# Stony Brook University

# Collaborative and Heterogeneous Signal Processing Methodology for Mobile Sensor Based Applications

A Dissertation Presented

by

Shung Han Cho

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

August 2010

**Stony Brook University**

The Graduate School

**Shung Han Cho**

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree,
hereby recommend acceptance of this dissertation.

Sangjin Hong, Advisor of Dissertation
Associate Professor, Department of Electrical and Computer Engineering

Muralidhara Subbarao, Chairperson of Defense
Professor, Department of Electrical and Computer Engineering

Monica Fernandez-Bugallo, Assistant Professor
Department of Electrical and Computer Engineering

Hongshik Ahn, Professor
Department of Applied Mathematics and Statistics

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

## Abstract of the Dissertation

# Collaborative and Heterogeneous Signal Processing Methodology for Mobile Sensor Based Applications

by

Shung Han Cho

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

2010

Multiple object tracking and association are key capabilities in mobile sensor based applications (i.e., a large scale flexible surveillance system and multiple robots application system). Such systems track and identify multiple objects autonomously and intelligently without human operators. They also flexibly control deployed sensors to maximize resource utilization as well as system performance. Moreover, methodologies for the tracking and association should be robust against non-ideal phenomena such as false or failed data processing. In this thesis, we address various issues and present approaches to resolve them in collaborative and heterogeneous single processing for the applications.

Multiple object association (finding the correspondence of objects among cameras) is an important capability in multiple cameras environment. We introduce a locally initiating line-based object association to support flexible camera movements. The method can be extended to support multiple cameras through pair-wise collaboration for the object association. While the pair-wise collaboration is effective for objects with the enough separation, the association is not well-established for objects without the enough separation and it may generate the false association. We extend the locally initiating homographic lines based association method to two different multiple camera collaboration strategies that reduce the false association. Collaboration matrices are defined with the required minimum separation for an effective collaboration. The first strategy uses the collaboration matrices to select the best pair out of many cameras having the maximum separation to efficiently collaborate on the object association. The association information in selected cameras is propagated to unselected cameras by the global information constructed from the associated targets. While the first strategy requires the long operation time to achieve the high association rate due to the limited view by the best pair, it reduces the computational cost using homographic lines. The second strategy initiates the collaboration process of objects association for all the pairing cases of cameras regardless of the separation. While the repetitive association processes improve the association performance, the transformation processes of homographic lines increase exponentially.

Identification of tracked objects is achieved by using two different signals. The RFID tag is used for object identification and a visual sensor is used for estimating object movements. Visual sensors find the correspondence among cameras and lo-

calize them. An association of tracked positions with identifications utilizes object dynamics of crossing the modeled boundary of identification sensors. The proposed association method provides association recovery against tracking and association failure. We also consider coverage uncertainty induced by identification signal characteristics or multiple objects near the boundary of identification sensor coverage. A group and incomplete group association are introduced to resolve identification problems with coverage uncertainty. The simulation results demonstrate the stability of the proposed method against non-ideal phenomena such as false detection, false tracking, and inaccurate coverage model.

Finally, a novel self localization method is presented to support mobile sensors. The algorithm estimates the coordinate and the orientation of mobile sensor using projected references on visual image. The proposed method considers the lens nonlinearity of the camera and compensates the distortion by using a calibration table. The algorithm can be utilized in mobile robot navigation as well as positioning application where accurate self localization is necessary.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Introduction

Many researchers have much interest in a multiple camera based surveillance system [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [23] [24]. While the tracking performance of a single camera is restricted by a finite view, multiple cameras can collaborate to track objets with redundant or broad views. However, the multiple cameras based surveillance system has several issues such as optimal camera placements [3] [4], calibration of multiple cameras [5] [6], finding the correspondence of objects [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [23], camera handoff [25], etc. In this dissertation, we focus on a multiple object association finding the correspondence of objects among cameras. Especially, it is considered for flexible camera movements in which a reference ground plane may not be viewable by cameras.

Recently, heterogeneous sensor network has also received much attention in the

field of multiple objects tracking to exploit advantages of using different functionalities [26] [27]. Visual sensor is one of the most popular sensors due to its reliability and ease of analysis [1] [28] [29]. However, the visual sensor based tracking system is limited only to recording the trajectory of objects because visual sensors have several limitations for object identification [30] [31] [32] [33]. One of the main difficulties for the visual sensor-based object tracking is that distinguishable characteristics of the objects are non-trivial to be constructed for all the detected targets due to the objects' similarity in color, size and shape. Moreover, accurate feature extraction is not always guaranteed. Therefore, identifying an object with features is a non-trivial problem. Also, several identification sensors, such as RFID (Radio Frequency Identification) system, fingerprint or iris recognition system, have been utilized for object identification. However, the functionality of these sensors is limited only to the object identification and they are difficult to be used for the object tracking [34] [35] [36]. They can only alarm human operators for events triggered by identification sensors but cannot make intelligent decisions for them. For example, they cannot monitor the movement pattern of authorized people in special areas. Therefore, an identification sensor can only complement the visual sensor based tracking system for the intelligent surveillance system.

In mobile sensor based applications, self-localization is a key functionality to support flexible movement of mobile sensors. The position is usually estimated based on odometry and the measurement obtained by peripheral devices such as ultra-sonar, range finder, and visual sensor. However, odometry is usually unreliable due to the internal factor of encoder device and the external factor of slippage. These kinds of

errors accumulated in navigation continuously affect the estimation accuracy [37] [38]. Range finder and sonar are often used in navigation [39] [40] [41] [42]. However such measurement is not reliable in highly-dynamic environments where the radar or sonar beams can be frequently blocked or confused by moving objects such as people. They are also not applicable to localization in large area because of their limited range. Also, passive sensor requires active landmarks such as beacon, which requires modification of environment and is not practical especially in an outdoor environment. Moreover, interference between several mobile sensors causes inability to properly localize their locations. Therefore, an efficient and accurate self-localization method is required to maintain the movement of mobile sensors.

## 1.2 Application Model

Heterogeneous sensor network in Fig. 1-1 consists of two types of sensors: one is a visual sensor and the other is an identification sensor (e.g. check-in at the airport is equivalent to the identification by an identification sensor). When more than two cameras detect an object, the system obtains the redundant information. The benefit of multiple cameras is to complement each other by using redundancy. When multiple cameras are used for the surveillance, it improves the occlusion situation arisen in the single camera and broadens the field of view of tracking. In order to consistently track objects, it is critical to find correspondent objects among cameras (i.e., multiple objects association among multiple cameras). Currently, the human operator analyzes and maintains the information. However, the efficiency and accuracy are restricted by

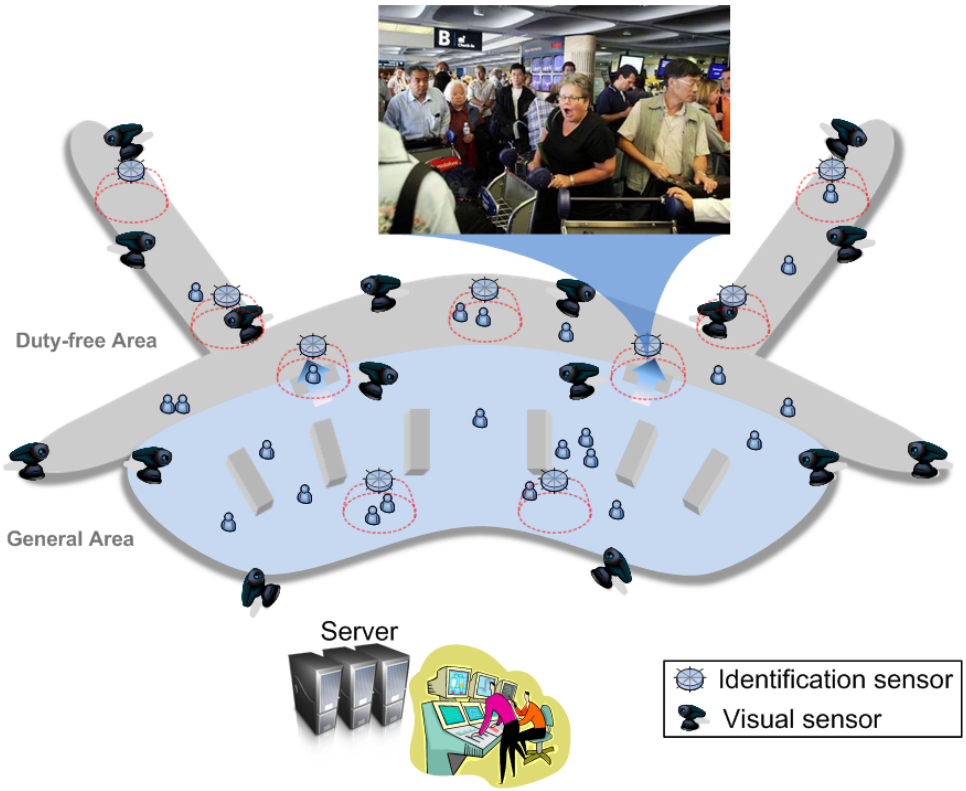



Figure 1-1: Example of an application model with heterogeneous mobile sensors such as visual sensors and identification sensors

the number of cameras. Thus, the association method supporting multiple cameras is required for the automated surveillance system.

While it is assumed that identification sensors operate correctly, they can be classified into two types in terms of the coverage issue in the proposed approach: one is a RFID-type ID sensor and the other is a non-RFID-type ID sensor. When non-RFID-type ID sensors are used for object identification, the effect of the coverage uncertainty is minimized since they usually identify a single object at one time. However, they usually require the long processing time to extract and analyze the features of a target. On the other hand, while RFID-type ID sensors have the benefit of the short processing time to identify objects, they suffer from the effect of the coverage

uncertainty (i.e., multiple objects can be registered simultaneously in the uncertain coverage of RFID-type ID sensors). Objects emit the radio signal to the RFID-type sensors and the effect of the coverage uncertainty is maximized with the RFID-type sensors. With respect to practical issues, object collision and tracking failure are common problems with both RFID-type and non-RFID-type sensors and coverage uncertainty is only for the RFID-type-sensor problem. The main problem is how to achieve data association of position information by a visual sensor with identification information by an identification sensor under these issues. In an ideal situation, it can only be a simple engineering task that one registered ID is associated with one estimated position within the coverage of an identification sensor. However, ID assignment becomes a non-trivial problem when objects are densely populated in the surveillance region, therefore the simple ID assignment cannot be achieved due to frequent collisions between objects or simultaneously entering objects. A collision between the objects can lead to a failure in tracking objects since they are too close to be differentiated for position and ID assignments.

The proposed approach can be applied to not only public areas (e.g. schools, hospitals and shopping malls) but also highly secured areas (e.g. airports, military facilities and government organizations). As an example of possible scenarios, serious offenders with attached ID tags can be tracked with the proposed method in order to ensure the safety at public places in cities. Also, the surveillance system with the proposed approach can keep tracking passengers in an airplane check-in or military personnel in a special area. It assumes that each object has its own identification such as a RFID tag, fingerprint, and iris. Identification sensors are usually installed

at the gates of restricted areas and a visual sensor tracks objects. For the airport application, the check-in counter can play the role of the ID-sensor. Whenever an object goes across the gates, the registered ID by an identification sensor is associated with the position estimated by a visual sensor. The system continuously watches the surveillance region by checking authorized IDs in the restricted areas.

## 1.3 Overview

In order to find the correspondences of objects among cameras, we present locally initiating line-based object association method. Targets on images are associated by checking if a locally generated homographic line uniquely crosses a correspondent target in an other camera. Reference planes of target heights are used in generating and transforming homographic lines while a common reference plane is not necessary shown to cameras. However, association performance is restricted by the expanded size of targets to compensate for uncertain parameters (i.e., height uncertainty of targets, variance of target locations by detection performance, and synchronization issues between multiple cameras). The issue of the uncertain parameters, which is dependant to camera and object locations, is addressed with respect to the effectiveness of homographic lines. In order to achieve distinctive crossing by transformed homographic lines, the proposed method transiently minimizes the expanded size of a target as compensating for the uncertain parameters. Height uncertainty is alleviated by a multi-camera localization scheme and synchronization issues are compensated for by object velocities estimation. A system can maintain and reconfirm effectively

object associations with the proposed method. The simulation results demonstrate that the proposed method significantly improves association performance as compared with the non-compensated method.

We extend the locally initiating homographic lines based association method to two different multiple camera collaboration strategies that reduce the false association. Collaboration matrices are defined with the required minimum separation for an effective collaboration because homographic lines for objects association are ineffective with the insufficient separation. The first strategy uses the collaboration matrices to select the best pair out of many cameras having the maximum separation to efficiently collaborate on the object association. The association information in selected cameras is propagated to unselected cameras by the global information constructed from the associated targets. While the first strategy requires the long operation time to achieve the high association rate due to the limited view by the best pair, it reduces the computational cost using homographic lines. The second strategy initiates the collaboration process of objects association for all the pairing cases of cameras regardless of the separation. In each collaboration process, only crossed targets by a transformed homographic line from the other collaborating camera generate homographic lines. While the repetitive association processes improve the association performance, the transformation processes of homographic lines increase exponentially. The proposed methods are evaluated with real video sequences and compared in terms of the computational cost and the association performance. The simulation results demonstrate that the proposed methods effectively reduce the false association rate as compared with basic pair-wise collaboration.

Moreover, an approach for dynamic object association and identification is proposed for heterogeneous sensor network consisting of visual and identification sensors. Visual sensors track objects by a 2-D localization and identification sensors (i.e., RFID system, fingerprint or iris recognition system) are incorporated into the system for object identification. This paper illustrates the feasibility and effectiveness of information association between the position of objects estimated by visual sensors and their simultaneous registration of multiple objects. The proposed approach utilizes the object dynamics of entering and leaving the coverage of identification sensors, where the location information of identification sensors and objects is available. We investigate necessary association conditions using set operations where the sets are defined by the dynamics of the objects. The coverage of identification sensor is approximately modeled by the maximum sensing coverage for a simple association strategy. The effect of the discrepancy between the actual and the approximated coverage is addressed in terms of the association performance. We also present a coverage adjustment scheme using the object dynamics for the association stability. Finally, the proposed method is evaluated with a realistic scenario. The simulation results demonstrate the stability of the proposed method against non-ideal phenomena such as false detection, false tracking, and inaccurate coverage model.

Finally, a novel self localization method is presented for mobile sensors. The algorithm estimates the coordinate and the orientation of mobile sensor using projected references on visual image. The proposed method considers the lens non-linearity of the camera and compensates the distortion by using a calibration table. The method determines the coordinates and orientations with iterative process, which is very ac-

curate with low computational demand. We identify various sources of error on the coordinate and orientation estimations, and present both static sensitivity analysis of the algorithm and dynamic behavior of the mobile sensor. The algorithm can be utilized in mobile robot navigation as well as positioning application where accurate self localization is necessary.

The rest of this dissertation is organized as follows. Chapter 2 proposes a locally initiating line-based object association in large scale multiple cameras environment. The effect of the non-ideal parameters is addressed and an improved approach is presented. Chapter 3 presents two different camera collaboration strategies for objects association. Chapter 4 presents association and identification approaches for a situation where objects are densely populated. In Chapter 5, we present a novel self localization method for mobile sensors. In Chapter 6, we finally conclude the dissertation along with future works.

# Chapter 2

# Locally Initiating Line-Based Object Association in Large Scale Multiple Cameras Environment

## 2.1 Introduction

When multiple cameras operate collaboratively, the detected targets in multiple cameras should be associated for the consistent and reliable tracking. The various methods are investigated for multiple objects association. In feature based matching method, the features such as shape, motion or color are used to find the corresponding targets [8] [9] [10] [18] [19]. However, these methods do not guarantee the association performance where objects have similar features. Some authors use both geometric information and visual information to find correspondence between cameras [13] [20]. Also, there is the approach that uses the multiple features at the same time [17].

The common problem for the feature-based methods is to utilize the probabilistic approach. This may cause the false association and the tracking becomes unreliable. When the geometry information is available, the globally defined homographic lines are used to find the corresponding targets [14] [21]. However, this approach requires the camera calibration or the training process before the system operates. Moreover, the calibration of surrounding is necessary whenever the camera movement changes.

A reliable association mechanism is necessary for maintaining the consistent tracking information in the large scale multiple cameras environment. The association needs to be performed whenever the system requires to establish or confirm the object association. Also, the system should support the dynamic change of the camera configuration for broadening the effective tracking coverage. However, the conventional camera model requires the calibration whenever the camera configuration changes for accuracy. This is not appropriate for the automatic surveillance where the change of camera configuration frequently occurs. Therefore, the large scale tracking system requires the association method supporting the flexible camera configuration on demand.

In this chapter, the locally initiating line-based object association method is presented so that association is established on demand. In order to avoid the calibration whenever the camera configuration changes, the parallel projection model is used for generating and transforming a homographic line [22] [55]. We investigate plausible non-ideal parameters (i.e, transformation error, height uncertainty and asynchronous problem) affecting the association performance. With this analysis, the threshold distance between the detected targets is defined to indicate the effectiveness of a locally

generated homographic line on the other cameras. The threshold distance based line generation algorithm is presented for multiple objects association. We verify the proposed method with actual image frames. Finally, we discuss the strategy to improve the association performance by using the temporal and spatial redundancy.

The remainder of this chapter has 4 sections. In Section 2.2, we present overview of the application model of visual surveillance system with multiple cameras. Section 2.3 presents an association method using local initiated homographic lines and investigates the non-ideal parameters affecting the association. In Section 2.4, we verify the proposed method with the actual image frames and discuss the strategy to improve the association performance by using the temporal and spatial redundancy. Finally, our contribution is summarized along with future works in Section 2.5.

## 2.2 Application Model and Problem Description

### 2.2.1 Application Model

When more than two cameras detect objects, the system obtains the redundant information. The benefit of multiple cameras is to complement each other by using the redundancy. When multiple cameras are used for the surveillance, it improves the occlusion situation arisen in the single camera and broadens the field of view of tracking. In order to consistently track objects, it is critical to find corresponding targets among cameras (i.e., multiple objects association among multiple cameras). Currently, the human operator analyzes and maintains the information. However, the

efficiency and accuracy are restricted by the number of cameras. Thus, the association method supporting multiple cameras is required for the automated surveillance system.

The aim of the proposed method is to support the situation that the common ground plane may not be shown on all the cameras and association targets are the faces of objects. The conventional association methods construct fundamental matrix or homography matrix with the predetermined corresponding points or known references. In a normal situation, the ground plane is usually shown to all the cameras and the homography matrix is easily constructed on the ground plane. Once a homography matrix is constructed on the common (ground) plane, the correspondence of objects can be easily found by transforming points with the homography matrix. However, it is not always guaranteed that the common plane is shown to all the cameras in the surveillance system. Even though they share the common plane, it is difficult for the system to automatically find corresponding points for constructing the homography matrix. Thus, the association methods using the fundamental matrix or homography matrix are not appropriate for an autonomous surveillance system with dynamically moving cameras.

### 2.2.2 Related Work

There are two approaches to find the corresponding targets among multiple cameras. One is feature based and another is geometric based approaches. Researchers have used various features such as color, histogram, height and motion for feature based

approaches. However, all of them are not the unique characteristics for objects. Although some features are used together to improve accuracy, they still rely on the probabilistic approach. Their performance is severely affected by objects having the similar features [9] [8] [10]. For instance, when people wear the similar uniform or the cloth which has the different color at the front and the back, it is hard to construct the color-based information and associate objects. Also, the accurate feature extraction requires the prohibitively large computation. In [43], the principal axis of a target is used for finding correspondences. Although this can be more reliable than other features, the decision may be confused when the floor is shown in one camera and not in another camera.

In the geometric based method, some authors try to align two different images by using the geometric transformation between cameras [11] [12]. Although the known geometric relationship between two different cameras facilitate an association process finding the correspondence of objects, a system requires known reference points between two different cameras to construct the geometric relationship. This kind of the pre-construction process can restrict the camera movement in a large scale multiple camera environment. Also, the performance is only ensured when disparity between the geometric relationship of cameras is small. Another method is to use the boundary of the field of view (FOV) of cameras on the ground plane [21]. The boundary information of FOVs are projected onto the image of the other cameras. The object association is established when corresponding targets cross the same boundary on each camera. Although this method finds the correspondence of objects effectively, there are several limitations. The association is established only when objects cross

over the boundaries of FOVs. This cannot support the re-association right after association failures due to miss detection or occlusion. Also, when a camera pans or tilts, the system needs the calibration or training to generate the FOV lines on the other cameras. Another method is to use the epipole line with the relationship between cameras [7]. A point on a camera is in a line with the camera's focal point. This line is shown in the other camera by using the camera geometry. If the epipole line from an object in a camera passes the object in other camera, the correspondence between detected targets is established. However, this method highly depends on the camera geometry and is sensitive to the accuracy of the camera calibration. The fundamental problem for objects association with these approaches is to restrict the camera movement because of the training or the calibration. They usually use the perspective projection model for the transformation and alignment. Cameras should be calibrated to decrease the non-ideal error before the system starts in the application. However, the calibration process usually needs the planar and regular target as reference, which is hard to obtain in real time tracking applications [44] [45] [46] [47] [48]. That is why the calibration is performed at the initial system stage with the known reference. The camera movement is also restricted to avoid the calibration while the tracking system operates. Moreover, in an autonomous surveillance system, the proper camera movement for the object tracking is necessary to secure the effective view and to manage the resource.

### 2.2.3 Approach Overview

In the proposed approach, the parallel projection model based transformation technique is applied for the object association [22]. At the initial system stage, the zoom factor and the non-ideal parameter compensation table are measured. Since the zoom factor table is independent of the camera configurations (i.e., panning, tilting and zooming), the camera does not require the calibration whenever the camera changes the movement or location. The object association is locally initiated by generating a homographic line on each target. Locally generated homographic lines are exchanged among different cameras to find intersections with corresponding targets. The detailed association process with locally generated homographic lines is explained in the following section.

## 2.3 Multiple Objects Association

### 2.3.1 Local Line-based Association

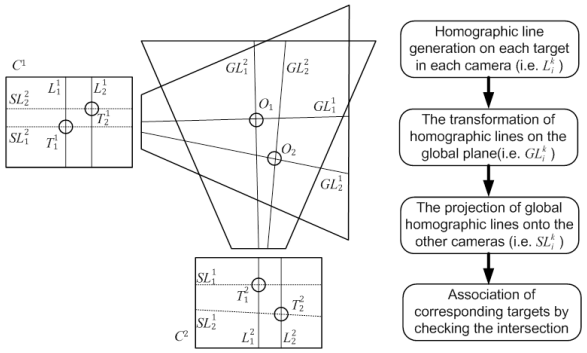



Figure 2-1: Illustration of the homographic lines based association method with the flow chart.

Object association is initiated by locally generating homographic lines on detected targets in each camera. The locally generated homographic lines are exchanged among different cameras to find intersections with targets through a global plane such as the ground plane or the reference plane. Fig. 2-1 illustrates the homographic lines based association method using two cameras. $L_i^k$ denotes a locally generated homographic line on target $T_i^k$, a detected target of object $i$ on camera $C^k$. $GL_i^k$ is a transformed homographic line on a global plane and $SL_i^k$ is a projected homographic line from $GL_i^k$ on the other camera $C^l$. In the figure, solid lines are locally generated homographic lines and dotted lines are projected homographic lines. The association between targets is established when their projected homographic lines intersect with corresponding targets each other. For example, $T_1^1$ intersects with a projected homographic line generated from $T_1^2$ and vice versa, and targets $\{T_1^1, T_1^2\}$ and $\{T_2^1, T_2^2\}$ are associated respectively.

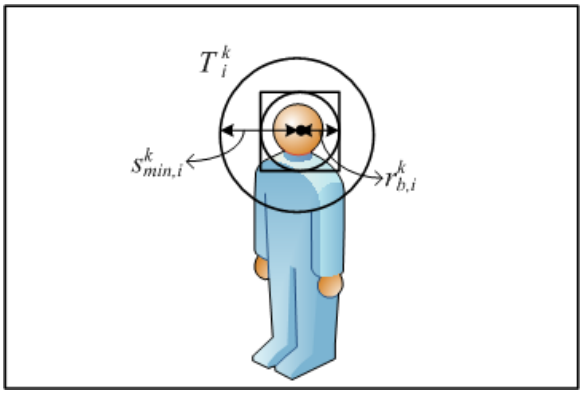



Figure 2-2: Illustration of key parameters in locally initiating homographic lines based association method.

The association establishment of locally generating homographic lines based association is determined by the correct intersection with the corresponding targets.

However, a projected homographic line can be deviated from the centroid of a target because of an inaccurate global plane and synchronization issues between cameras. The accuracy problem of a global plane becomes significant when the common ground plane may not be shown to all the cameras and the faces of objects are extracted [49] [50] as association targets. Because the heights of association targets are not known to a system, the global plane with the average height of targets is used for the transformation and the projection of homographic lines it can create the deviation problem. In order to ensure the correct intersection with the corresponding target, we introduce the tolerable region around a target with the size $r^k_{min,i}$ as shown in Fig. 2-2. Radius $s^k_{min,i}$ denotes the adjusted radius of the tolerable region for a projected homographic line deviated by an inaccurate global plane and synchronization issues between cameras. The circle with $s^k_{min,i}$ is called as an association circle throughout this chapter. In the following section, we identify non-ideal parameters causing the deviation problem and how to determine the size of $s^k_{min,i}$.

### 2.3.2 Non-ideal Parameters in Line Generation

The size of an association circle affects the association performance of the proposed method. While it ensures the correct intersection of homographic lines against the non-ideal parameters, the large size of an association circle may increase the possibility that the target intersects with homographic lines generated from irrelevant detected targets. Therefore, the appropriate size of an association circle is important in terms of the association performance. Since the size of an association circle depends on

the effect of the non-ideal parameters among multiple cameras, we predetermine the size of an association circle by considering all the possible camera movements. In the following subsections, we show the effect of the non-ideal parameters to determine the size of an association circle. While $C^1$ is fixed at $P_1(3m, 0m, 3m)$, $C^2$ moves to three positions (i.e., $P_2(6m, 3m, 3m)$, $P_3(3m, 6m, 3m)$ and $P_4(0m, 3m, 3m)$) as varying the tilting and panning angle also.

**Effect of Transformation Error**

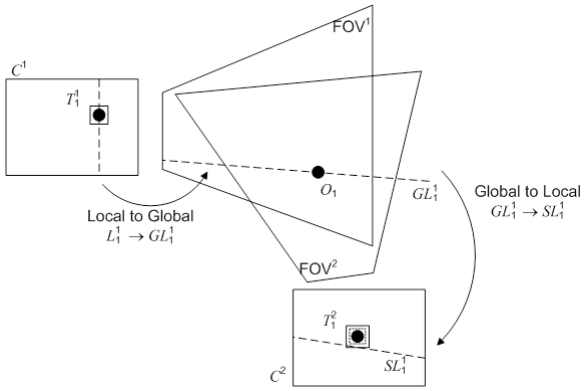
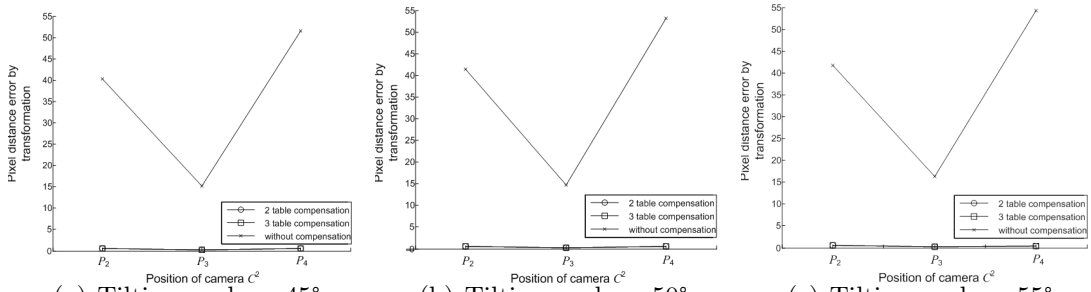



Figure 2-3: Illustration of the transformation error caused by the camera nonlinearity.

When the homographic line is generated, the image coordinates are transformed to the global coordinates by using the parallel projection model as shown in Fig. 2-3. This model uses the table based error compensation to support the dynamic camera configuration instead of using the known reference points. Since the image is affected by the lens distortion or the camera configurations (panning, tilting, and zooming), this deteriorates the transformation accuracy. This can cause the problem that a homographic line does not intersect with the detected target.

Fig. 2-4 shows the maximum transformation error of the generated homographic




(a) Tilting angle = 45° (b) Tilting angle = 50° (c) Tilting angle = 55°

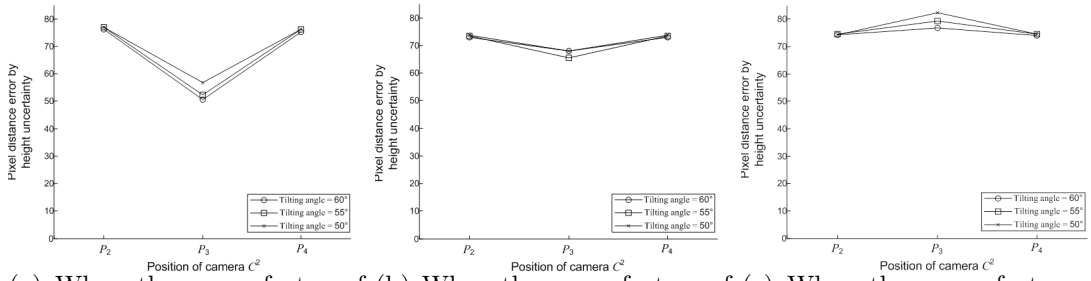
Figure 2-4: The transformation error on $C^2$ according to the camera status (i.e., position, panning and tilting angle).

line on camera $C^2$. For simplicity, the zoom factors of horizontal axis and vertical axis of the image are set to 1 and 1.7 respectively. Camera $C^1$ is fixed at $P_1$ and the panning angle of camera $C^2$ varies $-10°$to $10°$and the tilting angle varies $45°$to $60°$at each position. The homographic lines from $C^1$ are transformed onto the ground and the global lines are transformed onto $C^2$. Among the transformed lines, the maximum pixel distance from the ideally transformed line is shown in y-axis. When the homographic line is transformed without the compensation for the lens non-linearity, the pixel distance error is very larger than the homographic line with the compensation. The result shows that the pixel distance error is almost $1pixel$ on the image when the lens non-linearity is compensated and the transformation error is negligible with the compensation.

**Effect of Height Uncertainty**

Since the faces of objects are association targets, the system does not know the height of the global plane accurately where homographic lines are generated. Thus, the system uses the average height of objects for the global plane. The difference

between the actual height and the average height of objects induces homographic lines to be shifted from the corresponding targets. $\sigma_h$ denotes the amount of pixels to compensate for the shifted homographic lines due to height uncertainty.




(a) When the zoom factor of $C^2$ is 0.2 smaller than that of $C^1$

(b) When the zoom factors of two cameras are the same

(c) When the zoom factor of $C^2$ is 0.2 larger than that of $C^1$

Figure 2-5: The effect of the height uncertainty on the other camera according to the camera status (i.e., position, panning and tilting angle, zoom factor).

Fig. 2-5 shows how much the homographic line is shifted from the generated homographic line due to the height uncertainty. The effect of the height uncertainty is simulated similarly to the transformation error. The average height is assumed to be 1.6m. As varying the status of camera $C^2$ at each position with the height uncertainty, $-0.1m \sim 0.1m$, the maximum pixel distance error is measured. The value of y-axis indicates the maximum number of pixels between the projected homographic line with the known height and the projected homographic line with the height uncertainty. Since the effect of the height uncertainty is measured by transforming homographic lines, it also considers the effect of the transformation error. The result indicates that the effect of the height uncertainty increases when the zoom factor is large on the other camera. This is because an object is viewed more closely to a camera when the zoom factor is larger.

**Synchronization Issue**

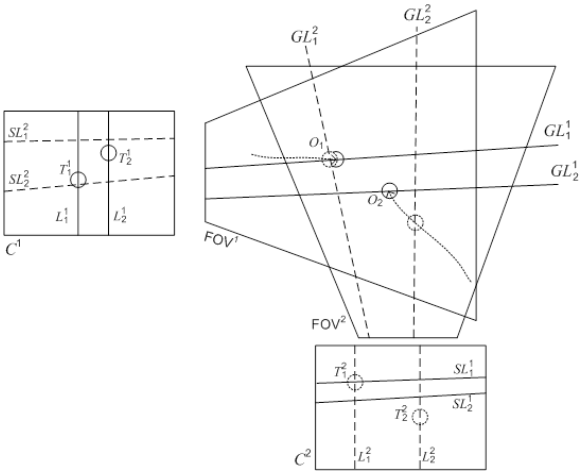



Figure 2-6: Illustration of association problem in asynchronous image frames.

Another problem in object association is synchronization issue among cameras. Suppose $N$ cameras process $P$ frames per second. Then, the maximum synchronization error is $1/P$ second (i.e., out of synchronization by 1 frame) because they sample an image at the same rate. In Fig. 2-6, camera $C^2$ is delayed by 1 frame two cameras are placed in the perpendicular way for the maximum effect of the synchronization issue. Object $O_2$ moves in the diagonal way to the optical axis of camera $C^2$. Since the captured views of cameras are different from each other, the projected homographic line does not intersect with the detected targets of object $O_2$.

Fig. 2-7 shows the effect of the synchronization problem. $v_f$ represents the object speed per the image sampling time. It is assumed that the maximum frame delay between cameras $C^1$ and $C^2$ is 1 frame as an object moves in the perpendicular way to the optical axis of $C^1$ at speed $v_f$. The homographic line is generated from the delayed image of camera $C^1$ to the image of camera $C^2$. The maximum pixel difference

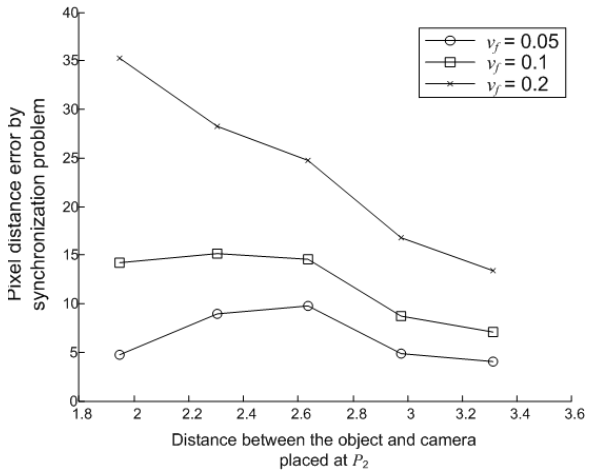



Figure 2-7: The effect of the synchronization problem depending on the distance between the object and camera.

error between the homographic lines from the delayed image and the detected target in camera $C^2$ is measured. As the object moves fast and closely to camera $C^2$, the pixel error increases. $\sigma_s$ is defined as the maximum pixel distance error caused by the synchronization issue depending on the applications (i.e., object speed and the camera position).

