Ph.D Dissertation

Online Multiple Objects Tracking with a MAP Optimization

최대 사후 확률 최적화를 통한 다중 물체 추적 기법

By

Soo Wan Kim

August 2014

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

Abstract

In an ideal surveillance scenario, the instant response to the crime/incident should be guaranteed for its purpose. For this reason, online approach is more preferred for the algorithms implemented in a surveillance system, such as moving objects detection and object tracking. Generally, online algorithms cannot break causality condition and only use past observations, which lead to lower performance than batch algorithms with future observations. However, online algorithms are more demanded than batch algorithms in a surveillance system because batch algorithms require heavy computation time. Moreover, batch algorithms need the whole video input, which makes the batch algorithms more suitable for video analysis, not for the surveillance system. While online tracking for the single object is quite normal and most current researches track its target object in online manner, most multiple objects tracking methods have been researched with offline scheme due to their heavy computation and lack of causality. Another reason why the offline scheme is widely adopted in the field of the multiple objects tracking is that the required quantity of clues to track each object and distinguish them simultaneously is much larger than the single object tracking problem. To handle this difficulty, the data association method is generally used to find temporal association of each object over frames. However, this complexity still increases when several number of cameras are used and both spatial and temporal association should be achieved.

In this thesis, we propose an online data association approach for tracking multiple number of people with both single camera and multiple cameras. Without delayed decision or future data input, we perform online data association between the detection results and tracking models and show robust performance

with a faster speed than offline data association. For multiple target tracking in the single camera case, we formulate an online MAP (Maximum A posteriori Probability) problem to find the temporal association among detection observations at the current frame and the tracking models from the last frame in the same image domain. Because a single camera can provide a limited information, the multiple target tracking with a single camera is especially weak for occlusions and overlaps. To overcome these limitations, we use the head detector which is robust against occlusions and overlaps. With head detection results and the tracking models, we encode the problem of multiple target tracking to the problem of finding matching in a graph and solve the matching problem on the formulated MAP problem considering object size, center distance, motion and appearance. During temporal association process to track multiple objects, our solution initializes new tracking model automatically. Moreover, the corruption of tracking models by missed detections from occlusions is prevented by selective update of the tracking model through occlusion reasoning method. This occlusion reasoning method prevents the tracking model from being corrupted with unreliable information. Since the proposed MAP formulation only uses the last tracking models and current observations, this proposed MAP formulation can be solved without heavy computation. In order to demonstrate the validity of the proposed method, we compare our method with the state-of-the-art methods and show improvement in performance.

Extending the proposed framework for the single camera case, we also propose an online framework to track multiple objects with multiple number of cameras. Multiple cameras can provide more information than a single camera for tracking especially when occlusions among objects happen or overlaps behind backgrounds occur. However, in the perspective of association, increased amount of information is not always preferred. The problem of multiple target tracking in

multiple cameras is much more complicated than single camera data association because spatial and temporal association should be handled at the same time. Moreover, most conventional approaches have large computational complexity by taking the global optimization scheme for the accuracy. To solve this problem of heavy computational load, we formulate an online MAP problem to find the spatial association among cameras and the temporal associations between recently consecutive frames at the same time. As the case of the single camera, we encode this online data association to a matching problem in the graph and formulate a MAP problem. In the association of the tracking models and the detection results, we use 2D position, appearance, motion, and 3D position (a reconstructed point on the world coordinate with camera matrix.) With these features of objects and tracking models, the geometric information (camera parameters) and assumptions for human models are considered in data association process. Through experiments with several datasets, we show the performance of the proposed online algorithm is comparable to the state-of-the-art offline method even in low computation load.

Keywords: visual tracking, online multiple target tracking, data association, matching graph, MAP optimization, multiple cameras

Student ID Number: 2008-22908

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of problem

Object tracking is one of the fundamental issues in building an intelligent visual surveillance system. Object tracking is a process of chasing objects and maintaining their labels through frames and it is usually initiated after detection of objects which users are interested in. With the object detection, object tracking is a very important task for surveillance purpose. This area has been studied extensively in decades and many single object tracking algorithms show good performance even in the hard conditions (severe occlusions, background clutters, and uncertain initialization). They used different dynamic models [1, 2, 3], features [4, 5, 6, 7], and learning skills [8, 9, 10, 11] to solve the upper problems. In general surveillance scenario, however, there usually exist more than one object and the single object tracking is not enough to achieve the goal of the surveillance system. While the single object tracking algorithms show good performance, extending those single object tracking methods for multiple target tracking by assigning an individual tracking model to each object does not work successfully because the multiple

target tracking problem is a much more complex problem than the single object tracking problem. The reason why this simple extension of the single tracking algorithm does not work well is that it cannot consider relationships among different objects and regards different objects as completely independent objects, which is not true. In reality, objects move along other objects or avoid each other or occlusions among them happen by crossover. This is the reason why the data association scheme is widely adopted as a solution for the multiple target tracking problem and recently shows good performance.

Tracking multiple pedestrians maintains identities of multiple people, provides their trajectories, and achieves information to recognize their personal behaviors in time sequences simultaneously. However, there exist several difficulties for tracking multiple people in crowded video. The main sources of the difficulties are occlusions, including inter-objects and object-background overlaps, and closely-located people with similar appearance. To solve these difficulties, there have been two types of approaches for multiple target tracking. The first type of approach is to use multiple number of cameras and the second type of approach is to develop a robust data association method in single camera with various cues. The former type of approach can provide more information of objects with several videos from different viewpoints and does not need to consider occlusion in one viewpoint because it can perform data association in world coordinate by projection of one view with camera calibration information or homography estimation. In the world coordinate domain, the occlusion between two different objects actually can not happen because two people locating at the same 3D position is physically impossible. However, even though using multiple number of cameras can handle the occlusion problem by providing 3D information of objects, spatial (between cameras) and temporal (between consecutive frames) association of objects should be found simultaneously and this is one of the well known *NP-hard*

problems, *K-partite matching*. On the other hand, the approaches for the data association in single camera are not suffered from large solution space unlike the multiple camera case requiring inter-cameras association as well as inter-frames association at the same time. However, it is hard to get reliable cues to associate data robustly when occlusion happens or there exist many objects with similar appearance. For this reason, the available amount of information and the computation time are very different depending on the number of cameras, and several algorithms have been proposed for both single and multiple camera cases. Currently, the number of researches with multiple number of cameras is increasing largely with the improvement of computation power and performance of detection algorithms.

1.2 Related works

• Approaches with a single camera

When multiple objects are tracked with a single camera, the amount of information is much less than the multiple camera case and falsely detected objects, occlusions, and missing detections are very crucial for reliable data association. For this reason, the performance of multiple target tracking algorithms in the single camera case is usually less than the multiple camera case. Without information from different camera views, multiple objects tracking algorithms in the single camera case generally cannot use geometric information, such as reconstruction error and positions in the world coordinate system. However, more cameras do not always guarantee better tracking result because the solution space for the data association is drastically increased with amount of observations to be associated. The conventional methods for multiple objects tracking based on the tracking-by-detection scheme can be categorized to the global optimization

method or the online optimization method with respect to the optimization domain, and the data association methods using a full-body detector or a head detector with respect to the type of the detector. In the sense of optimization domain, the global optimization methods [12, 13, 14, 15, 16, 17, 18, 19, 20, 21] breaks the causality condition using future information or deal with several frames for data association. They usually show better performance with a global optimization scheme, however, they suffer from a decision delay and high computational complexity. The online optimization methods [22, 23, 24, 25] consider only the last and current observations in association. These methods are suitable for online applications, however, the performance of data association might be degraded easily by noisy or missing detections.

For the type of detectors, most of tracking-by-detection approaches for tracking multiple objects in a single camera use full body detectors to detect interesting targets [26, 27, 25, 13, 22]. Full body detection can provide many different cues for tracking and discriminating objects. However, the object detection performance of full body detector is easily degraded by occlusions, which leads to the failure of data association. To overcome this problem, various types of detections were adopted in many approaches. Bo and Ram proposed edgelet based part detectors in [23] and Siyu et al. trained double person detector to detect occluded people and separated them to track each individual in [28]. In [18, 24], head detection results were used as observations for the data association problem. In crowded scenes, detecting heads is more suitable for video sequences with occlusions because the head part is less occluded and can provide more reliable detection results than the body of human in data association. However, the main reason why head detector is not commonly used in data association is that the head has insufficient features, such as shapes and colors, which makes the head detection results less discriminative than full body detection results.

• **Approaches with multiple cameras**

To solve the multiple object tracking problem with multiple number of cameras, various tracking approaches have been proposed. Since multiple cameras can watch the scene from different views, occluded objects in one camera view can be located far and not under occlusion situation in different view. These approaches can be categorized with respect to its input for data association and its optimization domain, and this is illustrated in Figure 1.1. In the perspective of input data, there are two types of approaches to track multiple objects with multiple cameras, which are object detector based approaches[29, 30, 31, 32, 33] and background subtraction based approaches[34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 43]. The object detector based approaches detect interesting objects such as human body or human head in the videos with pre-trained classifier first, and, then, perform the data association in spatial and temporal senses with detection results from multiple number of camera views. Detectors based on Histogram of gradients (HOG) [44] and deformable part model(DPM) [45] are widely used detectors. Several object detection algorithms, such as those mentioned above, show a robust performance even in moving cameras or under small occlusions. In [29], Leal-Taixe et al. modeled the input of detection result in a joint optimization framework over the complete sequence. They divided the data association process into two steps (spatial association and temporal association), and they solve those two associations separately. In [30], Hofmann et al. proposed hypergraphs for multiple object tracking and solve spatial and temporal association together using the network flow concept. In [31], Sternig et al. adopted the idea of generalized Hough voting in [46] and extended it for tracking with multiple cameras. By exploiting the geometric constraints, they performed Hough voting on each camera and projected the results to the top view map for the particle filter based

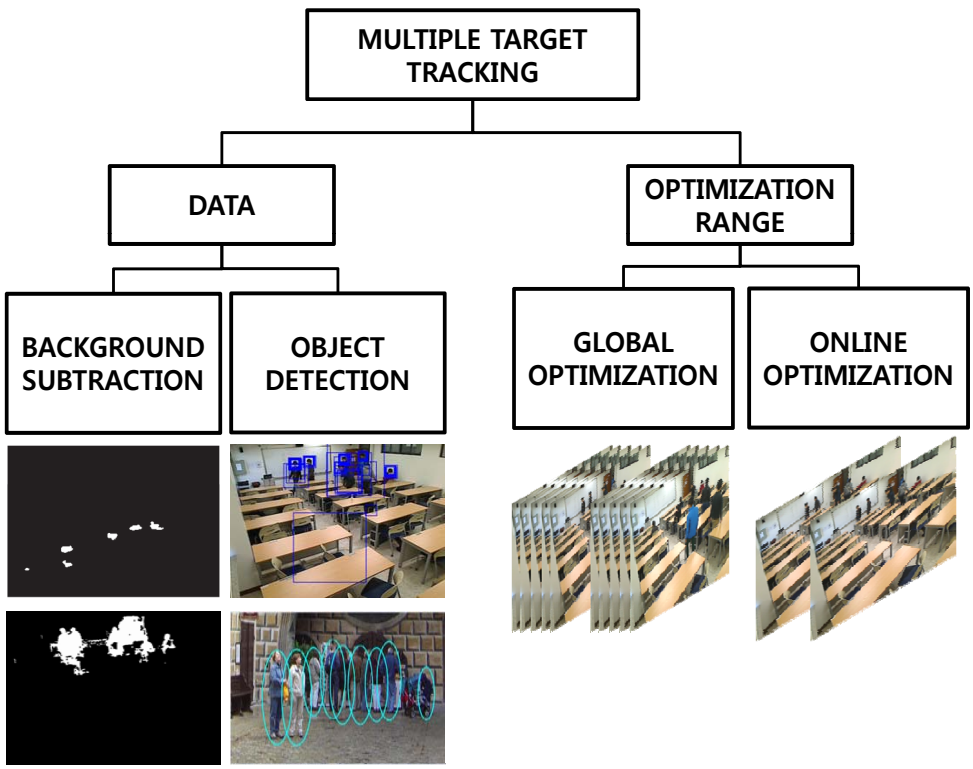

Figure 1.1: Conventional approaches for multiple target tracking. The conventional approaches are categorized with respect to its input for data association and its optimization domain. In the perspective of input data, conventional methods are divided into background subtraction based methods and object detector based methods. In the perspective of optimization domain, they are divided into global optimization methods and online optimization methods.

tracking. Mittal and Davis, in their paper [32], tracked the centroid of estimated object blobs, which was done by their own detector based on color and geometric information with Kalman filter, and Shitrit et al. built a multi-commodity network to compute flows on the directed acyclic graph in [33]. This type of approach is known as the Tracking-by-Detection method because they perform the object detection algorithm first and then give labels on the results of detection algorithm. The Tracking-by-Detection method has been widely accepted as a solution for multiple target tracking since the performance of detection algorithm becomes quite confident to be used for the data association. However, there are still false-positives or missing detection which makes the data association problem hard to solve. Because the detection of interesting object is processed by comparing the pattern of certain region with that of the pre-trained model (human body or head), false positive regions can be repeatedly detected and they are burdens to accomplish robust data association. Also, even though using multiple cameras can bring better results than using a single camera, there is no guarantee that every person is detected by exploiting multiple number of cameras. Once missing detection of certain object happens, its position could be estimated with different techniques to improve the performance of the tracking algorithm.

On the other hand, background subtraction based approaches find foreground objects in each camera view first, and, then, merge them into the single world coordinate system with camera calibration information or into the image coordinate system of one camera view with homography estimation. In [34], Possegger et al. introduced the concept of an occupancy volume and tracked each object using the local mass density score with a particle filter based tracking algorithm. They also adopted the idea of a Voronoi partitioning and divided the re-projected occupancy map of the constructed visual hull. Khan et al. in [35] used the motion detection algorithm to find the foregrounds in the ground plane and adopted

the homography constraint so that any 3D point inside the foreground object could be projected to a foreground pixel in every view. In [36], they applied multiple number of homographies on different heights for 3D shape recovery of non-occluded objects. Extending these previous works, Khan et al. in [37] built a 4D spatio-temporal sequence of synergy map, which was constructed by projecting foreground maps of each camera into 3D coordinate system and fusing them together, and performed a graph cut on the sequence of the synergy map. Kim and Davis [38] applied search-guided particle filtering and multiple hypotheses tracking scheme in [47]. In [39], Berclaz et al. reformulated the data association problem as a constrained flow optimization problem and solved it with a standard linear programming technique. The result of this paper is improved compared to that of their own previous paper which uses the sequential dynamic programming [40]. Eshel and Moses [41] estimated heads at different height by geometric information and tracked them to handle occlusion. This idea comes from the fact that head is less occluded than body and foot in crowded scene. The tracking is done only with motion information between consecutive frames. In [42], Wu et al. encoded multiple target tracking problem to multi-dimensional assignment problem and used a greedy randomized adaptive search procedure to solve the problem. They relaxed the one-to-one constraint for unmatched/newly-appeared objects and solved iteratively the relaxed problem. To improve the performance of both detection and tracking, Wu et al. [43] coupled the object detection and tracking problem together, which was solved with a single objective function. This single objective function considers object presence and network flow based data association at once. In [48], Liem and Gavrila used space volume carving technique to remove ghost artifacts in segmented foreground maps and they formulated the multiple target tracking problem as an edge selection task on a bipartite graph. Because background subtraction is much faster than object

detection algorithms, acquiring data for data association by background subtraction is easy and suitable for real-time and online surveillance scenario. However, the background subtraction itself cannot show good performance with illumination change and dynamic backgrounds. Moreover, merging foreground maps from each different camera view can bring a *phantom* effect which is the appearance of imaginary objects in merged world coordinate system by projection of adjacent objects into the same coordinate system. As shown in Figure 1.2, the foreground maps (black map) of each camera views are transformed to the single coordinate system (blue map), and both actual human-stand regions (orange circles) and phantoms (red circles) are detected by thresholding the fused score of the single coordinate system. For tracking multiple targets, not phantoms which are not actually interesting objects, those two regions should be distinguished first and data association is performed only with actual human-stand regions (orange circles). Since distinguishing phantoms from actual objects is generally a difficult task only with the fused score map of the single coordinate system, object detector based approaches are more preferred recently.

In the perspective of optimization domain, conventional approaches for multiple objects tracking with multiple cameras can be categorized into the global optimization method or the online optimization method. Most conventional methods which focus on high performance use the past and the future information altogether for data association [29, 30, 33, 37, 39, 40, 43] and perform a global optimization. This type of approach is known to find a better solution with abundant information from multiple number of frames, however, they break the causality condition and not suitable for online applications. Moreover, the computational complexity increases significantly with increasing number of cameras and objects in scenes, which causes the intractable solution or time-delayed results.

On the other hand, several online algorithms for multiple objects tracking

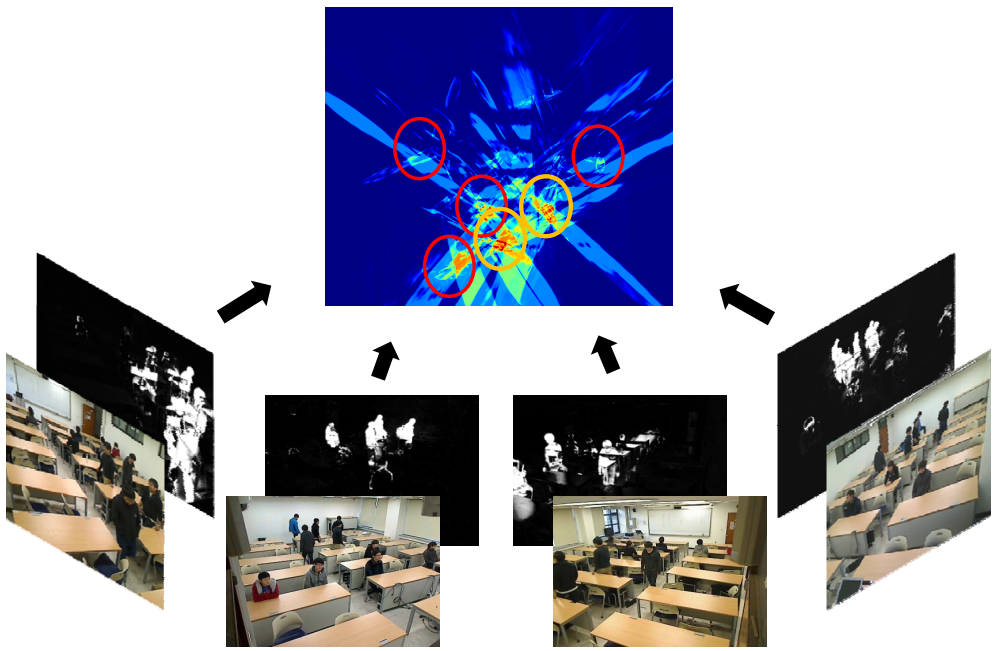

Figure 1.2: The phantom effect. When moving foregrounds are extracted by background subtraction methods (black map) and they are transformed to the single coordinate system (blue map), actual human-stand regions (orange circles) and phantoms (red circles) are both detected. For multiple target tracking purpose, those two regions should be distinguished first and data association is performed only with actual human-stand regions.

have been proposed [31, 32, 34, 35, 36, 38, 41, 42, 48] for online application purpose. These algorithms are fast and available in online application, however, they show poor tracking performance with large number of moving targets and large number of cameras. Moreover, their tracking fails frequently when the tracking model is drifted or deteriorated by neighbors because their algorithm is optimized in an online scheme and cannot be recovered from the wrong solution. If the solutions at the previous frames are incorrect, their solution at the current frame will be most probably incorrect either since the online scheme does not have the feedback procedure to fix the previous solutions. For this reason, this type of approach assumes the solutions at the previous frames are perfectly correct and choose/enumerate the solution at a certain frame based on the incorrect solution at the previous frames. This characteristic of the online approach is the main reason why online algorithms have lower tracking performance than the batch algorithms using the global optimization scheme.

1.3 Contents of research

In this thesis, we overcome the limitations of conventional approaches with various schemes and propose an online multiple objects tracking method with a single camera as well as multiple number of cameras. We first focus on the problem of the multiple objects tracking with a single camera and then extend our algorithm for multiple cameras. The inputs for data association in both cases are acquired at each frame by the object detection algorithms to get a human head detection results for the case of single camera and human body detection results for the case of multiple cameras. In the case of the single camera, the multiple object tracking problem is suffered from less amount of information than the multiple camera case. To decrease the number of the missing detection by occlusions and overlaps,

we use the head detection algorithm and use the results in the data association. Our algorithm constructs a matching graph whose nodes are detection results at each frame, 2D tracking models from the previous frame and null node for the initialization of new tracking model. The online MAP problem is formulated on this matching graph and we find matchings among those three types of nodes by Gibbs sampling method [49]. Because the proposed method does not break the causality condition and work only with observations at the current frame and the tracking models at the last previous frame, our method runs faster than global optimization methods with good performance. We evaluate the proposed multiple target tracking approach with several dataset and show improvement in performance.

In the algorithm for multiple objects tracking with multiple cameras, we extend the algorithm proposed for the case of the single camera case. We construct a matching graph whose nodes are detection results at each frame, 3D tracking models from the previous frame and null node for the initialization of new tracking model. As the single camera case, the online MAP problem is formulated on this matching graph considering the 2D information in image domain. However, in the multiple camera case, we additionally formulated the likelihood and prior probabilities with the 3D information in world domain, such as reconstruction positions with camera matrix and geometrical cues from camera installation information. Moreover, the unmatched tracking models in 2D image domain are differently updated with matching results in other cameras. Because the multiple number of cameras can provide additional information especially when occlusion happens, using multiple number of cameras to track multiple targets shows good performance even in crowded scene. Moreover, by using only the current detection and the last previous 3D tracking model, our method works faster than conventional methods based on the global optimization and shows reasonable

performance even with large number of objects and cameras.

1.4 Organization of the thesis

In chapter 2, we propose the online algorithm to track multiple objects with a single camera. With less information from the single camera, we successfully track multiple objects under several difficult conditions, such as occlusions and false positives. After describing our overall framework, we introduced the head detection algorithm we used and the online MAP formulation. The likelihood and prior probabilities are described in detail, and selective update scheme to handle missing detection and occlusion is explained. In chapter 3, we extend our algorithm with a single camera to the case of multiple cameras. After describing the used matching graph with extra added node, our online MAP formulation for the case of multiple number of cameras is introduced. In chapter 4, we finish this thesis by making conclusions and describing the possible future research directions.

Chapter 2

Multiple Target Tracking in a Single Camera

2.1 Introduction

In this chapter, we propose an online data association method with head detection results based on a matching graph to achieve a comparable performance to the global optimization approaches with a full-body detector. Because a single camera cannot provide additional information from various number of views of multiple cameras, the multiple targets tracking with a single camera is especially weak for occlusions and overlaps. To overcome these limitations, the posterior probability should be defined differently from the multiple camera case. For data association, we formulate a matching problem between two groups of nodes; nodes in one side of the matching graph are the current detection observations, and nodes in the other side represent recently updated tracking models. Different to the paper by Oh et al. [50], we construct a graph, which has a similar structure with bipartite graph, and add a new node connected to all observation vertices. By adding this

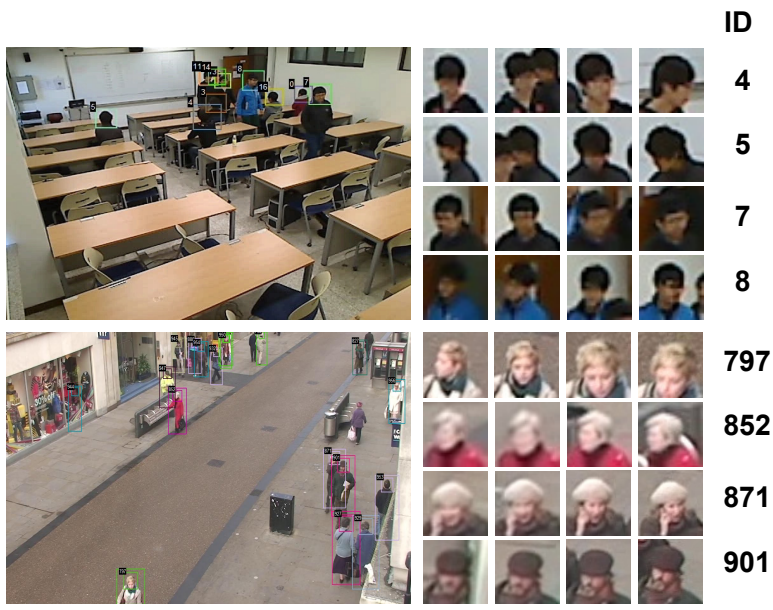

Figure 2.1: Multiple people tracking results by the proposed method in the single camera case. The images in the right are the tracked heads with the same identity at different time step. It shows that the label of the people is successfully maintained in the crowded scene.

new node, we can deal with the problem of the tracking model initialization, which is one of the main problems in both single object and multiple objects tracking. To handle insufficient discriminative features of the observations from the head detector, we build a MAP formulation for data association instead of the greedy bipartite matching scheme adopted in [22, 23, 25]. By performing sampling on the solution space of the MAP formulation considering the size, center distance, motion and appearance, a robust data association can be achieved.

