assignment $\mathbf{T}^{*(k-1)}$, we show that the problem of finding q that minimizes KL divergence equals to the problem of finding means and covariance matrices that maximize the lower bound $\mathcal{L}$. Starting from the problem of finding $q^{(k)}(\mathbf{X})$ given the $\mathbf{T}^{*(k-1)}$,

$$q^{*(k)}(\mathbf{X}) = \underset{q^{(k)}(\mathbf{X})}{\arg\min}\, KL\left(q^{(k)}(\mathbf{X})||p(\mathbf{X}|\mathbf{T}^{*(k-1)}, \mathbf{D})\right). \tag{3.45}$$

By the definition of KL divergence,

$$KL\left(q^{(k)}(\mathbf{X})||p(\mathbf{X}|\mathbf{T}^{*(k-1)}, \mathbf{D})\right) = -\int_{\mathbf{X}} q^{(k)}(\mathbf{X}) \cdot \ln \frac{p(\mathbf{X}|\mathbf{T}^{*(k-1)}, \mathbf{D})}{q(\mathbf{X})}, \tag{3.46}$$

$$= -\int_{\mathbf{X}} q^{(k)}(\mathbf{X}) \cdot \ln \frac{p(\mathbf{X}, \mathbf{T}^{*(k-1)}|\mathbf{D})}{q(\mathbf{X})} + \ln p(\mathbf{T}^{*(k-1)}|\mathbf{D}) \cdot \int_{\mathbf{X}} q(\mathbf{X}), \tag{3.47}$$

$$= -\int_{\mathbf{X}} q^{(k)}(\mathbf{X}) \cdot \ln \frac{p(\mathbf{X}, \mathbf{T}^{*(k-1)}|\mathbf{D})}{q(\mathbf{X})} + \ln p(\mathbf{T}^{*(k-1)}|\mathbf{D}). \tag{3.48}$$

Assuming $q(\mathbf{X})$ follows a Gaussian distribution, $q(\mathbf{X})$ is represented by its mean vector and covariance matrix;

$$\int_{\mathbf{X}} q^{(k)}(\mathbf{X}) \cdot \ln \frac{p(\mathbf{X}, \mathbf{T}^{*(k-1)}|\mathbf{D})}{q(\mathbf{X})} = \mathcal{L}(\widehat{\mu}^{(k)}, \widehat{\Sigma}^{(k)}, \mathbf{T}^{*(k-1)}). \tag{3.49}$$

Since the term $\ln p(\mathbf{T}^{*(k-1)}|\mathbf{D})$ in is a constant w.r.t $\mathbf{X}$, the problem in (3.45) is equivalent to

$$\widehat{\mu}^{*(k)}, \widehat{\Sigma}^{*(k)} = \underset{\widehat{\mu}^{(k)}, \widehat{\Sigma}^{(k)}}{\arg\min} -\mathcal{L}(\widehat{\mu}^{(k)}, \widehat{\Sigma}^{(k)}, \mathbf{T}^{*(k-1)}). \tag{3.50}$$

3.3.2 Derivation of equation (3.27-3.32)

Starting from the definition of the lower bound,

$$-\mathcal{L}(q(\mathbf{X}), \mathbf{T}) = -\int_{\mathbf{X}} q(\mathbf{X}) \cdot \ln \frac{p(\mathbf{T}, \mathbf{X}|\mathbf{D})}{q(\mathbf{X})}. \tag{3.51}$$

According to the Bayes rule,

$$-\mathcal{L}(q(\mathbf{X}),\mathbf{T}) = -\int_{\mathbf{X}} q(\mathbf{X})\cdot(\ln p(\mathbf{T},\mathbf{X}) + \ln p(\mathbf{D}|\mathbf{T},\mathbf{X}) - \ln p(\mathbf{D})) + \int_{\mathbf{X}} q(\mathbf{X})\ln q(\mathbf{X}), \tag{3.52}$$

where the likelihood and prior are defined in (3.14-3.17) as follows:

$$\ln p(\mathbf{D}|\mathbf{X},\mathbf{T}) = \sum_{\tau\in\mathbf{T}\setminus\{\tau_0\}}\sum_{I\in\tau}(\ln\psi_{tar}(D_I,\mathbf{x}_I) - \ln\psi_{fak}(D_I)), \tag{3.53}$$

$$\ln p(\mathbf{X},\mathbf{T}) = \sum_{\tau\in\mathbf{T}\setminus\{\tau_0\}}\left(\ln\psi_s(\mathbf{x}_{I^s}) + \ln\psi_e(\mathbf{x}_{I^e}) + \sum_{I,J\in adj(\tau)}\psi_{link}(\mathbf{x}_I,\mathbf{x}_J)\right). \tag{3.54}$$

Substituting (3.53,3.54) to (3.52), the lower bound is decomposed into a likelihood+prior term $\mathcal{G}$, an entropy term $\mathcal{H}$, and a constant,

$$-\mathcal{L}(q(\mathbf{X}),\mathbf{T}) = \mathcal{G}(q(\mathbf{X}),\mathbf{T}) - \mathcal{H}(q(\mathbf{X})) + \ln p(\mathbf{D}). \tag{3.55}$$

Assuming that $q(\mathbf{X})$ is factorized as $\prod_{I\in\Omega} q_I(\mathbf{x}_I)$, $\mathcal{G}$ and $\mathcal{H}$ are defined as

$$\mathcal{G}(q(\mathbf{X}),\mathbf{T}) = \sum_{\tau\in\mathbf{T}\setminus\{\tau_0\}}\left(g^s_I(q_{I^s}(\mathbf{x}_{I^s})) + g^e_I(q_{I^e}(\mathbf{x}_{I^e})) + \sum_{I\in\tau} g_I(q_I(\mathbf{x}_I)) + \sum_{I,J\in adj(\tau)} g_{IJ}(q_I(\mathbf{x}_I),q_J(\mathbf{x}_J))\right), \tag{3.56}$$

$$\mathcal{H}(q(\mathbf{X})) = \sum_{I\in\Omega} h_I(q_I(\mathbf{x}_I)), \tag{3.57}$$

where

$$
\begin{aligned}
g_I^s(q_{I^s}(\mathbf{x}_{I^s})) &= -\int q_{I^s}(\mathbf{x}_{I^s}) \cdot \ln \psi_s(\mathbf{x}_{I^s}) d\mathbf{x}_{I^s} \\
&= \int q_{I^s}(\mathbf{x}_{I^s}) \cdot (\lambda_{fse} c_{fse} + \lambda_{eez} c_{eez}) d\mathbf{x}_{I^s}, \qquad (3.58) \\
g_I^e(q_{I^e}(\mathbf{x}_{I^e})) &= -\int q_{I^e}(\mathbf{x}_{I^e}) \cdot \ln \psi_e(\mathbf{x}_{I^e}) d\mathbf{x}_{I^e} \\
&= \int q_{I^e}(\mathbf{x}_{I^e}) \cdot (\lambda_{fse} c_{fse} + \lambda_{eez} c_{eez}) d\mathbf{x}_{I^e}, \qquad (3.59) \\
g_I(q_I(\mathbf{x}_I)) &= -\int q_I(\mathbf{x}_I) \cdot (\ln \psi_{tar}(D_I, \mathbf{x}_I) - \ln \psi_{fak}(D_I))\, d\mathbf{x}_I \\
&= \int q_I \cdot (\lambda_{obs} c_{obs} + \lambda_{mid} c_{mid} + \lambda_{fak} c_{fak}) d\mathbf{x}_I, \qquad (3.60) \\
g_{IJ}(q_I(\mathbf{x}_I), q_J(\mathbf{x}_J)) &= -\iint q_I(\mathbf{x}_I) q_J(\mathbf{x}_J) \psi_{link}(\mathbf{x}_I, \mathbf{x}_I) d\mathbf{x}_I d\mathbf{x}_J \\
&= \iint q_I q_J (\lambda_{mot} c_{mot} + \lambda_{jmp} c_{jmp}) d\mathbf{x}_I d\mathbf{x}_J, \qquad (3.61) \\
h_I(q_I(\mathbf{x}_I)) &= -\int q_I(\mathbf{x}_I) \cdot \ln(q_I(\mathbf{x}_I)) d\mathbf{x}_I. \qquad (3.62)
\end{aligned}
$$

Using the definitions of probability model in Section 3.2.1, the seven integral terms in (3.58 - 3.61) are given by

$$\lambda_{obs} \int q_I(\mathbf{x}_I) \cdot c_{obs}(D_I, \mathbf{x}_I)\, d\mathbf{x}_I = \frac{\lambda_{obs}}{|I|} \cdot E(\sum_{i \in I} \|A_i \mathbf{x}_I - bi\|^2), \qquad (3.63)$$

$$\lambda_{fak} \int q_I(\mathbf{x}_I) \cdot c_{fak}\, d\mathbf{x}_I = \lambda_{fak} \cdot |I|\,, \qquad (3.64)$$

$$\lambda_{mid} \textstyle\int q_I(\mathbf{x}_I) \cdot c_{mid}(D_I, \mathbf{x}_I)\, d\mathbf{x}_I = \lambda_{mid} \cdot \big(\int q_I(\mathbf{x}_I) \cdot v(\mathbf{x}_I)\, d\mathbf{x}_I - |I|\big), \qquad (3.65)$$

$$\lambda_{mot} \textstyle\iint q_I(\mathbf{x}_I) q_J(\mathbf{x}_J) \cdot c_{mot}(\mathbf{x}_I, \mathbf{x}_J)\, d\mathbf{x}_I d\mathbf{x}_J = \lambda_{mot} \left(\frac{|I|+|J|}{2}\right) E(\|\mathbf{x}_I - \mathbf{x}_J\|^2), \qquad (3.66)$$

$$\lambda_{jmp} \int q_I(\mathbf{x}_I) \cdot c_{jmp}\, d\mathbf{x}_I = \lambda_{jmp} \left(\frac{|I| + |J|}{2}\right) \Delta t_{IJ}, \qquad (3.67)$$

$$\lambda_{fse} \int q_I(\mathbf{x}_I) \cdot c_{fse}\, d\mathbf{x}_I = \lambda_{fse} \left(\frac{|I|}{2}\right), \qquad (3.68)$$

$$\lambda_{eez} \int q_I(\mathbf{x}_I) \cdot c_{eez}(\mathbf{x}_I)\, d\mathbf{x}_I = \lambda_{eez} \left(\frac{|I|}{2}\right) \left(\int q_I(\mathbf{x}_I) \cdot u(\mathbf{x}_I)\, d\mathbf{x}_I\right). \qquad (3.69)$$

Assuming that each $\mathbf{x}$ follows a Gaussian distribution, the expectation terms in (3.63),

(3.66) and the entropy term are calculated by

$$E(\sum_{i\in I}\|A_i\mathbf{x}_I - b_i\|^2) = \sum_{i\in I}\Big(tr(A_i\Sigma_I A_i^T) + \|A_i\mu_I - b_i\|^2\Big), \quad (3.70)$$

$$E(\|\mathbf{x}_I - \mathbf{x}_J\|^2) = tr(\Sigma_I + \Sigma_J) + \|\mu_I - \mu_J\|^2, \quad (3.71)$$

$$h_I(q_I(\mathbf{x}_I)) = \frac{1}{2}\ln\Big((2\pi e)^3\cdot|\Sigma_I|\Big). \quad (3.72)$$

where tr denotes the trace operation.