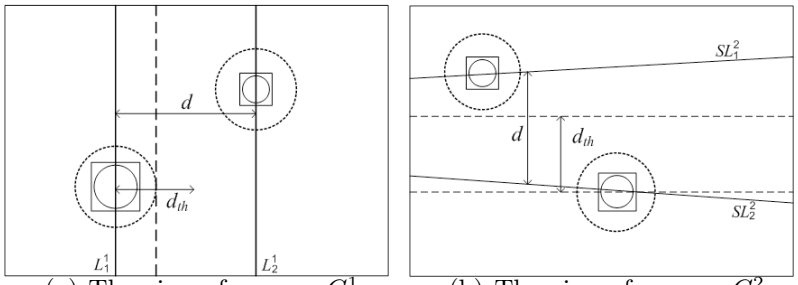
### 2.3.3 Parameters Affecting Association Performance

Since the effect of transformation is included in the height uncertainty, the required minimum size of an association circle is represented as,

$$s_{min} \simeq max(\sigma_h + 2\sigma_s)/2, \tag{2.1}$$

where $\sigma_h$ and $\sigma_s$ are measured in all the possible camera positions for the application. Based on this, the system increases the box size to $s_{min}$ so that a projected homo-

graphic line intersects with the detected box. On the other hand, the increased size of the detected target also increases the case that an object is intersected with multiple homographic lines. The proposed algorithm is affected by this case and the increased size may decrease the association performance. Moreover, the association condition is not satisfied when a homographic line intersects with more than two same objects on both cameras. Thus, the parameter indicating the effectiveness of the homographic line should be defined.




(a) The view of camera $C^1$ (b) The view of camera $C^2$

Figure 2-8: The view of each camera with $d_{th}$ when the homographic line is generated from camera $C_1$.

The threshold distance is the minimum distance between targets, which is required to generate the effective vertical homographic lines on the other cameras. Fig. 2-8 illustrates the threshold distance in each camera. The system knows which vertical homographic line is generated from which object in the local camera. Hence, although generated vertical homographic lines intersect with two detected targets in the local camera, it does not affect the association process. However, when the projected homographic lines from them intersect with two detected targets in the other camera, the system cannot determine the corresponding target. The one way to expect this is to use the globally defined threshold distance, $d_{th}$, which is represented as

$$d_{th} \simeq max(\sigma_h + 2\sigma_s), \tag{2.2}$$

where the maximum error of non-ideal parameters is considered. When the homographic lines are generated in one camera, the separation of the transformed lines from them is anticipated with $d_{th}$ in other cameras. However, we need to consider that $d_{th}$ is the reference value obtained with the predefined zoom factor.

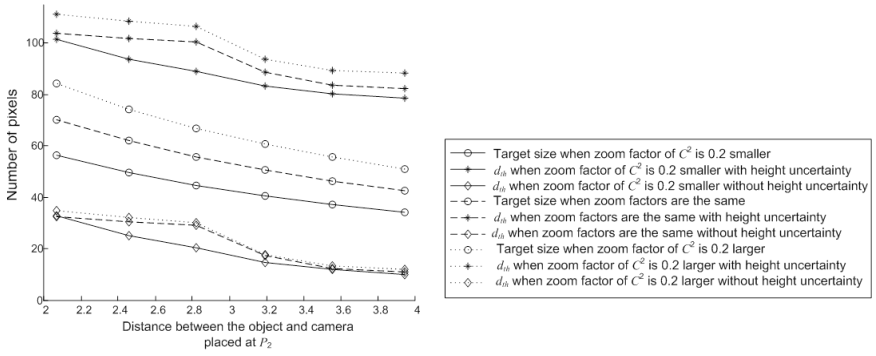



Figure 2-9: Relationship between the size of the detected object and $d_{th}$.

Fig. 2-9 illustrates the detected object size compared with $d_{th}$ and the zoom factor of other camera. $d_{th}$ is determined by (2.2) based on the simulation with the predefined zoom factor for the non-ideal parameters. The size of most detected objects is smaller than $d_{th}$ with the height uncertainty. Thus, the box size of them needs to be expanded so that the non-ideal factors can be compensated. However, $d_{th}$ decreases when the height is known because the effect of the height uncertainty is negligible. Moreover, if the other cameras use the different zoom factor from the camera where a homographic line is generated, $d_{th}$ also changes in proportion to the zoom factor.

When $d_{th}$ is considered for the line generation, the system has the case that two

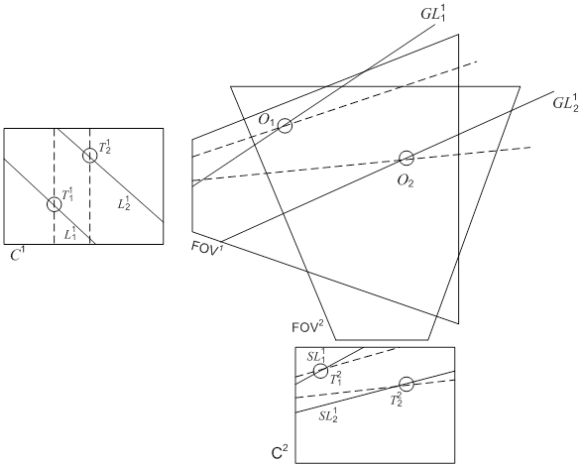
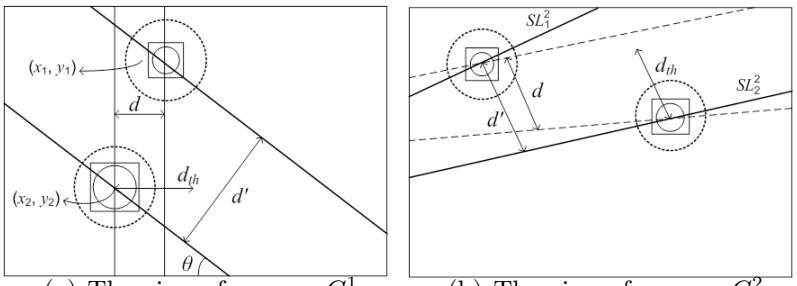



Figure 2-10: Illustration of the case that the vertical homographic line is not effective.




(a) The view of camera $C^1$ (b) The view of camera $C^2$

Figure 2-11: The view of each camera when the slant homographic line is generated from $C^1$.

objects are too close to be associated with the vertical homographic lines as shown in Fig. 2-10. If $d$ is smaller than $d_{th}$, the vertical homographic line is not effective as shown in Fig. 2-11. Hence we introduce a slant homographic line. As generating a slant homographic line, the distance between two homographic lines satisfies with the separation $d_{th}$. The angle $\theta$ is the slope of the homographic line. In order to obtain the new distance between the homographic lines, we use the distance between a point and a line. For example, the homographic line passing the center of object $O_1$ is defined by

$$tan(\theta)x - y - tan(\theta)x_1 + y_1 = 0. \tag{2.3}$$

The distance between the line and the center of $O_2$ is calculated by

$$d' = |\frac{tan(\theta)x_2 + y_2 - tan(\theta)x_1 + y_1}{(tan(\theta))^2 + 1}|. \tag{2.4}$$

The angle $\theta$ is chosen as finding $d' \geq d_{th}$. In order to decrease the number of comparison cases, the system can choose $\theta$ among the limited angle candidates (i.e., 0°, 45°and 135°).

**Vertical Line Based Association**

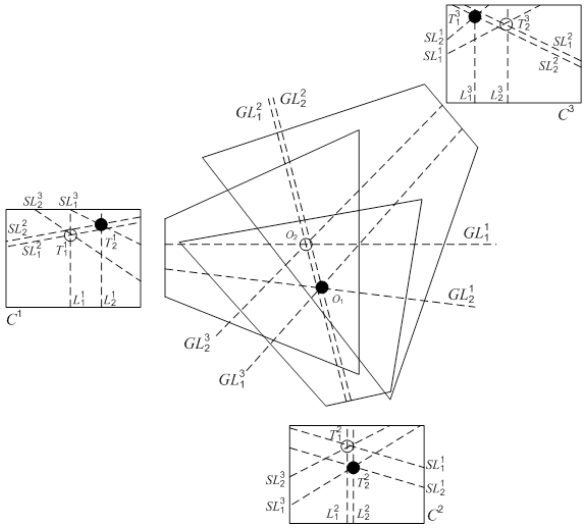



Figure 2-12: Illustration of object association by the vertical homographic lines.

This association method uses only the vertically generated homographic lines. Each camera generates the vertical homographic lines on the detected object considering the threshold distance. If the homographic line is not satisfied with the threshold

distance, the homographic line is not used for object association. As shown in Fig. 2-12, all cameras generate the homographic lines on the detected object. However, the homographic lines generated by camera $C^2$ do not satisfy the threshold distance. The distance between $L_1^2$ and $L_2^2$ is smaller than $d_{th}$. Thus, they are not effective anymore and only cameras $C^1$ and $C^3$ associate the detected objects.

**Slant Line Based Association**

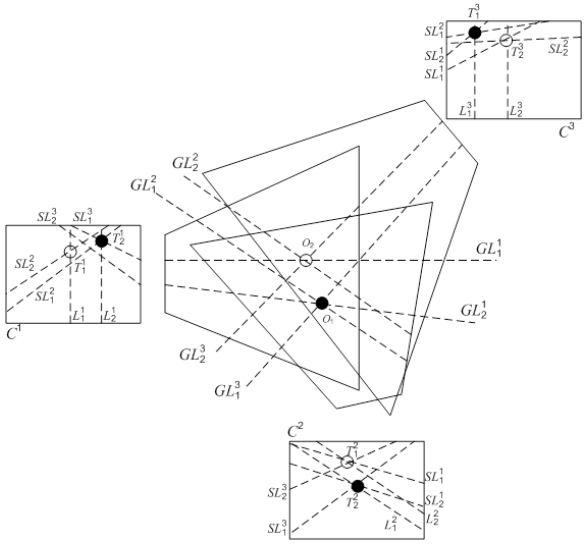



Figure 2-13: Illustration of object association by the possible slant homographic lines.

When slant homographic lines are used for object association, there are two methods to generate them. One is the locally selected line and another is the globally selected line. In the locally selected line method, the angle $\theta$ is chosen when the homographic line is generated. Fig. 2-13 shows the example how the slant homographic line is applied to object association. In camera $C^2$ which does not participate in object association with the vertical homographic lines, a slant homographic line is generated on each detected object. In the figure, $d$ of lines generated by camera $C^2$

is larger than $d_{th}$. Hence they are still effective and camera $C^2$ can also participate in object association. In the second approach, all kinds of effective homographic lines including the vertical lines are generated and transferred to the system. The system determines which lines are used for association after testing the threshold distance in all cameras. Although this can choose the best homographic lines for object association, this requires the large amount of computation for comparison with the threshold distance.

**Algorithm 1**: the line generation algorithm with $d_{th}$

```
Input : d_th,
Coordinates of detected objects with the expanded box size from each camera
for k = 1 to K do
    Calculate min d^90° between detected objects at C^k
    if d^90° < d_th then
        find max d^θ satisfying d^θ ≥ d_th
    end
    if d^θ is found then
        for i = 1 to I do
            if A detected target T_i^k has at least d_th of separation then
                Generate homographic lines with θ on detected objects
            end
        end
    end
end
```

Algorithm 1 explains the line generation method based on $d_{th}$. The purpose of this algorithm is to find the homographic lines with $\theta$ having the maximum separation in other cameras so that the association performance increases. First, the system determines the set of $\theta$s to be tested. As increasing the number of elements in the set, the comparison cases increase. Then, the system calculates the minimum distances when the homographic line is generated with each $\theta$. Finally, it chooses $\theta$ which has

the maximum distance among the minimum distances. Then, the system tests if each object has at least $d_{th}$ of the separation with other objects. If this is satisfied, the system generates the homographic line with $\theta$.

## 2.4 Simulation and Analysis

### 2.4.1 Simulation Setup

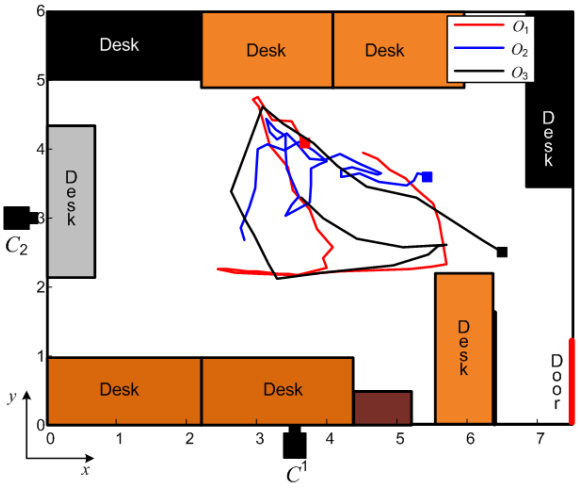



Figure 2-14: Illustration of our simulation setup showing objects trajectories and camera placements (Each square denotes the starting position of each object).

Fig. 2-14 shows objects trajectories and camera placements for analyzing the proposed association algorithm. Camera $C^1$ is placed at $(x = 3.57m, y = 0.05m, z = 2.24m)$ with tilting angle 73°and panning angle 0°and camera $C^2$ is placed at $(x = 0.05m, y = 2.97m, z = 2.34m)$ with tilting angle 68°and panning angle 0°. The total number of frames is 45 and object $O_3$ is shown to both cameras after frame 23. The height of object $O_1$ is $1.75m$, the height of object $O_2$ is $1.87m$, and the height of object $O_3$ is $1.72m$. The average height of objects is $1.78m$. Since a target height is

considered to be the center of an object face in height estimation, the target height is shorter than the real height of an object. Thus, $z = 1.7m$ is used as an initial global plane where homographic lines are transformed. If an object height is estimated, an object has its own global plane of an estimated object height. 60 pixels are used for the initial value of $\sigma_h + \sigma_s$.

### 2.4.2 Performance With Constant Radius of Association Circle

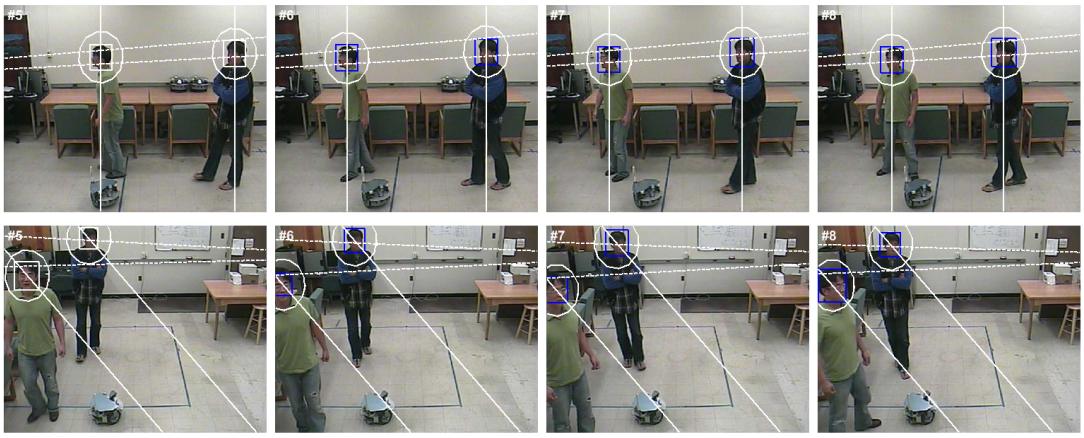


Figure 2-15: Image snapshots of object association with constant threshold based association algorithm at frame $5 \sim 8$ (The upper images are captured by camera $C^1$ and the lower images are captured by camera $C^2$. Blue rectangles of targets indicate associated targets; white solid lines, locally initiated homographic lines; and white dotted lines, transformed homographic lines from locally initiated homographic lines).

Fig. 2-15 shows snapshots for objects association with a constant radius. The system generates homographic lines on images only when the distance between targets is satisfied with the threshold. At frame 5, targets are not associated because transformed homographic lines cross over multiple targets. This indicates that the

threshold does not always lead to the correct decision for generating homographic lines. However, if the distance between targets is smaller than the threshold, the generated homographic lines are even less effective for objects association. After frame 5, targets between two images are associated by locally initiating homographic lines. As shown in the figure, an adjusted region size $s_{min}$ of each target is constant. Moreover, the simulation shows that transformed homographic lines deviate from the centroid of targets because of the non-ideal parameters. The association status of each object is shown in Fig. 2-16.

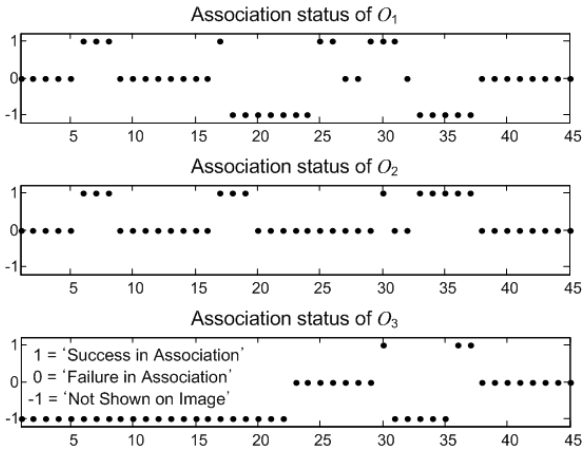



Figure 2-16: Simulation result of the association status of each object with the constant radius based association algorithm.

### 2.4.3 Association Performance Improvement Using Redundancies

This section discusses the strategy to improve the association performance by minimizing the adjusted radius of an association circle after targets are associated between cameras. The first non-ideal parameter to be compensated for is the height uncer-

tainty of targets. The effect of the height uncertainty is maximized when there is a significant discrepancy between actual height and average height of a target. If we can estimate a target height within a certain range, it is not necessary to consider significant uncertainty. A target can be localized in 3-D through finding the intersection of projected lines from targets in cameras to a ground [11]. However, the intersection of the projected lines may not exist because targets are detected in different views. Hence, we utilize the shortest line between two projected lines.

A corresponding point $p_i$ in each camera is projected onto the global ground plane (z = 0) using $p_i = Cp'_i$ where $C$ is a known camera matrix and $p'_i$ denotes a transformed point on a ground plane. Ideally, constructed lines between the camera and the global plane should intersect each other if the centroid of targets is the same position of an object in 3-D. However, this is not always guaranteed because the centroid of targets can be deviated by detection algorithms. Thus, we obtain a line which has the shortest distance between the constructed lines. Then, an object position is estimated using the middle point between two points that form the shortest line between constructed two lines.

The average height of targets(i.e., centroid of detected faces of objects) is set to be $1.7m$. The target height of object $O_1$ is $1.65m$, the target height of object $O_2$ is $1.73m$, and the target height of object $O_3$ is $1.6m$. Fig. 2-17 shows the estimated heights of three objects assuming that their corresponding targets are known. Estimated heights can be deviated from an actual target height because targets are detected in different views. However, the estimated heights converge on the actual target heights.

Fig. 2-18 shows how many pixels homographic lines are deviated from correspond-

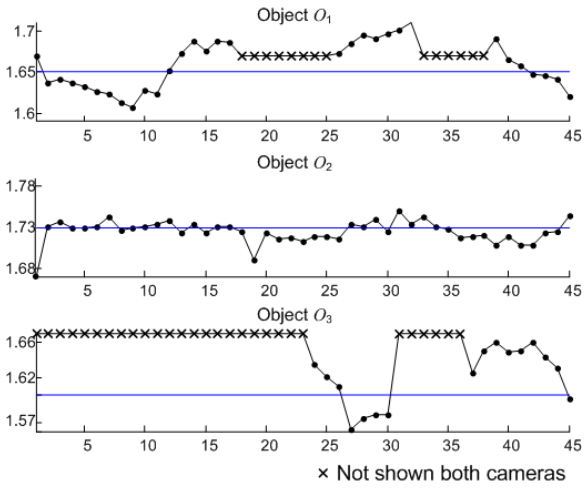
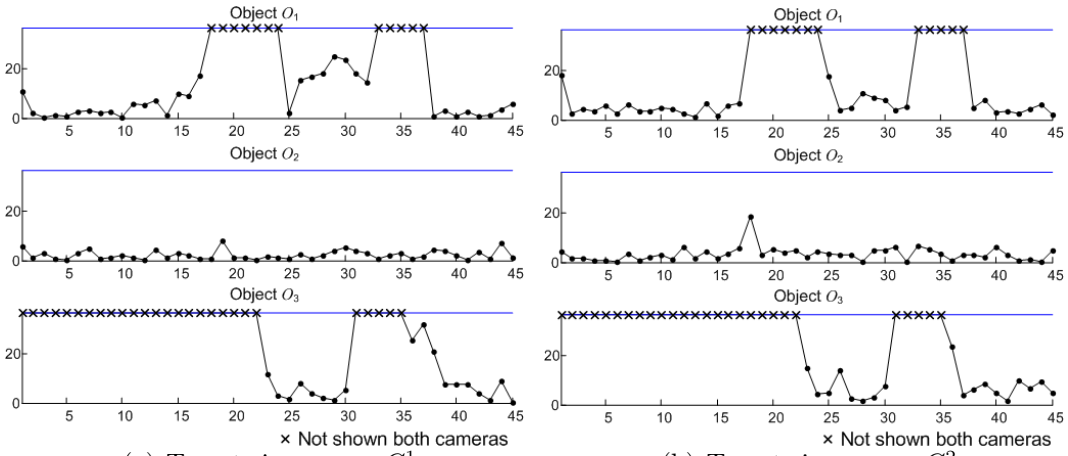



Figure 2-17: Estimated target heights of objects by the shortest distance between two lines after corresponding targets are found (a blue solid line is the average height of targets).

ing targets when estimated heights are utilized for the generation of homographic lines. The initial value of height uncertainty $\sigma_h$ is set to be 37 pixels assuming that the degree of height uncertainty is $0.1m$. When an object is not shown on both cameras, the initial value is used for the number of deviated pixels. The result indicates that estimated heights alleviate the deviation of homographic lines.

Since synchronization issues are caused by the physical network delay, it is difficult to correct the effect of synchronization issues. However, $\sigma_s$ can be optimally selected by using information such as an object speed and a distance between an object and a camera. In order to estimate $\sigma_s$ for the next frame, the global position of an object needs to be predicted. The next position is estimated based on the previous position and velocity. The velocity of an object is estimated based on the positions for at least two consecutive frames. The velocity of object $O_i$ at time $n$ (i.e. $v_{x,i}(n), v_{y,i}(n)$)is obtained by




(a) Targets in camera $C^1$ (b) Targets in camera $C^2$

Figure 2-18: The number of deviated pixels from the centroid of targets when estimated heights are utilized to generate and transform homographic lines.

$$v_{x,i}(n) = x_i(n) - x_i(n-1),$$

$$v_{y,i}(n) = y_i(n) - y_i(n-1),$$

where $(x_i(n), y_i(n))$ is global position at time $n$. This assumes a constant velocity object model. Since the sampling rate is usually higher than object velocity, the predicted position error by the incorrect object model is not significant. The object speed $v_f^i$ of object $O_i$ per a frame is also obtained by

$$v_f^i = \sqrt{{v_{x,i}}^2 + {v_{y,i}}^2}. \tag{2.5}$$

Fig. 2-19 shows the estimated speeds of objects per a frame when a frame rate is $8 frames/sec$ (the maximum time difference between cameras is $0.125 sec$). The

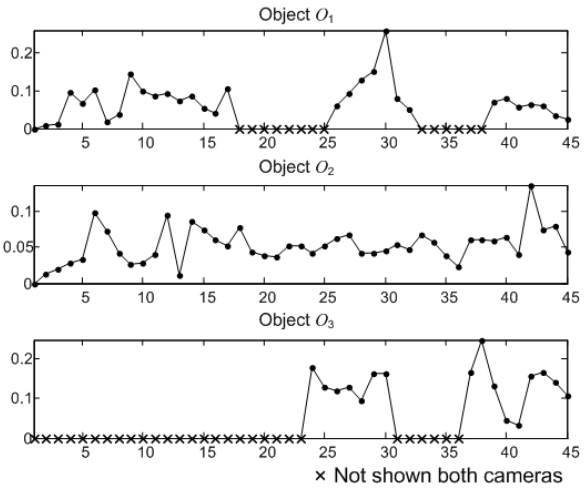


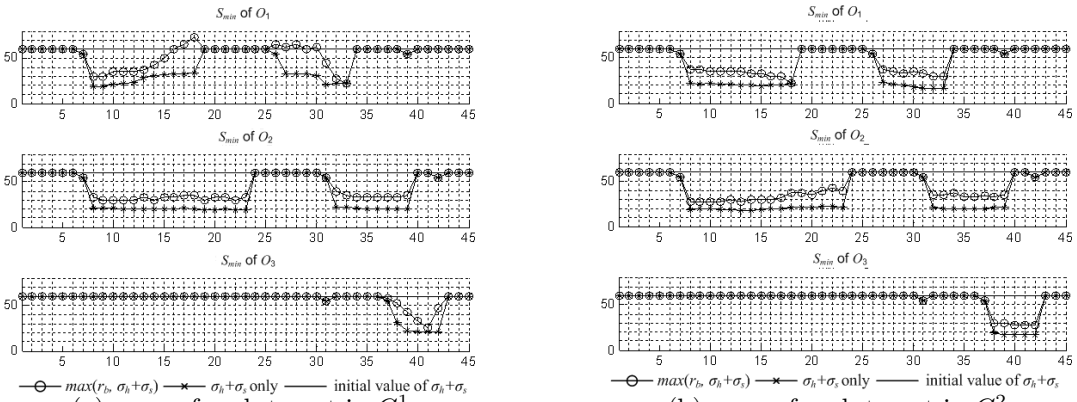

Figure 2-19: The estimated speeds of objects per a frame assuming that the frame rate is $8 frames/sec$ (the maximum time difference between cameras is $0.125 sec$).

| $v_f/2$ | $D_i^{k,l} < 3$ | $D_i^{k,l} >= 3$ |
|---|---|---|
| 0.05 | 10 | 5 |
| 0.1 | 22 | 12 |
| 0.2 | 45 | 25 |

Table 2.1: Constructed pixel error table for synchronization effect according to Fig. 3-7.

estimated speed is used to determine the optimal value for $\sigma_s$ with the preconstructed table representing the influence of the synchronization effect. Table 2.1 shows the example of constructed pixel error table by using the simulated data from Fig. 2-7. $v_f/2$ corresponds to the sampling period of Fig. 2-7 since the synchronization effect is measured with the average object speed $2m/s$. $D_i^{k,l}$ denotes a distance between the position of target $T_i^k$ and camera $C^l$. The table can be constructed in more detail for more subdivided selection.

Fig. 2-20 shows the variation of an adjusted radius $s_{min}$ as objects move around the surveillance region. The maximum object speed is slower than $0.2m/s$ in this




(a) $s_{min}$ of each target in $C^1$ (b) $s_{min}$ of each target in $C^2$

Figure 2-20: The variation of $s_{min}$ of each object as a function of time.

simulation. Thus, 22 pixels are used when the distance between an object and a camera is closer than $3m$ and 12 pixels for greater distances with TABLE 2.1. The line "$\sigma_h + \sigma_s$ only" is the smallest $s_{min}$ that can be achieved by estimating the object height and compensating for the synchronization issue. However, it is not necessary to decrease $s_{min}$ smaller than the detected box size. Thus, the system uses the line "$max(r_b, \sigma_h + \sigma_s)$" for the association process. The line "initial value of $\sigma_h + \sigma_s$" is a radius that the constant radius based method uses for the entire time.

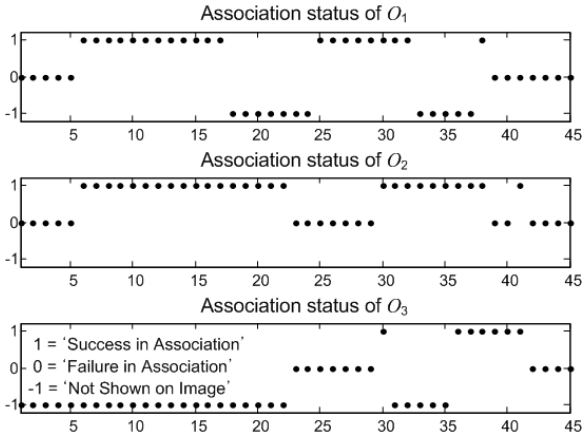
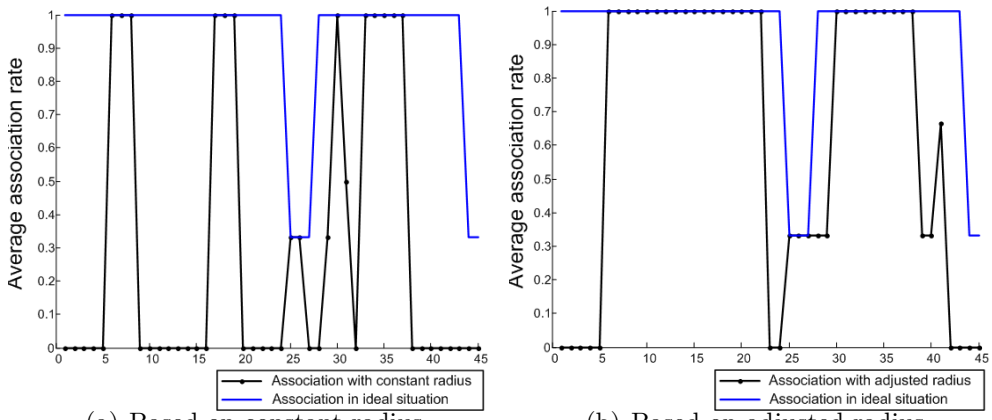



Figure 2-21: Improvement over Fig. 2-16 by adjusted radius based association algorithm.

Fig. 2-21 shows the association status of each object when the adjusted radius based algorithm is used. The figure shows a prominent association improvement as compared with Fig. 2-16. A new object (i.e., object $O_3$) or a failed object is also associated or recovered as time elapses.




(a) Based on constant radius (b) Based on adjusted radius

Figure 2-22: Simulation result of average association performance with constant radius based association algorithm and adjusted radius based association algorithm.

Fig. 2-22 compares the average association performance between the constant radius based association algorithm and the adjusted radius based association algorithm. An ideal situation indicates that heights of targets are known and cameras are synchronized. Although the proposed association algorithm outperforms the constant radius based association algorithm, the result indicates that homographic lines based association is not always the best solution. A system utilizes rather local tracking information to maintain association information when homographic lines cannot be guaranteed to distinctively cross targets.

## 2.5 Summary

In this chapter, we showed a novel association method with local initiated homographic line for surveillance application with multiple cameras. This method is based on the parallel projection model to support the camera movement. We investigate the plausible parameters to affect the association performance. This information is used to define the threshold distance to indicate the effectiveness of the locally generated homographic line. We also discuss the strategy to improve the association performance using the temporal and spatial redundancies.

# Chapter 3

# Multiple Camera Collaboration Strategies for Dynamic Object Association

## 3.1 Introduction

In order to dynamically establish the association for objects, the previous chapter presents an association method that homographic lines are locally generated on targets in each camera and they are projected to among the other cameras. Since it is not necessary to have a ground plane as a common reference plane, all the cameras do not need to see the ground plane. Homographic lines are generated when the degree of separation between them is satisfied. The required minimum separation between each pair of cameras is predetermined by incorporating the effect of targets height uncertainty and frame synchronization errors because the reference plane may

not be the same as the actual height of targets. The method can be extended to support multiple cameras through pair-wise collaboration for the object association and combine the association information from each pair of collaborating cameras. While the pair-wise collaboration is effective for objects with the enough separation, the association is not well-established for objects without the enough separation and it may generate the false association. Therefore, an effective camera collaboration is necessary to reduce inconsistent and uncertain information.

In this chapter, we extend the locally initiating homographic lines based association method to two different multiple camera collaboration strategies that reduce the false association. Collaboration matrices are defined with the elements of the required minimum separation presented in [23]. The first strategy compares the collaboration matrices with the minimum separation of objects for each pair of cameras and selects the best pair out of many cameras satisfying the required minimum separation. After targets are associated in selected cameras, the association information is propagated to unselected cameras by transforming the global information constructed from the associated targets. The selection based strategy efficiently collaborates on the object association with the best pair of the cameras as reducing the false association. However, it requires the long operation time to increase the association rate due to unsatisfied separation when a large number of targets are detected. In order to shorten the operation time for the high association rate, the second strategy initiates the collaboration for all the pairing cases of cameras regardless of the separation. When each pair of cameras collaborates on object association, a homographic line is generated on each target and it is projected to the other collaborating camera. The

other camera generates homographic lines on only the crossed targets by the projected homographic lines and they are re-projected to the one camera. This association process is iteratively operated for all the unassociated targets in all the pairing cases of cameras. The proposed methods are evaluated with real video sequences and they are compared with the basic pair-wise collaboration to demonstrate the effective and efficient association.

The remainder of this chapter has 4 sections. In Section 3.2, we present the overview of homographic lines based association method and describe the association problem in terms of the false association and the computational costs. Section 3.3 investigates two collaboration strategies for objects association to minimize the inconsistency in the system and to improve the efficiency of using homographic lines. In Section 3.4, we verify the proposed methods with the real video sequences. Finally, our contribution is summarized in Section 3.5.

## 3.2 Problem Description

### 3.2.1 Problem Description and Approach

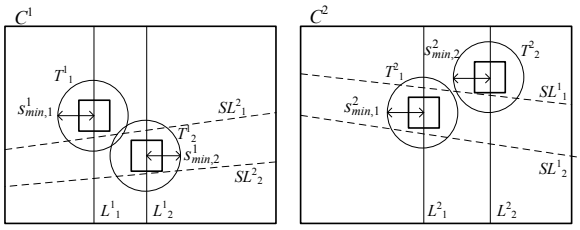



Figure 3-1: Illustration of an unsuccessful association of detected targets due to insufficient separation in a non-ideal situation.

The successful objects association depends on the separation between homographic lines on the other cameras. When the sufficient separation between homographic lines is not guaranteed, homographic lines can cross multiple targets and intersections with multiple targets can create ambiguity in determining the correspondence of objects. Moreover, detection uncertainty as well as lack of common reference may create uncertainty in deciding the intersections. In order to guarantee the correct intersections of homographic lines with corresponding targets, a tolerance circle, $s_{min}$, is defined for each target by considering the effect of targets height uncertainty, frame synchronization errors and detection uncertainty as shown in Fig. 3-1. Because the required separation for objects association increases, the association performance is affected by the tolerance circle of targets.

When many cameras (i.e. more than two cameras) are involved in the object association, the locally generated homographic lines based association method can be extended to support them through pair-wise collaboration. However, each pair of collaborating cameras may contradict the object association due to the insufficient separation. It may generate false associations and create the inconsistent information of uncertain association in the system. They are also propagated in time and continuously degrade the association performance. Moreover, when multiple cameras collaborate on objects association without the sufficient separation between targets, homographic lines are unnecessarily generated on targets and projected to the other cameras without establishing objects association. The association failure due to the insufficient separation wastes the computational costs of transforming homographic lines. In general, each target requires transformation of a local homographic line to

a global homographic line or transformation of a global homographic line to a local homographic line. The number of the intersection tests is proportional to the multiplication of the number of targets and the number of transformed homographic lines. It is assumed that a pair-wise association is utilized when more than two cameras are used. Then, the computational costs for using homographic lines are defined by the number of transformations and intersection tests, $C_T$ and $C_C$, respectively

$$C_T = 2I \times 2(K-1),$$
$$C_C = I^2 \times 2(K-1), \tag{3.1}$$

where $I$ denotes the number of commonly detected targets in cameras and $K$ denotes the number of cameras. Fig. 3-2 shows the computational costs for using homographic lines according to the number of targets and the number of cameras. When the number of targets is greater than three, the computational costs become extremely high. Thus, a proper collaboration camera strategy is necessary to efficiently associate targets among different cameras as minimizing the inconsistent and uncertain information.