By using only the last tracking models and current observations for online application, the proposed method decreases the computational complexity for the MAP problem (data association for matching) and it can be solved with sampling on the limited solution space. Moreover, the proposed solution on the modified matching graph and selective update scheme can successfully track the heads and maintain their identity even in the highly crowded scene without any prior scene knowledge.

2.2 Overall framework

The overall framework of our algorithm is illustrated in Figure 2.2. With input image, we perform object detection algorithm to find human or car, and filter out unreliable detection results by the outlier rejection method with distance and the meanshift clustering method. The reason why head detection algorithm is applied instead of human detection algorithm is that human body is easily occluded by other humans and backgrounds while human head is less occluded even in the crowded scene. Because we cannot use the different view in the single camera case, the occluded objects may not be detected, which can make the data association problem much hard. For this reason, we adopt human head detection algorithm while the human head has less information than human body,

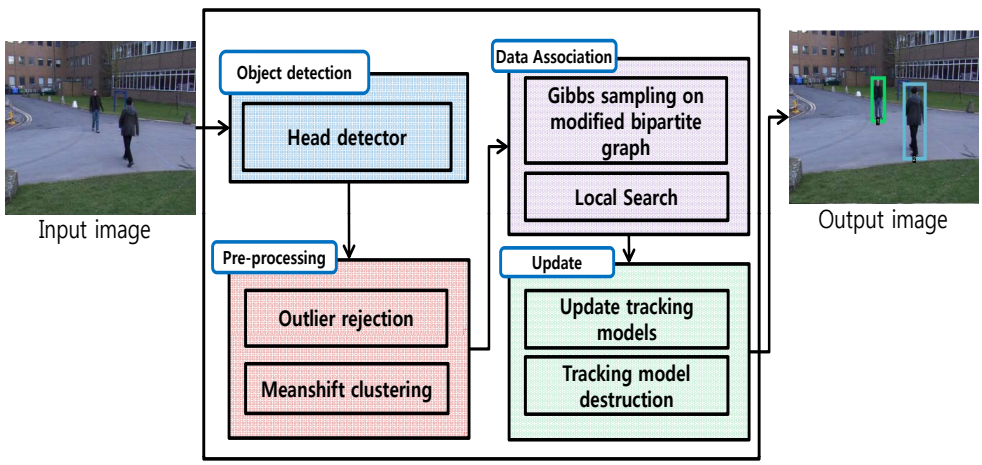

Figure 2.2: Overall framework of the proposed multiple targets tracking method in single camera case. With input image from a single camera, we detect human head in each frame and track multiple objects and maintain their labels by data association process. To associate detections and tracking models, we formulate a MAP problem and solve it by sampling.

for example, the object position in the ground plane. After noisy detection boxes are filtered by outlier rejection and meashift clustering methods, we associate them with tracking models from the previous frames. The association problem is formulated by solving a MAP problem on the matching graph. To solve the occlusion/missing detection problems, we propose a selective local search which updates/destructs the tracking models selectively with the matching results.

2.3 Detection of heads

To acquire head detection results for data association, we apply the GPU version [51] of the Histogram of Oriented Gradients (HOG) based detection algorithm

Figure 2.3: Upper row: Row head detection results by [51]. Bottom row: Filtered head detection result by the outlier rejection and the meanshift clustering algorithm. The proposed data association method is processed with these filtered boxes.

by Dalal and Trigg [44]. This method detects heads by convolution of pre-trained HOG of head sample in various scales to the current frame. Without any prior knowledge about the size of the head of the people, this method brings multiple head detection results on a single person from different scales of HOG template, and false positive detection results on human-similar region in HOG sense. This is illustrated in the upper row of the Figure 2.3. Because these noisy detection results make the data association problem much hard, we adopt two filtering methods before performing the data association to remove multiple boxes on a single person and false positives.

Each detection box, a blue box in Figure 2.3, is represented by a 4-D tuple (x, y, w, h) where x,y is its top left corner and w,h is its width and height

respectively. To filter out noisy observations, we build a voting distribution of the detection boxes in 4-D coordinate and reject the boxes which are not located in the frequently voted regions of the distribution. After rejection of outliers, we apply meanshift clustering to cluster multiple boxes to a single box which corresponds to a single person. The kernel bandwidth for the clustering is set to the same value for the whole experiments in this chapter. The results of the two filtering methods is shown in the bottom row of the Figure 2.3. However, filtering scheme cannot handle noisy observation completely (e.g. a region which has similar HOG to the head template is continuously detected as heads in the left bottom of the Figure 2.3). To handle remaining problems, we propose an online data association method based on the matching scheme using these filtered heads as observations.

2.4 MAP formulation on the matching graph

To track multiple objects over frames, the recursive Bayesian estimation method is one of the well known solutions for this purpose. The recursive Bayesian estimation method is a mathematically well-posed method and its performance is quite guaranteed. However, the recursive Bayesian estimation method requires detection results at all frames, which is not available in online application. In this section, we explain how the recursive Bayesian estimation method works for multiple targets tracking problem, and propose our own online framework to track multiple objects by solving the formulated MAP problem in the following section.

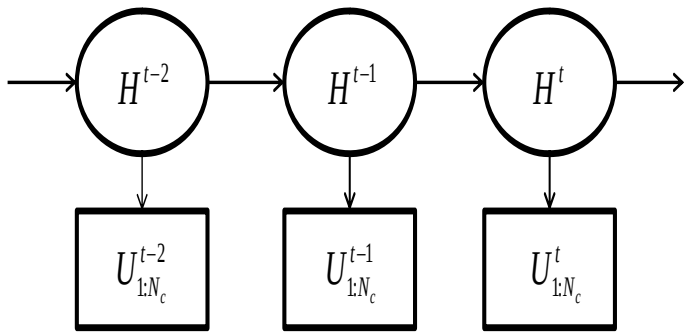

Figure 2.4: The Bayesian network for the problem of multiple target tracking. The true state of human models (H^t) can be assumed to be an unobserved Markov process, and the detection results from a camera (U^t) are the observation of a Hidden Markov Model. This Bayesian network can be solved by recursive filtering scheme, such as Kalman filter [52].

2.4.1 Recursive Bayesian estimation

Recursive Bayesian estimation, also known as Bayes filter, is a probabilistic approach to estimate an unknown probability distribution. This filter works recursively with incoming measurements and can be used to estimate the posterior probability which is the probability of states given observations. Because data association for multiple targets tracking is to find a human model corresponding to an observed detection result, we can apply the Recursive Bayesian estimation method which estimates the states (human models) for given observations (detection results). The Bayesian network for this purpose is shown in Figure 2.4. Because we want to find the tracking models associated with the detection results at every frame, the tracking model H^t can be regarded as the state and the detection results U^t can be regarded as observation.

Mathematically, we define the posterior probability function of human models given observations and estimate the human models which maximizes the fol-

lowing posterior function

$$P(H^0, H^1, \cdots, H^t | U^1, \cdots, U^t), \tag{2.1}$$

where H^t represents the set of tracking models at time t and U^t is the total set of detection results at time t. With the Bayes rule, the equation (2.1) can be rewritten as

$$\begin{aligned} & P(H^0, H^1, \cdots, H^t | U^1, \cdots, U^t) \\ = & \frac{P(H^0, \cdots, H^t, U^1, \cdots, U^t)}{P(U^1, \cdots, U^t)} \\ = & \frac{P(U^1, \cdots, U^t | H^0, \cdots, H^t) P(H^0, \cdots, H^t)}{P(U^1, \cdots, U^t)}. \end{aligned} \tag{2.2}$$

To expand the equation (2.2) more, we use the Markov assumption. The Markov assumption refers to the property of the distribution with respect to the temporal domain, which enforces that the conditional probability distribution of future states depends only upon the present state. In detail, with the Markov property, the probability of current state (H^t) given the last previous state (H^{t-1}) is conditionally independent of all the other earlier states ($H^{0:t-2}$) as

$$P(H^t | H^{t-1}, \cdots, H^0) = P(H^t | H^{t-1}), \tag{2.3}$$

and the observations at time t (U^t) is conditionally independent of all other states ($H^{0:t-1}$) given the current state (H^t) as

$$P(U^t | H^t, \cdots, H^0) = P(U^t | H^t). \tag{2.4}$$

If we apply the equation (2.3) and the equation (2.4) of the Markov assumption to the equation (2.2), then, the posterior probability for multiple objects tracking

can be expanded as

$$
\begin{aligned}
& P(H^0, H^1, \cdots, H^t | U^1, \cdots, U^t) \\
= \; & \frac{P(U^1, \cdots, U^t | H^0, \cdots, H^t) P(H^0, \cdots, H^t)}{P(U^1, \cdots, U^t)} \\
= \; & \frac{1}{P(U^1, \cdots, U^t)} P(U^1 | H^0, \cdots, H^t) \cdots P(U^t | H^0, \cdots, H^t) \\
& \cdot P(H^t | H^0, \cdots, H^{t-1}) P(H^0, \cdots, H^{t-1}) \qquad (2.5) \\
= \; & \frac{1}{P(U^1, \cdots, U^t)} P(U^1 | H^1) \cdots P(U^t | H^t) \\
& \cdot P(H^t | H^{t-1}) P(H^{t-1} | H^0, \cdots, H^{t-2}) P(H^0, \cdots, H^{t-2}) \qquad (2.6) \\
= \; & \frac{P(U^1 | H^1) \cdot \cdot P(U^t | H^t) P(H^t | H^{t-1}) \cdot \cdot P(H^1 | H^0) P(H^0)}{P(U^1, \cdot\cdot, U^t_{1:N_c})} \qquad (2.7) \\
= \; & \frac{P(H^0)}{P(U^1, \cdot\cdot, U^t)} \sum_{i=1}^{t} P(U^i | H^i) P(H^i | H^{i-1}). \qquad (2.8)
\end{aligned}
$$

With the posterior probability defined above, the state model H^t at time t can be estimated via the prediction and the update steps of the Kalman filter [52]. The prediction part of Kalman filter (prediction of current state with earlier observations) is

$$
P(H^t | U^{t-1}, \cdots, U^1) = \int P(H^t | H^{t-1}) P(H^{t-1} | U^{t-1}, \cdots, U^1) dH^{t-1}, \qquad (2.9)
$$

and the update part of Kalman filter (update current state with all observations including the observation at current time step) is

$$
\begin{aligned}
P(H^t | U^t, \cdots, U^1) &= \frac{P(U^t | H^t) P(H^t | U^{t-1}, \cdots, U^1)}{P(U^t | U^{t-1}, \cdots, U^1)} \qquad (2.10) \\
&\propto P(U^t | H^t) P(H^t | U^{t-1}, \cdots, U^1). \qquad (2.11)
\end{aligned}
$$

As we can see in the equation (2.9) and the equation (2.11), both the prediction and the update parts of the Kalman filter require all observations from the first frame to the current frame. It means that the recursive Bayesian estimation method estimates the state model at time t with all detection results until the current tth frame. However, storing all of the observations until the current frame t requires a large memory space and considering all of them for the whole video sequences to calculate the several terms (the second term of the left handside of the equation (2.9) and that of the equation (2.11)) makes the problem more complex. Moreover, because the computation time of the estimation process in this case is very closely related to the number of considered observations in posterior probability, the situation becomes even worse as the time index of the frame increases. Because the large computational load of the recursive Bayesian estimation is not suitable for online application, we propose an online MAP formulation for estimation of the state model at time t, the human tracking model H^t, only using the current observations at time t, U^t, and the one-step ahead tracking model, H^{t-1}.

2.4.2 Online MAP formulation for the single camera case

For the online data association, we propose a new matching graph scheme and solve a node matching problem in the graph. In this graphical model, we add an extra node for the automatic tracking initialization and the MAP formulation to associate detection observations and tracking models. An example of the proposed matching graph with an added node is illustrated in the Figure 2.5. The modified matching graph $G^t = (U^t, H^{t-1}, H_0, E^t)$ is composed of a vertex set of observations ($U^t = \{u_i^t | 1 \leq i \leq N_{U^t}\}$), a vertex set of human tracking models ($H^{t-1} = \{h_j^{t-1} | 1 \leq j \leq N_{H^{t-1}}\}$), an extra added node ($H_0$) for initialization of a new tracking model, and the edges between vertices

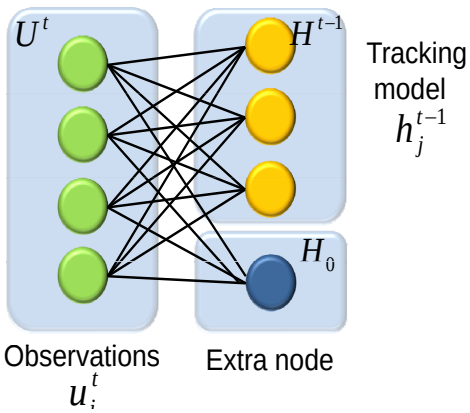

Figure 2.5: An example of the modified matching graph. U^t is a vertex set of observations, H^{t-1} is a vertex set of tracking model, H_0 is an extra added node for tracker initialization, and they are connected by non-directional edges, E. On this graph, a matching problem is solved for data association.

$(E = \{(u_i^t, h_j^{t-1}) | u_i^t \in U^t, h_j^{t-1} \in H^{t-1}\})$ which contains the similarity between nodes on the each end of edges, u_i^t and h_j^{t-1}. Then, a matching problem on the graph is defined with the following criteria: 1) the matching prefers nodes with similar features; 2) every node in the set U^t is connected to either a node in the set H^{t-1} or the extra node H_0; 3) no two nodes from the set U^t should be connected to the same node in the set H^{t-1}; 4) plural nodes from the set U^t can be matched to the extra node H_0. Instead of using the greedy bipartite matching algorithm, we formulate these criteria as a MAP problem and solve it with a sampling method. For the probabilistic formulation, we define the random variables representing the components of the matching graph as follows. The observation random vector U^t is defined as $U^t = [U_1^t, U_2^t, \cdots, U_{N_{U^t}}^t]$ and the human tracking model at time t, H^t, is defined as $H^t = [H_1^t, H_2^t, \cdots, H_{N_{H^t}}^t]$, in which H_j^t contains the random variables of position D_j^t, velocity V_j^t, and appearance model A_j^t.

As described above, we want to find matchings between nodes in the graph G.

From the second criteria, this matching should be achieved for the given current observation vector U^t, so that it maximizes the posterior probability distribution. This problem is to find the human tracking model at time t, H^t, by maximizing the following conditional probability, that is,

$$P(H^t|U^t, \hat{H}^{t-1}), \tag{2.12}$$

where $\hat{H}^{t-1}$ is defined as

$$\hat{H}^{t-1} = \underset{H^{t-1}}{\operatorname{argmax}} \; P(H^{t-1}|U^{t-1}, \hat{H}^{t-2}). \tag{2.13}$$

The equation in (2.12) can be expanded as

$$\begin{aligned} P(H^t|U^t, \hat{H}^{t-1}) &= \frac{P(\hat{H}^{t-1}, H^t, U^t)}{P(U^t, \hat{H}^{t-1})} && (2.14)\\ &= \frac{P(\hat{H}^{t-1})P(U^t, H^t|\hat{H}^{t-1})}{P(U^t, \hat{H}^{t-1})} && (2.15)\\ &= \frac{P(\hat{H}^{t-1})P(U^t|H^t)P(H^t|\hat{H}^{t-1})}{P(U^t, \hat{H}^{t-1})} && (2.16)\\ &\propto P(U^t|H^t)P(H^t|\hat{H}^{t-1}). && (2.17) \end{aligned}$$

In our MAP formulation of the equation (2.17), the first term in the right hand side stands for the likelihood probability. With the assumption of independence of each vertex given matching result, which is the current 2D tracking model H^t, we can define the likelihood probability as

$$P(U^t|H^t) = \prod_i P(U_i^t = u_i^t|H_{m(u_i^t)}^t = h_{m(u_i^t)}^t), \quad h_{m(u_i^t)}^t \in \{H^{t-1}, H_0\}, \tag{2.18}$$

where $h_{m(u_i^t)}$ is the matched human tracking node of u_i^t (either h_j^{t-1} or H_0). Then, the likelihood probability for each vertex u_i^t is,

$$P(U_i^t = u_i^t | H_{m(u_i^t)}^t = h_{m(u_i^t)}^t) = P_{2D}(U_i^t = u_i^t | H_{m(u_i^t)}^t = h_{m(u_i^t)}^t) \cdot P_M(U_i^t = u_i^t | H_{m(u_i^t)}^t = h_{m(u_i^t)}^t), \tag{2.19}$$

where P_{2D} is the 2D likelihood probability and P_M is matching likelihood probability, which are defined for the similarity measure between two nodes and the physically plausible solution respectively. First, the 2D likelihood probability is defined as

$$\begin{aligned} & P_{2D}(U_i^t = u_i^t | H_{m(u_i^t)}^t = h_{m(u_i^t)}^t \in H^{t-1}) \\ = \; & \alpha_{m(u_i^t)} \cdot exp(-\|Pos(u_i^t) - d_{m(u_i^t)}^t\|_2^2) \\ & + (1 - \alpha_{m(u_i^t)}) \cdot exp(-\sum_{q=1}^{Q} (IM_q(u_i^t) - a_{m(u_i^t),q}^t)^2), \end{aligned} \tag{2.20}$$

where $h_{m(u_i^t)}^t = (d_{m(u_i^t)}^t, v_{m(u_i^t)}^t, a_{m(u_i^t)}^t)$ and $IM(u_i^t)$ is the image patch of the detection u_i^t, $IM_q(u_i^t)$ is the qth pixel of the image patch, $a_{m(u_i^t),q}^t$ is the qth pixel of appearance model of the $m(u_i^t)$th tracking model, $a_{m(u_i^t)}^t$, and Q is the size of the image patch of the detection u_i^t.

In the equation (2.20), $\alpha_{m(u_i^t)}$ is the weight factor of Euclidean distance considering the position and velocity information in 2D likelihood probability, which is initially set to 0.5. This weight factor controls the importance of position and appearance, and is updated during the matching process: $\alpha_{m(u_i^t)}$ keeps decreasing by $\triangle\alpha$ until zero if $h_{m(u_i^t)}^t$ is not matched to any detection vertices. The value of $\triangle\alpha$ in our experiment is 0.05, which means if the 2D tracking model of the $m(u_i^t)$th 3D tracking model ($h_{m(u_i^t)}^t$) is not matched to any detection within 10 frames, appearance information is only used in the calculation of the 2D likeli-

hood probability; Once matched, the value of α is reset to 0.5. Changing α is effective to improve the data association performance because it emphasizes the appearance feature (SSD) more than the distance (EUC) in calculating the 2D likelihood probability for the tracking model which has not been matched with any detection results for periods. For example, when background-people overlaps and inter-people occlusions happen, the detection is missed for long period and actual position of the object at current time can be largely different to the position of the current tracking model. For this reason, the position information is not reliable to be used in the data association process. In this case, emphasizing appearance features more than the position in the matching likelihood can recover the drifted tracking model because the appearance of the tracking model is only updated when it is matched to the detection observation in update process, which will be described later. Controlling α with this scheme can improve the performance of the tracking algorithm.

In the calculation of of the equation (2.20), the sizes of tracking models and the detection observation are usually different because the objects keep moving. For this reason, we resize the size of the appearance model to the size of each detection when we compute the sum of squared difference. This likelihood value is only defined for the case that $m(u_i^t)$ is a tracking model h_j^{t-1} which has the position and the appearance from the past frames. When comparing the appearance models of the currently detected boxes and the tracking models, the size of the appearance model is resized for calculating the pixelwise SSD(Sum of Squared Distance).

On the other hand, the matching likelihood for plausible solution, P_M, is

defined as

$$P_M(U_i^t = u_i^t | H_{m(u_i^t)}^t = h_{m(u_i^t)}^t \in H^{t-1}) \tag{2.21}$$

$$= \begin{cases} exp(-\beta) & \text{if there exists } u_k^t, m(u_k^t) = m(u_i^t), k \neq i \\ 1 & \text{otherwise} \end{cases}, \tag{2.22}$$

where β is a large constant (100 for all experiments in this chapter). This matching likelihod probability prevents two different detection results from being matched to the same tracking model, which is not physically plausible. As described before, the tracking model initialization node (H_0) does not have appearance information, so, only matching likelihood can be defined between the detection node and this tracking model initialization node. The likelihood value for the extra added node (H_0) is defined as

$$P_M(U_i^t = u_i^t | H_{m(u_i^t)}^t = h_{m(u_i^t)}^t \in H_0) \tag{2.23}$$

$$= \begin{cases} 0.5 & \text{if } \max_{h_{s(u_i^t)}} P(U^t = u_i^t | H_{s(u_i^t)}^t = h_{s(u_i^t)}) \leq T, h_{s(u_i^t)} \in H^t \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

where T is a threshold to check whether the detection u_i^t has a close tracking model or not. If the detection u_i^t has no close tracking models, then the probability $\max_{h_{s(u_i^t)}} P(U^t = u_i^t | H_{s(u_i^t)}^t = h_{s(u_i^t)})$ will be less than the predefined T and we increase the probability to initiate a new tracking model for that detection. Otherwise, we give no chance to a new tracking model to be initiated. Unlike the matching to the tracking models, there is no penalty for the case that multiple number of detections are matched to this initialization node. Since multiple number of objects can appear at the same frame, it is required to allow multiple matchings to the node H_0. Because this likelihood probability is influenced by the matching configuration, discrete bipartite matching algorithm, such as Hun-

garian method, cannot model this likelihood.

The prior probability, $P(H^t|\hat{H}^{t-1})$, is designed to enforce motion dynamics of objects over frames. This prior probability is defined as

$$P(H^t|\hat{H}^{t-1}) = \prod_{i}^{N_p(H^t,\hat{H}^{t-1})} P(H_i^t = h_i^t|\hat{H}_i^{t-1} = \hat{h}_i^{t-1}) \tag{2.25}$$

with the assumption of the independence between motion dynamics of objects. N_p is the number of pairs, and only paired tracking models in consecutive frames are calculated for the prior probability value. Then, the equation (2.25) is calculated as

$$\begin{aligned} & P(H_i^t = h_i^t|\hat{H}_i^{t-1} = \hat{h}_i^{t-1}) \\ = \; & P(D_i^t = d_i^t|\hat{D}_i^{t-1} = \hat{d}_i^{t-1}, \hat{V}_i^{t-1} = \hat{v}_i^{t-1}) && (2.26) \\ = \; & exp(-|d_i^t - (\hat{d}_i^{t-1} + \hat{v}_i^{t-1})|). && (2.27) \end{aligned}$$

This prior probability gives high values for the closely located people or slowly moving objects, which is used to find closely located tracking model at time t with respect to the tracking model at time $t-1$.