Note that the integrals of v in (3.65) and u in (3.69) are intractable. Instead, we use the zeroth-order Taylor-series approximation of each function at the recently obtained mean μ_I^*, i.e., $v(\mathbf{x}_I) \approx v(\mu_I^*)$, $u(\mathbf{x}_I) \approx u(\mu_I^*)$. By substituting (3.63-3.72) to (3.58-3.62), we can express the lower bound $\mathcal{L}$ as a function of means and covariance matrices of $\mathbf{x}$. The lower bound in (3.55) is rewritten by

$$-\mathcal{L}(\widehat{\mu},\widehat{\Sigma},\mathbf{T}) = \mathcal{G}(\widehat{\mu},\widehat{\Sigma},\mathbf{T}) + \mathcal{H}(\widehat{\Sigma}) + \ln p(\mathbf{D}). \quad (3.73)$$

Similarly, the terms in (3.58-3.62) become

$$g_I^s = g_I^e = \lambda_{fse}\cdot\frac{|I|}{2} + \lambda_{eez}\cdot\frac{|I|}{2}u(\mu_I^*), \quad (3.74)$$

$$g_I(\mu_I,\Sigma_I) = \lambda_{obs}\cdot\left(\frac{1}{|I|}\sum_{i\in I}tr(A_i\Sigma_I A_i^T) + \|A_i\mu_I - b_i\|^2\right) + \lambda_{mid}\cdot(v(\mu_I^*) - |I|) + \lambda_{fak}\cdot|I|\,, \quad (3.75)$$

$$g_{IJ}(\mu_I,\Sigma_I,\mu_J,\Sigma_J) = \lambda_{mot}\cdot\left(\frac{|I|+|J|}{2}\right)\Big(tr(\Sigma_I+\Sigma_J) + \|\mu_I-\mu_J\|^2\Big) + \lambda_{jmp}\cdot\left(\frac{|I|+|J|}{2}\right)\Delta t_{IJ}, \quad (3.76)$$

$$h_I(\Sigma_I) = \frac{1}{2}\ln\Big((2\pi e)^3\cdot|\Sigma_I|\Big). \quad (3.77)$$

3.3.3 Deriving optimal mean and covariance matrix (3.33-3.35)

For each τ, we derive the optimal mean $\widehat{\mu}_\tau^*$ that maximize $\mathcal{L}$. The terms related to $\widehat{\mu}_\tau$ are summarized as the following weighted least square problem;

$$\widehat{\mu}_\tau^* = \arg\min_{\widehat{\mu}_\tau}\ \lambda_{obs}\cdot\sum_{I\in\tau}\frac{1}{|I|}\sum_{i\in I}\|A_i\mu_I - \mathbf{b}_i\|^2 + \lambda_{mot}\cdot\sum_{I,J\in adj(\tau)}\left(\tfrac{|I|+|J|}{2}\right)\cdot\|\mu_I-\mu_J\|^2. \quad (3.78)$$

We show that each term can be rewritten by a matrix form. The inner summation of the first term is rewritten as

$$\frac{\lambda_{obs}}{|I|}\sum_{i\in I}\|A_i\mu_I-\mathbf{b}_i\|^2=\left\|(W_I^r)^{1/2}\left(\widehat{A}_I\mu_I-\widehat{\mathbf{b}}_I\right)\right\|^2 \tag{3.79}$$

where

$$W_I^r=\frac{\lambda_{obs}}{|I|}\cdot\mathbb{I}\in\mathbb{R}^{3|I|\times 3|I|},\ \ \widehat{A}_I=\begin{bmatrix}A_{i_1}\\ \vdots\\ A_{i_{|I|}}\end{bmatrix}\in\mathbb{R}^{3|I|\times 3},\ \ \widehat{\mathbf{b}}_I=\begin{bmatrix}\mathbf{b}_{i_1}\\ \vdots\\ \mathbf{b}_{i_{|I|}}\end{bmatrix}\in\mathbb{R}^{3|I|\times 1}. \tag{3.80}$$

Substituting (3.79) to (3.78), the first term is also rewritten as the following equation;

$$\lambda_{obs}\cdot\sum_{I\in\tau}\frac{1}{|I|}\sum_{i\in I}\|A_i\mu_I-\mathbf{b}_i\|^2=\left\|\widehat{W}_{obs}^{1/2}(\widehat{A}\mu_\tau-\widehat{\mathbf{b}})\right\|^2, \tag{3.81}$$

where μ_τ is a concatenated vector of means s.t. $\mu_\tau=\left[\mu_{I_1}^{\mathsf{T}}\dots\mu_{I_{|\tau|}}^{\mathsf{T}}\right]^{\mathsf{T}}$ and

$$\widehat{W}_{obs}=\begin{bmatrix}W_{I_1}^r & & \\ & \ddots & \\ & & W_{I_{|\tau|}}^r\end{bmatrix},\ \widehat{A}=\begin{bmatrix}\widehat{A}_{I_1} & & \\ & \ddots & \\ & & \widehat{A}_{I_{|\tau|}}\end{bmatrix},\ \widehat{\mathbf{b}}_I=\begin{bmatrix}\mathbf{b}_{I_1}\\ \vdots\\ \mathbf{b}_{I_{|\tau|}}\end{bmatrix}. \tag{3.82}$$

On the other hand, the second term is rewritten as

$$\lambda_{mot}\cdot\sum_{I,J\in adj(\tau)}\left(\frac{|I|+|J|}{2}\right)\cdot\|\mu_I-\mu_J\|^2=\left\|\widehat{W}_{mot}^{1/2}(\widehat{C}\cdot\mu_\tau)\right\|^2, \tag{3.83}$$

where $W_I^m=\lambda_{mot}\left(\frac{|I|+|J|}{2}\right)\cdot\mathbb{I}_{3\times 3}$ such that J denote the index adjacent to I,

$$\widehat{W}_{mot}=\begin{bmatrix}W_{I_1}^m & & \\ & \ddots & \\ & & W_{I_{|\tau|-1}}^m\end{bmatrix}\in\mathbb{R}^{3(|\tau|-1)\times 3(|\tau|-1)},$$

$$\widehat{C}=\begin{bmatrix}-\mathbb{I}_{3\times 3} & \mathbb{I}_{3\times 3} & \\ & \ddots & \\ & -\mathbb{I}_{3\times 3} & \mathbb{I}_{3\times 3}\end{bmatrix}\in\mathbb{R}^{3(|\tau|-1)\times 3|\tau|}. \tag{3.84}$$

Substituting (3.81) and (3.83) to (3.78), we have

$$\widehat{\mu}_\tau^* = \arg\min_{\mu} \left\| \widehat{W}_{obs}^{1/2}(\widehat{A}\mu - \widehat{\mathbf{b}}) \right\|^2 + \left\| \widehat{W}_{mot}^{1/2}(\widehat{C}\mu) \right\|^2, \tag{3.85}$$

which can be solved in a closed-form;

$$\widehat{\mu}_\tau^* = (\widehat{A}^\mathsf{T}\widehat{W}_{obs}\widehat{A} + \widehat{C}^\mathsf{T}\widehat{W}_{mot}\widehat{C})^{-1}\widehat{A}^\mathsf{T}\widehat{W}_{obs}\widehat{\mathbf{b}}. \tag{3.86}$$

When the length of τ is 1, the second term $\left\| \widehat{W}_{mot}^{1/2}(\widehat{C}\mu) \right\|^2$ vanishes. Thus $\widehat{\mu}_\tau^*$ becomes

$$\widehat{\mu}_\tau^* = (\widehat{A}^\mathsf{T}\widehat{A})^{-1}\widehat{A}^\mathsf{T}\widehat{\mathbf{b}}. \tag{3.87}$$

Next, we derive the optimal covariance Σ_I^* that maximizes $\mathcal{L}$. In the lower bound $\mathcal{L}$, the terms related to Σ_I are three;

$$\frac{\lambda_{obs}}{|I|}\sum_{i\in I} tr(A_i\Sigma_I A_i^\mathsf{T}) + \frac{1}{2}\lambda_{mot}\cdot w_{mot}\cdot tr(\Sigma_I) - \frac{1}{2}\ln|\Sigma_I|, \tag{3.88}$$

where

$$w_{mot} = \begin{cases} |I_{prev}| + |I|\,, & \text{if } I \text{ is a start index,} \\ |I_{next}| + |I|\,, & \text{if } I \text{ is an end index,} \\ |I_{prev}| + 2\cdot|I| + |I_{next}|\,, & \text{otherwise.} \end{cases} \tag{3.89}$$

Taking partial derivative of $\mathcal{L}$ w.r.t Σ_I, we have the following equation;

$$\frac{\partial\mathcal{L}}{\partial\Sigma_I} = \frac{\partial}{\partial\Sigma_I}\left(\frac{\lambda_{obs}}{|I|}\sum_{i\in I} tr(A_i\Sigma_I A_i^\mathsf{T}) + \frac{1}{2}\lambda_{mot}\cdot w_{mot}\cdot tr(\Sigma_I) - \frac{1}{2}\ln|\Sigma_I|\right). \tag{3.90}$$

Using the matrix derivative lemmas,

$$\begin{aligned} &\frac{\partial\sum_{i\in I} tr(A_i\Sigma_I A_i^\mathsf{T})}{\partial\Sigma_I} = \frac{\partial\sum_{i\in I} tr(\Sigma_I A_i^\mathsf{T}A_i)}{\partial\Sigma_I} = \sum_{i\in I} A_i^\mathsf{T}A_i, \\ &\frac{\partial tr(\Sigma_I)}{\partial\Sigma_I} = I_{3\times3}, \quad \frac{\partial\ln|\Sigma_I|}{\partial\Sigma_I} = \Sigma_I^{-1}, \end{aligned} \tag{3.91}$$

the derivative becomes

$$\frac{\partial\mathcal{L}}{\partial\Sigma_I} = \frac{\lambda_{obs}}{|I|}\sum_{i\in I} A_i^\mathsf{T}A_i + \frac{1}{2}\lambda_{mot}\cdot w_{mot}\cdot I_{3\times3} - \frac{1}{2}\Sigma_I^{-1} = 0. \tag{3.92}$$

Taking an inverse, we finally have the equation,

$$\Sigma_I^* = \left(2 \cdot \frac{\lambda_{obs}}{|I|} \sum_{i \in I} A_i^{\mathsf{T}} A_i + \lambda_{mot} \cdot w_{mot} \cdot \mathbb{I}_{3\times 3} \right)^{-1}. \tag{3.93}$$

3.3.4 Definition of A and b in (3.22)

Let a back-projection line $\Phi(d)$ of detection d be $\frac{x-\mathsf{c}}{\mathsf{a}} = \frac{y-\mathsf{d}}{\mathsf{b}} = z$, where the constants $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}$ are constant parameters determined by using a 2D point of the detection d and its corresponding camera projection matrix. We define the 3D Euclidean distance at the same z-value as the distance between a back-projection line Φ and a 3D point $\mathbf{x}$. If the 3D position $\mathbf{x} = (x', y', z')^{\mathsf{T}}$, the point $\mathbf{x}_\Phi$ at $z = z'$ in the back-projection line is $(\mathsf{a}z' + \mathsf{c},\ \mathsf{b}z' + \mathsf{d},\ z')^{\mathsf{T}}$. Finally, the distance between $\mathbf{x}_\Phi$ and $\mathbf{x}$ can be rewritten as a form of linear equation, i.e.,

$$\begin{aligned} \|\mathbf{x} - \mathbf{x}_\Phi\| &= \left\| (x', y', z')^{\mathsf{T}} - (\mathsf{a}z' + \mathsf{c},\ \mathsf{b}z' + \mathsf{d},\ z')^{\mathsf{T}} \right\| && (3.94) \\ &= \left\| (x' - \mathsf{a}z' - \mathsf{c}, y' - \mathsf{b}z' - \mathsf{d}, 0)^{\mathsf{T}} \right\| && (3.95) \\ &= \|A\mathbf{x} - \mathbf{b}\|, && (3.96) \end{aligned}$$

where $A = \begin{pmatrix} 1 & 0 & -\mathsf{a} \\ 0 & 1 & -\mathsf{b} \\ 0 & 0 & 0 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} \mathsf{c} \\ \mathsf{d} \\ 0 \end{pmatrix}$.

Chapter 4

Experiments

4.1 Datasets

4.1.1 PETS 2009

The IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) has provided several datasets in various surveillance setting since the year 2000. Each PETS dataset was used as a benchmark dataset, such as tracking, detection and people counting. Among the PETS datasets the *PETS 2009* dataset proposed in year 2009 is still actively used in the computer vision community. Spcifically, it has been widely used as a set of benchmark data for tracking multiple people in both single camera [58, 18, 17, 35] and a multi-camera setup [38, 36, 54, 41]. The *PETS 2009* dataset was recorded at the campus of the University of Reading, UK. A total of eight calibrated cameras monitored the overlapping space and were synchronized in time. There exists a static obstacle (e.g. a light pole in the middle) which makes consistent labeling hard.

The *PETS 2009* dataset consists of three sequences for tracking scenario with different density of people; from low density, *S2.L1*, *S2.L2*, and *S2.L3*. In the case of the *S2.L1* sequence, eight cameras were recorded, and the *S2.L2*, and *S2.L3* sequences were recorded in four cameras. However, The PETS organizers did not disclose the

(a) *S2.L1* sequence

(b) *S2.L2* sequence

(c) *S2.L3* sequence

Figure 4.1: An overview of the *PETS 2009* dataset.

video corresponding to the camera 2 for *S2.L1* for reasons of cross validation. Some example frames from three sequences are shown in Figure 4.1. In each sequence, people move in a relatively constant direction from relatively irregular *S2.L1* to *S2.L3*.

The PETS 2009 organizers do not publish the ground truth because it is planned as a challenge. We have utilized the bounding boxes annotated by Milan *et al.* [58] [1]. To evaluate performance in a 3D space, we moved 2D foot positions (assumed bottom center of each bounding box) to a 3D ground plane using camera calibration information.

4.1.2 PSN-University

Existing multi-camera datasets, such as the PETS2009 dataset, assume a situation where the full body is visible. In reality, however, there are many cases where people

[1] http://www.milanton.de/data/

(a) *standing* sequence

(b) *sitting* sequence

(c) *standing & sitting* sequence

Figure 4.2: An overview of the *PSN-University* dataset.

are not fully visible to the whole body by various structures. Contrast to the previous multi-camera datasets, we have constructed a new multi-camera dataset that includes both occlusion by static obstacles (e.g. chairs, desks) as well as mutual occlusion by people. The ultimate goal of the proposed new dataset is to solve the 3W problem (Who, Where, What). That is, we create a perception sensor network (PSN) that knows who is doing what and where. The dataset was recorded in a university lecture room where students move around or sit. Thus the dataset is called PSN-University. The goal of this dataset is to localize 3D positions of all students, track them, identify their faces, and recognize their actions.