We consider two effective collaboration strategies to reduce false associations as well as to improve the efficiency of using homographic lines. Collaboration matrices are defined to indicate the feasibility of successful association between any two cameras. The elements in the collaboration matrices represent the required minimum separation obtained by incorporating the effect of targets height uncertainty,

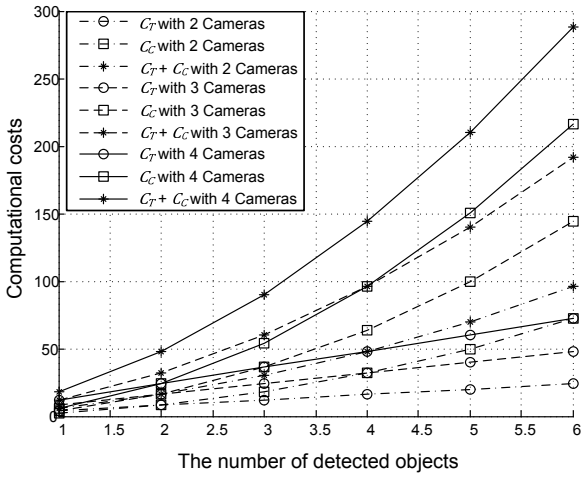



Figure 3-2: Illustration of the computational costs for using homographic lines according to the number of targets and cameras.

frame synchronization errors and detection uncertainty. The first strategy uses the collaboration matrices to select the best pair out of many cameras by using the degree of separation between homographic lines. After targets in the selected cameras are associated, the association information is propagated to unselected cameras by transforming the global information constructed from the associated targets.

However, when a large number of objects is detected, the selected cameras cannot cover all the targets and the threshold is hardly satisfied due to targets overlapping and occlusion. The second strategy initiates the collaboration process of objects association for all the paring cases of cameras regardless of the separation. In each pair of cameras, a homographic line is generated on each target and it is projected to the camera. Then, the other collaborating cameras generate homographic lines on only the crossed targets by the projected homographic lines and they are re-projected to the one camera. Since the system tests the association for each target at a time, it minimizes association ambiguity caused by homographic lines generated from all

the targets. This association process is iteratively operated for all the unassociated targets in all the paring cases of cameras.

## 3.3 Multiple Camera Collaboration

### 3.3.1 Collaboration Matrices and Characterization

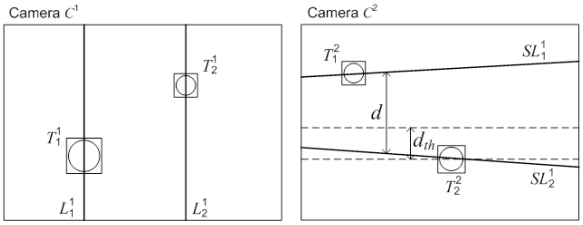



Figure 3-3: The required separation of homographic lines to be effective for objects association in the other collaborating camera with the known reference plane such as a ground plane.

Two collaborating cameras participate in an association process by generating a homographic line on each target in each camera. The homographic line is transformed to a global reference plane such as a ground plane and its transformed homographic line is projected to the other collaborating camera. While the system knows each target that generates its homographic line in a local camera, its corresponding target in the other collaborating camera is determined by the intersection with its projected homographic line. The association of the corresponding targets is established when a projected homographic lines intersects a corresponding target in each camera. However, the association is not established if a projected homographic line intersects multiple targets in each camera. Thus, the separation of projected homographic lines is a key parameter determining the successful association and the size of a target

determines the required separation of projected homographic lines. For example, homographic lines are generated on each target in camera $C^1$ in Fig. 3-3 and the distance between projected homographic lines is denoted by $d$ in camera $C^2$. In order for projected homographic lines $SL_1^1$ and $SL_2^1$ to be effective in camera $C^2$, the separation of them should be greater than the twice size of targets to be associated. We denote the required minimum separation of homographic lines as a threshold and it is represented by $d_{th}$. Since the threshold can be used to indicate the effectiveness of projected homographic lines, the determination of an appropriate threshold is critical in homographic lines based association.

The critical issue in determining the threshold is that the separation of projected homographic lines depends on the location and the orientation of cameras. Also, if cameras are flexibly titling and panning, the threshold should incorporate the effect of tilting and panning on the separation. When all the possible variations for a camera configuration are considered, the threshold needs to be determined for each case of a camera configuration. Since it is not trivial to construct all the thresholds according to all the camera configurations, we consider only the worst effect among them on the separation. The threshold is determined for each pair of cameras by measuring the maximum size of targets and the threshold matrix with the threshold for each pair of cameras is represented by

$$D_{th} = \begin{bmatrix} d_{th}^{1,1} & d_{th}^{1,2} & \dots \\ d_{th}^{2,1} & d_{th}^{2,2} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix},$$

where its size is $K \times K$ and $d_{th}^{k,l}$ denotes the threshold between camera $C^k$ and $C^l$. A negative value indicates that the threshold is always satisfied in a local camera because a system knows which targets generate homographic lines.

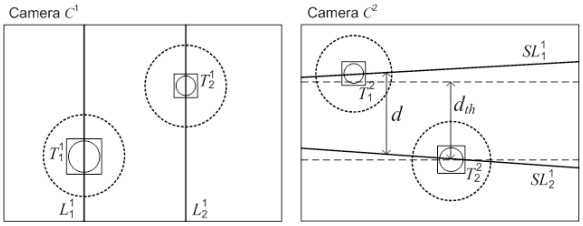



Figure 3-4: The required separation of homographic lines to be effective for objects association in the other collaborating camera with the unknown heights of objects and the frame synchronization errors between cameras.

There are two additional factors influencing the threshold which increases the size of a target. One is the unknown heights of objects and another is the frame synchronization errors between cameras in a real situation. The dotted circle of targets represents the tolerance circle incorporating the effect of them as shown in Fig. 3-4. Due to the effect of the additional factors, $d$ is increased and may be smaller than $d_{th}$ in the figure. The size of the tolerance circle is determined by measuring the amount of pixels that can be deviated from a corresponding target by the additional factors.

For the unknown heights of objects, the size of a tolerance circle is defined by

$$s^k_{min,i} = \max(r^k_{b,i}, \sigma^k_{h,i}), \tag{3.2}$$

where $r^k_{b,i}$ denotes the original size of a target and $\sigma^k_{h,i}$ denotes the possible number of deviated pixels from the centroid of a target by the mismatched height of a target in camera $C^k$. $\sigma^k_{h,i}$ depends on camera configurations (locations, tilting angles and panning angles) generating homographic lines. Then, $\sigma^k_{h,i}$ is represented by

$$\sigma^k_{h,i} = \max_{l \neq k}(\acute{\sigma}^{k,l}_{h,i}), \tag{3.3}$$

where $\acute{\sigma}^{k,l}_{h,i}$ denotes the maximum number of deviated pixels from the centroid of a target when homographic lines are generated from camera $C^l$ to $C^k$ with possible camera configurations. We use the reference plane with the average height of targets to transform and project homographic lines to a different camera since the actual heights of targets are unknown. When a homographic line is transformed to the reference plane and its transformed homographic line is projected to a different camera, a projected homographic line is deviated from the point of a target with the actual height. In order to measure the amount of pixels to include the deviation by the height uncertainty, it is assumed that only the height range of targets is given to the system. The amount of pixels is measured by comparing a projected homographic line with the average height with a projected homographic line with the maximum or the minimum height because the heights of targets are unknown.

Each value of $\acute{\sigma}^{k,l}_{h,i}$ between cameras $C^k$ and $C^l$ is determined by finding the

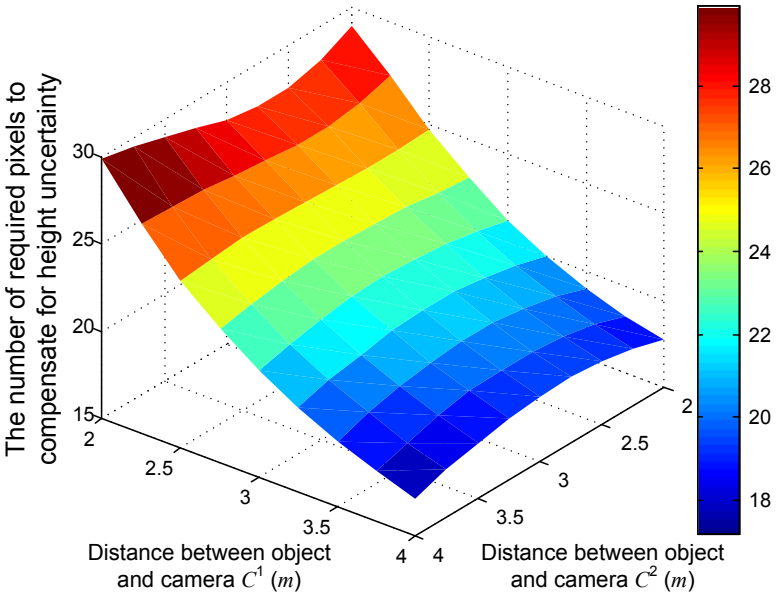



Figure 3-5: Illustration of the effect of height uncertainty between cameras $C^1$ and $C^2$ according to the distance to an object.

maximum effect of the given amount of height uncertainty according to object locations. Fig. 3-5 shows an example how many pixels are deviated from an original point between cameras $C^1$ placed at $(x = 3m, y = 0m, z = 3m)$ and $C^2$ placed at $(x = 6m, y = 3m, z = 3m)$. In order to measure the number of deviated pixels, two homographic lines are generated from camera $C^2$ to camera $C^1$ according to an object's location. One homographic lines is generated with an actual height and the other is generated with an average height different from the actual height by $0.1m$. The number of deviated pixels is maximized when an object is close to a camera onto which homographic lines are transformed. The simulation to measure the amount of pixels is repeated for different camera configurations and the maximum value among them is selected. Fig. 3-6 illustrates the amount of pixels with other cameras incorporating the effect of height uncertainty $(0.1m \sim 0.4m)$ where $\sigma_h^{k,l}$ denotes the

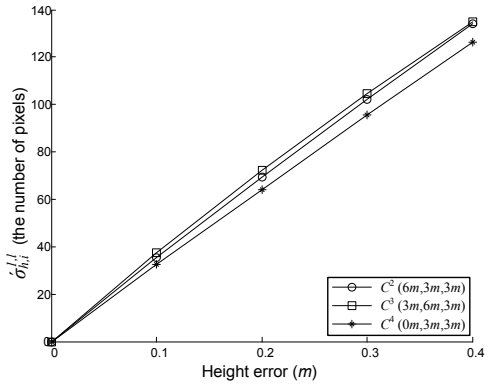



Figure 3-6: Illustration of the number of pixels required to compensate for the effect of height uncertainty in camera $C^1$ placed at $(x = 3m, y = 0m, z = 3m)$ with other cameras (Image size : $704 \times 480$).

amount of pixels. They are proportional to the amount of height uncertainty.

Another factor that influences the size of the tolerance circle is frame synchronization errors between cameras. It also causes the deviation of a projected homographic line because targets can be different locations due to differently captured time. It is assumed that the differently captured time of cameras is within at most 1 frame. Fig. 2-6 shows the example of the frame synchronization errors. Solid circles and dotted circles represent two different locations where objects are detected by cameras. Object $O_2$ cannot be associated because $SL_2^1$ does not intersect with $T_2^2$. In order to incorporate frame synchronization errors, the radius of a tolerance circle needs to be adjusted by

$$s^k_{min,i} = \max(r^k_{b,i}, \sigma^k_{s,i}), \tag{3.4}$$

where $\sigma^k_{s,i}$ denotes the possible number of deviated pixels from the centroid of a target by the synchronization issue between cameras in camera $C^k$. The synchronization

effect depends on the sampling period of a camera $T_F$ and the direction toward which an object moves. Then, $\sigma_{s,i}^{k}$ is obtained by

$$\sigma_{s,i}^{k} = \max_{l \neq k}(\acute{\sigma}_{s,i}^{k,l}), \tag{3.5}$$

where $\acute{\sigma}_{s,i}^{k,l}$ denotes the maximum number of deviated pixels from the centroid of a target by the synchronization issue between cameras $C^k$ and $C^l$. Since the effect is maximized when the optical axes of two cameras are perpendicular to each other, only paired cases of perpendicular cameras can be considered.

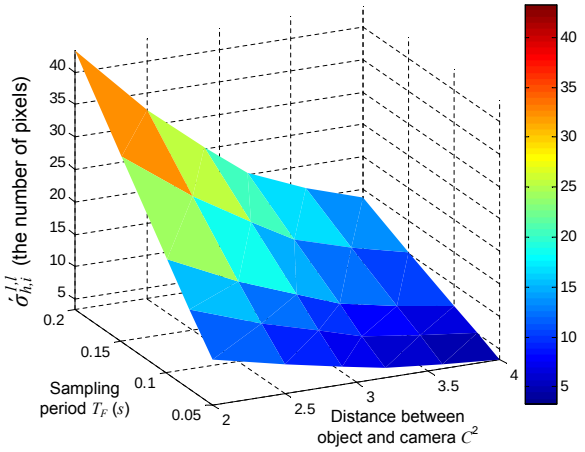



Figure 3-7: Illustration of the number of pixels required to compensate for the synchronization effect according to sampling periods and distance between an object and a camera.

Fig. 3-7 shows the amount of pixels to incorporate the effect of frame synchronization errors where $\acute{\sigma}_{s,i}^{k,l}$ denotes the amount of pixels. It is noted that the deviation of a projected homographic line is related to the relative speed of an object per each frame, not the absolute speed of an object. When the deviation of a projected homographic line is estimated by the simulation, the speed of an object is set to $2m/sec$.

Since the frame rate varies from $0.05sec$ to $0.2sec$ in the simulation, it has the same effect of having the relative speed of an object, $0.1m/frame$ to $0.4m/frame$. Since the frame synchronization errors are maximized with perpendicularly placed cameras, a homographic line is generated from the delayed image of $C^1$ to the image of $C^2$. The maximum pixel distance error between a projected homographic line and a corresponding target in camera $C^2$ is measured. As the sampling rate increases, the amount of pixels decreases.

When the effects of the additional factors are considered at the same time, they can compensate for each other. For example, when a homographic line is generated from the deviated position of a target by detection algorithm, a transformed homographic line with an average height of a target can be accidentally shifted to the position with an actual height of a target. A similar effect can also occur with synchronization issues. However, the system cannot predict the compensation effect by non-ideal parameters. Thus, the size of a tolerance circle should consider the worst effect by

$$s^k_{min,i} = \max(r^k_{b,i}, \sigma^k_{h,i} + \sigma^k_{s,i}). \tag{3.6}$$

Fig. 3-8 illustrates $s_{min}$ in terms of the number of required pixels by (3.6) assuming that the height uncertainty is set to be $0.1m$. If the effect of detection performance is considered, the values are expected to be increased. While expanded radii of association circles guarantee that targets are crossed by corresponding transformed homographic lines, they can degenerate association performance because homographic lines generation can be ineffective due to insufficient separations of targets. A threshold

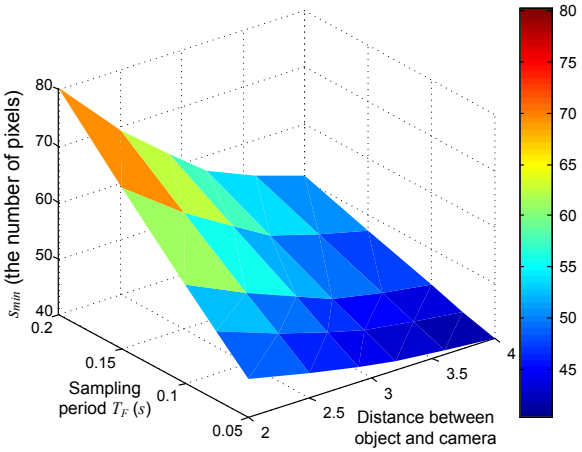



Figure 3-8: Illustration of the number of deviated pixels by (3.6).

indicating effectiveness of homographic lines can be represented by radii of association circles. A threshold $d_{th}^k$ in camera $C^k$ is defined as

$$d_{th}^k = arg\min_{i,j}(s_{min,i}^k + s_{min,j}^k), \tag{3.7}$$

where $i, j$ denote indices of neighboring targets. The smallest sum of radii of two neighboring association circles indicates effectiveness of homographic lines in an association process in camera $C^k$.

### 3.3.2 Camera Selection Based Approach

A camera selection based approach is to select a pair of the cameras to increase the effectiveness of projected homographic lines. Since the effectiveness of projected homographic lines depends on the threshold in the other collaborating camera, a system tests the separation of homographic lines for each pair of cameras. Each camera determines the shortest distance between neighboring homographic lines to be tested

for the effectiveness in the other collaborating camera. If the transformed shortest distance has enough separation in the other collaborating camera, the separations of other homographic lines are also satisfied since the transformation of a homographic line is a linear process.

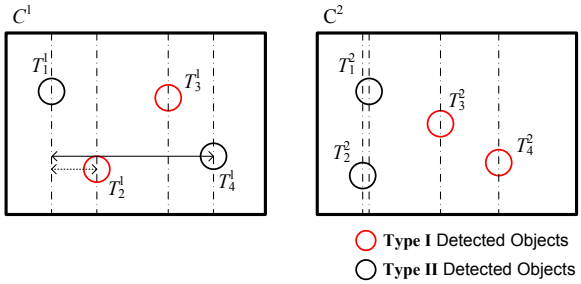



Figure 3-9: Illustration of grouping targets by using the local tracking and association information.

In order to reduce the dependence of the shortest distance on the separation of targets, targets are grouped into two types as shown in Fig. 3-9. Type I is a newly detected target and Type II is a locally tracked target having the association. Each of the shortest distances is determined for each type of targets. If they are not grouped, the shortest between targets in camera $C^1$ is the distance between targets $T_1^1$ and $T_2^1$ otherwise, the distance increases to the distance between targets $T_1^1$ and $T_4^1$ for Type I targets. $G^k$ denotes a set of Type I targets and $\widetilde{G}^k$ denotes a set of Type II targets in camera $C^k$. Then, sets for targets are represented by, in Fig. 3-9,

$$G^1 = \{T_1^1, T_4^1\}, \widetilde{G}^1 = \{T_2^1, T_2^1\},$$

$$G^2 = \{T_1^2, T_4^2\}, \widetilde{G}^2 = \{T_2^2, T_3^2\}.$$

The association processes are operated on sets of equivalent types by using the ho-

mographic line based association.

$$H^{1,2}(G^1, G^2), \widetilde{H}^{1,2}(\widetilde{G}^1, \widetilde{G}^2),$$

where function $H^{i,j}$ is the homographic line based association for Type I targets between camera $i$ and $j$ and function $\widetilde{H}^{i,j}$ for Type II targets. $H$ and $\widetilde{H}$ are equivalent but the type of targets are different. The target grouping also decreases the number of crossing tests by projected homographic lines. When the targets are not grouped in this example, $C_T + C_C$ is $16+32 = 48$. Otherwise, $C_C$ decreases to 16 with grouped targets and $C_T + C_C$ becomes 32.

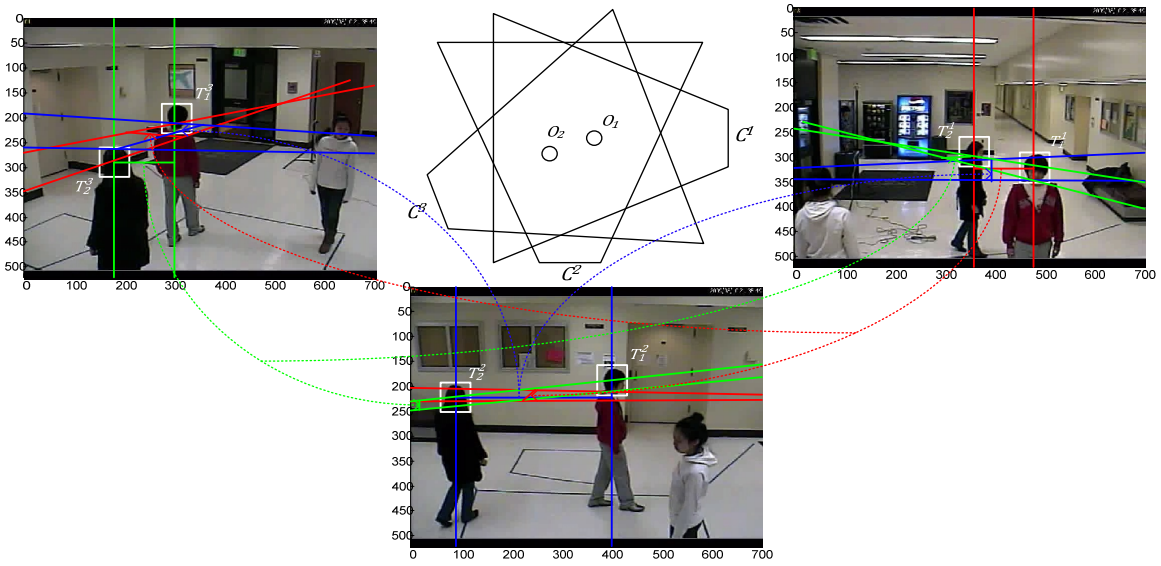



Figure 3-10: Illustration of the camera selection strategy in multiple cameras with more than two objects (red homographic lines from $C^1$, blue homographic lines from $C^2$, and green homographic lines from $C^3$).

When the shortest distance between neighboring homographic lines is determined in each camera, it is represented as a starting point and an ending point to be projected to the other cameras. It is assumed that an image has the top left ori-

gin. The x-coordinates of the starting point and the ending point are the same as the x-coordinates of the targets respectively and the y-coordinates are set to be the greater of y-coordinates of targets. The reason for selecting the greater of the two y-coordinates is that a closer line to a camera has a shorter length when it is transformed and the shorter one should be tested for the separation in the other cameras. For example, the coordinates of two targets are (482, 323) and (363, 289) in Fig. 3-10. The shortest distance between them consists of two points (363, 482) and (482, 323). $d_{i,j}^k$ denotes the length of the shortest line between neighboring targets $T_i^k$ and $T_j^k$, and $T_{i,j}^{k,l}$ denotes the transformed length of $d_{i,j}^k$ at camera $C^l$. The length of the shortest line for each pair of cameras can be represented by a matrix with the size of $K \times K$ for convenience. In this example, the distance matrix $D$ is obtained by

$$D = \begin{bmatrix} 119 & 32.177 & 112.8608 \\ 43.0299 & 314 & 179.2715 \\ 59.1755 & 16.9171 & 121 \end{bmatrix}.$$

The distance values on diagonals of this matrix are the pixel distance of two targets in each local camera and others are projected pixel distances in the other cameras. If only one object is detected by a camera, distance is $\infty$. Since this matrix is determined by the coordinates of targets, it is updated every frame.

The threshold matrix $D_{th}$ for Fig. 3-10 is obtained by by using the simulated data from Fig. 3-6 and Fig. 3-7

$$D_{th} = \begin{bmatrix} -1 & 74.46 & 87.23 \\ 76.61 & -1 & 64.27 \\ 82.70 & 74.46 & -1 \end{bmatrix},$$

where the height error is set to $0.1m$ and the frame rate is set to $8 frames/sec$. When matrices $D$ and $D_{th}$ are compared, a pair of $C^1$ and $C^3$ and a pair of $C^2$ and $C^3$ are possibly selected to cooperate for association. If multiple choices are possible, a pair of cameras having the maximum difference between matrices $D$ and $D_{th}$ can be chosen.

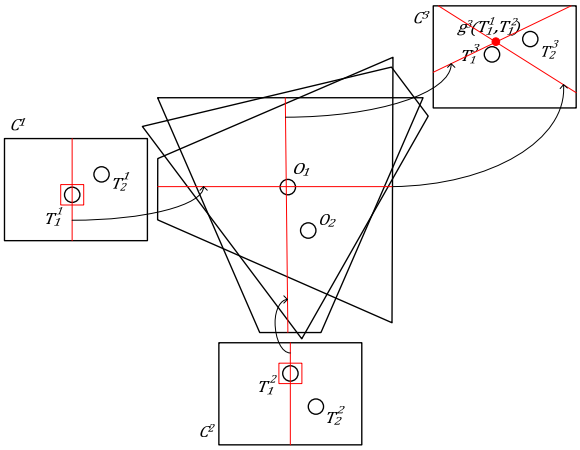



Figure 3-11: Illustration of an association information update in unselected cameras by using global information.

Targets in unselected cameras are associated by global information constructed by the associated targets in selected cameras. When targets are associated in two selected cameras, the system constructs global information such as height and position by a multi-camera localization scheme. Fig. 3-11 illustrates an association information update by selected cameras. When homographic lines generated with the known

height from associated targets are transformed to other cameras, they should intersect at near another corresponding target. $g^{\widetilde{k}}(T_{i_1}^{k_1}, T_{i_2}^{k_2})$ denotes an intersection point of homographic lines on camera $C^{\widetilde{k}}$ from targets $T_{i_2}^{k_1}$ and $T_{i_3}^{k_2}$ where $k_l \in K$ with the positive integer $l$. If several unassociated targets exist in unselected cameras, a target having the minimum distance to the transformed point is associated with $\{T_{i_1}^{k_1}, T_{i_2}^{k_2}\}$. Then, index $\widetilde{i}$ of unassociated targets to be associated with them is determined by,

$$\arg\min_{\widetilde{i}} D(g^{\widetilde{k}}(T_{i_1}^{k_1}, T_{i_2}^{k_2}), T_{\widetilde{i}}^{\widetilde{k}}), \tag{3.8}$$

where $D(a, b)$ returns the distance between points $a$ and $b$. This may falsely associate targets when they are occluded by each other. Hence, if an intersection point is satisfied with $s_{min}$ of more than two targets, association information is not updated to prevent from false association. Even with these strategies, false association is still possible when targets are occluded each other. However, their association can be confirmed after they are separated enough for the association. For the computational costs, this process requires one global to local transformation and the crossing tests by the number of targets in (3.8). The camera selection based approach is summarized in Algorithm 2.

### 3.3.3 Iteration Based Approach

The association performance of the camera selection based method is ineffective due to a large number of targets. Any elements of matrix $D$ may not be satisfied with the threshold matrix $D_{th}$. A distance matrix for Fig. 3-12 is obtained by

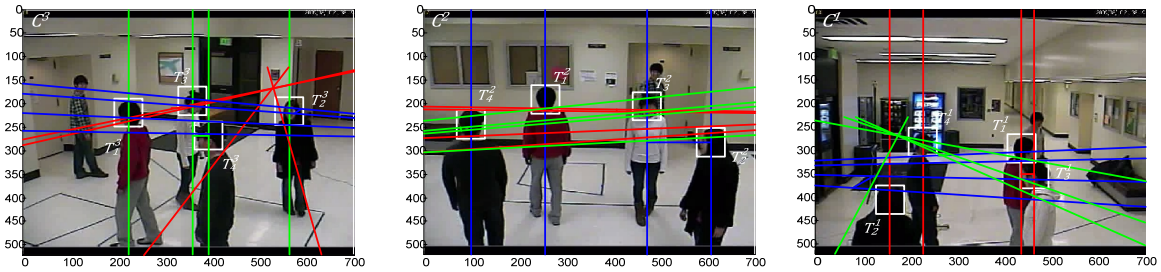


Figure 3-12: Illustration of a case in which any pairs of cameras do not satisfy thresholds (red homographic lines from $C^1$, blue homographic lines from $C^2$, and green homographic lines from $C^3$).

$$D = \begin{bmatrix} 27 & 3.90 & 17.98 \\ 47.52 & 136 & 20.14 \\ 20.71 & 6.64 & 34 \end{bmatrix}.$$

The figure shows the ineffectiveness of homographic lines for associating targets when any pairs of cameras do not satisfy the threshold matrix $D_{th}$. Thus, the system needs to wait until thresholds are satisfied. This may cause undetermined delay for the association process.

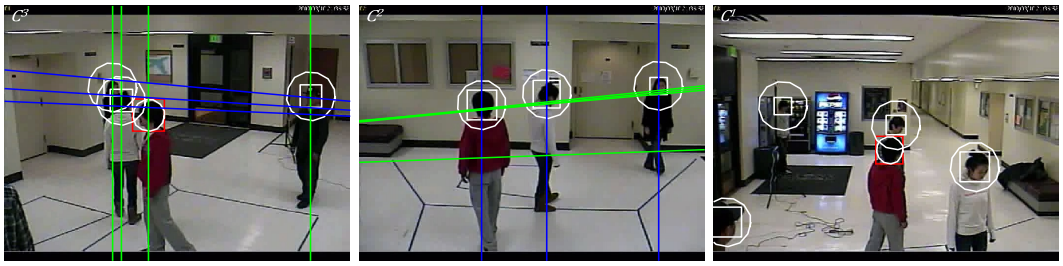


Figure 3-13: A limitation of camera selection strategy (blue homographic lines from $C^2$, and green homographic lines from $C^3$).

Moreover, the camera selection based method does not always provide a correct

decision in selecting cameras because the average height of targets is used to transform and project a horizontal line to other cameras. Also, since projected homographic lines are not parallel in other cameras, the association process is not always successful. These limitations waste projected homographic lines without establishing objects association. For example, although cameras $C^2$ and $C^3$ are selected by the satisfied threshold in Fig. 3-13, any targets are not associated due to the insufficient separation. Another limitation of the camera selection based method is that some pairs of cameras having possible collaboration are disregarded.

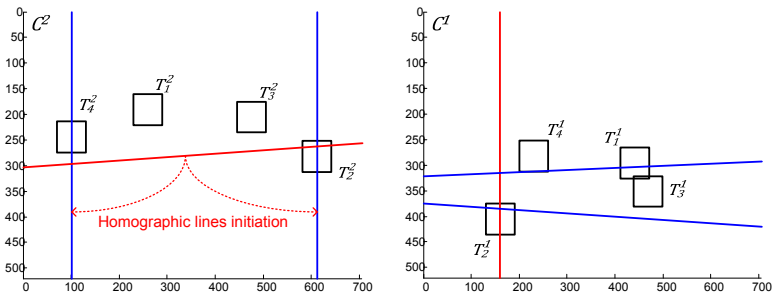
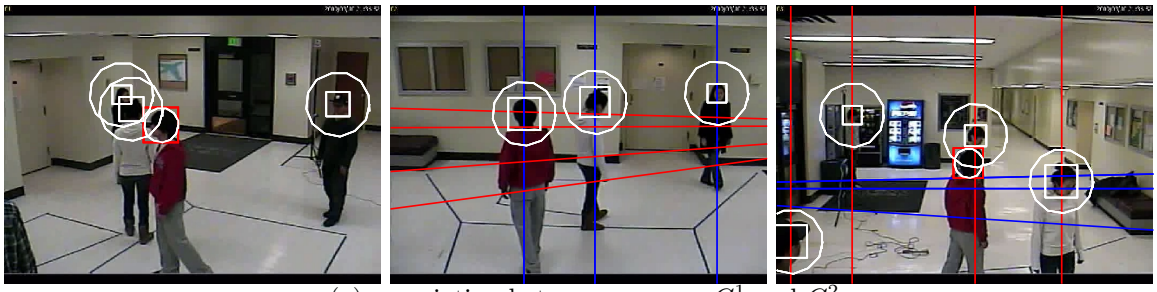



Figure 3-14: The illustration of an iterative association method.

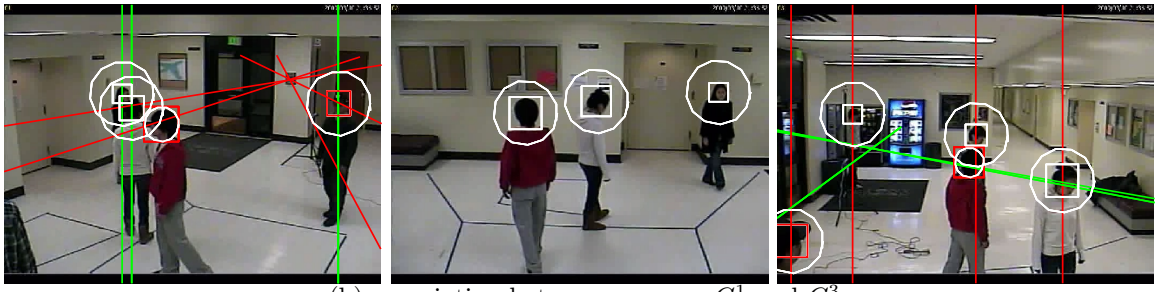
In order to remedy the limitations of the camera selection based method, a system initiates the collaboration process of objects association for all the pairing cases of cameras regardless of the separation. Similarly to the camera selection based approach, the targets in each camera are grouped by the two types defined in the previous section to minimize the unnecessary association process for the different types of targets. When each pair of cameras is operated for objects association, a homographic line is generated on each target at a time and it is transformed to the other collaborating camera. A homographic line can be generated in any order of targets such as the left to the right on image. After a homographic line is transformed to the

other collaborating camera, all the crossed targets by the transformed homographic line generate homographic lines. The association is established for only the targets generating homographic lines. Fig. 3-14 illustrates an example of a homographic line generation on one target for two cameras. Camera $C^1$ generates a homographic line on only target $T_2^1$ first. Its homographic line is projected to camera $C^2$ and camera $C^2$ generates each homographic line on the intersected targets $T_4^2$ and $T_2^2$. Since only the two of the four targets participate in an association process for target $T_2^1$, it has a higher chance that the projected homographic lines have the sufficient separation in camera $C^1$. Table 3.1 illustrates the crossed targets by homographic lines of an iteration based strategy for Fig. 3-14. As a result, targets $T_2^1$ and $T_2^2$ are associated. This process is repeated for each target in each pair of collaborating cameras. The iteration based approach is summarized in Algorithm 3.

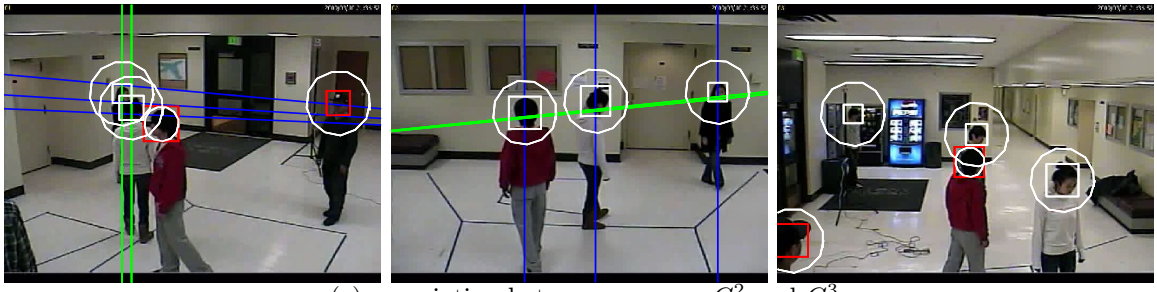
Unassociated targets by the camera selection based approach in Fig. 3-13 are associated by the iterative association approach in Fig. 3-15. Since three cameras are used, six iterated association cases exist ($C^1$ to $C^2$, $C^1$ to $C^3$, $C^2$ to $C^3$, and vice vera). One additional association for targets is established in Fig. 3-15(b). Although the iterative association approach has an advantage to check every association case, the computational costs of the transformation and the crossing test are much higher than those of the camera selection based association.



