Stony Brook University

Robust Object Detection and Localization for Real-Time Autonomous Surveillance Applications

A Dissertation Presented

by

Kyoung-Su Park

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

May 2008

Stony Brook University

The Graduate School

Kyoung-Su Park

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 & Computer Engineering

Ridha Kamoua, Chairperson of Defense
Associate Professor, Department of Electrical & Computer Engineering

Alex Doboli,
Associate Professor, Department of Electrical and Computer Engineering

Hongshik Ahn,
Associate 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

Robust Object Detection and Localization for Real-Time Autonomous Surveillance Applications

by

Kyoung-Su Park

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

2008

In this dissertation, we present robust object detection and localization methods for real-time autonomous surveillance applications. The hyperspectral image processing technology can provide high performance object detection, while requiring special purpose camera and high computational complexity. For real-time processing, the complexity reduction is critical. On the other hand, conventional detection methods can be easily deployed in surveillance system using general purpose camera, although it suffers from limited detection performance. We propose a human body and face joint detection method with multiple cameras where the detection performance is significantly improved. We also propose an object localization method which collaborates with the detection method so that the overall performance can be improved.

First, the spectral characterization for efficient image detection using hyperspectral processing techniques is presented. We investigate the relationship between the number of used bands and the performance of the detection process in order to find the optimal number of bands. The band reduction significantly reduces computation and implementation complexity of the algorithm. Specifically, we define and characterize the contribution coefficient for each band. Based on the coefficients, we heuristically select the required minimum bands for the detection process. We have shown that the small number of bands are efficient for effective detection. The proposed algorithm is suitable for low complexity and real-time applications.

Next, a simplified algorithm to localize object position using multiple images is proposed. We use a parallel projection model which supports both zooming and panning capabilities of the imaging devices. Our proposed algorithm is based on a virtual viewable plane for creating a relationship between an object position and a reference coordinate. The reference point is obtained from a rough estimate which may be obtained from the pre-estimation process. The algorithm minimizes localization error through the iterative process with relatively low computational complexity. In addition, non-linearity distortion of the digital image devices is compensated during the iterative process. The performances of several scenarios are evaluated and analyzed in both indoor and outdoor environments.

Finally, we propose a human body and face joint detection method in a multiple camera environment. The limitations of single camera based human detection are addressed. Through the multiple cameras, the observable range becomes broader with additional perspectives. Multiple cameras with different perspectives pave the

way to collaborate one another, and enable to support additional information among cameras. Each detected human from multiple cameras is transferred to a global localization, which enables us to monitor all-around human movement in a global coordinate. The global information reversely assists the original detection which suffers from the single camera limitations. Furthermore, our proposed application supports the camera panning and zooming through the global information. The performances of multiple human detections are evaluated and analyzed in a variety of multiple camera environments.

All glory to my Lord, Jesus Christ

To my wife, Wonkyoung Cho

Contents

List of Figures

Chapter 1

Introduction

The human monitoring using tracking and localization is particularly useful in security sensitive area such as airports, banks, and building lobbies [1] [2]. In these surveillance systems, the human detection is primarily known to be a significant and difficult research problem [1] [3]. Besides, the object localization is one of the key operations in surveillance systems so that the accuracy of the object localization is very critical and poses a considerable challenge [4] [5] [6].

The hyperspectral image processing technology can provide high performance object detection in surveillance systems. The hyperspectral sensor typically gets one hundred to several hundred of bands for exact spectral classification. The property of the hyperspectral sensor is similar to that of the sensor used in advanced digital cameras. The hyperspectral sensor is capable of covering infrared and/or ultraviolet radiation as well as visible light using the enormous number of bands; a typical digital camera sensor covers only visible light using three bands which are called RGB. The hyperspectral processing technology is gradually incorporated into modern civil and

military remote sensing systems along with other sensors such as imaging radar and laser systems [7].

Hyperspectral processing requires an extremely large amount of input data for the spectral classification. Moreover, the computational requirement for processing input is significant. We characterize key parameters used in hyperspectral processing in order to minimize computational requirements, which are essential for high-speed real-time processing. We are focusing on target detection problems used in surveillance applications.

Most of localization methods use geometric relationship between the object and sensors. Acoustic sensors have been widely used in many localization applications due to their flexibility, low cost and easy deployment. The acoustic sensor provides directional information in angle of the source with respect to the sensor coordinates which are used to create a geometry for localization. However, an acoustic sensor is extremely sensitive to its surrounding environment with noisy data and does not fully satisfy the requirement of consistent data [8]. Thus as a reliable tracking method, visual sensors are often used for tracking and monitoring systems as well [9] [10]. The visual localization has a potential to yield non-invasive, accurate and low-cost solution [11] [12] [13].

We propose a simplified algorithm for localizing multiple objects in a multiple-camera environment. We use the 2-D global coordinate to represent the object location. In our localization algorithm, the distance between an object and a camera is provided by a reference point. Since the reference point is initially a rough estimate, we are motivated to obtain a more accurate reference point. Here, we use an iter-

ative process which substitutes a previously localized position with a new reference point close to a real object location. In addition, the proposed localization method has an advantage of using a zooming factor without concerning about a focal length. Thus, the computational complexity is simplified in determining an object's position which supports both zooming and panning features. In addition, the localization algorithm sufficiently compensates a non-ideal property such as optical characteristics of a camera lens.

A typical human detection method is to extract motion information in order to denote a human body. While the human body is sufficiently obtained from each separate moving object, the detected body frequently suffers from an overlapping problem, especially in a crowded environment [2] [14]. In order to alleviate the overlapping problem, smaller region detections such as a face, an upper body and a leg have been considered [14]. Among the smaller sub-bodies detection, a face detection has been most actively studied because the face characteristics do not easily change compared to an upper body and a leg with a variety of clothes and shoes [2]. However, the face detection is very sensitive according to a resolution and a human-camera orientation. A low resolution suffers from a face recognition and sometimes leads to detection failure [14]. A variety of human-camera orientations require the consideration of all perspective faces such as front-face, side-face and back of the head [2].

The body and face joint detection has several benefits against only body or face detection [15] [16]. The body and face joint detection can keep tracking in the situations even when one of the parts is missing. We selectively use one of body and face information, and possibly estimate or recover a corresponding missing part. Nev-

ertheless, even the joint detection method has several limitations such as occlusion, overlapping as well as face recognition inability due to low resolution. It is rather to say a limitation of single camera. Above all, single camera has a narrow view, which leads an observable range limited. In addition, it is not sufficient to cover a wide area from one perspective. In order to extend the observable range as well as view from additional perspectives, multiple cameras based multiple people detection and tracking have been studied [2] [17] [18]. Through the additional view point, each camera supports another camera to sufficiently solve the single camera limitation, and possibly enable to make a more reliable detection.

We propose a robust human body and face joint detection method with multiple cameras, which enables to collaborate one another in order to alleviate the single camera limitation such as occlusion, body overlapping without face detection, and face recognition inability due to low resolution. The original detection called by primary detection searches body and face information, and draw a rectangular for each in a view of camera. Through the primary detection in a multiple camera environment, it is possible to localize each person in a global coordinate [19]. The global localization and tracking enable us to monitor all-around human movement without disturbances such as overlapping, occlusion and limited viewable range. Thus, the global information reversely assists the primary detection suffering from the above single camera limitations. The assistant detection through global localization and tracking is called as a secondary detection. The secondary detection is performed when the primary detection from one of cameras is failed. Furthermore, we may easily support the camera panning and zooming factors through the global information; thus, the detection

algorithm supports the dynamic camera change.

The rest of this dissertation is organized as follows. Chapter 2 describes spectral content characterization for high performance image detection and then propose an efficient detection algorithm which has minimized computational complexity. Chapter 3 proposes an object localization algorithm in multiple camera environment. Chapter 4 presents an body and face joint detection approach which provides robust human detection with multiple cameras. In Chapter 5, we finally conclude the research works in this dissertation, and mention the future works.

Chapter 2

Spectral Content Characterization for Efficient Image Detection Algorithm Design

2.1 Introduction

The objective of this chapter is to characterize key parameters used in hyperspectral processing in order to minimize computational requirements, which are essential for high-speed real-time processing [20]. We are focusing on target detection problems used in surveillance applications.

There are many approaches for analyzing hyperspectral data. Hardware clusters may be a feasible solution because these are used to achieve high performance, high availability, or horizontal scaling. Cluster technology can also be used for highly scalable storage or data management. These computing resources could be utilized

to efficiently process the remotely sensed data before transmission to the ground [21]. Digital signal processors are also suitable for hyperspectral computations because it can be optimized for performing multiply-and-accumulate operations. It is usually implemented in digital signal processor (DSP) clusters for parallel processing [7] [21]. Even though these processing systems have been applied for hyperspectral processing, high speed image processing and efficient communication within processors are still hot issues. In addition, new processing algorithms and the highly effective memory management are essential for the new hyperspectral sensor which contains higher resolution and much more bands. For a real-time processing hyperspectral system, these are some of the key issues [22].

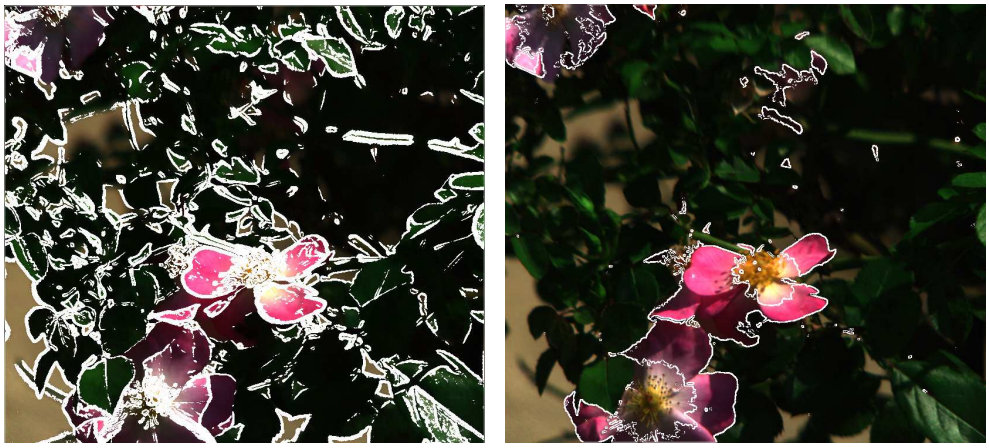
The rest of this chapter is organized as follows. Section 2.2 describes the background of hyperspectral signal processing. The image data structures as well as processing data flow are described. We also characterize various key parameters involved in the detection process. Section 2.3 discusses detection characteristics as a function of the bands and libraries. In Section 2.4, we present a heuristic band selection strategy. The algorithm design and the evaluation are discussed in Section 2.5.

2.2 Background and Problem description

2.2.1 Hyperspectral Image Processing for Detection Problems

Consider the problem of detecting flowers in a garden where a mixture of flowers and various plants are present [23]. Figure 2-1 illustrates the results where detection based on hyperspectral image processing is compared to that of conventional image processing. As shown in Figure 2-1(a), the object is detected in conventional image processing with edge detection using RGB information. Since this image contains many fragmented detected edges, isolating the desired target image becomes a challenge [24]. On the other hand, edge detection can be carried out after the hyperspectral image processing. The result is shown in Figure 2-1(b) in which only the images of flowers are detected. Such detection is possible because every material has an essential spectral property [25]. In this chapter, Figure 2-1(b) to be the ground truth image for comparisons.

Hyperspectral processing involves three key stages. The first step is the calibration stage. The image data produced by a sensor is manipulated to minimize sensor non-uniformity. The sensor is also calibrated by using the initially measured samples to consider the environment of measurement [20] [26]. Each image cube contains a number of bands of spectral contents. For example, the image cube representing the garden of flowers as shown in Figure 2-2 consists of 30 bands of spectral information. Each band represents the information corresponding to a specific frequency range.



(a) Conventional (b) Hyperspectral

Figure 2-1: Comparison of detected images based on conventional approach and hyperspectral approach.

Thus, a library (or spectral information) is constituted by a set of values where the number of values corresponds to the number of bands. In other words, every pixel in the cube is represented by a set of values; thus, a target (i.e., object image to be detected) is represented by numerous sets of values in a library. The second step is the detection stage. In the detection stage, target images are detected via isolating the portion of data which is highly correlated with the given target library. The target library contains spectral information about the object intended to be detected. The objective of the detection stage is to find out the image from the input cube that correlates with the spectral information stored in the target library. The third step is the visualization stage which collects detected image pixels and visualizes through color composition [26].

In this chapter, we focus our discussion on the detection stage. Figure 2-3 illustrates the block diagram of hyperspectral processing. The main challenge of general

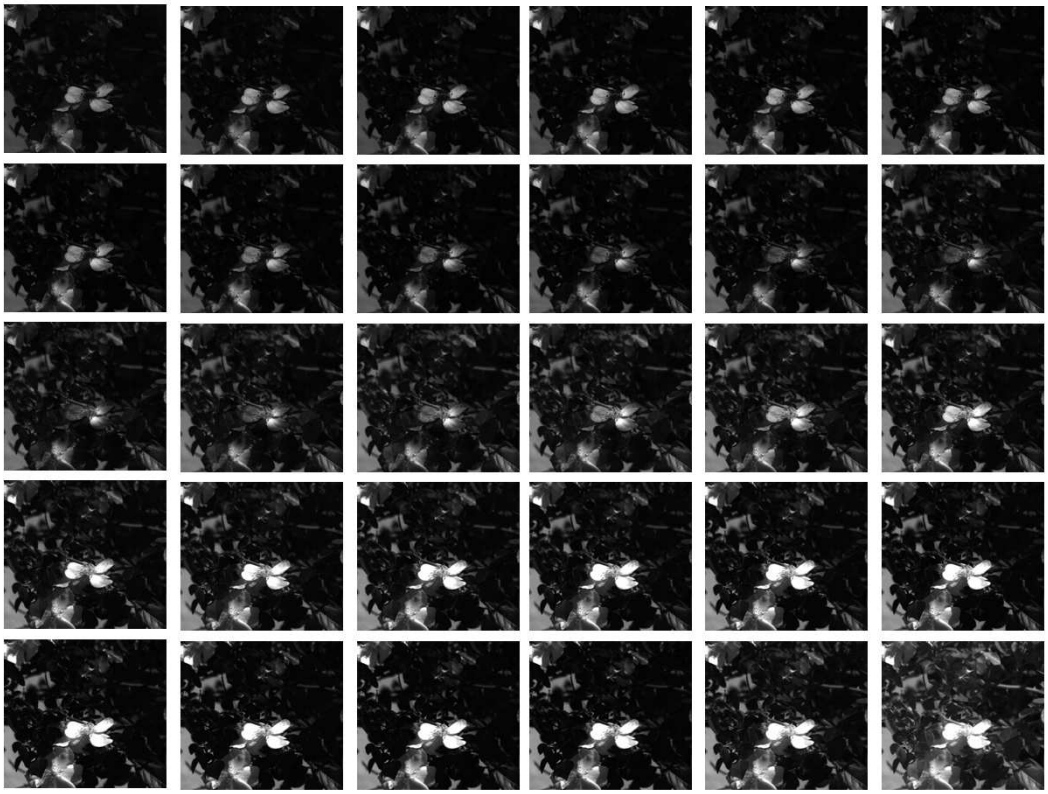


Figure 2-2: Illustration of images corresponding to different bands of the hyperspectral cube.

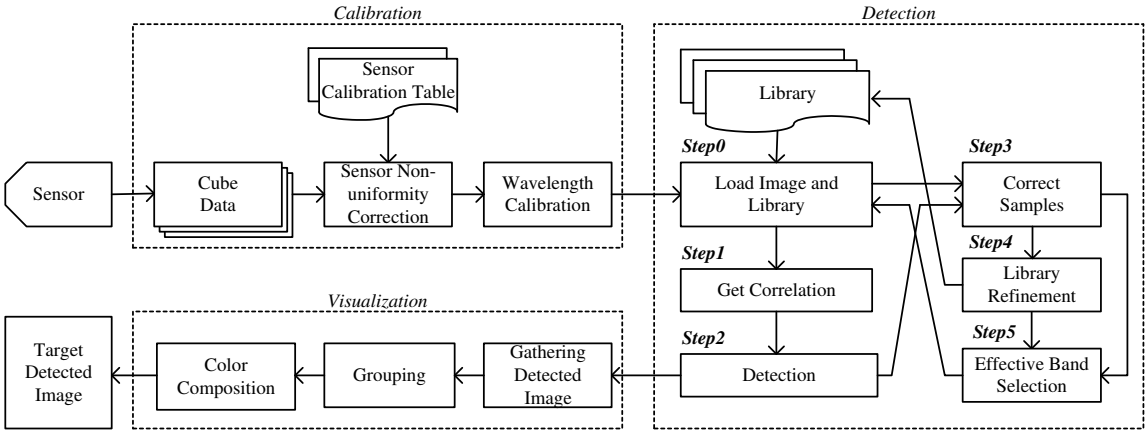


Figure 2-3: Illustration of block diagram about overall hyperspectral processing. A detailed description of steps are explained in Section 2.5.

hyperspectral image processing is the backside of its advantages: high volume and complexity of hyperspectral data. The performance of detection depends on the quality of spectral information stored in the target library. The main operation in the hyperspectral processing for target detection is to compare the input cube with the target library to determine correlation in terms of spectra. The detection is based on perceptual segmentation where spectra contents for each sub-bands are correlated with the spectra contents stored in the library. However, not all bands are necessary since some may contain redundant information where they are compared to the target library. The easiest approach is to reduce the number of bands and the amount of library for processing. However, such reduction may eliminate the merit of hyperspectral processing. Hence, one of our objectives is to determine which bands are effective in detecting the target and selecting them accordingly. The effectiveness is measured in terms of the amount of target being detected with a fewer number of bands. In practice, a perfect target library, which is a set of all spectra comprising the target image, does not exist since objects exhibit different spectral characteristics which are sensitive to environmental factors such as lighting [20] [26] [27]. In the application of target detection , the basic library is a target spectrum which is generated in laboratories or measured in typical environments. Hence, the spectrum of the target image measured by different conditions results in mismatching the target library. Thus, we propose to refine the target library dynamically so that effective detection can be achieved with a small amount of target library information.