For 3D localizing and tracking (Where) task, the main purpose is to estimate 3D locations of heads. The PSN-University dataset consists of three sequences of *standing*, *sitting*, and *standing & sitting* (See Figure 4.2). The main challenge is to handle the different heights and poses (e.g. standing, sitting) of the students. The video was cap-

Dataset	Sequence	# Frames	FrameRate	Resolution	Density	# Cameras
PETS2009	*S2.L1*	795	7	med	med	8
PETS2009	*S2.L2*	436	7	med	high	4
PETS2009	*S2.L3*	240	7	med	high	4
PSN-University	*standing*	177	3	high	med	4
PSN-University	*sitting*	292	3	high	med	4
PSN-University	*standing & sitting*	330	3	high	med	4

Table 4.1: Summary of datasets

tured using four synchronized cameras with a 10 mega-pixel 3648×2752 resolution at 3 fps. This is for high-level analysis such as face recognition and action recognition.

For the ground truth trajectories, we annotate the human head of each camera. For accurate head coordinates, we put markers on people's heads. In order to create the 3D ground truth trajectory, we used the calibration information to estimate each person's 3D head coordinates (See the forth column of Figure 4.2). The 3D head coordinates of each person were triangulated by several cameras. All detections, the ground truth of 3D trajectories, evaluation scripts and camera calibrations used in our experiments are publicly available[2].

4.2 Evaluation Metrics

To quantitatively evaluate localization and tracking performance, we adopted widely used metrics for multiple target tracking evaluation: MOTA, MOTP [59] and MT, ML, PT, IDS, FM [60].

MOTA, MOTP. The *multiple object tracking accuracy* (MOTA) evaluates the overall

[2]http://sites.google.com/site/byeonmoonsub/home/mcmtt

Table 4.2: Parameters of mixed multidimensional assignment approach

Dataset	λ_{rec}	λ_{mot}	λ_{mid}	λ_{tse}	λ_{tfm}	λ_{fpt}
PETS 2009	$1/20^2$	$1/50^2$	10^2	10^2	20^2	12.5^2
PSN-Univ.	$1/20^2$	$1/80^2$	12^2	10^2	20^2	12.5^2

performance of a multi-target tracking by looking at missed targets, false alarms, and identity switches. The multiple object tracking precision (MOTP), on the other hand, averages the localization error between the estimated and the ground truth trajectory. We calculated the MOTA and the MOTP using a 3D world coordinate. The matching criterion between the estimated trajectory and the ground truth trajectory is the distance between them in a 3D world coordinate. The estimated trajectory matches the ground truth only if the distance is less than a threshold (set to 1 meter in our experiments).

MT, ML, PT, IDS, FM. Each abbreviation stands for the number of trajectories mostly tracked (MT) and mostly lost (ML); the number of fragments (FM); and identity switches (IDS). Each ground truth trajectory is classified as MT if it is successfully tracked over 80%, ML if tracked less than 20%.

Recall, Precision. Other metric including recall (Rcll) and precision (Prcn) are also presented.

4.3 Results and Discussion

4.3.1 Mixed Multidimensional Assignment Approach

Parameter Settings

As we discussed in Section 2.1.2, the parameters for cost function need to be set. We empirically found each parameters but the most of them were fixed to prevent overfit a specific sequence. Table 4.2 shows the parameter setting for weights of each cost term in (2.12). Although two dataset have different properties such as resolution, in-

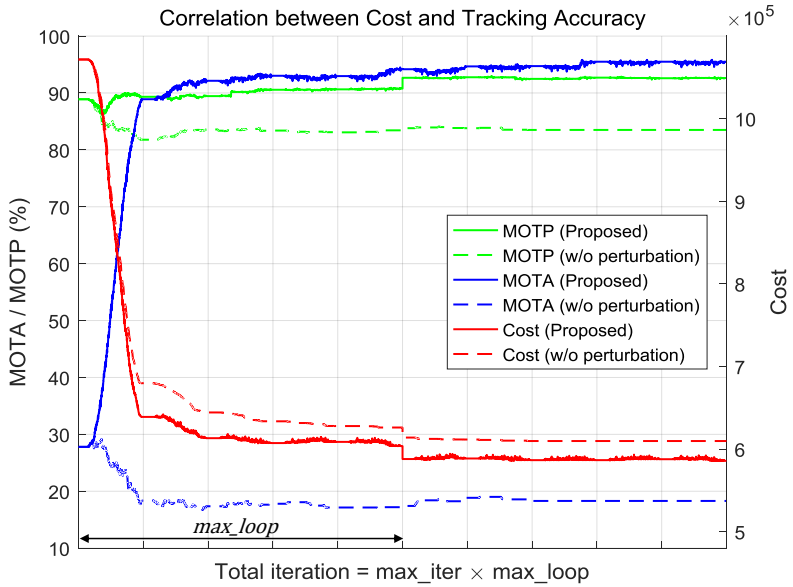


Figure 4.3: Correlation between a cost value and tracking accuracy on *PSN-University sitting* sequence at FNR 50%. The proposed scheme without perturbation moves to a solution decreasing a cost value (red dashed). To escape from a local basin, we adopts perturbation that allows a cost increase (red solid). The proposed scheme with perturbation outperforms without perturbation (blue, green).

door/outdoor, and the number of targets, the only two parameters λ_{mot} and λ_{mid} are different; λ_{mot} related to moved distance per a frame is trivially dependent on a framerate of recoded video. λ_{mid} is relevant to the density of people in the scene, since it penalized missed detections. In addition, we set $r = 300$ mm and $\alpha_m = 0.5$ for all experiments, which indicates the diameter of a person and a weight of motion smoothness defined in (2.13) and (2.15) respectively. The maximum number of iterations max_iter and max_loop is set to 2 and $5 \times (\mathsf{F} \times (2^{\mathsf{K}} - 2) \times 2)$ respectively.

Investigation of Perturbation & Convergence

Purturbation. To investigate the effect of purturbation, we used the *PSN-University sitting* sequence at FNR 50% (see Figure 4.3). For this purpose, two optimization

Table 4.3: Comparison the proposed splitting/re-mering strategy with the optimal values.

Trajectory #	T	K	Detection#	Optimal Value	Proposed
885	5	2	10	92.6401	92.6401
1770	5	2	20	91.5869	91.5869
910	5	2	16	135.4429	135.4429
11186	5	2	20	84.9256	84.9256
54722	5	2	20	134.9607	134.9607
80200	5	2	20	170.1519	177.5030
180665	5	3	30	150.7589	150.7589
291074	5	3	30	134.6287	134.6287
148436	5	3	30	148.1003	148.1003
826348	5	4	29	163.0143	163.0143

schemes were evaluated depending on whether purturbation is used or not. The proposed scheme without perturbation moves to a solution decreasing a cost value only (red dashed), whereas the proposed scheme with perturbation allows a cost increase (red solid). Nevertheless, the proposed scheme with perturbation finally found a lower cost solution. There was a significant difference between tracking performances by the two optimization schemes. In this exepriment, the proposed scheme without perturbation was stuck in a local basin and finally converged to the local basin (green and blue dashed). In contrast, the proposed scheme with perturbation significantly outperforms the proposed scheme without perturbation. These results support that the perturbation is helpful to escape from a local basin.

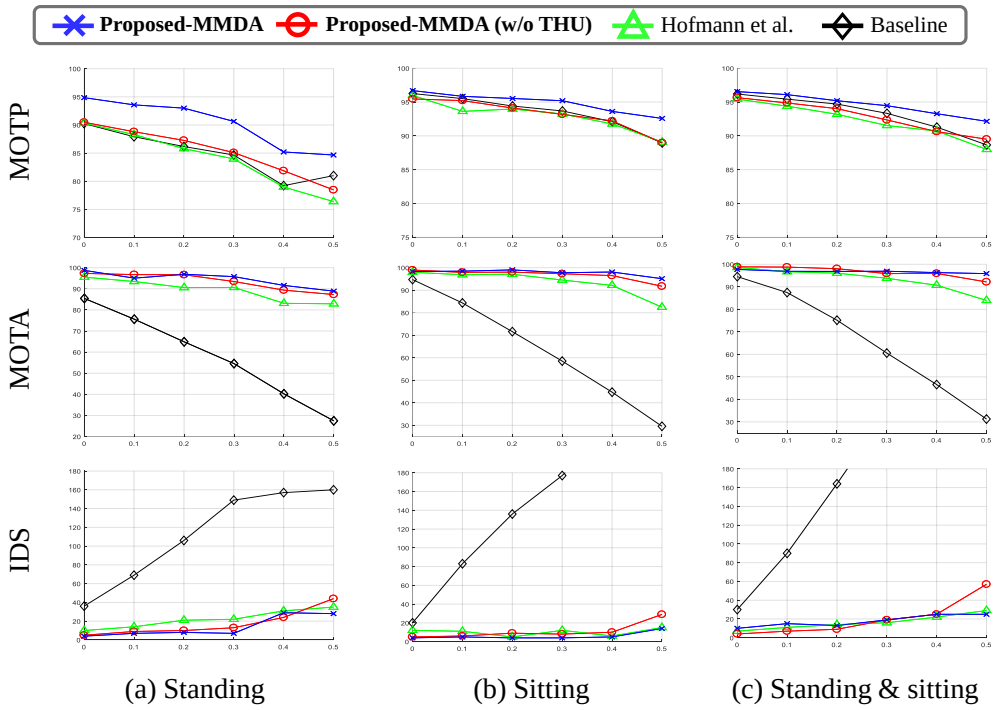
Convergence. We have conducted experiments to illustrate the convergence of the splitting/re-merging. To evaluate the convergence, we made *Synthesized* dataset where optimums can be calculated. The *Synthesized* dataset was constructed by randomly sampling from the *PSN-University standing* sequence. Since the MDA problem is NP-hard, the solution space of the MDA problem exponentially increases when the number of cameras, frames, and people increases. In our experiment, we set $\mathsf{K} = 4, \mathsf{F} = 5$

and assume two people are moving. Using the *Synthesized* dataset, we evaluated the convergence to the optimum by our method. To calculate the optimum, we generated all possible assignment matrices and trajectory hypotheses, calculated their costs, and solved the MDA problem by a BIP solver[3]. Table 4.3 shows the comparison of our solutions with the optimal values. Note that the number of possible assignments exponentially increases with order of $O((\mathsf{P}+1)^{\mathsf{KF}})$ in K cameras, F frames, and P people case although we restrict possible candidates whose locations are close to each other. In most cases, our method finds the optimum by iteratively improving the initial solution as shown in Table 4.3.

Quantitative and Qualitative Evaluation

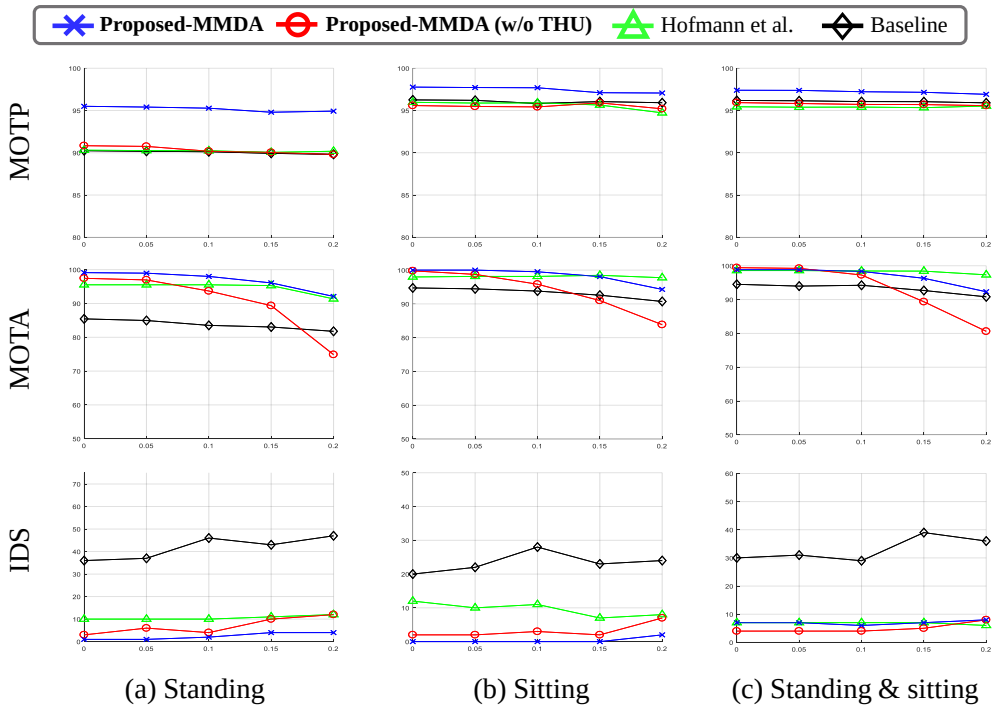
We report the results of four methods: "Proposed-MMDA", "Proposed-MMDA (w/o THU)", [38], "Baseline". For a fair comparison, they are evaluated with the same detection results as an input. We report the tracking result of greedy spatial and temporal data association method as "Baseline" in our experiments (see Details in Sec 2.2.3). The method of Hofmann *et al.* [38] has a difficulty in directly applying head detection results as an input since the algorithm needs to be set 3D coordinates of each combination of detections. Although the 3D coordinate of the person detected by a single camera can be estimated with ground plane assumption in full body case, it is difficult to determine its location in head case because the height of the person is unknown. Therefore, we assumed that the person has an approximate height (1700 millimeters in our experiments). Instead of the height assumption, we have also experimented with ignoring the case detected in one camera only. Unfortunately, resulting trajectories are frequently split because in many cases people are detected in one camera when occlusions occur. In order to show the importance of the trajectory hypothesis update process, we also reported the tracking performance when the trajectory hypothesis update process was not performed (denoted as Proposed-MMDA (w/o THU)).