With the posterior distribution from the likelihood and the prior defined in the modified matching graph, a Gibbs sampling method is adopted to get a MAP solution to find the matching between nodes. Because the solution space of our data association problem is small by using only U^t and H^{t-1}, iterative solution by Gibbs sampling does not require a large number of iterations and can solve the matching problem fast. After performing the data association with Gibbs sampling, the different update procedures are processed from the association result, which is shown in the Figure 2.6. If a detection vertex is connected to the tracking model vertex as the Figure 2.6 (a), the appearances and motions of

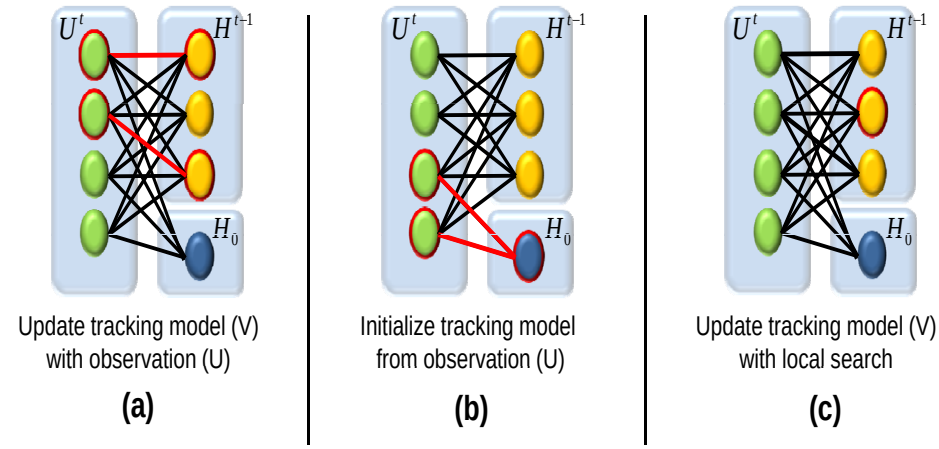

Figure 2.6: The three different procedures with association results. (a) Update the tracking model (H^{t-1}) with the observation (U^t). (b) Initialize the tracking model (H^{t-1}) from the observation (U^t). (c) Update the tracking model (H^{t-1}) with local search result.

the tracking model is updated with the matched observation. A detection vertex, which is connected to the extra added node H_0 as the Figure 2.6 (b), initializes a new tracker and gives it a new label. This MAP solution on the modified matching graph performs a robust data association even with head detection results which are less discriminative than full body detections. Moreover, because the solution space of sampling contains the tracking model initialization with the node H_0, the data association step itself can initialize the tracking model different from the traditional bipartite graph. In the case of the Figure 2.6 (c), the tracking model is not matched with any detection vertex. This means that there was a tracking model for certain person but the person is not detected in current frame. In this perspective, the position of the person should be estimated without observation, and we propose a selective update method to estimate the position of the tracking model which is not matched to any detection node as Figure 2.6 (c). This scheme

is explained in the next section.

2.5 Selective update to handle occlusions

After the data association step, there can exist the 2D tracking models which are not matched to any detection observations from the incomplete performance of the detectors as the Figure 2.6 (c). In this case, we search the image in the local region of the all the 2D positions of the unmatched tracking model in each camera and set the most plausible position in SSD sense as the new position of the 2D tracking model in image coordinate system, similar to the conventional tracking methods. However, some of the tracking models are unmatched from the reason of the missing detection by occlusions. When occlusion happens, the true position of the object is actually invisible and there exists no reliable information to update the tracking model. This is illustrated in the Figure 2.7 (a) and (b). In Figure 2.7, because of the occlusion by the background structure and other person, the person of the red box (Figure 2.7 (a)) are hidden in the yellow box (Figure 2.7 (b)) and the simply searched result of the person of the red box in the yellow box can be incorrect. For this reason, updating the tracking model with local search result can cause the drift or the corruption of the tracking model and should be done carefully.

To decide whether we accept or reject the local search result as the source to update the tracking model, we apply a selective accepting scheme for the occlusion reasoning method. This algorithm is explained in Algorithm 1. In Algorithm 1, the candidate positions are chosen in grid around the position of the tracking model. Then, for all candidate positions in the local region, SSD values are calculated and their spatial distribution is considered in decision. In detail, we trust the local search result if the spatial distribution of the SSD has a solid minimum point,

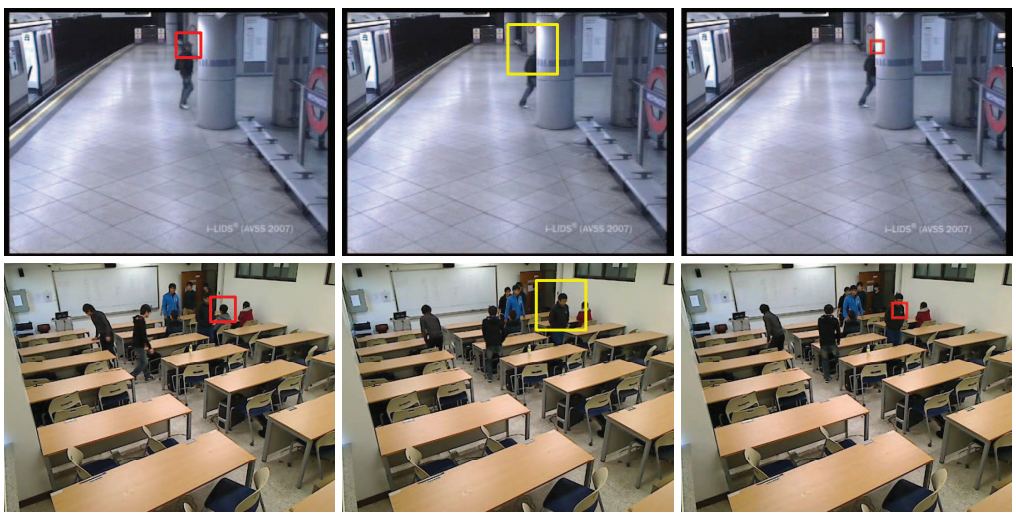

Figure 2.7: The examples of bacgkround-object and inter-object occlusions. (a) currently tracked model (b) occlusion happens in yellow box and detection is missing: update tracking model can corrupt the model. (c) the result by the proposed selective update

Algorithm 1 The selective update for a tracking model

Input: The unmatched tracking model at $t-1$ time step, h_j^{t-1}
(2D position: d_j^{t-1}, 2D velocity: v_j^{t-1}),
A set of candidate positions for the tracking model h_j^{t-1}, $CP(h_j^{t-1})$.

Output: The position of the tracking model h_j^t at t time step, d_j^t.

for all candidate points in the set $CP(h_j^{t-1})$ **do**
evaluate $SSD(d_j^{t-1}, P'), P' \in CP(h_j^{t-1})$
(evaluate SSD value between the image patch with center in d_j^{t-1} and the image patch with center in one of the candidate points)
end for
calculate
$\mu = \frac{1}{n} \sum_{P'} SSD(P, P')$,
$mn = \min_{P'} SSD(P, P')$,
$\sigma^2 = \frac{1}{n} \sum_{P'} (SSD(P, P') - \mu)^2$, $P' \in CP(h_j^{t-1})$
if $\frac{\mu - mn}{\sigma} \leq T$ **then**
$d_j^t := d_j^{t-1} + v_j^{t-1}$
else
$d_j^t := \arg\min_{P'} SSD(d_j^{t-1}, P')$,
$P' \in CP(v_j^{t-1})$
end if

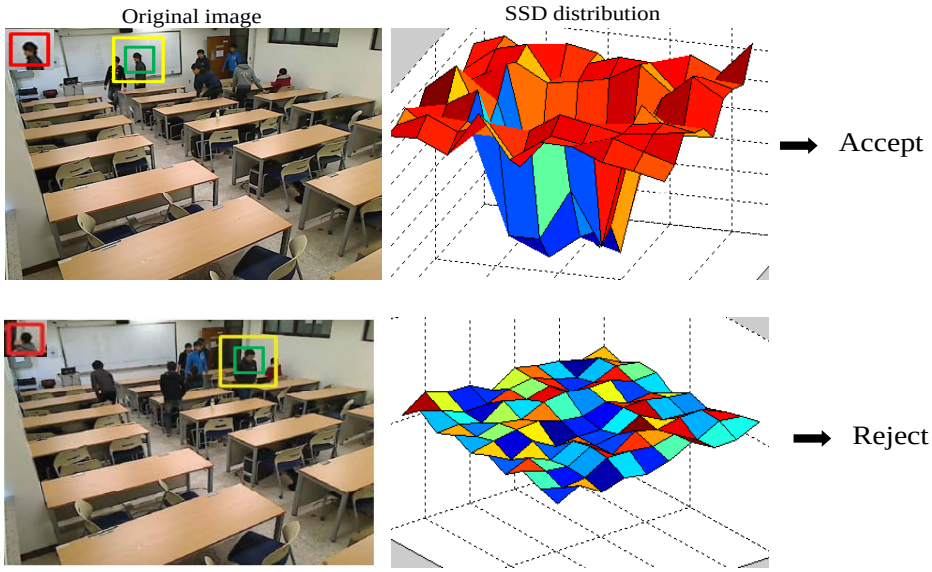

Figure 2.8: The example of accepting and rejecting the local search results with SSD distribution. When SSD distribution has strong minimum point, we accept the local search results. Otherwise, the local search result is rejected and the position of the tracking model is estimated with its previous position and its velocity.

that is, the mean and the minimum of the distribution show large difference. Otherwise, we reject the local search result and update the position of the tracking model with the velocity of it at the previous time-step. The velocity of the tracking model h_j^t is calculated by

$$v_j^t = v_j^{t-1} + \gamma \cdot (d_j^t - d_j^{t-1}), \tag{2.28}$$

where γ is a velocity smoothing factor, which is set to 0.9. The velocity of the tracking model is updated with the equation (2.28) with the data association and the selective update results. The example of accepting and rejecting the local search results with SSD distribution is illustrated in Figure 2.8.

When a tracking model is not matched to any detection observations in data

association step and the model is updated by its previous velocity from the rejection of the search result more than a certain number of frames, then, the tracking model is terminated. The number of frames for termination used in the experiment was 20.

The proposed selective update method can prevent the tracking models from being corrupted by unreliable observations and being drifted to incorrect positions. As a result, it significantly reduces the number of identity switches and false positives. The effect of the selective update method can be seen in the quantitative experimental results.

2.6 Experimental results

In this section, we present the test results of the proposed algorithm on several dataset, four public datasets, i-Lids AVSS 2007 dataset, Oxford Town Center dataset [21], PETS 2007 dataset [53] and PETS 2009 dataset [54] and two our own video sequences from the *Smart Class* dataset. The quantitative results are evaluated by CLEAR MOT metrics, Multiple Object Tracking Precision (MOTP), Multiple Object Tracking Accuracy (MOTA), the detection precision, and the detection recall from the paper [55]. MOTP is the precision score from intersection over union of estimated tracking box and ground truth, and MOTA is calculated with the number of false positives, false negatives, and identity switches. These scores are measured in 2D image domain with detection and tracking box information. In detail, MOTP is the precision score from intersection over union of estimated tracking box and ground truth, and it is defined as

$$\text{MOTP} = \frac{\sum_{i,t} e_t^i}{\sum_t c_t}, \tag{2.29}$$

where e_t^i is the number of errors in estimated position for the ith matched object-hypothesis pairs and c_t is the number of total matches made at time t. If the intersection over union of estimated tracking box and ground truth is less than threshold, the matching is counted as error, otherwise, correct match. This MOTP measures the ability of the tracking algorithm to estimate the positions of the object precisely. For this reason, the value of MOTP measure is closely related to the ground truth annotation quality. Since the ground truth positions of people are generally annotated on the image plane by users, it can be not as precise as the actual positions of the moving targets. On the other hand, MOTA is calculated with the number of false positives, false negatives, and identity switches as

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \tag{2.30}$$

where m_t, fp_t, mme_t, and g_t are the number of misses, false positives, miss matchings, and ground truth, respectively, at time t. For both $MOTP$ and $MOTA$, the high value of MOTP/MOTA is better than the low values. We compared the proposed method to the state-of-art methods and to the proposed method without the selective update scheme. Also, the qualitative results visualize the performance of the proposed multiple people tracking method. In all experiments, data association does not assume any prior knowledge about the scene structures, but estimating body regions of the people from their tracked head position is done with the ground plane calibration to compare with performance of existing methods. For the *Smart Class* dataset, the torso region is used to evaluate the tracking performance because some people sit on the chair and their body regions are not visible. For PETS 2007 and PETS 2009 dataset, we only show qualitative results in figure since we do not have ground truth data for those datasets.

To show the effectiveness of the proposed varying likelihood probability between detection and the initialization node H_0, we modified the Hungarian method

[56] by adding the same number of initialization nodes with the detection nodes and assigned the same likelihood probability for all initialization nodes. For simplicity, we refer this algorithm as *m-hung* below in tables. The measure values of this algorithm are in Tables of each dataset, and we can see that our proposed algorithm works better than *m-hung* with every dataset while it works 3 times faster than our algorithm. However, our algorithm already works about 5 fps in MATLAB without code optimization, which is expected to work in real-time if the code is implemented in C++ with code optimization. The reason why our varying likelihood probability works better than the conventional fixed likelihood probability for the tracking model initialization is that we assign the probability of the initialization of a certain detection considering its similarity with currently existing tracking models. By comparing the similarities between detection and tracking models, we encourage the initialization of the tracking model when a certain detection is most likely to be newly appeared one or block the initialization completely if there exist a similar tracking model to the detection observation. On the other hand, the fixed probability for initialization can initialize the tracking model even when the detection is not a newly appeared object or do not initialize the tracking model when the corresponding detection is a new appeared target with the probability of the fixed value. Also, the tracking performance is very sensitive to the value of this fixed likelihood probability for the tracking model initialization. When this value is too high, too many labels are initialized and one object has several number of labels, and when the initialization probability is too low, different objects can share one label, which is incorrect. The performance of the algorithm *m-hung* is described in each dataset.

2.6.1 iLids 2007 AVSS dataset

The iLids 2007 AVSS dataset is a widely used dataset to evaluate the performance of multiple people tracking methods. We tested iLids AB easy dataset. The resolution of this dataset is 720×576 and the frame rate is 25fps. In this dataset, there exists a few inter-object occlusions, but people are frequently occluded by the pillar in the background. Also, the HOG-based head detector shows weak performance for the people close to the platform. For the quantitative comparison, we estimated the body regions from our head tracking with ground plane information as [21] and compared the result to state-of-art methods in Table 2.1. The proposed method outperforms the existing methods in MOTP and has comparable performance in MOTA, precision and recall compared to them. Figure 2.9 shows that our algorithm successfully maintains the label of tracked people.

Method	MOTP	MOTA	Prec	Rec
Stalder et al. [17]	-	-	89.4%	53.3%
Benfold et al. [21]	73.6%	59.9%	80.3%	82.0%
m-hung	85.5%	59.8%	81.5%	77.4%
Breitenstein et al. [22]	67.0%	78.1%	-	83.6%
Ours w/o s.update	85.4%	59.8%	68.7%	72.1%
Ours	87.1%	63.6%	84.3%	78.2%

Table 2.1: The quantitative results for the iLids 2007 AVSS dataset. We evaluated the proposed method and the proposed method without selective update method on full-body region estimated from the head tracking results with camera calibration.

Figure 2.9: The qualitative results for the iLids 2007 AVSS dataset. The squares are detected heads and the rectangles are estimated body of each person.

2.6.2 Oxford Town Center dataset

We evaluated our method with the Oxford Town Center dataset with resolution of 1920×1080 and frame rate of 25fps. This sequence has a semi-crowded people and few long-term occlusions. Most of people show a linear motion. We did experiment with the same settings in several state-of-art methods and the quantitative result comparison is shown in Table 2.2. For head region, only [21] has the result, and our method works better in MOTA, precision and recall. For body region, the proposed method shows the best MOTP, precision, and recall values and comparable MOTA to the state-of-art methods. This quantitative result of the state-of-art algorithms on the Oxford Town Center dataset is from [21] and [25]. The output images by the proposed method are shown in the Figure 2.10.

2.6.3 PETS 2007 and PETS 2009 dataset

For PETS 2007 and PETS 2009 dataset, only qualitative results are shown in the Figure 2.11. Among different video sequences in PETS 2007 data set, we used the sequence *S06-BAG-STEAL* because this video is more suitable to evaluate the multiple people tracking method rather than the sequence used by [21]. For PETS 2009 dataset, we tested the *S2.L1 walking* sequence with the proposed method. In both sequences, our algorithm shows reliable performance in maintaining the identities of people.

2.6.4 Smart Class dataset

The Smart Class dataset is composed of four synchronized video sequences captured by different cameras in the same classroom and we used two sequences from them. The frame resolution of these sequences is 640×480, and the frame length is 883. In these videos, tables cover the lower body of people and it is

		Head regions				Body regions			
	Method	MOTP	MOTA	Prec	Rec	MOTP	MOTA	Prec	Rec
Delayed *or* Global optimization	Benfold et al. [21]	69.9%	56.6%	73.8%	71.0%	80.4%	64.8%	80.5%	64.9%
	Zhang et al. [16]	-	-	-	-	71.5%	65.7%	71.5%	66.1%
	Leal-Taixe et al. [19]	-	-	-	-	71.5%	67.3%	71.6%	67.6%
	Pellegrini et al. [20]	-	-	-	-	70.7%	63.4%	70.8%	64.1%
	Izadinia et al. [57]	-	-	-	-	71.6%	75.7%	70.8%	64.1%
	Zamir et al. [58]	-	-	-	-	71.9%	75.6%	-	-
Online optimization	Shu et al. [25]	-	-	-	-	71.3%	72.9%	71.4%	73.5%
	Wu et al. [59]	-	-	-	-	68.7%	69.5%	-	-
	Yamaguchi et al. [60]	-	-	-	-	71.7%	66.6%	-	-
	m-hung	63.6%	56.2%	77.8%	77.0%	84.5%	64.2%	82.4%	79.5%
	Ours w/o s.update	63.1%	57.6%	72.2%	71.6%	81.1%	65.2%	79.3%	77.1%
	Ours	63.7%	61.4%	83.3%	77.1%	85.6%	66.8%	86.3%	79.8%

Table 2.2: The quantitative results for the Oxford Town Center dataset. The full-body region is estimated from the head tracking results with camera calibration.

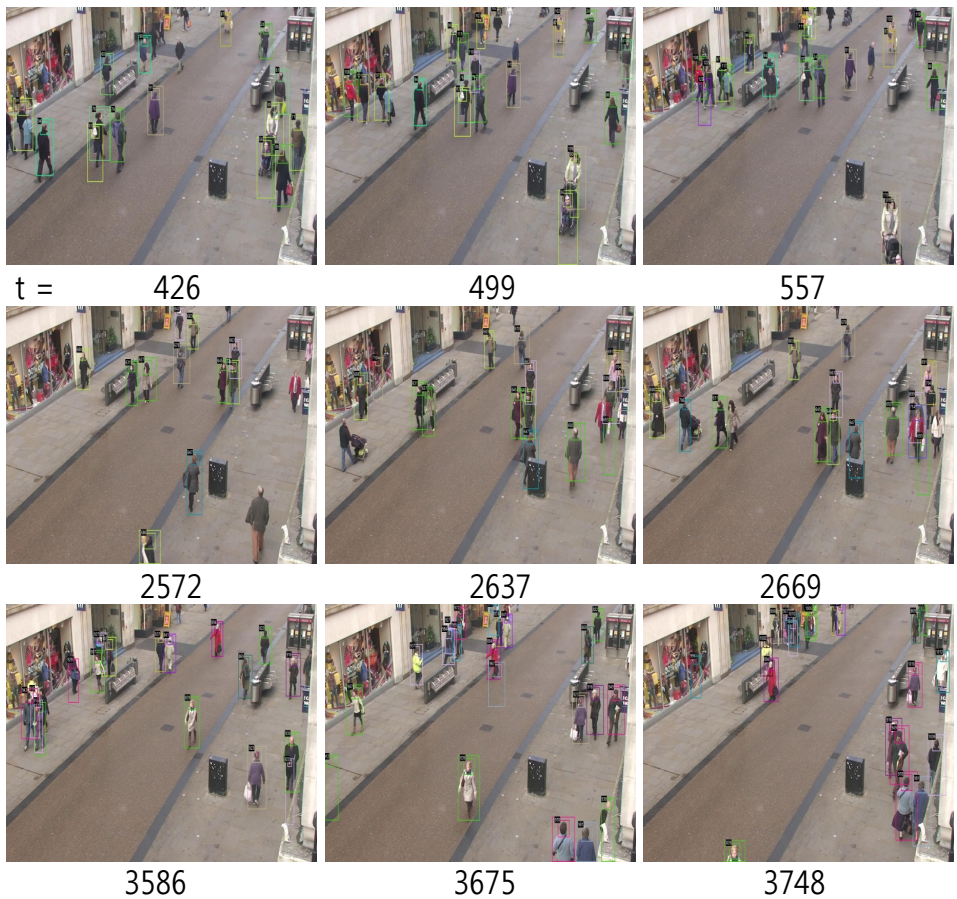

Figure 2.10: The qualitative results for the Oxford Town Center dataset. The squares are detected heads and the rectangles are estimated body of each person.

Figure 2.11: The qualitative results for the PETS 2007 (upper two rows) and the PETS 2009 (bottom two rows). The squares are detected heads and the rectangles are estimated body of each person.

hard to use other types of detectors, such as the full-body detector. There exist long-term occlusions with highly-crowded people, some of them sitting on the chair for long period or walking around with non-linear motions and making even full occlusions. Missing detections by occlusion happen more than other public datasets. For tracking, the similar appearances of heads and different scales of the head boxes with respect to their positions are the challenging problems in these sequences, however, our algorithm maintains the identities of people successfully as shown in the Table 2.3, the Figure 2.12, and the Figure 2.13.

Method	Parts	Video	MOTP	MOTA	Prec	Rec
m-hung	Head	1	84.5%	75.5%	91.8%	83.7%
		2	85.9%	91.5%	95.0%	97.0%
Ours	Head	1	85.8%	79.1%	94.1%	85.2%
		2	86.5%	93.5%	95.6%	98.0%
Ours w/o s.update	Head	1	85.1%	74.3%	91.0%	83.4%
		2	86.1%	88.7%	91.5%	97.6%
m-hung	Torso	1	84.7%	75.6%	91.2%	84.7%
		2	86.1%	91.7%	94.9%	97.4%
Ours	Torso	1	85.4%	78.9%	93.4%	86.1%
		2	86.3%	93.9%	95.5%	98.5%

Table 2.3: The quantitative results for the first and the second video sequence of *Smart Class* dataset. We evaluate the proposed method, the proposed method without selective update method (both on heads) and the proposed method on torso regions.

2.7 Final remarks and discussion

In this chapter, we proposed an online data association for tracking multiple people with a single camera in highly crowded scenes. We encoded the multiple people tracking problem to the matching problem on the modified matching graph and solved it with the MAP formulation. The solution can be calculated fast

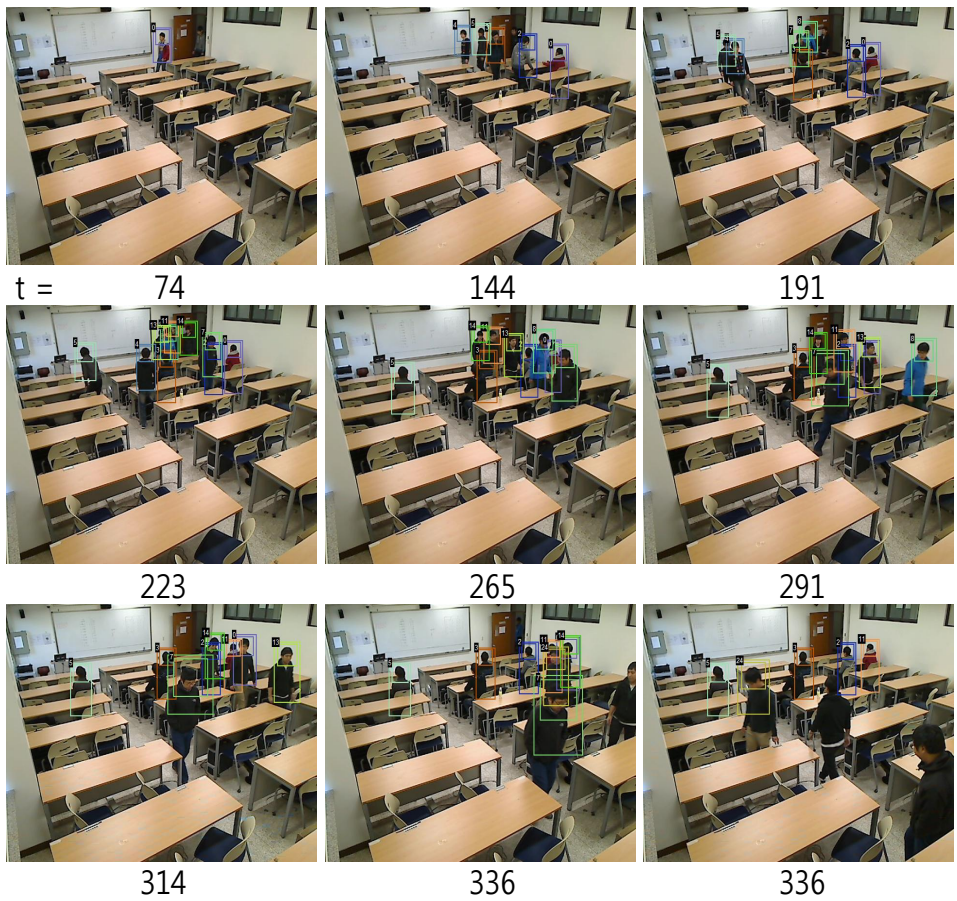

Figure 2.12: The qualitative results for the first video sequence of *Smart Class* dataset. The squares are detected heads and the rectangles are estimated torso of each person.