(a) association between cameras $C^1$ and $C^2$



(b) association between cameras $C^1$ and $C^3$



(c) association between cameras $C^2$ and $C^3$

Figure 3-15: Illustration of the multiple objects association by the iterative association approach (red homographic lines from $C^1$, blue homographic lines from $C^2$, and green homographic lines from $C^3$).

## 3.4 Simulation and Analysis

### 3.4.1 Simulation Setup

Fig. 3-16 illustrates a simulation setup with six objects and three cameras for analyzing the association performance and the computational costs of the transformation and the crossing test. We also compare the proposed methods with basic pair-wise collaboration extended from [23] to prove the improvement on the false association

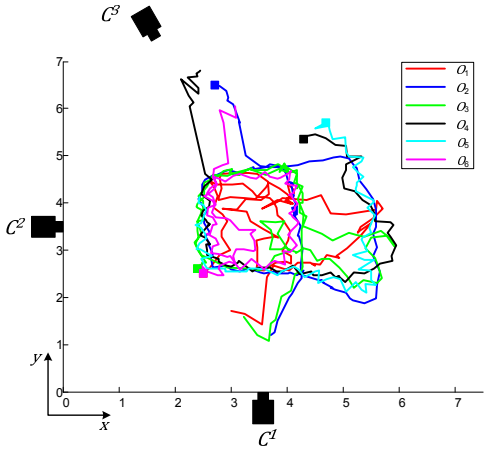
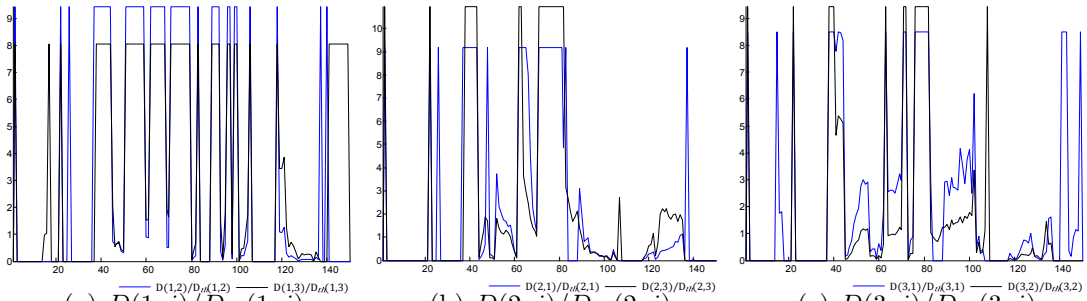



Figure 3-16: Illustration of a simulation setup with six objects trajectories and three cameras.

rate. Because the basic pair-wise approach initiates the association process for each pair of participating cameras, three different pairs of cameras execute the association process redundantly. In order to compare the performance between the methods, we measure three different rates such as successful association rate, failed association rate, and false association rate to check the inconsistency. Camera $C^1$ is placed at $(x = 3.65m, y = 0m, z = 2.37m)$ with tilting angle 82.3°and panning angle 0°, camera $C^2$ is placed at $(x = 0m, y = 3.5m, z = 2.45m)$ with tilting angle 76°and panning angle 90°, and camera $C^3$ is placed at $(x = 1.83m, y = 7.32m, z = 2.37m)$ with tilting angle 78°and panning angle 156°. The total number of frames is 150 and the average height of targets is $1.7m$. Another important issue is how to locally track targets in each camera. When targets (i.e., faces) are occluded each other by three fourths of their size, they fail in the local tracking and also lose association information. Since Type II targets are already associated, only Type I targets are considered for the association in the simulation.

### 3.4.2 Association Performance and Complexity Comparison




(a) $D(1,j)/D_{th}(1,j)$ (b) $D(2,j)/D_{th}(2,j)$ (c) $D(3,j)/D_{th}(3,j)$

Figure 3-17: Simulation results of $D(i,j)/D_{th}(i,j)$ to show possible cameras for the cooperation at each association time.

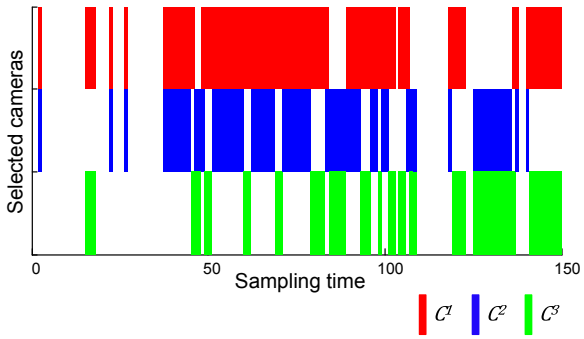
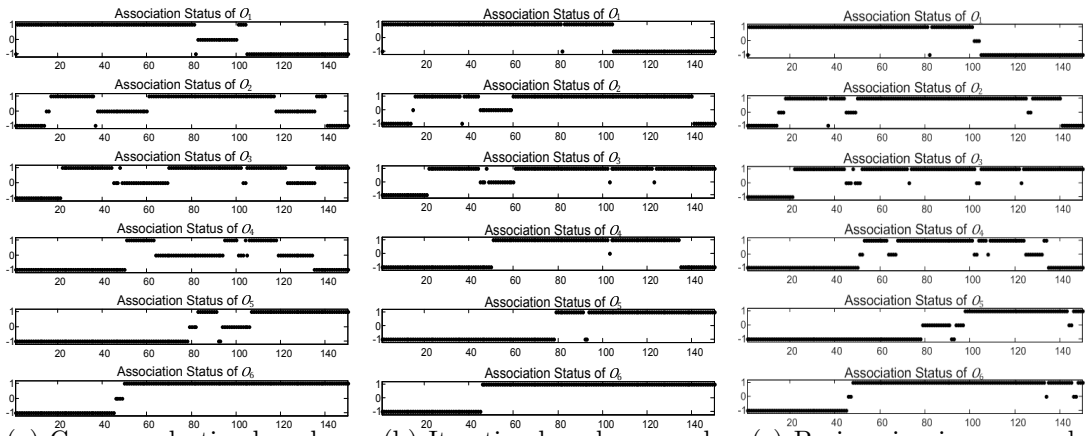



Figure 3-18: Selected cameras by the proposed camera selection method at each association time.

The first collaboration strategy selects a pair of cameras satisfying the threshold. The values for $D_{th}$ are the same as the values used in Section 3.3. The shortest distance between homographic lines in each camera is determined and its projected distance is compared with the corresponding element of $D_{th}$ at every frame. Fig. 3-17 shows the variation of each element of distance matrix $D$ with respect to a corresponding element of threshold matrix $D_{th}$ for the camera selection based approach. If the value of $D(i,j)/D_{th}(i,j)$ is greater than 1, the transformed minimum distance is satisfied with the threshold for a pair of corresponding cameras $C^i$ and $C^j$. When

one unassociated target is remained in each camera, the distance between targets is set to be the width of image instead of the infinity for the simulation. Fig. 3-18 shows the result of selected cameras at each sampling time with the information of Fig. 3-17. If any thresholds are not satisfied, none of cameras is selected.




(a) Camera selection based approach (b) Iteration based approach (c) Basic pair-wise approach

Figure 3-19: The objects association status comparison of the camera selection based approach and the iteration based approach with the basic pair-wise approach at each association time ('$1'$ indicates successful association, '$0'$ failed association, and '$-1'$ that object is shown on only one camera).

| $O_i$ | No asso. (%) | Camera selection based approach | | | Iteration based approach | | | Basic pair-wise approach | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Success (%) | Failure (%) | False (%) | Success (%) | Failure (%) | False (%) | Success (%) | Failure (%) | False (%) |
| $O_1$ | 32.0 | 56.0 | 12.0 | 0.0 | 68.0 | 0.0 | 0.0 | 66.0 | 2.0 | 6.0 |
| $O_2$ | 16.7 | 54.7 | 28.6 | 0.7 | 71.3 | 12.0 | 0.7 | 76.7 | 6.6 | 7.3 |
| $O_3$ | 14.0 | 60.0 | 26.0 | 0.7 | 73.3 | 12.7 | 0.0 | 79.3 | 6.7 | 2.7 |
| $O_4$ | 44.0 | 22.0 | 34.0 | 0.0 | 51.3 | 4.7 | 0.0 | 44.7 | 11.3 | 2.7 |
| $O_5$ | 53.3 | 35.4 | 11.3 | 0.0 | 46.7 | 0.0 | 0.7 | 34.0 | 12.7 | 2.7 |
| $O_6$ | 30.0 | 67.3 | 2.7 | 0.0 | 70.0 | 0.0 | 0.0 | 66.7 | 3.3 | 4.0 |
| Avg. | 31.7 | 49.2 | 19.1 | 0.2 | 63.4 | 4.9 | 0.2 | 61.2 | 7.1 | 4.2 |

Figure 3-20: Performance comparison for Fig. 3-19.

Fig. 3-19 and Fig. 3-20 show the comparison of the proposed approaches with

the basic pair-wise approach. In Fig. 3-20, "no asso." case means that objects are detected on only one single camera, and the collaboration for the object association is not required for multiple cameras. The basic pair-wise association approach may generate false associations because of the insufficient separation of homographic lines. The proposed approaches reduce the inconsistency of false association as compared with the basic association approach. When the iteration based approach is used for multiple objects association, the number of successful association is greater than that of the camera selection based approach. This is mainly because the camera selection based approach usually selects only a pair of cameras satisfying the threshold and disregards the rest of possible association cases. On the other hand, the iteration based approach initiates the association process for each target and it leads to improve the association performance. Thus, the camera selection based approach requires the longer operation time to increase the association rate than the iteration based approach.

Fig. 3-21 shows the simulation result to measure the number of transformations (i.e. $C_T$) and the number of intersection tests (i.e. $C_C$) by the association approaches according to the number of objects. The results show that the proposed methods are more efficient than the basic pair-wise approach. Also, the number of transformations and intersections tests in the camera selection based approach is lower than that of the iteration based approach. This is because only two selected cameras are participating in the association process. However, in terms of the association performance, the selected cameras cannot cover occluded targets as the number of targets increases. As a result, the association performance is degraded as compared with the iteration

based approach in Fig. 3-22. The iteration based approach increases the association performance with the cost of the increased number of transformations and the number of intersection tests.

### 3.4.3 Discussion

The simulation result shows that the camera selection based approach has the lower successful association rate than the others. However, it is not critical because the result is obtained in the limited amount of time and the objects can be associated when the sufficient separation is satisfied. Hence, the successful association rate can be improved as objects move around the surveillance region in time. A more important issue is to minimize the false association so that the consistent global view of multiple cameras is maintained in the system. Once objects are falsely associated, they are propagated in time and it may continuously generate the inconsistent information through tracking. Especially, when objects are more densely populated than the simulation, there is a high possibility that homographic lines are not effective for objects association due to the insufficient separation. Then, the basic pair-wise collaboration may create false associations more and they can corrupt the consistent information in the system. Thus, the effective association collaboration scheme is important to accurately and effectively maintain association information for the insufficient separation.

## 3.5 Summary

We present two different strategies for multiple camera collaborations to reduce the false association for the object association. Collaboration matrices are defined with the required minimum separation for each pair of cameras and used to select a pair of cameras having the maximum separation of homographic lines in the first strategy. We have shown that the first strategy reduces the number of transformations and intersection tests using homographic lines for the object association. However, as a large number of objects are detected, it may require the long operation time to achieve the high association rate due to the unsatisfied separation. In order to remedy the limitation of the first strategy, the second strategy initiates the collaboration process of objects association for all the pairing cases of cameras regardless of the separation. The simulation result demonstrates that the association performance is improved by the repetitive association processes while the computational costs of using homographic lines increase exponentially. The comparison simulation with the basic pair-wise approach also shows that the proposed methods reduce the false association effectively.

**Algorithm 2**: Camera selection based approach

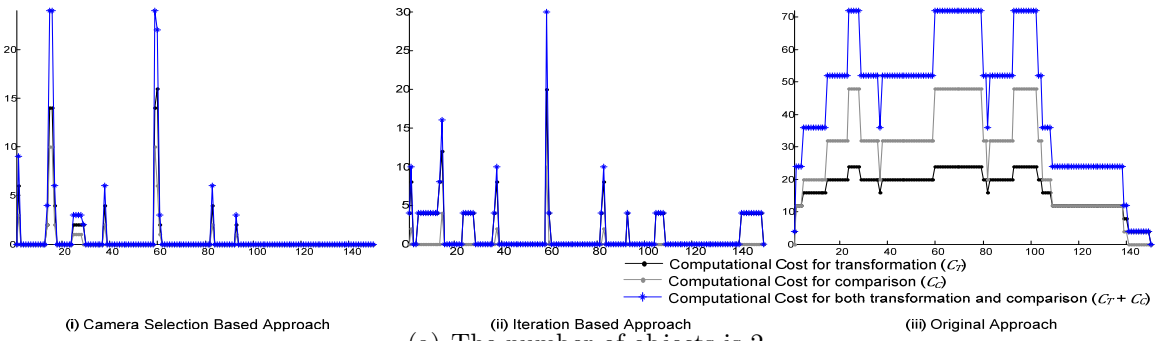
**Input**: Detected targets at each camera $C^k$, threshold matrix $D_{th}$
**Output**: Associated targets
**repeat**
*Classify targets into Type I and Type II targets by local tracking information in each camera* $C^k$
*Construct distance matrix* $D$ *with the minimum distance between targets in each camera* $C^k$
*Select the pair of the best cameras* $\boldsymbol{C}_p = \{(C^i, C^j)|i, j \in k\}$ *based on matrices* $D$ *and* $D_{th}$
**for** $C^k \in \boldsymbol{C}_p$ **do**
**for** $T^k_{i_k} \in \boldsymbol{T}^k_{i_k}$ **do**
**if** $T^k_{i_k}$ *is Type I target* **then**
Generate a vertical homographic line and transform it to other selected camera
**end**
**end**
**end**
*Find associated targets in two selected cameras with transformed homographic lines and extra blob size* $s_{min} \longrightarrow$ associated objects list $\mathbf{A}$
**for** $A \in \boldsymbol{A}$ **do**
**for** $C^k \notin \boldsymbol{C}$ **do**
**for** $T^k_{i_k} \in \boldsymbol{T}^k_{i_k}$ **do**
**if** $T^k_{i_k}$ *is Type I target* **then**
Two homographic lines from $a$ are transformed onto $C^k$ and check association with $s_{min}$ by (3.8)
**end**
**end**
**end**
**end**
**until** *System stops*

| | Crossed Objects in $C^1$ | Crossed Objects in $C^2$ |
|---|---|---|
| $T^1_2$ | - | $\{T^2_2, T^2_4\}$ |
| $T^2_2$ | $\{T^1_2\}$ | - |
| $T^2_4$ | $\{T^1_1, T^1_3, T^1_4\}$ | - |

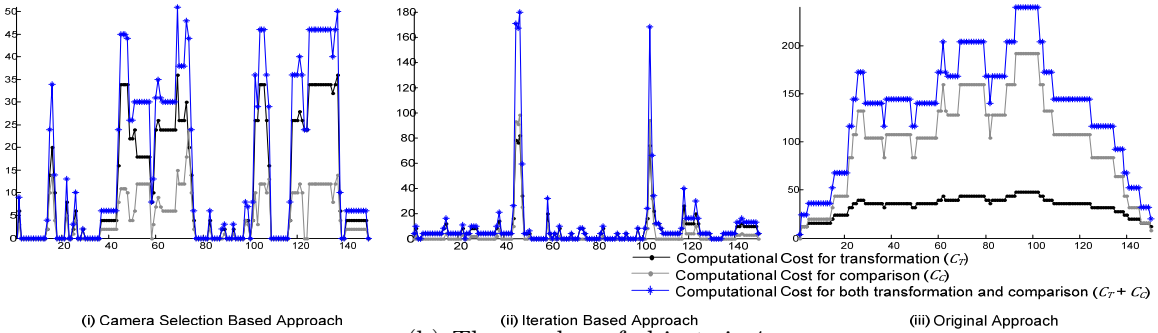
Table 3.1: Crossed targets by the homographic lines for Fig. 3-14.

**Algorithm 3**: Iteration based approach

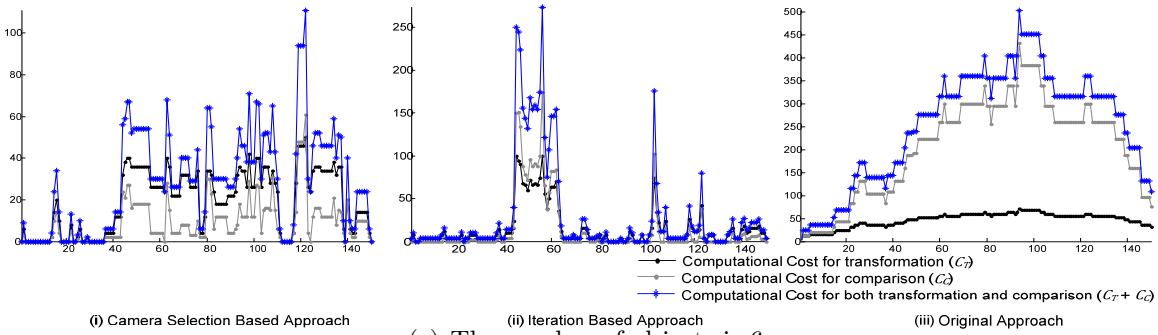
**Input**: Detected targets at each camera $C^k$
**Output**: Associated targets
**repeat**
*Classify targets into Type I and Type II targets by local tracking information in each camera* $C^k$
*Construct all possible pairing cases* $\boldsymbol{C}_p = \{(C^i, C^j)|i, j \in k\}$
**for** $C_p \in \boldsymbol{C}_p$ **do**
**for** $C^k \in C_p$ **do**
**for** $T^k_{i_k} \in \boldsymbol{T}^k_{i_k}$ **do**
**if** $T^k_{i_k}$ *is Type I target* **then**
Generate a vertical homographic line and transform it to the other camera $C^{\widehat{k}}$
Check association with $s_{min}$ and new initiated homographic lines
**if** *only one* $T^{\widehat{k}}_{i_{\widehat{k}}}$ *in Type I target is associated with* $T^k_{i_k}$ **then**
Append association information into associated objects list **A**
**end**
**end**
**end**
**end**
**end**
**for** $A \in \boldsymbol{A}$ **do**
**for** *all* $C^k$ **do**
**for** $T^k_{i_k} \in \boldsymbol{T}^k_{i_k}$ **do**
**if** $T^k_{i_k}$ *is Type I target* **then**
Two homographic lines from $a$ are transformed onto $C^k$ and check association with $s_{min}$ by (3.8)
**end**
**end**
**end**
**end**
**until** *System stops*




(a) The number of objects is 2

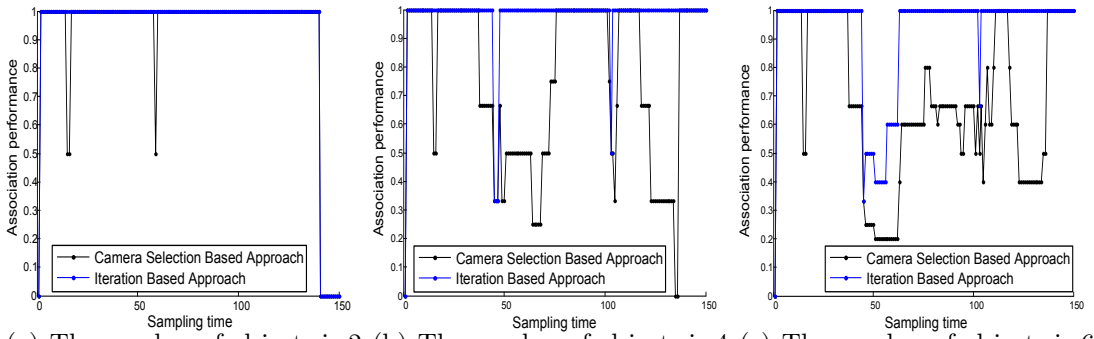



(b) The number of objects is 4




(c) The number of objects is 6

Figure 3-21: The computational costs comparison of the camera selection based approach and the iteration based approach with the basic pair-wise approach according to the number of objects.




(a) The number of objects is 2 (b) The number of objects is 4 (c) The number of objects is 6

Figure 3-22: The association performance comparison of camera selection based approach and iteration based approach with the basic pair-wise approach according to the number of objects.

# Chapter 4

# Object Association and Identification in Heterogeneous Sensors Environment

## 4.1 Introduction

There have been some related works regarding the issue of surveillance using heterogeneous types of sensors. The specific issues considered are various such as heterogeneous data association and efficient network architecture. Schulz et al. [51] proposed the method to track and identify multiple objects by using ID-sensors such as infrared badges and anonymous sensors such as laser range-finders. Although the system successfully associates the anonymous sensor data with ID-sensor data, the transition of the two phases is simply done by the heuristic of the average number of of different assignments in Markov chains. Moreover, it does not provide a recovery method

against losing the correct ID and the number of hypotheses grows extremely fast whenever several people are close to each other. Shin et al. [52] proposed the network architecture for a large scale surveillance system that supports heterogeneous sensors such as video and RFID sensors. Although the event driven control effectively minimizes the system load, it does not deal with the association problem of heterogeneous data but only the mitigation of the data overload. Cho et al. [53] [54] proposed the heterogeneous sensor node with an acoustic and RFID sensor where the coverage of an acoustic sensor is identical to the coverage of a RFID sensor. The association of the estimated position and the identification of an object is achieved by using a simple association rule that one and only one identification is registered within the coverage of the sensor node while its corresponding position is estimated within the coverage of the sensor node. The performance of these approaches, however, can be significantly degraded by the coverage uncertainty of the acoustic and RFID sensors. The coverage uncertainty is caused by the characteristics of acoustic and RFID signal. The system cannot accurately calibrate the time-varying coverage of those sensors. Moreover, multiple objects near the boundary of the sensor coverage may obscure the object identification by identification sensors and the object localization by acoustic sensors. Therefore, an effective association algorithm is needed which can manage the inconsistent registrations of identifications.

In this chapter, we present an approach for dynamic object identification in heterogeneous sensor networks where two functionally different sensors are incorporated. Visual sensors associate objects and track them using the geometric relationship of multiple cameras [23] [55]. The visual sensor-based tracking system is assisted by

identification sensors in identifying the estimated positions of objects. The coverage of identification sensors is assumed by its maximum sensing coverage and the association system applies the simple association strategy for the estimated position from the visual sensor and the identification from the identification sensor. The important issue in heterogeneous sensor networks is to provide the association system with a common reference information fusing heterogeneous data. The visual sensors-based tracking system utilizes the known coverage of the identification sensors to associate the heterogeneous data. The locations of identification sensors are known and they are jointly used with the locations of objects to check the object dynamics of entering and leaving the sensor coverage. The sets of estimated positions and identifications are defined for the coverage of each identification sensor. The association of them is established by checking the temporal change of the sets. In order to solve the association problem with the coverage uncertainty issue, a group and incomplete group association are introduced. The group and incomplete group association enable the association system to maintain identification candidates for the corresponding estimated positions until a single association is established. Also, a group association can stabilize the association performance against the inconsistent registration of identifications by an identification sensor. Additional association cases are investigated to increase the association performance by checking the object dynamics. We also identify more association problems with the discrepancy between the actual coverage by the identification sensor and the approximated coverage by the visual sensor, and present a coverage adjustment scheme using the object dynamics. Finally, the proposed association method is evaluated with a realistic scenario and is analyzed

to show the stability of the proposed method according to degree of the discrepancy between approximated and actual identification sensor coverage, variance of actual identification sensor coverage, and tracking performance.

The remainder of this chapter has 4 sections. In Section 4.2, we present the overview of an application model and problem descriptions. Section 4.3 explains an association method for multiple objects by a group association and incomplete group association with the consideration of the coverage uncertainty problems. In section 4.4, the proposed method is evaluated with a realistic application scenario and is analyzed with non-ideal problems such as the discrepancy between approximated and actual identification sensor coverage, variance of actual identification sensor coverage. Finally, this chapter is summarized in Section 4.5.

## 4.2 Application Model and Problem Description

### 4.2.1 Application Model

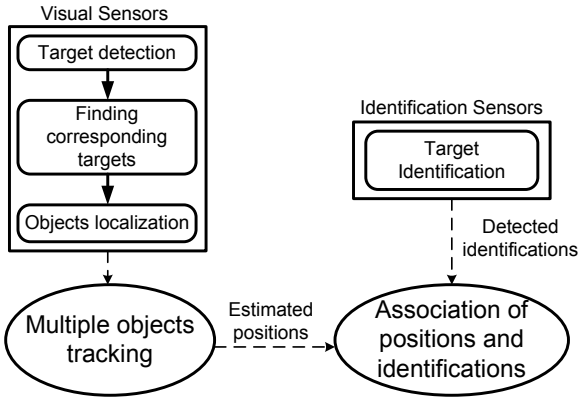



Figure 4-1: Illustration of the overall architecture of the proposed heterogeneous sensor system using visual sensor and identification sensor.

Fig. 4-1 shows the architecture of the association system that we consider in this chapter. Visual sensors continuously detect and track objects by various techniques [56] [57] [58]. In order to find the corresponding targets of objects among multiple cameras, locally initiating homographic line method is used [23]. Objects are localized by a simple 2-D localization algorithm in [55]. On the other hand, identification sensors register identifications of objects within their own coverage. The association of an object at time $t$ is defined as

$$O_{(t)} : x_{(t)} \leftrightarrow \text{ID}, \tag{4.1}$$

where $x_{(t)}$ is the estimated 2-D position from the visual sensor and ID is the identification obtained from the identification sensor.

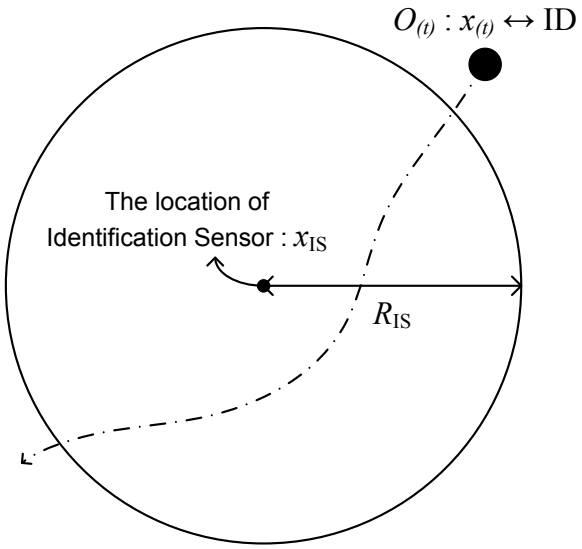



Figure 4-2: Example of an association and identification with an ideal sensor coverage

Fig. 4-2 shows an association approach of two different types of signals to identify the estimated position of an object. Let $\left\{x^1_{(t)}, x^2_{(t)} \cdots, x^M_{(t)}\right\}$ denote the set of objects'

positions inside the coverage of the identification sensor at time $t$, where the actual coverage radius of identification sensor is $R_{\mathrm{IS}}$ in the ideal case. Since the association system knows the locations of identification sensors and the estimated positions of objects, it can check whether an object is within the coverage of the identification sensor by using the distance between them. Define the set of objects' positions within the coverage of the identification sensor but not associated with an ID at time $t$ as

$$S^x_{(t)} := \{x^m | \mathrm{dist}(x^m_{(t)}, x_{\mathrm{IS}}) \leq R_{\mathrm{approx}} \text{ and } \mathbf{1}_{\left\{x^m_{(t)} \leftrightarrow \mathrm{ID}\right\}} = 0, \\ \forall \mathrm{ID}, \text{ for } m = 1, 2, \cdots, M'\}, \tag{4.2}$$

where $x_{\mathrm{IS}}$ denotes the location of the identification sensor and $\mathrm{dist}(x^m_{(t)}, x_{\mathrm{IS}})$ is the distance between $x^m_{(t)}$ and $x_{\mathrm{IS}}$. $\mathbf{1}_{\left\{x^m_{(t)} \leftrightarrow \mathrm{ID}\right\}}$ is the indicator function, where $\mathbf{1}_{\left\{x^m_{(t)} \leftrightarrow \mathrm{ID}\right\}} = 0$ means that an estimated position does not have an associated identification. Note that $R_{\mathrm{approx}}$ is the maximum radius of an identification sensor where there are $M'$ positions of objects while there are $M$ objects. As the actual coverage of an identification sensor can vary and a visual sensor tracks the objects for the radius of $R_{\mathrm{approx}}$, $M$ can be different from $M'$. Similarly, $S^{\mathrm{ID}}_{(t)}$ is defined as the set of identifications not being associated but registered by the identification sensor.

The simple association condition for a single object is given by

$$\mathcal{N}\left(S^x_{(t)}\right) = \mathcal{N}\left(S^{\mathrm{ID}}_{(t)}\right) = 1, \tag{4.3}$$

where $\mathcal{N}(S)$ represents the number of elements in the set $S$ [53] [54]. In other words, for an identification sensor at a time instance, if there is one unassociated ID (from identification sensor) and one unassociated object position (from visual sensor), the association can simply be made. However, in practical applications, the condition in (4.3) may not be satisfied.

### 4.2.2 Problem Description

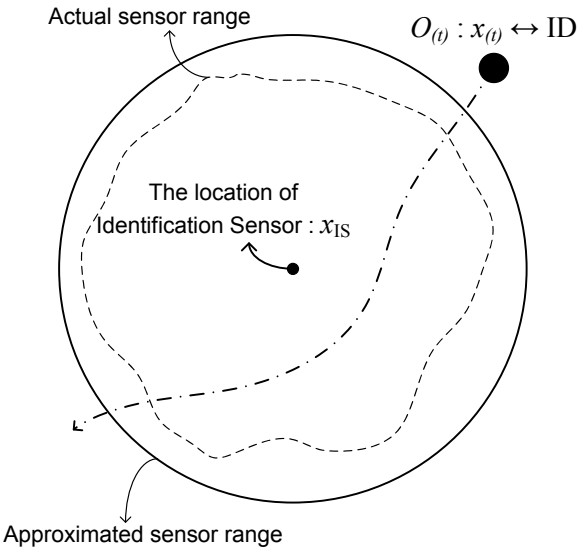



Figure 4-3: Illustration of a non-ideal identification region model. Range is variable because of the radio frequency characteristics.

The association problems can be non-trivial, especially when RFID-type identification sensors are used as shown in Fig. 4-3. For those types of sensors, as they are based on the reception of the radio frequency signal, which can be easily distorted by the environment, the coverage of the sensor can become time-varying without being known to the visual sensor. Then, the actual coverage of an identification sensor can be different from the approximated coverage by a visual sensor and the condition in

(4.3) may not be satisfied – there are more than one unassociated objects' positions but fewer number of unassociated ID's, or vice versa. Even if the coverage of the identification sensor is not time-varying, there can still be the coverage uncertainty problem, when objects are densely populated near the boundary of the coverage. In order to adapt to the time-varying coverage of the identification sensor, the maximum sensing coverage of the identification sensor can be assumed by the visual sensor.

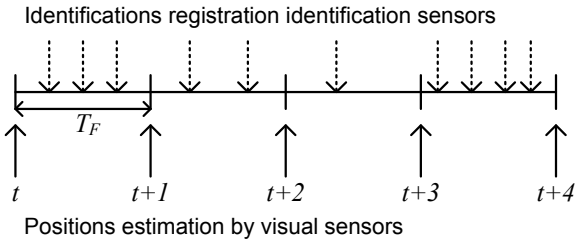



Figure 4-4: Illustration of timing diagram of an identification sensor and a visual sensor

Violation of the condition in (4.3) can happen due to the coverage discrepancy between the sampling intervals of two sensors as shown in Fig. 4-4. For example, an ID registered during one sampling interval of the visual sensor can be associated with multiple estimated positions within the coverage of an identification sensor. An ideal situation for an association is that one and only one ID is registered during one sampling interval of the visual sensor and one position is newly added and estimated at each sampling time within the coverage of an identification sensor. However, the registration of identifications within the approximated coverage of the visual sensor is not always guaranteed due to the coverage uncertainty. Identifications may not be registered sequentially as multiple objects enter the approximated coverage of the visual sensor. Also, the registration times of identifications may not coincide with

the estimation time of the corresponding positions. Then, it is difficult to associate identifications with estimated positions by using only the simple association condition in (4.3).

The association problems become more difficult when objects with and without identifications coexist. Especially, when there is the coverage uncertainty issue, the association system cannot clearly determine whether an object has an ID or not. The deterministic association approach by one-to-one assignment may falsely associate identifications with unassociated estimated positions. Moreover, the association system may switch identification while tracking multiple objects when objects collide with each other. Therefore, the association system requires an effective association algorithm that can recover association failures by managing the coverage uncertainty.

## 4.3 Association and Identification with Coverage Uncertainty

### 4.3.1 Multiple Objects Association

#### Association without Coverage Uncertainty

Even when the coverages of the identification sensor and the visual sensor are identical, the association failure, the violation of the condition in (4.3), can happen mainly due to two reasons – the simultaneous entrance and the collision as shown in Fig. 4-5. A collision between the objects can lead to a failure in tracking objects since they are too close to be differentiated for position and ID assignments. When multiple objects

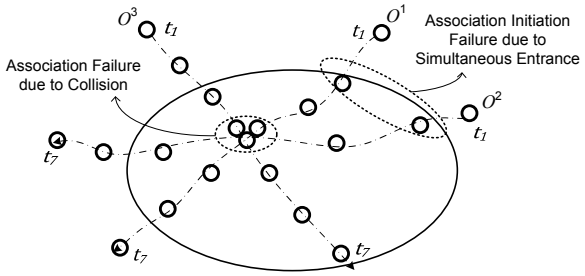



Figure 4-5: Illustration of association failure cases for multiple objects in identification sensor region without coverage uncertainty.

simultaneously enter the coverage of the identification sensor, the condition in (4.3) is not satisfied, that is multiple objects are registered during a single sampling time of the visual sensor and $\mathcal{N}\left(S^x_{(t)}\right) = \mathcal{N}\left(S^{\mathrm{ID}}_{(t)}\right) > 1$. As investigated in [53], increasing the sampling time of the visual sensor can alleviate the problem, but it cannot be the fundamental solution to the simultaneous entrance problem. When there is a collision between the objects, the association can also fail. Although the visual sensor can track multiple objects after the collision, the associations between the objects and the ID's are no longer valid. If the dynamic transition model of objects is known, an identification assignment can be estimated through the tracking. However, the accurate model is not always known to the association system. The existing method shown in [53] [51], waits for a new association until the association-failure objects enter the coverage of a new identification sensor. Although this method can provide an association recovery, all the established associations are lost by the collision.