[3] we used the Gurobi optimization library at http://www.gurobi.com



(a) Standing (b) Sitting (c) Standing & sitting

Figure 4.4: Quantitative evaluation for MOTA, IDS and Recall in the *PSN-University* dataset for increasing false negative rate (FNR).



(a) Standing (b) Sitting (c) Standing & sitting

Figure 4.5: Quantitative evaluation for MOTA, IDS and Recall in the *PSN-University* dataset for increasing false positive rate (FPR).

PSN-University. Two sets of experiments on *PSN-University* were conducted: using hand-labeled detections and state-of-the-art detectors [50, 61]. First, we did not use any detectors to decouple the tracking performance from the detection performance. Instead, to evaluate the tracking performance depending on the detection performance, we added false negatives (missing detections) and false positives (false detections) to the hand-labeled detections. To simulate false negatives, we removed true detections randomly, where we changed the false negative rate (FNR) from 0% to 50%. To create false positives, we made false detections at random locations chosen uniformly, where we changed the false positive rate from 0% to 20% of true detections.

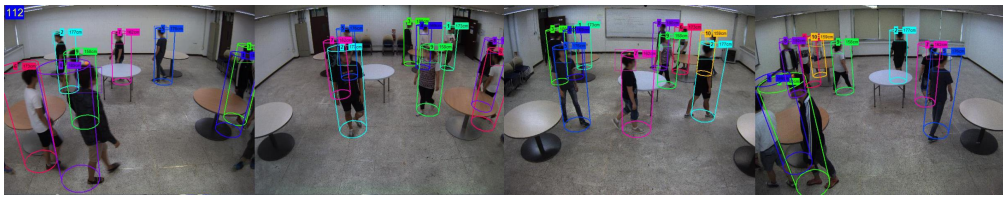
We plot the tendency of MOTP, MOTA, and IDS by increasing FNR (see Figure 4.4) and FPR (see Figure 4.5). Note that the "Baseline" method in 0% FNR and 0% FPR even achieved over 85% MOTA, but tracking performance is significantly degraded for increasing detector errors. It can be seen that problem becomes the more challenging by adding the more detector errors.

The proposed method shows clear advantages in 3D localization accuracy when increasing missing detections, achieving the best MOTP in all three sequences (see Figure 4.4). Since we optimize each 3D trajectory hypotheses in our unified optimization framework, it significantly improves localization performance. In addition, the improvement of localization finally increases MOTA and IDS. The proposed method outperforms against the state-of-the-art method [38] with more than 5% and 10% gain in MOTA, while achieving minimal IDS in 50% FNR.

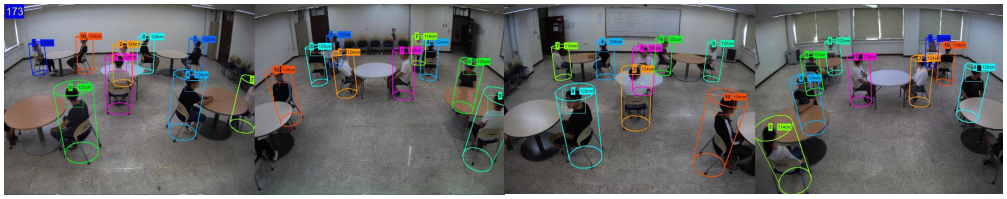
When the targets move in z-direction such as *sitting* and *sit.&stand.* sequences, it is difficult for other methods [38] and "Proposed-MMDA (w/o THU)" to robustly localize 3D position of the targets, especially when a target is detected by only one camera. Note that the proposed method finds 3D positions robustly by fusing observations across multiple cameras within multiple frames, thereby achieving a small loss of tracking performance. Our method degrades 3.2% in *sitting* and 0.9% in *sit.&stand.* sequence, whereas the method of Hofmann *et al.* [38] degrades 15.5% in *sitting* and

Table 4.4: Quantitative evaluation on the *PSN-University* dataset using a head detector as an input. we also reported the number for recall and precision of the used detector by averaging over all visible cameras.

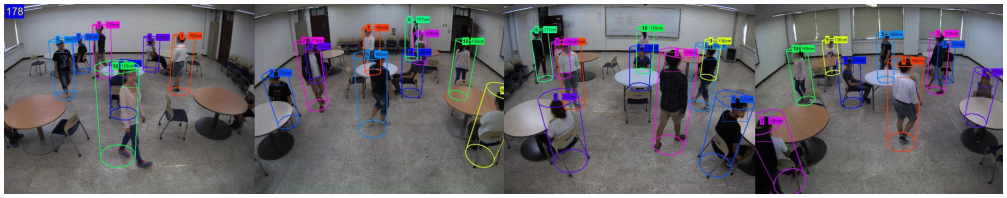
Dataset	Method	MOTA↑	MOTP↑	Rcll↑	Prcn↑	IDS↓	FM↓	GT	MT↑	PT↓
	Proposed-MMDA	**90.1%**	**85.6%**	**96.6%**	94.4%	10	**6**	10	**100%**	**0%**
PSN-Univ.	**Proposed-MMDA** (w/o THU)	89.5%	82.0%	93.4%	97.4%	17	16	10	90%	10%
standing	Hofmann *et al.* [38]	88.1%	82.3%	91.5%	97.1%	**9**	12	10	90%	10%
	Baseline	67.0%	85.0%	73.4%	**98.5%**	65	68	10	10%	90%
	Detection (Head)	-	-	79.4%	95.1%	-	-	-	-	-
	Proposed-MMDA	**88.2%**	**89.1%**	**95.1%**	93.9%	**12**	**4**	10	**100%**	**0%**
PSN-Univ.	**Proposed-MMDA** (w/o THU)	85.0%	87.0%	93.8%	92.1%	14	14	10	**100%**	**0%**
sitting	Hofmann *et al.* [38]	84.5%	88.0%	92.9%	92.4%	12	10	10	**100%**	**0%**
	Baseline	68.4%	**89.1%**	76.8%	**95.4%**	85	85	10	60%	40%
	Detection (Head)	-	-	82.8%	93.1%	-	-	-	-	-
	Proposed-MMDA	75.6%	85.4%	**93.0%**	84.9%	21	**10**	10	**100%**	**0%**
PSN-Univ.	**Proposed-MMDA** (w/o THU)	**77.2%**	84.8%	90.2%	88.4%	29	27	10	90%	10%
sit.&stand.	Hofmann *et al.* [38]	77.1%	86.4%	89.1%	88.8%	**18**	19	10	80%	20%
	Baseline	69.8%	**87.4%**	78.1%	**95.8%**	120	131	10	60%	40%
	Detection (Head)	-	-	75.1%	90.5%	-	-	-	-	-



(a) *standing* sequence



(b) *sitting* sequence



(c) *sitting&standing* sequence

Figure 4.6: Qualitative results of mixed multidimensional assignment approach using a head detector as an input. (top to bottom) PSN-University *standing*, *sitting*, and *sit.&stand.* sequences.

8.7% in *sit.&stand.* sequence.

Similarly, most of the compared methods decreases their overall tracking performance as increasing false detections, which increases IDSs and decreases MOTA (see Figure 4.5). Since the proposed method finds the more realistic trajectories taking account into their physical properties, “fake trajectories” can be constructed rather plausibly describing false detections. In fact, a MOTA decrease of the proposed method was larger than that of the method of Hofmann *et al.* [38] in *sitting* and *sit.&stand.* sequences. Nevertheless, the proposed method robustly maintains its tracking performance, achieving best MOTP and IDS.

Table 4.5: Quantitative evaluation on *PETS 2009* dataset using a full body detector as an input. we also reported the number for recall and precision of the used detector by averaging over all visible cameras. * mark denotes that the results are copied from tables of the paper.

Dataset	Method	Rcll	Prcn	MOTA	MOTP	IDS	FM	MT	PT	ML
	Proposed-MMDA	**99.5%**	**99.6%**	99.0%	77.2%	5	2	**100.0%**	**0.0%**	**0.0%**
PETS 2009	**Proposed-MMDA** (w/o THU)	99.1%	99.1%	98.1%	77.6%	3	0	**100.0%**	**0.0%**	**0.0%**
S2.L1	Hofmann *et al.* [38]*	-	-	99.4%	**83.0%**	1	2	**100.0%**	**0.0%**	**0.0%**
	Yoo *et al.* [41]*	-	-	**99.5%**	78.1%	**0**	**0**	**100.0%**	**0.0%**	**0.0%**
	Baseline (3)	97.8%	83.8%	89.8%	74.4%	63	23	**100.0%**	**0.0%**	**0.0%**
	Detection (FullBody)	85.6%	96.0%	-	-	-	-	-	-	-
	Proposed-MMDA	**89.9%**	93.1%	**81.5%**	70.8%	142	**88**	78.4%	17.6%	4.1%
PETS 2009	**Proposed-MMDA** (w/o THU)	89.8%	91.6%	79.8%	69.8%	147	94	**79.7%**	16.2%	4.1%
S2.L2	Hofmann *et al.* [38]*	-	-	79.7%	**74.2%**	**132**	129	69.8%	27.9%	**2.3%**
	Yoo *et al.* [41]*	-	-	72.9%	63.1%	246	132	73.0%	24.3%	2.7%
	Baseline (3)	85.1%	90.6%	71.0%	70.3%	437	302	70.3%	27.0%	2.7%
	Detection (FullBody)	65.0%	**94.1%**	-	-	-	-	-	-	-
	Proposed-MMDA	**72.5%**	93.8%	**66.6%**	66.7%	**36**	**24**	43.2%	34.1%	22.7%
PETS 2009	**Proposed-MMDA** (w/o THU)	**72.5%**	91.6%	64.5%	57.2%	43	28	**45.5%**	29.5%	25.0%
S2.L3	Hofmann *et al.*[38]	-	-	65.4%	**73.9%**	116	88	40.9%	34.1%	25.0%
	Yoo *et al.* [41]	-	-	54.5%	57.0%	101	78	34.1%	56.8%	**9.1%**
	Baseline (3)	69.9%	89.9%	57.5%	62.1%	150	104	31.8%	50.0%	18.2%
	Detection (FullBody)	40.4%	**99.6%**	-	-	-	-	-	-	-

In Table 4.4, we also report the results when the current state-of-the-art detector is used [50]. The classifier of Dollar *et al.* [50] was trained for a head-shoulder detector using positive samples from NLPR dataset [52] and negative samples from INRIA dataset [51]. Note that detection performance at each sequence was about recall 80% and over precision 90%, i.e., FNR 20% and FPR 10%. Especially, the proposed method improved Recall metric significantly from the input detections after linking the detections of each person and filling the missing gap of the detections. As a result, the proposed method reported the best performance in terms of MT, Recall, and FM metric, which mostly tracked 10 people at each sequence (see also Figure 4.6 for a visual illustration).

(a) *S2.L1* sequence

(b) *S2.L2* sequence

(c) *S2.L3* sequence

Figure 4.7: Qualitative results of mixed multidimensional assignment approach using a fullbody detector as an input. (top to bottom) PETS 2009 *S2.L1*, *S2.L2*, and *S2.L3* sequences.

PETS2009. We evaluated our method on three sequences in the *PETS 2009* benchmark dataset [62], where the deformable part model (DPM) [61] were used for the detection of full body (see Figure 4.7 for a visual illustration). For a fair comparison with state-of-the-art methods [38, 41], we adopted the same ground truth provided by Milan *et al.* [58] and used the same number of cameras. The 3D trajectories of the ground truth are defined on ground plane i.e., 2D Euclidean space, $Z = 0$, therefore we projected to the ground plane with reference to the "View 001". As shown in Table 4.5, the proposed method achieved comparable performance at *S2.L1* sequence and outperformed the state-of-the-art methods at *S2.L2* and *S2.L3*, recording the best MOTA. Note that by using multiple cameras, "Baseline" method also achieved MOTA 89.0% and MT 100% in low density sequence such as *S2.L1* sequence. We also note that the proposed method fixed the parameters of algorithm along three sequences in contrast to the methods of Hofmann *et al.* [38] and Yoo *et al.* [41].

Application: Real-time 3D localizing and tracking system (3DLTS)

Figure 4.8 shows that qualitative visualizations of the proposed system on the PETS 2009 *S2.L1* and the PSN-University *standing* sequences. The full body model was used in the PETS 2009 dataset and head model in the PSN-University dataset. To evaluate the quantitative performance, we used the PETS 2009 dataset, which is a public benchmark dataset. In order to calculate the performance and computation time of the whole system, the overall performance evaluation was performed by changing the meta parameters such as engaged cameras and maximum number of iterations (max_iter). Next, we compared with other state-of-the-art methods.