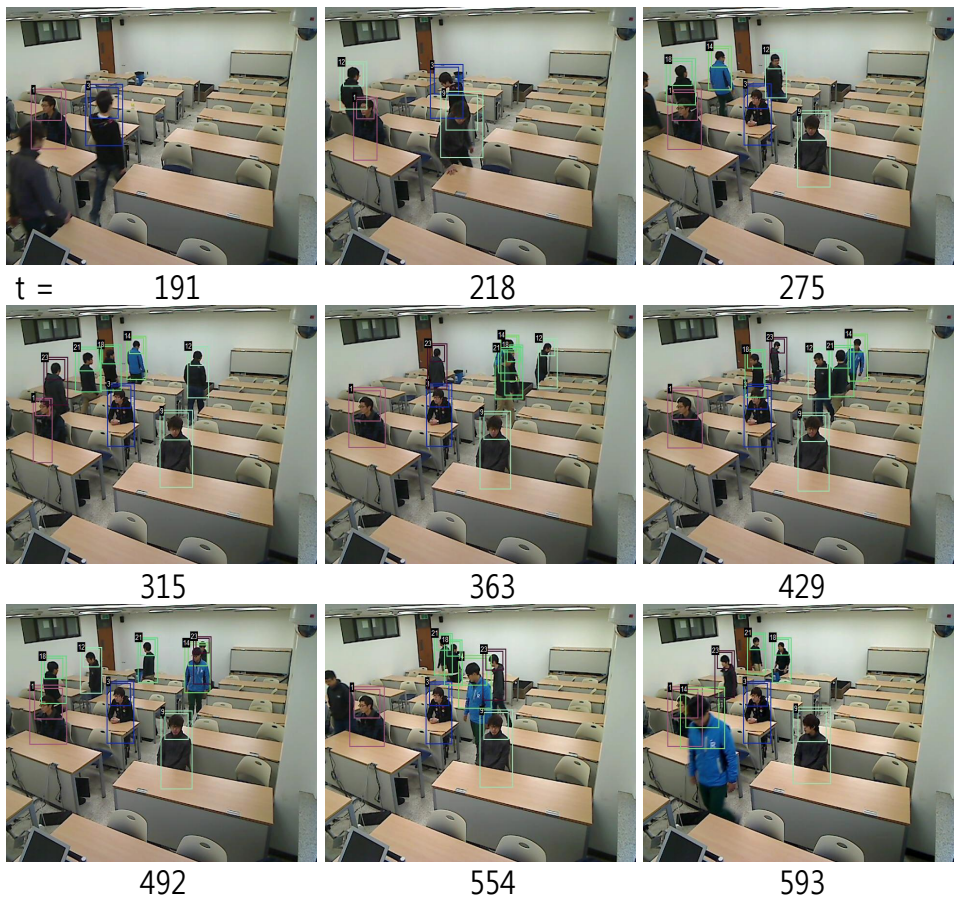

Figure 2.13: The qualitative results for the second video sequence of *Smart Class* dataset. The squares are detected heads and the rectangles are estimated torso of each person.

with a sampling method because the solution space of our formulation is small. Selectively updating scheme for tracking model handles the occlusions and deals with the missing detection from poor performance of detectors. Our quantitative and qualitative evaluations show that our method tracks multiple people and maintains their identity successfully comparable to the state-of-art algorithms.

Chapter 3

Multiple Target Tracking in Multiple Cameras

With increased number of cameras and improved system architecture in software and hardware sense, current surveillance system or sport broadcasting system are mostly composed of several number of cameras. The examples of areas using multiple cameras are illustrated in Figure 3.1. Multiple number of cameras can cover more wide areas than a single camera and handle occlusions among people or hidden objects behind backgrounds by watching scene from different views: overlapped people in one view can be located apart in other view and invisible people behind backgrounds in one view can be visible from other camera views. In this sense, using multiple cameras increases the effect of the surveillance system significantly. However, building a surveillance system to track multiple objects with multiple cameras has several issues to be solved. One of the most important problems is the "*who is who*" problem. In multiple cameras, one person can appear differently in each camera and assigning the same label to that person in all camera views is required as well as the tracking of that person in each camera.

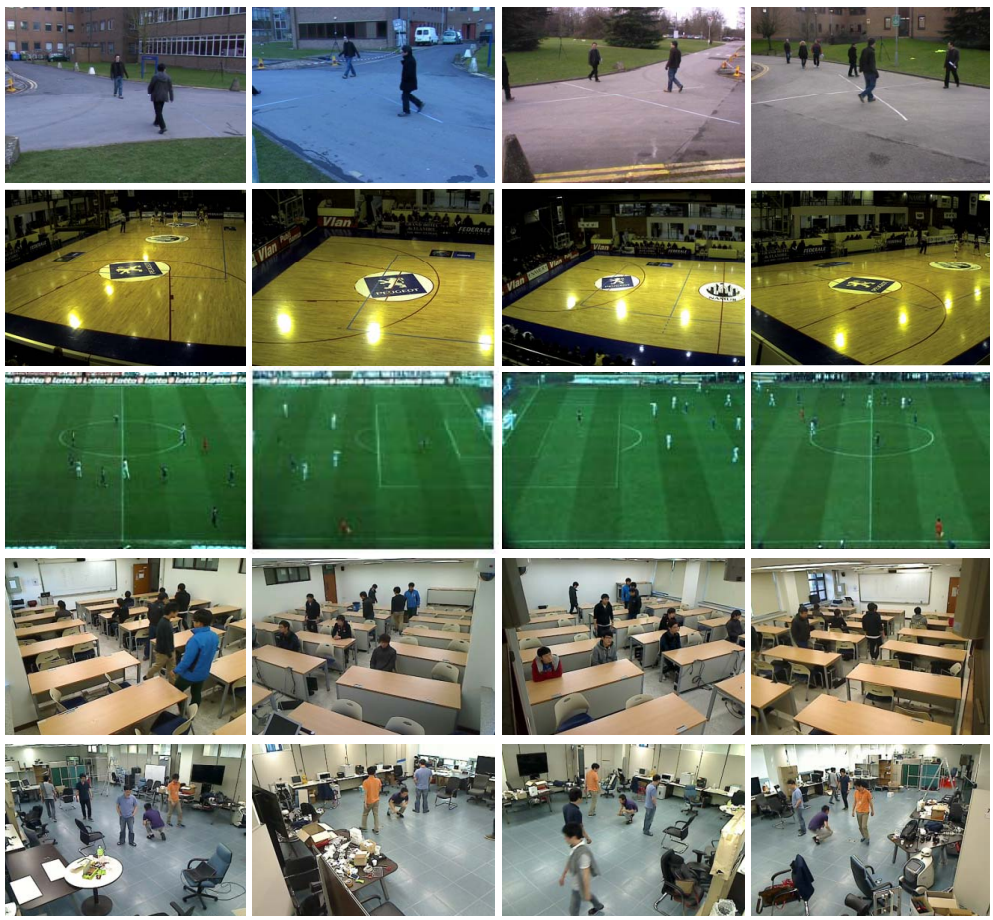

Figure 3.1: Examples of using multiple cameras. The multiple cameras can be used for surveillance purpose and in sports analysis. Because multiple number of cameras can cover wide areas and handle occlusions among people or hide behind backgrounds, using multiple cameras increases the effect of the surveillance system significantly.

The assignment of the labels over views is called spatial association, and that of the labels over frames is called temporal association. The second problem is the online availability and computational load. For the purpose of surveillance and sports analysis, the tracking should be done in online manner and fast for the instantaneous response to the criminal/abnormal behaviors of people. However, the amount of data to be associated and the possible number of matching configuration in tracking increases largely with the number of cameras and it is hard to show good tracking performance within the online framework.

In this chapter, we extend the proposed multiple target tracking method for the case of the single camera and propose a novel online method for tracking multiple objects with multiple cameras. By exploiting various number of views from different cameras, we aim to find spatial and temporal association of objects with less computation load using objects information, such as 2D information (position, velocity, and appearance) and 3D information (position), and geometric information, such as camera matrix. Our association maintains the labels of objects in the spatial and temporal domain. For this purpose, we formulate an online MAP problem on the matching graph whose nodes are detection results at each frame, 3D tracking models from the previous frame and null node for the initialization of new tracking model. We solve the formulated MAP problem with the Gibbs sampling method [49]. Our method can automatically initialize tracking models and successfully handle occlusion and missing detection problems by selective local search scheme. Since the performance of detection algorithms is incomplete, missing detection usually happens and it decreases the performance of the data association. To solve this problem, existing approaches apply the particle filter [22, 24] and the meanshift tracker [61, 23], and use the result of the tracking algorithms to update the tracking model. However, in occlusion scene, updating the tracking model with observations can corrupt the model since the

observations can be unreliable. To prevent the tracking model from being corrupted, we reason an occlusion for the tracking model with spatial information and selectively update the tracking models only with reliable observations. By updating the tracking model selectively with the evidence from it, the proposed scheme works more robust to occlusions than the existing tracking-by-detection methods. We evaluate the proposed approach with several dataset composed of multiple number of cameras and show improvement in performance.

3.1 Overall framework

Figure 3.2 shows the overall framework of the proposed method to track multiple objects with multiple cameras. With each input sequence from cameras, we perform a human detection algorithm using cascade deformable part model (DPM) [45] at each frame. Any detection algorithm, such as HOG-SVM based detector [44] or crosstalk [62] can be used. Then, we build a matching graph whose nodes are detection results at current frame, 3D tracking models from previous frames, and an extra node for the initialization of new tracking model. To find matchings between nodes in the graph, we formulate an online MAP problem to find the matching in the constructed graph. Our formulation considers 2D positions, velocities, appearances in image coordinate system, and 3D position in world coordinate system for the similarities among nodes. In addition, we enforce some conditions for matchings, for example, two detection nodes in one camera should not be matched to the same 3D human tracking model. The matchings on this graph are found by applying Gibbs sampling method to maximize the formulated posterior probability. After that, we update all 3D tracking models with the associated detection results and update the unmatched tracking models by the selective local search scheme within the neighborhood. The proposed selective

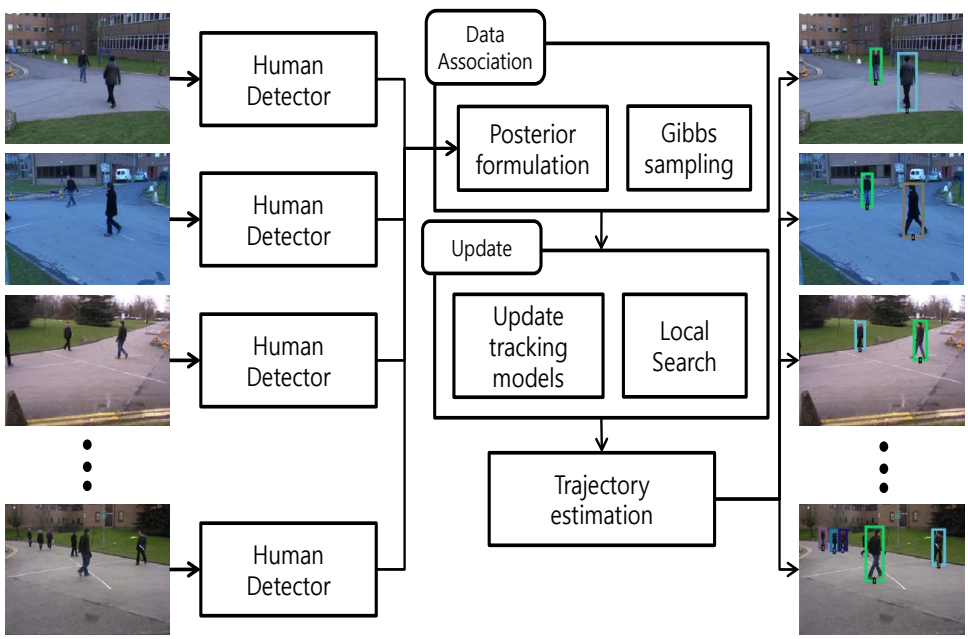

Figure 3.2: Overall framework of the proposed algorithm. In the proposed method, we perform human detection algorithm on images from each camera view. Then, we formulate an online MAP problem to associate those detection results at current frame and tracking models at last previous frame, and the solution is found by Gibbs sampling method. With calculated matching configuration, we update the tracking model with different strategies. When a certain human tracking model is not matched to any detection in all cameras, we update the models by the selective update scheme.

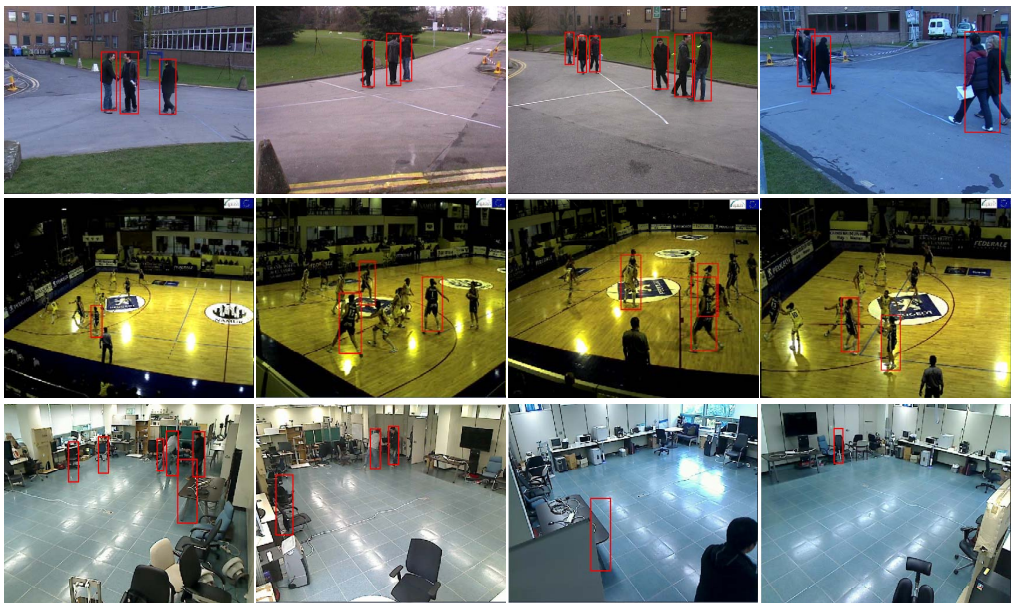

Figure 3.3: Human detection results for multiple camera case. The detection of human is performed by cascade deformable part model [45]. The false positives and missed detections may exist and they are burden to achieve a good performance in tracking multiple objects.

local search scheme can prevent the tracking model from being corrupted by incorrect association or missing detection from occlusions. Finally, the trajectories of all people are estimated in the world coordinate domain and the association results are re-projected to the each camera coordinate to show the multiple targets tracking results in all camera sequences of image domain.

3.2 Detection of humans

To acquire the input for data association, we apply one of the well known human detection algorithms, which is the cascade DPM [45], to the input images from multiple number of cameras at every frame. This method is the improved

version of the original DPM based detection algorithm [63] in the sense of computational time. In the original DPM based detection algorithm [63], it trains the interesting object model which is composed of several deformable parts. Then, this algorithm detects the interesting object, which is a human in our case, in the scene by performing the convolution operation between every part of the various sized pre-trained models and the input image to detect the human in the scene. For this reason, it requires a large computation time. On the other hand, the cascade DPM [45] works much faster than the original algorithm by cutting the convolution process with the similarity score of each part in the trained model. When one part of the trained model shows low convolution score with a certain region of the input image, then this region is eliminated in the convolution process with the other parts of the trained model. The results of this human detection algorithm in various scenes are illustrated in Figure 3.3.

The cascade DPM method is a good object detection algorithm especially when occlusion occurs because they divide the human model into several deformable parts and compare those parts separately with the image candidate region. As described in [64, 65], this algorithm is known to work better than conventional object detection algorithms such as [66, 67, 68, 69, 70]. When some parts of a human are occluded, the other parts of that person show a good convolution score and it makes the person under part-occlusion be detected well. As you can see in Figure 3.3, some of people in the scene are detected even under occlusions. However, the performance of this detection algorithm is not perfect, and there still exist many missed detections and false positives. The missed detection is the undetected human in the scene and the false positives mean the detected regions which are not actually part of human. The missed detection usually happen when the training data is irrelevant and it is very different from the current test data. However, it can also happen when one human is occluded by others partly or

completely and when the human is overlapped by backgrounds. In the middle row of Figure 3.3, many missed detections happens in the case of the people with dynamic shapes and the people under severe occlusions while some of them under small occlusions are successfully detected. When the missing detection problem happens, the position and the velocity of people should be updated with other information, such as motion model or local search based on appearance, to assign the correct labels to people. These alternative solutions can maintain the tracking performance in short period, but they are failed easily when complete occlusion happens or the undetected target human moves dynamically. This leads to the deterioration of the tracking models which makes the data association hard and, at last, the change of labels of objects or drift of the tracking model happens.

On the other hand, the false positives, which are usually located in the stationary background, can be repeatedly detected when they have similar patterns to the pre-trained human model because the stationary background rarely changes in frames. For example, as we can see in the bottom row of Figure 3.3, false positives are detected in the corner of the table or near the region of the chairs, and these false alarms are detected over and over because they do not move during the entire frame of the sequences but have similar patterns to the pre-trained human body model. It is hard to distinguish those false positives from the true positive detection results, however, they should be removed because tracking those uninteresting objects is not suited for surveillance purpose. Moreover, since humans keep moving over frames, they can approach to the false positives and closely located false positives might steal the label of the approaching people. From this perspective, the false positives are burden to accomplish high performance in data association.

As described above, the missed detections and the noisy detection results, such as false positives, can decrease the tracking performance significantly. To

compensate these insufficient detection results in the tracking phase, we formulate a MAP problem based on several 2D and 3D information, and use an online data association method based on the matching scheme in graph.

3.3 MAP formulation on the matching graph

3.3.1 The matching graph

For tracking multiple objects with the data association scheme, we encode the data association problem to the matching problem in a graph. In this section, we explain the conventional K-dimensional matching scheme in the K-partite graph for tracking multiple objects with multiple number of cameras and then describe our proposed matching graph for the same purpose to show the difference between conventional matching graphs and the proposed one. The conventional K-partite graph is usually used to associate detection observations in spatial domain and assign the same label to the same objects in different camera views, and they find the spatial association and the temporal association separately by concepts of *Reconstruction-Tracking* or *Tracking-Reconstruction*. On the other hand, the proposed matching graph is used to find both the spatial and temporal association simultaneously. For this purpose, the proposed matching graph has the nodes of 3D tracking models and matches them with detection observations in various cameras.

3.3.1.1 The conventional K-partite matching graph

The K-partite matching problem is a generalization of *the bipartite matching problem*. While the bipartite matching problem is defined as finding the connection between nodes in the two disjoint set, the K-partite matching problem deals with

K number of disjoint sets. It is one of the well known *NP-hard* problems. When K-partite matching problem finds matchings among K disjoint sets, the matching can be represented by K-tuples, where each element of the tuple is from each disjoint set and any distinct two matchings have no common node from the same set. The examples of K-partite graph and K-partite matching are illustrated in Figure 3.4. Figure 3.4 (a) is general K-partite matching graph with K number of disjoint set, and Figure 3.4 (b) is the examples of K-partite matching, where a shaded cylindrical link from the set S_1 to the set S_K is one matching. For example, in the top right graph in Figure 3.4, there exist two people associated together, and four people in the bottom left and in the bottom right graph. In K-partite graph, the number of matchings is decided via the defined objective function to optimize, and it is not required to find the maximum number of K-partite matching.

The conventional algorithms using K-partite matching scheme use the set of detection observations from each camera view as the disjoint set for the K-partite graph. That is, every node in the K-partite graph is actually a detection observation. From this perspective, the K-partite matching in the K-partite graph represents the spatial association among detection observations of different cameras. This across-camera data association problem is known as multidimensional assignment problem, and many conventional approaches solve this problem by minimizing the following linear cost function

$$c = \min \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_K=1}^{n_K} c_{i_1,i_2,\cdots,i_K} x_{i_1,i_2,\cdots,i_K} \tag{3.1}$$

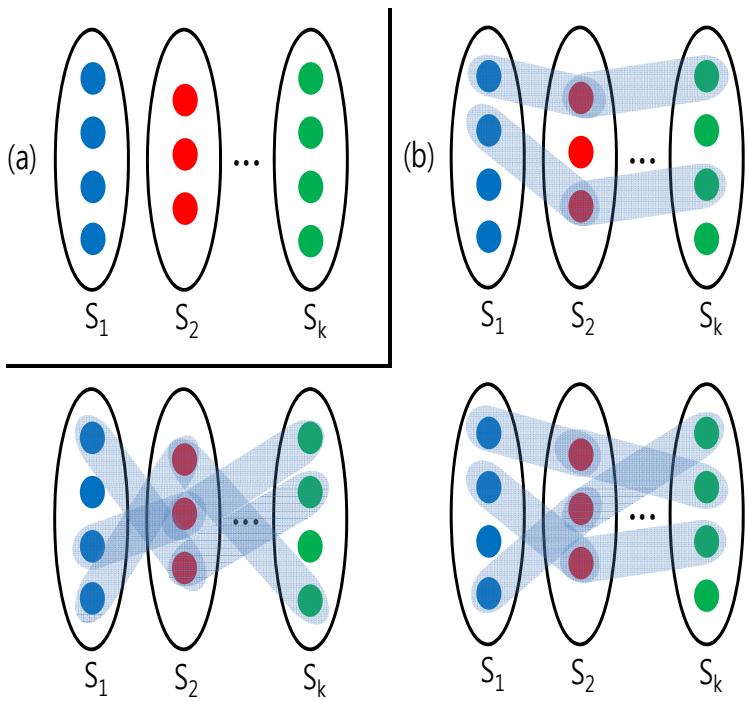

Figure 3.4: The examples of K-partite graph and K-partite matching. (a) The K-partite graph. The total sets of nodes is composed of K number of disjoint sets. (b) The examples of K-partite matching. A shaded cylindrical link from the set S_1 to the set S_K is one matching. The matchings have a single element from each disjoint set and any distinct two matchings have no common node from the same set. From the matching process, two people are associated together in the top right graph, and four people in the bottom left and in the bottom right graph.

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{i_2=1}^{n_2}\sum_{i_3=1}^{n_3}\cdots\sum_{i_K=1}^{n_K} x_{i_1,i_2,\cdots,i_K} = 1; \;\; i_1 = 1,2,...,n_1 \\
& \sum_{i_1=1}^{n_1}\sum_{i_3=1}^{n_3}\cdots\sum_{i_K=1}^{n_K} x_{i_1,i_2,\cdots,i_K} = 1; \;\; i_2 = 1,2,...,n_2 \\
& \qquad\qquad\qquad\qquad \vdots \\
& \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\cdots\sum_{i_{K-1}=1}^{n_{K-1}} x_{i_1,i_2,\cdots,i_K} = 1; \;\; i_K = 1,2,...,n_K,
\end{aligned}
$$

where $c_{i_1,i_2,\cdots,i_K}$ is the association cost and $x_{i_1,i_2,\cdots,i_K}$ is the binary variable to indicate if the corresponding observation is associated or not.