2.2.2 Related Work

Traditional store-and-processing system performance is inadequate for real-time hyperspectral image processing without data reduction [22]. In this work, a fine-grain, low-memory and single-instruction-multiple-data (SIMD) processor is presented as an efficient computational solution for hyperspectral processing. However, the SIMD processor does not fully solve the higher resolution and a large number of band problems.

To minimize the volume of hyperspectral image processing, several data compression algorithms are proposed [28]. They achieve impressive compression ratios but could lose valuable information for detection or classification even though the error can be minimized by the clever compression algorithm. However, overall process is affected by the decompression complexity [29]. Statistical approach based on pattern recognition is one of the solutions for high dimensionality of hyperspectral image processing. It uses a small number of reference measurements to distinguish material identification. However, it requires a large number of sample pixels to determine accurate probability density function [29].

Even though hyperspectral image processing uses hundreds of bands to detect or classify targets, there is redundancy which means partial bands efficiently accomplish the edge detection as described in [29] and [30]. In [29], the band selection is based on the band add-on (BAO) procedure that chooses an initial pair of bands and classifies two spectra by correlation, and then adds additional bands that increase the correlation of two spectra. It is a feasible solution to determine effective bands when

an unknown pixel is classified by using many reference classes. A set of best-bases feature extraction algorithms is proposed for classification of hyperspectral data as well [31]. This method is simple, fast, and highly effective so that it can reduce the input space from 183 dimensions to less than four dimensions in many cases. However, this approach is based on classification so that it is suitable when a spectrum of a pixel is classified by many numbers of libraries. In the application domain of target detection, the input image is compared to a few libraries which represent the spectrum contents of the target.

2.2.3 Correlation Coefficient of Image (A)

Correlation coefficient, A, is a measure of similarity between the stored spectra in a target library and the obtained spectra from sensors. The high value of correlation indicates the high degree of similarity between two spectrums [32]. The correlation coefficient is defined as

$$A = 1 - cos^{-1}\left(\frac{\sum_{i=1}^{N_T} t_i r_i}{\sqrt{\sum_{i=1}^{N_T} t_i^2}\sqrt{\sum_{i=1}^{N_T} {r_i}^2}}\right), \tag{2.1}$$

where N_T is the number of bands in input spectrum, t_i is the test spectrum of i^{th} band, and r_i is the reference spectrum of i^{th} band. The value of correlation defines a degree of similarity between input spectrum and target spectrum stored in the target library.

The input spectra of an object is compared to the spectra in the target library. This comparison is based on the correlation coefficient. In this chapter, we define

A_t as the minimum correlation coefficient value, which recognizes the target between unknown spectra. When the correlation value is higher than or equal to A_t, the object is assumed to be matched with the data in the target library. Thus, the value is used as an indicator for the degree of confidence in detection.

If we use lower A_t to detect targets, it increases the possibility of wrong detection which means some backgrounds are detected as a target. However, if the numbers of libraries and bands applied in detection are increased, the performance of target detection is improved. However, even if all possible information is used to detect targets, there is a limit value where target and background cannot be isolated. Thus, the minimum correlation coefficient (A_t) is related to the similarity within the target and background. We define A_b as a maximum correlation value where any correlation value below A_b is considered to be a background, which means the pixel is not a target at least. The detected image with the correlation value below A_b may not be the interest of objects which may capture a large portion of the background.

2.2.4 Percentage of Detected Image (P)

Percentage of detected image (P) shows the effectiveness of selected bands in the detection process. Figure 2-4 illustrates the relationship between the correlation coefficients and percentage of detected image (P) where three types of target libraries are used. When the given correlation coefficient A_t is 1.0, the value of percentage of detected image (P) is very low (i.e., approaches to zero). For all libraries, when the correlation coefficient is increased, the percentage of detected image (P) is decreased.

We define A_t as the correlation value where the change in the percentage of detected image (P) is smaller than some value δ as we increase the value of the correlation coefficient.

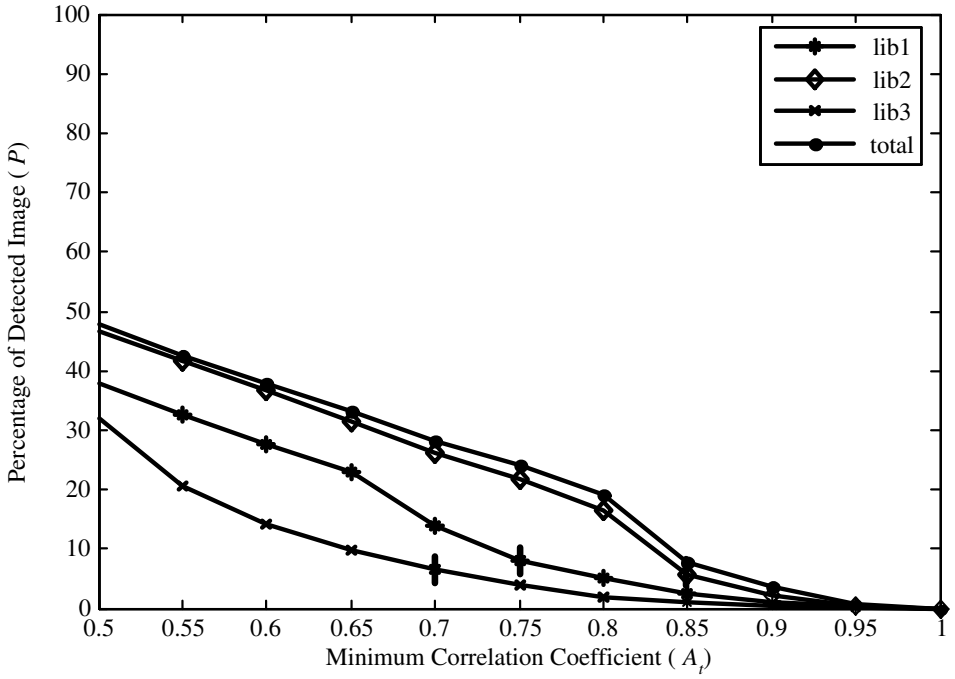


Figure 2-4: Relationship between the correlation value used and the percentage of detected image (P). 31 bands of input image data are used in the simulation.

Figure 2-5 shows the simulation results of the detected image as a function of the minimum correlation values for one target library, *lib1*. The detected images are shown for different minimum correlation values; 0.70, 0.75, and 0.85. In the case where A_t of *lib1* is 0.70, unwanted objects that satisfy the minimum correlation value are detected as a target. However, as A_t is increased to 0.85, the unwanted objects almost disappear in the detection at the cost of losing the target image. At the minimum correlation A_t of 0.85, the process tries to find only the image from the input that is highly correlated with the target library.

(a) $A_t = 0.70$ (b) $A_t = 0.75$ (c) $A_t = 0.85$

Figure 2-5: Result of detected image as a function of correlation values A_t for *lib1*. 31 input bands are used and processed with one library.

The values of percentage of detected image (P) have two interpretations. First, the higher value of percentage of detected image (P) (i.e., more image has been detected) implies that more target image is detected. Second, the higher value of percentage of detected image (P) can imply that some of the detected image is not the target. Hence, detection depends on the number of libraries (spectral information) and their qualities as well as the minimum correlation values used in the process.

Under the assumption which multiple libraries are used in the detection, we define total percentage of detected image (PT) as following:

$$P_T = \sum_l P(l, A_t), \tag{2.2}$$

where l is the index of each library and $P(l, A_t)$ is the percentage of detected image (P) value at the correlation value A_t when library l is used. We will use the total percentage of detected image (P) as an indicator for detection performance.

2.3 Target Detection

2.3.1 Effects of Number of Bands

Since the motivation of our work is to use the smaller number of bands for detecting the target, we investigate the effects of the number of bands on detection performance. Thus, the goal is to minimize the total percentage of detected image (P_T) at the minimum correlation (A_t) given the number of bands (N_E).

Figure 2-6 shows the detected image where a partial number of bands are used to detect flowers. When the number of bands, N_E, is equal to 2, the detected image includes the target image as well as other unwanted background images. It implies that two bands are not effectively isolating the target image. When the number of bands is more than 4, the detected images become isolated and the percentage of detected image (P) is lower than that of the image generated with 2 bands. However, there is only slight improvement (the total percentage of detected image (P) is decreased) from 4 bands to 16 bands.

We define the degree of effectiveness in terms of the total percentage of detected image (P_T). As shown in Figure 2-6(a), the total percentage of detected image (P_T) is higher than that shown in Figure 2-6(b) and Figure 2-6(c) (i.e., more images are shown). However, the total percentage of detected image (P_T) is improved (reduced) very slightly from 4 bands to 16 bands. This shows that the complete use of the bands is not always necessary for detecting the target from the input image.

(a) 2 bands (b) 4 bands (c) 16 band

Figure 2-6: Result of detected image as a function of the number of bands used out of 31 input bands.

2.3.2 Redundancy Between Bands

To use the partial number of bands, the simplest approach is to select bands in random. In this section, we consider two types of band selection in order to characterize the effect of band selection on detection performance. We investigate the redundancy within the bands.

Clustered Bands

Cluster band selection selects N_E consecutive bands. Figure 2-7 shows the relationship between the correlation coefficient and the percentage of detected image (P) when 4 consecutive bands are selected out of 31 possible bands. The selected bands are (27, 28, 29, 30). The figure shows a much higher percentage of detected image (P) for the entire range of correlation values when it is compared to that of Figure 2-4. Thus, the figure indicates that it has detected more image from the background. In this situation, it is likely that the detected image contains a lot of unwanted images.

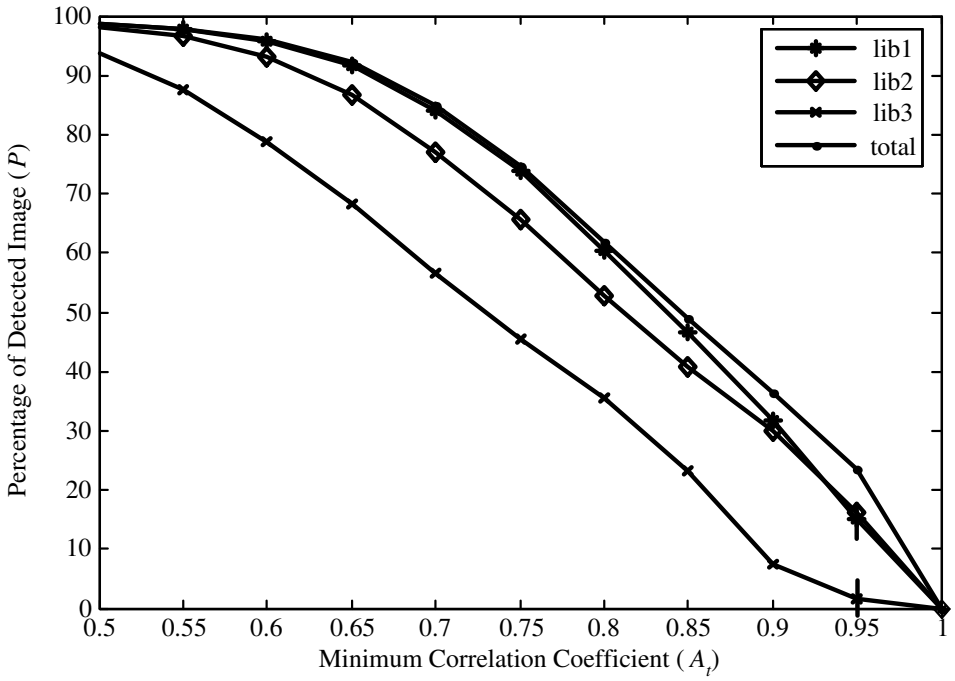


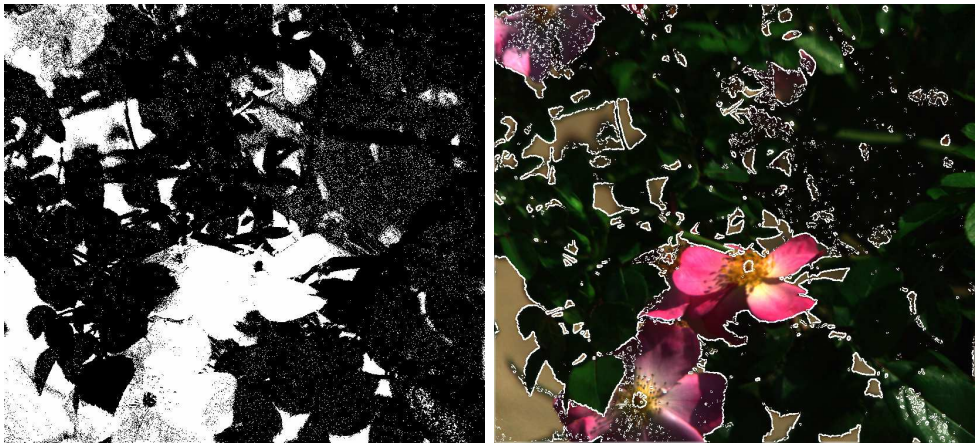
Figure 2-7: Relationship between the correlation values and the percentage of detected image (P) when clustered bands (27, 28, 29, 30) are used in the detection.

The analysis with the percentage of detected image (P) is proven by the detected image illustrated in Figure 2-8. Each of the three libraries were not effective in detecting the flowers. Even with the correlation coefficient of 0.95, the target is not separated from the background. This simulation suggested that those clustered bands contain redundancy and the clustered bands are not effective in detecting the target. Similar results were obtained when the other sets of clusters are used. Thus, the clustering is not an effective way to select the bands for detection.

Maximum Separation Bands

On the other hand, we select the bands that are maximally separated. There are several combinations of sets of bands. Figure 2-9 shows the relationship between correlation and the percentage of detected image (P) where bands are selected by

(a) With *lib1* (b) With *lib2* (c) With *lib3*



(d) Detection with clusters (e) Detected image with full colors

Figure 2-8: Result of detected image when clustered bands are used in the detection. Bands used are (27, 28, 29, 30).

maximal separation as (2, 10, 18, 26).