In order to efficiently deal with the association failures, a group association can be used as shown in Fig. 4-6. It can be initiated by the simultaneous entrance or the

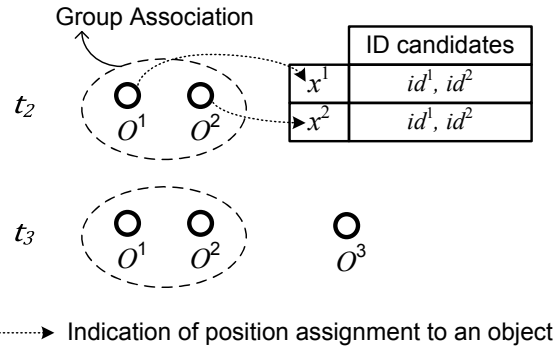



Figure 4-6: Illustration of a possible association initiation by a group association.

collision. Consider the set of association groups and each group $G$ is defined by

$$G_{(t)}^{x,p} \leftrightarrow G_{(t)}^{\mathrm{ID},p}, \text{ for } p = 1, 2, \cdots, P, \tag{4.4}$$

where $G_{(t)}^{x,p}$ and $G_{(t)}^{\mathrm{ID},p}$ are the set of positions and the set of identifications respectively, for group association index $p$ at time $t$, and $P$ is the number of group associations for an identification sensor. A group association within the coverage of an identification sensor is established by

$$\mathcal{N}\left(S_{(t)}^{x} - \bigcup_{p=1}^{P} G_{(t)}^{x,p}\right) = \mathcal{N}\left(S_{(t)}^{\mathrm{ID}} - \bigcup_{p=1}^{P} G_{(t)}^{\mathrm{ID},p}\right) > 1. \tag{4.5}$$

In other words, for an identification sensor at a time instance, if there are more than one unassociated ID's (from the identification sensor) and the same number of unassociated object positions (from the visual sensor), then a group association can be made.

Once multiple objects are associated as a group with the same number of identifications, they are considered to have associated identifications, but still included in

the set $S^{x}_{(t)}$ and $S^{\mathrm{ID}}_{(t)}$. Suppose that $x^1$ and $x^2$ are associated with $\mathrm{ID}^1$ and $\mathrm{ID}^2$ as a group by the simultaneous entrance or a collision. If a newly estimated position, $x^3$ is not associated with any identification, a different identification from $\mathrm{ID}^1$ and $\mathrm{ID}^2$, say $\mathrm{ID}^3$ is registered in the sensor coverage, then a newly registered identification is associated with the estimated position $x^3$ by

$$\mathcal{N}\left(S^{x}_{(t)} - \bigcup_{p=1}^{P} G^{x,p}_{(t)}\right) = \mathcal{N}\left(S^{\mathrm{ID}}_{(t)} - \bigcup_{p=1}^{P} G^{\mathrm{ID},p}_{(t)}\right) = 1, \tag{4.6}$$

which is the condition of association, modified from the condition in (4.3). Although the condition in (4.6) establishes a single association for a newly added object, such a single association cannot be established for an object in a group association by the condition in (4.6).

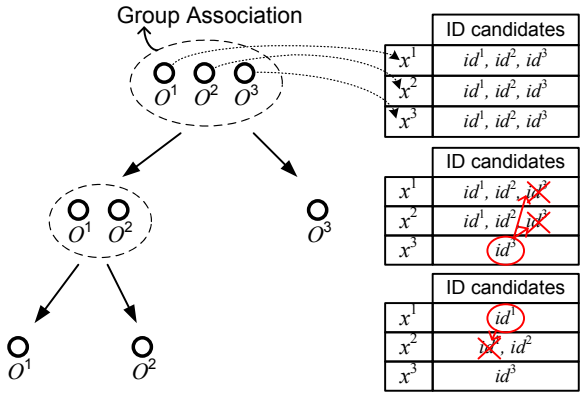



Figure 4-7: Illustration of how a system split a group association into single associations.

When there are multiple objects inside the coverage, the association system can utilize the object dynamics of entering or leaving the coverage to establish a single association for an object in a group association. Fig. 4-7 shows an example of how a group association is divided into single association. The association condition for an

entering object at the coverage of an identification sensor is represented by

$$\begin{aligned} \mathcal{N}\left(\left\{x^m \;\middle|\; x^m_{(t)} \in S^x_{(t)}, x^m_{(t-1)} \notin S^x_{(t-1)}\right\}\right) &= \\ \mathcal{N}\left(\left\{\mathrm{ID}^l \;\middle|\; \mathrm{ID}^l_{(t)} \in S^{\mathrm{ID}}_{(t)}, \mathrm{ID}^l_{(t-1)} \notin S^{\mathrm{ID}}_{(t-1)}\right\}\right) &= 1, \end{aligned} \tag{4.7}$$

and for a leaving object at the coverage of an identification sensor, the condition is

$$\begin{aligned} \mathcal{N}\left(\left\{x^m \;\middle|\; x^m_{(t)} \notin S^x_{(t)}, x^m_{(t-1)} \in S^x_{(t-1)}\right\}\right) &= \\ \mathcal{N}\left(\left\{\mathrm{ID}^l \;\middle|\; \mathrm{ID}^l_{(t)} \notin S^{\mathrm{ID}}_{(t)}, \mathrm{ID}^l_{(t-1)} \in S^{\mathrm{ID}}_{(t-1)}\right\}\right) &= 1. \end{aligned} \tag{4.8}$$

These conditions in (4.7) and (4.8) can be extended to associate multiple objects in group associations with their own identifications. If the estimated position $x^m_{(t)}$ is in a group association, this can be differentiated from the added positions those are not in a group association. Suppose that $G^x(x^m_{(t)})$ is the set of positions of $x^m_{(t)}$ and $G^{\mathrm{ID}}(x^m_{(t)})$ is the set of identifications corresponding to $G^x(x^m_{(t)})$. Then, the conditions in (4.7) and (4.8) for entering and leaving objects are modified to

$$\begin{aligned} \mathcal{N}\left(\left\{x^m \,\middle|\, x^m_{(t)} \in S^x_{(t)} \cap G^x(x^m_{(t)}), x^m_{(t)} \notin S^x_{(t-1)}\right\}\right) &= \\ \mathcal{N}\left(\left\{\mathrm{ID}^l \,\middle|\, \mathrm{ID}^l_{(t)} \in S^{\mathrm{ID}}_{(t)} \cap G^{\mathrm{ID}}(x^m_{(t)}), \mathrm{ID}^l_{(t-1)} \notin S^{\mathrm{ID}}_{(t-1)}\right\}\right) &= 1, \end{aligned} \tag{4.9}$$

and

$$\mathcal{N}\left(\left\{x^m \,\middle|\, x^m_{(t)} \in S^x_{(t-1)} \cap G^x(x^m_{(t)}), x^m_{(t)} \notin S^x_{(t)}\right\}\right) =$$

$$\mathcal{N}\left(\left\{\mathrm{ID}^l \,\middle|\, \mathrm{ID}^l_{(t)} \in S^{\mathrm{ID}}_{(t-1)} \cap G^{\mathrm{ID}}(x^m_{(t)}), \mathrm{ID}^l_{(t)} \notin S^{\mathrm{ID}}_{(t)}\right\}\right) = 1, \tag{4.10}$$

respectively. A group association is divided into single association(s) or other group associations by these conditions.

**Effects of Coverage Uncertainty**

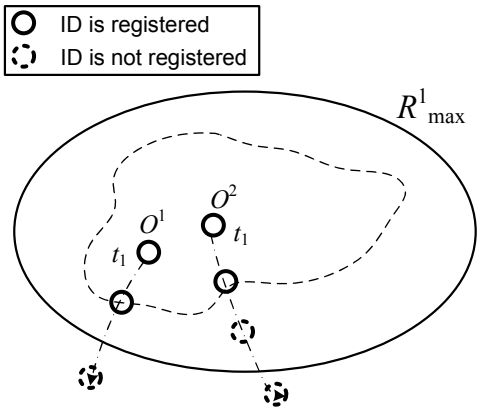



Figure 4-8: Illustration of a case in which an object in group association may not satisfy (4.10) due to coverage uncertainty.

The entering or leaving condition in the group association (4.9) and (4.10) can only be satisfied when the coverages of the identification sensor and the visual sensor are identical. The discrepancy between the actual coverage by the identification sensor and the approximated coverage by the visual sensor may generate cases where the conditions in (4.9) or (4.10) are not satisfied. The registered identifications of objects within the actual coverage may not be consistent with the estimated positions of

them as shown in Fig. 4-8. For example, suppose that $x^1$ and $x^2$ are associated with $\mathrm{ID}^1$ and $\mathrm{ID}^2$ as a group. $x^1$ enters or leaves the coverage before $x^2$ does. In order to establish a single association for $x^1$ to $\mathrm{ID}^1$ or for $x^2$ to $\mathrm{ID}^2$, $\mathrm{ID}^1$ and $\mathrm{ID}^2$ need to be registered or de-registered sequentially in the order that they enter or leave the coverage. However, regardless of the entering or leaving order by the visual sensor, $\mathrm{ID}^1$ and $\mathrm{ID}^2$ can be occasionally registered or de-registered at the same time due to the coverage uncertainty. In this case, the entering or leaving conditions in the group association (4.9) or (4.10) is not satisfied for a single association.

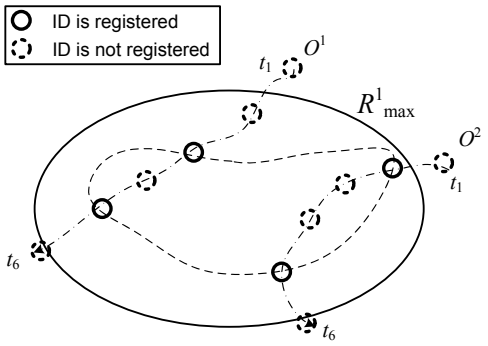



Figure 4-9: Illustration of multiple objects association in sensor coverage with uncertainty.

Another association problem to be considered is due to the inconsistent registration of identifications within the approximated coverage by the visual sensor as shown Fig. 4-9. Since all identifications are not always registered in the coverage of the identification sensor due to the coverage uncertainty, $S^{\mathrm{ID}}_{(t)}$ or $S^{\mathrm{ID}}_{(t-1)}$ may not be consistent in the entering or leaving conditions in the group association. It indicates that the association system may not always correctly determine whether an object enters or leaves the coverage of identification sensors.

The incomplete group association is introduced to effectively utilize the inconsis-

tent registrations of identifications. An incomplete group association is established by

$$\mathcal{N}\left(S^{x}_{(t)} - \bigcup_{p=1}^{P} G^{x,p}_{(t)}\right) \neq \mathcal{N}\left(S^{\mathrm{ID}}_{(t)} - \bigcup_{p=1}^{P} G^{\mathrm{ID},p}_{(t)}\right), \tag{4.11}$$

where each object is registered as an element of the incomplete group association with possible identification candidates.

| | $x^1 \blacktriangleleft\cdots O^1$ | $x^2 \blacktriangleleft\cdots O^2$ | $x^3 \blacktriangleleft\cdots O^3$ |
|---|---|---|---|
| $t_2$ | $id^2$ | $id^2$ | |
| $t_3$ | $id^1$, $id^2$ | $id^1$, $id^2$ | $id^1$ |
| $t_4$ | $id^1$, $id^2$, $id^3$ | $id^1$, $id^2$, $id^3$ | $id^1$, $id^3$ |

Figure 4-10: Illustration of incomplete group association in sensor coverage with uncertainty.

Suppose that identification $\mathrm{ID}^1$ is not registered but $\mathrm{ID}^2$ is registered while both $x^1$ and $x^2$ are estimated within the coverage as shown in Fig. 4-10. Then, $x^1$ and $x^2$ are registered as elements of an incomplete group association. At every time instance when the condition in(4.11) is satisfied, new possible identifications are added to the candidates. However, due to the coverage uncertainty, it is not guaranteed that an object in an incomplete group association has its identification in its candidates. Also, objects without identifications may have irrelevant identifications in their candidates. Elements in an incomplete group are removed when they are associated with other estimated positions by a single or group association. While an associable identification in a group association is limited to the identification candidates of an object, the estimated position of an object in an incomplete group association can be associated with an identification beside its candidates. Therefore, an object in an incomplete group association establish a single association by using

$$\mathcal{N}\left(S^{x}_{(t)} \cap G'^{x}\left(x^{m}_{(t)}\right)\right) = \mathcal{N}\left(S^{\mathrm{ID}}_{(t)} \cap G'^{\mathrm{ID}}\left(x^{m}_{(t)}\right)\right) = 1, \tag{4.12}$$

where $G'^{x}\left(x^{m}_{(t)}\right)$ is the set of positions in relation to incomplete group association with $x^{m}_{(t)}$ and $G'^{\mathrm{ID}}\left(x^{m}_{(t)}\right)$ is the set of the candidate identifications corresponding to $G'^{x}\left(x^{m}_{(t)}\right)$.

### 4.3.2 Group Association by Temporal Set Maintenance

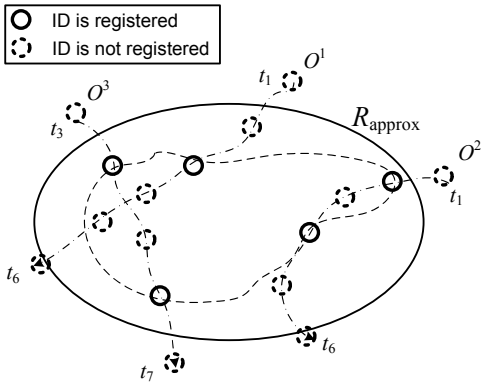



Figure 4-11: Illustration of a case in which group association is not established by the registration uncertainty of identifications.

| Sampling Time | $S^{x}_{(t)}$ | $S^{\mathrm{ID}}_{(t)}$ |
|---|---|---|
| $t_2$ | $\left\{x^{1}_{(t_2)}, x^{2}_{(t_2)}\right\}$ | $\left\{\mathrm{ID}^{2}_{(t_2)}\right\}$ |
| $t_3$ | $\left\{x^{1}_{(t_3)}, x^{2}_{(t_3)}\right\}$ | $\left\{\mathrm{ID}^{1}_{(t_3)}\right\}$ |
| $t_4$ | $\left\{x^{1}_{(t_4)}, x^{2}_{(t_4)}, x^{3}_{(t_4)}\right\}$ | $\left\{\mathrm{ID}^{2}_{(t_4)}, \mathrm{ID}^{3}_{(t_4)}\right\}$ |
| $t_5$ | $\left\{x^{1}_{(t_5)}, x^{2}_{(t_5)}, x^{3}_{(t_5)}\right\}$ | $\{\}$ |

Table 4.1: The variation of sets of the estimated positions and identifications for Fig. 4-11.

The group maintenance algorithm discussed before is based on the set of estimated positions and the set of identifications at each sampling time. However, the regis-

tration uncertainty of identifications may delay establishment of a group association. For example, TABLE 4.1 shows the variation of sets of the estimated positions and identifications at each sampling time for Fig. 4-11. Since $\mathrm{ID}^1$ and $\mathrm{ID}^2$ are registered at different sampling times, they are associated as an incomplete group association. The problem of an incomplete group association is to generate another incomplete group association until they are associated as a single or group association. For example, $\mathrm{ID}^3$ is registered in the coverage at $t_4$, but the association system cannot clearly recognize it as a newly added ID due to its unassociated identifications. They all become an incomplete group association again by the condition in (4.11).

In order to increase the establishment of a group association, the association system can keep temporally registered identifications at different sampling time, until objects does stay within the coverage. $\widetilde{S}^{\mathrm{ID}}_{(t)}$ denotes the temporally maintained set of identifications in the coverage and this set is updated by

$$\widetilde{S}^{\mathrm{ID}}_{(t)} = \begin{cases} \widetilde{S}^{\mathrm{ID}}_{(t-1)} \cup S^{\mathrm{ID}}_{(t)} & \text{for } S^{x}_{(t-1)} \subseteq S^{x}_{(t)} \\ S^{\mathrm{ID}}_{(t)} & \text{otherwise.} \end{cases} \tag{4.13}$$

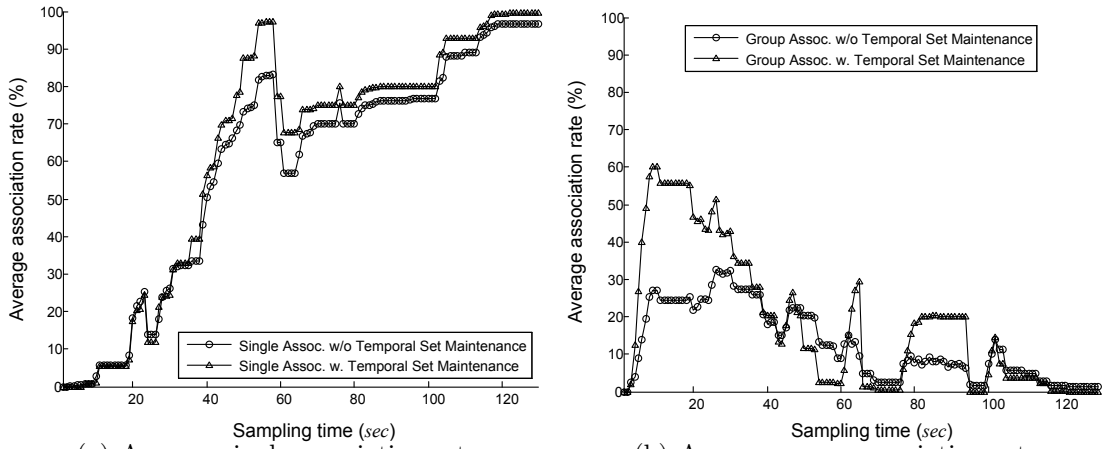
If an object leaves the coverage, $\widetilde{S}^{\mathrm{ID}}_{(t)}$ should not keep the previously registered identifications because the association system does not know which object leaves the coverage. By using the temporally maintained identification set, the association system has a group association condition by,

$$\mathcal{N}\left(S^{x}_{(t)} - \bigcup_{p=1}^{P} G^{x,p}_{(t)}\right) = \mathcal{N}\left(\widetilde{S}^{\mathrm{ID}}_{(t)} - \bigcup_{p=1}^{P} G^{\mathrm{ID},p}_{(t)}\right) > 1. \tag{4.14}$$

TABLE 4.2 shows how the sets of estimated positions and identifications vary using the temporal set maintenance. $\{x^1, x^2\}$ are associated with $\{\mathrm{ID}^1, \mathrm{ID}^2\}$ as a group at $t_3$. Since $x^3$ is associated with $\mathrm{ID}^3$ at the next sampling time, $x^3$ and $\mathrm{ID}^3$ are removed in $S^{\mathrm{ID}}_{(t)}$ and $\widetilde{S}^{\mathrm{ID}}_{(t)}$.

| Sampling Time | $S^x_{(t)}$ | $S^{\mathrm{ID}}_{(t)}$ | $\widetilde{S}^{\mathrm{ID}}_{(t)}$ |
|---|---|---|---|
| $t_2$ | $\left\{x^1_{(t_2)}, x^2_{(t_2)}\right\}$ | $\left\{\mathrm{ID}^2_{(t_2)}\right\}$ | $\left\{\mathrm{ID}^2_{(t_2)}\right\}$ |
| $t_3$ | $\left\{x^1_{(t_3)}, x^2_{(t_3)}\right\}$ | $\left\{\mathrm{ID}^1_{(t_3)}\right\}$ | $\left\{\mathrm{ID}^1_{(t_3)}, \mathrm{ID}^2_{(t_3)}\right\}$ |
| $t_4$ | $\left\{x^1_{(t_4)}, x^2_{(t_4)}\right\}$ | $\left\{\mathrm{ID}^2_{(t_4)}\right\}$ | $\left\{\mathrm{ID}^1_{(t_4)}, \mathrm{ID}^2_{(t_4)}\right\}$ |
| $t_5$ | $\left\{x^1_{(t_5)}, x^2_{(t_5)}\right\}$ | $\{\}$ | $\left\{\mathrm{ID}^1_{(t_5)}, \mathrm{ID}^2_{(t_5)}\right\}$ |

Table 4.2: The variation of sets of the estimated positions and identifications using temporal set maintenance for Fig. 4-11.




(a) Average single association rate. (b) Average group association rate.

Figure 4-12: Comparison between the association performance with and without temporal set maintenance.

Fig. 4-12 shows the performance comparison between association algorithms with and without the temporal set maintenance. Ten objects dynamically move around the surveillance region where four identification sensors are installed. At every time interval of identification sensor, each object is registered with probability of 0.5. It is

assumed that the system fails in tracking when objects are adjacent within $0.3m$. The association simulation is repeated 100 times and the results are averaged in order to reflect the effect of the coverage uncertainty. The blue line indicates the simulation result with the temporal set maintenance. When the identification set is temporally maintained by the condition in (4.13), temporally unregistered identifications are still maintained in the set of $\widetilde{S}^{\mathrm{ID}}_{(t)}$. Then, it increases the possibility of establishing a group association increases rather than an incomplete group association. Since the objects in a group association are distinguished from other objects, the chance of establishing a single association also increases. As a result, the association rate increases faster with the temporal set maintenance than without the temporal set maintenance.

### 4.3.3 Association Stability in Mismatched Model

Association performance is also influenced by the discrepancy between the approximated coverage and the actual coverage. When the approximated coverage is greater than the actual coverage, positions of objects with non-registered identifications can be estimated within the approximated coverage. Then, a group or incomplete group association increases by the condition in (4.5) or (4.11). This can frequently occur when objects move around the boundary of coverage of an identification sensor. Moreover, the effect of the smaller approximated coverage than actual coverage is similar to the effect of the larger approximated coverage than actual coverage. Since the number of registered identifications is different from the number of estimated positions within the approximated coverage, this may increase group or incomplete

group associations. However, the estimated positions of objects are eventually identified when single associations are established. While the inaccurate coverage model may delay the establishment of single associations, the number of single associations eventually increases by the object dynamics.

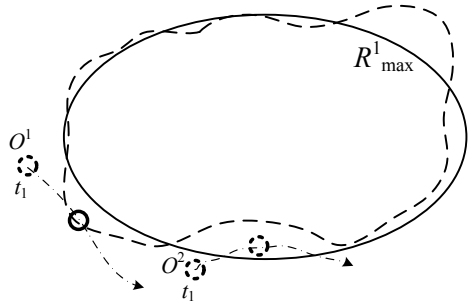



Figure 4-13: Illustration of a false association when actual sensor coverage is irregular.

The irregular sensor coverage causes a false association with an non-corresponding identification when objects move around the boundary of the modeled coverage as shown in Fig. 4-13. For example, $x^1$ is not estimated but $x^2$ is estimated inside the coverage of the visual sensor. Also, on the other hand, only $\mathrm{ID}^1$ is registered inside the coverage. Then, $x^2$ can be falsely associated with $\mathrm{ID}^1$ by the condition in (4.3). Since a single association is established, the association system cannot confirm the false association immediately. However, the association system can cope with false associations using two approaches. One is a passive approach that uses the property of a group association. If objects in relation with a false association collide inside or outside the coverage, a false association naturally becomes a group or incomplete group association. The other approach is to confirm the association by checking whether duplicated identifications exist in the association system. If the false association is confirmed, the falsely associated position changes to an unassociated

position. Therefore, false associations are eventually resolved by a group association or checking the identification with duplicate registrations at the coverage of different identification sensors.

### 4.3.4 Coverage Adjustment Scheme

At the initial state, the approximated radius of an identification sensor is set as a physical variable in the system. Since the radius is used to determine whether objects enter or leave the coverage of an identification sensor, it needs to be accurately estimated for the improved association performance. However, the association performance is also affected by the simultaneous entrance and the collision. These phenomena frequently occurs where objects are densely populated. The association performance is not improved proportionally to the degree of the accurate estimation of the radius but the time to stabilize the association performance is inversely proportional to the degree of the accurate estimation of the radius. In order to adjust the initial radius of an identification sensor, we utilize the object dynamics of entering and leaving the coverage of an identification sensor.

The basic idea of the coverage adjustment scheme is to compare the number of estimated positions with the number of registered identifications within the coverage of an identification sensor. If the approximated radius of an identification sensor is accurate enough, the number of the estimated positions is mostly equivalent to the number of the registered identifications. Otherwise, it means that the approximated coverage differs from the actual coverage. The radius of an identification sensor is

adjusted by checking the difference between them. In some cases, the system needs to check the farthest or closet estimated position from the center of an identification sensor. For example, when the number of the estimated positions is equivalent to the number of the registered identifications, the coverage of an identification should be adjust to the farthest estimated position. Then, the problem in the coverage adjustment scheme is to determine how degree the radius is adjusted by at each sampling time. Since the coverage of an identification sensor can vary temporally, the large change of the radius may cause a reverse effect and the association performance may degenerate. Thus, we use the average speed of tracked objects measured by the association system as the degree of the radius adjustment to be unsusceptible to the object dynamics.

The temporal change of sets of positions and identifications is utilized to adjust the initial coverage, while the coverage of an identification sensor is assumed to slowly vary. Since an association can be established at every sampling time $t$, the approximated coverage of the visual sensor is also adjusted by the change of a radius $\Delta r$ at time $t$. The average speed of tracked objects, measured by the association system, can be used to determine $\Delta r$, since the registration is related to the object dynamics. Define $R_{(t)}$ as the adjusted radius between radii $R_{\min}$ and $R_{\max}$ for an identification sensor at time $t$. The set of estimated positions within $R_{(t)}$ is denoted by $X_{(t)}$ and the set of registered identifications within $R_{(t)}$ is denoted by $\mathrm{ID}_{(t)}$.

At time $t$, the set of newly added estimated positions and registered identifications

are represented, respectively by $E^x_{(t)}$ and $E^{\mathrm{id}}_{(t)}$ as

$$E^x_{(t)} = X_{(t)} - X_{(t-1)}, \text{ and } E^{\mathrm{id}}_{(t)} = \mathrm{ID}_{(t)} - \mathrm{ID}_{(t-1)}. \tag{4.15}$$

When the number of changes for each set are equal by $\mathcal{N}\left(E^x_{(t)}\right) = \mathcal{N}\left(E^{\mathrm{id}}_{(t)}\right)$, the radius is kept by

$$R^E_{(t)} = R_{(t-1)}, \tag{4.16}$$

where $R^E_{(t)}$ denotes the adjusted coverage determined by added objects and its coverage is between $R_{\min}$ and $R_{\max}$. On the other hand, when the number of newly registered identifications is smaller than the number of newly estimated positions at time $t-1$ by $\mathcal{N}\left(E^x_{(t)}\right) > \mathcal{N}\left(E^{\mathrm{id}}_{(t)}\right) > 0$, the current radius is reduced by

$$R^E_{(t)} = R_{(t-1)} - \Delta r. \tag{4.17}$$

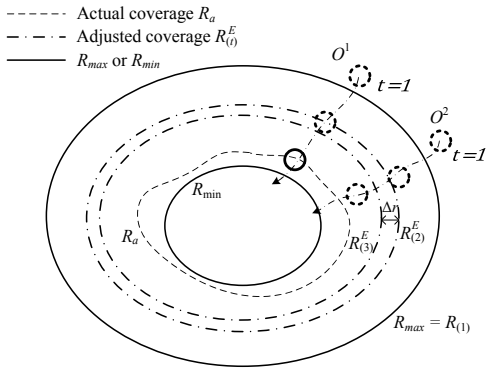



Figure 4-14: Illustration of coverage reduction when objects enter the coverage of an identification sensor

If no identification is registered, $\mathcal{N}\left(E^x_{(t)}\right) > 0$ and $\mathcal{N}\left(E^{\mathrm{id}}_{(t)}\right) = 0$ as shown in Fig. 4-14, the current radius of the approximated coverage can be much larger than

the radius of actual coverage. In this case, the estimated position with the minimum distance from a sensor position is used to determine the adjusted radius by

$$R^E_{(t)} = \min_{x^m_{(t)} \in E^x_{(t)}} \left( \text{dist}\left( x^m_{(t)}, x_{\text{IS}} \right) \right) - \Delta r. \tag{4.18}$$

On the contrary, when the number of registered identifications is greater than the number of added estimated positions, $\mathcal{N}\left(E^{\text{id}}_{(t)}\right) > \mathcal{N}\left(E^x_{(t)}\right) > 0$, the current radius is enlarged by

$$R^E_{(t)} = R_{(t-1)} + \Delta r. \tag{4.19}$$

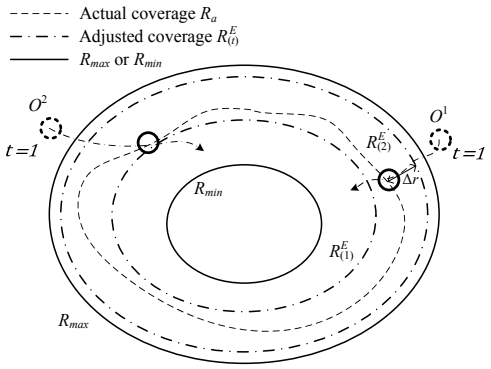



Figure 4-15: Illustration of coverage enlargement when objects enter the coverage of an identification sensor

In particular, if the number of added identifications is equal to the number of estimated positions within $R_{\max}$, $\mathcal{N}\left(E^{\text{id}}_{(t)}\right) = \mathcal{N}\left(E^x_{(t)}\right) > 0$ as shown in Fig. 4-15, the current radius of approximated coverage can be much smaller than the radius of actual coverage. Then, the radius is enlarged by

$$R^E_{(t)} = \max_{x^m_{(t)} \in E^x_{(t)}} \left( \text{dist}\left( x^m_{(t)}, x_{\text{IS}} \right) \right) + \Delta r, \tag{4.20}$$

where $\Delta r$ is added for the extra coverage to prevent false associations by the irregular property of actual coverage.

A similar radius adjustment can be applied to the case where objects leave the coverage of an identification sensor. A set of leaving positions $L^x_{(t)}$ and a set of leaving identifications $L^{\mathrm{id}}_{(t)}$ at time $t$ is represented respectively by

$$L^x_{(t)} = X_{(t-1)} - X_{(t)}, \text{ and } L^{\mathrm{id}}_{(t)} = \mathrm{ID}_{(t-1)} - \mathrm{ID}_{(t)}. \tag{4.21}$$

When the number of leaving identifications is equal to the number of leaving positions, $\mathcal{N}\left(L^x_{(t)}\right) = \mathcal{N}\left(L^{\mathrm{id}}_{(t)}\right)$, the current radius is kept by

$$R^L_{(t)} = R_{(t-1)}, \tag{4.22}$$

where $R^L_{(t)}$ denotes the adjusted coverage determined by leaving objects with the coverage between $R_{\min}$ and $R_{\max}$. On the other hand, when the number of leaving identifications is greater than the number of leaving estimated positions, $\mathcal{N}\left(L^x_{(t)}\right) < \mathcal{N}\left(L^{\mathrm{id}}_{(t)}\right)$, the radius is reduced as

$$R^L_{(t)} = R_{(t-1)} - \Delta r. \tag{4.23}$$

If the number of leaving identifications and estimated positions are equal, $\mathcal{N}\left(L^{\mathrm{id}}_t\right) = \mathcal{N}\left(X_{(t-1)}\right)$ as shown in Fig. 4-16, the radius is reduced by an estimated position hav-

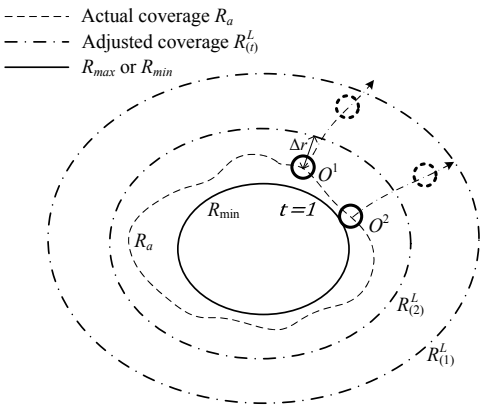



Figure 4-16: Illustration of coverage reduction when objects leave the coverage of an identification sensor

ing the maximum distance from the position of an identification sensor by

$$R^L_{(t)} = \max_{x^m_{(t-1)} \in X_{(t-1)}} \left(\text{dist}\left(x^m_{(t-1)}, x_{\text{IS}}\right)\right) + \Delta r, \tag{4.24}$$

where $\Delta r$ is added for the extra coverage to prevent false associations by the irregular property of actual coverage. When the number of leaving identifications is smaller than the number of leaving estimated positions, $\mathcal{N}\left(L^x_{(t)}\right) > \mathcal{N}\left(L^{\text{id}}_{(t)}\right)$, the radius is enlarged by

$$R^L_{(t)} = R_{(t-1)} + \Delta r. \tag{4.25}$$

If the number of leaving identifications is zero $\mathcal{N}\left(L^x_{(t)}\right) > 0$ and $\mathcal{N}\left(L^{\text{id}}_{(t)}\right) = 0$ as shown in Fig. 4-17, the current radius of approximated coverage is much smaller than the radius of actual coverage. In this case, the leaving estimated position with the maximum distance from a sensor position is used to determine the adjusted radius by

$$R^L_{(t)} = \min_{x^m_{(t-1)} \in X_{(t-1)}} \left(\text{dist}\left(x^m_{(t-1)}, x_{\text{IS}}\right)\right) + \Delta r, \tag{4.26}$$

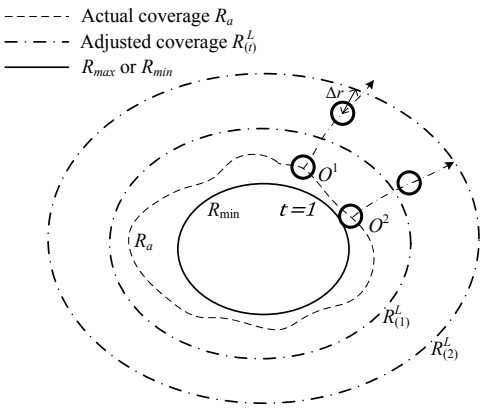



Figure 4-17: Illustration of coverage enlargement when objects leave the coverage of an identification sensor

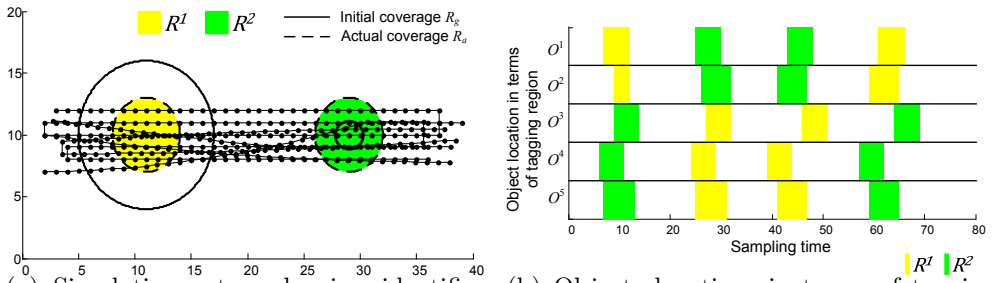
where $\Delta r$ is added for the extra coverage to prevent false associations by the irregular property of actual coverage.