However, this formulation and matching graph assumes that the performance of detection is perfect and no missing detection happens. Moreover, it even assumes that all people in the scene are detected in every view of cameras. Within this formulation and graph, every K-partite matching should have exactly K number of nodes, where each of them is selected in each disjoint observation set from different cameras. This assumption is unrealistic because every people cannot be in all the views of every camera unless all of their field of view is exactly same. Moreover, the detection algorithm cannot be perfect with 100% precision and recall values. To relax this assumption and solve the data association problem in spatial domain practically, the objective function in the equation (3.1) can be modified as

$$
c = \min \sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_2}\cdots\sum_{i_K=0}^{n_K} c_{i_1,i_2,\cdots,i_K} x_{i_1,i_2,\cdots,i_K} \tag{3.2}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{i_2=0}^{n_2}\sum_{i_3=0}^{n_3}\cdots\sum_{i_K=0}^{n_K} x_{i_1,i_2,\cdots,i_K} = 1; \;\; i_1 = 1,2,...,n_1 \\
& \sum_{i_1=0}^{n_1}\sum_{i_3=0}^{n_3}\cdots\sum_{i_K=0}^{n_K} x_{i_1,i_2,\cdots,i_K} = 1; \;\; i_2 = 1,2,...,n_2 \\
& \qquad\qquad\qquad\qquad \vdots \\
& \sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_2}\cdots\sum_{i_{K-1}=0}^{n_{K-1}} x_{i_1,i_2,\cdots,i_K} = 1; \;\; i_K = 1,2,...,n_K,
\end{aligned}
$$

and the modified K-partite graph is illustrated in Figure 3.5. The difference between the formulations in the equation (3.1) and the equation (3.2) is that one extra node is added for each disjoint set. This node has the index of 0 in the equation (3.2) and is the blank node in each disjoint observation set in Figure 3.5 (a). This blank node is an imaginary node for handling the missing detection problem. Different to the previous definition of the K-partite matching, this blank node can be selected multiple times in different matchings as shown in Figure 3.5 (b). For example, if we see the bottom left matching graph in Figure 3.5, we can see that the blank node in the set S_2 is selected in the matchings for the first and the second objects. Also, this graph can describe the situation where the first object is located only in the last camera, the second object in the first and the last camera, and the third object in all camera views while the original K-partite graph and K-partite matching cannot describe this situation because they do not consider the missing detection case from the geometrical condition and the low performance of detection algorithms.

However, this modified conventional K-partite matching and the graph is not enough to solve the problem of multiple objects tracking with multiple number of cameras since this conventional K-partite matching and the graph only target to solve the spatial data association problem. To consider both the spatial and the temporal association, conventional approaches use one of the two concepts,

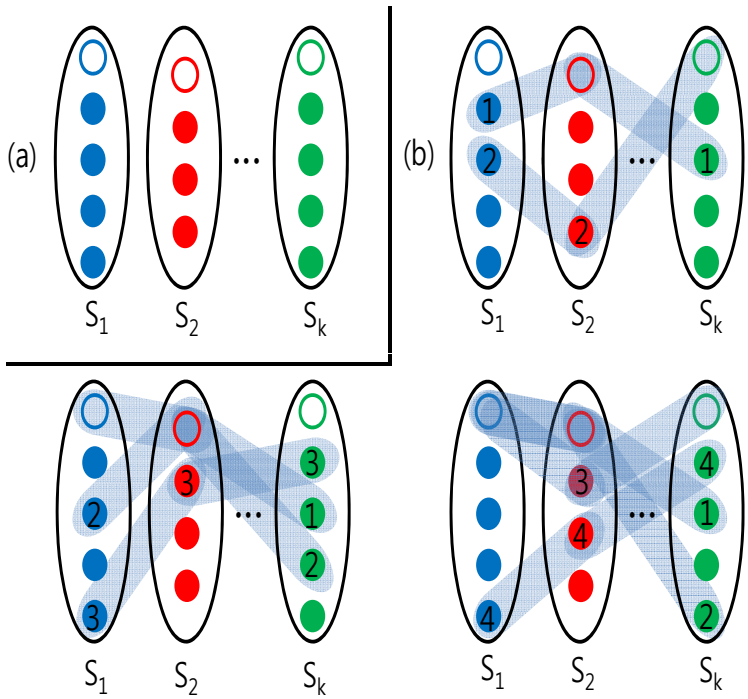

Figure 3.5: The examples of modified K-partite graph and K-partite matching. (a) The modified K-partite graph. The total sets of nodes is composed of K number of disjoint sets and each disjoint set has an extra node (blank node) for missed detection. (b) The examples of K-partite matching in the modified K-partite graph. A shaded cylindrical link from the set S_1 to the set S_K is one matching. The matchings have a single element from each disjoint set and any distinct two matchings has no common node from the same set. However, the added node can be selected in several matchings. The label of objects is indicated with numbers.

Reconstruction-Tracking or *Tracking-Reconstruction*. The difference between two concepts is illustrated in Figure 3.6. *Reconstruction-Tracking* firstly reconstructs 3D tracking model (yellow nodes) by spatial association process among 2D detections (green nodes), and then finds temporal association between these 3D tracking models ($3D^{t-1}$ and $3D^{t}$ in Figure 3.6). On the other hand, *Tracking-Reconstruction* finds temporal association (2D tracking) between detections at current frame and tracking models at the last previous frame in image domain, and then perform spatial association process among cameras. These two types of approaches, *Reconstruction-Tracking* and *Tracking-Reconstruction*, finds spatial and temporal association but separately, and this can make incorrect solution by considering tracking and reconstruction process independently. To overcome this limitation of the conventional approaches, we modify the conventional K-partite graph and matching scheme to associate data in both spatial and temporal domain simultaneously.

3.3.1.2 The proposed matching graph

Different to the conventional *Reconstruction-Tracking* and *Tracking-Reconstruction* approaches, we need to perform the data association process considering both spatial and temporal domain at once. For this purpose, we modify the conventional K-partite matching graph by adding the node set of the 3D tracking models as shown in Figure 3.7 and associate those tracking models with detection observations from each camera. In our K-partite graph $G^t = (U^t, H^{t-1}, H_0, E^t)$ as shown in Figure 3.7, there exist detection observations (U^t), 3D tracking models (H^{t-1}), and a null node for the tracking model initialization (H_0). For the probabilistic formulation, we define the random variables representing the components of the matching graph as follows. The observation random vector U^t is defined as $U^t = [U_1^t, U_2^t, \cdots, U_{N_c}^t]$, in which U_k^t is the vector for detec-

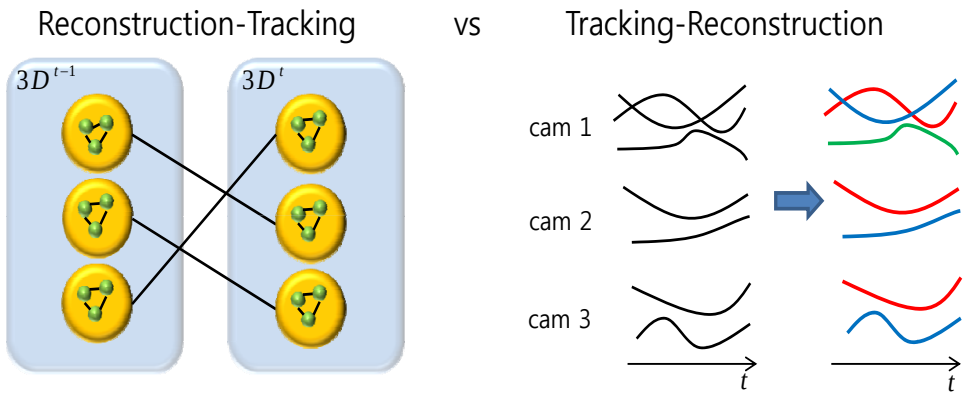

Figure 3.6: The difference between the two concepts of *Reconstruction-Tracking* or *Tracking-Reconstruction*. *Reconstruction-Tracking* firstly reconstructs 3D tracking model (yellow nodes) by spatial association process among 2D detections (green nodes), and then finds temporal association between these 3D tracking models ($3D^{t-1} and 3D^{t}$). On the other hand, *Tracking-Reconstruction* finds temporal association between detections at current frame and tracking models at the last previous frame in image domain, and then perform spatial association process.

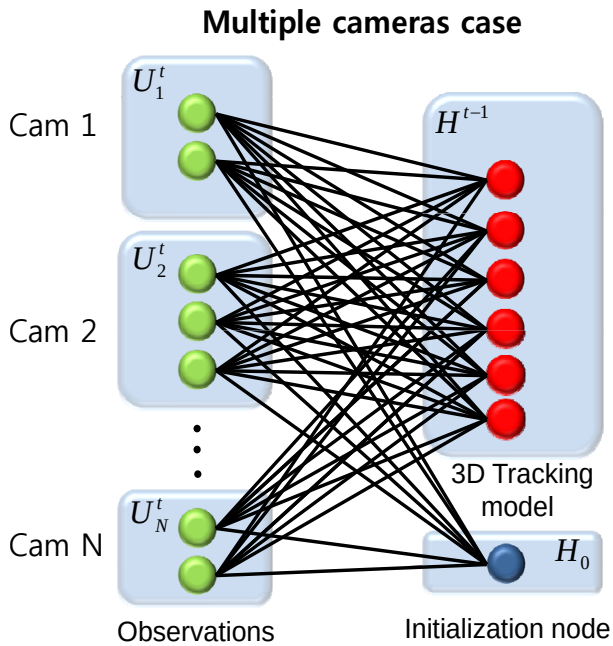

Figure 3.7: The K-partite matching graph for multiple camera case. Multiple objects tracking with multiple cameras is a data association problem between observations in current frame and 3D tracking models (each 3D tracking model represent an individual object).

tion observation on the kth camera. U_k^t is a random vector composed of N_k number of random variables as $U_k^t = [U_{k,1}^t, U_{k,2}^t, \cdots, U_{k,N_k}^t]$, that is, N_k is the number of detection observation in the kth camera. The 3D human tracking model H^{t-1} is defined as $H^{t-1} = [H_1^{t-1}, H_2^{t-1}, \cdots, H_{N_{H^{t-1}}}^{t-1}]$, in which H_i^{t-1} is the random variable of the ith 3D human tracking model. Each 3D tracking model (H_i^{t-1}) is composed of position in the three dimensional world coordinate system ($D_{3,i}^{t-1}$), and positions, velocities and appearances in every camera coordinate system is defined as ($D_{2,i,1:N_c}^{t-1}, V_{2,i,1:N_c}^{t-1}, A_{2,i,1:N_c}^{t-1}$) respectively. That is, $H_i^t = [D_{3,i}^t, D_{2,i,1:N_c}^t, V_{2,i,1:N_c}^t, A_{2,i,1:N_c}^t]$. D represents position information, V is for velocity, and A for appearance. Those nodes of detection observations and 3D tracking models are linked each other with edges (E), and no edges are connected between nodes in the same camera set and between nodes in the tracking model set.

To track multiple objects and maintain their labels over frames, we find matchings in this graph among human detection results of each camera at current frame ($U_{k,j}^t, 1 \leq k \leq N_c, 1 \leq j \leq N_k$), which is extracted by the method explained in the previous section, and 3D tracking models at the last previous frame ($H_i^{t-1}, 1 \leq i \leq N_{H^{t-1}}$) which represent individual objects being tracked until the last previous frame. Here, $N_{U_k^t}$ is the number of detection on the kth camera at time t, and $N_{H^{t-1}}$ is the number of 3D human tracking model at time $t-1$. In detail, matching between two nodes $U_{k,j}^t$ and H_i^{t-1} means that the jth detection from the kth camera at time t is matched to the ith 3D tracking model at time $t-1$. By assigning the label of the ith tracking model at time $t-1$ to the jth detection from the kth camera at time t, the temporal tracking process is accomplished, and the spatial tracking process is finished by assigning the same label to all detections which are connected to the same 3D tracking model. On the other hand, the matching between the detection node ($U_{k,j}^t$) and the added

imaginary node (H_0) is defined for different purpose. When the detection result $U^t_{k,j}$ is from an individual who appears newly in the scene, there is no current 3D tracking model which can explain this detection result. In this case, all of those nodes of newly appeared detections are connected to the imaginary node (H_0). This matching between $U^t_{k,j}$ and H_0 initializes a new tracking model with a new label for the detection $U^t_{k,j}$.

With this definition of matching and graph, we assume several conditions in matching solutions for the physically plausible tracking result. First, the nodes with similar features, such as 2D, 3D positions, velocities, etc., are likely to be matched. Because we compare the observed detection at time t and the 3D tracking models at time $t-1$, we assume the object, which is a human in our case, does not move or change significantly and they look similar between short time steps. This is reflected in the likelihood probability, which will be described in the next section, MAP formulation. Second, all of the nodes of the detection results are in matching. This means all detection nodes are matched to either 3D tracking model or the imaginary node for tracking model initialization. Because the detection at current time t should be either the object in the last previous frame or the newly appeared object in current frame, it is adequate to enforce the entire detection node to be in one of the matchings. Third, two detections from different camera can be connected to one 3D tracking model. This comes from the fact that the actual object which corresponds to a single 3D tracking model can appear and be detected differently in each camera. However, the link which connects two detections from the same camera to a single 3D tracking model should be avoided because one single target cannot be detected twice in the same camera, which is physically implausible. Fourth, any number of detection can be linked to the imaginary node even the detection nodes from the same camera. The first-appeared objects cannot be described by previous tracking models and all

those detections should be initialized via the imaginary node for the next frame.

In our matching solution, detection nodes which are connected to the same 3D tracking model are regarded as spatially associated detections and the matching between one detection node and one 3D tracking model is temporally associated. The spatially and temporally associated nodes share the same label and this achieves the multiple target tracking. This spatial and temporal association on the matching graph will be found by a sampling method on the proposed online framework to solve a MAP problem described in the next section.

3.3.2 MAP formulation

With the matching graph and several matching conditions described in the previous section, we formulate a MAP problem to find matchings in the graph and to solve the multiple targets tracking problem. Among various possible matching configurations, we find a matching configuration composed of several number of matchings which maximizes the defined posterior probability. However, this posterior probability distribution is very complicated with its large number of possible combinatorial solutions. It is difficult to know the shape of this posterior probability exactly, which leads to the fact that it is hard to find a solution which maximizes this posterior probability distribution. For this reason, the problem of multiple target tracking with multiple cameras is well known as a NP-hard problem. To solve this NP-hard problem fast in online manner, we propose an online framework to track multiple objects by solving our formulated MAP problem in the following section. Our online framework to solve multiple target tracking problem with multiple number of cameras is based on the similar framework with a single camera in the previous chapter.

3.3.2.1 Online MAP formulation for the multiple cameras case

To build an online framework for multiple target tracking, the posterior probability should be formulated in an online form, which means it has to deal with only the detection at current frame and the state model at the last previous frame. In this perspective, we define our online posterior probability for tracking multiple objects as

$$P(H^t|U^t, \hat{H}^{t-1}), \tag{3.3}$$

where $\hat{H}^{t-1}$ is defined as

$$\hat{H}^{t-1} = \underset{H^{t-1}}{\operatorname{argmax}} \; P(H^{t-1}|U^{t-1}, \hat{H}^{t-2}). \tag{3.4}$$

Different to the recursive Bayesian estimation, the state model at the current frame (H^t) is calculated given the current observation from multiple number of cameras ($U^t = [U_1^t, \cdots, U_{N_c}^t]$) and the last previous state model ($\hat{H}^{t-1}$)in the equation (3.3). The reason why we use the hat mark on the state model at the time $t-1$ is that we want to distinguish the usage of the state model at time $t-1$ (H^{t-1}) between the recursive Bayesian estimation method and the proposed online method. While we assume that the estimated state at the last previous frame can explain the observations until the last previous frame well and regard it as the fixed variable like observations for the posterior probability, the recursive Bayesian estimation method regards it as the state model which is an unobserved variable required to be estimated. For this reason, we use the last state model as observation ($\hat{H}^{t-1}$) instead of the state model to be estimated (H^{t-1}). In the equation (3.4), we remove the observations from the first frame to the last frame in the equation (2.11) and substitute them with the state at the last previous frame ($\hat{H}^{t-1}$). The resulting graphical model of our online framework for estimation of the posterior probability in the equation (3.3) is illustrated in Figure 3.8.

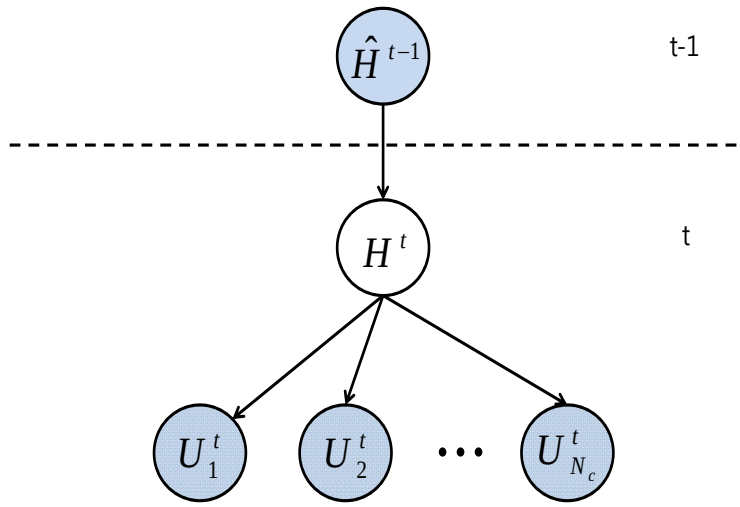

Figure 3.8: The graphical model of the proposed online framework for estimation of the posterior probability. The state at the current frame is estimated given the solution at the last previous frame and current obsetvations from multiple number of cameras.

The directed graph in Figure 3.8 describes the process of estimation of the 3D tracking model at time t (H^t) given the 3D tracking model at time $t-1$ (H^{t-1}) and the detection observation at time t (U^t). In this perspective, this directed graph and the defined posterior probability in the equation (3.3) can describe well our online matching problem on the graph in Figure 3.8. Similar to the matching process between the 3D tracking model at time $t-1$ (H^{t-1}) and the detection observation at time t (U^t) and the building process of the 3D tracking model at time t (H^t) with their matching results on the graph in Figure 3.8, estimation process in this directed graph and posterior probability deals with the 3D tracking model at time $t-1$ (H^{t-1}) and the detection observation at time t (U^t) to find 3D tracking model at time t (H^t).

With this online framework, the MAP problem for multiple targets tracking

can be defined as

$$
\begin{aligned}
P(H^t|U^t,\hat{H}^{t-1}) &= \frac{P(\hat{H}^{t-1},H^t,U^t)}{P(U^t,\hat{H}^{t-1})} && (3.5)\\
&= \frac{P(\hat{H}^{t-1})P(H^t|\hat{H}^{t-1})P(U_1^t|H^t)\cdots P(U_{N_c}^t|H^t)}{P(U^t,\hat{H}^{t-1})} && (3.6)\\
&= \frac{P(\hat{H}^{t-1})P(U^t|H^t)P(H^t|\hat{H}^{t-1})}{P(U^t,\hat{H}^{t-1})} && (3.7)\\
&\propto P(U^t|H^t)P(H^t|\hat{H}^{t-1}), && (3.8)
\end{aligned}
$$

where $H^t = [H_1^t, \cdots, H_{N_{H^t}}^t]$ and $H^{t-1} = [H_1^{t-1}, \cdots, H_{N_{H^{t-1}}}^{t-1}]$, and N_c is the number of camera in the system. Here, N_{H^t} is the number of 3D tracking models at the tth time step, and $N_{H^{t-1}}$ is the number of 3D tracking models at the $t-1$th time step. In the equation (3.8), the first term $P(U^t|H^t)$ represents a likelihood probability and the second term $P(H^t|\hat{H}^{t-1})$ represents a prior probability. Without calculation of the normalization term of the equation (3.7), finding a solution which maximizes the product of the likelihood probability and the prior probability can guarantee the solution of the original posterior probability in the equation (3.3). In following sections, the likelihood probability and the prior probability are defined with several features.

• The likelihood probability

For the calculation of likelihood probability of this MAP formulation, we define four types of likelihood probability, which are 2D likelihood probability, 3D assignment likelihood probability, camera overlap likelihood probability, and separation likelihood probability. Each likelihood is calculated based on different features or current matching status. The features we used for the likelihood probability are 2D position, velocity, appearance, 3D position, and geometry of the scene which is closely related to the view of camera. On the other hand, the likelihood probabil-

ity considering the current matching status is related to the physically plausible matching conditions we described in the previous section. The total likelihood probability is the production of the four likelihood probabilities, 2D likelihood probability (P_{2D}), 3D assignment likelihood probability (P_{3D}), camera overlap likelihood probability (P_{CO}), and separation likelihood probability (P_S), as

$$\begin{aligned} P(U^t|H^t) &= P_{2D}(U^t|H^t) \cdot P_{3D}(U^t|H^t) \\ &\quad \cdot P_{CO}(U^t|H^t) \cdot P_S(U^t|H^t). \end{aligned} \tag{3.9}$$

In the equation (3.9), 2D likelihood probability deals with information in each camera domain, and 3D assignment likelihood considers the 3D positions of the detections which have the same label by association process. The camera overlap likelihood is based on the geometric information of the scene such as the internal matrices and the positions of cameras, and the separation likelihood gives low probability for implausible solution, such as the matchings of two detections in the same camera and the single tracking model. Assuming these informations are independent to each other, we multiply the likelihoods on the informations to define the total likelihood probability. The detailed description of each likelihood probability is described in the following sections.

2D likelihood probability

In the equation (3.9) of the total likelihood probability, the 2D likelihood probability is defined as

$$P_{2D}(U^t|H^t) = \prod_{k,j} P_{2D}(U^t_{k,j} = u^t_{k,j}|H^t_{m(u^t_{k,j})} = h^t_{m(u^t_{k,j})}), \tag{3.10}$$

where k is the camera index, j is the object index in a single camera view. In this sense, $u^t_{k,j}$ is the jth detection observation in the kth camera at time t.

$m(u^t_{k,j})$ is the index of the matched human model of the object $u^t_{k,j}$ and $h^t_{m(u^t_{k,j})}$ is the matched 3D tracking model, where $h^t_{m(u^t_{k,j})} = (d^t_{3,i}, d^t_{2,i,1:N_c}, v^t_{2,i,1:N_c}, a^t_{2,i,1:N_c})$. This 2D likelihood probability is calculated with all of the detection observations. This probability is related to the 2D information of the observed detection results in each camera and that of 3D human tracking models. In detail, we use the informations from each single camera, which are $D^t_{2,m(u^t_{k,j}),k}$ and $A^t_{2,m(u^t_{k,j}),k}$ of the tracking model $h^t_{m(u^t_{k,j})}$ as the equation (2.20). The features we use in this likelihood probability are their positions and appearances in image domain of the each camera view as below,

$$\begin{aligned} & P_{2D}(U^t_{k,j} = u^t_{k,j} | H^t_{m(u^t_{k,j})} = h^t_{m(u^t_{k,j})}) \\ = \;& \alpha_{m(u^t_{k,j})} \cdot exp(-\|Pos(u^t_{k,j}) - d^t_{2,m(u^t_{k,j}),k}\|^2_2) \\ & + (1 - \alpha_{m(u^t_{k,j})}) \cdot exp(-\sum_{q=1}^{Q} (IM_q(u^t_{k,j}) - a^t_{2,m(u^t_{k,j}),k,q})^2), \end{aligned} \tag{3.11}$$

where Pos, IM_q and α is defined in chapter 2. $a^t_{2,m(u^t_i),k,q}$ is the qth pixel of appearance model of the $m(u^t_{k,j})$th tracking model in the kth camera, $a^t_{2,m(u^t_{k,j}),k}$.

The above 2D likelihood probability can be well defined if the index of $m(u^t_{k,j})$ indicates one of the indices of the 3D human models h^t, which means $1 < m(u^t_{k,j}) < N_{H^t}$. However, if $u^t_{k,j}$ is a detection result from a newly appeared human at the tth frame, as it is explained in the previous section, this detection node is connected to the tracking model initialization node (H_0). Since the tracking model initialization node (H_0) is an imaginary node, it does not have position and velocity information, and it exists independently in the time domain. For this reason, the 2D likelihood probability between the detection node and the imaginary node (H_0) cannot be calculated with the definition in the equation (3.11). To handle this case, the likelihood probability for the extra added node (H_0) is

defined as

$$P_{2D}(U^t_{k,j} = u^t_{k,j} | H^t_{m(u^t_{k,j})} = H_0) \tag{3.12}$$

$$= \begin{cases} \delta & \text{if } \max\limits_k P_{2D}(U^t_k = u^t_{k,j} | H^t_{m(u^t_{k,j})} = h^t_{m(u^t_{k,j})}) \leq T, \\ 0 & \text{otherwise} \end{cases}, \tag{3.13}$$

where T is a threshold to check whether the detection $u^t_{k,j}$ has a similar 3D tracking model in the 2D likelihood probability (P_{2D}) sense. If the detection $u^t_{k,j}$ has no similar 3D tracking model, then the value of $\max\limits_k P_{2D}(U^t_k = u^t_{k,j} | H^t_{m(u^t_{k,j})} = h^t_{m(u^t_{k,j})})$ will be less than the predefined threshold value of T. In this case, we assign a constant value of δ to the probability to initiate a new 3D tracking model for that detection. Otherwise, we assign 0 to the probability and give no chance to a new 3D tracking model to be initiated. The values of δ and T we used in this thesis are 0.2 and 0.5 respectively.