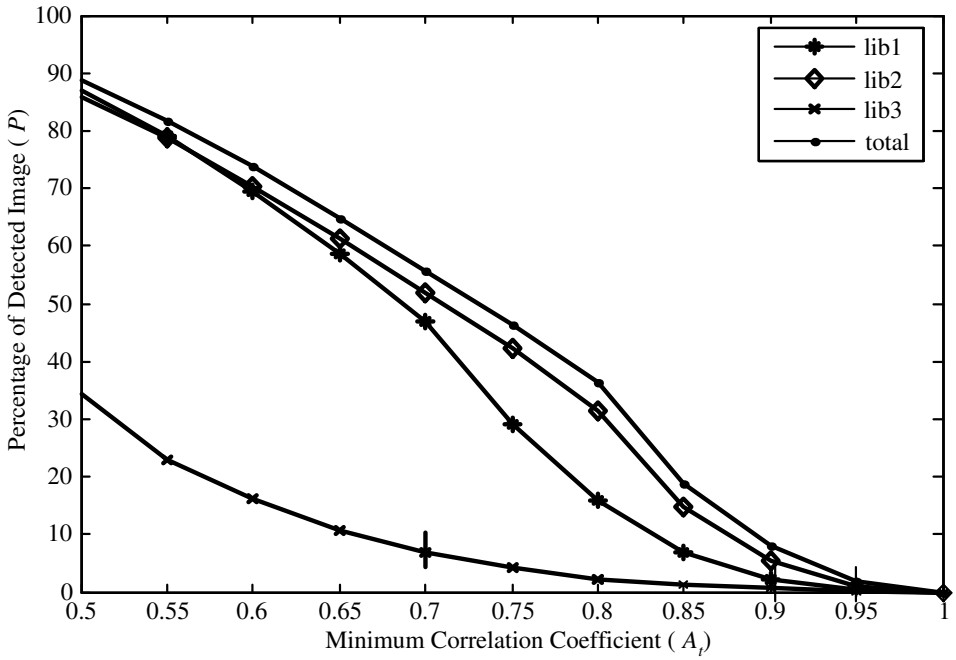
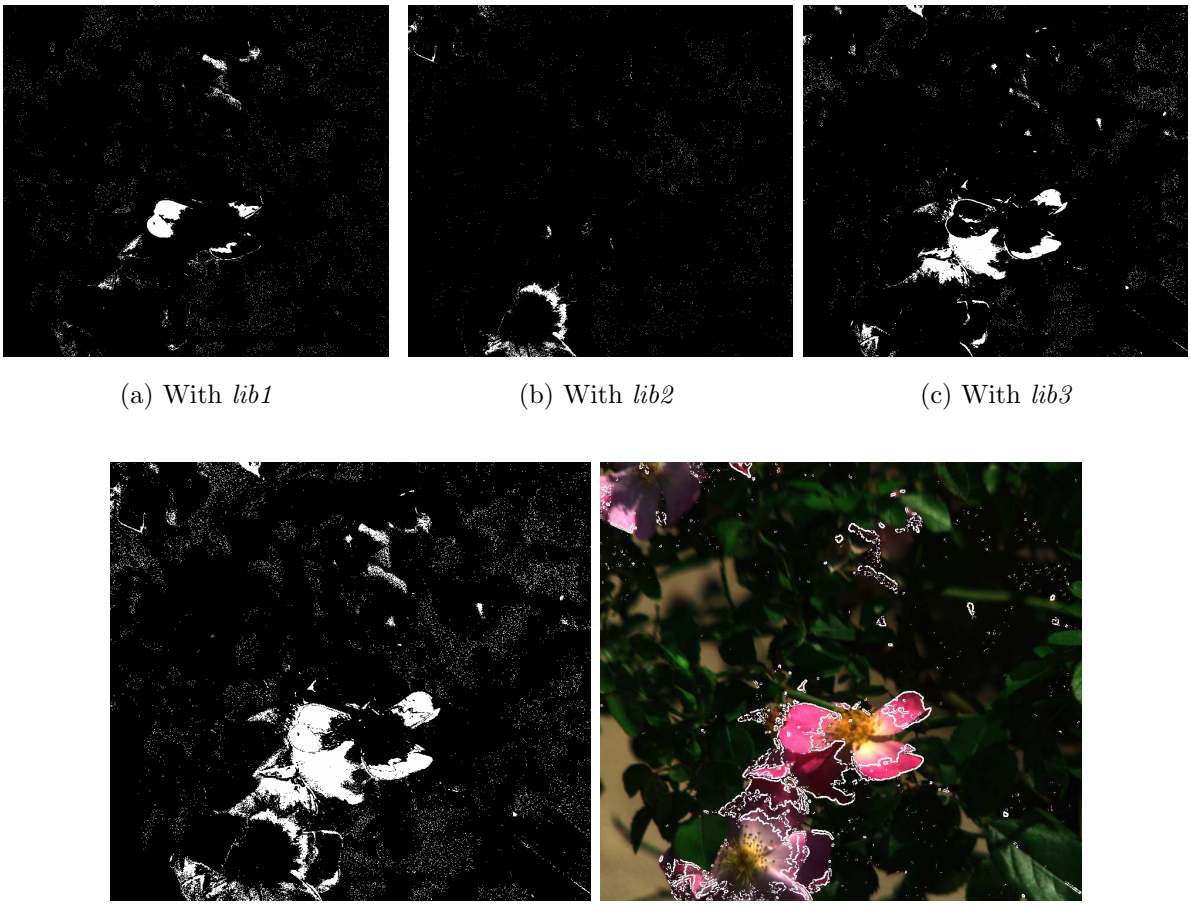


Figure 2-9: Relationship between the correlation values and the percentage of detected image (P) when maximum separation bands are used in the detection. Band used (2, 10, 18, 26)

As shown in Figure 2-9, the percentage of detected image (P) values of each library as well as the total percentage of detected image (P_T) are much lower than that for the entire range of the correlation values. For example, the total percentage of detected image (P_T) of clustering case at $A_t = 80$ is 70 while maximum separation case at $A_t = 80$ is 40. This implies that the maximal separation performs better than the clustering at any minimum correlation value. The detected image by each library shown in Figure 2-10 contains only the flowers. This is much improved detection over the clustering method. Figure 2-10(d) illustrates the detected image when all three libraries are used.

However, in the results generated by the maximum separation, some of the targets



(a) With *lib1*

(b) With *lib2*

(c) With *lib3*

(d) Detection with maximum separation

(e) Detected image with full colors

Figure 2-10: Result of detected image when maximum separation bands are used in the detection. Bands used (2, 10, 18, 26).

were lost. Similar results are obtained with a different set of bands (4, 12, 20, 28). The detected images by three target libraries are illustrated in Figure 2-11. The band set (4, 12, 20, 28) performs better than the band set (2, 10, 18, 26) in detecting and isolating the target images. This implies that while the maximum separation scheme is better than the clustering, more bands may be necessary since the total percentage of detected image (P_T) value obtained is much higher than the case of 31 bands. We will present an effective band selection scheme in Section 2.4.

(a) With *lib1* (b) With *lib2* (c) With *lib3*

Figure 2-11: Result of detected image when maximum separation bands are used in the detection. Bands used (4, 12, 20, 28).

Observation

We can observe from the results that detected images are improved when the percentage of detected image (P) value is low for the given correlation values. This observation coincides when we compare Figure 2-4, Figure 2-7, and Figure 2-9. the percentage of detected image (P) is the lowest when all bands are used for given correlation value. We will consider an approach for selecting bands in the next section.

When the number of bands is increased, the percentage of detected image (P) is reduced and then it is saturated. This means that a target can be detected by using only partial bands because some bands have enough information to detect a target.

2.4 Complexity Reduction Strategy

The main objective in reducing computational complexity is to determine the minimum number of bands used in the detection process as well as selecting a specific set of bands. In this section, we first define the band contribution coefficient and present a band selection strategy based on the coefficient.

2.4.1 Band Contribution in Detection

Library usually has several spectra for a target because the spectrum depends on the measurement part of the target and the condition of light sources. Figure 2-12 is an example of spectra for library and background, which shows three libraries and two background spectra. When the spectral information of the target is highly different from the background, the target detection is easier. In Figure 2-12, the spectrum of *lib1* from the 18th band to the 31st band is saturated. Also, spectrum waveform of *lib2* is similar to *lib3*. However, the magnitude is different within the two libraries. *background1* is extracted from leaves and *background2* is from the back of a scene.

The effectiveness of the k^{th} band of the l^{th} library, $e_{l,k}$, is defined as

$$e_{l,k} = \frac{|\sum_{b=1}^{N_B}(l_{l,k} - b_{b,k})|}{N_B}, \tag{2.3}$$

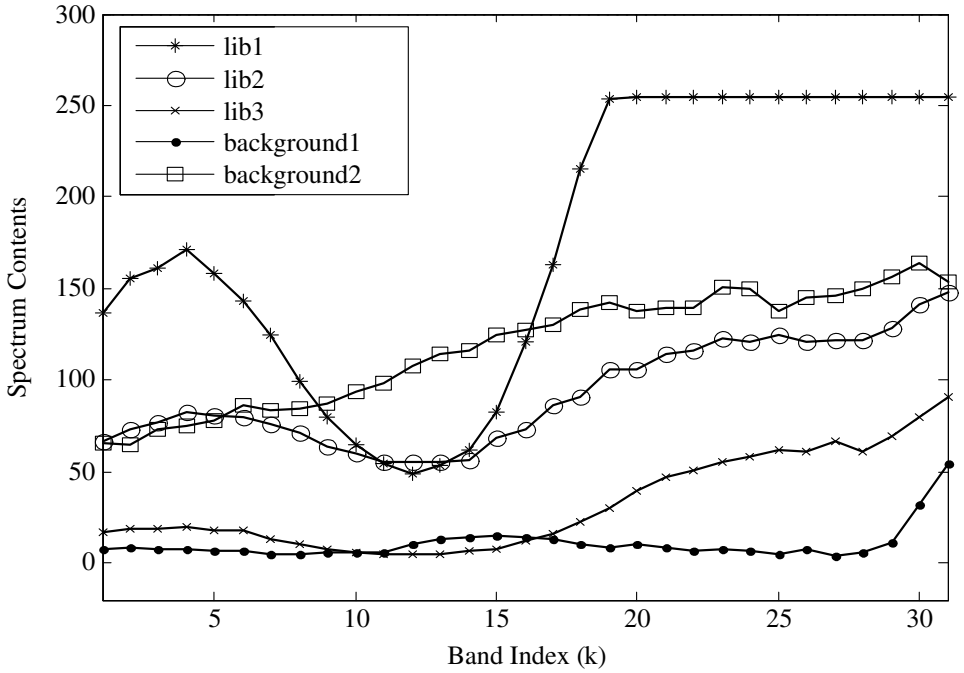


Figure 2-12: Comparison between spectrum of target libraries and the spectrum of the background of input bands.

where N_B is the number of backgrounds, $l_{l,k}$ is the k^{th} spectrum content in the l^{th} library and $b_{b,k}$ is the k^{th} spectrum content in the b^{th} background.

If a spectrum of a target is similar to that of data in the library, target detection is achieved more effectively; we will define the effectiveness as contribution. The contribution coefficient (c) is defined as

$$c_k = \frac{\sum_{l=1}^{N_{LIB}} e_{l,k}}{N_{LIB}}, \tag{2.4}$$

where c_k is the contribution of the k^{th} band and N_{LIB} is the number of libraries.

The relationship between the contribution factor and the number of bands is illustrated in Figure 2-13. Contribution of *lib2* and *lib3* is less than 20 while *lib1* has much higher contribution than other two libraries. Thus, the contribution of *lib1* is

dominant as shown in Figure 2-13.

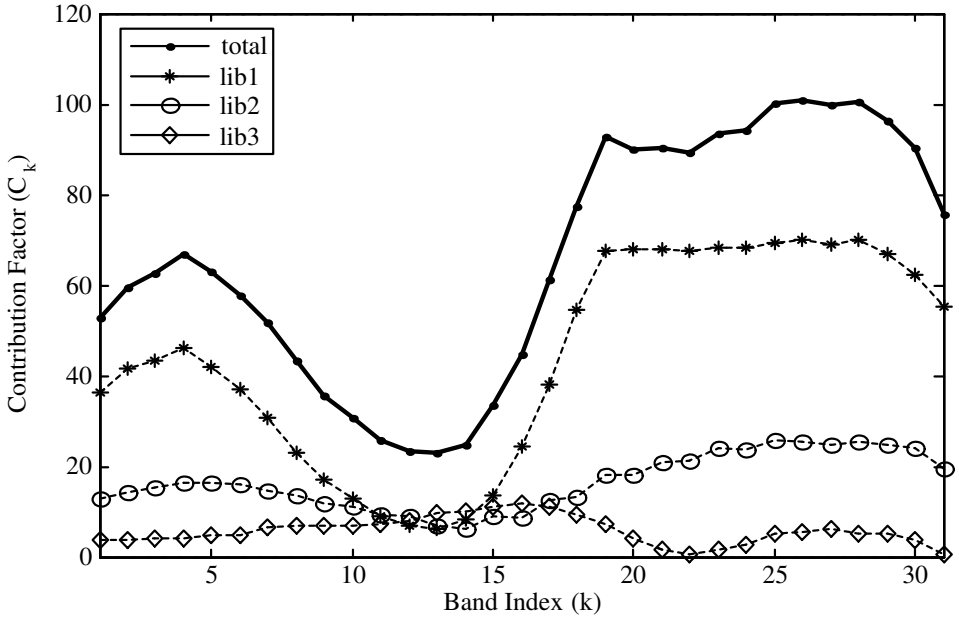


Figure 2-13: Illustration of contribution coefficient of each band.

Even though the contribution coefficient is not an absolute indicator for detection, the coefficient is considered to be one of the factors for isolating the target. To obtain the contribution, we need to choose samples of backgrounds. Samples are randomly selected in a scene, and then each sample is verified to be a background or an applicant of a target by using the maximum correlation coefficient (A_b). If the correlation coefficients between an input spectrum and all of the libraries are lower than A_b, the input spectrum is considered a background. Also, A_b is experimentally decided depending on an application. Although background and library can be highly correlated, the contribution factor is a powerful factor under the condition of which A_b is lower than A_t.

2.4.2 Effective Band Selection

Since the contribution coefficient represents the effectiveness to detect targets, it has a benefit for effective band selection. However, if the high contribution bands are selected, it may lead to select clustered bands (i.e., bands 27, 28, 29, 30).

From the definition of correlation in Equation (2.1), the correlation of library and background is basically the variation of the difference in two spectra. For example, if the spectrum contents in a reference are $(10, 20, 40, 60, 50, 30)$ and the test spectrum has 10 times higher value of contents like $(100, 200, 400, 600, 500, 300)$, the correlation between two spectra is 1 which means two spectra are perfectly correlated since the variations of spectrum contents between adjacent bands are same.

Thus, effective bands represent the variation of differences between the library and the background. Since contribution is related to the difference between the library and the background, isolating the target and background in lower A_t can be one of the solutions in maximally separated bands. To maximally separate the contribution of selected bands, the first band has minimum contribution and the last band has maximum contribution. The contribution of the k^{th} bands is $((maxC) - (minC))/(N_E - 1) \times k + (minC)$ where $(maxC)$ and $(minC)$ are the values of maximum and minimum contribution, respectively.

For example, let us assume a series of contributions is $(90, 180, 360, 540, 450, 270)$. Since the contribution of the 1^{st} band is minimum and the 4^{th} band is maximum, the 1^{st} and the 4^{th} are selected. Then, since the gap of selected bands is $150(= (540 - 90)/3)$, contributions of second and third bands are approximately 240 and

390, respectively. Since the contribution values of the 6^{th} and the 3^{rd} bands are close to 240 and 390, the 6^{th} and the 3^{rd} bands are selected as effective bands. Figure 2-14 shows the result of target detection when the effective bands are selected. The result is similar to the one in the case where full bands are used.

(a) With *lib1*

(b) With *lib2*

(c) With *lib3*

(d) Detection with effective band selection

(e) Detected image with full colors

Figure 2-14: Result of detected image when the effective band selection strategy is used in the detection.

2.4.3 Library Selection

We have observed that some target libraries work better in detecting the target than other target libraries. Theoretically, a larger set of target libraries will enhance the detection but at the cost of computational complexity. We investigate the target library selection in cases where the finite number of target libraries is to be used for reducing the computational complexity. However, the best possible sets of target libraries cannot be generated or obtained before the processing. However, the target library can be improved during the detection process.

In Figure 2-15, the total percentage of detected image (P_T) from *lib1*, *lib2*, and *lib3* is 14% when A_t is equal to 0.8. Even though *lib1* is more effective to detect targets than other libraries, *lib2* or *lib3* can detect the different part of the targets.

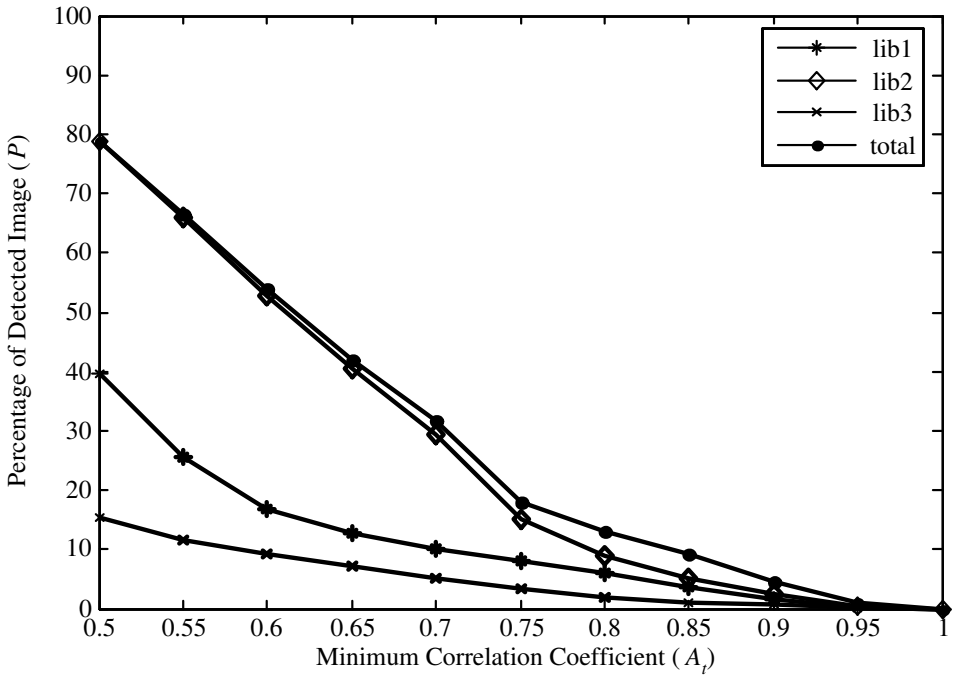


Figure 2-15: Relationship between the correlation values and the percentage of detected image (P) when effective band selection strategy is used.

Note that the lower value of P_T does not imply that the performance is better. It merely suggests that there is a high probability that the detected image is only a target. Figure 2-16 shows the relationship between the percentage of detected image (P) and correlation coefficient when it has two libraries (*lib2* and *lib3*). In addition, when several libraries are used, more effective libraries will produce bigger contributions.