System overall performance. In order to evaluate overall system performance, algorithm speed and tracking performance were measured for various system configurations (engaged cameras and max_iter). First, we performed a comparative experiment on the maximum number of iterations max_iter of the proposed tracking algorithm. The proposed tracking algorithm is an iterative algorithm that iteratively performs ran-

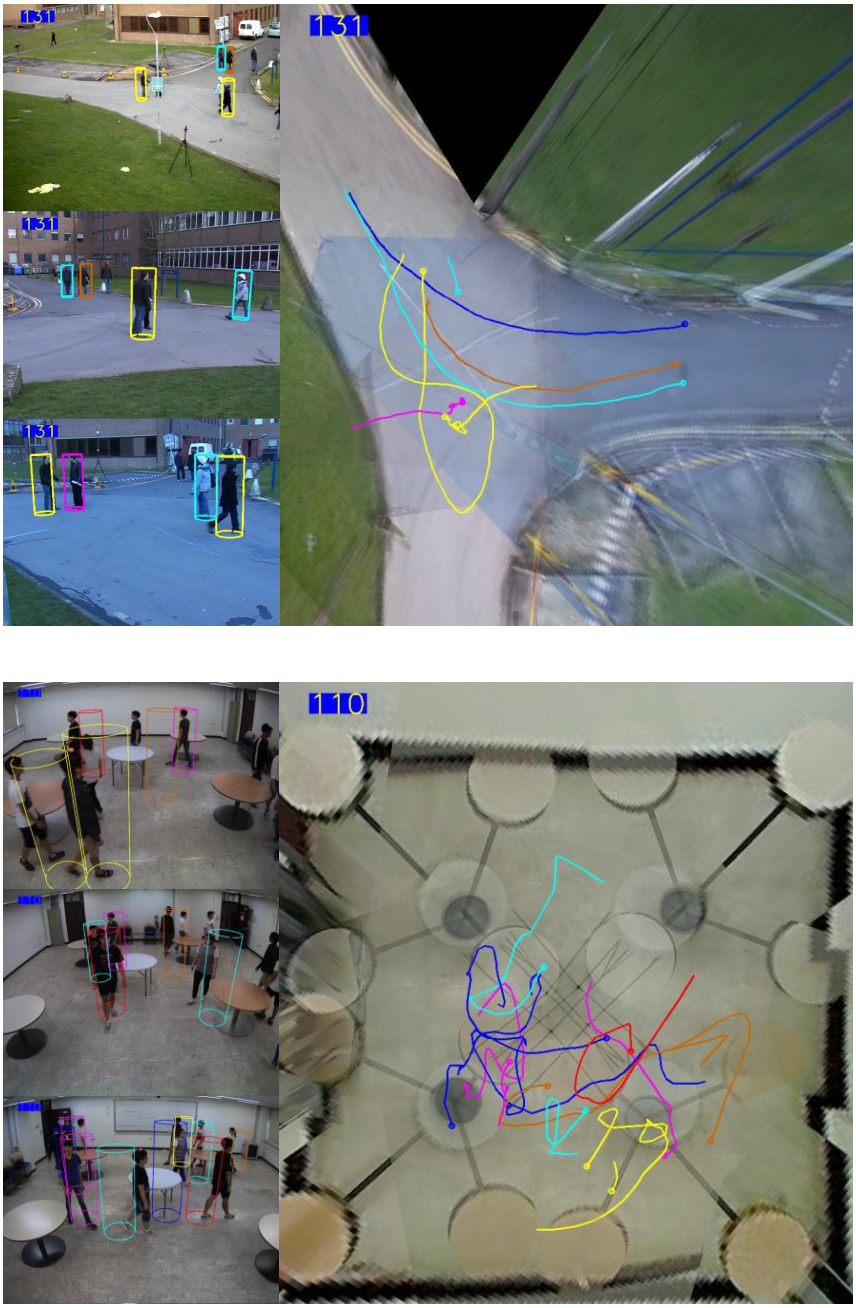


Figure 4.8: Qualitative results of real-time 3D localizing and tracking system for the PETS 2009 *S2.L1* (Left) and the PSN-University *standing* sequence (Right).

Table 4.6: Dependency of max_iter for 4 cameras on the PETS 2009 *S2.L1*.

Method	FPS	MOTP↑	MOTA↑
Proposed-3DLTS (max_iter 25)	5.24	76.9%	92.0%
Proposed-3DLTS (max_iter 50)	5.20	77.3%	95.6%
Proposed-3DLTS (max_iter 75)	5.18	77.3%	95.8%
Proposed-3DLTS (max_iter 100)	5.05	77.3%	95.9%
Proposed-3DLTS (max_iter 150)	4.98	77.3%	96.1%
Proposed-3DLTS (max_iter 200)	4.87	77.3%	96.1%

Table 4.7: Computational time (four cameras and $max\_iter = 100$).

	w/o OpenMP	w/ OpenMP
Detection	285.70 ms (96.4%)	184.5 ms (93.2%)
Feature Extraction	220.85 ms (74.5%)	155.29 ms (78.4%)
Classifying	63.71 ms (21.5%)	27.86 ms (14.1%)
Post-Processing & Etc	1.14 ms (0.4%)	1.35 ms (0.7%)
Tracking	10.82 ms (3.6%)	13.42 ms (6.8%)
Total	3.37FPS	5.05 FPS

dom split and re-merging to improve the initial solution to a better solution. Table 4.6 shows the relationship between tracking performance and system speed according to max_iter for four cameras. If max_iter is large, the tracking performance, that is, MOTA and MOTP, increases. When max_iter becomes larger than a specific value, the performance is no longer improved and the solution converges as shown in Table 4.6. In addition, the number of max_iter until the solution converges sufficiently, is related to the size of the problem space. If the number of cameras increases, the max_iter value should be higher. Therefore, we need to determine the appropriate max_iter and camera numbers for the trade-off between real-time speed and satisfactory performance.

Computational time. The proposed system achieves a real-time speed (>5 FPS) with satisfactory performance at $max\_iter = 100$ for four cameras. Table 4.7 shows the computational time required for the detection and tracking module for the case of four

Table 4.8: Quantitative evaluation of real-time 3D localizing and tracking system for the PETS 2009 *S2.L1*. Also shown in parentheses is the number of cameras.

Method	FPS	O/B	MOTP↑	MOTA↑	MT↑	ML↓	Rcll↑	Prcn↑
Proposed-3DLTS (3)	5.49	Online	73.9%	95.0%	**100.0%**	**0.0%**	97.1%	98.6%
Proposed-3DLTS (4)	5.05	Online	77.3%	95.9%	**100.0%**	**0.0%**	98.1%	98.4%
Proposed-3DLTS (5)	3.69	Online	77.7%	96.6%	**100.0%**	**0.0%**	98.2%	98.8%
Yoo *et al.*[41] (3)	<1	Online	72.9%	98.9%	**100.0%**	**0.0%**	-	-
Hofmann *et al.*[38] (3)	<0.82	Batch	74.7%	**99.0%**	**100.0%**	**0.0%**	99.6%	**99.5%**

cameras and $max\_iter = 100$. Detection module occupies 96.4% of the total time and occupies most of the time. In particular, the feature extraction using image pyramid occupies 74.5%, which is a bottleneck of the proposed system. With simple parallel processing with OpenMP, the detection time was reduced by 35.4% compared to the previous one, resulting in real-time performance. The proposed tracking module occupies only 7% of the total time and shows fast performance at 74.5 FPS in 4 cameras. In Table 4.8, we compared our method to state-of-the-art batch methods [38] as well as the online method [41]. We used three configurations for the proposed scheme: 1) max_iter=100, 3 cameras; 2) max_iter=100, 4 cameras; 3) max_iter=500, 5 cameras. Although the online approach is difficult to recover from missing detections because it only uses inputs up to the current frame, the proposed scheme achieves competitive performance to the state-of-the-art batch methods. In addition, the proposed method shows real-time performance that is five times faster than the existing online method [41]. Other state-of-the-art methods [38, 41] are based on detections using deformable part model (DPM) [61]. DPM takes over 0.6s to process one frame and is therefore not suitable for real-time applications.

4.3.2 Variational Inference Approach

Parameters

Our probabilistic model shown in (3.18-3.21) contains some predefined weighting parameters. Each parameter is empirically found but most of them were fixed regardless of the kind of sequences. Even though two datasets are different in resolution, indoor/outdoor, number of targets and many other aspects, only λ_{mot} and λ_{mid} were set to different values depending on the video characteristics. This λ_{mot} is related to the frame rate of a recoded video, where a lower frame rate leads to a lower λ_{mot}. On the other hand, λ_{mid} is correlated with the density of the people because it penalizes the missed detections that occur more often in crowded scenes. In our experiments, λ_{mot} was set to $1/160^2$ for PSN-University and $1/100^2$ for PETS 2009. In PSN-University, λ_{mid} was fixed to 10^2 in all sequences. In PETS 2009, λ_{mid} was set to 10^2 for *S2.L1*, 8^2 for *S2.L2* and 7^2 for *S2.L3*. The other weight parameters were fixed as follows: $\lambda_{obs} = 1/20^2$, $\lambda_{fse} = 20^2$, $\lambda_{eez} = 10^2$, $\lambda_{fak} = 14^2$, $\lambda_{jmp} = 3^2$.

Convergence

First, we analyze the convergence trends of MOTP and MOTA in the proposed expectation-maximization framework. In the E-step, 3D positions of given trajectory assignments are updated. Based on the new 3D positions, the new trajectory assignments are obtained in the M-step. As shown in Figure 4.9, the tracking performance mainly increases in the M-step, which is influenced by the improved accuracy of the 3D position estimation in E-step. In most cases, the algorithm converged within 5-10 iterations.

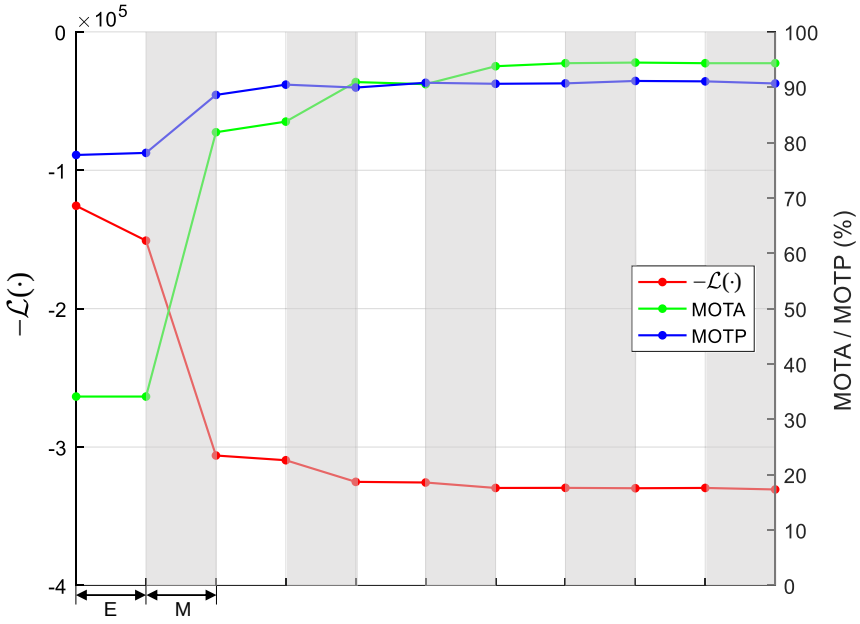


Figure 4.9: Convergence trends of MOTP and MOTA on the PSN-University *sitting* sequence at FNR 50%.

Robustness Evaluation in PSN-University

To show the tendency of tracking performance depending on detection performance, false negatives (missing detection) and false positives (false detections) were added to the hand-labeled detections. To simulate the false negatives of a detector, true detections were randomly removed to increase the false negative rate (FNR) from 0% to 50%. To create the false positives of a detector, false detections were made at random locations to adjust the false positive rate (FPR) from 0% to 20%.

By changing the FNR (see Figure 4.10) and the FPR (see Figure 4.11), we compared our method to the state-of-the art methods [38] and "Proposed-MMDA (w/o THU)". Note that the method of Hofmann *et al.* [38] needs to use 3D coordinates of each combination of detections. However, it is impossible to determine 3D position from 2D detection in a single camera if the height of the person is unknown. For this case, 3D positions are calculated under the assumption of person's height. In our experiments, we assume the person's height be 1700 millimeters. For both FNR and FPR cases, our method has clear advantages in 3D localization accuracy, outperforming [38] in terms of MOTP. By increasing the FNR (see Figure 4.10), the tracking performance degrades with decreasing MOTA and MOTP and increasing IDS+FM. In most cases, our method is more robust against false positives and negatives than Hofmann *et al.* [38] and is comparable to "Proposed-MMDA (w/o THU)" in terms of MOTA. In terms of FM and IDS, our method is mostly better than "Proposed-MMDA (w/o THU)". In terms of the performance degradation due to increase of FPR, our method is better than "Proposed-MMDA (w/o THU)".