If $R^E_{(t)}$ and $R^L_{(t)}$ conflict with each other, the coverage of an identification sensor needs to be adjusted passively to prevent false associations. Therefore, the final radius $R_{(t)}$ is determined by

$$R_{(t)} = \max(R^E_{(t)}, R^L_{(t)}). \tag{4.27}$$

Moreover, the goal of the coverage adjustment is to prevent a significant discrepancy between the initial approximated coverage and the actual coverage as conserving current association information of objects. Hence, the adjusted radius should not violate the positions of objects having association information.

Fig. 4-18 illustrates a simulation setup showing identification sensors and object trajectories. The range of $R^k_{g,t}$ is between $R_{min}$ $1m$ and $R_{max}$ $6m$. The initial value of $R^1_{g,t}$ is $6m$ and the initial value of $R^2_{g,t}$ is $1m$ for extreme cases. The actual radius $R^k_a$ is $3m$ for both sensors. The simulation assumes that identifications of objects are

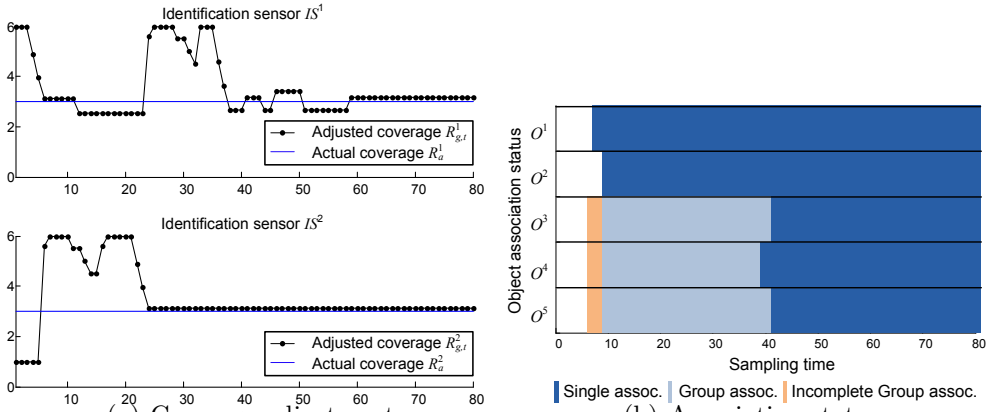



(a) Simulation setup showing identification sensors and objects trajectories.

(b) Objects locations in terms of tagging regions.

Figure 4-18: Illustration of simulation setup for coverage adjustment and objects locations in terms of tagging regions.

perfectly registered within the actual radii of the identification sensors.




(a) Coverage adjustment

(b) Association status

Figure 4-19: Simulation result of coverage adjustment and association status for Fig. 4-18

Fig. 4-19 is the corresponding result of the coverage adjustment and association status for Fig. 4-19. In identification sensor $IS^1$, initial coverage is slowly adjusted as objects enter the coverage. Every time any identifications are not registered, the coverage is changed by (4.18). When the number of entered or left positions may differ from the number of entered or left registrations, the adjusted radius is slowly changed by $\Delta r$. In identification sensor $IS^2$, initial coverage is abruptly changed as

reacting to certain registration by (4.20). The coverage of $IS^2$ also has the similar variation by the mismatched number of positions and identifications. Eventually, the initial radii of the sensors converge on actual coverage as an association rate increases.

### 4.3.5 Association Algorithm

Algorithm 4 summarizes the conditions for multiple objects association with the coverage uncertainty. If $x^m_{(t)}$ is in a group association, possible associable identifications are limited to $G^{\mathrm{ID}}(x^m_{(t)})$. Objects in incomplete group associations also have identification candidates. Therefore, the possibility is increased that an estimated position will be uniquely paired with its identification. After the association system finishes checking the association conditions for each object in the coverage, it determines whether remaining objects are a group association or incomplete group association. Then, the association system removes associated identifications and estimated positions in all the sets of group associations and incomplete group associations. Single associations can also be established in this process if the number of elements in group associations is two.

## 4.4 Evaluation

### 4.4.1 Simulation Setup

Fig. 4-20 shows a simulation configuration which can be applied to a bank or an airport. An object enters and exits through gates where identification sensors are

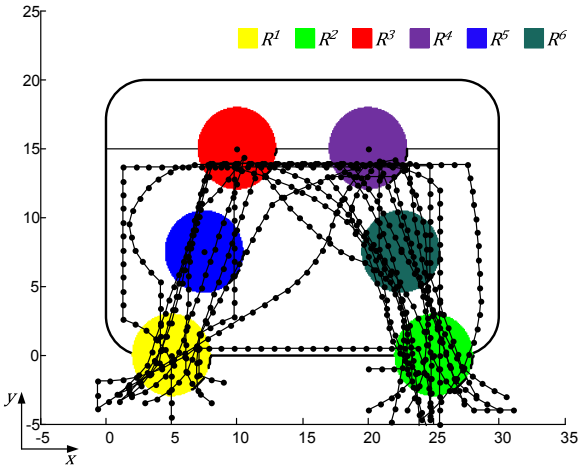



Figure 4-20: Simulation configuration with the trajectories of ten objects (unit : meter).

installed. The colored circular areas are the coverage of identification sensors. $O^1$ starts at $(2, -3)$, $O^2$ at $(5, -4.5)$, $O^3$ at $(9, -2)$, $O^4$ at $(3, -3.5)$, $O^5$ at $(30, -4)$, $O^6$ at $(30.5, -3)$, $O^7$ at $(20, -4)$, $O^8$ at $(8, -3.5)$, $O^9$ at $(23, -4)$ and $O^{10}$ at $(20, -1)$. The identification sensor $R^1$ is placed at $(5, 0)$, $R^2$ at $(25, 0)$, $R^3$ at $(10, 15)$, $R^4$ at $(20, 15)$, $R^5$ at $(7.5, 7.5)$ and $R^6$ at $(22.5, 7.5)$. Every visual sensor approximates the coverage radius of 3 meter. Objects are localized and tracked by visual sensors. The total number of sampling time is 130. In the simulation, the registration of identification is probabilistically determined to reflect the effect of the coverage uncertainty. The sampling interval of the identification sensor is 1 sec and the trajectories of the objects are dotted by also the sampling interval of 1 sec in the figure. At every time interval of identification sensor, each object is registered with probability of 0.5.

The association performance for the identification is compared against the simple association rule. In the simple association rule, a position and an identification of a single object are associated when each signal exists in the sensor coverage [53] [54].

It is assumed that an object is localized and tracked by multiple cameras without failure. We use a simple object tracking algorithm since object models are not known to the association system.

### 4.4.2 Effect of Modeled Region Accuracy

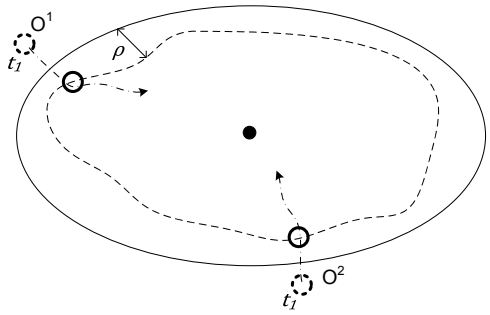


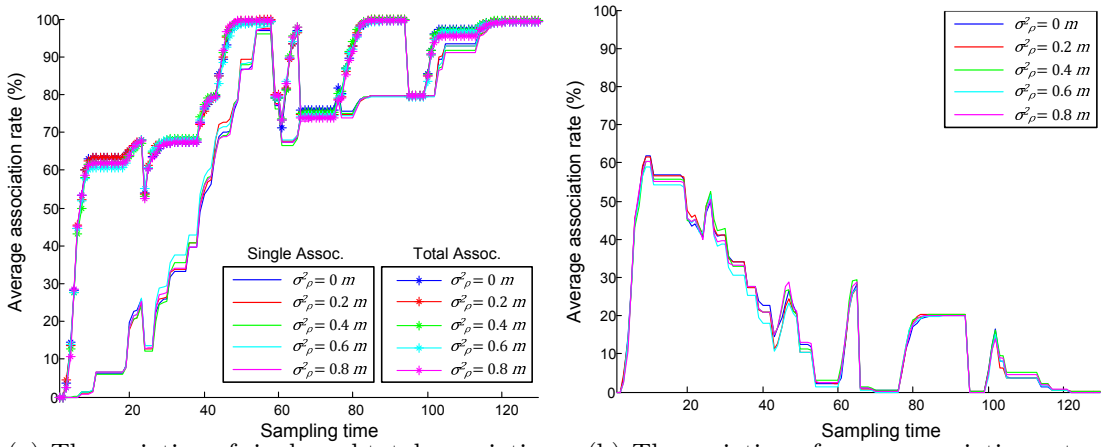

Figure 4-21: Illustration of the effect of modeled region accuracy in association condition.

When an object is associated with its identification by the object dynamics, (4.9) or (4.10) should be satisfied. The necessary condition is that an identification should be registered immediately after an object enters or right before an object leaves the region. However, satisfying these conditions depends on how accurately actual coverage is approximated, as shown in Fig. 4-21. Also, localization errors by visual sensors cause ambiguity in the boundary of coverage. In order to analyze the effect of the modeled region accuracy, we utilize a parameter

$$\rho \sim N(0, \sigma_\rho^2),$$

where $\rho$ is a distance between a modeled boundary and an actual boundary and $\sigma_\rho^2$ is a corresponding variance. The actual size of coverage is determined by adding $\rho$ to

the modeled size of coverage. Only when the position of an object is estimated within the actual size of coverage, an identification is considered registered in the system.



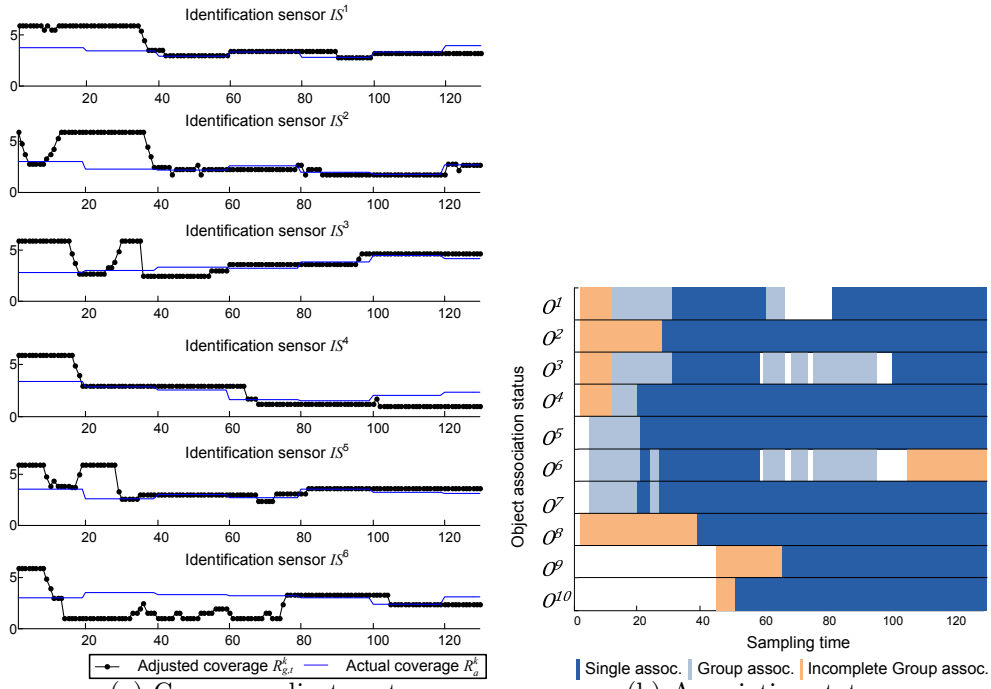

(a) The variation of single and total association rate.

(b) The variation of group association rate.

Figure 4-22: The simulation of the association performance according to the variation of the modeled region.

Fig. 4-22 shows the simulation result of association performance according to the variance of the actual size of coverage. The association simulation is repeated 100 times and the results are averaged in order to reflect the effect of the mismatched coverage model. The result indicates that establishment of group association is affected by the discrepancy between the actual and modeled coverage. However, association performance is not significantly affected by the coverage variance. Group associations are eventually resolved to single associations by the object dynamics.

The coverage adjustment scheme also alleviates the effect of the discrepancy between the actual coverage and the approximated coverage. Fig. 4-23 shows the simulation result of the coverage adjustment scheme with the current simulation configuration. The maximum and minimum radius of each identification sensor is set to be $1m$ and $6m$ respectively while the actual radius of each identification sensor varies




(a) Coverage adjustment (b) Association status

Figure 4-23: Simulation result of coverage adjustment and association status

from the initial radius $3m$ at every 20 sampling times. The amount of each variation is chosen from the uniform distribution $-1m \sim 1m$. The result demonstrates that the approximated coverage of each identification sensor is adaptively adjusted to the actual coverage of each identification sensor. Moreover, as the association rate increases, the accuracy of the approximated coverage also increases since the object dynamics are utilized for the coverage adjustment.

### 4.4.3 Effect of Region Overlapping

Fig. 4-24 shows a case in which identification sensor regions overlap each other due to largely approximated coverage. The overlapped regions may confuse the system. However, it does not affect association performance since each region has its own

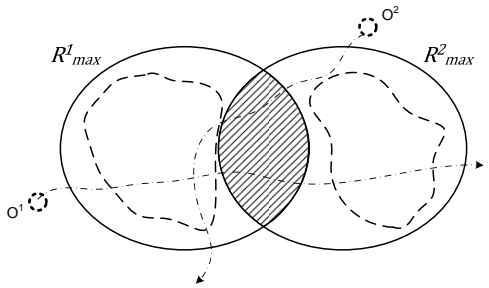



Figure 4-24: Illustration of the effect of region overlapping.

data sets for estimated positions and identifications. Instead, an overlapped region can be considered a separate region. Then, the system has the effect of having one more region. For example, the overlapped region of $R^1$ and $R^2$ is denoted as $R^{\{1,2\}}$. Naturally, the sets of unassociated estimated positions and identifications in this region are represented by $S_t^{x,\{1,2\}}$ and $S_t^{ID,\{1,2\}}$ respectively. The sets for overlapped regions are generated based on initially obtained data,

$$\begin{aligned} S_t^{x,\{1,2\}} &= S^{x,1} \cap S_t^{x,2} \\ S_t^{ID,\{1,2\}} &= S^{ID,1} \cap S_t^{ID,2}. \end{aligned} \tag{4.28}$$

This can increase a case to make a single or group association. However, this cannot significantly improve association performance because it is hard to define an optimal overlapping. Especially, when actual regions are not overlapped, the sets for overlapped regions become useless.

Fig. 4-25 shows a simulation configuration with the overlapped identification sensor regions ($R^{1,5}$ and $R^{2,6}$). Fig. 4-26 shows a corresponding simulation results

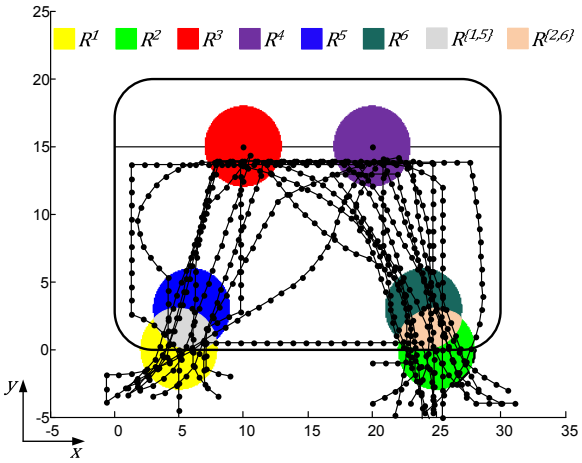
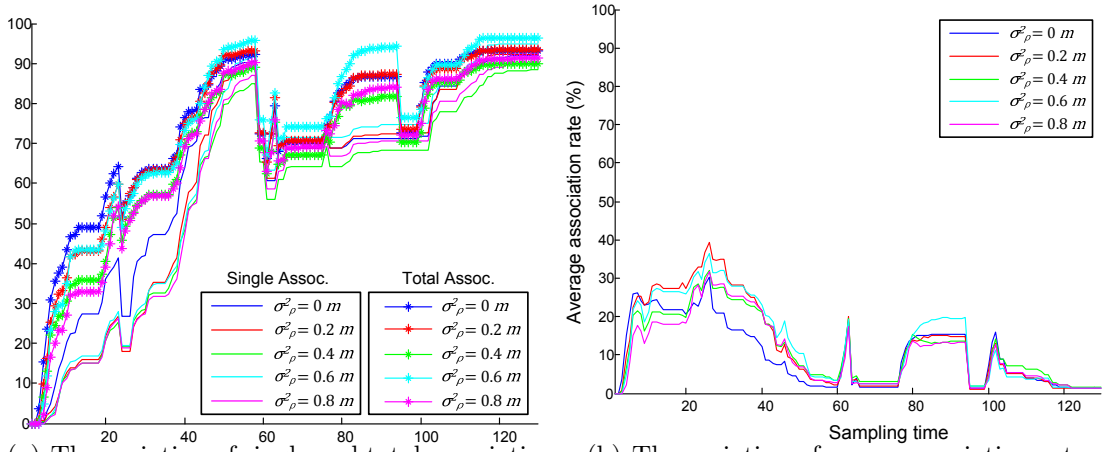



Figure 4-25: Illustration of the simulation configuration with overlapping regions ($R^{1,5}$ and $R^{2,6}$).

for Fig. 4-25. The association simulation is repeated 100 times and the results are averaged in order to reflect the effect of the coverage uncertainty. Since the system can have more regions, single associations can be established faster. However, this does not indicate that a total association rate is increased. Overlapped regions split two sets of data into three sets of data. This can decrease the establishment of group associations depending on object movement patterns and registration performance of identification sensors. Therefore, this scheme has both sides in terms of the association performance.

### 4.4.4 Association Performance

Fig. 4-27 shows how the object association status changes with the coverage uncertainty. Fig. 4-27(a) shows the registered identification of the objects as a function of the time for each identification sensor. Each color corresponds to each coverage of an identification sensor and the white color indicates that an object dose not belong



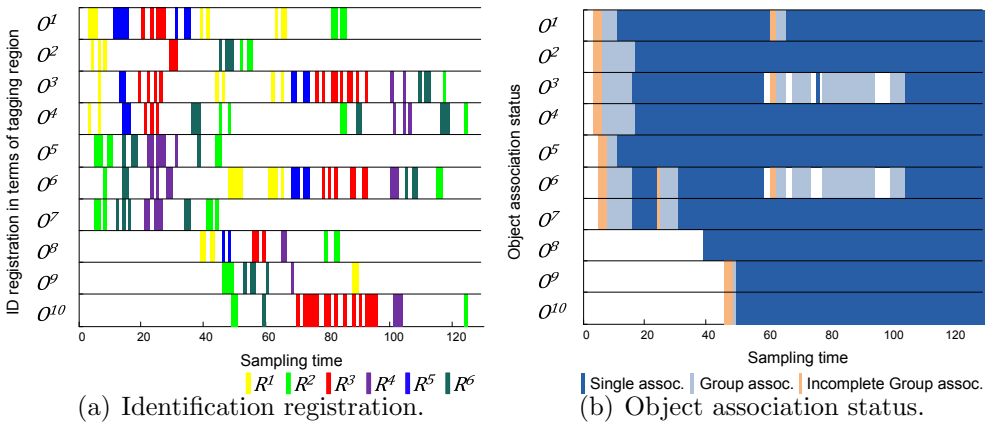

(a) The variation of single and total association rate.

(b) The variation of group association rate.

Figure 4-26: The simulation result of the association performance according to the variance of the modeled region for Fig. 4-25.

to the coverage of any identification sensor. Although identifications of objects are probabilistically registered due to the coverage uncertainty, positions of objects are eventually associated their own identifications as shown in Fig. 4-27(b). Fig. 4-27(c) shows which IDs are registered for each object as a function of the time. Each color corresponds to each ID of an object and objects are eventually associated with their IDs respectively.

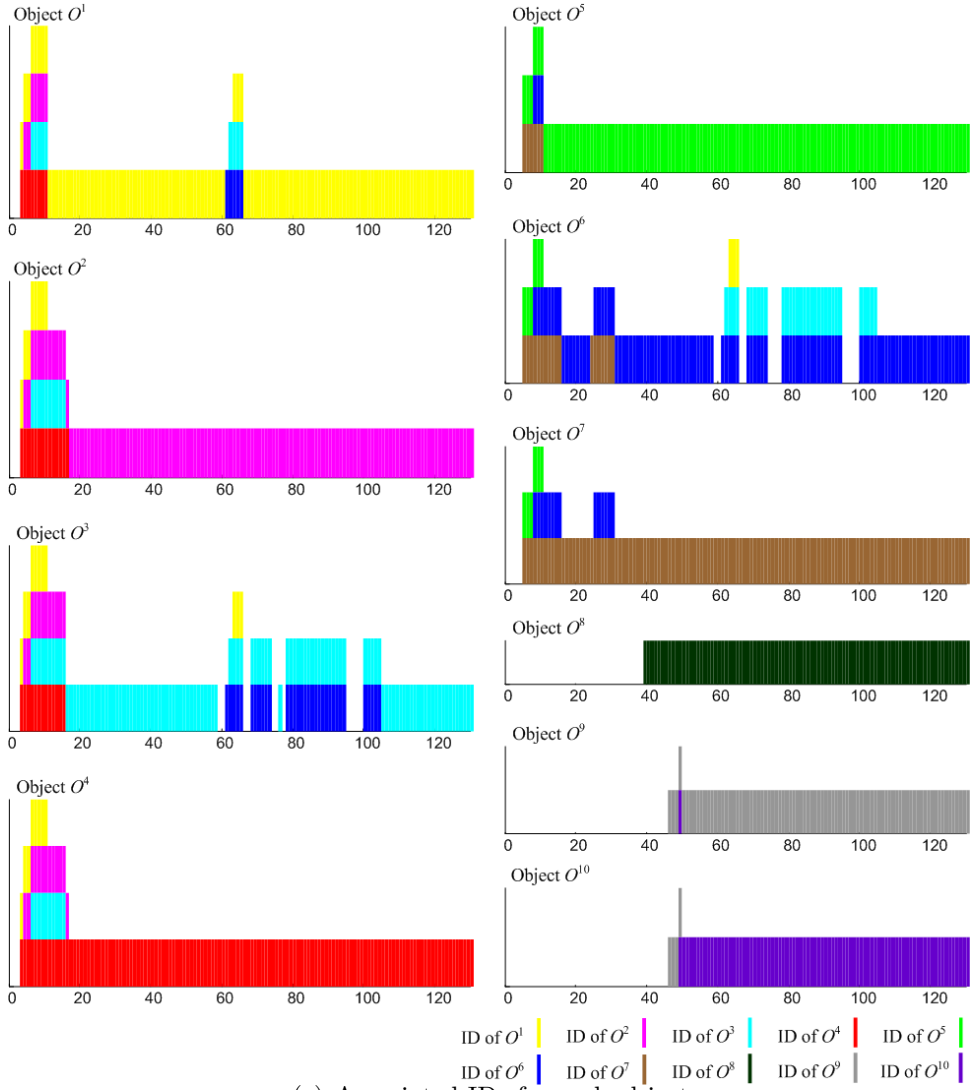
Fig. 4-28 shows the comparison of the association performances between the existing association method and the proposed association method in terms of the tracking performance. The tracking performance is defined as the case when objects tracking fails due to the collision. One case fails in tracking when objects are adjacent within $0.3m$ and the other case uses $0.6m$. When an object fails in tracking due to the collision, it loses all associated identifications regardless of status such as a single association, group association, and incomplete group association. The association simulation is repeated 100 times and the results are averaged in order to reflect the




(a) Identification registration.

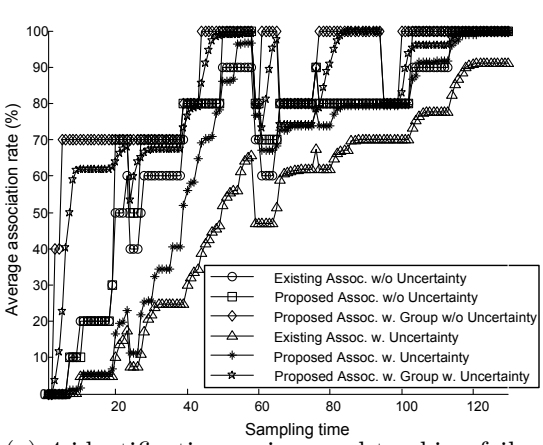



(b) Object association status.

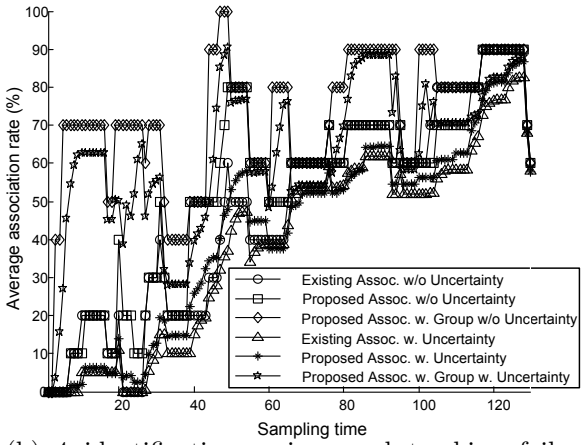



(c) Associated IDs for each object.

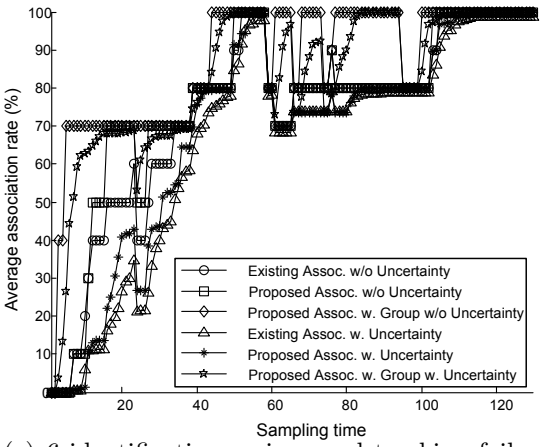
Figure 4-27: Object association status with the inconsistent registration of identifications when the proposed association method is used.



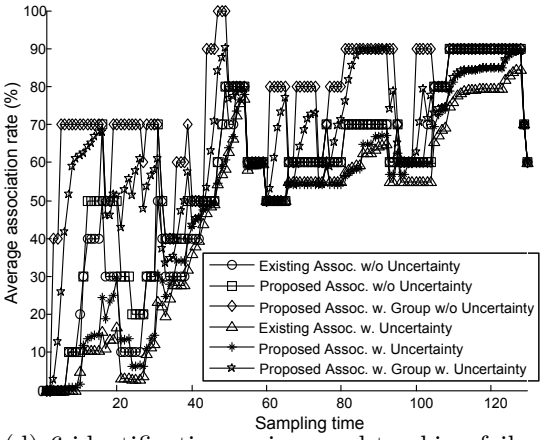

(a) 4 identification regions and tracking failure within $0.3m$.




(b) 4 identification regions and tracking failure within $0.6m$.




(c) 6 identification regions and tracking failure within $0.3m$.




(d) 6 identification regions and tracking failure within $0.6m$.

Figure 4-28: The simulation of association performance comparison in terms of the number of identification regions and the tracking performance.

effect of the coverage uncertainty. The proposed association method establishes single associations faster than the existing method regardless of the effect of the coverage uncertainty. Especially, a group and incomplete group association increases the average association rate. The result also demonstrates that the tracking performance has less influence on the proposed association method in terms of the average association performance. Although objects tracking fails more often, their identifications are maintained by a group or incomplete group association. The result also demonstrates that the proposed method is less vulnerable with an smaller number of identification regions in terms of association performance.

### 4.4.5 Robustness Against False Detection and False Tracking

The proposed method has robustness against two non-ideal phenomena possibly caused by visual sensors. One case involves falsely detected objects according to the classification capability of detection algorithms. When objects are falsely detected inside the region, this leads to a group or incomplete group association. However, this is eventually resolved when the true position of an object is associated with its identification. The other issue is false tracking, which usually occurs when objects collide with each other. Identifications can be switched depending on the tracking capability. In this case, the proposed method utilizes a group association. Then, their identifications are also eventually found by the object dynamics. However, the system cannot clearly determine whether an object have identification information or not because of the coverage uncertainty.

## 4.5 Summary

The data association and management scheme is proposed to complement two different types of signals in heterogeneous sensor environment. Visual sensors estimate and track positions of objects, and identification sensors register identifications of objects. The uncertain sensing coverage of an identification sensor is approximately modeled for a simple association strategy. The location information of identification sensors and objects is utilized to resolve the association problems with the object dynamics. We also present a coverage adjustment method using the object dynamics around the coverage of the identification sensor. The simulation based analysis shows that the association performance is improved as the time elapses even with realistic problems such as error of estimated positions, a discrepancy between approximated and actual identification sensor overage, variance of actual identification sensor coverage, and imperfect tracking. To improve the association performance, the identification sensors should be installed at the places where objects dynamically move around for a fast association establishment or recovery, as the associations are established by the object dynamics of crossing the coverage of identification sensors.

**Algorithm 4**: The proposed association algorithm

**repeat**
- Estimate $x_{(t)}^{m}$ of all detected objects by visual sensors at $t$
- Register $\mathrm{ID}_{(t)}^{m}$ by identification sensors at $t$
- Generate $S_{(t)}^{x}$ and $S_{(t)}^{\mathrm{ID}}$ at $t$
- **for** $m = 1$ *to* $\mathcal{N}\left(S_{(t)}^{x}\right)$ **do**
  - **if** $\mathcal{N}\left(S_{(t)}^{x}\right) = \mathcal{N}\left(S_{(t)}^{ID}\right) = 1$ **then**
    - $x_{(t)}^{m}$ and $\mathrm{ID}_{(t)}^{l}$ are associated
  - **else**
    - **if** $G^{x}\left(x_{(t)}^{m}\right) \neq \phi$ **then**
      - **if** *the entering condition in the group association is satisfied* **then**
        - $x_{(t)}^{m}$ and $\mathrm{ID}_{(t)}^{l}$ satisfying above conditions are associated
        - Remove them in group associations
      - **end**
    - **else**
      - **if** *the condition in (4.12)* **then**
        - $x_{(t)}^{m}$ and $\mathrm{ID}_{(t)}^{l}$ satisfying above conditions are associated
        - Remove them in incomplete group associations
      - **end**
    - **end**
  - **end**
- **end**
- **for** $m = 1$ *to* $\mathcal{N}(S_{(t-1)}^{x})$ **do**
  - **if** *the leaving condition in the group association is satisfied* **then**
    - $x_{(t)}^{m}$ and $\mathrm{ID}_{(t)}^{l}$ satisfying above conditions are associated
    - Remove them in group associations
  - **end**
- **end**
- **if** *The remaining objects satisfy the condition in (4.3)* **then**
  - Register them as a single association
- **else if** *The remaining objects satisfy the condition in (4.5)* **then**
  - Register them as a group association
- **else**
  - Update candidate identifications of objects in group associations
- **end**

**until** *Association system stops*

# Chapter 5

# Self Localization Method for Mobile Sensors

## 5.1 Introduction

Visual information based Self-localization has advantages over above methods in two-folds. First, it does not require active landmarks (i.e. reference objects) such as beacon and also natural objects can serve as landmarks. The other advantage is that it is more effective and reliable in dynamic environment as the sensible range is not limited by the line-of-sight. These advantages have fostered much efforts in research community of navigation application [59] [60] [61].

Reference objects can be either artificial or natural objects [62] [63] [64] [65]. We assume that reference objects can be uniquely identified by a mobile robot. The global coordinates of landmarks or reference objects are known to mobile robot. There were extensive researches about robot localization with known reference objects. The

conventional self localization algorithms use the bearings of the landmarks relative to each other. This is called "visual angle" formed by the rays from a mobile robot to each reference point [66]. The location of a mobile robot is estimated by finding the intersection point of the circles passing the reference point. In the ideal case, the mobile robot is localized with three landmarks. However, although the matched landmarks from the image are found in the known map, the visual angle is usually distorted by the nonlinear property of a camera lens. Thus, the solution is estimated by minimizing the error of all possible landmarks pairs and the estimation error is minimized in proportion to the number of landmarks pairs [62] [67]. Moreover, multiple solutions exist when all landmarks form a circle. Another approach uses the perspective projection model to identify the relationship between the view point and the landmark. Although this method is simple in calculation, the performance is also deteriorated by the projection method without calibrating the camera nonlinearity [65] [68].

In this chapter, we propose a self localization method using a single visual image with the simple iteration technique. We assumed that reference objects can be reliably extracted and identified. The proposed method identifies the relationship between the landmarks on the image and the known global reference points by the parallel projection model. The parallel projection model calibrates the non-linearity of optical lens distortion without the computational complexity. The coordinates and the orientations are estimated with minimum relation equations by the simple iteration method. Our method can be used in large area with artificial or natural references as long as their coordinates are known and they can be reliably identified.

The possible error source of the self localization method is explained and analyzed in terms of the performance of the self localization.

The rest of this chapter is organized as follows. Section 5.2 discusses background on self localization and problem description. In Section 5.3, parallel projection model and its basic concept are discussed. Section 5.4 proposes a self localization algorithm for determining the coordinates and the orientation from external reference points. In Section 5.5, we present an experiment and the analysis of simulation results with extensive analysis of the effect of the error of the measurement on the performance of the algorithm. Section 5.6 summaries the chapter.

## 5.2 Background and Problem Description

### 5.2.1 Related Works on self-localization

In general, self localization with visual information is related to photometry. [69] presents a general method for determining the three-dimensional position and orientation of an object relative to a camera based on a two-dimensional image of known feature points located on the object. [70] analytically deals with the perspective $n$-point (PnP) problems with four correspondence of scene objects. Our approach does not analytically solve matrix transformation, but calculates the orientation and the location using computationally efficient iterative algorithm.

A simple method for localization which allows a robot to determine its absolute position with a view of single landmark in one image is presented in [65]. The land-

mark is chosen as the intersection of natural lines easily found in indoor environment such as edge of doors or walls. Another localization algorithm which is based on comparing the images taken in advance and taken during navigation is discussed in [62]. In this scheme, the shape and the coordinate of images are stored in memory efficient format for quick retrieval and comparison. Similar method is presented where planar landmarks such as posters are used in visual localization of a mobile robot in indoor environment [71]. This algorithm has a restriction on the shape of landmark and is not suitable in an open area. Similar to these approaches, our assumption requires a map of global coordinates of reference objects that can either be natural or artificial objects.

Scale Invariant Feature Transform (SIFT) developed for image feature generation in object recognition application is used for robot localization in [72]. The invariant characteristics of SIFT are captured by three images and stereo-matched to elect landmark that was later used to compute 3-D world coordinate relative to the robot. This algorithm does not require modification of the environment or map of reference objects, but needs more than two cameras with expensive computation to compare the features in the image.