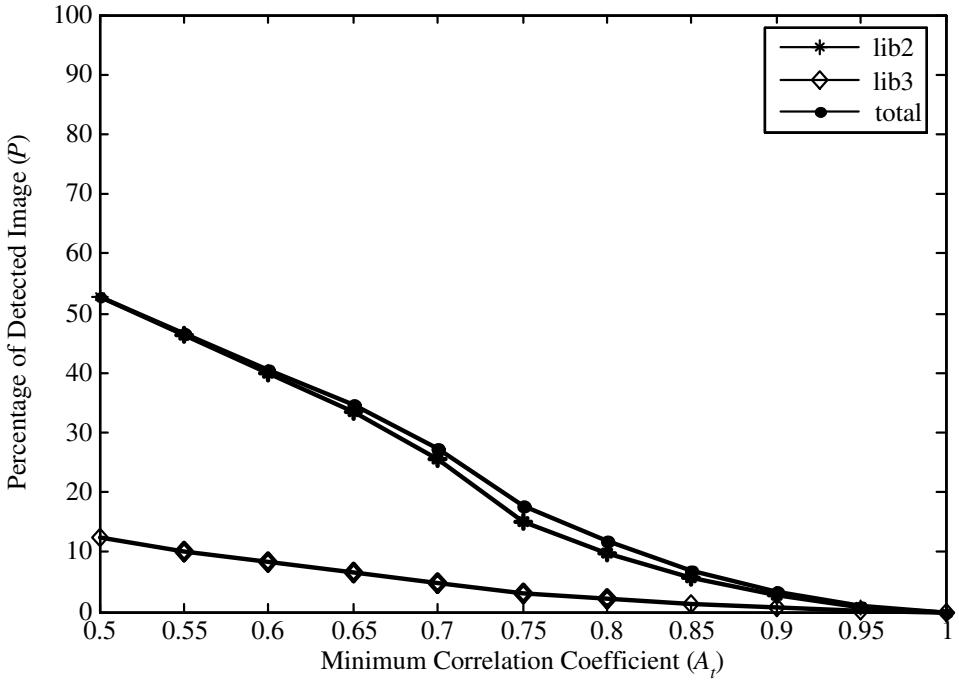


Figure 2-16: Relationship between the correlation values and the percentage of detected image (P) when two libraries are used.

Figure 2-17 illustrates of the library selection where *lib2* and *lib3* are used. Figure 2-17(a) and 2-17(b) have 5.71% and 4.71% of percentage of detected image (P), respectively. Since the total percentage of detected image (P_T) is 10.39% , two detected areas are slightly overlapped.

(a) With *lib1 (5.71%)* (b) With *lib2 (4.71%)* (c) With *lib1* and *lib2 10.39%*)

Figure 2-17: Illustration of the library selection

2.4.4 Library Refinement

One important aspect that we have discussed in this chapter is that the performance depends on the quality of the target library. Library refinement improves the detection process. The overall process starts with a set of basic libraries. Once a target image is detected, the target library from the detected image is refined. The refined library has all spectrums of the detected target. Once the refined library is generated, the library is applied in lieu of the basic library.

Figure 2-18 shows the results of library refinement where the detected image has 0.9 of the correlation coefficient. Figure 2-18(a) uses the basic library and Figure 2-18(b) and Figure 2-18(c) use the refined library. Since A_t is not 1.0 (perfect correlation value), a background image is detected as a target. Hence, the chosen target image with library refinement is a candidate of the new library. The randomly selected target image is compared to the basic library each time. If the correlation between the new library candidate and basic library satisfies the condition ($\geq A_t$), the current

library is replaced by the new library candidate. Otherwise, the basic library is used in the process.

(a) Basic Library (b) Case 1 of Refined Library (c) Case 2 of Refined Library

Figure 2-18: Result of detected images when the libraries are refined from detected samples ($A_t = 0.9$).

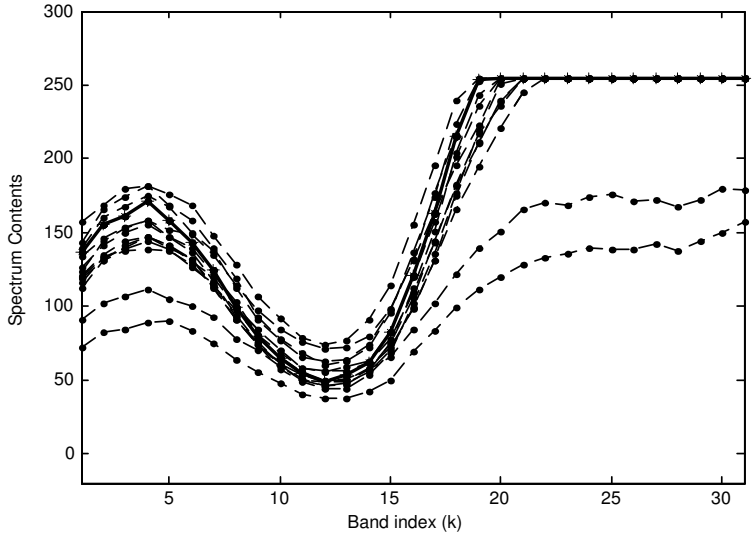
In Figure 2-19, refined libraries are shown by the dashed line where all refined libraries satisfy the condition of correlation ($A_t = 0.9$). The refined library can be adopted in a variety of light source conditions.

2.5 Algorithm Design

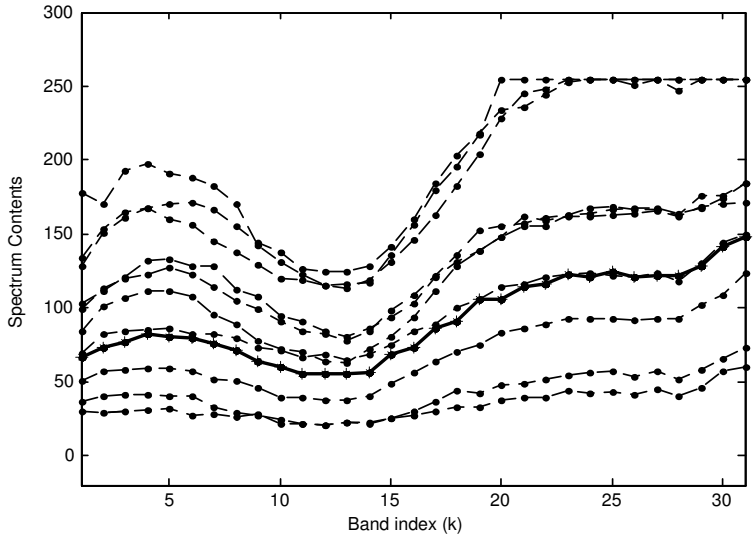
2.5.1 Algorithm Overview

Figure 2-20 illustrates the overall algorithm for detecting and isolating target images in Processing where the algorithm has two processing flows. The right side is for comparing the input cube with the target libraries. The left side has two parts where the target library is refined and the effective band selection is performed.

We assume the basic parameters are loaded in Step 0. The basic parameters are



(a) With *lib1*



(b) With *lib2*

Figure 2-19: Refined libraries of lib1 and lib2 ($A_t = 0.9$)

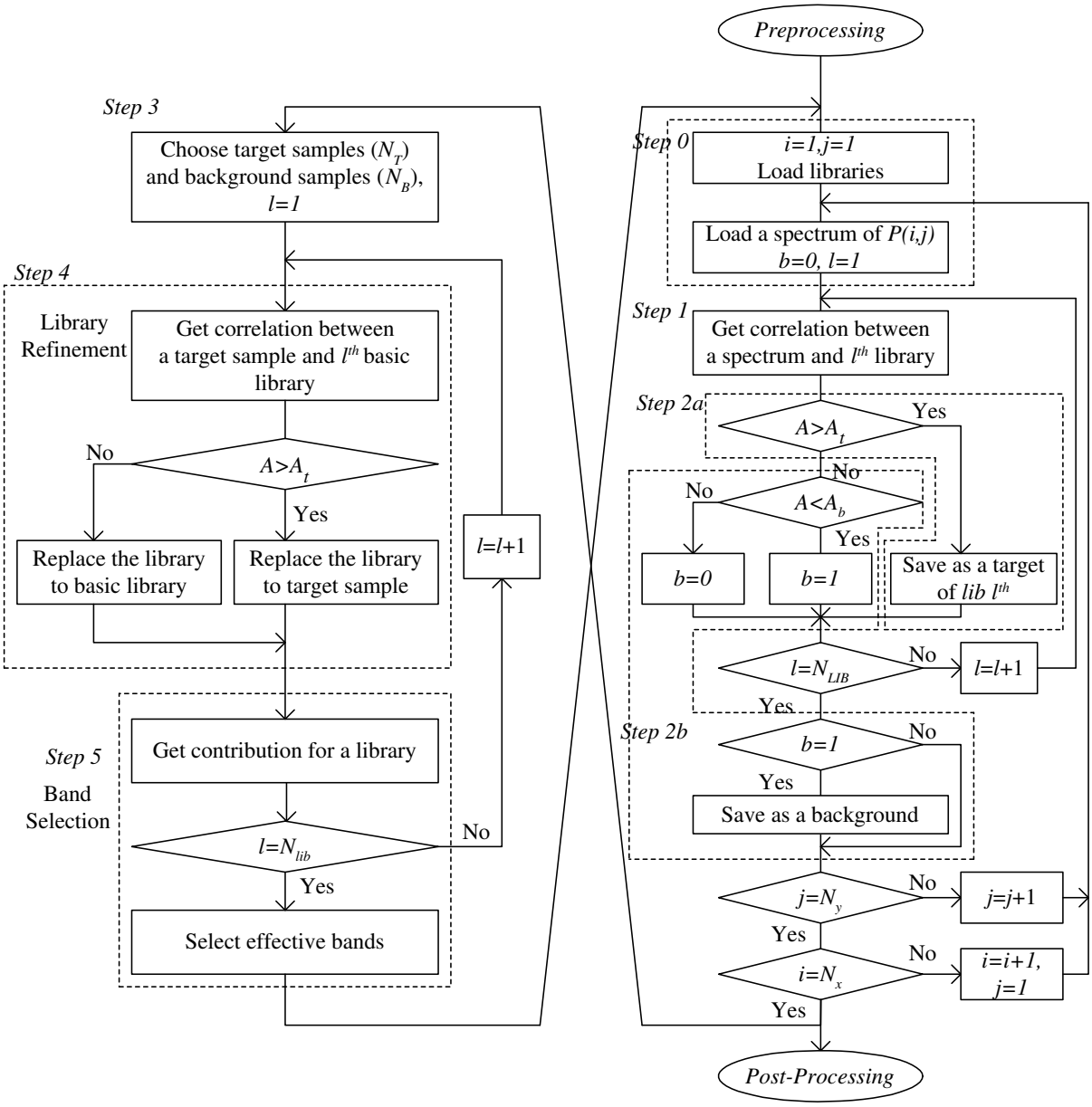


Figure 2-20: Flowchart of proposed algorithm for the detection process.

the number of bands (N_E), the number of libraries (N_{LIB}), the number of background samples (N_B) and the number of target samples (N_T), the minimum correlation coefficient between library and target (A_t), and the maximum correlation coefficient between library and background (A_b). The basic parameters are based on the type of the target and detecting environment. The output of Processing is a series of endmembers which represents a type of target.

2.5.2 Iteration Process

The algorithm repeats the following steps until $i = N_x$ and $j = N_y$ for a cube.

Step 0: Load spectrum contents in a pixel (i, j) and libraries. Initially, maximally separated bands are selected as effective bands. Then, from the next cube, effective bands are selected by Step 5. Thus, the number of spectrum contents is the same as the number of effective bands (N_E).

Step 1: Compute the correlation coefficient between an input spectrum and the l^{th} library.

Step 2: Classify each pixel (i, j) whether it is a target or a background; Step 2a is for target detection, and Step 2b is for background detection.

Step 2a: If the correlation coefficient (A) is higher than A_t, it is considered to be a target. Even though the libraries are only for a target, the detected results are saved separately for library refinement.

Step 2b: If A is lower than A_b, it can be a candidate for the background. Even if a spectrum of a pixel is not considered to be a target, it can be a target of other

libraries so that there is a tag bit which takes either False (0) or True value (1). After the loop for library refinement is completed with tag bit 1, it is classified as a background.

If the value A is between A_b and A_t, it is impossible for the pixel to be classified due to insufficient information. Thus, to save endmembers, $N_x \times N_y \times (N_{LIB} + 1)$ size of bit memories are required since the area size of x-y plane is $N_x \times N_y$ and each endmember requires a bit memory to save the information where 1 is the endmember and 0 is the unknown object. In addition, since the number of bits to save the type of the endmembers in a pixel is the sum of the number of libraries (N_{LIB}) and a background, the $(N_{LIB} + 1)$ bits are required for endmembers. For example, if there are three libraries, the required endmember bits are 4-bits. Furthermore, if all endmember bits are 0 (where background bit is also 0), it is classified as a background.

Step 3: Choose samples for background and target. To represent the spectrum of the background area, the samples of background are randomly selected where the number of background samples is N_B. For library refinement, each library uses one sample as a candidate to replace the current library. We assume the area of targets is much smaller than the area of background. All of the detected targets are counted and randomly selected in endmembers. If we count all backgrounds to select randomly, it makes excessive data loading so that we select N_B random pixels from the entire image.

Step 4: Refine current library. The sample is a candidate for the new library. Since the partial number of bands are used to obtain correlation in Step 1 and Step 2, the sample is compared to the basic library again for entire bands. If A is higher than

A_t where the correlation between the l^{th}library and a spectrum of a sample uses all of the bands of which size is N_z, the candidate replaces the current library. Otherwise, the current library goes back to the basic library. The refined library is saved to a memory for libraries.

Step 5: Select effective bands. From Step 4, we obtained the new library so that effective bands are changed to support the new library. Since the band selection is based on contribution, $(N_{LIB} \times N_B)$ operations are required to get contribution (c). From the distribution of contribution coefficient, N_E bands are selected.

Figure 2-21 shows the timing flow of hyperspectral processing algorithm. T_{init} represents the time interval for loading libraries and several coefficients such as the minimum correlation coefficient between library and input image (A_t), the maximum correlation coefficient between a library and an input image (A_b), the number of libraries (N_{LIB}), the number of target samples (N_T), the number of background samples (N_B) and the number of effective bands (N_E). T_{pixel} is the processing time for a pixel and the sum of T_{load}, T_{corr} and T_{detect} from Step 0 to Step 2 where T_{load} is the required time for the function $load()$ in Step 0, T_{corr} is for the function $corr()$ in Step 1 and T_{detect} is the required time for the function $detect()$ in Step 2. Thus, the total required time for a cube is $T_{init} + T_{pixel} \times N_x \times N_y + T_{choose\_samples} + T_{refine\_lib} + T_{get\_ebands}$ where $T_{choose\_samples}$ is the required time for the function $choose\_samples()$ in Step 3, T_{refine_lib} is for the function of $refine\_lib()$ in Step 4 and $T_{get\_ebands()}$ is the required time for the function $get\_ebands()$ in Step5.

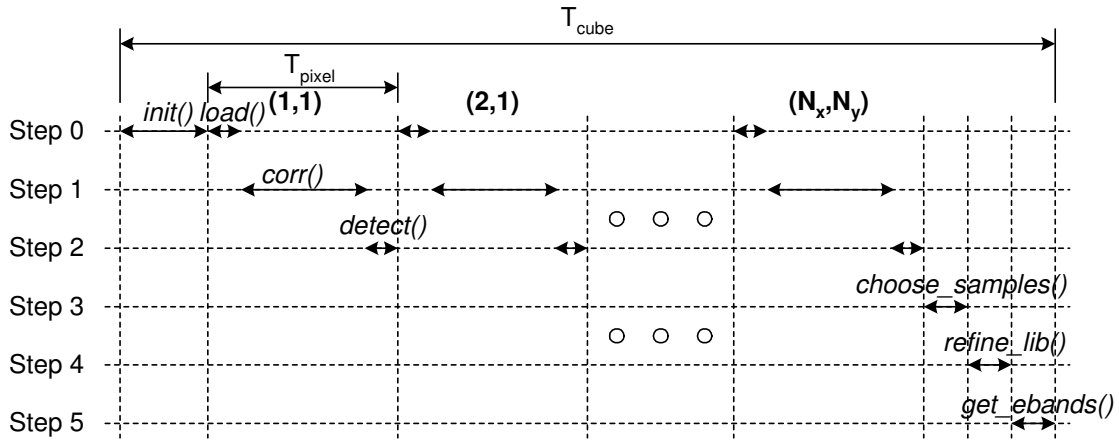


Figure 2-21: Illustration of time flow in Processing.