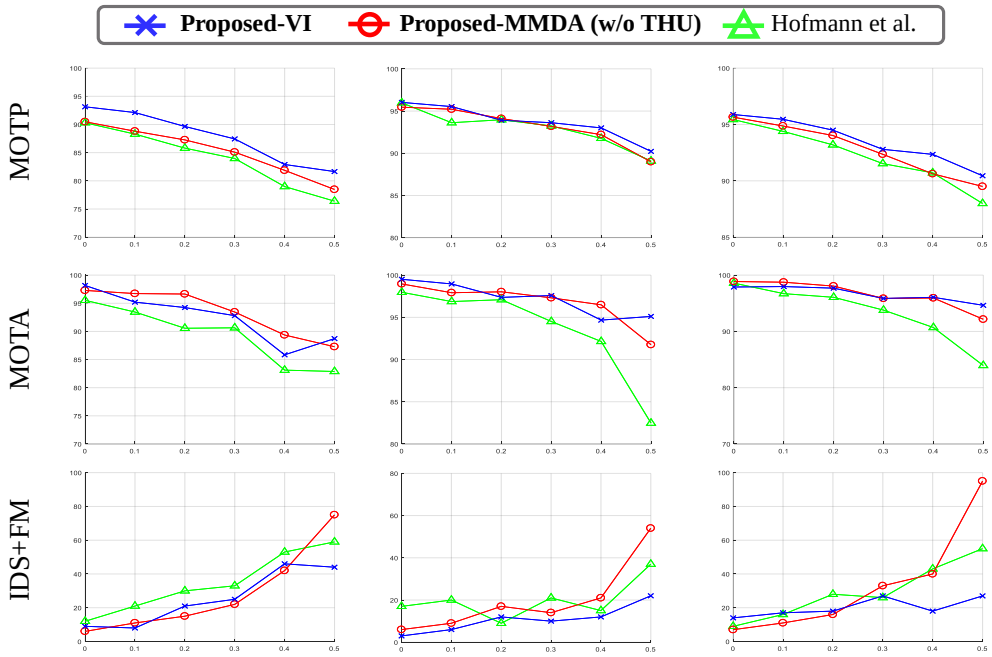
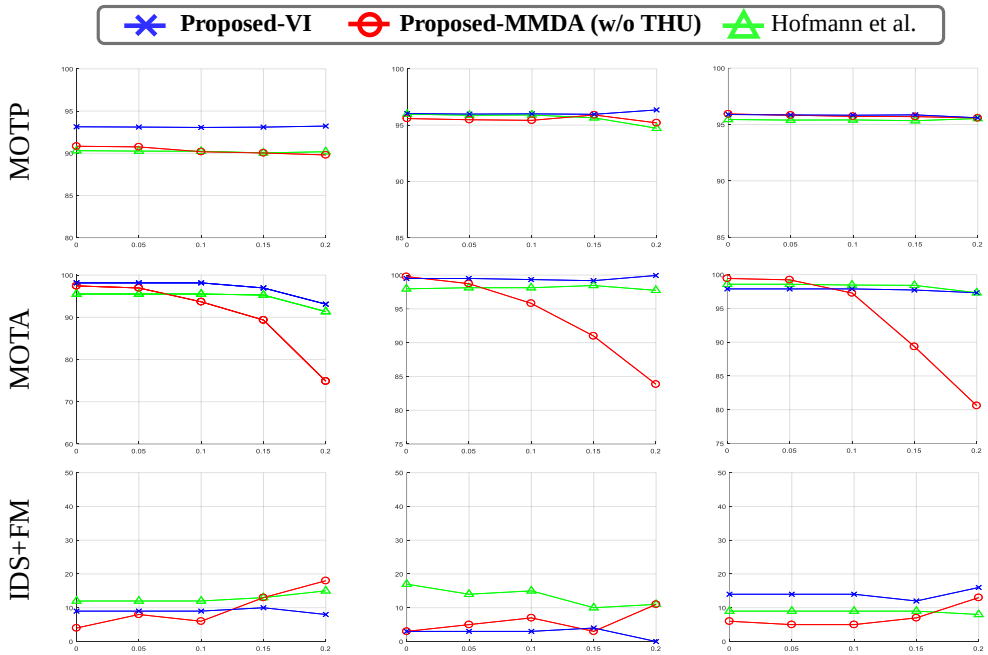


Figure 4.10: Robustness evaluation against increase of false negative rate (FNR). The evaluation was conducted for MOTP, MOTA, and IDS+FM using the *PSN-University* dataset.



(a) Standing (b) Sitting (c) Standing & sitting

Figure 4.11: Robustness evaluation against increase of false positive rate (FPR). The evaluation was conducted for MOTP, MOTA, and IDS+FM using the *PSN-University* dataset.

Benchmark evaluation in PETS 2009

The proposed method has been compared with the state-of-the-art methods [38, 41]. In tracking-by-detection framework, tracking performance is highly dependent on detection performance. For a fair comparison, all methods are evaluated with the same detections obtained using [61], and ground truth trajectories annotated by [18]. The ground truth includes the 2D trajectories of the foot positions on the ground plane. Thus, the estimated 3D trajectories were projected to $z = 0$ plane for evaluation. In Table 4.9, we report average recall and precision of detections used in our experiments. The density of people in *S2.L3* is highest and that of *S2.L2* is higher than that of *S2.L1*. As the density increases, the recall performance deteriorates, which leads to the increase of missing detections. Therefore, *S2.L2* and *S2.L3* are more challenging than *S2.L1*.

In *S2.L2* and *S2.L3* sequences, MOTA and MOTP of the proposed method are much better than the state-of-the-art methods [38, 41]. In *S2.L1* sequence, tracking performance tends to be saturated since the average recall is over 85%. Compared to other methods using predefined 3D position [38, 41], our method shows that the performance of MOTA as well as MOTP is improved in all sequences. This implies that the estimation accuracy of 3D position is improved owing to the effect of the proposed variational expectation(position estimation) and maximization (trajectory assignment) framework. For other metrics such as MT, ML, IDS, and FM metrics, the proposed method is still competitive. For reference, Figure 4.12 illustrates the qualitative performance of our method. We can see that even in a high density scene, our method shows a satisfactory performance.

Table 4.9: Quantitative evaluation on *PETS 2009* dataset. * mark denotes that the results are copied from tables of the paper. Also shown in parentheses is the number of cameras.

Dataset	Method	MOTA↑	MOTP↑	IDS↓	FM↓	MT↑	ML↓	Rcll↑	Prcn↑
	Proposed-VI	98.3%	**78.1%**	3	1	**100.0%**	**0.0%**	**99.8%**	98.6%
PETS 2009	Hofmann *et al.*[38]	99.0%	74.7%	3	1	**100.0%**	**0.0%**	99.6%	**99.5%**
S2.L1	Yoo *et al.*[41]*	**99.5%**	**78.1%**	**0**	**0**	**100.0%**	**0.0%**	-	-
	Baseline	89.8%	74.4%	63	23	**100.0%**	**0.0%**	97.8%	83.8%
	Detection [61]	-	-	-	-	-	-	85.6%	96.0%
	Proposed-VI	**86.5%**	**73.7%**	**57**	**38**	**79.7%**	5.4%	**91.3%**	95.8%
PETS 2009	Hofmann *et al.*[38]	85.2%	72.2%	68	53	74.3%	4.1%	89.3%	**96.4%**
S2.L2	Yoo *et al.*[41]*	72.9%	63.1%	246	132	73.0%	**2.7%**	-	-
	Baseline	71.0%	70.3%	437	302	70.3%	2.7%	85.1%	90.6%
	Detection [61]	-	-	-	-	-	-	65.0%	94.1%
	Proposed-VI	**64.6%**	**62.3%**	44	27	43.2%	18.2%	**74.0%**	90.2%
PETS 2009	Hofmann *et al.*[38]	62.1%	59.7%	**26**	**15**	43.2%	27.3%	68.8%	92.1%
S2.L3	Yoo *et al.*[41]*	54.5%	57.0%	101	78	34.1%	**9.1 %**	-	-
	Baseline	57.5%	62.1%	150	104	31.8%	18.2%	69.9%	89.9%
	Detection [61]	-	-	-	-	-	-	40.4%	**99.6%**

(a) *S2.L1* sequence

(b) *S2.L2* sequence

(c) *S2.L3* sequence

Figure 4.12: Qualitative results of variational inference approach for the *PETS 2009* dataset.

Benchmark evaluation in PSN-University

The benchmark experiments were conducted using actual detections for PSN-university (see Figure 4.13). The PSN-University dataset is a challenging dataset because it is a classroom environment, where the desk is covering the lower body part of the person, and the students sitting or standing are mixed together to include the uncertainty of the bounding box position. Since full body detection is difficult in the PSN-University dataset, we use head detection as in [54]. The proposed method estimates the probability distribution of each person's 3D head and their assignment variables together. The average recall and precision for all cameras in the head detector used in the experiment are shown in the Table 4.10. We compared the proposed method with the state-of-the-art method [38] and a baseline method based on separate data association. The baseline method is a two-step approach in which the inter-camera association is obtained in a greedy manner and the inter-frame association is achieved by bipartite matching between the inter-camera associations. As shown in the below table, the proposed method in all three sequences achieved a higher MOTA than other unified data association approaches as well as the baseline. Especially in sitting and sitting & standing sequence, the variation of the z-value of the 3D trajectory is large because a person is seated or standing in a classroom chair. The proposed method improved the MOTA metric by 9% for sitting sequence and 15% for sitting & standing sequence. At the same time, IDS and FM also achieved the best performance. Note that the parentheses in the tables denote the best performance in each column.

Table 4.11 shows three tracking performance of the PSN-University dataset: Proposed-VI, without the additional update (AU) in E-step and the initial solution. In *sitting* sequence, the proposed method improves the MOTA by 12% compared to the initial solution by performing an additional update, but only 3% improvement from the initial solution when not doing the AU in E-step. Therefore, it is important to update the means and covariance matrices for unselected detection indices in the trajectory assignment. If the unselected detection indices are not updated, the means and covari-

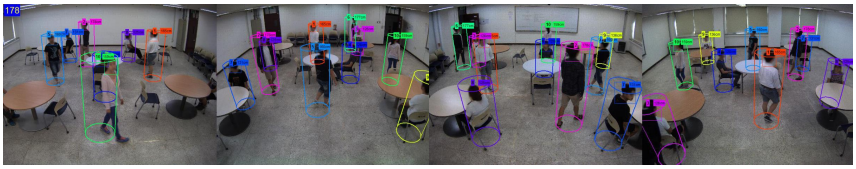
Table 4.10: Quantitative evaluation on the *PSN-University* using a head detector as an input. we also reported the number for recall and precision of the used detector by averaging over all visible cameras.

Dataset	Method	MOTA↑	MOTP↑	Rcll↑	Prcn↑	IDS↓	FM↓	GT	MT↑	PT↓
	Proposed-VI	**92.6 %**	83.0%	95.4%	**98.0%**	12	**6**	10	**100%**	**0%**
PSN-Univ.	Hofmann *et al.* [38]	88.1%	82.3%	91.5%	97.1%	**9**	12	10	90%	10%
standing	Baseline	67.0%	85.0%	73.4%	**98.5%**	65	68	10	10%	90%
	Detection (Head)	-	-	79.4%	95.1%	-	-	-	-	-
	Proposed-VI	**93.4 %**	88.2%	**95.1%**	**99.0%**	13	7	10	**100%**	**0%**
PSN-Univ.	Hofmann *et al.* [38]	84.5%	88.0%	92.9%	92.4%	12	10	10	**100%**	**0%**
sitting	Baseline	68.4%	**89.1%**	76.8%	95.4%	85	85	10	60%	40%
	Detection (Head)	-	-	82.8%	93.1%	-	-	-	-	-
	Proposed-VI	**92.3%**	**87.5%**	**93.3%**	**99.2%**	**7**	**10**	10	**100%**	**0%**
PSN-Univ.	Hofmann *et al.* [38]	77.1%	86.4%	89.1%	88.8%	18	19	10	80%	20%
sit.&stand.	Baseline	69.8%	87.4%	78.1%	95.8%	120	131	10	60%	40%
	Detection (Head)	-	-	75.1%	90.5%	-	-	-	-	-

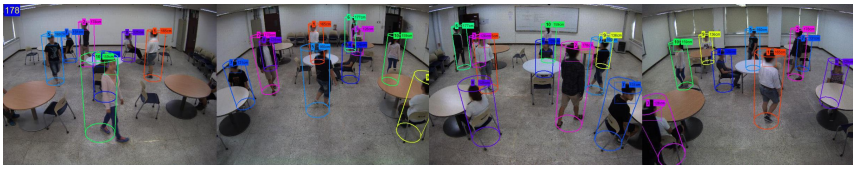
ance matrices of the selected trajectory assignments are optimized. Thus, it is easily stuck in the current trajectory assignment, which becomes harder to select better trajectory assignments in the M-step.

Table 4.11: Self-comparisons of variational inference approach for the *PSN-University* dataset whether additional update (AU) is performed.

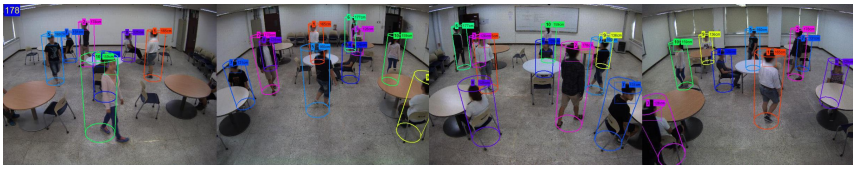
Dataset	Method	MOTA↑	MOTP↑	Rcll↑	Prcn↑	IDS↓	FM↓	GT	MT↑	PT↓
PSN-Univ.	**Proposed-VI**	**92.6 %**	83.0%	95.4%	98.0%	12	**6**	10	**100%**	**0%**
standing	**Proposed-VI** (w/o AU)	77.5%	**86.3%**	80.1%	98.9%	21	21	10	70%	30%
	Initial	73.6%	85.9%	75.7%	**99.9%**	25	23	10	60%	40%
PSN-Univ.	**Proposed-VI**	**93.4 %**	88.2%	**95.1%**	99.0%	13	7	10	**100%**	**0%**
sitting	**Proposed-VI** (w/o AU)	84.6%	89.0%	87.0%	98.5%	19	14	10	80%	20%
	Initial	81.5%	**89.9%**	83.2%	**99.5%**	24	26	10	80%	20%
PSN-Univ.	**Proposed-VI**	**92.3%**	**87.5%**	**93.3%**	**99.2%**	**7**	**10**	10	**100%**	**0%**
sit.&stand.	**Proposed-VI** (w/o AU)	86.8%	87.4%	88.5%	99.1%	22	21	10	90%	10%
	Initial	80.8%	**87.5%**	82.8%	99.5%	38	40	10	70%	30%



(a) *standing* sequence



(b) *standing* sequence



(c) *standing* sequence

Figure 4.13: Qualitative results of variational inference approach for the *PSN-University* dataset.