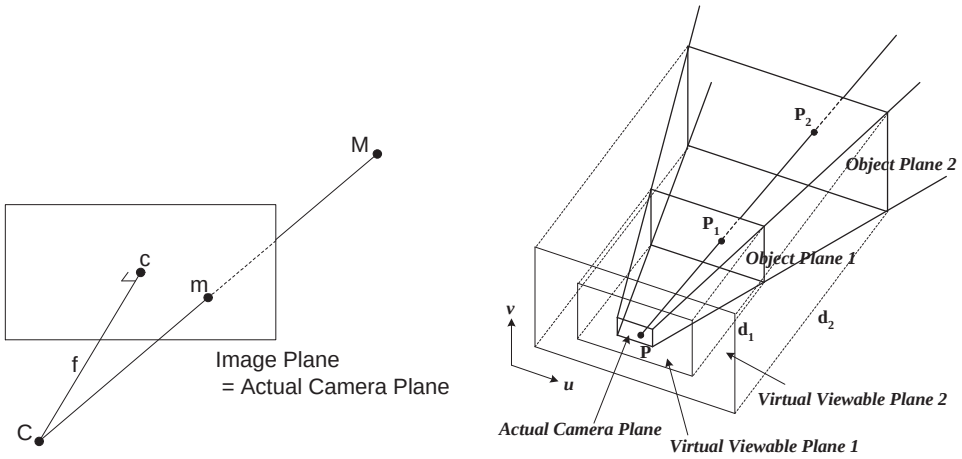
In this chapter, we use the parallel projection model to simplify the computational complexity in determining the location and the orientation of mobile robot. In the parallel projection model, a *virtual viewable plane* is defined to formulate the relationship between a real object and an actual camera. By using this model, the minimum number of relation equations between the detected landmarks on the image and known corresponding reference points are obtained. This enables to decrease the

computational complexity in finding the solution from all possible landmark pairs with the minimum mean square error.

The parallel projection model is similar to the pin-hole camera model in perspective projection model since the camera angle is relatively narrow and the size of image sensor is much smaller than real object area [73]. However, in the perspective projection model, the calibration process usually uses a flat plate with a regular pattern or the known several reference points [74] [75] [76]. In addition, the calibration information should be updated for projection accuracy whenever the camera status is changed. This is not appropriate for the highly dynamic application such as the robot navigation. On the other hand, the parallel projection model uses zoom factor instead of focal length and scale factor. This model simplifies the calibration process of the camera non-linear property by using the pre-obtained calibration table. Therefore, it is easily applied to the reference projection in the robot navigation. As a result, the relation equations are less affected by the camera model. This minimizes the self localization error due to the projection model. Fig 5-1 illustrates the difference of our parallel projection model and the perspective projection model [77] where $d_1$ and $d_2$ denote the distance of camera and *Object Plane*. *Object Plane* contains the object and is parallel to *Actual Camera Plane*.

### 5.2.2 Problem Overview

In this chapter, we define the term 'mobile sensor' to describe a mobile robot with visual sensor. We assume that mobile sensor can identify the reference objects and




(a) Illustration of the Perspective Projection Model

(b) Parallel Projection Model

Figure 5-1: The difference of the perspective projection model and the parallel projection model.

their coordinates are known (i.e., stored in a database). We limit our discussion to the self localization problem and the method of how to identify such objects is not considered. The mobile sensor navigates by itself and visual image is obtained periodically. Based on the captured image data, the self localization comprises of determining both the orientation and the location of mobile sensor. We use global coordinate system, the origin of which is arbitrarily chosen but used to represent the coordinates of the reference points and the location of the mobile sensor. The objective is to utilize the projected reference points to determine the location of the mobile sensor. In this chapter, we focus on two aspects of the proposed method. The first aspect is to maintain accuracy of the self localization and the second aspect is to maintain computational efficiency.

Since our method uses captured image data through the application of a typical digital imaging device, several sources of error are possible. Since the proposed ap-

proach relies on the point that is chosen from an area of pixels which is the projected image of the reference object, there can be inherent errors from image processing that selects one point from the area of an object image. This error can vary depending on the many factors such as distance from mobile sensor to reference objects, distance between reference objects, etc. In addition, non-linearity of the lens of imaging device causes shifting of projected point when the distance to reference points is not known. This shifting also affects the fidelity of self localization if compensation is not done.

Since the mobile sensor changes its location and orientation continuously, the reference points may be changed accordingly. The self location method should be computationally efficient by effectively utilizing available reference points. As we will show later in this chapter, the selection of reference points affects the self localization errors. When more than three reference points are inside the viewable range of the mobile sensor at the same time, mobile sensor has freedom to choose the reference objects in such a way that can minimize such error. Therefore, multiple reference objects should be strategically distributed to harness self localization of individual mobile sensor. A computationally efficient iterative algorithm using the relationship between the location of reference points is proposed.

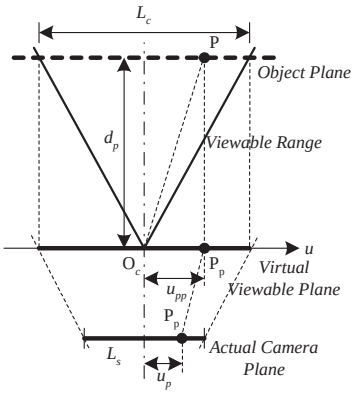



Figure 5-2: Illustration of the parallel projection model and the relationship with the actual image projected on visual sensor (Camera).

## 5.3 Characterization of Viewable Images

### 5.3.1 Basic Concept of Parallel Projection Model

In this section, we introduce *parallel projection model*. In order to simplify the process of projected image on the camera device, we define three planes: the *object plane*, the *virtual viewable plane*, and the *actual camera plane* as shown in Fig. 5-2. An object $P$ which is in the viewable area of mobile sensor, is considered to be on the *object plane*. As opposed to the traditional model of a camera, in a parallel projection model, the object $P$ is projected in parallel onto *virtual viewable plane* and the projected point is denoted as $P_p$. The *Virtual viewable plane* is parallel to the *object plane* with distance $d_p$. $L_c$ denotes the length of the *object plane*, which is the length of viewable area at distance $d_p$. $L_s$ denotes the length of *actual camera plane* on which the measurement of projected image is done. The positions of the projected object on a *virtual viewable plane* and an *actual camera plane* is denoted as $u_{pp}$ and $u_p$, respectively.

In the parallel projection model, a real object located on an *object plane* is pro-

jected to an *actual camera plane* through a *virtual viewable plane*. Hence, as formulated in (5.1), $u_{pp}$ is expressed as $L_c$, $L_s$ and $u_p$ using the proportionality of the length of a *virtual viewable plane* and an *actual camera plane*.

$$u_{pp} = u_p(\frac{L_c}{L_s}). \tag{5.1}$$

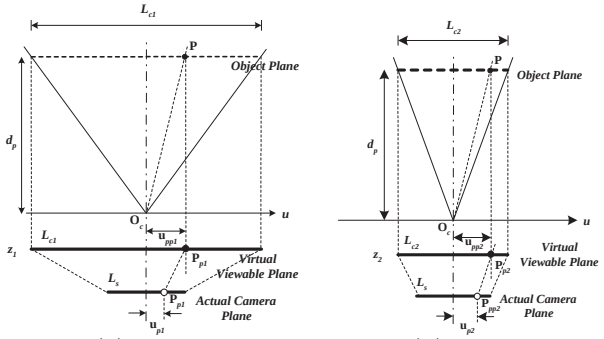
The position of real object can be obtained from $u_{pp}$ and $d_p$, once the ratio of $L_c$ and $d_p$ is known. This ratio is defined to be zoom factor, $z$, which is the property of the image device.

$$z = \frac{d_p}{L_c}. \tag{5.2}$$

Fig. 5-3 shows the effect of different zoom factors. Although the positions of $u_{p1}$ and $u_{p2}$ on an *actual camera plane* are different with different zoom factors, $z_1$ and $z_2$, the positions of $u_{pp1}$ and $u_{pp2}$ on *virtual viewable plane* remain same. From (5.1), $u_{pp1}$ is expressed as $u_{p1}(L_{c1}/L_s)$ and $u_{pp2}$ is $u_{p2}(L_{c2}/L_s)$. Since the positions $u_{pp1}$ and $u_{pp2}$ are same, $u_{p1}$ can be expressed as $u_{p2}(L_{c2}/L_{c1})$. Also, since $z_1 = d_p/L_{c1}$ and $z_1 = d_p/L_{c1}$ from (5.2), the position of the projected object on *actual camera plane* with one zoom factor can be expressed with the position with the other zoom factor.

$$u_{p1} = u_{p2}\frac{z_2}{z_1}. \tag{5.3}$$

Fig. 5-4 illustrates the case where two objects of $P_1$ and $P_2$ on different *object plane*, that is, at different distance, appear as a single point on *actual image plane*.




(a) $z = z_1$ (b) $z = z_2$

Figure 5-3: Illustration of zooming effects on the *Virtual Viewable Plane*.

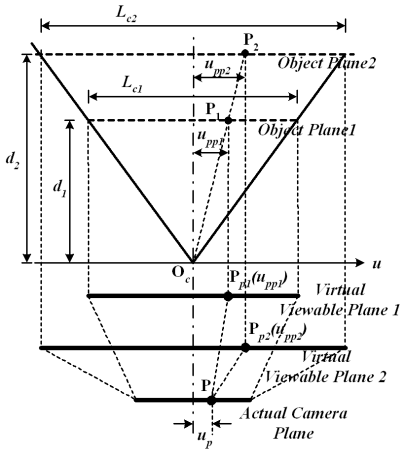



Figure 5-4: Illustration of zooming model in the Parallel Projection Model.

The objects are considered to be projected to $u_{pp1}$ and $u_{pp2}$ on two different *virtual viewable plane*s, but meet at the same point on an *actual camera plane*. From equation (5.1), $u_{pp1}$ is expressed as $u_p(L_{c1}/L_s)$ and $u_{pp2}$ as $u_p(L_{c2}/L_s)$. For this case, the following condition

$$\frac{u_{pp1}}{d_1} = \frac{u_{pp2}}{d_2} \tag{5.4}$$

should be met. Here, we used the definition of zoom factor, $L_{c1} = d_1/z$ and $L_{c2} = d_2/z$ from (5.2).

### 5.3.2 Relationship of Reference Points on Different Planes

Given the parameters of visual sensor, $z$ and $L_s$, we can derive the relationship between a projected reference point on the *virtual viewable plane* and one on the *actual camera plane* with their distance to the origin of each plane. The origin of each plane is defined to be the cross point between a plane and its perpendicular line, the view axis, which also crosses the location of the mobile sensor. Specifically, the origin of the *actual camera plane* is the axis of panning. In Fig. 5-5, the origin on the actual camera plane is denoted as $O_c$ and the origins of the virtual viewable planes are denoted as $O_{v1}$ and $O_{v2}$, respectively. Even though the planes are rotated as visual sensor is panned, the relationship based on the distance on each plane remains unchanged. When $p_1$ and $p_2$ denote the distance to the view axis on each virtual plane, and $i_1$ and $i_2$ denote the corresponding measurement on the *actual image plane*, using (5.1) and (5.2), we can derive

$$\begin{aligned} \frac{p_1}{i_1} &= \frac{L_{p1}}{L_s} = \frac{D_1}{z_1 L_s}, \\ \frac{p_2}{i_2} &= \frac{L_{p2}}{L_s} = \frac{D_2}{z_2 L_s}, \end{aligned} \tag{5.5}$$

where $z_1$ and $z_2$ are the zoom factors of mobile sensor corresponding to distance, $D_1$ and $D_2$, from the *actual camera plane* to the *object plane* for each reference point. $L_s$ is the size of the image on which $i_1$ and $i_2$ are measured.

In practice, the location of a projected point on the image device is obtained from the image processing of the target objects such as edge detection and/or feature

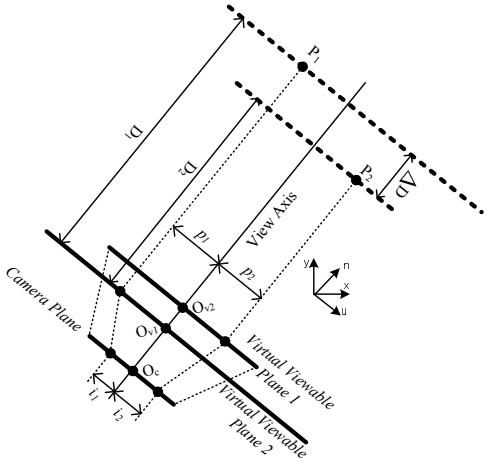



Figure 5-5: Illustration of the self localization with two reference points.

extraction. Thus, the projected point on the image device usually contains some uncertainty. In later sections, how this uncertainty affects self localization algorithm is discussed in detail.

## 5.4 Self Localization Algorithm

### 5.4.1 Self Localization with Known Orientation

In this section, we introduce self localization when two reference points and the orientation of visual sensor are known. We define $\theta$ as the angle formed between the camera plane and global $x$ axis. We define a unit vector $\hat{u}$ to have the direction of the camera plane and $\hat{n}$ to be the unit vector along the view axis, the direction to which the visual sensor is facing. Therefore, $\theta$ is the angle between the $x$ axis and $\hat{u}$. Using the following equations, we can obtain values $p_1$, $p_2$, $D_1$ and $D_2$ in (5.5). In the previous section, we described $i_1$ and $i_2$ as the distance to the view axis on the camera plane, but, from now on, they are considered to be able to have negative

values when the projected reference point is in the left side of the view axis. It does not change the distance relationship described in the previous section by allowing $p_1$ and $p_2$ to have negative values as well when they are also in the left side of the view axis.

For $p_1$ and $p_2$,

$$\begin{aligned} p_1 &= \overrightarrow{CP_1} \cdot \hat{u}, \\ p_2 &= \overrightarrow{CP_2} \cdot \hat{u}. \end{aligned} \tag{5.6}$$

For $D_1$ and $D_2$,

$$\begin{aligned} D_1 &= \overrightarrow{CP_1} \cdot \hat{n}, \\ D_2 &= \overrightarrow{CP_2} \cdot \hat{n}. \end{aligned} \tag{5.7}$$

Here, $\overrightarrow{CP_1}$ and $\overrightarrow{CP_2}$ denote the vectors from the location of the mobile sensor, $O_c$, to each reference point. Above the two sets of equations are simply the decomposition of $\overrightarrow{CP_1}$ and $\overrightarrow{CP_2}$ to the $\hat{u}$ and $\hat{n}$ components. Then, from (5.5), we have

$$\begin{aligned} \frac{p_1}{i_1} &= \frac{\overrightarrow{CP_1} \cdot \hat{u}}{i_1} = \frac{\overrightarrow{CP_1} \cdot \hat{n}}{zL_s}, \\ \frac{p_2}{i_2} &= \frac{\overrightarrow{CP_2} \cdot \hat{u}}{i_2} = \frac{\overrightarrow{CP_2} \cdot \hat{n}}{zL_s}. \end{aligned} \tag{5.8}$$

Or,

$$\begin{aligned}\overrightarrow{CP_1}\cdot(\hat{u}-\frac{i_1}{zL_s}\hat{n})=0,\\ \overrightarrow{CP_2}\cdot(\hat{u}-\frac{i_2}{zL_s}\hat{n})=0.\end{aligned}\tag{5.9}$$

The first terms, $\overrightarrow{CP_1}$ and $\overrightarrow{CP_2}$, of the dot product can be expressed with their $x$ and $y$ component in global $x$-$y$ coordinate system as

$$\begin{aligned}\overrightarrow{CP_1}=(P_{1x}-x_c)\hat{x}+(P_{1y}-y_c)\hat{y},\\ \overrightarrow{CP_2}=(P_{2x}-x_c)\hat{x}+(P_{2y}-y_c)\hat{y},\end{aligned}\tag{5.10}$$

where $P_{1x}$ and $P_{2x}$ are the $x$ components of $P_1$ and $P_2$, respectively, and $P_{1y}$ and $P_{2y}$ are the $y$ components of $P_1$ and $P_2$, respectively.

The $x$ and $y$ components of the second terms of the dot products are expressed by

$$\begin{aligned}(\hat{u}-\frac{i_1}{zL_s}\hat{n})\cdot\hat{x}=\cos\theta+\frac{i_1}{zL_s}\sin\theta,\\ (\hat{u}-\frac{i_1}{zL_s}\hat{n})\cdot\hat{y}=\sin\theta-\frac{i_1}{zL_s}\cos\theta,\\ (\hat{u}-\frac{i_2}{zL_s}\hat{n})\cdot\hat{x}=\cos\theta+\frac{i_2}{zL_s}\sin\theta,\\ (\hat{u}-\frac{i_2}{zL_s}\hat{n})\cdot\hat{y}=\sin\theta-\frac{i_2}{zL_s}\cos\theta.\end{aligned}\tag{5.11}$$

Then, equations (5.9) are equivalent to

$$(P_{1x} - x_c)(\cos\theta + \frac{i_1}{z_1 L_s}\sin\theta) + (P_{1y} - y_c)(\sin\theta - \frac{i_1}{z_1 L_s}\cos\theta) = 0,$$
$$(P_{2x} - x_c)(\cos\theta + \frac{i_2}{z_2 L_s}\sin\theta) + (P_{2y} - y_c)(\sin\theta - \frac{i_2}{z_2 L_s}\cos\theta) = 0.$$

Let us introduce intermediate variables to simplify the final equations for $x_c$ and $x_y$. They are

$$\begin{aligned}
\alpha_1 &= \cos\theta + \frac{i_1 \sin\theta}{z L_s}, \\
\beta_1 &= \sin\theta - \frac{i_1 \cos\theta}{z L_s}, \\
\alpha_2 &= \cos\theta + \frac{i_2 \sin\theta}{z L_s}, \\
\beta_2 &= \sin\theta - \frac{i_2 \cos\theta}{z L_s}.
\end{aligned}$$

Thus, we can obtain the coordinate of mobile sensor expressed as

$$\begin{aligned}
x_c &= -\frac{\alpha_1\beta_2 P_{1x} + \beta_1\beta_2 P_{1y} - \beta_1\alpha_2 P_{2x} - \beta_1\beta_2 P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}, \\
y_c &= -\frac{\alpha_1\alpha_2 P_{1x} - \alpha_2\beta_1 P_{1y} + \alpha_1\alpha_2 P_{2x} + \alpha_1\beta_2 P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}.
\end{aligned} \tag{5.12}$$

Since the reference object is projected onto the camera plane, the coordinates of reference objects correspond to the centroid of the reference objects. Then, we can obtain the coordinate of mobile sensor using (5.12). However, even though the coordinates of reference points are accurately known in advance, the measurement $i_1$

and $i_2$ on the image may not be corresponding to the true reference points. Possible sources of the uncertainties may arise from the pixel resolution of the image planes as well as incorrect determination of the centroid of the detected reference shape. This uncertainty is evident even with perfect lens view characteristics. We will introduce the effect of non-linearity of camera lens in the later section.

### 5.4.2 Orientation Determination

Thus far, we have considered determining the position of mobile sensor when its orientation is given. However, it is necessary to determine the orientation of mobile sensor as well as its position. Determining both position and orientation concurrently requires a third reference point. From the parallel projection model, using (5.5), we can obtain the angle of the line that crosses the center of the camera plane and the reference point, where the angle is formed between the line and the camera plane. With two reference points, we have two lines with known angle respect to the camera plane and we know each reference point is on one of them. Since there are infinite numbers of ways to position a line segment having two reference points as vertexes sitting on those lines, we cannot determine the position and the orientation of mobile sensor with two reference points. With one more reference point, the problem becomes to position three vertexes of a triangle with known length onto three lines with a known angle. There is only one way to position the triangle in such way if we limit the orientation of mobile sensor to 180° range. From the above, we can conclude that three reference points are enough for determining both the orientation and the location of

mobile sensor when the general direction of the reference points are assumed in the following discussion.

We can find a solution by solving three simultaneous solutions using (5.12), but its non-linearity requires large computational complexity to be implemented on resource limited devices, such as mobile robot. Instead, we developed an effective iteration algorithm which involves solving only two simultaneous equations and the solution is given in (5.12). In our iteration approach, we determine the orientation of the mobile sensor. Once we found the orientation, we obtain the location of mobile sensor using (5.12).

For a given orientation angle, $\theta$, using (5.12), we can obtain two sets of coordinates, ($x_{c1}$, $y_{c1}$) and ($x_{c2}$, $y_{c2}$) using two pairs of reference points out of three. When three reference points, $P_1$, $P_2$ and $P_3$ are chosen for self-localization, using $P_1$ and $P_2$, we have

$$\begin{aligned} x_{c1} &= -\frac{\alpha_1\beta_2 P_{1x} + \beta_1\beta_2 P_{1y} - \beta_1\alpha_2 P_{2x} - \beta_1\beta_2 P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}, \\ y_{c1} &= -\frac{\alpha_1\beta_2 P_{1x} + \beta_1\beta_2 P_{1y} - \beta_1\alpha_2 P_{2x} - \beta_1\beta_2 P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}, \end{aligned} \tag{5.13}$$

and by using another pair, $P_2$ and $P_3$, we have

$$\begin{aligned} x_{c2} &= -\frac{\alpha_2\beta_3 P_{2x} + \beta_2\beta_3 P_{2y} - \beta_2\alpha_3 P_{3x} - \beta_2\beta_3 P_{3y}}{\beta_2\alpha_3 - \alpha_2\beta_3}, \\ y_{c2} &= -\frac{\alpha_2\beta_3 P_{2x} + \beta_2\beta_3 P_{2y} - \beta_2\alpha_3 P_{3x} - \beta_2\beta_3 P_{3y}}{\beta_2\alpha_3 - \alpha_2\beta_3}. \end{aligned} \tag{5.14}$$

In order to develop an effective iterative strategy, we investigate the behavior of the differences of the two coordinates, $d_{cx} = x_{c1}$ - $x_{c2}$ and $d_{cy} = y_{c1}$ - $y_{c2}$ while varying the angle of orientation. We define *error_distance* as

$$error\_distance(\theta) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}, \tag{5.15}$$

where $\theta$ is the angle of mobile sensor's orientation. Fig. 5-6 shows the behavior of this function, as $\theta$ varies from 0° to 180°. The figure shows a case when the true orientation of mobile sensor is 80°, and, at this angle, *error_distance*($\theta$) becomes zero. We call this angle *solution point*. Around this point, the function is found to be symmetric and periodical with 180°. Based on the characteristics of the simulated distance error function, there is always one global minimum between 0° to 180°. Thus, there will be two possible orientations between 0° to 360°. The good initial point is approximated by the direction which a robot moves toward.

If we start iteration inside 45° range from *solution point*, and if we follow down the slope, it is guaranteed to find the solution. In order to find such initial iteration point, $i_0$, inside the range, we arbitrarily choose two angles separated with 90°. Since one of them will be inside 45° range from the *solution point* and the other one will be outside, we simply choose the angle that gives smaller *error_distance*($\theta$) as our initial iteration angle, $\theta_0$.

Once we choose an initial point, we have the initial iteration point, $i_0$, determined by $\theta$ and *error_distance*($\theta$). In order to estimate the slope at that point, another *error_distance* function is evaluated using an $\theta_{-1}$ which is chosen to be very close to

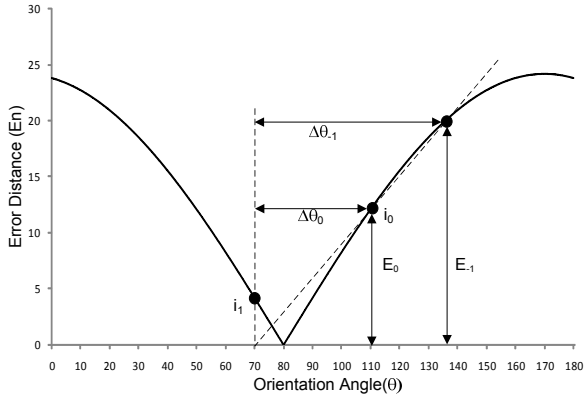



Figure 5-6: Illustration of the distance error function as a function of orientation error. The slope estimations of initial iteration points are shown.

$\theta_0$ such as 0°. We call this estimated slope as $slope_0$ and the relationship of the initial iteration variables are

$$\begin{aligned} \Delta\theta_0 &= \theta_0 - \theta_{-1}, \\ \Delta E_0 &= E_0 - E_{-1} \\ slope_0 &= \frac{\Delta E_0}{\Delta\theta_0}, \end{aligned} \tag{5.16}$$

where $E_n = error\_distance(\theta_n)$.

Depending on the sign of the estimated slope, we choose the direction of the iteration, $dir$. If $slope_0 > 0$, we set $dir_0 = -1$, and, swap $\theta_0$ with $\theta_{-1}$, and, $E_0$ with $E_{-1}$. Otherwise, $dir_0 = 1$.

First, by assuming the slope at our initial point being close to be linear, we choose the next angle where the slope line crosses $x$-axis. Since the slope increases as approaching to the *solution point*, the next iteration step will overshoot albeit very close to *the solution point*. As shown in Fig. 5-6, the *error_distance* function

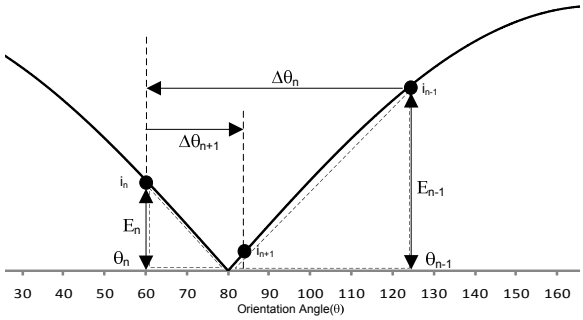



Figure 5-7: Illustration of the convergence steps of the iteration algorithm.

evaluated at $\theta_1$ is the other side of the *solution point*.

From the second step, instead of using the slope, we choose the angle of next iteration step based on the two previous angle and *error_distance* evaluated with them. In this case, since the two triangles shown in Fig. 5-7 are approximately proportional near to the solution point, the angle for the next step is evaluated by the following equations. From

$$\frac{\Delta\theta_n - \Delta\theta_{n+1}}{E_{n-1}} = \frac{\Delta\theta_{n+1}}{E_n}, \tag{5.17}$$

the next iteration angle is calculated as

$$\Delta\theta_{n+1} = dir * \frac{E_n}{E_n + E_{n-1}} * \Delta\theta_n. \tag{5.18}$$

The iteration continues until, the change of estimated orientation, $\Delta\theta_n$, becomes smaller than the threshold value, $\epsilon$. Otherwise, we change the direction, $dir = dir*-1$ and continue. Fig. 5-8 shows that the algorithm is converging very rapidly. The figure shows the iteration algorithm is applied when three initial estimation angles are used, 10°, 40° and 80°. The value of $error\_disance(\theta)$ is plotted at each iteration step.

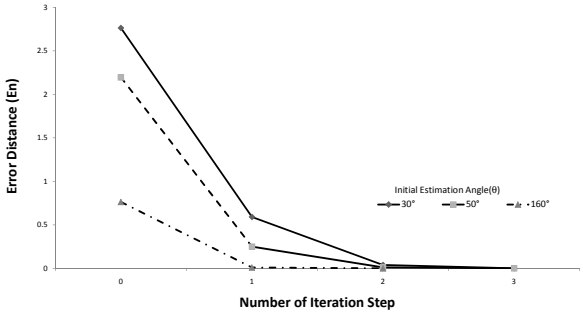


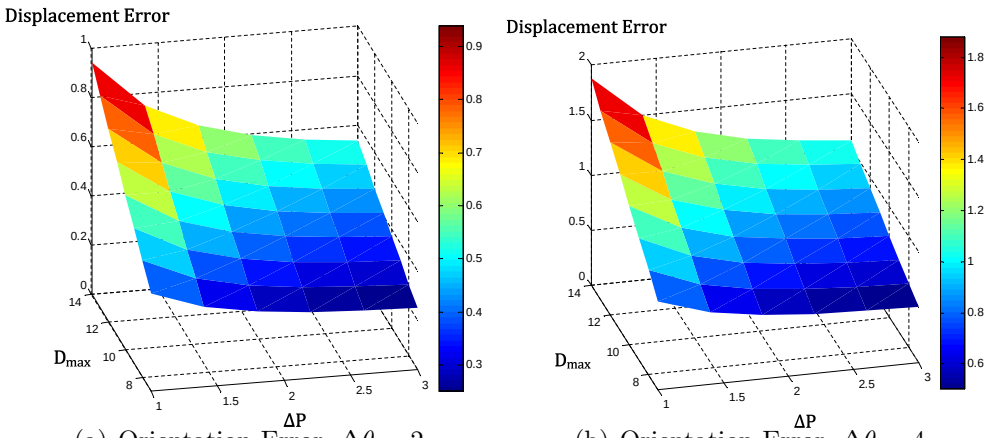

Figure 5-8: Illustration of convergence of the iteration algorithm as a function of the number of iterations.

Note that the iteration starts with two initial iterations (i.e., as shown in the figure, the iteration starts at -1 index).

Fig. 5-9 illustrates the importance of the orientation error on the localization. The plot shows the displacement error for several orientation errors. Throughout this chapter, the displacement error is defined as

$$\sqrt{(x_{c,true} - x_{c,est})^2 + (y_{c,true} - x_{y,est})^2}, \tag{5.19}$$

where $(x_{c,true}, y_{c,true})$ is the true coordinate and $(x_{c,est}, y_{c,est})$ is the estimated coordinate. The results are plotted as a function of $\Delta P$ and $D_{max}$ where $\Delta P$ represents the separation (in parallel to the projected plane) between the reference points and the $D_{max}$ represents the largest distance (perpendicular to the projected plane) of the reference points. The angle determination is very critical since the displacement is computed after the orientation is determined. Thus, if the orientation computation is not accurate, the localization may not successfully estimate the coordinates.




(a) Orientation Error, $\Delta\theta = 2$ (b) Orientation Error, $\Delta\theta = 4$

Figure 5-9: Illustration of the displacement error as a function of the orientation error $\Delta\theta$.

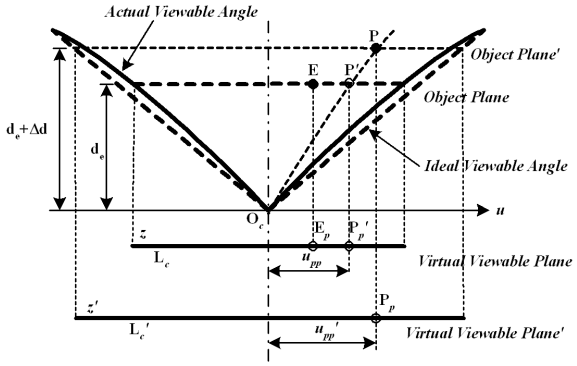



Figure 5-10: The illustration of the actual zooming model caused by lens distortion.

### 5.4.3 Lens Distortion

Fig. 5-10 illustrates the actual (non-ideal) zooming model caused by lens distortion where the dashed line and the solid line indicate ideal viewable angle and actual viewable angle, respectively with zoom factors $z_1 = \frac{D_1}{L_{c,1}}$ and $z_2 = \frac{D_2}{L_{c,2}}$. Hence, $z_1 = z_2$ when the lens view is ideal and linear. But most of practical lens views have non-linear zoom factors.

Fig. 5-11 illustrates zooming non-linearity which is a function of the distance from the lens. The figure shows zooming factor obtained with a typical commercial lens

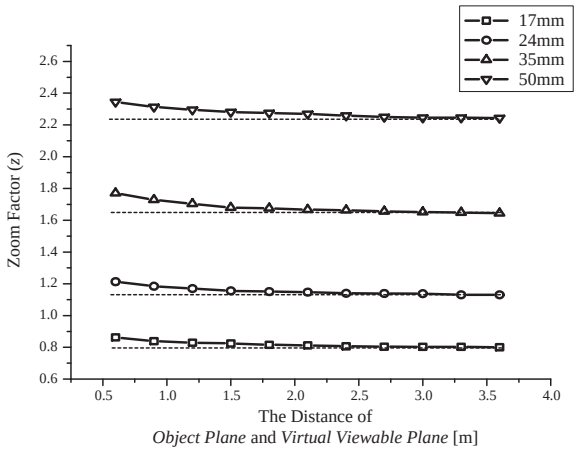



Figure 5-11: Illustration of the zooming distortion as a function of distance from the camera.

(i.e., *Tamron SP AF 17-50mm Zoom Lens* [78]) where the dashed line is the ideal zoom factor (i.e., when infinite distance model is assumed) and the solid line is the actual measured (non-ideal) zoom factor. Note that the zoom factor depends on the distance from the imaging device, as well as the distance from the axis of the lens. The calibration table for the zoom factor is constructed according to the the distance from the imaging device and the distance from the axis of the lens at the interval of $0.3m$.

The non-linear distortion of non-ideal lens affects the scale in the parallel projection model. Since the distances between the mobile sensor and the references are not known, we compute the coordinate of the mobile sensor using the value of $z$ corresponding to the value when the distance is large (i.e., the value of $z$ converges to a specific value). Once initial value of the coordinate is obtained, we use specific values of $z$ (i.e. the one for the first reference and the other for the second reference) to compensate for the non-linearity to obtain the final coordinate of the mobile sensor.

Fig. 5-12 illustrates the nonlinear effect on the coordinate determination. In the

figure, the orientation is chosen to be 0 (i.e., view axis is perpendicular to the x axis). The camera is located at (4.8m, 3.0m) with respect to the global coordinate (0, 0) and the first reference $P_1$ is located at (4.5m, 4.5m) and the second reference $P_2$ is located at (5.1m, 5.1m). These coordinates are used to illustrate the severe non-linearity near the camera as shown in Fig. 5-10. The lens is set at 17mm zoom range. The value of $L_s$ is 18.4cm (i.e., the image size in $x$ direction). The projected position of the first reference $i_1$ is at 3.85cm and the projected position of the second reference $i_2$ is at 2.45cm from the center of the image. These positions are from the center of the reference objects to the center of the image. The reference objects both have finite widths of 0.16cm and 0.12cm corresponding to $\Delta i_1 = 0.0087$ and $\Delta i_2 = 0.0065$, respectively. In this chapter, $\Delta i$ is defined as the uncertainty range or the measurement error with the respect to the overall image size (i.e., 18.4cm in the example). Since the centroid of the reference points are determined from the image, potential measurement errors will be in within $\Delta i$. The actual zoom factor corresponding to the first reference $z_1$ is 0.8238 and the zoom factor corresponding to the second reference $z_2$ is 0.8119. In the initial estimation, the zoom factor corresponding to the infinite distance of 0.8 is used. The estimated coordinate without compensation is (4.8017m, 3.03815m) which is 3.87cm off from the true position of mobile sensor.

### 5.4.4 Effects of Reference Measurement Uncertainty

As briefly discussed in the previous section, the measurement error cannot be avoided. The measurement error directly affects the accuracy of the localization including the

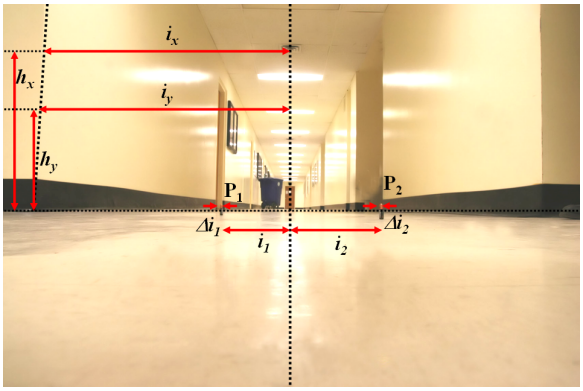



Figure 5-12: The image used to illustrate nonlinear effect of the lens on estimation error.

orientation.