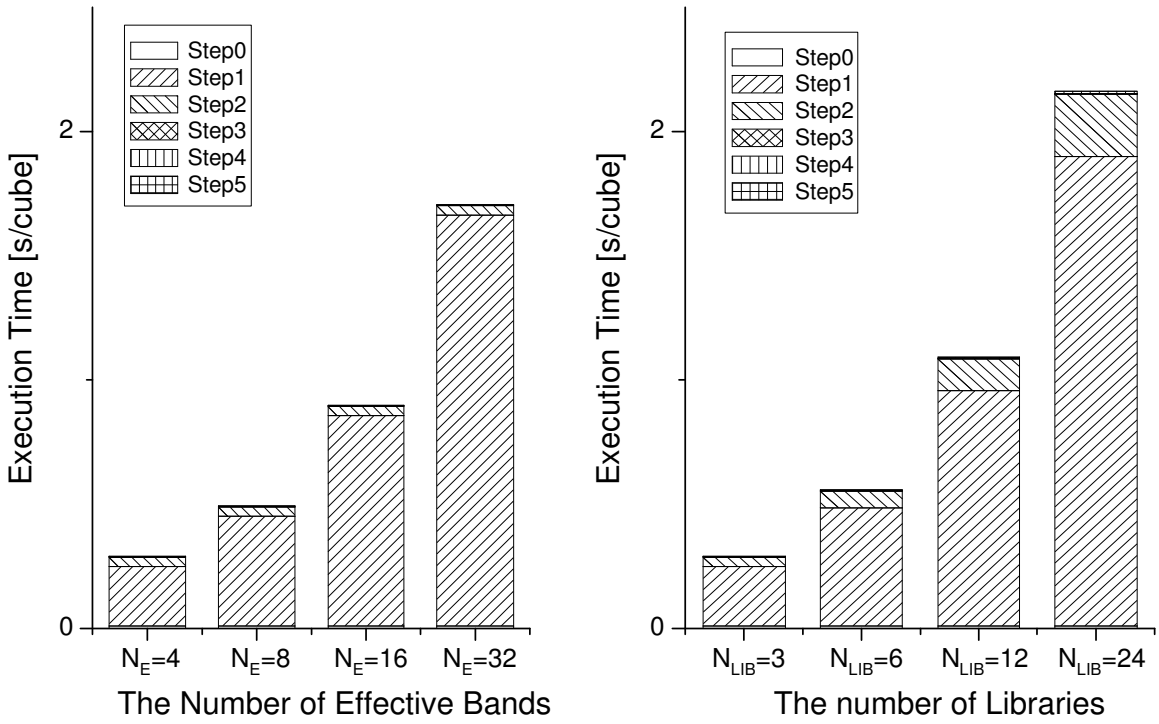
2.5.3 Complexity

The complexity of this algorithm has been estimated by TMS320C6713 (300MHz) based on the VLIW architecture. The internal program memory is structured so that a total of eight instructions can be fetched in every cycle [33] [34]. We estimate the execution time from the instruction cycle count using Code Composer Studio 3.1.

Figure 2-22(a) shows the execution time in terms of the number of bands used. The complexity of the system is directly related to the execution time. When the number of effective bands is increased, the complexity as well as the execution time are increased.

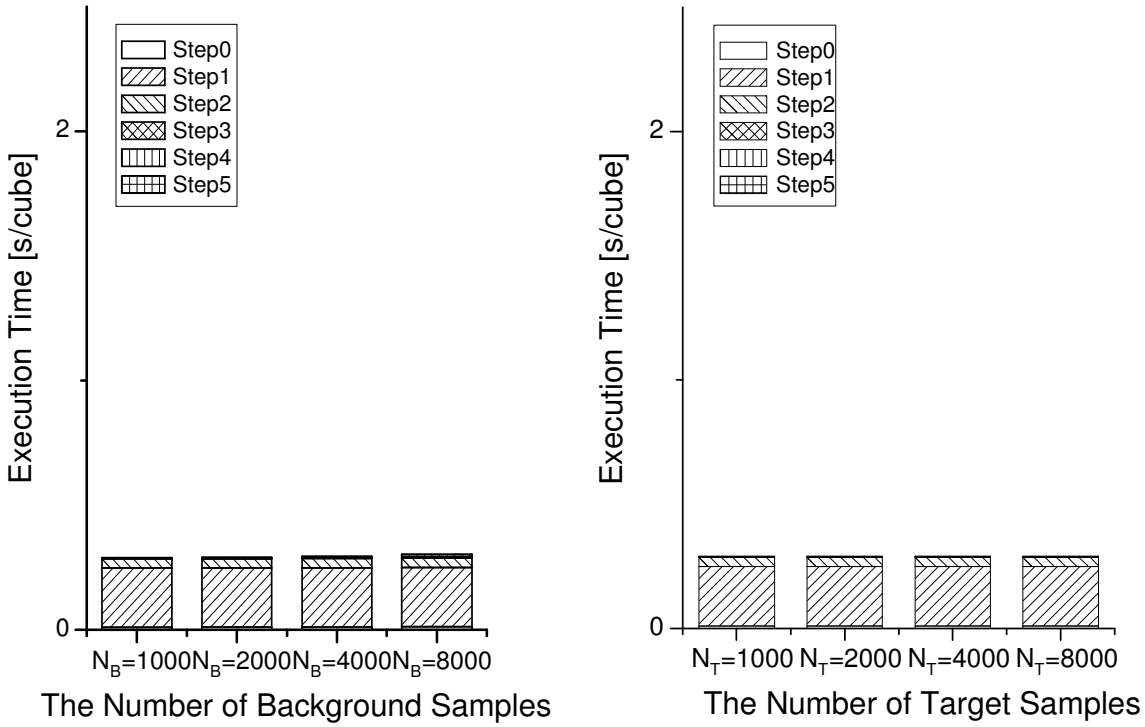
The computation complexity in terms of the number of target libraries is shown in Figure 2-22(b). The increasing rate of complexity is higher than the case shown in Figure 2-22(a) since the complexity of Step 2 is also increased as the number of libraries is increased.

The band selection is based on the relationship between backgrounds and libraries. The background samples represent the background area. Thus, the number of back-



(a) N_E (b) N_{LIB}

Figure 2-22: Illustration of the execution time in function of number of effective bands and the number of libraries. where (a) $N_{LIB} = 3$, $N_x = 820$, $N_y = 748$, $N_B = 1000$, $N_T = 1000$ (b) $N_E = 4$, $N_x = 820$, $N_y = 748$, $N_B = 1000$, $N_T = 1000$



(a) N_B (b) N_T

Figure 2-23: Illustration of the execution time in function of number of background samples or the number of target samples, where (a) $N_E = 4$, $N_{LIB} = 3$, $N_x = 820$, $N_y = 748$, $N_T = 1000$ (b) $N_E = 4$, $N_{LIB} = 3$, $N_x = 820$, $N_y = 748$, $N_B = 1000$

ground samples is important for the effective band selection. Figure 2-23(a) shows the complexity in terms of the number of background samples. When the number of background samples is larger, the complexity of Step 3 and Step 5 is increased. However, the total computation complexity is slightly increased.

The number of target samples is important for library refinement since the sample represents the detected image. Figure 2-23(b) shows the variation of computation complexity in terms of the number of background samples.

Chapter 3

Iterative Object Localization Algorithm Using Visual Images with Reference Coordinate Estimates

3.1 Introduction

In this chapter, a simplified algorithm for localizing multiple objects in a multiple-camera environment is proposed. Multiple-image based multiple-object detection and tracking are used in indoor and outdoor surveillance, and give a delicate and complete history of an interested object's action [2] [17] [18]. The object tracking can be simply concerned into a 2-D tracking problem on the ground plane [17] [35] [36] [37]. The

establishment of correspondences in multiple images can be achieved by using a field of view lines [17] [38]. Besides, for the selection of the best view about interested objects, a camera movement such as zooming and panning is required [35].

There are many localization methods which use image sensors [12] [39] [40] [41] [42] [43] [44] [45]. Most of conventional localization methods follow two steps of operation. Initially, the camera parameters are computed off-line using known objects or pattern images. Then using additional information such as control points in the scene or techniques such as structure from motion, the relative displacements of a camera are estimated [41] [46]. Basically, these studies can sufficiently localize objects from 3-D reconstruction. Once the sufficient number of points is observed in multiple images from different positions, it is mathematically possible to deduce the locations of the points as well as the positions of the original cameras, up to a known factor of scale [41]. In the localization method based on a perspective projection model, the camera calibration is critical. The calibration usually uses a flat plate with a regular pattern [13] [47] [48]. However, in many applications, it is not easy to obtain calibration patterns [49] [50]. In order to alleviate the effect of the calibration patterns, some methods based on self-calibration use the point matching from image sequences [49] [50] [51] [52] [53] [54]. In these methods, the image feature extraction should be very accurate since this procedure is very sensitive to the noise [41] [47] [55]. Moreover, if a pair of stereo images for a single scene are not calibrated and the motions between two images are unknown, the image matching requires prohibitively high complexity [47] [54] [55] [56].

The localization method based on affine reconstruction can be used for object

localization without the concern of the complex calibration [57] [58] [59] [60]. The method uses two uncalibrated perspective images where an image is induced by a plane to infinity [57] [58] [59] [61] [62] [63] [64]. Especially, the factorization method based on the paraperspective projection model can be used for localization [62] [64] [65]. However, the localization method based on the affine structure requires at least five correspondences in two images [57] [58] [59]. On the other hand, our proposed method requires only one correspondence (i.e., a centroid coordinate of the detected object) in two images, where each correspondence represents the same object. Thus, the critical requirements of an effective localization algorithm in tracking applications are the computational simplicity with a simpler model where 3-D reconstruction is not necessary, and the robust adaptation of camera's movement during tracking (i.e., zooming and panning) without requiring any additional imaging device calibration from the images.

Figure 3-1 illustrates the application model where multiple people are localized in a multiple-camera environment. The cameras can freely move with zooming and panning capabilities. Within a tracking environment, the proposed method uses detected object points to find object location.

The rest of this chapter is organized as follows. Section 3.2 briefly describes a parallel projection model with a single camera. Section 3.3 illustrates the visual localization algorithm in a 2-D coordinate with multiple cameras. In Section 3.4 we present analysis and simulation results where the localization errors are minimized by compensating for non-linearity of the digital imaging devices. An application that uses the proposed algorithm for tracking people within a closed environment is

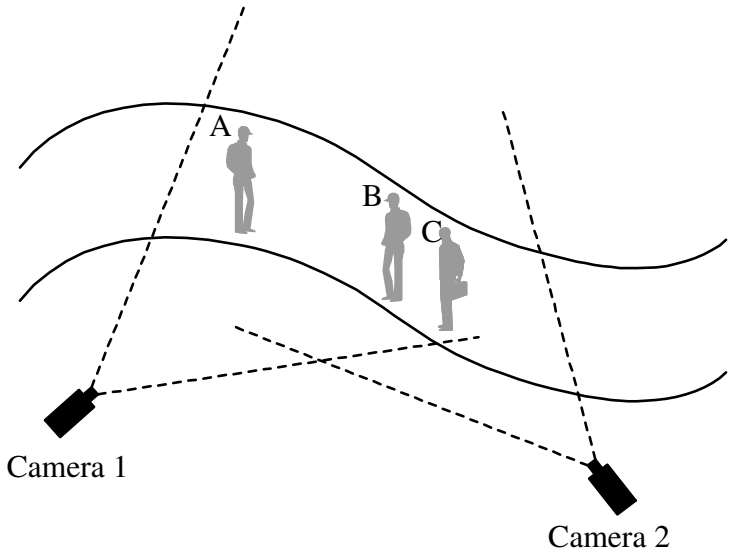


Figure 3-1: Illustration of the model of application.

illustrated.

3.2 Characterization of Viewable Images

3.2.1 Basic Concept of a Parallel Projection Model

In this section, we introduce a parallel projection model to simplify the visual localization, which is basically comprised of three planes: an object plane, a virtual viewable plane and an actual camera plane. In Figure 3-2, an object P placed on an object plane is projected to both a virtual viewable plane and an actual camera plane and P_p denotes the projected object point on the virtual viewable plane. The distance d_p denotes the distance between a virtual viewable plane and an object plane. u_p and u_{pp} denote the position of projected object P_p on both the actual camera plane and the virtual viewable plane. The virtual viewable plane is parallel with the object

plane by distance d_p. L_c and L_s denote each length of the virtual viewable plane and the actual camera plane, respectively.

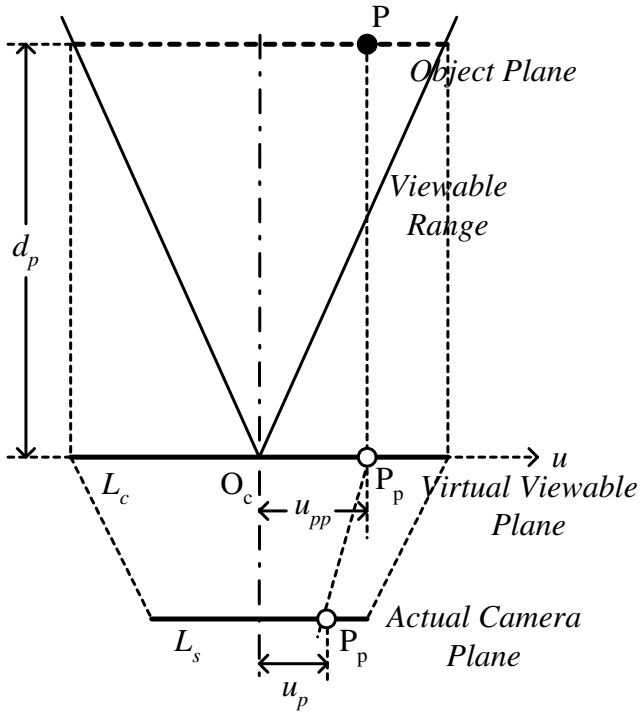


Figure 3-2: Illustration of the parallel projection model.

Since the size of image sensor is much smaller than the virtual viewable plane, the viewable range starts from a point O_c. Thus the camera model of parallel projection model is similar to a pin-hole camera. All planes are represented as u- and v-axis but we use u-axis for the explanation of the parallel projection model in this section. Since O_c represents the origin of both the virtual viewable plane and the camera plane, two planes are placed on the same camera position. However, in Figure 3-2, we drew two planes separately to show the relationship between three planes.

In the parallel projection model, an object is projected from an object plane through a virtual viewable plane to an actual camera plane. Hence, as formulated in

Equation (3.1), u_{pp} is expressed as L_c, L_s and u_p through the proportional lines of two planes as the following:

$$u_{pp} = u_p \left(\frac{L_c}{L_s} \right). \tag{3.1}$$

Thus the object P is represented from u_{pp} and the distance d_p between the virtual viewable plane and the object plane.

3.2.2 Zooming and Panning

Since the size of the virtual viewable plane and the object plane are proportional to the distance between the object and the camera (d_p), the length of the virtual viewable plane (L_c) is derived from the distance d_p and the viewable range.

Zooming factor represents the relationship between d_p and L_c. The zooming factor z is defined as a ratio of d_p and L_c as the following:

$$z = \frac{d_p}{L_c}. \tag{3.2}$$

Since both d_p and L_c use metric units, zooming factor z is a constant.

Figure 3-3 illustrates the model of zooming in terms of two different zooming factors. Even though the zooming factor of a camera has changed from z_1 to z_2, if the distance between object and camera is not changed, the position of projected object on the virtual viewable plane is not changed. In the figure, since the distance d_{p1} is equal from the distance d_{p2}, the position of the object on the virtual viewable plane is invariant but the position on the actual camera plane is variant. Thus the distance u_{pp1} is equal to u_{pp2} but the distance u_{p1} is different to the distance u_{p2}. The

projected positions u_{p1} and u_{p2} on the actual camera plane 1 and 2 are expressed as $u_{pp1} = u_{p1}(L_{c1}/L_s)$ and $u_{pp2} = u_{p2}(L_{c2}/L_s)$. Since $z_1 = d_{p1}/L_{c1}$ and $z_2 = d_{p2}/L_{c2}$, the relationship between u_{p1} and u_{p2} is represented as $u_{p1} = u_{p2}(z_2/z_1)$.

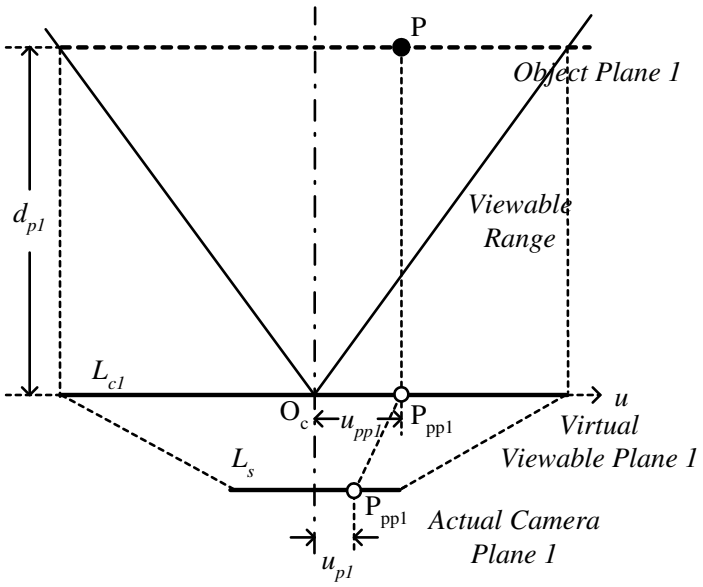
Figure 3-4 illustrates a special case in which two different objects denoted P_1 and P_2 are projected to the same spot on the actual camera plane. P_{p1} and P_{p2} denote the projected objects on the virtual viewable plane 1 and 2.

The object P_1 and P_2 are projected to a point on the actual camera plane while two objects are separated as two different points on the virtual viewable plane 1 and 2. Since the zooming factor z is equal to d_1/L_{c1} and d_2/L_{c2}, the relationship between the distance u_{pp1} and u_{pp2} is expressed as $u_{pp1} = u_{pp2}(d_1/d_2)$. The distance u_{p1} is equal to the distance u_{p2} and the distance u_{pp1} is different from the distance u_{pp2}. It is shown that the distance in projection direction between an object and a camera is an important parameter for the object localization.