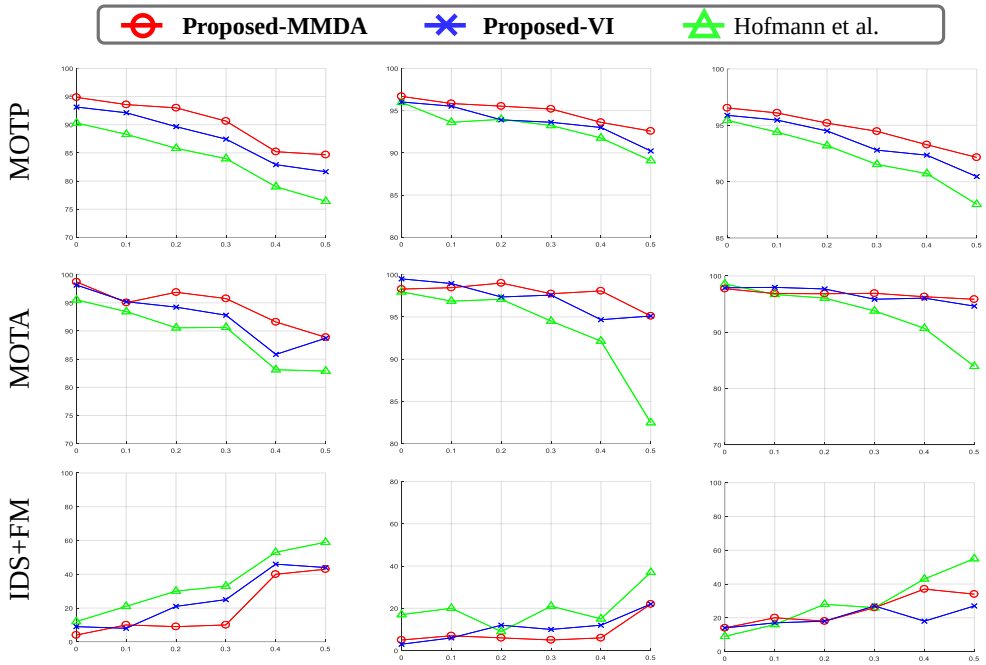
Table 4.12: Quantitative comparisons of the proposed two approaches (MMDA and VI) on the *PETS 2009* dataset.

Dataset	Method	MOTA↑	MOTP↑	Rcll↑	Prcn↑	IDS↓	FM↓	MT↑	ML↓
PETS 2009	**Proposed-VI**	98.3%	**78.1%**	**99.8%**	98.6%	3	1	**100.0%**	**0.0%**
S2.L1	**Proposed-MMDA**	**99.0%**	77.2%	99.5%	**99.6%**	5	2	**100.0%**	**0.0%**
	Hofmann *et al.* [38]	**99.0%**	74.7%	99.6%	99.5%	3	1	**100.0%**	**0.0%**
PETS 2009	**Proposed-VI**	**86.5%**	**73.7%**	**91.3%**	95.8%	**57**	**38**	**79.7%**	5.4%
S2.L2	**Proposed-MMDA**	81.5%	70.8%	89.9%	93.1%	142	88	78.4%	4.1%
	Hofmann *et al.* [38]	85.2%	72.2%	89.3%	**96.4%**	68	53	74.3%	4.1%
PETS 2009	**Proposed-VI**	64.6%	62.3%	**74.0%**	90.2%	44	27	**43.2%**	18.2%
S2.L3	**Proposed-MMDA**	**66.6%**	**66.7%**	72.5%	93.8%	36	24	**43.2%**	22.7%
	Hofmann *et al.* [38]	62.1%	59.7%	68.8%	92.1%	**26**	**15**	**43.2%**	27.3%

4.3.3 Comparisons of Two Approaches

PETS 2009

Table 4.12 shows the results of evaluating the two proposed frameworks in the three sequences of the PETS 2009 dataset. The deformable part model (DPM) [61] was used for full body detection. In PETS 2009, there was a different method for state-of-the-art performance per sequence. In case of *S2.L1* sequence with low density, all three methods showed performance close to perfect MOTA (100%). By contrast, *S2.L2* and *S2.L3* sequences have highly density. From the viewpoint of MOTA and MOTP metric, the VI approach in *S2.L2* and the MMDA approach in *S2.L3* showed the best performance respectively. These performance differences are related to the behavioral trends of each sequence. In the case of *S2.L2* sequence, people move freely irregularly, whereas in *S2.L3* sequence, very dense people move in a certain direction. Therefore, the MMDA approach using constant velocity motion dynamics showed some performance advantage in the case of *S2.L3* sequence where people move at a constant speed. In the case of *S2.L2*, on the other hand, the VI approach using the simpler motion dynamics showed better performance than the MMDA approach.



(a) Standing (b) Sitting (c) Standing & sitting

Figure 4.14: Robustness evaluation of the proposed two approaches (MMDA and VI) against increase of false negative rate (FNR). The evaluation was conducted for MOTP, MOTA, and IDS+FM using the *PSN-University* dataset.

PSN-University

In the experiments using ground truth detection, we can confirm that the two proposed approaches (MMDA and VI) clearly outperform Hofmann et al. [38] which is the existing state-of-the-art method (see Figure 4.14 and 4.15). First, in the experiments conducted by adjusting the false negative rate (FNR), MOTA and MOTP metrics of the MMDA approach showed better than those of the VI in most cases (see Figure 4.14). Especially in the case of MOTP metric, the VI approach using the second derivative of the trajectory is better than the MMDA approach using only the first derivative. In case of IDS and FM, the MMDA approach performed better than the VI approach for *standing* and *sitting* sequences, and the VI approach achieved better performance than

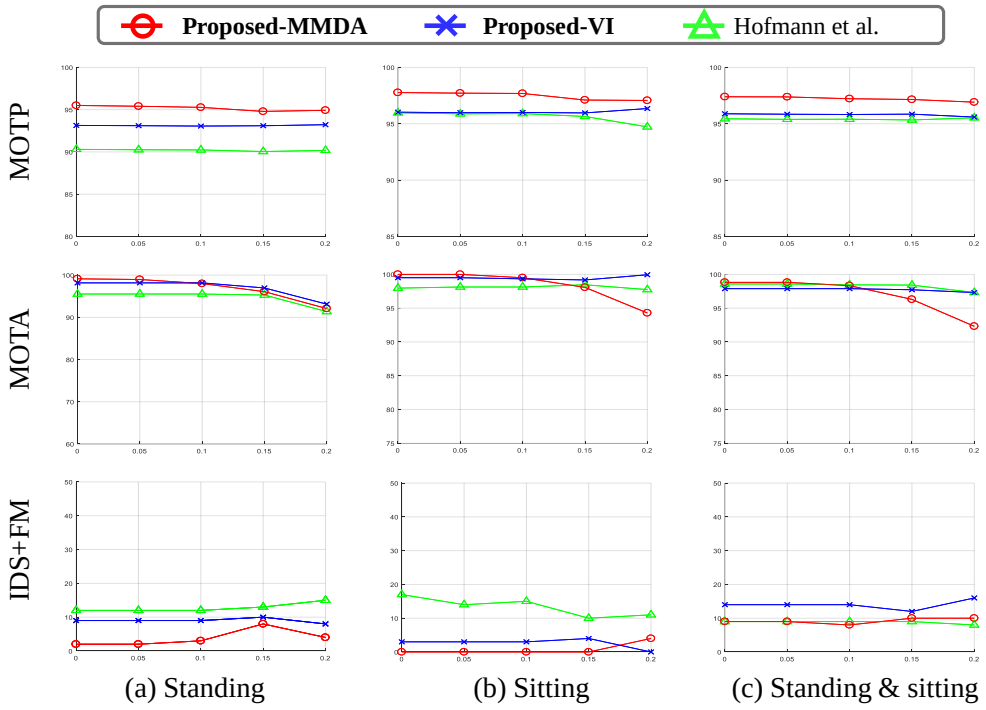


Figure 4.15: Robustness evaluation of the proposed two approaches (MMDA and VI) against increase of false positive rate (FPR). The evaluation was conducted for MOTP, MOTA, and IDS+FM using the *PSN-University* dataset.

Table 4.13: Quantitative comparisons of the proposed two approaches (MMDA and VI) on the *PSN-University* dataset using a head detector as an input.

Dataset	Method	MOTA↑	MOTP↑	Rcll↑	Prcn↑	IDS↓	FM↓	MT↑	PT↓
PSN-Univ.	**Proposed-VI**	**92.6 %**	83.0%	95.4%	**98.0%**	12	**6**	**100%**	**0%**
standing	**Proposed-MMDA**	90.1%	**85.6%**	**96.6%**	94.4%	10	**6**	**100%**	**0%**
	Hofmann *et al.* [38]	88.1%	82.3%	91.5%	97.1%	**9**	12	90%	10%
PSN-Univ.	**Proposed-VI**	**93.4 %**	88.2 %	**95.1%**	**99.0%**	13	7	**100%**	**0%**
sitting	**Proposed-MMDA**	88.2%	**89.1%**	**95.1%**	93.9%	**12**	**4**	**100%**	**0%**
	Hofmann *et al.* [38]	84.5%	88.0%	92.9%	92.4%	12	10	**100%**	**0%**
PSN-Univ.	**Proposed-VI**	**92.3%**	**87.5%**	**93.3%**	**99.2%**	**7**	**10**	**100%**	**0%**
sit.&stand.	**Proposed-MMDA**	75.6%	85.4%	93.0%	84.9%	21	**10**	**100%**	**0%**
	Hofmann *et al.* [38]	77.1%	86.4%	89.1%	88.8%	18	19	80%	20%

the MMDA approach for *standing & sitting* sequence. In the case of the experiment adjusting false positive rate (FPR), the MMDA approach method performed better in the case of the MOTP as in the FNR experiment (see Figure 4.15). This phenomenon suggests that motion model of the MMDA approach is more degradation in false detection as it attempts to describe higher-order motion. The tendency of IDS and FM was the same as the experiment with adjusting FNR. For the *standing* and *sitting* sequences, the MMDA approach performed better and the VI approach achieved better performance for the *standing & sitting* sequence.

As shown in Table 4.13, we also report the results when the state-of-the-art detector is used for detecting heads [50]. Two frameworks outperform the state-of-the-art method as the experiments using ground truth detections. When using the actual detections, the VI approach showed better overall tracking performance than the MMDA approach. This is because the uncertainty in the three-dimensional space increases due to the 2D position noise, i.e., the noise of the bounding box. The VI approach, which treats the position of objects in 3D space as a probability distribution, seems more robust to such noise than the MMDA approach that treats it as a three dimensional deterministic point.

Summary

Both of the proposed approaches have outperformed the state-of-the-art method [38] in most experiments. In the PSN-University dataset experiments using ground truth detection, the MMDA approach achieved better performance than the VI approach. In experiments using real detection, the VI approach achieved better performance than the MMDA approach in all sequences of PSN-University and *S2.L2* sequence of PETS 2009. In real detection, the 3D ambiguity increases with the noise of the bounding box. Therefore, the VI approach models the 3D position with probability distribution showed better performance.

Chapter 5

Conclusion

5.1 Concluding Remarks

In this dissertation, we have proposed two approaches to solve the spatio-temporal data association and 3D localization problem at the same time. In the mixed multidimensional assignment approach, we have shown that the two coupled problems are formulated as a minimization of the proposed *discrete-continuous* cost function that models physical properties of a trajectory. The proposed alternative optimization scheme efficiently minimizes the resulting non-convex and non-submodular cost function by alternately optimizing two different type of objective variables. In the *spatio-temporal* data association, the approximation algorithm of the multidimensional assignment (MDA) problem iteratively improves a feasible solution by two operations: random splitting and optimal merging. Experimental results show that the proposed method achieves accurate 3D trajectories of interesting targets and robust tracking performance against the state-of-the-art methods.

In variational inference approach, we have formulated the two problems as a maximum a posterior (MAP) problem on highly correlated variables, i.e., trajectory assignments and 3D positions. In addition, the intractable MAP problem has been analytically solved through a variational approximation framework. The variational expectation-

maximization (V-EM) scheme derived in the variational framework could effectively find the optimal solution owing to the explicit formula describing the probabilistic properties of the multi-camera settings.