3D assignment likelihood probability

In the equation (3.9) of the total likelihood probability, we formulate the 3D assignment likelihood probability by considering the 3D positions of detection nodes which are matched to the same human tracking model. Because the assignment of different detection nodes to the same human tracking model means that those detection nodes are actually from the same object in each different camera, their positions should be close enough in the three dimensional world coordinate system. To check whether they are close, we assume that the probability follows the exponential distribution with respect to the distance among detections. In this perspective, we transform the position of the detection result in image coordinate system of the each camera to the world coordinate system together and compute the mean of the Euclidean distances from the mean position to the each trans-

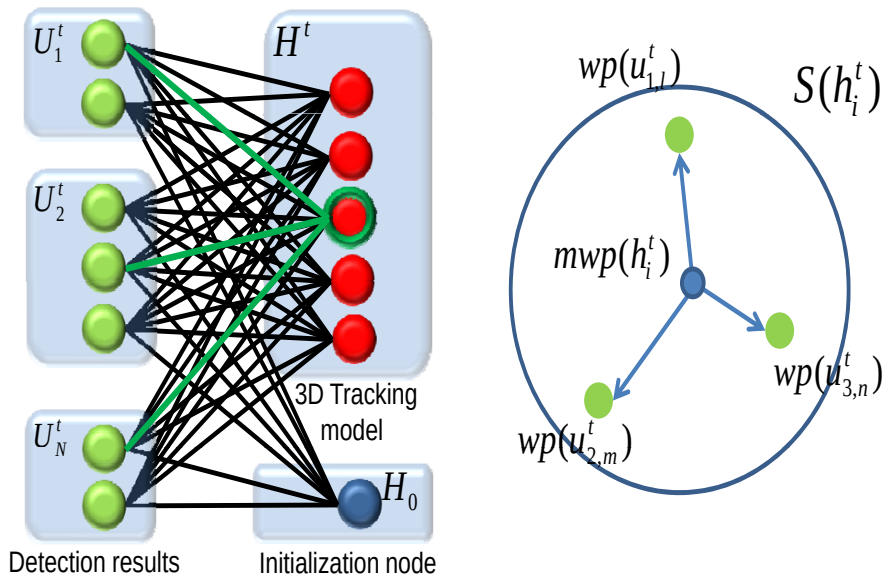

Figure 3.9: The illustration of the definitions in 3D assignment likelihood. The set of detections connected to the the same human model h_i^t (linked with green edges) are defined as $S(h_i^t)$. After transformation of positions of detection nodes with camera calibration matrix, the mean of those 3D positions in world coordinate system is calculated as $mwp(h_i^t)$.

formed points. The definition of the 3D assignment likelihood is illustrated in Figure 3.9. If we think the case that there are three detection nodes $u^t_{1,l}, u^t_{2,m}$ and $u^t_{3,n}$ which are connected together to the same 3D tracking model (h^t_i). Define $wp(g)$ be a function of transforming the position vector g in the image coordinate system into the world coordinate system with the pre-calculated camera calibration matrix. The set $S(h^t_i)$ is the set of positions of the detection observations which are connected to the tracking model h^t_i in the three dimensional world coordinate system and $S_l(h^t_i)$ is the lth element of the set $S(h^t_i)$. Then, the mean point among the 3D points transformed from the positions of detections at the cameras, $mwp(h^t_i)$, is defined as

$$mwp(h^t_i) = \frac{1}{N_{S(h^t_i)}} \sum_{l=1}^{N_{S(h^t_i)}} wp(S_l(h^t_i)), \tag{3.14}$$

where $N_{S(h^t_i)}$ is the number of element in the set $S(h^t_i)$. The matching assignment which produces the small distance from each of them to the mean point $mwp(h^t_i)$ is more preferred in our formulation than the matchings with large gap among transformed points. Finally, the 3D assignment likelihood probability is defined as

$$\begin{aligned} P_{3D}(U^t|H^t) &= \prod_i^{N_{Ht}} P_{3D}(U^t_{m^{-1}(h^t_i)} = u^t_{m^{-1}(h^t_i)}|H^t_i = h^t_i) \\ &= \prod_i^{N_{Ht}} exp(-\max_l(|mwp(h^t_i) - wp(S_l(h^t_i)|))), \end{aligned} \tag{3.15}$$

where $m^{-1}(h^t_i)$ is the index of detection observation which is connected to the tracking model h^t_i.

This 3D assignment likelihood is well known as the reconstruction error of the assignment and widely used to solve the multiple targets tracking problem in

several conventional approaches. However, using this likelihood probability of the reconstruction error only can cause severely incorrect data association because assigning different labels to all of the detection results can maximize this likelihood probability. For this reason, several different types of likelihood probability should be applied together or a constant probability should be defined for the case of a single detection result for a human tracking model.

camera overlap likelihood probability

In the equation (3.9) of the total likelihood probability, the camera overlap likelihood probability is adopted to prevent the insufficient spatial association among the detections of different cameras. The insufficient spatial association is the case that the detection observations, which are actually from a single object, are not associated together. The example of this insufficient spatial association is illustrated in Figure 3.10. In Figure 3.10, there exist four number of cameras, where the black box means images from each camera and the shaded regions represent the overlapped area among these four cameras. In the images from the multiple cameras, the circles ($(u^t_{1,l}, u^t_{2,m}$, $u^t_{3,n}$, and $u^t_{4,o})$) are the observed detections of the same object. Because the detection at the first camera is located within the overlapped area (shaded region), the corresponding object should be detected in other cameras and all of the four detection observations should be spatially associated as $S^*(h^t_i)$. However, if we see $S(h^t_i)$, there exists only one observation $u^t_{1,l}$ in $S(h^t_i)$, and insufficient spatial association happens in this case. The insufficient association occurs a lot in the sampling process by initializing all of the detection nodes as new tracking models. This initialization from insufficient association is mainly caused by the 3D assignment likelihood probability in the equation (3.15) because the tracking model connected to the single detection observation brings the 3D assignment likelihood of 1. To give lower likelihood when insufficient as-

signment happens, we formulate the camera overlap likelihood probability.

To prevent insufficient spatial assignment, we need to know the set $S^*(h_i^t)$ from the set $S(h_i^t)$ first. The set $S^*(h_i^t)$ can be obtained by using all of the elements in the set of $S(h_i^t)$ and camera calibration matrices of all camera views. After knowing the set $S^*(h_i^t)$, we penalize the current association $S(h_i^t)$ by comparing the number of $S(h_i^t)$ and that of the set $S^*(h_i^t)$. That is, we formulate the camera overlap likelihood probability to give lower likelihood when the numbers of above two sets are different. In detail, the camera overlap likelihood is defined as

$$\begin{aligned} P_{CO}(U^t|H^t) &= \prod_i^{N_{H^t}} P_{CO}(U^t_{m^{-1}(h_i^t)} = u^t_{m^{-1}(h_i^t)}|H_i^t = h_i^t) \\ &= \prod_i^{N_{H^t}} exp(-|N_{S(h_i^t)} - N_{S^*(h_i^t)}|), \end{aligned} \tag{3.16}$$

where $N_{S(h_i^t)}$ is the number of elements in the set $S(h_i^t)$ and $N_{S^*(h_i^t)}$ is the number of elements in the set $S^*(h_i^t)$. When a certain object is located within overlapped area and the number of spatially associated set of that object is not enough which is not a good solution, these two numbers of $N_{S(h_i^t)}$ and $N_{S^*(h_i^t)}$ are different. In this respect, those numbers should be similar to prevent the insufficient matching assignment.

separation likelihood probability

In the equation (3.9) of the total likelihood probability, the separation likelihood probability is defined to prevent physically implausible matching configuration, for example, it is implausible that two different detections in one camera is connected to the single tracking model. The illustration example of this separation likelihood is shown in Figure 3.11. In Figure 3.11, two detection nodes (the node

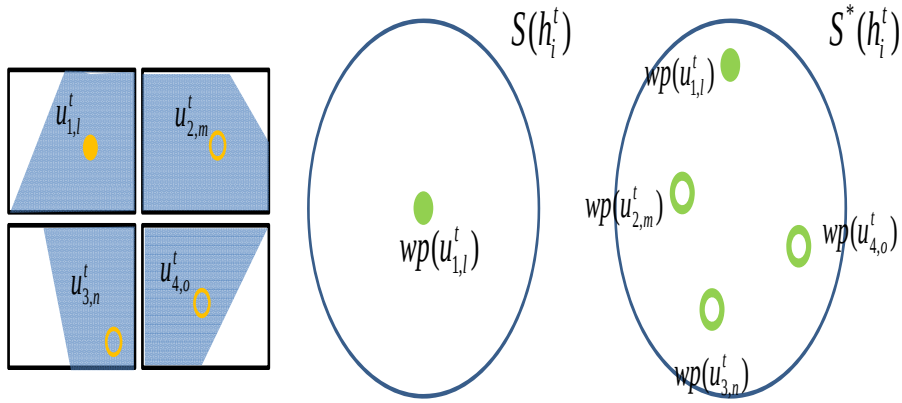

Figure 3.10: The illustration example for the camera overlap likelihood. The black box means images from each camera, and the shaded regions are overlapped area. The circles ($(u^t_{1,l}, u^t_{2,m}$, $u^t_{3,n}$, and $u^t_{4,o})$) in each camera view are the observed detections of the same person. Because the detection at the first camera ($u^t_{1,l}$) is located within the overlapped area (shaded region), it should also be detected in other cameras and the association should be done to include all of the four elements of detections as the set $S^*(h^t_i)$ rather than $S(h^t_i)$.

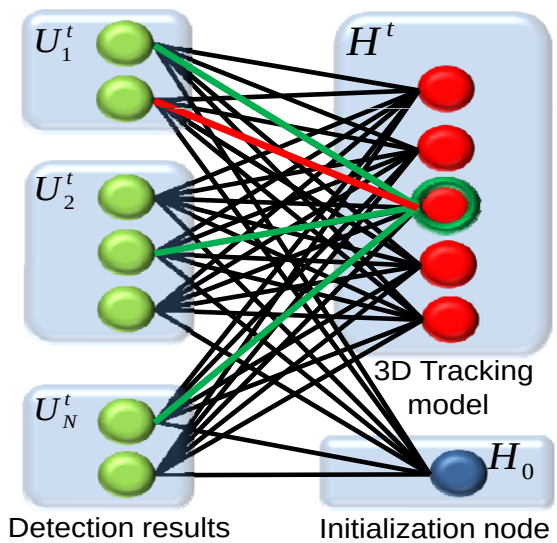

Figure 3.11: The illustration example for the separation likelihood probability. When two detection nodes from the same camera (the node on green edge and the node on red edge in the first camera) are in the $S(h_i^t)$) at the same time, it means one object is detected separately in the same camera. Because this association is not physically plausible, we assign low likelihood probability for this type of the matching solution.

on green edge and the node on red edge in the first camera) are connected to the same 3D tracking model. However, this association means one object is detected twice in the same camera, which is physically impossible. On the other hand, multiple detections from different cameras (the nodes on the green edges) can be associated together. To prevent the physically impossible solution, we formulate the separation likelihood probability as

$$\begin{aligned} P_S(U^t|H^t) &= \prod_i^{N_{H^t}} P_S(U^t_{m^{-1}(h_i^t)} = u^t_{m^{-1}(h_i^t)}|H_i^t = h_i^t) \\ &= \prod_i^{N_{H^t}} exp(-N_{sc}(S(h_i^t))), \end{aligned} \tag{3.17}$$

where N_{sc} is a function that counts the number of nodes in the set which have the same camera index. For each solution, we count the number of detection observations in the same camera connected to the same 3D tracking model. As the number of this type of the matching increases, we assign lower likelihood probability.

• The prior probability

The prior probability from the equation (3.8) is dealing with the motion information of the tracking models without considering the detection observations. With the assumption of the independence of each 3D tracking model, we calculate the prior probability considering each tracking model independently. The illustration of the prior probability is shown in Figure 3.12. As shown in Figure 3.12, the 3D positions of tracking models with the same label at time $t-1$ and t are compared and their distances are measured for the prior probability. On the other hand, the fourth 3D tracking model $\hat{h}_4^{t-1}$ is not connected to any 3D tracking model at current t frame because it is a disappeared human. Also, the seventh 3D tracking

model h_7^t is not connected to any 3D tracking model at the $t-1$ frame because it is a newly appeared human.

With the assumption of the indepdence of each 3D tracking model, the prior probability in the equation (3.8) is formulated as

$$P(H^t|\hat{H}^{t-1}) = \prod_{i}^{N_p(H^t,\hat{H}^{t-1})} P(H_i^t = h_i^t|\hat{H}_i^{t-1} = \hat{h}_i^{t-1}), \tag{3.18}$$

where N_p the number of pairs of H^t and $\hat{H}^{t-1}$. Because this prior probability is defined with respect to the possible motion dynamics of the objects, we do not calculate the prior probability for newly appeared objects (H^t is existed but no corresponding $\hat{H}^{t-1}$) and disappeared objects ($\hat{H}^{t-1}$ is existed but no corresponding H^t). Then, the prior probability of each 3D tracking model, $P(H_i^t|\hat{H}_i^{t-1})$, is defined as

$$\begin{aligned} & P(H_i^t = h_i^t|\hat{H}_i^{t-1} = \hat{h}_i^{t-1}) \\ = \;& P(D_{3,i}^t = d_{3,i}^t|\hat{D}_{3,i}^{t-1} = \hat{d}_{3,i}^{t-1}) \\ & \cdot \prod_{k}^{N_c} P(D_{2,i,k}^t = d_{2,i,k}^t|\hat{D}_{2,i,k}^{t-1} = \hat{d}_{2,i,k}^{t-1}, \hat{V}_{2,i,k}^{t-1} = \hat{v}_{2,i,k}^{t-1}), \end{aligned} \tag{3.19}$$

where $h_i^t = (d_{3,i}^t, d_{2,i,1:N_c}^t, v_{2,i,1:N_c}^t, a_{2,i,1:N_c}^t)$ (the ith 3D tracking model at time t) and $\hat{h}_i^{t-1} = (\hat{d}_{3,i}^{t-1}, \hat{d}_{2,i,1:N_c}^{t-1}, \hat{v}_{2,i,1:N_c}^{t-1}, \hat{a}_{2,i,1:N_c}^{t-1})$ (the ith 3D tracking model at time $t-1$). Each term in the equation (3.19) is defined as the exponential distribution, i.e.,

$$P(D_{3,i}^t = d_{3,i}^t|\hat{D}_{3,i}^{t-1} = \hat{d}_{3,i}^{t-1}) = exp(-|d_{3,i}^t - \hat{d}_{3,i}^{t-1}|), \tag{3.20}$$

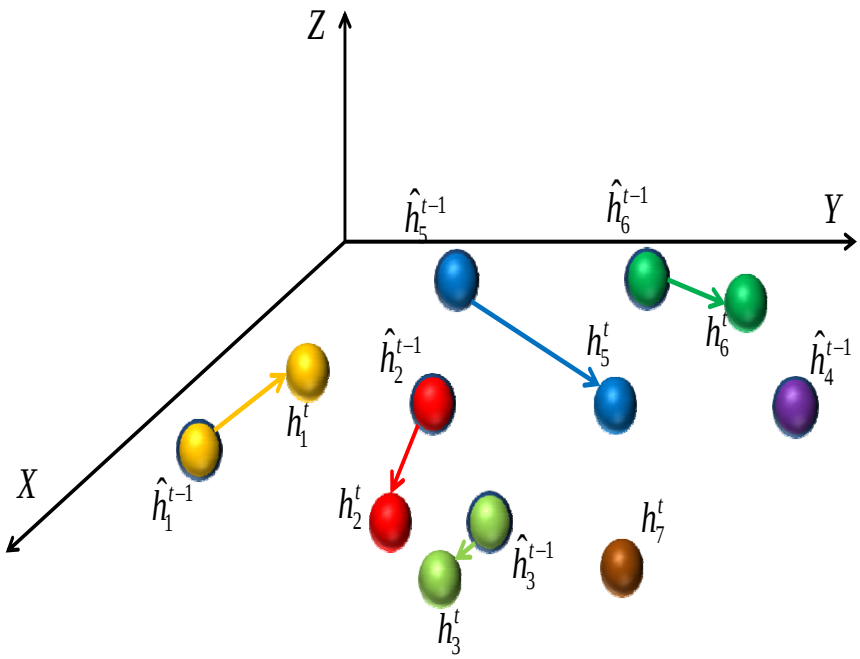

Figure 3.12: The illustration example for the prior probability. For the human model which corresponds to h_i^t and $\hat{h}_i^{t-1}$, the prior probability is calculated according to the exponential distribution on the random variable of Euclidean distance between h_i^t and $\hat{h}_i^{t-1}$. Because $\hat{h}_4^{t-1}$ is a disappeared object and h_7^{t-1} is a newly appeared object at the current frame, we do not calculate the prior probability for these two human models.

and

$$P(D^t_{2,i,k} = d^t_{2,i,k} | \hat{D}^{t-1}_{2,i,k} = \hat{d}^{t-1}_{2,i,k}, \hat{V}^{t-1}_{2,i,k} = \hat{v}^{t-1}_{2,i,k}) \\ = \quad exp(-|d^t_{2,i,k} - (\hat{d}^{t-1}_{2,i,k} + \hat{v}^{t-1}_{2,i,k})|). \tag{3.21}$$

From equations (3.20) and (3.21), the closely located models h^t_i and $\hat{h}^{t-1}$ gets high probability by exponentially defined prior with respect to the 3D distance. In detail, we assume a human cannot move large distance at a single time step and the data association result which associates two human models at a long distance gets less prior value. For the distance in the world domain, assuming that the data association result at time t is known, the 3D position of the ith 3D tracking model can be calculated with averaging the 3D positions of associated detection results by camera calibration matrix.

On the other hand, in the equation (3.19), we use the velocity information of the tracking model. Considering the velocity in the association can handle the occlusion case when two objects are approaching to each other from distance. When those two objects are located closely, the position without motion information is not enough to distinguish them and even changes their labels. However, by exploiting the velocity information of the tracking models, the labels of two objects approaching each other are well maintained after the occlusion. This is illustrated in Figure 3.13. As we can see in the last column of Figure 3.13 (a) and (b), using velocity information can prevent the label switch of the objects approaching each other from different directions.

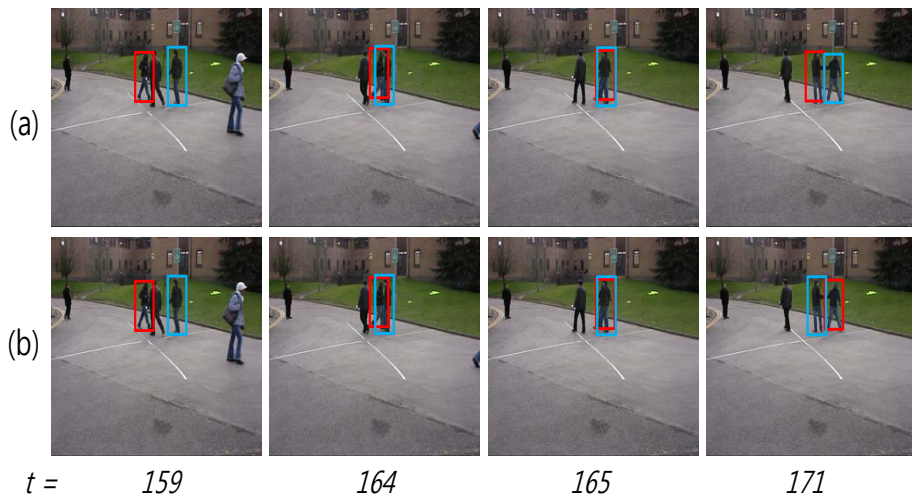

Figure 3.13: The illustration of the effect of using the velocity information. (a) the tracking results without considering the velocity information in the image domain. (b) the tracking results of considering the velocity information in image domain. By using the velocity information, we can solve the occlusion case when two objects are approaching from different directions.

3.3.2.2 Solving the MAP Problem

With the posterior probability of the product of the defined likelihood probability and the prior probability, the Gibbs sampling method [49] is adopted to get a MAP solution and to find the matchings between nodes. Because the solution space of our data association problem to find H^t is reduced by using only U^t and $\hat{H}^{t-1}$, the iterative method using the Gibbs sampling does not require a large number of iterations to solve the matching problem. On the other hand, the conventional global optimization methods require large number of detections through all the frames. As a result, the computational time increases significantly with the number of detections and frames.

3.4 Tracking model update processes

After the data association process with the Gibbs sampling method, the tracking models are updated according to the resultant types of matchings as illustrated in Figure 3.14: (a) Update of 2D tracking model matched to a detection node. (b) Update of the tracking model initialization node H_0. Because the newly appeared detection $u_{i,j}^t$ is not explainable by currently existing 3D human models, new tracking model is initialized with the detection $u_{i,j}^t$. (c) Update of 2D tracking models unmatched to any detection nodes. In some cameras, the object may not be detected or not visible by occlusion and in this case, the corresponding 2D models can not be matched to any detection nodes. (d) Update of 3D human models unmatched to any detection nodes. This case happens when some objects disappear. In Figure 3.14, the green nodes are detection observations, the red nodes are 3D human models, the yellow nodes are 2D tracking models within 3D human models, and the blue node is the tracking model initialization

node. In (a) and (b), the red line represents accomplished matching results and related nodes are the encircled nodes in blue color. Those matchings are used to update and initialize tracking models respectively. In the cases of (c) and (d), the encircled nodes in blue color are not matched with any detection nodes and they are required to be updated with different cues. Here, the black edges are also matchings, but the cases of (c) and (d) concern only the blue encircled 2D tracking models and 3D human models.

In Figure 3.14 (a), the detection-tracking model matchings are shown in red lines. When the detection node is matched to the 3D tracking model, it is actually matched to a 2D tracking model within the 3D human tracking model. In this case (a), the 2D tracking models (yellow nodes) within the 3D tracking model, which are matched to the detection nodes (green nodes), are updated with the information of the detection results such as position and size. The velocities of the tracking model are also computed in each camera domain with the information of detections. After every tracking models are updated, the corresponding 3D tracking model is also modified with the updated 2D tracking models. In (b), the detection-initialization node matchings are illustrated in red lines. Because the detection node is actually from the object appeared newly in the scene at the current frame, there exist no appropriate tracking model which can explain the detection observation. In this case, the detection node should initiate a new tracking model with a new label for tracking purpose. Even though two different detection nodes from the same camera cannot be matched to a single human model, any number of detection nodes can be linked to the same H_0 node for initializing multiple number of different tracking models with different labels.