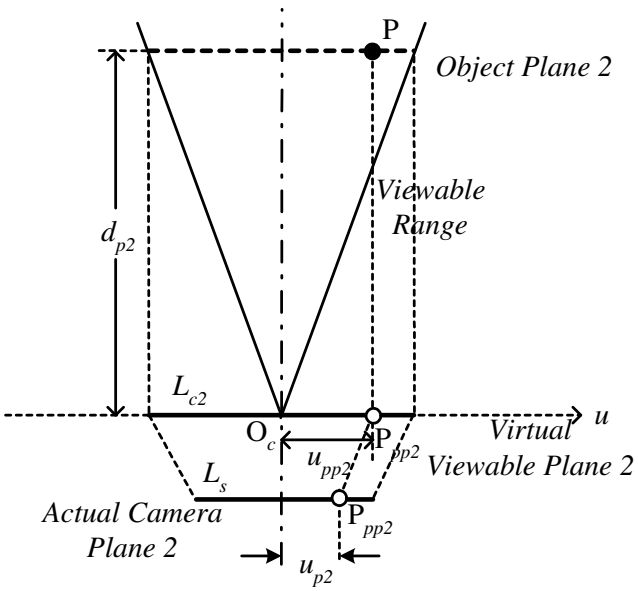
Now, we consider a panning factor denoted as θ_p that represents camera rotation. The panning angle is defined as the angle difference between n-axis and u-axis where n-axis represents the normal direction of the virtual viewable plane. Thus the panning angle can exist in the range of $-\pi/2 < \theta_p < \pi/2$. The sign of θ_p is determined: the left rotation is positive and the right rotation is negative.

To get the global coordinate of the object, u-axis and v-axis in camera coordinate are translated to x-axis and y-axis in global coordinate. We define camera angle factor (θ_c) to represent the absolute camera angle in global coordinate. The camera angle θ_c is useful to translate the object coordinate from camera images.

Figure 3-5 illustrates the relationship between the camera angle θ_c and the panning



(a) Small zooming factor(z_1)



(b) Large zooming factor (z_2)

Figure 3-3: Illustration of the model of zooming in terms of two different zooming factors.

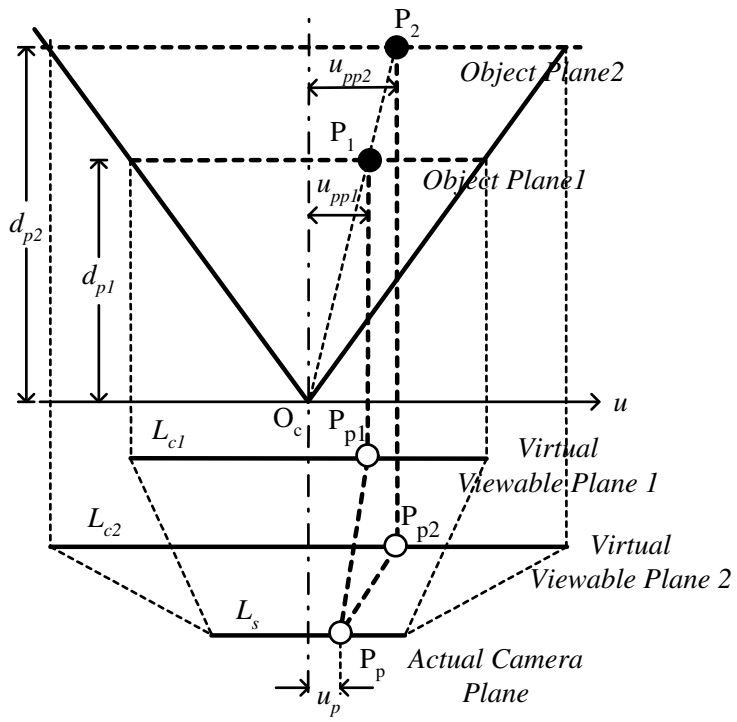


Figure 3-4: Illustration of a special case in which different objects are projected to the same spot on the actual camera plane.

angle θ_p in global coordinate. The global coordinate is represented as x-axis and y-axis. For example, in the position of Camera A, panning angle θ_p is the angle between n- and y-axis while in Camera D, the panning angle is the angle between n-axis and x-axis. Thus four cases of camera deployment such as Camera A, B, C, D, have different relationships between θ_c and θ_p. Thus the projected object $P_p(x_{pp}, y_{pp})$ on the virtual viewable plane is derived from $x_{pp} = x_c + u_{pp} \cos\theta_c$ and $y_{pp} = y_c + u_{pp} \sin\theta_c$. $O_c(x_c, y_c)$ denotes the origin on the virtual viewable plane in global coordinate.

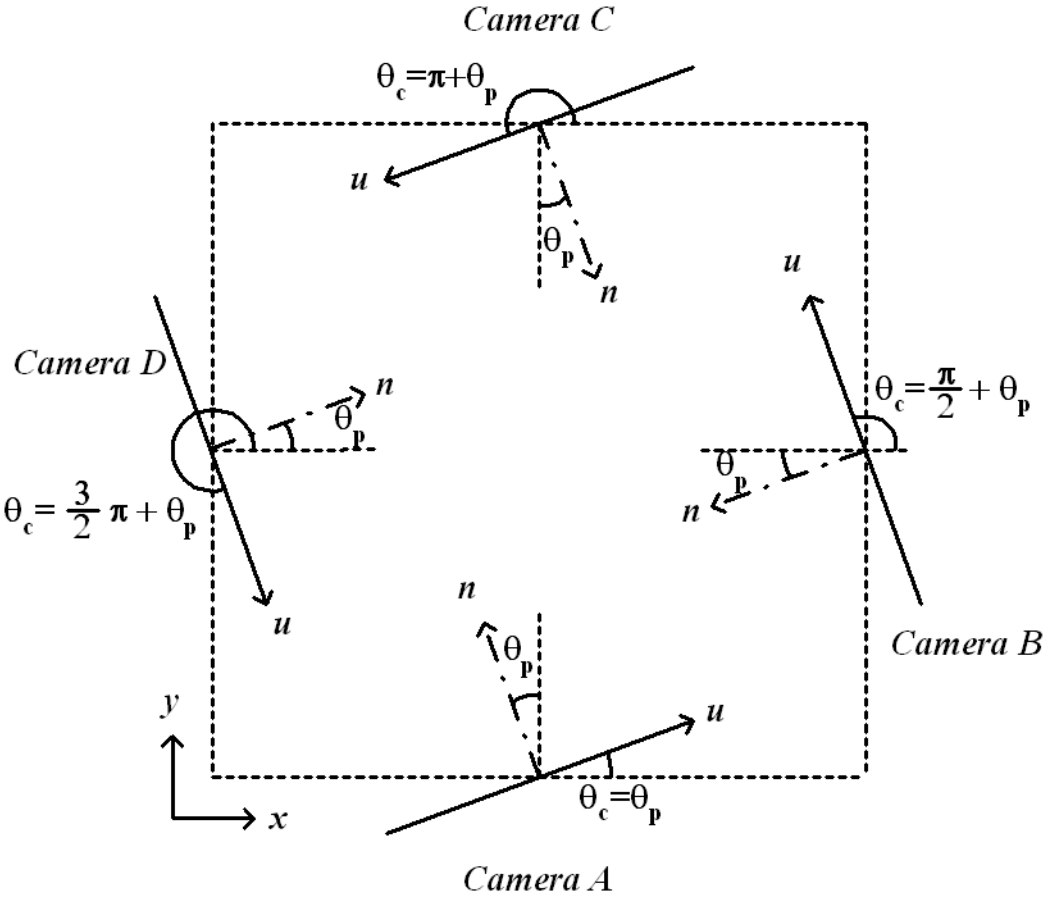
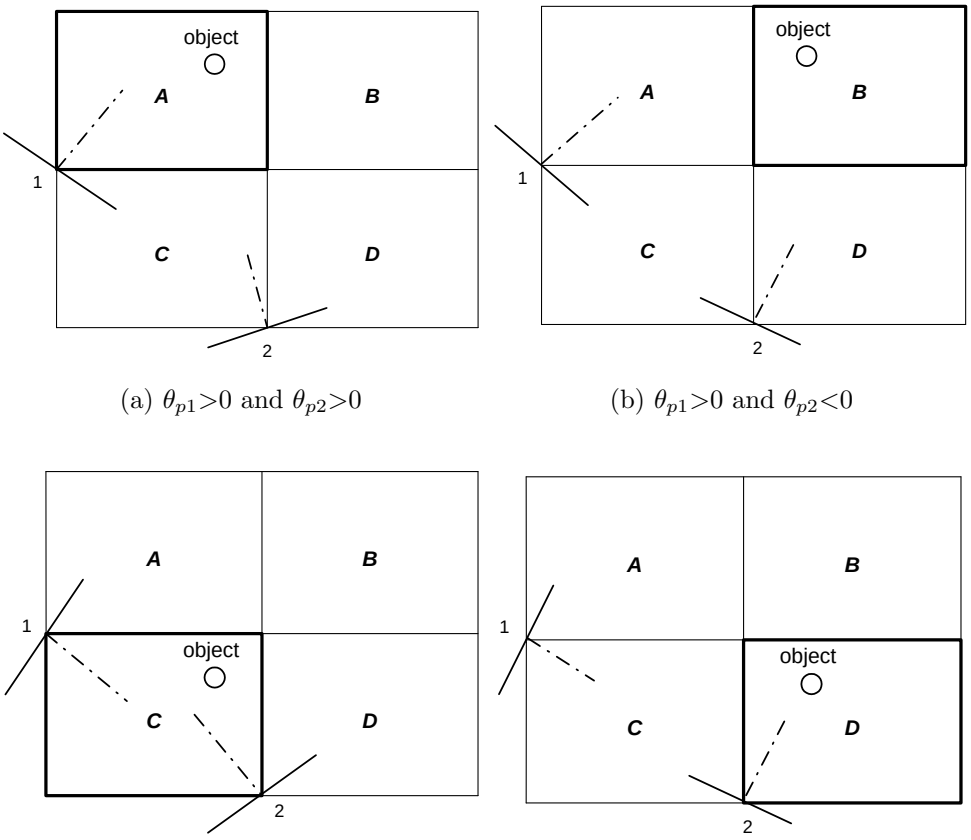


Figure 3-5: Illustration of individual panning factors with respect to a global coordinate.

3.2.3 The Relationship between Camera Positions and Pan Factors

Figure 3-6 illustrates the panning factor selection in a pair of cameras depending on an object position. Among deployment of four possible cameras, such as Camera A, B, C, and D, a pair of cameras located in adjacent axes are chosen.



(a) $\theta_{p1}>0$ and $\theta_{p2}>0$

(b) $\theta_{p1}>0$ and $\theta_{p2}<0$

(c) $\theta_{p1}<0$ and $\theta_{p2}>0$

(d) $\theta_{p1}<0$ and $\theta_{p2}<0$

Figure 3-6: Illustration of panning factor selection in a pair of cameras depending on an object position.

In this chapter, we choose Camera A and D for the deployment of two cameras for the sake of the localization formulation. The camera angle in Camera A and D

are expressed as $\theta_c = \theta_p$ and $\theta_c = \theta_p + \frac{3}{2}\pi$ in terms of the panning angle θ_p.

3.3 Visual Localization Algorithm In A 2-Dimensional Coordinate

3.3.1 The Concept of Visual Localization

Turning to the object localization with an estimate, consider a single camera based localization. In the single camera localization, we use the estimate plane as an object plane. Figure 3-7 illustrates the object localization using the estimate E based on a single camera where E denotes the estimate which is used for a reference point. The estimate E and the object P are projected to two planes: virtual viewable plane and actual camera plane. Here, the reference point E generates the object plane. The distance d_e denotes the distance between the estimate and the virtual viewable plane. In view of the projected positions, the length l_p is obtained by the length l_{ps}. Hence the object $P(x_p, y_p)$ is determined from the estimate $E(x_e, y_e)$.

Once we use the estimate plane as an object plane, the object position P is different from the object position P_r. In other words, since any points on the ray between the object and origin are projected to the same spot on the actual camera plane, the real object P_r is distorted to the point P. Thus the localization has an error from the distance difference of the distance d_p and d_e. Through the single image sensor based visual projection method, it is shown that an approximated localization is accomplished with a reference point.

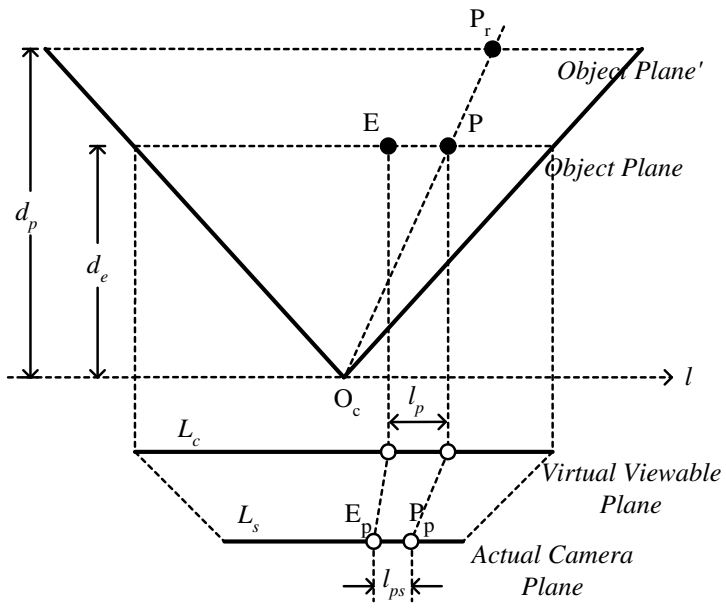


Figure 3-7: Illustration of the visual localization in a single camera.

We are now motivated to use multiple image sensors in order to reduce the error between P_r and P. In the case of single camera, the distance difference between the distance d_p and d_e cannot be found by a single camera view. However, if an additional camera is available for localizing the object within different angles, the distance difference can be compensated by the relationship between two camera views.

Figure 3-8 illustrates the localization using two cameras for a simple case where both panning factors are zero and the direction of l_1- and l_2-axis are aligned to y- and x-axis. Given by a reference point E, the virtual viewable planes for two cameras are determined. P_{r1} and P_{r2} are the obtained object coordinates in each single camera. In view of camera 1, the length l_{p1} between the projected points P_{p1} and E_{p1} supports the distance between the object plane of camera 2 and the point P. Similarly, in the view of camera 2, the length l_{p2} between the projected points P_{p2} and E_{p2} supports a distance between the object plane of camera 2 and the point P. Therefore, the basic

compensation algorithm is that camera 1 compensates y-direction by the length l_{p1}, and camera 2 compensates x-direction by the length l_{p1} given by a reference point E.

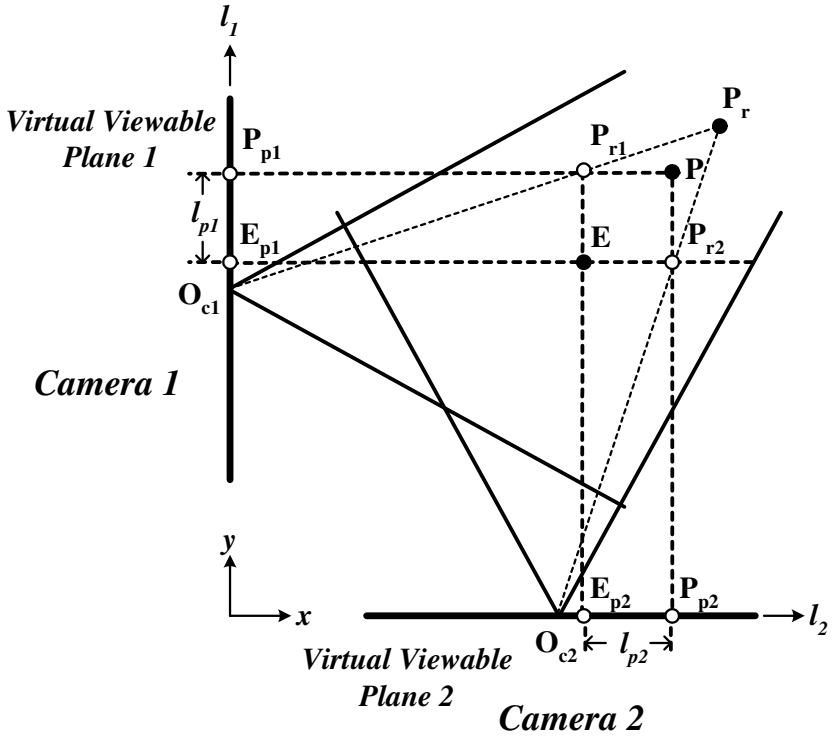


Figure 3-8: Illustration of the localization in multiple cameras.

Through one additional image sensor, both l_1 in y-direction and l_2 in x-direction make a reference point $E(x_e, y_e)$ closer to a real object position. Hence $P(x_p, y_p)$ is computed by $x_p = x_e + l_{p2}$ and $y_p = y_e + l_{p1}$. Note that P is the localized object position through the two cameras, which still results in an error with the real object position P_r. The error can be reduced by obtaining a reference point E closer to a real position P_r. In Section 3.3.5, an iterative approach is introduced for improving localization. In the next section, we formulate the multiple image sensors based localization.

3.3.2 2-D Localization

2-D Localization Model

In this section, we introduce a simplified localization model. If the estimate E and the object P have the same z-coordinate and v-axis is aligned with z-axis, all points are placed on a plane. Thus the localization is simplified in 2-D coordinate. The 2-D localization is simple and has an advantage for mapping the test environment. Moreover, once the object is represented as $P(x_p, y_p)$ in global coordinate, the 2-D localization gives a feasible solution.