Since the reference object is usually projected as an area on the image, in order to apply the parallel projection model, one point should be determined from the area. In the parallel projection model, we only take the horizontal component of the determined point. If we designate the coordinate of the reference object as its centroid, we can choose its position on the camera plane, $i$, as the center of the projected area. However, if the shape of reference object is not symmetrically round, there is always certain amount of error in determining $i$. This type of error is usually influenced by the image processing in finding the boundary of the projected references. Quantization error due to limited resolution of the visual sensor may affect the accuracy but it is not the biggest source of error.

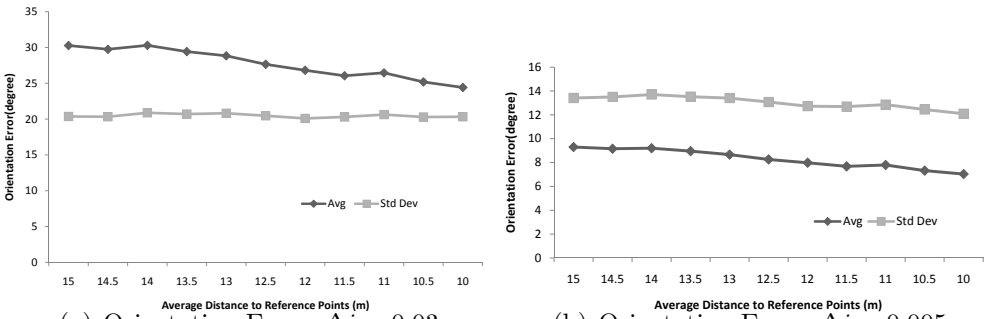
Another source of measurement error is when the reference object is not projected in the center of the horizontal line (i.e., illustrated as dotted line in Fig. 5-12). This is because the lens distorts the projected objects. In the figure, the edge of a wall has one coordinate value. However, multiple reference values can be obtained for the edge of a wall. For example, both $i_x$ measured at $h_x$ and $i_y$ measured at $h_y$ should

represent the same coordinate, but the projected values are different. The difference between $i_x$ and $i_y$ contributes as $\Delta i$ in the localization. However, [55] shows that the projection error is compensated by using the parallel projection model if the height of a landmark is known. Since the height of a landmark can be known, the localization performance is not deteriorated in this case.

The orientation errors due to the incorrect determination of $i_1$ and $i_2$ are illustrated in Fig. 5-13. Fig. 5-13(a) and 5-13(b) show the results for two different values of the measurement errors. The average of orientation error is measured using all the possible combinations of reference points located on 5m by 5m grid. Each reference point is located at 50cm interval in the grid with a small amount of additional random variation. Due to the variation of $i_1$ and $i_2$, the estimated orientation can be different from the true orientation of mobile sensor. The figures show that overall range of error and standard deviation are larger when $\Delta i = 0.03$ than when $\Delta i = 0.05$. Also, when the mobile sensor is closer to the reference points, the orientation error is very critical since the actual coordinates are obtained by computing the orientation first.

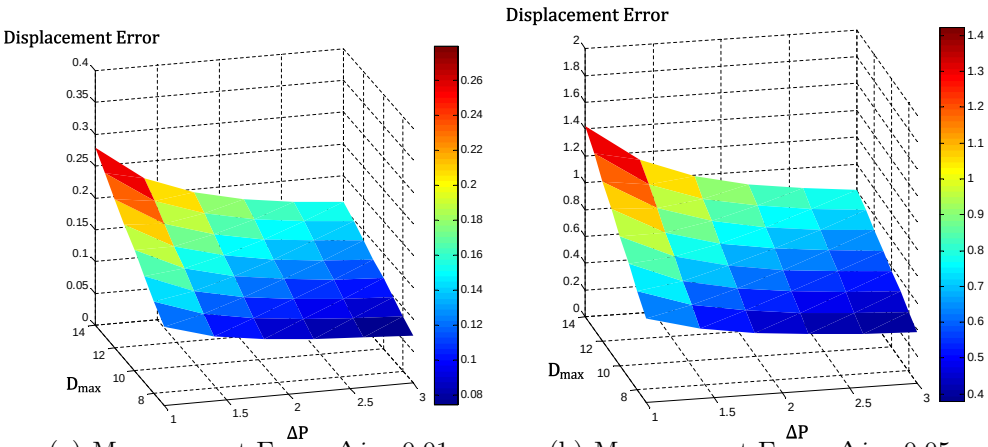
The similar results are obtained for the displacement errors in Fig. 5-14. Similarly, Fig. 5-14(a) and 5-14(b) show the results for two different values of the measurement errors. The displacement error is plotted as a function of two variables, $D_{max}$, the largest distance from the mobile sensor to the reference points, and $\Delta P$, the distance between two most separated reference points measured from the *view axis* of the mobile sensor. As the figure shows, the overall error range increases as $\Delta i$ increases.

Both results show that the algorithm is more prone to error when distance from mobile sensor to reference points is larger, and when the references are closer to one




(a) Orientation Error, $\Delta i = 0.03$ (b) Orientation Error, $\Delta i = 0.005$

Figure 5-13: Illustration of the orientation error, $\Delta\theta$, as a function of the projected measurement error $\Delta i$.



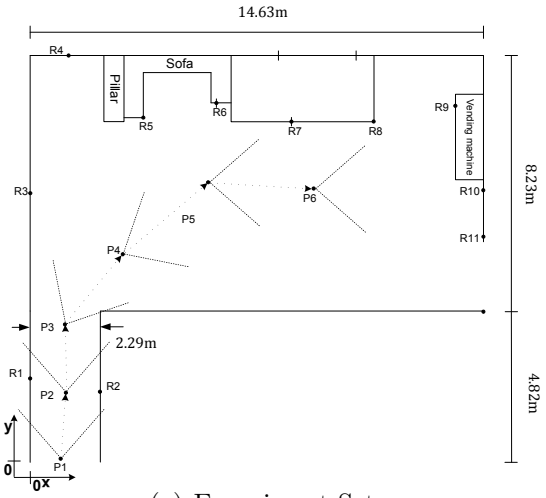

(a) Measurement Error, $\Delta i = 0.01$ (b) Measurement Error, $\Delta i = 0.05$

Figure 5-14: Illustration of displacement error as a function of projected measurement error $\Delta i$.

another. From Fig. 5-13 and Fig. 5-14, we know that estimation error is smaller when the distance between the reference objects along the camera plane is larger. Since our iteration algorithm uses two pairs of reference objects out of three pairs that can be made from three reference objects, given three reference points, $R_1$, $R_2$ and $R_3$, we can choose two pairs that give maximum distance on the camera plane to minimize error. This selection criterion can be applied also when there are more than three reference objects viewable and three of them need to be selected for self-localization.

## 5.5 Analysis and Simulation

### 5.5.1 Experiment Setup




(a) Experiment Setup

| | X(m) | Y(m) | Orientation(degree) |
|---|---|---|---|
| P1 | 0.98 | 0.12 | 0 |
| P2 | 1.15 | 2.25 | 0 |
| P3 | 1.12 | 4.45 | 30 |
| P4 | 2.98 | 6.71 | 60 |
| P5 | 5.73 | 9.02 | 90 |
| P6 | 9.13 | 8.82 | 90 |

| | X(m) | Y(m) |
|---|---|---|
| R1 | 0.00 | 2.71 |
| R2 | 2.25 | 2.28 |
| R3 | 0.00 | 8.67 |
| R4 | 1.24 | 13.11 |
| R5 | 3.65 | 11.11 |
| R6 | 6.00 | 11.58 |
| R7 | 8.42 | 10.98 |
| R8 | 11.07 | 10.98 |
| R9 | 13.70 | 11.49 |
| R10 | 14.60 | 8.77 |
| R11 | 14.60 | 7.27 |

(b) Coordinates of reference points and mobile sensor locations

Figure 5-15: Experimental setup used in the self localization illustration. 10 reference points are used by the mobile sensor located at 6 distinct coordinates. The actual coordinates are shown in the table.

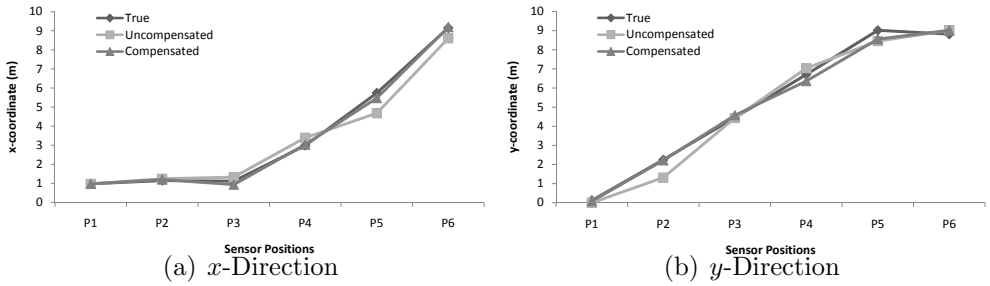
Fig. 5-15 shows the experimental setup for the verification of the proposed

method. A mobile sensor is placed at several positions indicated by $P1, P2, \cdots, P6$ with $P1$ as the starting position. Several reference points, $R1, R2, \cdots, R11$, are designated and measured beforehand. At each position, the mobile sensor estimates the position and orientation before moving on to the next position. Navigation application utilizing our method can employ two strategies as to when to run the algorithm. In the first case, the mobile sensor can move until three reference points are in its sight before running the algorithm. In the other case, the mobile sensor can move to a designated position directed by a navigation algorithm, and search for three reference points by rotating before running the localization algorithm. In the table on right side of the figure, the coordinates of the position of the mobile sensor and those of reference objects are listed. Three reference points are extracted using edge detection and color matching. We evaluate two cases. The first case assumes the orientation is known and the displacement errors are obtained. In the second case, the orientation is computed first, and the displacement errors are obtained from the orientation.

### 5.5.2 Localization Performance with Known Orientation

We first assume that mobile visual sensor knows its orientation. In the experiment, the orientation is fixed according to 5-15(b). Hence the localization performance depends solely on the choice of the reference points and the measurement errors.

Fig. 5-16(a) and Fig. 5-16(b) show the true position of the mobile sensor, in $x$-direction and $y$-direction separately, and their estimated trajectory obtained from using the algorithm. The deviation from the true position is shown as the distance

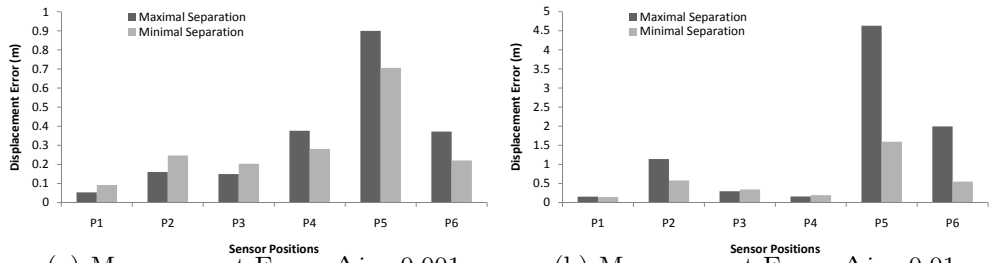



(a) $x$-Direction (b) $y$-Direction

Figure 5-16: Illustration of displacement error at each mobile sensor location with known orientation. Both compensated and uncompensated coordinates are compared to that of true coordinates.

error from the true position to the estimated position of mobile sensor. Two estimated positions are plotted to show the effect of a zoom factor compensation. For the uncompensated estimation, the average value of the zoom factors is used. While displacement error in $x$-direction as shown in Fig. 5-16(a) is negligible, the displacement error in $y$-direction as shown in Fig. 5-16(b) illustrates that the uncompensated estimation deviates from the true position as much as 0.5m. It indicates the zooming factor is very sensitive to the distance from the visual sensor to the reference points. It is because zoom factor has non-linear property only along the $y$-direction or the distance from mobile sensor to the reference objects. However, when the zoom factors are compensated within the algorithm, the distance error in $y$-direction disappears.

The effect of reference points pairing is also investigated with the inclusion of the measurement errors, $\Delta i$. In order to illustrate the effect, we compare the results from the maximal separation pairing to that of the minimal separation pairing. The value of $\Delta i$ is added to or subtracted from the projected reference points $i$'s to make the separation between the two reference points smaller, which maximizes the localization error.




(a) Measurement Error, $\Delta i = 0.001$ (b) Measurement Error, $\Delta i = 0.01$

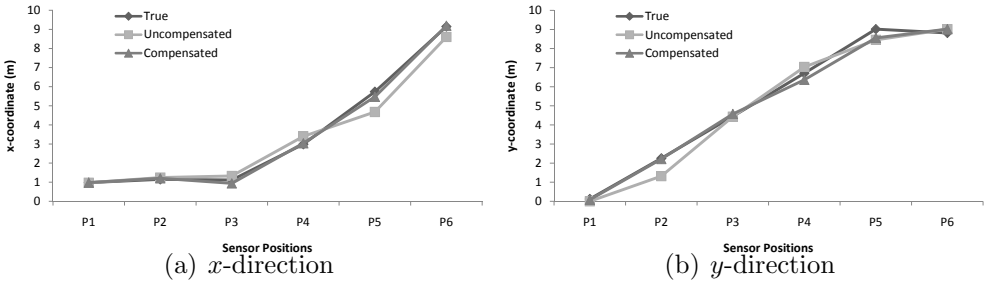
Figure 5-17: Illustration of displacement error for different measurement errors. Each figure illustrates the displacement error corresponding to the maximal and minimal separation methods.

The values of $\Delta i$ are chosen to be 0.001, 0.01. These values are the potential measurement errors caused by having to choose a point from the range of projected objects.

Fig. 5-17 illustrates the simulation results for three different measurement errors. As illustrated in the figures, the average displacement error with the maximal separation pairing is smaller than those with the minimal separation pairing. Note that the displacement error gets larger when reference points are close to one another so it is hard to differentiate the reference points. This confirms the findings shown in Fig. 5-14, where the localization error gets larger when the distance between the reference points, $\Delta P$, become smaller.

Similar to the previous investigation, the effect of orientation errors is shown in Fig. 5-18. The figure illustrates how the displacement error is affected by the orientation error. Note that the simulation result still assumes the orientation is known. The orientation error is simply included in the algorithm. Without the measurement error, the orientation error does not affect the displacement error. In the next section, we investigate the inclusion of the measurement errors on the orientation


Figure 5-18: Illustration of displacement error for different orientation errors.




(a) $x$-direction (b) $y$-direction

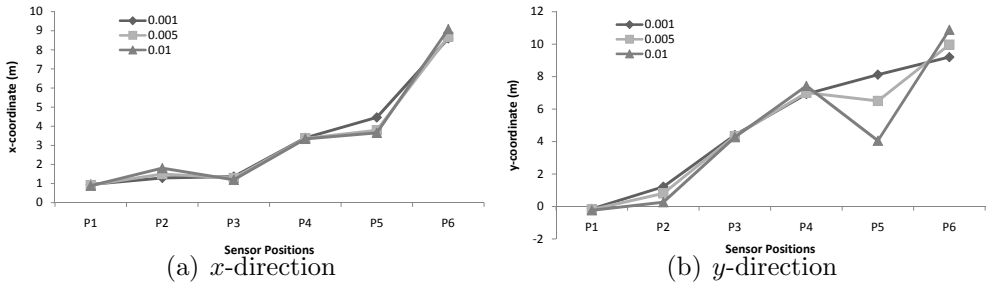
Figure 5-19: Illustration of displacement error at each mobile sensor location with unknown orientation. Both compensated and uncompensated coordinates are compared to that of true coordinates.

determination.

### 5.5.3 Localization Performance with Unknown Orientation

In practice, the mobile sensor estimates the orientation as well as the coordinates. The proposed localization algorithm can determine the orientation of the sensor from the reference points. Since the coordinate of mobile sensor is determined from the estimated orientation, the minimization of the error in estimating the orientation is very critical.

Fig. 5-19(a) and Fig. 5-19(b) illustrate the displacement error in $x$-direction and $y$-direction, respectively. As before, the displacement error in $x$-direction is negligible even without the compensation. Similar result is also shown in $y$-direction.



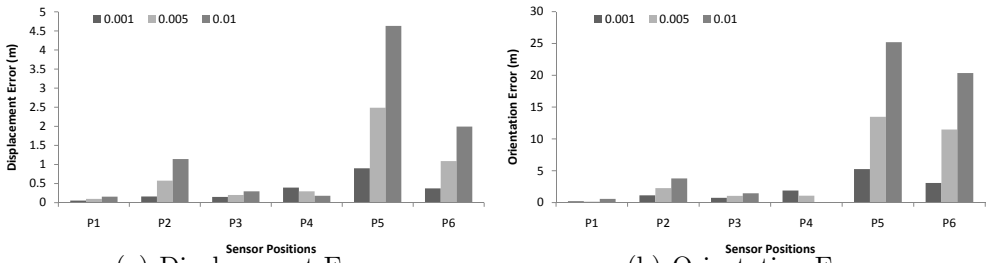

(a) $x$-direction (b) $y$-direction

Figure 5-20: Illustration of displacement error for different measurement errors for unknown orientation. The displacement error is obtained after the orientation estimation with the measurement errors.

The difference between these two figures from the previous figures obtained for the known orientation is that the displacement error is computed after the orientation is determined.

Fig. 5-20 shows the effect of $\Delta i$ on the error of coordinates of the mobile sensor. For this simulation, we used maximal separation criterion for reference objects selection. As shown in the figure, when mobile sensor is farther from the reference objects, the coordinate error is more sensitive to $\Delta i$. In our experiment setup, the mobile sensor is the closest to its selected reference points at the position $P4$. In the figure, at $P4$, the coordinate error is the least sensitive to $\Delta i$. When the mobile sensor at $P5$, where its distance to the reference objects is the farthest, the coordinate error is very sensitive to $\Delta i$. Especially, the $y$-direction error at $P5$, in Fig. 5-20(b) shows the large sensitivity to $\Delta i$ of the coordinate error in $y$-direction. It is because the captured image does not contain any depth information, the variation to $i$ can be mapped to large range of location of mobile sensor in $y$-direction. 0.01 as $\Delta i$ value, is unrealistic considering the real dimension of the image.

Fig. 5-21 shows the orientation error caused by $\Delta i$. Three $\Delta i$ values are used




(a) Displacement Error (b) Orientation Error

Figure 5-21: Summary of the displacement error and corresponding orientation error as a function of measurement errors.

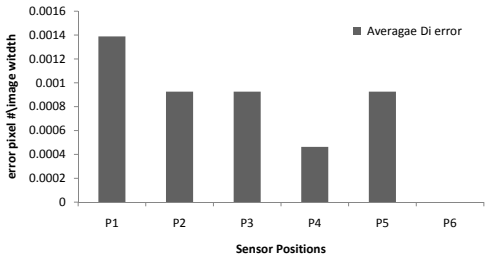



Figure 5-22: Actual average $\Delta$ i error from the experiment

in the illustration. Here, the distance error is calculated from the true position of mobile sensor to the estimated position. As shown in the figure, the uncertainty of the projected reference point, $\Delta i$, affects the coordinate and the orientation error at the same time with the same degree. As it is shown, the error is more sensitive to $\Delta i$, when the mobile sensor is farther from the reference objects.

Finally, Fig. 5-22 shows the actual average $\Delta i$ error. The error is measured by counting number of pixels between the actual reference coordinates of reference points used in calculation and the center point of extracted visual tag. This value is divided by the total number of pixels of the image in horizontal direction. For detection algorithm, we used color based target isolation. As it is shown, the $\Delta i$ value is a fraction of the minimum value we used for simulation.

## 5.6 Summary

In this chapter, we present a novel self localization method using parallel projection model for mobile sensor in navigation applications. The algorithm estimates the coordinate and the orientation of mobile sensor using projected references on single visual image. The camera lens non-linearity is compensated using lens specific calibration table. The method utilizes a simple iterative algorithm, which is very accurate with low computational demand. We identify various sources of measurement error that affect the estimation accuracy. We demonstrate with an example that the algorithm can be utilized in robot navigation as well as positioning application where accurate self localization is necessary.

# Chapter 6

# Conclusions and Future Work

## 6.1 Conclusions

We address various issues for mobile sensors based applications (i.e., a large scale flexible surveillance system and multiple robots application system) and present approaches to resolve them. Object dynamics are utilized to identify tracked positions with the approximated coverage of identification sensors. Although the performance of this scheme depends on the object dynamics, it provides the association recovery against false or failed associations. Moreover, managing redundant information among cameras is an important capability in autonomous surveillance systems. We propose an association method to locally generate homographic lines. This scheme supports flexible camera movements since a reference ground plane is not necessarily shown to cameras. We also consider a situation where objects are densely populated. The simulations show that the situation is effectively managed by using group and incomplete group associations. Finally, the self location method is presented for mo-

bile sensors. The proposed methods can be applied to large scale flexible surveillance systems.

## 6.2 Future Work

### 6.2.1 Map Based Indoor Robot Navigation using Rangefinders in Complex Environment

The performance of the proposed self-localization method highly depends on the accuracy of extracted features. In order to achieve automated mobile sensors, each mobile sensor should be localized by itself and they move according to the received path from the server. Since the global localization system (i.e. the GPS system) is not available in indoor environments, the relative position should be used. Without any map information and absolute reference, it is hard to plan the path for a robot and robot navigation is also not trivial. Thus, we divide the building by the certain size of the sector. This map based approach with the grid facilitates the path planning for robots. The path is easily represented by the lists of the grids.

### 6.2.2 Tracking System Performance Modeling Method for Large Scale Surveillance System Design

Evaluation of tracking performance in surveillance system is difficult because environment and system behavior are complex. Surveillance system model make it easy to evaluate and quantify performance. Tracking performance of surveillance system

depends on application domain, purpose of surveillance, and algorithm in surveillance system. Purpose of surveillance system can be people monitoring, dangerous area monitoring, and human behavior monitoring. The application domain can be a mall, a mine, and a hospital. As properties of domains, surveillance system can have large non-overlapping area or overlapping area, and can be indoor or outdoor. Feature of algorithm can be face, foot, or motion. Due to these various conditions, we install camera and operate application physically to measure tracking performance of surveillance. However, it spends time and costs, and moreover tracking performance of physical operation is not convinced because result is not considered by a lot of situations. When surveillance system covers large area with a lot of cameras, operation of surveillance system is more complex. Consequently, surveillance system model is required for measuring tracking performance and certifying a proper algorithm.

# Bibliography


[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, Vol. 34, No. 3, Aug. 2004.

[2] S. L. Dockstader and A. M. Tekalp, "ultiple camera tracking of interacting and occluded human motion," *Proc. of IEEE*, vol. 89, pp. 1441-1455, Oct. 2001.

[3] K. Charkrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for suveillance and target location in distributed sensor networks," *IEEE Trans. on Computers*, Vol. 51, no. 12, pp. 1448-1453, Dec. 2002.

[4] E. Hörster and R. Lienhart, "Approximating optimal visual sensor placement," *IEEE Int'l Conf. on Multimedia and Expo*, pp. 1257-1260, Jul. 2006.

[5] A. W. Senior, A. Hampapur, and M. Lu, "Acquiring multi-scale images by pan-tilt-zoom control and automatic multi-camera calibration," *IEEE Workshop on Application of Computer Vision*, pp. 433-438, Jan. 2005.

[6] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 758-767, Aug. 2000.

[7] J. Black and T. Ellis, "Multi camera image tracking," *Image and Vision Computing*, Vol. 24, pp. 1256-1267, 2006.

[8] J. Orwell, P. Remagnino and G.A. Jones, "Multiple camera color tracking," *IEEE Int'l Workshop on Visual Surveillance*,, pp. 14-24, Jun. 1999.

[9] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easy living," *Proc. of IEEE Int'l Workshop on Visual Surveillance*, Dublin, Ireland, pp. 3-10, Jul. 2000.

[10] A. Mittal and L.S. Davis, "M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo," *Proc. of European Conf. on Computer Vision*, pp. 18-36, May 2002.

[11] H. Tsutsui, J. Miura and Y. Shirai, "Optical flow-based person tracking by multiple cameras," *Proc. of IEEE Conf. on Multisensor Fusion and Integration in Intelligent Systems*, pp. 91-96, Aug. 2001.

[12] A. Utsumi, H. Mori, J. Ohya and M. Yachida, "Multiple human tracking using multiple cameras," *Proc. of IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pp. 498-503, Apr. 1998.

[13] S. Chang and T.-H. Gong. "Tracking multiple people with a multi-camera system," *Proc. of IEEE Workshop on Multi-Object Tracking*, pp. 19-26, 2001.

[14] S. Calderara, A. Prati, R. Vezzani, and R. Cucchiara, "Consistent labeling for multi-camera object tracking," *Image Analysis and Processing; ICIAP*, pp. 1206-1214, 2005.

[15] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou and S. Maybank, "Principal axis-based correspondence between multiple cameras for people tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4, pp. 663-671, Apr. 2006.

[16] P. H. Kelly, A. Katkere, D. Y. Kuramura, S. Moezzi, S. Chatterjee, and R. Jain, "An architecture for multiple perspective interactive video," *Proc. ACM International Conf. on Multimedia*, pp. 201-212, Nov. 1995.

[17] Q. Cai and J. K. Aggarwal, "Tracking human motion using multiple cameras", *Proc. Int'l Conf. on Pattern Recognition* ,Vienna, Austria, pp. 68-72, 1996.

[18] J. Li, C.S. Chua, and Y.K. Ho., "Color based multiple people tracking," *Proc. of IEEE Int'l Conf. on Control, Automation, Robotics and Vision*, Vol. 1, pp. 309-314, 2002.

[19] Y. Caspi, D. Simakov and M. Irani, "Feature-based sequence-to-sequence matching," *International Journal of Computer Vision*, pp. 53-64, 2006.

[20] J. Kang, I. Cohen, and G. Medioni. "Continuous tracking within and across camera streams," *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. I-267 - I-272, 2003.

[21] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlaping fields of view," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10, Oct. 2003.

[22] S. H. Cho, Y. Kyong and S. Hong, "Homographic line generation and transformation technique for dynamic object association," Proc. of Int'l Workshop on Machine Learning for Signal Processing, Oct. 2008.

[23] Y. Kyong, S. H. Cho, S. Hong, and W. D. Cho, "Local initiation method for multiple object association in surveillance environment with multiple cameras," *IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, Sept. 2008.

[24] S. Velipasalar and W. Wolf, "Multiple object tracking and occlusion handling by information exchange between uncalibrated cameras," *IEEE Int'l Conf. Image Processing*, Sept. 2005.

[25] C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan, and M. Abidi, "Camera handoff with adaptive resource management for multi-camera multi-target surveillance," *Proc. of IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, Sept. 2008.

[26] N. Strobel, S. Spors, and R. Rabenstein, "Joint audio-video object localization and tracking," IEEE Signal Processing Magazine, vol. 18. no. 1, pp. 22-31, Jan. 2001.

[27] H. Zhou, M. Taj, and A. Cavallaro, "Target detection and tracking with heterogeneous sensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 503-513, Aug. 2008.

[28] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, Dec. 2003.

[29] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, Jan. 2002.

[30] R. Brunelli and T. Poggio, "Face recognition: features vs. templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, p. 1042-1052, Oct. 1993.

[31] M. Grudin, "On internal representation in face recognition systems," *Pattern Recognition*, vol 33, pp. 1161-1177, 2000.

[32] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Trans. on Multimedia*, vol. 1, no. 3, pp. 264-277, Sept. 1999.

[33] R.-L. Hsu, M. Abdel-Mottaleb and A K. Jain, "Face detection in color images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, May 2002.

[34] K. Römer, T. Schoch, and F. Mattern, "Smart identification frameworks for ubiquitous computing applications," *Wireless Networks*, vol. 10, no. 6, pp. 689-700, Nov. 2004.

[35] C.M. Roberts, "Radio frequency identification (RFID)," *Computer & Security*, vol. 25, no. 1, pp. 18-26, Feb. 2006.

[36] G. Roussos and V. Kostakos, "RFID in pervasive computing: state-of-the-art and outlook," *Pervasive and Mobile Computing*, vol. 5, no. 1, pp. 110-131, Feb. 2009.

[37] J. Borenstein, L. Feng. "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. on Robotics and Automation*, 12(6): 869–880, 1996.

[38] A. Martinelli, N. Tomatis and R. Siegwart. "Simultaneous localization and odometry self calibration for mobile robot," *Autonomous Robots*, 22(1): 75–85, 2007.

[39] J. Borenstein, H. Everett, L. Feng, et al. "Mobile robot positioning: sensors and techniques," *Journal of Robotic Systems*, 14(4): 231–249, 1997.

[40] J. Borenstein, B. Everett, L. Feng. *Navigating mobile robots: systems and techniques*, MA: AK Peters. Feb, 1996.

[41] A. Gro$\beta$mann and R. Poli. "Robust mobile robot localization from sparse and noisy proximity readings using hough transform and probability grids abstract," *Robotics and Autonomous Systems*, 37(1): 1–18, 2001.

[42] K. Demirli and M. Molhim. "Fuzzy dynamic localization for mobile robots," *Fuzzy Sets and Systems*, 144(2): 251–283, 2004.

[43] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou and S. Maybank, "Principal axis-Based correspondence between multiple cameras for people tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4, pp. 663-671, Apr. 2006.

[44] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, Nov. 2000.

[45] L. Lucchese, "Geometric calibratio of digital cameras through multi-view rectification," *Image and vision computing*, Vol. 23, Issue 5, pp. 517-539, May, 2005.

[46] J. Wang, F. Shi, J. Zhang and Y. Liu, "A new calibration model of camera lens distortion," *Pattern Recognition*, Vol. 41, Issue 2, pp. 607-615, Feb. 2008.

[47] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correlation," *Conference on Computer Vision and Pattern Recognition*, pp. 1106-1112, 1997.

[48] R. Hassanpour, Volkan Atalay, "Camera auto-calibration using a sequence of 2D images with small rotations," *Pattern Recognition Letters* Vol. 25 pp. 989-997, 2004.

[49] S. Z. Li and Z.Q. Zhang, "Floatboost learning and statistical face detection," *IEEE Trans. on Patteran Analysis and Machine Intelligence*, Vol. 26. No. 9, pp. 1112-1123, Sept. 2004.

[50] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137-154, May. 2004.

[51] D. Schulz, D. Fox, and J. Hightower, "People tracking with anonymous and ID-sensors using rao-blackwellised particle filters," *Proc. of Int'l Joint Conf. on Artificial Intelligence*, Aug. 2003.

[52] J. Shin, R. Kumar, D. Mohapatra, U. Ramachandran, and M. Ammar, "ASAP: A camera sensor network for situation awareness," *Proc. of Int'l Conf. on Principles of Distributed Systems*, Dec. 2007.

[53] S. H. Cho, J. Lee, X. Deng, S. Hong, and W.D. Cho, "Passive sensor based dynamic object association method in wireless sensor network," *Proc. of MWSCAS07 and NEWCAS07*, Aug. 2007.

[54] S. H. Cho, J. Lee, and S. Hong, "Passive sensor based dynamic object association with particle filtering," *Proc. of IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, Sept. 2007.

[55] K. S. Park, J. Lee, M. Stanaćević, S. Hong, and W-D. Cho, "Iterative object localization algorithm using visual images with reference coordinate," *EURASIP Journal on Image Processing*, Article ID 256896, 2008.

[56] S. M. Yoon and H. Kim, "Real-time multiple people detection using skin color, motion and appearance information," *Int'l Workshop on Robot and Human Interactive Communication*, pp. 331-334, 2004.

[57] H. Eng, J. Wang, A. Kam, and W. Yau, "A bayesian framework for robust human detection and occlusion handling using a human shape model," *Proc. of 17th Int'l Conf. on Pattern Recognition*, vol. 2, pp. 257-260, Aug. 2004.

[58] H. Elzein, S. Lakshmanan, and P. Watta, "A motion and shape-based pedestrian detection algorithm," *Proc. of IEEE Intelligent Vehicles Symposium*, pp. 500-504, Jun. 2003.

[59] G. Jang, S. Kim and W. Lee, et al. "Color landmark-based self-localization for indoor mobile robots," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, 2002, pp.1037–1042.

[60] J.B. Hayet, F. Lerasle and M. Devy. "A visual landmark framework for mobile robot navigation," *Image and Vision Computing* 25(8): 1341–1351, 2007.

[61] S. Se, D.G. Lowe, and J. Little. "Global localization using distinctive visual features," *Proc. of Int'l Conf. on Intelligent Robots and Systems*, 2002, pp. 226–231.

[62] M. Betke and L. Gurvits. "Mobile robot localization using landmarks," *IEEE Trans. on Robotics and Automation*, 13(2): 251–263, 1997.

[63] A. Briggs, D. Scharstein, D. Braziunas, et al. "Mobile robot navigation using self-similar landmarks," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, 2000, pp.1428–1434.

[64] S. Thrun. "Finding landmarks for mobile robot navigation," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, 1998, pp.958–963.

[65] N. X. Dao, B-J. You, S-R. Oh, et al. "Simple visual self-localization for indoor mobile robots using single video camera," *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2004, pp.3767–3772.

[66] K. T. Sutherland and W. B. Thompson. "Inexact navigation," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, 1993, pp. 1–7.

[67] C. Cohen and F. V. Koss. "A comprehensive study of three object triangulation," *Proc. of SPIE Conf. on Mobile Robots*, 1993, pp. 95–106.

[68] W. Liu and Y. Zhou. "Robot self-localization based on a single image of identified landmarks," *Proc. of IEEE Int'l Symposium on Computational Intelligence in Robotics and Automation*, 2007, pp. 248–253.

[69] J. Yuan. "A general photogrammetric method for determining object position and orientation," *IEEE Trans. on Robotics and Automation*, 5(2): 129–142, 1989.

[70] R. Horaud, B. Conio, O. Leboulleux, et al. "An analytic solution for the perspective 4-point problem," *Computer Vision, Graphics, and Image Processing*, 47(1): 33–44, 1989.

[71] V. Ayala, J. B. Hayet, F. Lerasle, et al. "Visual localization of a mobile robot in indoor environments using planar landmarks," *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2000, pp.275–280.

[72] S. Se, D. owe, J. Little. "Vision-based mobile robot localization and mapping using scale-invariant features," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, 2001, pp.2051–2058.

[73] R. Swaminathan, M. D. Grossberg, S. K. Nayar. "A perspective on distortions," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, pp.594–602.

[74] R. Lenz and R. Tsai. "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(5): 713–720, 1988.

[75] J. Heikkila, O. Silven. "A four-step camera calibration procedure with implicit image correlation," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1997, pp.1106–1112.

[76] F. Lv, T. Zhao, R. Nevatia. "Camera calibration from video of a walking human," *IEEE Trans. on Pattern Analysis and Machine intelligence*, 28(9): 1513–1518, 2006.

[77] V. Lepetit, P. Fua. "Monocular model-based 3D tracking of rigid objects: a survey," *Foundations and Trends in Computer Graphics and Vision*, 1(1), 2006.

[78] http://www.tamron.com/.