The Figure 3.14 (c) is the case for the 3D tracking models which have unmatched 2D tracking models to any detection node. The blue encircled node in the graph of Figure 3.14 (c) is the undetected 2D tracking model within the

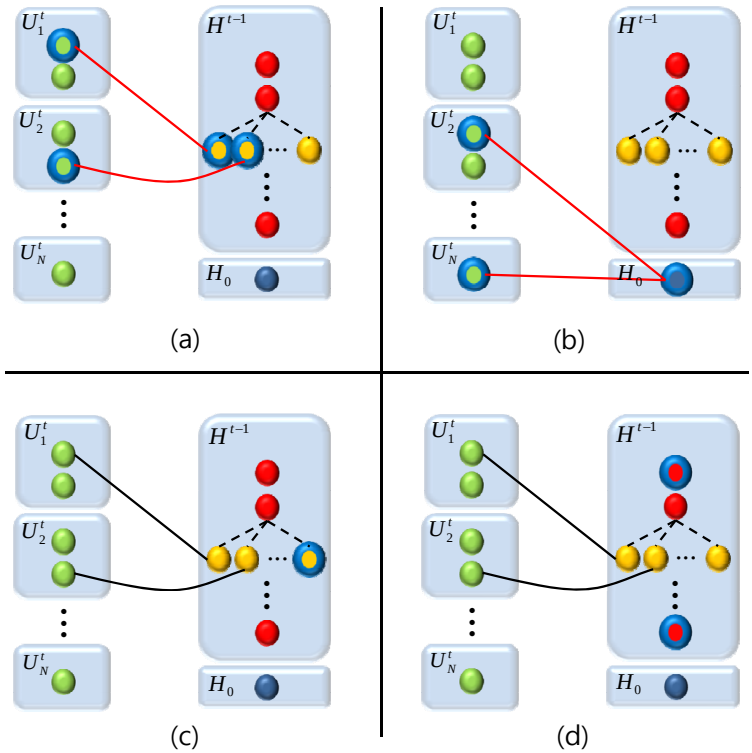

Figure 3.14: The update types according to the matching results. Green, red, and yellow nodes indicate detection, 3D tracking model, and 2D tracking model, respectively.(a) Update of 2D tracking model matched to a detection node. (b) Update of the tracking model initialization node H_0. Because the newly appeared detection $u_{i,j}^t$ is not explainable by currently existing 3D human models, new tracking model is initialized with the detection $u_{i,j}^t$. (c) Update of 2D tracking models unmatched to any detection nodes. In some cameras, the object may not be detected or not visible by occlusion and in this case, the corresponding 2D models can not be matched to any detection nodes. (d) SUpdate of 3D human models unmatched to any detection nodes. This case happens when some objects disappear.

3D tracking model. Because we have multiple number of cameras, several objects can be detected in some cameras but not in other cameras. If we see the graph of Figure 3.14 (c), the second 3D human tracking model (red node) is composed of several number of 2D tracking models, but only two 2D tracking models are matched with the detection nodes in the first and the second camera. This matching means the second human is detected in the first and the second camera, so the first and the second 2D tracking model can be updated with the matched detection nodes. However, the other 2D tracking models of the second 3D human tracking model are not matched with detection nodes. This missing detection problem can happen by occlusion or that the position of the object is not yet visible in that camera from geometrical reason. In former case, the position of this tracking model should be updated with different cues. By exploiting multiple number of cameras, the position of unmatched 2D tracking model can be estimated by other matched 2D tracking models within the same 3D tracking model. This estimation is achieved by projection of 2D tracking models from other cameras and reprojection to the camera where unmatched 2D tracking model is located. However, when 2D tracking model in certain camera is not initialized, we regard the object is not yet visible in that area of the corresponding camera and nothing is done for the 2D tracking model. In the graph of Figure 3.14 (d), two blue encircled 3D tracking models are not matched to any detections in multiple cameras views, so all of the 2D tracking models of those 3D tracking models should be updated either. Because those 3D tracking models are not matched to any detection node, there is no reliable information of detections as the case of (c). For this reason, to find the appropriate positions and velocities of those blue encircled tracking models in Figure 3.14 (d), we use a selective update method which was described in the single camera case.

3.5 Computational complexity analysis

In this section, we analyze the computational complexity of the proposed algorithm to track multiple targets with multiple number of cameras. This analysis is for the case at time t, in which there exist C number of cameras, D_i number of detection observation on the ith camera, and $N_{H^{t-1}}$ number of human models. Before the sampling process, we find the initial configuration of the 3D tracking model H^{t-1} by assigning the labels to the each detection on a certain camera independently to the other cameras. In this perspective, the computational complexity of calculation of the initial state is

$$O(D_1 \times N_{H^{t-1}} + \cdots + D_C \times N_{H^{t-1}}) = O((D_1 + \cdots + D_C) \times N_{H^{t-1}}). \quad (3.22)$$

In each step of the sampling process, we change the matching configuration of one detection observation probabilistically by calculating the posterior probability of the possible move of the matching configuration. When this move of the matching configuration is done for all of the detection observations of multiple number of cameras, one iteration of the sampling process is done. If we want to perform this sampling process until R number of iteration, the computational complexity of the total sampling process is

$$O(\sum_{r=1}^{R} \sum_{k=1}^{D_1+\cdots+D_C} N_{H^t,k,r}), \quad (3.23)$$

where $N_{H^t,k,r}$ is the number of 3D human tracking model at the kth step of sampling of the rth iteration. $N_{H^t,k,r}$ keeps changing with the result of the sampling process, which is the step for the kth detection out of total number of detections at the rth iteration. To validate this analysis experimentally, we test PETS 2009 dataset with the measures of Multiple Object Tracking Precision (MOTP) and

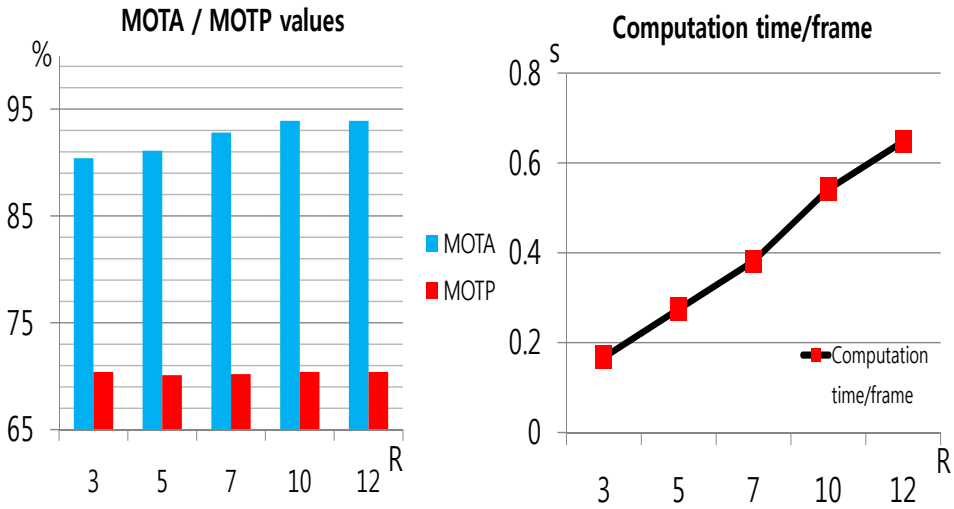

Figure 3.15: The computational complexity analysis for PETS 2009 dataset with different iteration number. The left graph shows that the MOTA values increase with large number of iterations and the right graph shows the linear increase of the computation time with the iteration number.

Multiple Object Tracking Accuracy (MOTA) which will be described detail in the next section with different number of iterations in the sampling process. This result is illustrated in Figure 3.15. In Figure 3.15, the left graph shows the tracking performance of MOTA and MOTP values with different iteration number R. As we iterate the sampling process more, the values of MOTA increases until $R = 10$. After that, the posterior probability is not increased with more number of iteration and the measure values are maintained. On the other hand, MOTP values are not changing much with different iteration numbers. The right graph of Figure 3.15 shows the change of the computation time of the sampling process with respect to the iteration number. As the complexity is defined in the equation (3.23), the computational complexity of the sampling process is proportionally increased with the iteration number, however, the term of $N_{H^t,r}$ is changing in each iteration. With this varying number of 3D human model at each iteration, the computational complexity is not perfectly linearly increased as shown in the right graph of Figure 3.15.

3.6 Experimental results

To evaluate our proposed algorithm and compare it to the conventional state-of-the-art algorithms, we tested PETS 2009 dataset [54] and APIDIS basketball dataset [71, 72], and we made ETRI dataset for solving the multiple target tracking problem with multiple number of camera in the case of indoor scene (a small room). There are two video sequences in this ETRI dataset. The detail description of each dataset is described in each section. The quantitative results were evaluated by CLEAR MOT metrics, Multiple Object Tracking Precision (MOTP), Multiple Object Tracking Accuracy (MOTA), the detection precision, and the detection recall from the paper [55]. The definitions of MOTP, MOTA are in

the equation (2.30) and the equation (2.29). Moreover, we additionally used the other metrics to measure the tracking performance that were presented in [73], including identity switches (IDS), mostly tracked (MT), mostly lost (ML), and partly tracked (PT). The measure of IDS is the total number of times that a tracked trajectory changes its identity with respect to the labels of the ground truth. For this reason, the smaller value of IDS is more preferred. MT is the percentage of the tracking successes which mean that the tracker should track the ground truth trajectories for more than the predefined threshold in length. In conventional algorithms and our proposed method, the threshold was set to 80%. ML is the percentage of the tracking failures which mean that the tracker could not track the ground truth trajectories for less than the predefined threshold in length. The threshold was set to 20%. Finally, PT is the percentage of partially tracked objects, which can be calculated by $1 - MT - ML$. The different techniques used to calculate the above measures for each data set is explained in each section below.

3.6.1 PETS 2009 dataset

The publicly available PETS 2009 dataset [54] is composed of three scenarios with different levels of difficulty. The easiest one is the set (S2.L1) with low density of people in the scene, and the sets of (S2.L2) and (S2.L3) contains much more number of people with higher densities. We tested our algorithm on the set of (S2.L1) and compared our algorithm with several state-of-the-art algorithms including online and global optimization algorithms. In this dataset, there exist eight camera sequences from eight cameras which have overlapping areas. These eight camera views of PETS 2009 (S2.L1) set are illustrated in Figure 3.16. As we can see in Figure 3.16, several people are walking across or along, and small number of occlusions happen. The motions of people are quite linear and easy

Figure 3.16: PETS 2009 dataset (S2.L1). The PETS 2009 dataset is composed of eight number of cameras which have overlapping areas, and the set of (S2.L1) in PETS 2009, which we used for the experiment, has low density number of pedestrians.

to estimate. However, the frame rate of the video sequences of PETS 2009 is only 7 frames per seconds, and it makes people move fast/jump in far distance between consecutive frames and it makes the tracking problem in this dataset challenging.

To evaluate the performance of conventional tracking algorithms and the proposed one, we used the ground truth provided in [74]. This ground truth is composed of actual 3D positions of multiple people in the world coordinate system. We reconstructed this 3D positions of multiple people with four number of cameras, whose numbers are 5,6,7, and 8. The quantitative and qualitative results are shown in Table 3.1 and in Figure 3.17 respectively. As we can see in Table 3.1, our online algorithm shows better performance in every measures than the previous works [39, 29], and comparable results to the paper [30]. All of these conventional algorithms are based on global optimization method, which

Sequence	Method	Camera IDs	MOTA[%]	MOTP[%]	MT[%]	PT[%]	ML[%]	IDS
PETS S2.L1	Berclaz et al. [39]	1+3+5+6+8	82	56	-	-	-	-
	Leal-Taixe et al. [29]	1+5	76.0	60	-	-	-	-
	Leal-Taixe et al. [29]	1+5+6	71.4	53.4	-	-	-	-
	Hofmann et al. [30]	1+5	99.4	82.9	100.0	0.0	0.0	1
	Hofmann et al. [30]	1+5+7	99.4	83.0	100.0	0.0	0.0	2
	Ours	5+6+7+8	93.9	70.4	100.0	0.0	0.0	3
	Ours	1+5+7+8	93.4	70.0	100.0	0.0	0.0	5

Table 3.1: The quantitative results for the PETS 2009 dataset (S2.L1).

Figure 3.17: The qualitative result of PETS 2009 dataset. The labels of multiple objects are temporally and spatially associated successfully. Each row is for the same frame index, and each column represents each camera view.

uses all of the detection observations of the video sequence. As described in the introduction, this type of methods is not suitable for online application, which is the most important thing for the surveillance purpose, and they are preferred in, rather, video analysis. Moreover, they could not provide the results fast because of the large computation space by the characteristic of the combinatorial problem. For the fair comparison, we tested our algorithm with different combinations of cameras (camera index (5,6,7,8) and camera index (1,5,7,8)). In Figure 3.17, we can see that the labels of multiple objects are successfully maintained over frames and among different cameras. Each column shows the results in each individual camera, and each raw represents the images from different cameras at the same frame index. Because we use the limitied tracking area as the paper [74] not the whole image region, the labels of people are disappeared when they exit this area and they get new labels when those people re-enter the tracking region. In the third row of Figure 3.17, all people except the human with the label 1 get new labels with this limitation of the tracking area. Afterwards, the labels of people keep changing with exiting and re-entering the tracking zone as we can see in the next frames.

3.6.2 APIDIS basketball dataset

The publicly available APIDIS basketball dataset [71, 72] is composed of 7 cameras. Five cameras are located on ground, and two fish-eye cameras are installed on ceiling and looking from above. The frame speed of this dataset is 25 fps. As conventional tracking algorithms, the proposed algorithm for multiple objects tracking with multiple cameras is only processed within the left half court to ensure the fair comparison the tracking performance. This APIDIS dataset is shown in Figure 3.18.

This dataset contains various challengeable things to be solved for tracking

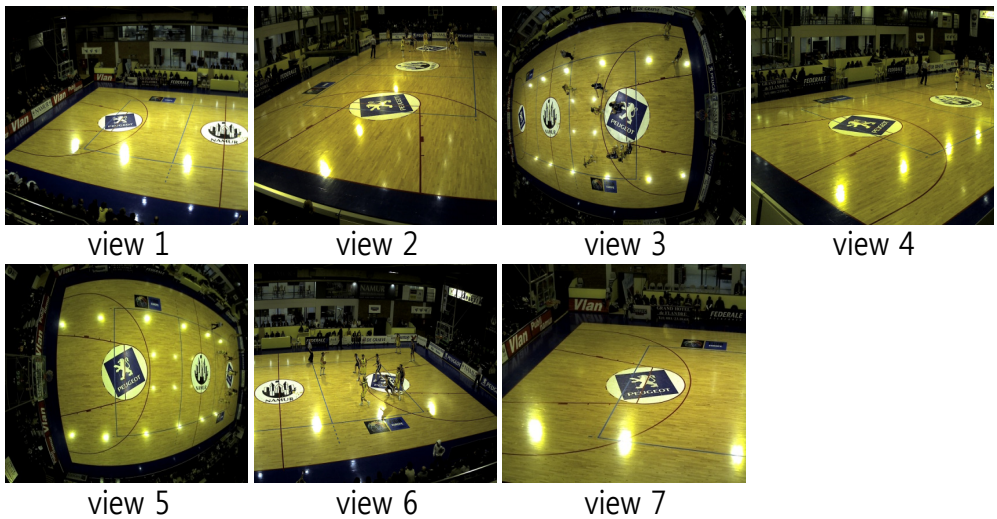

Figure 3.18: APIDIS basketball dataset. APIDIS basketball dataset is composed of seven number of cameras which have overlapping areas. Five of these cameras are installed on ground and two fish-eye cameras are on ceiling.

purpose. First, because this dataset is tracking basketball players and referees in a basketball game, the occlusions between players and referees happen often and severely. Different from the dataset which is taken under controlled scenario, the players in this dataset actually play the basketball, defending the others and blocking the opponent team players, and this makes severe occlusion among people. Secondly, the motions of players are so dynamic and they show nonlinear and abrupt motion. They even jump and violate the assumption that people are walking on the ground plane. With this respect, simple motion dynamic models of conventional algorithms cannot explain this type of motion. Thirdly, all players of the same team have very similar appearance except the height by wearing the same uniforms. In conventional multiple targets tracking algorithms, the two main features widely used to distinguish multiple number of objects are motion information and appearance cue. However, in this dataset, the motion informa-

tion is not reliable as explained before, and the appearance cue is also not a good feature. Lastly, several cameras share almost the same view point and using those cameras only increases the solution space without providing additional information from multiple number of cameras. The main reason in using multiple number of cameras is exploiting the abundant number of views and handling the occlusion by information from different views. However, in this APIDIS basketball dataset, the effect of adding more cameras is not large compared to the increase of solution space, which leads to low performance of tracking by the difficult association process. With these hard conditions of the dataset, almost all of the conventional algorithms show bad performance in this dataset.

Among the camera views of the APIDIS basketball dataset shown in Figure 3.18, the views 1,2,4, and 7 are used to track people and localize them in 3D world coordinate. The quantitative and qualitative results are shown in Table 3.2 and in Figure 3.19 respectively. In Table 3.2, KSP represents the algorithm using K-shortest path optimization in [75], and POM represents a method using the probabilistic occupancy map for multiple target tracking [40]. Different to the case of PETS 2009 dataset, the conventional algorithms for this dataset is measured with the number of TP (true positives), FP (false positives), FN (false negatives), and IDS (label switch). In detail, the true positive is the case of assigning the same label to the same objects as the ground truth, and the false positive counts the number of the case that more labels are assigned to the objects than the ground truth. Finally, the false negative is the case of the missed human. Compared to the conventional algorithms, our proposed algorithm shows better MOTA, MOTP, TP, FN values. For fair comparison with conventional approaches, we tested every 10th frames of data association results as [34], and showed the result with all frames. As shown in Table 3.2, the values of TP, FP, and FN increase proportionally when we testing more number of frames. However,

the value of IDS does not increase significantly with more number of frames. The difference between these two cases are from the characteristic of measures that TP, FP, FN are calculated in every frame independently while IDS is measured by comparing the label between two frames. When we sampled every 10th frame and performed data association, the false positives and missings which happened between two sampled frames was not counted. On the other hand, when the label of the objects was changed between two sampled frames, this label switch was counted. For this reason, when we tested every frame, our solution showed that the value of IDS is not proportionally increased with number of frames but gave better MOTA value. In the qualitative results of Figure 3.18, we can see that the spatial and temporal association is successfully done and people have the same labels over frames. This label may change when false positive or false negative happens.

3.6.3 ETRI dataset

In the ETRI dataset, there are two video sequences in which one is a video sequence with three people and the other with six people. For the video sequence with six people, *ETRI-S1*, the qualitative result is illustrated in Figure 3.20 respectively, and, for the video sequence with three people, *ETRI-S2*, the qualitative result is illustrated in Figure 3.21. This dataset is taken in a small room, so the camera calibration matrix can be more precisely computed without errors than the previous two datasets. However, in this dataset, there exist many false positive detections by human-shaped backgrounds, such as chairs, robots, etc., and these falsely detected objects become burdens to accomplish the good performance of the multiple targets tracking algorithms. Different from the false positives in other dataset, these false positives of detection algorithm are repeatedly detected in the same position on the image, which makes it difficult to remove it with

Sequence	Method	Camera IDs	MOTA[%]	MOTP[%]	TP	FP	FN	IDS
APIDIS	KSP[75]/POM[40]	1+2+4+5+7	0.490	0.538	607	156	330	46
	Possegger et al. [34]	1+2+4+5+7	0.675	0.59	656	88	172	9
	Possegger et al. [34] w/o color	1+2+4+5+7	0.597	0.578	625	121	202	10
	Ours for every 10th frame	1+2+4+5+7	0.682	0.764	783	134	101	46
	Ours for every frame	1+2+4+5+7	0.725	0.770	7803	1348	1010	62

Table 3.2: The quantitative results for the APIDIS basketball dataset.

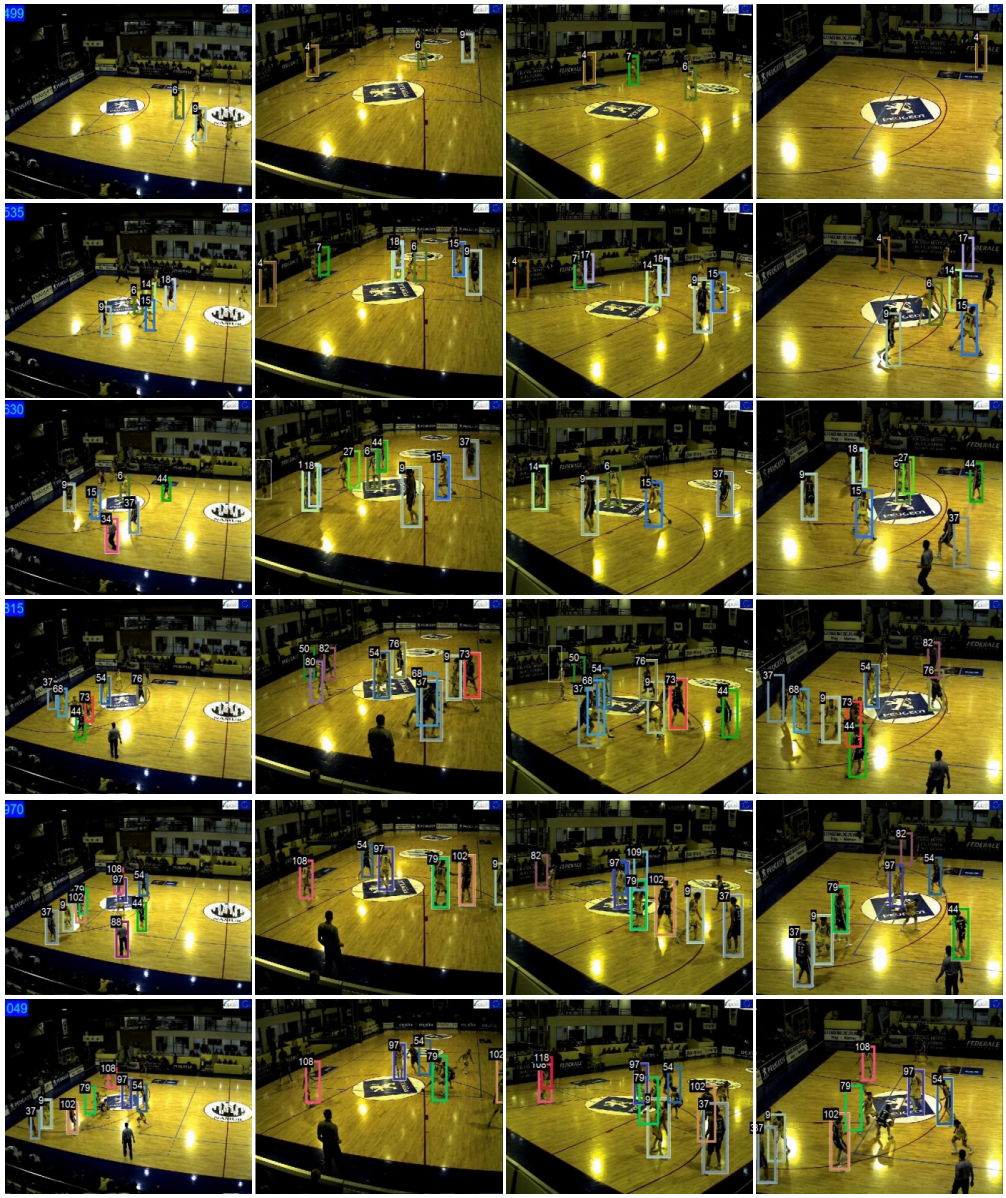

Figure 3.19: The qualitative result of APIDIS basketball dataset. The labels of multiple objects are temporally and spatially associated successfully. Each row is for the same frame index, and each column represents each camera view.

temporal frequency. However, as we can see in Figure 3.20, and Figure 3.21, our proposed algorithm maintains the labels of moving targets well over frames and within images from multiple number of cameras compared to the conventional algorithms.

3.7 Final remarks and discussion

In this chapter, we proposed an online data association for tracking multiple people using multiple cameras. We encoded the multiple people tracking problem to the matching problem on the matching graph and solved the spatial and temporal data association problem with the MAP formulation. We considered the 2D position in image coordinate, the 3D position in world coordinate, the velocity, and several other information to track objects in temporal domain and connect the same objects in the different cameras. The solution can be calculated fast with a sampling method because the solution space of our formulation is small by using only the observation at current frame and the 3D tracking models at the last previous frame. Moreover, we can solve the missing detection problem and the tracking model drift problem by selectively update scheme for the tracking model with local information. Our quantitative and qualitative evaluations showed that our method could track multiple people and maintains their identity successfully comparable to the state-of-art algorithms.