To derive 2-D localization equations, we use vector notations which has a benefit to express the relationship between the estimate and the object where $'\widehat{}'$ denotes an unit vector and $'\rightarrow'$ represents a vector. For example, one vector $\overrightarrow{r}$ is represented as $A\widehat{a}_x + B\widehat{a}_y + C\widehat{a}_z$ where $\widehat{a}_x$, $\widehat{a}_y$ and $\widehat{a}_z$ denote unit vectors toward x-, y- and z-axis and A, B and C are the magnitude of x-, y- and z-axis, respectively. Figure 3-9 shows the basic model of object localization. The vector $\overrightarrow{l}$, $\overrightarrow{l}_{p1}$ and $\overrightarrow{l}_{p2}$ denote the vector from the estimate E to the object P, the vector from the projected estimate E_{p1} to the projected object P_{p1} on the virtual viewable plane 1, and the vector from the projected estimate E_{p2} to the projected object P_{p2} on the virtual viewable plane 2, respectively. The length l_{p1} and l_{p1} are the projections of the vector $\overrightarrow{l}$ on the virtual viewable plane 1 and 2.

Figure 3-10 shows the projected image on the virtual viewable plane 1 and 2 where the projected point P_{p1} and P_{p2} are expressed as $P_{p1}(u_{pp1}, v_{pp1})$ and $P_{p2}(u_{pp2}, v_{pp2})$ on the virtual viewable plane 1 and 2. z_{p1} and z_{p2} denote the z-coordinates of the

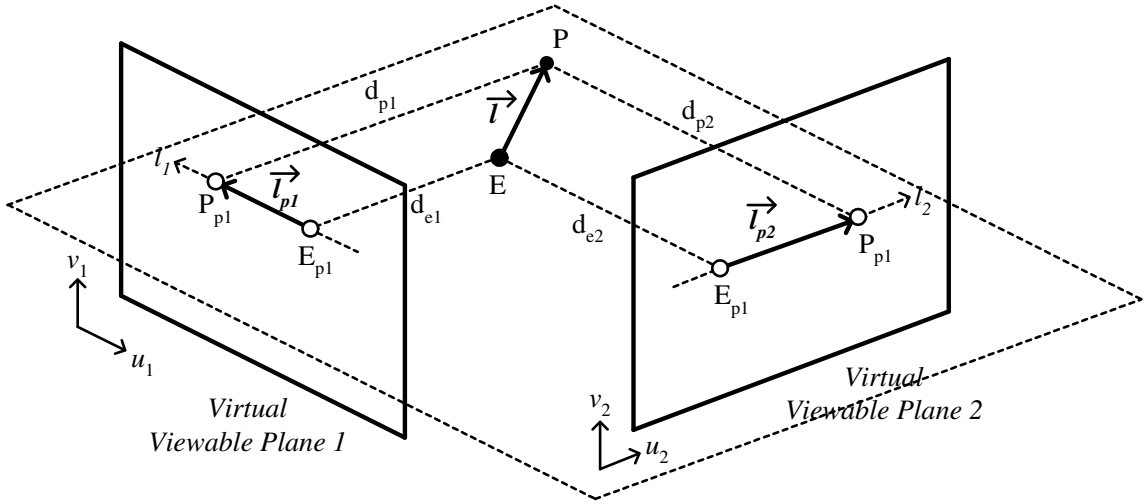


Figure 3-9: Illustration of basic localization algorithm.

projected objects in global coordinate and equal to $z_{c1} + v_{pp1}$ and $z_{c2} + v_{pp2}$. Since the estimate has some height with the object, the projected estimate and object have the same z-coordinate on the virtual viewable plane 1 and 2. Thus in the figure, v_{pp1} is different from v_{pp2} while z_{p1} is equal to z_{p2}. Since an estimate is a reference point, the actual estimates in the figure are not displayed on the actual camera plane. Since the projected vectors $\overrightarrow{l}_{p1}$ and $\overrightarrow{l}_{p2}$ are the projection of vector $\overrightarrow{l}$ toward l_1-axis and l_2-axis, the length l_{p1} and l_{p2} are equal to $\overrightarrow{l} \cdot \widehat{a}_{l_1}$ and $\overrightarrow{l} \cdot \widehat{a}_{l_2}$.

Object Localization based on a Single Camera

The projected object $P_p(l_{pp})$ in l-axis is transformed into $P_p(x_{pp}, y_{pp})$ in global coordinate. The origin $O_c(x_c, y_c)$ is the center of virtual viewable plane. The camera deployment is expressed as the origin $O_c(x_c, y_c)$ and camera angle θ_c.

Figure 3-11 shows the estimation with a reference point and a projected object. $\overrightarrow{l_p}$ denotes the vector from the origin O_c to the estimate E. The object P, estimate E, projected objects P_p and projected estimates E_p are denoted as $P(x_p, y_p)$, $E(x_e, y_e)$,

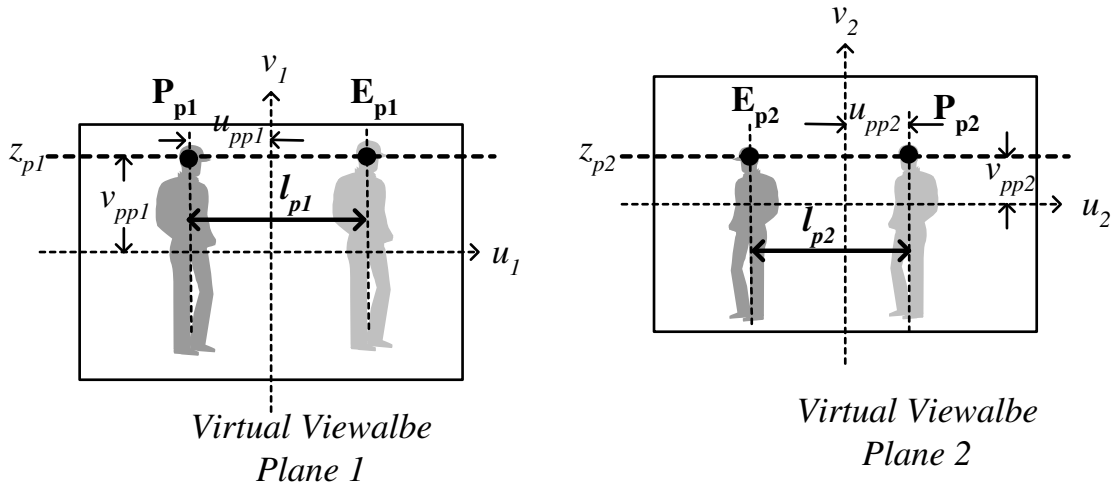


Figure 3-10: Illustration of the projected images on the virtual viewable plane 1 and 2.

$P_p(x_{pp}, y_{pp})$ and $E_p(x_{pe}, y_{pe})$ in global coordinate. The vector $\overrightarrow{l_p}$ is expressed in two ways which have different points of view; $\overrightarrow{l_p} = (l_{pp} - l_{pe})\widehat{a}_l$ on the virtual viewable plane and $\overrightarrow{l_p} = (x_p - x_c)\widehat{a}_x + (y_p - y_c)\widehat{a}_y$ in global coordinate.

The unit vector $\widehat{a}_l$ is represented in global coordinate as $\widehat{a}_l = \cos\theta_c\widehat{a}_x + \sin\theta_c\widehat{a}_y$. The vector $\overrightarrow{e}$ is expressed as $(x_e - x_c)\widehat{a}_x + (y_e - y_c)\widehat{a}_y$. Since the length l_{pe} is equal to the projection of vector $\overrightarrow{e}$ toward l-axis ($\overrightarrow{e} \cdot a_l$), the length l_{pe} is represented as:

$$l_{pe} = (x_e - x_c)\cos\theta_c + (y_e - y_c)\sin\theta_c. \tag{3.3}$$

Once we assume the estimate is close to the object, the length l_{pp} is represented as:

$$l_{pp} = l_{ps}\left(\frac{d_p}{zL_s}\right) \simeq l_{ps}\left(\frac{d_e}{zL_s}\right), \tag{3.4}$$

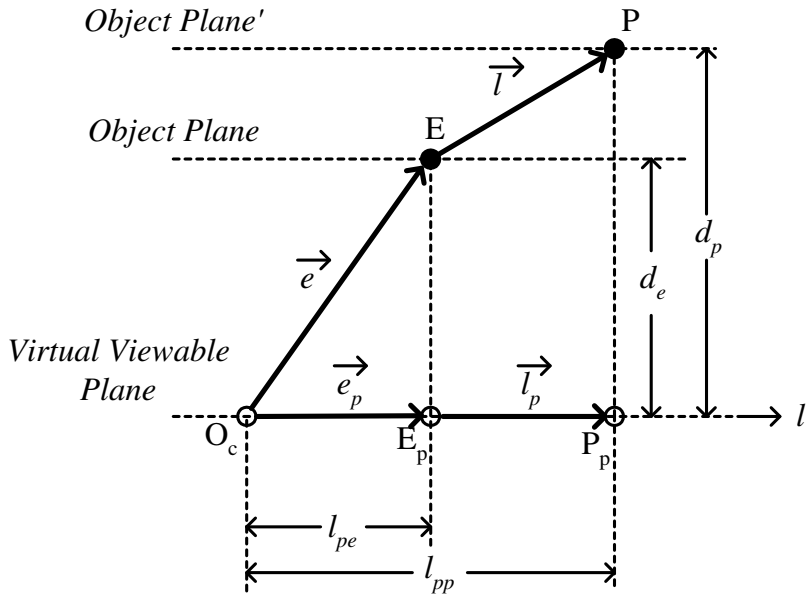


Figure 3-11: The estimation of a projected object.

where the length l_{ps} is the length of the projected estimate and object on the actual camera plane.

In Figure 3-11, since the vector $\overrightarrow{l}_p$ is equal to $l_{pp}\widehat{a}_l - l_{pe}\widehat{a}_l$, the length of vector l_p is represented as the following:

$$l_p = l_{pp} - l_{pe}. \tag{3.5}$$

Since the length l_p is the projection of the vector $\overrightarrow{l}$ toward l-axis ($\overrightarrow{l} \cdot \widehat{a}_l$), the global coordinate $P(x_p, y_p)$ is related with $E(x_e, y_e)$ as the following:

$$(x_p - x_e)\cos\theta_{c1} + (y_p - y_e)\sin\theta_c = l_p. \tag{3.6}$$

Note that since there are two unknown values of $P(x_p, y_p)$, two equations are necessary.

Object Localization Based on Multiple Cameras

As shown in Figure 3-9, Once there are two available cameras which show an object at the same time, two cameras have the following relationship:

$$\begin{aligned}(x_p - x_e)\cos\theta_{c1} + (y_p - y_e)\sin\theta_{c1} &= l_{p1},\\ (x_p - x_e)\cos\theta_{c2} + (y_p - y_e)\sin\theta_{c2} &= l_{p2}.\end{aligned} \tag{3.7}$$

The projected vector sizes of the vector $\overrightarrow{l}_{p1}$ and $\overrightarrow{l}_{p2}$ are derived from $l_{p1} = l_{pp1} - l_{pe1}$ and $l_{p2} = l_{pp2} - l_{pe2}$ in Equation (3.5). The length l_{pp1} and l_{pp2} are represented as $l_{pp1} \simeq l_{p1}(d_{e1}/z_1 L_{s1})$ and $l_{pp2} \simeq l_{p2}(d_{e2}/z_2 L_{s2})$ in Equation (3.4). The length between O_{c1} and P_{p1} in an actual camera plane (l_{p1}) and the length between O_{c2} and P_{p2} in an actual camera plane (l_{p2}) are obtained from displayed images.

Therefore, the object position $P(x_p, y_p)$ is represented as the following:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} \cos\theta_{c1} & \sin\theta_{c1} \\ \cos\theta_{c2} & \sin\theta_{c2} \end{bmatrix}^{-1} \begin{bmatrix} l_{p1} \\ l_{p2} \end{bmatrix}. \tag{3.8}$$

3.3.3 Effect of Zooming and Lens Distortion

The errors caused by zooming effect and lens distortion are the reason of scale distortion. In practice, since every general camera lens has non-linear viewable range, the zooming factor is not a constant. Moreover, since a reference point is a rough estimate, the distance d_p could be different from the distance d_e. However, in Equation 3.4, the distance d_e, instead of the distance d_p, is used to get the length l_{pp}.

Figure 3-12 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.

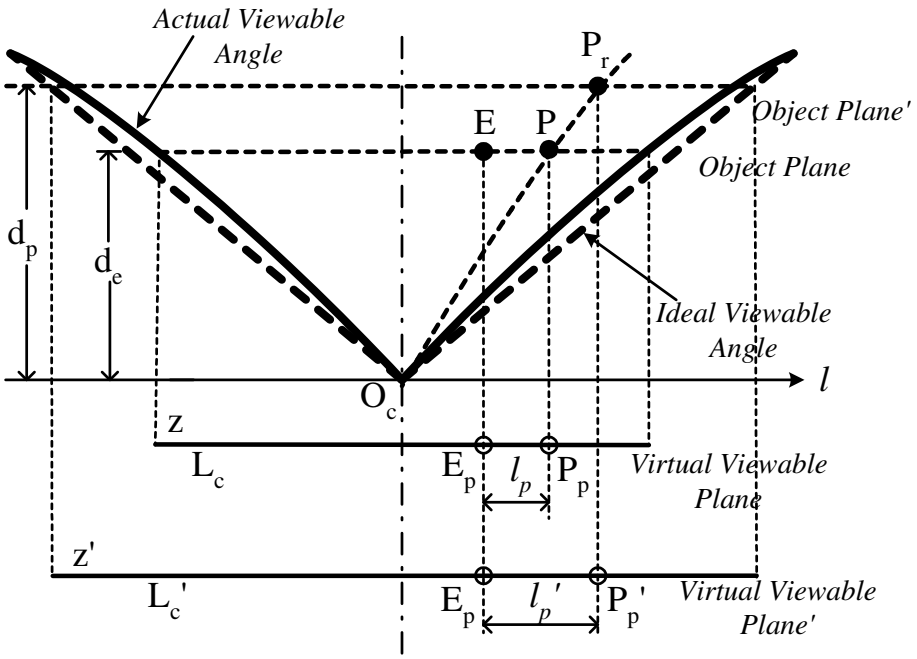


Figure 3-12: Illustration of actual zooming model caused by lens distortion.

For reference, zooming distortion is illustrated in Figure 3-13 with the function of distance from the camera and various actual zooming factors measured by Canon Digital Rabel XT with Tamron SP AF 17-50mm Zoom Lens [66] [67] where the

dashed line is the ideal zooming factor and the solid line is the actual (non-ideal) zooming factor. As the distance increases, the non-linearity property of zooming factor decreases.

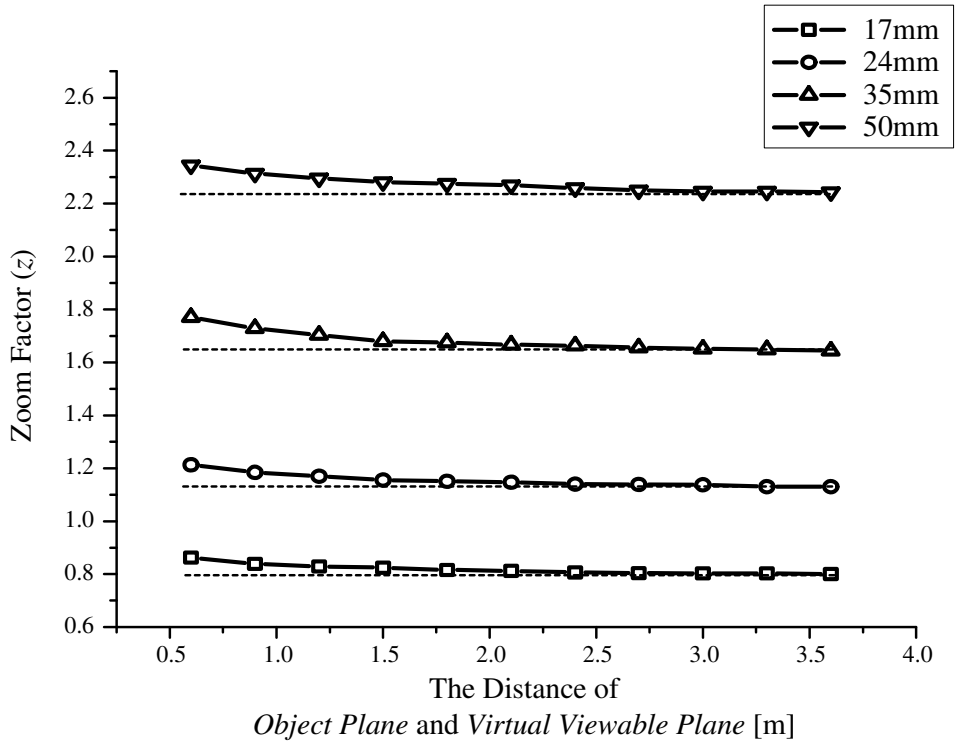


Figure 3-13: Illustration of zooming distortion on a function of distance from the camera and various actual zooming factors used.