5.2 Future Work

Although many technological advances have been made in multi-camera multi-target tracking over the last several years, tracking each object in 3D space is not completely solved. The two proposed approaches achieve the state-of-the-art performance for multi-camera multi-target tracking problems respectively. Finally, we will point out the remaining limitations of the proposed models and discuss how to improve tracking performance and apply them to other application applications.

One of the most desirable directions for improving tracking performance is to extract more features from the image. The proposed methods are based solely on geometric information using calibration information from multiple cameras. For example, we use geometric clues that the bounding box corresponding to the same person is near the three-dimensional location or that the motion of the adjacent frame in the three-dimensional space moves smoothly (see details in Section 2.1.2). Obviously, extracting more clues from the image with this geographic information can help track each trajectory accurately.

Appearance feature is one of the additional features that can be used with geometric information in the proposed methods. In the multi-target tracking problem, the appearance is used mainly on the assumption that appearance does not change much over time. Most of single-camera based approaches have modeled these assumptions as a cost function or a probability distribution, and have used them for tracking multiple objects. In addition, some methods adopt online appearance learning scheme [63, 64]. In multiple cameras, appearance in time as well as appearance in different cameras should be considered. However, it is difficult to obtain the inter-camera appearance

similarity of the same person with the assumption that the appearance does not change much. This is because the appearance with different viewpoints changes drastically. Therefore, 3D appearance modeling [65] or view-invariant appearance feature extraction [66, 67] can be one solution. It is important to note that the performance of the person re-identification problem has been improved notably using deep convolutional neural networks (CNN).

Next, another piece of information that can be extracted from an image is motion information. Since detections used as an input of currently proposed methods is processed based on an independent frame, motion information between frames is lost. Therefore, we can use optical flow as an additional observation to the proposed methods. On the other hand, tracklet-based approaches [68, 69, 70] use motion to connect sufficiently reliable detections to short trajectories and use these short trajectories as the unit of association. Extending the proposed methods to these tracklet-based methods can not only focus on solving longer-term data associations, but it can also help reduce computational complexity.

Similarly, another approach that can be complemented is the combination with silhouette-based approaches [33, 37, 71]. The silhouette-based approaches find the silhouette of each camera and to fuse it in various ways, such as creating a 3D volume or creating a synergy map. The silhouette-based approaches have the challenge of solving the "ghost effect" where no real object exists but false positives. This is basically because the silhouette has a 2D-3D ambiguity. If the silhouette of different objects simultaneously affect the 3D reconstruction, the 3D reconstruction is also made in the region where the actual object does not exist. If the data association based method and the silhouette based method are combined, it is expected that the trajectories of the accurate three dimensional volume can be found while reducing the "ghost effect".

One of the additional applications of the proposed method could be a robot or automobile for autonomous driving. Since the overlapping area may be smaller than the current setting, the view-invariant appearance feature and the geometry feature should

be combined well. Applications in the other direction utilize the proposed optimization framework for new problems. A typical example is to track multiple object poses [72, 73]. In the problem of tracking multiple object poses, we also combine the problem of associating each person's parts on the time axis and the localization problem of estimating the position of each part. Therefore, it is expected that the proposed framework can be extended and applied to various type of applications.

Bibliography

[1] Michael Friedewald, Olivier Da Costa, Yves Punie, Petteri Alahuhta, and Sirkka Heinonen, "Perspectives of ambient intelligence in the home environment," *Telematics and informatics*, vol. 22, no. 3, pp. 221–238, 2005.

[2] Diane J Cook, Michael Youngblood, and Sajal K Das, "A multi-agent approach to controlling a smart environment," in *Designing smart homes*, pp. 165–182. Springer, 2006.

[3] Giuseppe Riva, Fabrizio Davide, and Wijnand A IJsselsteijn, *Being there: Concepts, effects and measurements of user presence in synthetic environments*, Ios Press, 2003.

[4] Ichiro Satoh, "Software agents for ambient intelligence," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[5] Szymon Bobek, Grzegorz J Nalepa, Antoni Ligza, Weronika T Adrian, and Krzysztof Kaczor, "Mobile context-based framework for threat monitoring in urban environment with social threat monitor," *Multimedia Tools and Applications*, vol. 75, no. 17, pp. 10595–10616, 2016.

[6] Giovanni Acampora, Diane J Cook, Parisa Rashidi, and Athanasios V Vasilakos, "A survey on ambient intelligence in healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, 2013.

[7] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[8] Thomas E Fortmann, Yaakov Bar-Shalom, and Molly Scheffe, "Multi-target tracking using joint probabilistic data association," in *Proceedings of IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1980.

[9] Charles Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, pp. 302–312, 1977.

[10] Donald Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[11] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[12] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua, "Multiple Object Tracking Using K-Shortest Paths Optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[13] Li Zhang, Yuan Li, and Ramakant Nevatia, "Global data association for multi-object tracking using network flows," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[14] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1820–1833, 2011.

[15] Zheng Wu, Ashwin Thangali, Stan Sclaroff, and Margrit Betke, "Coupling detection and data association for multiple object tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] Robert T Collins, "Multitarget data association with higher-order motion models," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[17] Horst Possegger, Thomas Mauthner, Peter M Roth, and Horst Bischof, "Occlusion Geodesics for Online Multi-object Tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[18] Anton Andriyenko, Konrad Schindler, and Stefan Roth, "Discrete-continuous optimization for multi-target tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19] Ernesto Brau, Jinyan Guan, Kyle Simek, Luca Del Pero, Colin Reimer Dawson, and Kobus Barnard, "Bayesian 3D Tracking from Monocular Video," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2013.

[20] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah, "Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] Visesh Chari, Simon Lacoste-Julien, Ivan Laptev, and Josef Sivic, "On pairwise costs for network flow multi-object tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] Afshin Dehghan, Yicong Tian, Philip HS Torr, and Mubarak Shah, "Target identity-aware network flow for online multiple target tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] Philip Lenz, Andreas Geiger, and Raquel Urtasun, "Followme: Efficient online min-cost flow tracking with bounded memory and computation," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[24] Yu Xiang, Alexandre Alahi, and Silvio Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[25] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon, "Online multi-object tracking via structural constraint event aggregation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] Songhwai Oh, Stuart Russell, and Shankar Sastry, "Markov Chain Monte Carlo Data Association for Multi-Target Tracking," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 481–497, 2009.

[27] Kyungnam Kim and Larry S Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2006.

[28] Weiming Hu, Min Hu, Xue Zhou, Tieniu Tan, Jianguang Lou, and Steve Maybank, "Principal axis-based correspondence between multiple cameras for people tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 663–671, 2006.

[29] Wei Du and Justus Piater, "Multi-camera people tracking by collaborative particle filters and principal axis-based integration," in *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2007.

[30] Simone Calderara, Rita Cucchiara, and Andrea Prati, "Bayesian-competitive consistent labeling for people surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, 2008.

[31] Simone Calderara, Andrea Prati, and Rita Cucchiara, "Hecol: Homography and epipolar-based consistent labeling for outdoor park surveillance," *Computer Vision and Image Understanding*, vol. 111, no. 1, pp. 21–42, 2008.

[32] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua, "Multi-camera People Tracking with a Probabilistic Occupancy Map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 267–282, 2008.

[33] Saad M Khan and Mubarak Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 505–519, 2009.

[34] Zheng Wu, Nickolay I Hristov, Tyson L Hedrick, Thomas H Kunz, and Margrit Betke, "Tracking a large number of objects from multiple views," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2009.

[35] Michael Bredereck, Xiaoyan Jiang, Marco Körner, and Joachim Denzler, "Data association for multi-object Tracking-by-Detection in multi-camera networks," in *Proceedings of International Conference on Distributed Smart Cameras (ICDSC)*, 2012.

[36] Laura Leal-Taixe, Gerard Pons-Moll, and Bodo Rosenhahn, "Branch-and-price global optimization for multi-view multi-target tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] Horst Possegger, Sabine Sternig, Thomas Mauthner, Peter M Roth, and Horst Bischof, "Robust Real-Time Tracking of Multiple Objects by Volumetric Mass Densities," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[38] Martin Hofmann, Daniel Wolf, and Gerhard Rigoll, "Hypergraphs for Joint Multi-view Reconstruction and Multi-object Tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[39] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua, "Multi-Commodity Network Flow for Tracking Multiple People," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1614–1627, 2014.

[40] Martijn C Liem and Dariu M Gavrila, "Joint multi-person detection and tracking from overlapping cameras," *Computer Vision and Image Understanding*, vol. 128, no. C, pp. 36–50, Nov. 2014.

[41] Haanju Yoo, Kikyung Kim, Moonsub Byeon, Younghan Jeon, and Jin Young Choi, "Online scheme for multiple camera multiple target tracking based on multiple hypothesis tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 454–469, 2017.

[42] Aubrey B Poore, "Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking," *Computational Optimization and Applications*, vol. 3, no. 1, pp. 27–57, Mar. 1994.

[43] Donald Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[44] Somnath Deb, Murali Yeddanapudi, Krishna Pattipati, and Yaakov Bar-Shalom, "A generalized S-D assignment algorithm for multisensor-multitarget state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 523–538, Apr. 1997.

[45] James Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[46] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello, *Assignment Problems, Revised Reprint*, Siam, 2009.

[47] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[48] Ross Girshick, "Fast r-cnn," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2015.

[50] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona, "Fast Feature Pyramids for Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[51] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[52] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2008.

[53] Roger Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal on Robotics and Automation*, 1987.

[54] Moonsub Byeon, Songhwai Oh, Kikyung Kim, Haan-Ju Yoo, and Jin Young Choi, "Efficient spatio-temporal data association using multidimensional assignment for multi-camera multi-target tracking," in *Proceedings of British Machine Vision Conference (BMVC)*, 2015.

[55] Christopher M Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[56] Mustafa Ayazoglu, Binlong Li, Caglayan Dicle, Mario Sznaier, and Octavia I Camps, "Dynamic subspace-based coordinated multicamera tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[57] Gurobi, "Gurobi optimizer," *URL: http://www.gurobi.com*, 2012.

[58] Anton Milan, Stefan Roth, and Konrad Schindler, "Continuous Energy Minimization for Multitarget Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, 2014.

[59] Keni Bernardin and Rainer Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[60] Yuan Li, Chang Huang, and Ram Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[61] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[62] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter)*, 2009.

[63] Cheng-Hao Kuo, Chang Huang, and Ramakant Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[64] Bo Yang and Ram Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[65] Yu Xiang, Changkyu Song, Roozbeh Mottaghi, and Silvio Savarese, "Monocular multiview object tracking with 3d aspect parts," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[66] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[67] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," *arXiv preprint arXiv:1704.01719*, 2017.

[68] Chang Huang, Bo Wu, and Ramakant Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2008.

[69] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang, "Tracklet association with online target-specific metric learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[70] Seung-Hwan Bae and Kuk-Jin Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[71] Taiki Sekii, "Robust, real-time 3d tracking of multiple objects with similar appearances," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[72] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic, "3d pictorial structures revisited: Multiple human pose estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 1929–1942, 2016.

[73] Umar Iqbal, Anton Milan, and Juergen Gall, "PoseTrack: Joint multi-person pose estimation and tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

초 록

본 논문에서는 겹쳐진 영역을 바라보는 다중 카메라의 영상에서 다중 물체 추적 및 3차원 위치 추정을 위한 통합 프레임웍을 제안한다. 이 때 주요한 문제는 3차원 위치 추정 문제와 궤적 할당 문제를 동시에 해결 하는 것이다. 하지만 대부분의 기존의 방법들은 추적과 위치 추정 문제를 각각 독립적으로 분리해서 풀고자 하였다. 왜냐하면 단일 카메라 다중 물체 추적과는 달리 다중 카메라에서는 카메라 간의 관계도 고려해야 하기 때문에 두 가지 문제를 모두 해결하는 것이 훨씬 더 복잡하기 때문이다. 제안하는 방법은 다중 카메라에서의 데이터 연관 문제와 3차원 위치 추정 문제를 동시에 해결 하는 두 가지 접근 방식을 제안 한다. 첫 번째는 혼합 다중 할당 접근 방법이고 두 번째는 베이지안 변분 추론 접근 방법이다. 먼저, 혼합 다중 할당 접근 방법에서는 시공간 데이터 연계 문제와 3D 궤적 추정 문제의 두 가지 결합 문제를 공동으로 해결하고자 한다. 이때 큰 솔루션 공간을 다루기 위해 제안하는 프레임웍은 두 결합 문제를 번갈아 가며 합리적인 계산 부하로 최적화하는 효율적인 프레임웍이다. 두 번째로 베이지안 변분 추론 접근 방법에서는 다중 카메라로부터 탐지된 관측 값들에 대한 궤적 할당과 3차원 위치의 사후 확률을 최대화한다. 이때 3차원 위치를 확률 분포로 나타내어 3차원 공간의 불확정성이 존재하는 경우에 더 강인한 추적 성능을 달성 하였다. 사후 확률 최대화를 위해 볼츠만 분포를 따르는 다중 카메라 추적을 위한 7가지 요소들을 디자인하고 이 사후 확률로부터 해를 구하기 위한 기대값 최대화 기법을 개발하였다.

주요어: 다중 물체 추적 및 3차원 위치 추정, 다중 객체 추적, 다중 카메라, 다차원 할당, 변분 추론, 3차원 궤적 추정
학번: 2012-30211