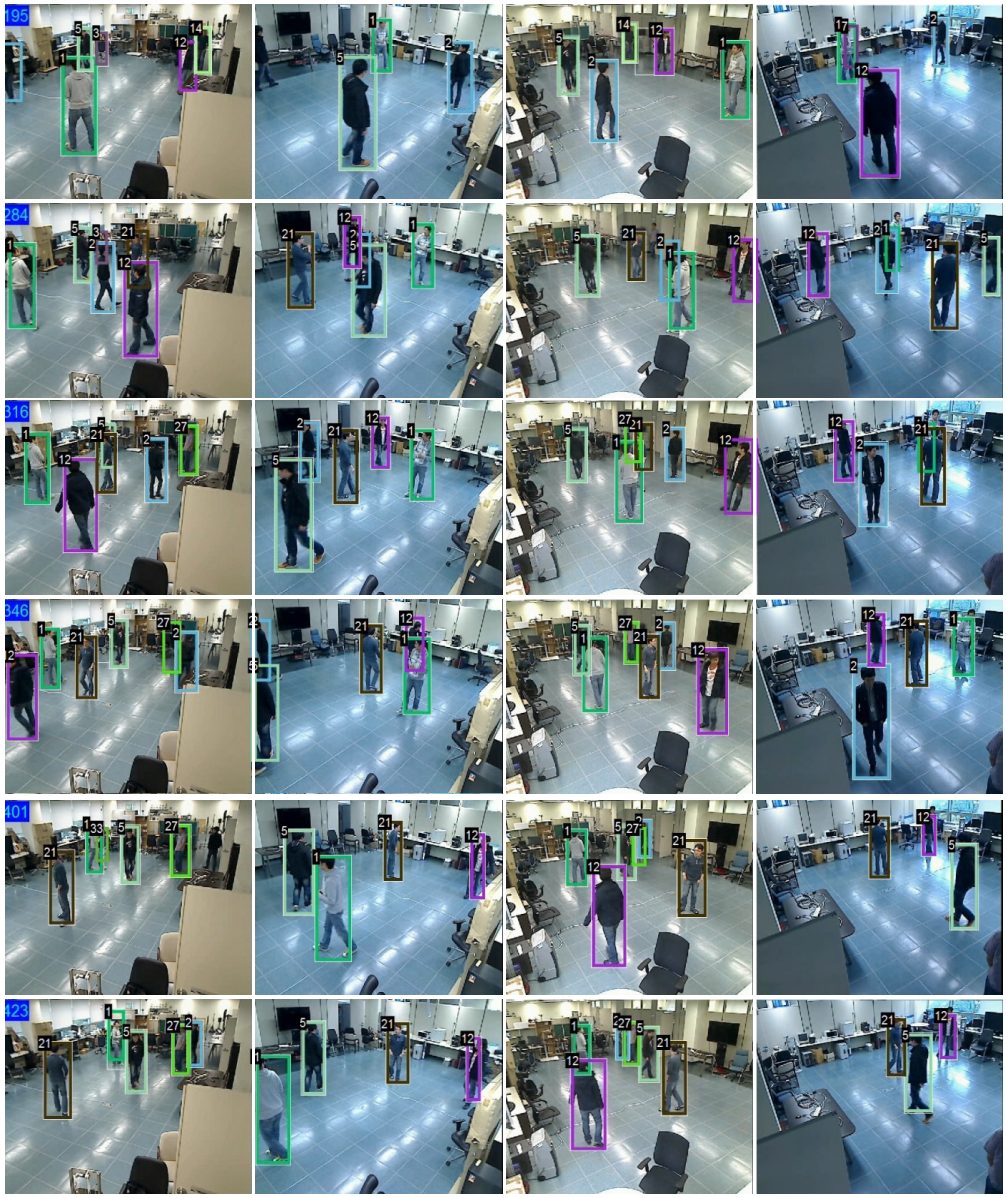

Figure 3.20: The qualitative result of ETRI-S1 dataset. The labels of multiple objects are temporally and spatially associated successfully. Each row is for the same frame index, and each column represents each camera view.

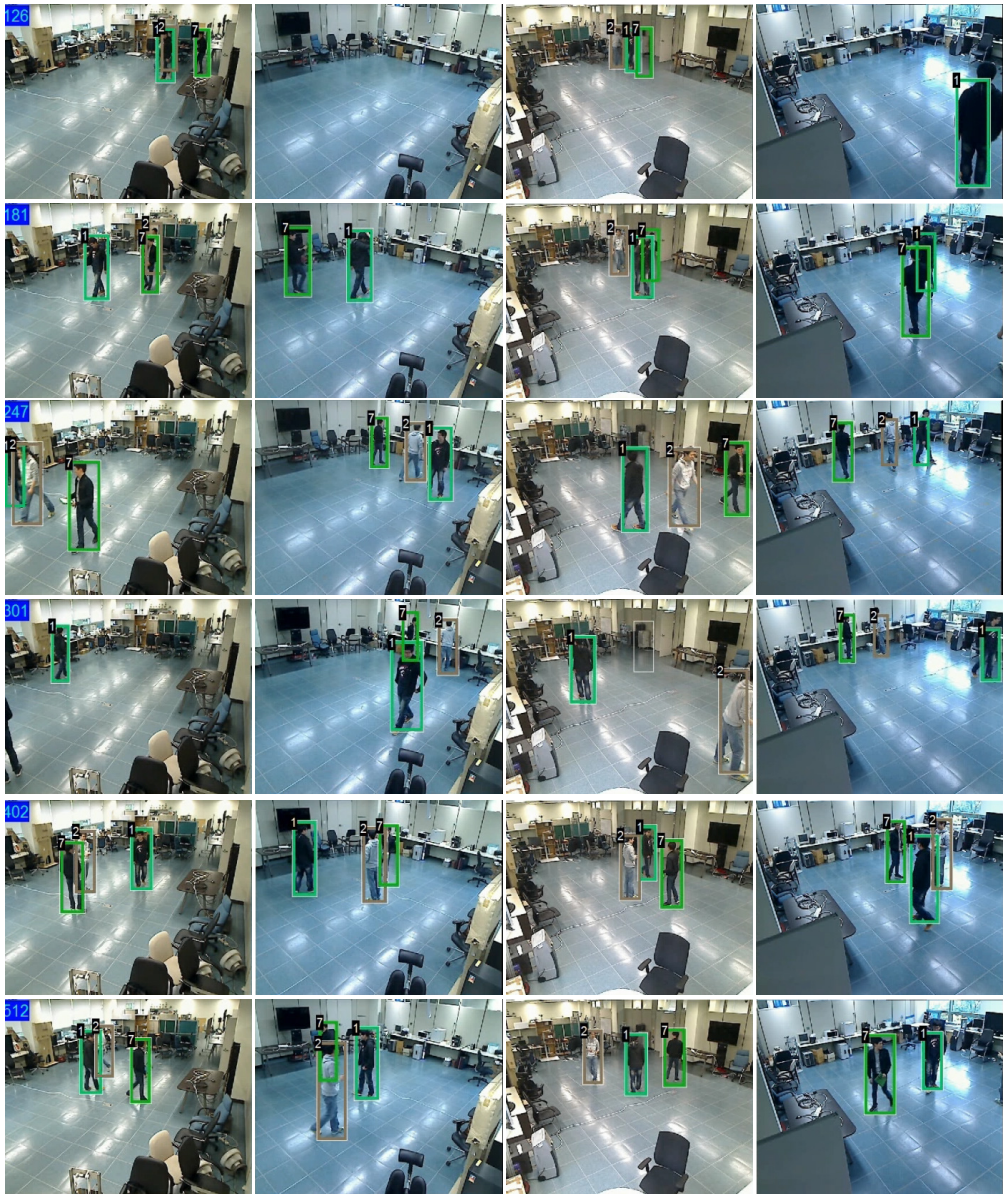

Figure 3.21: The qualitative result of ETRI-S2 dataset. The labels of multiple objects are temporally and spatially associated successfully. Each row is for the same frame index, and each column represents each camera view.

Chapter 4

Concluding Remarks

4.1 Conclusions

In this thesis, we proposed an online multiple objects tracking method with both multiple number of cameras and a single camera via a MAP optimization. In the case of the single camera, to track multiple objects in temporal domain, we formulate a MAP problem on the matching graph which is composed of detection observation and 2D tracking model in the same camera view. By constructing a matching graph and finding a matching configuration which maximizes the formulated MAP problem, we can track multiple objects in a single camera. Our MAP formulation considers only 2D information in image domain. The data association between the detection results at the current frame and the 2D tracking models from the last previous frame enables an online framework which runs faster than conventional methods which mostly use a global optimization framework. The missing detection problem, which is generally caused by occlusions or overlaps, is solved by our occlusion reasoning scheme and the selective update scheme. When missing detection happens, updating tracking models should be

done carefully because incorrectly estimated position of the tracking models can corrupt the tracking model and cause the drift. We selectively update the tracking model with its local neighborhood information and prevent this problem from being happened.

On the other hand, to track multiple objects in multiple cameras, we need to find the association in spatial as well as temporal domain. For this purpose, we build a matching graph and encode the tracking problem to the matching problem on the graph, which is composed of the detection results from each camera, 3D tracking models which represent each human in the world coordinate system, and extra added node for tracking model initialization. To find the matchings in our graph, we extended our formulation in the case of single camera and formulated a MAP problem on the matching graph. In our matching graph, the spatial and temporal association is achieved simultaneously by solving the formulated MAP problem with the Gibbs sampling method. Because we used only the detection observations at current frame and the 3D tracking models at the last frame, which we assume that they can describe all the previous association results, the solution space for the sampling is small and the solution can be calculated efficiently in online framework. To handle the occlusion and missing detection problems from poor performance of detectors in data association, we also used the proposed selective updating scheme as multiple camera case. Our quantitative and qualitative evaluations showed that our method tracked multiple people and maintained their identity successfully comparable to the state-of-art algorithms in both the single camera case and the multiple camera case.

4.2 Future Works

Even though the proposed method in this thesis shows reliable performance to the existing approaches, there are still many future works to improve the performance of the algorithm. Adopting online scheme for multiple objects tracking is the most important factor to be applied in online applications, however, the online optimization solution has limitation that the recovery from the degraded solution is very difficult. This is why we require the recovery method. One of the possible direction is the K-best matching scheme to enrich the solution space and find better solutions. However, selecting and storing multiple number of matching solutions should be done carefully. There can be different strategies in picking K number of solutions, such as picking the K matching solutions with the highest posterior probabilities or choosing K matching solutions which have significantly different matching configurations. Moreover, the way to visualize the tracking results from multiple number of hypotheses and how to measure the performance should be decided either. In the perspective of the design of the MAP formulation, the association likelihood can be improved by considering more complex information, such as volumetric shape of human and interactions among people. With these future researches, the multiple objects tracking method might improve the performance significantly and can be used in online applications.

Bibliography

[1] G. Welch and G. Bishop, *An introduction to the kalman filter.*, Chapel Hill, NC, USA, 1995.

[2] Shimin Yin, Jin Hee Na, Jin Young Choi, and Songhwai Oh, "Hierarchical kalman-particle filter with adaptation to motion changes for object tracking," *Computer Vision and Image Understanding*, vol. 115, no. 6, pp. 885–900, 2011.

[3] Jung Uk Cho, Seung-Hun Jin, Xuan Dai Pham, Jae Wook Jeon, Jong-Eun Byun, and Hoon Kang, "A real-time object tracking system using a particle filter," *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 2822–2827, Oct 2006.

[4] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 798–805, 2006.

[5] Xue Mei and Haibin Ling, "Robust visual tracking using l1 minimization," *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1436–1443, 2009.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[7] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1285–1292, 2010.

[8] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-n learning: Bootstrapping binary classifiers by structural constraints," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 49–56, 2010.

[9] B. Babenko, Ming-Hsuan Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[10] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," *Proceedings of the British Machine Vision Conference*, pp. 6.1–6.10, 2006.

[11] S. Avidan, "Ensemble tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 261–271, 2007.

[12] Chang Huang, Bo Wu, and Ramakant Nevatia, "Robust object tracking by hierarchical association of detection responses," *Proceedings of the 10th European Conference on Computer Vision - Volume Part II*, pp. 788–801, 2008.

[13] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," *ICCV*, pp. 1–8, 2007.

[14] Yuan Li, Chang Huang, and R. Nevatia, "Learning to associate: Hybrid-boosted multi-target tracker for crowded scene," *CVPR, IEEE Conference on*, pp. 2953–2960, 2009.

[15] Junliang Xing, Haizhou Ai, and Shihong Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses," *CVPR, IEEE Conference on*, pp. 1200–1207, 2009.

[16] Li Zhang, Yuan Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," *CVPR, IEEE Conference on*, pp. 1–8, 2008.

[17] Severin Stalder, Helmut Grabner, and Luc Van Gool, "Cascaded confidence filtering for improved tracking-by-detection," *Proceedings of the 11th European Conference on Computer Vision - Volume Part I*, pp. 369–382, 2010.

[18] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," *CVPR, IEEE Conference on*, pp. 3457–3464, 2011.

[19] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn, "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker," *Computer Vision Workshops (ICCV Workshops), IEEE International Conference on*, pp. 120–127, 2011.

[20] Stefano Pellegrini, Andreas Ess, and Luc Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," *Proceedings of the 11th European Conference on Computer Vision - Volume Part I*, pp. 452–465, 2010.

[21] Ben Benfold and Ian Reid, "Guiding visual surveillance by tracking human attention," *Proceedings of the British Machine Vision Conference*, 2009.

[22] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," *ICCV*, pp. 1515–1522, 2009.

[23] Bo Wu and Ram Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *Int. J. Comput. Vision*, vol. 75, no. 2, pp. 247–266, Nov. 2007.

[24] Irshad Ali and Matthew N. Dailey, "Multiple human tracking in high-density crowds," *Image and Vision Computing*, vol. 30, no. 12, pp. 966–977, 2012.

[25] Guang Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," *CVPR, IEEE Conference on*, pp. 1815–1821, 2012.

[26] Bo Yang and R. Nevatia, "An online learned crf model for multi-target tracking," *CVPR, IEEE Conference on*, pp. 2034–2041, 2012.

[27] Cheng-Hao Kuo, Chang Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," *CVPR, IEEE Conference on*, pp. 685–692, 2010.

[28] Siyu Tang, Mykhaylo Andriluka, and Bernt Schiele, "Detection and tracking of occluded people," *Proceedings of the British Machine Vision Conference*, 2012.

[29] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn, "Branch-and-price global optimization for multi-view multi-target tracking," *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1987–1994, 2012.

[30] M. Hofmann, D. Wolf, and G. Rigoll, "Hypergraphs for joint multi-view reconstruction and multi-object tracking," *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3650–3657, 2013.

[31] S. Sternig, T. Mauthner, A. Irschara, P.M. Roth, and H. Bischof, "Multi-camera multi-object tracking by robust hough-based homography projections," *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1689–1696, 2011.

[32] Anurag Mittal and Larry S. Davis, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo," *International Journal of Computer Vision*, pp. 189–203, 2002.

[33] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[34] H. Possegger, S. Sternig, T. Mauthner, P.M. Roth, and H. Bischof, "Robust real-time tracking of multiple objects by volumetric mass densities," *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2395–2402, 2013.

[35] S. M. Khan and Mubarak Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," *Proceedings of the 9th European Conference on Computer Vision*, vol. 3954, pp. 133–146, 2006.

[36] S.M. Khan, Pingkun Yan, and M. Shah, "A homographic framework for the fusion of multi-view silhouettes," *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, 2007.

[37] S.M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 3, pp. 505–519, March 2009.

[38] Kyungnam Kim and Larry S. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," *Proceedings of the 9th European Conference on Computer Vision - Volume Part III*, pp. 98–109, 2006.

[39] J. Berclaz, F. Fleuret, and P. Fua, "Multiple object tracking using flow linear programming," *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–8, 2009.

[40] 7. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 267–282, 2008.

[41] Ra8 Eshel and Yael Moses, "Tracking in a dense crowd using multiple cameras," *International Journal of Computer Vision*, vol. 88, no. 1, pp. 129–143, 2010.

[42] Nickolay I. Zheng Wu 9nd Hristov, Tyson L. Hedrick, T.H. Kunz, and M. Betke, "Tracking a large number of objects from multiple views," *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1546–1553, Sept 2009.

[43] A. Zheng Wu 10nd Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1948–1955, June 2012.

[44] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *CVPR, IEEE Conference on*, vol. 1, pp. 886–893 vol. 1, 2005.

[45] P.F. Felzenszwalb, R.B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2241–2248, 2010.

[46] R. Okada, "Discriminative generalized hough transform for object dectection," *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2000–2005, Sept 2009.

[47] D.B. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, 1979.

[48] Mart1jn Liem and DariuM. Gavrila, "Multi-person localization and track assignment in overlapping camera views," *Pattern Recognition*, vol. 6835, pp. 173–183, 2011.

[49] Stuart Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-6, no. 6, pp. 721–741, Nov 1984.

[50] Songhwai Oh, S. Russell, and S. Sastry, "Markov chain monte carlo data association for multi-target tracking," *Automatic Control, IEEE Transactions on*, vol. 54, no. 3, pp. 481–497, 2009.

[51] Victor Prisacariu and Ian Reid, "fasthog - a real-time gpu implementation of hog," Tech. Rep. 2310/09, Department of Engineering Science, Oxford University, 2009.

[52] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME Journal of Basic Engineering*, , no. 82 (Series D), pp. 35–45, 1960.

[53] James Ferryman, "Pets 2007 dataset: Performance and evaluation of tracking and surveillance," 2007.

[54] James Ferryman, "Pets 2009 dataset: Performance and evaluation of tracking and surveillance," 2009.

[55] Keni Bernardin and Rainer Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *J. Image Video Process.*, vol. 2008, pp. 1:1–1:10, jan 2008.

[56] Harold W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, 1955.

[57] Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah, "(mp)2t: multiple people multiple parts tracker," *Proceedings of the 12th European Conference on Computer Vision*, vol. 7577, pp. 100–114, 2012.

[58] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," *Proceedings of the 12th European Conference on Computer Vision*, pp. 343–356, 2012.

[59] Margrit Betke (Boston University) ZHENG WU (Boston University), Jianming Zhang, "Online motion agreement tracking," *Proceedings of the British Machine Vision Conference*, 2013.

[60] K. Yamaguchi, A.C. Berg, L.E. Ortiz, and T.L. Berg, "Who are you with and where are you going?," *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1345–1352, June 2011.

[61] Tao Zhao, R. Nevatia, and Bo Wu, "Segmentation and tracking of multiple humans in crowded environments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 7, pp. 1198–1211, 2008.

[62] Piotr Dollár, Ron Appel, and Wolf Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, pp. 645–659, 2012.

[63] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.

[64] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 4, pp. 743–761, April 2012.

[65] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009.

[66] P. Dollar, Zhuowen Tu, Hai Tao, and S. Belongie, "Feature mining for image classification," *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

[67] Zhe Lin and LarryS. Davis, "A pose-invariant descriptor for human detection and segmentation," *Proceedings of the 10th European Conference on Computer Vision*, vol. 5305, pp. 423–436, 2008.

[68] S. Maji, A.C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[69] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis, "Human detection using partial least squares analysis," *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 24–31, Sept 2009.

[70] Christian Wojek and Bernt Schiele, "A performance evaluation of single and multi-feature people detection," *Proceedings of the 30th DAGM Symposium on Pattern Recognition*, pp. 82–91, 2008.

[71] Fan Chen, D. Delannay, and C. De Vleeschouwer, "An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study," *Multimedia, IEEE Transactions on*, vol. 13, no. 6, pp. 1381–1394, Dec 2011.

[72] C. De Vleeschouwer, Fan Chen, D. Delannay, C. Parisot, C. Chaudy, E. Martrou, and A. Cavallaro, "Distributed video acquisition and annotation for sport-event summarization," *NEM Summit*, 2008.

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

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

[75] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 9, pp. 1806–1819, Sept 2011.

국문 초록

이상적인 영상감시 시스템은 영상 감시의 본연의 목적에 부합하기 위해서는 범죄나 사건에 대한 즉각적인 대응이 보장되어야 한다. 이러한 이유로 움직이는 물체를 탐지하거나 추적을 하는 등의 영상 감시 알고리즘은 온라인으로 동작하는 것이 더 선호 된다. 일반적으로 이러한 온라인 알고리즘들은 인과관계 (causality condition) 을 어길 수 없기 때문에 과거의 입력 데이터만을 사용하기 때문에 영상 전체를 사용하는 알고리즘들 (일괄 처리에 기반한 알고리즘) 에 비해 낮은 성능을 보일 수 밖에 없다. 하지만 일괄 처리에 기반한 알고리즘들은 연산 량과 연산 시간이 많기 때문에 영상 감시 시스템 에서는 여전히 온라인 알고리즘이 더 요구된다. 단일 물체에 대한 추적 알고리즘은 일반적으로 온라인으로 동작하는 반면에 대부분의 다중 물체 추적 방법은 그 어려움 때문에 일괄 처리방법을 사용하는 방향으로 개발되고 있다. 일괄 처리 기반의 알고리즘이 더 널리 이용되는 이유는 각각의 단일 물체를 추적하며 동시에 그들을 구별하는 데에 필요한 정보양이 단일 물체에 비해서 훨씬 많기 때문에 좋은 성능을 위해선 많은 양의 데이터가 필요하기 때문이다. 다수의 물체를 시간적으로 추적하는 데 있어 많은 양의 데이터를 동시에 고려해야 하는 어려움을 해결하기 위해 일반적으로 데이터 연관 기법이 많이 사용된다.

본 논문에서는 먼저 복잡한 상황에서도 단일 카메라만을 이용하여 다중 물체를 강인하게 추적하는 방법을 개발하였다. 시간적으로 지연된 결과나 미래의 입력 데이터 없이 우리는 현재 시간의 입력 데이터와 바로 이전 시간의 추적 모델간의 온라인 데이터 연관 기법을 통해 강인한 성능을 보이면서 일괄 처리 기법에 비해 빠른 속도로 알고리즘을 수행한다. 우리는 다중 물체 추적 문제를 그래프에서 물체간 연결을 찾는 문제로 변환하고 이 문제를 풀기 위하여 물체의 크기, 중심간 거리, 움직임, 모양 정보 등을 이용하여 사후확률을 정의하였다. 그 결과 매우 혼잡한 환경에서도 정보양이 적은 머리 부분 탐지기를 잘 활용하여

좋은 추적 성능을 보였다. 또한 본 논문에서 제안된 방법은 현재의 탐지 결과가 기존의 추적 모델로 설명되지 않을 시 자동적으로 새로운 추적 모델을 생성하고 물체 간의 겹침 등으로 만들어 질 수 있는 부정확한 정보에 의한 추적 모델의 오염을 막기 위한 겹침 추정 알고리즘을 사용하였다. 제안된 단일 카메라 기반 다중 물체 추적 알고리즘의 성능을 보이기 위해 다양한 데이터 셋에서 실험을 하고 기존 알고리즘과 비교를 하였다.

이어 본 논문에서는 단일 카메라에서 제안된 방법을 확장하여 다중 카메라에서 다중 물체를 추적하는 온라인 데이터 연관 기법을 제안하였다. 다중 카메라는 물체간 겹침이나 배경 뒤에 가려짐이 발생할 때 단일 카메라 보다 좋은 양질의 정보를 제공할 수 있지만 데이터 연관 알고리즘의 입력 데이터에 대한 관점에서 보면 이러한 증가된 정보량이 항상 더 선호되는 것은 아니다. 다중 카메라에서의 데이터 연관 기법을 수행하는 것은 데이터를 시,공간적으로 동시에 연결을 해야 하므로 단일 카메라에서의 데이터 연관 기법보다 훨씬 복잡하다. 이 문제의 가능한 해 공간 (solution space) 가 매우 크기 때문에 이 문제는 NP-난해 문제 (NP-hard) 로 알려져 있다. 하지만 대부분의 기존의 방법들은 정확도를 위해서 영상 전체를 모두 사용하는 일괄 처리 기반의 알고리즘을 채택함으로써 문제의 복잡도를 매우 크게 한다는 단점이 있다. 이러한 기존 알고리즘들의 문제를 풀기 위하여 우리는 온라인 데이터 연관기법을 단일 카메라와 마찬가지로 그래프에서 물체간 시,공간적 연결을 찾는 문제로 바꾸고 이를 위하여 사후확률 최대화 방법을 통하여 이를 해결하였다. 제안된 방법은 현재 시간의 탐지 결과와 바로 이전 시간까지의 추적 모델만을 연결함으로써 온라인 어플리케이션에 적용할 수 있다는 장점이 있다. 데이터간의 연결 및 유사도를 측정하기 위하여 영상 내에서의 위치, 모양, 속도 정보 및 카메라 정보를 활용한 3D 좌표 상에서의 위치 정보를 사용하였다. 마지막으로 본 논문에서는 여러 다양한 데이터 셋에 대해 제안된 알고리즘을 실험함으로써 기존의 뛰어난 알고리즘들과 비교성능을 보였다.

주요어: 물체 추적, 온라인 물체 추적, 데이터 연관기법, 매칭 그래프, 사후확률 최대화 기법, 다중 카메라

학번: 2008-22908