To reduce the localization error, we update the length l_p. The lengths l_{pp} and l'_{pp} are equal to $l_{ps}(L_c/L_s)$ and $l_{ps}(L'_c/L_s)$,respectively. Due to the definition of zooming factor, z and z' are expressed as d_e/L_c and d_p/L'_c. Since the object P and P_r are projected at the same point on the actual camera plane in Figure 3-12, P and P_r have the same length l_{ps} on the actual camera plane. Thus the actual length l'_p is represented as the following:

$$l'_p = l_{pp}\left(\frac{d_p}{d_e}\right)\left(\frac{z}{z'}\right) - l_{pe}. \tag{3.9}$$

The distance d_e and d_p are derived from:

$$\begin{aligned} d_e &= \sqrt{(x_e - x_{pe})^2 + (y_e - y_{pe})^2}, \\ d_p &= \sqrt{(x_p - x_{pp})^2 + (y_p - y_{pp})^2}, \end{aligned} \tag{3.10}$$

where x_{pe}, y_{pe}, x_{pp} and y_{pp}, are equal to $x_c + l_{pe}\cos\theta_c$, $y_c + l_{pe}\sin\theta_c$, $x_c + l_{pp}\cos\theta_c$ and $y_c + l_{pp}\sin\theta_c$, respectively.

Finally, the compensated object position $P(x_{pr}, y_{pr})$ is determined as the following:

$$\begin{bmatrix} x_{pr} \\ y_{pr} \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} \cos\theta_{c1} & \sin\theta_{c1} \\ \cos\theta_{c2} & \sin\theta_{c2} \end{bmatrix}^{-1} \begin{bmatrix} l'_{p1} \\ l'_{p2} \end{bmatrix}, \tag{3.11}$$

where the length l'_{p1} and l'_{p2} are equal to $l_{pp1}\,(d_{p1}/d_{e1})\,(z_1/z'_1) - l_{pe1}$ and $l_{pp2}\,(d_{p2}/d_{e2})\,(z_2/z'_2) - l_{pe1}$, respectively.

3.3.4 Effect of Lens Shape

The virtual viewable plane is a plane and real camera displays a curved space. Thus unit distances per pixel in u- and v-axis are non-linear on the actual camera plane. Figure 3-14 shows the error caused by lens shape where the distance d_{p1} and d_{p2}

denote two different distances between the estimates and the camera.

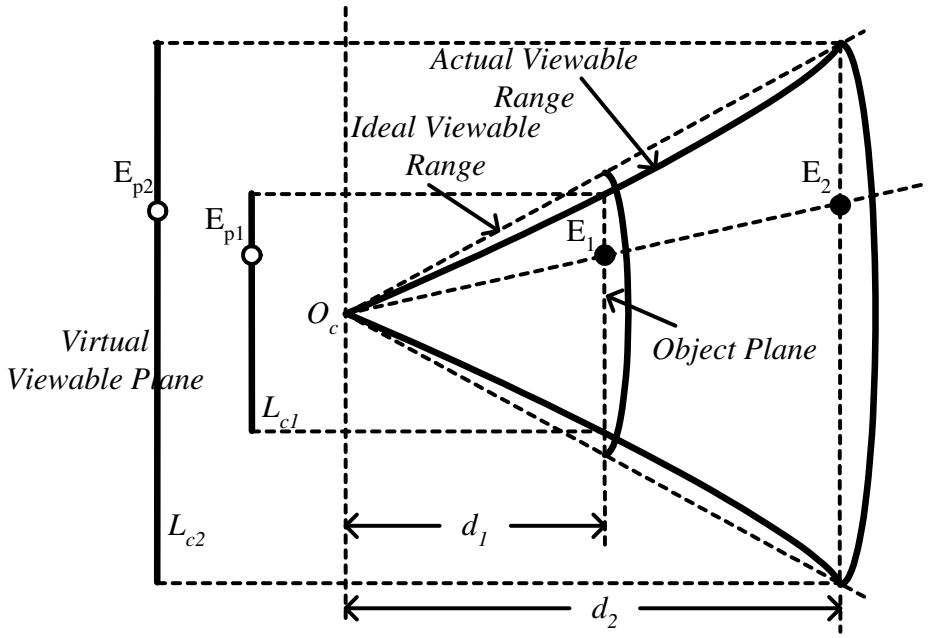
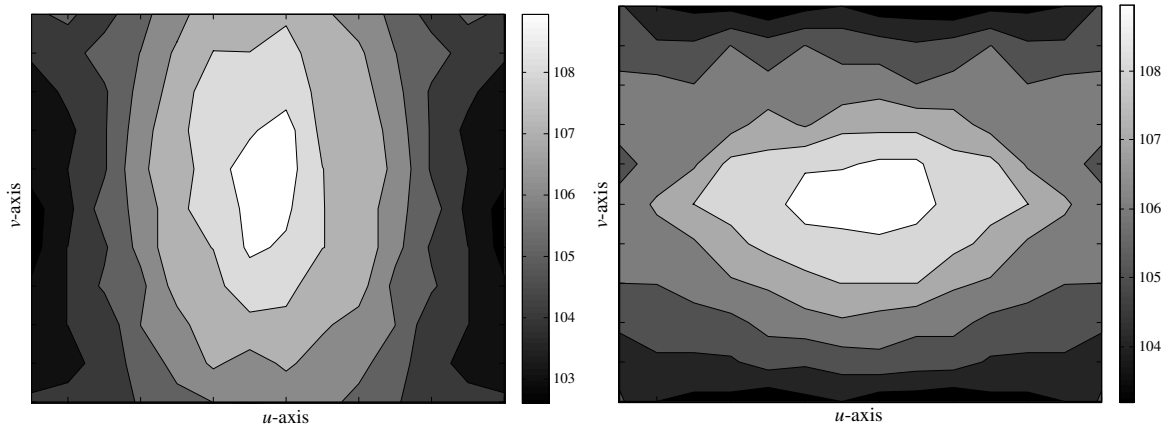


Figure 3-14: Illustration of the error caused by lens shape.

Figure 3-15 illustrate the distribution of unit distance of u- and v-axis on the actual camera plane. The distance between camera and calibration sheet is 35 inch and an unit distance is 1 inch.

The translation of the distance between the estimate and the object needs the compensation for the non-linearity by camera calibration. In Figure 3-15(a), the unit distance for u-axis is invariant in v-axis and in Figure 3-15(b), the unit distance for v-axis is also invariant in u-axis. Hence in Fig 3-10, the height difference of two different cameras have little effect for the overall localization error.



(a) Δu (b) Δv

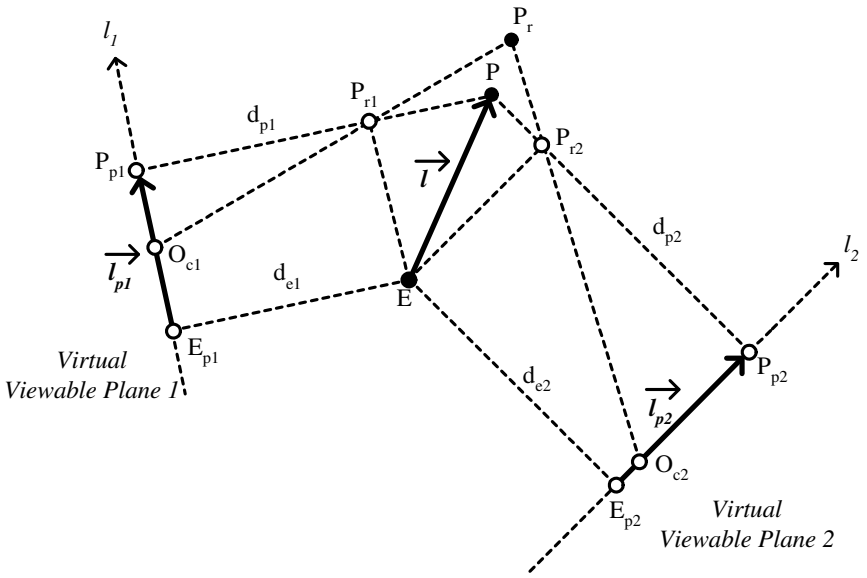
Figure 3-15: Illustration of unit distance distribution due to camera non-linearity on the actual camera plane.

3.3.5 Iterative Localization for Error Minimization

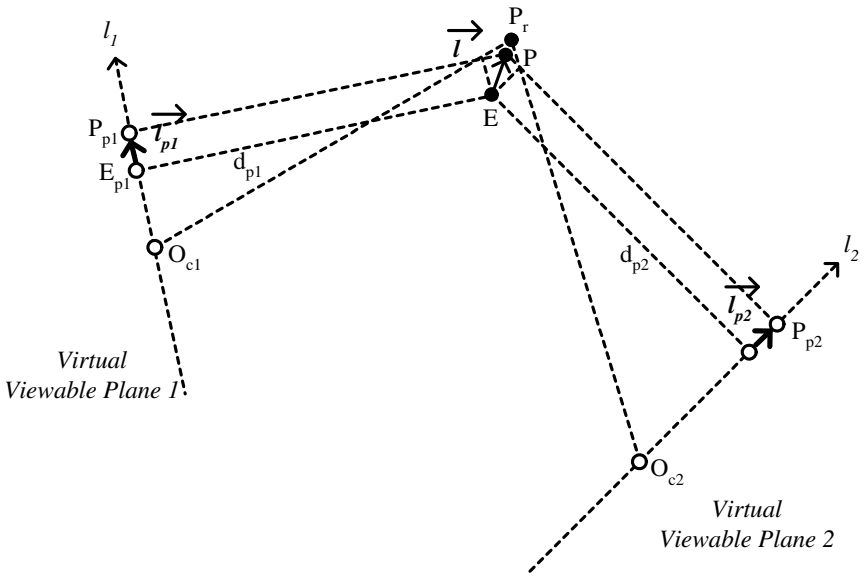
Once the virtual viewable plane is defined by the estimate, the localized result has the error caused by the distance difference between the estimate E and the real object P_r. Thus the distance between the object and the estimate is important for reducing the localization error.

The basic concept of iterative approach is to use the previous localized position P as a new reference point E for the localization of object P_r. Thus since the reference point E is closer to a real position P_r, the localized position P is getting closer to a real position P_r.

Figure 3-16(a) illustrates the basic localization based on two cameras where P_r represents the real object. If the distance d_p is equal to the distance d_e, the obtained object coordinate uses the coordinate of P_{p1} and P_{p2} to translate the global coordinate of the object. Thus the object point P is closer to the real object point P_r.



(a) Basic



(b) First Iteration

Figure 3-16: Illustration of iterative localization.

Figure 3-16(b) shows the iterative localization. Each iteration gives closer object coordinate with relative computational complexity. Thus the iterative approach can reduce the localization error. Furthermore, through the iteration process, the localization is becoming insensitive to the non-linear properties.

3.3.6 Discussion

Object Height Insensitivity

So far, we have discussed an object's localization which is mapped in 2-D coordinate. Since the observed object is localized in 2-D coordinate, the different up-down angles (azimuth of the camera viewable direction) of cameras do not affect any results. This is because the solution space is on the 2-D plane. Hence the localization principle still works in the x-y plane even in a situation where there is any height mismatch between the cameras and/or object in z-axis direction. Hence the proposed parallel projection model is insensitive to the object height.

Reference Point

The proposed localization algorithm localizes multiple objects with the same number of reference points. The reference initially decides the distance between the object and the camera. While it is not necessary, if the reference point is established by Kalman filter or Particle filter (i.e., estimation of the object using acoustic sensor), it has a benefit of lowering the computation since closer initial estimate to the actual object coordinate can reduce the number of iterations. However, if a reference point

is not given, the reference point can selected arbitrarily, such as the center of localization coordinate. In this case, through the iterative approach, the assigned point is becoming closer to the object.

3.3.7 Effect of Tilting Angle

In surveillance system, a camera can have tilting angle to increase viewable area. The tilting angle ϕ_c represents the angle difference between z-axis and v-axis on the virtual viewable plane. The tilting angle has the range as $-\pi/2 \leq \phi_c < \pi/2$.

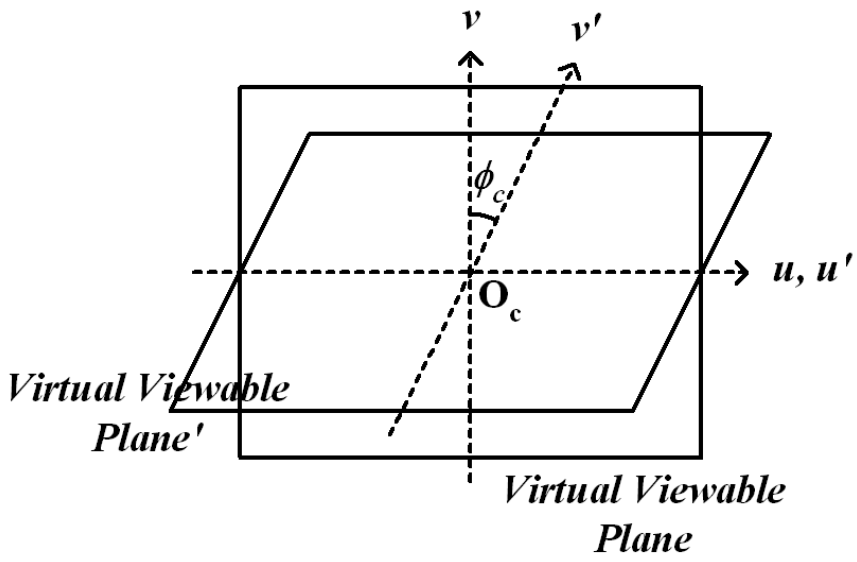


Figure 3-17: Illustration of an example of the tilting angle.

Figure 3-17 illustrates an example of the tilting angle where one plane is placed on z-axis and the other has θ_c tilting angle. The tilting angle ϕ_c equals to the angle difference between virtual viewable plane and virtual viewable plane$'$. Since u-axis is invariant for the variation of tilting angle, u-axis on the virtual viewable plane is the same as u'-axis on the virtual viewable plane$'$.

The tilting angle is the reason of distortion in u- and v-axis as shown in Figure 3-18. $P(u_p, v_p)$ and $P'(u'_p, v'_p)$ denote the project object positions of the same object within different tilting angles. The tilting angle is not affect the variation in u-axis. However, the tilting angle changes the distance of the object and camera. Thus once the distance of object and camera is changed, the zooming factor is also changed. Therefore, the tilting angle distorts the object position in u-axis.

In Figure 3-18, the distance u_p is different from the distance u'_p even if the position of camera and object is not changed. Since u_p and u'_p on the actual camera plane are translated to u_{pp} and u'_{pp} using the zooming factor and the distance between the object and camera, the tilting angle is the reason for the localization error.

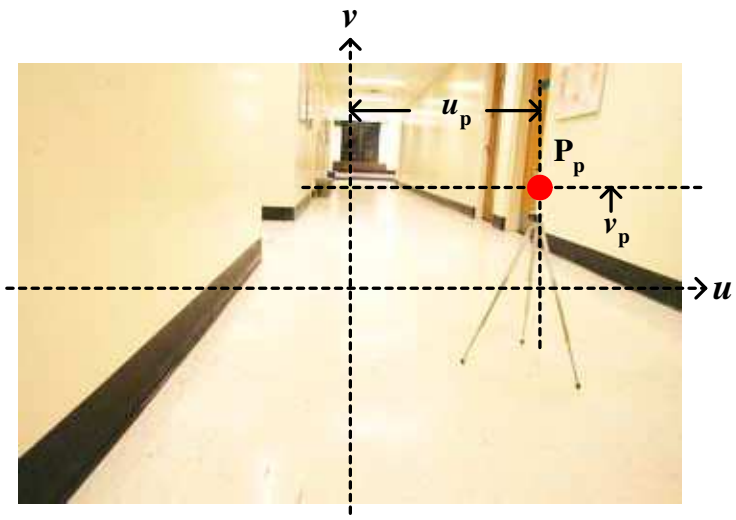
Figure 3-19 illustrates the effect of tilting angle in terms of the distance between the object and the virtual viewable plane. The height h_p and h_c denote the object height and the camera height. If the camera has ϕ_c tilting angle, the distance d_p is changed by the distance d'_p.

In order to compensate the localization error from tilting angle, we update the distance d_p to d'_p and then change the zooming factor for the distance d'_p. Thus the length l'_p in Equation 3.9 is updated as the following:

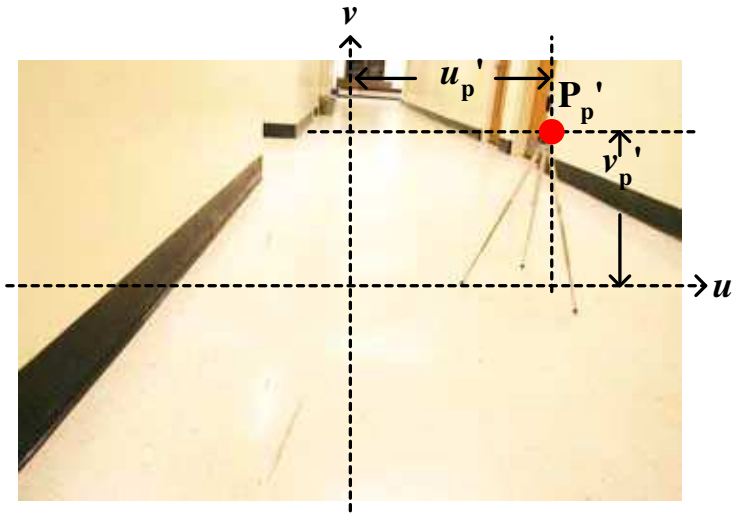
$$l'_p = l_{pp}\left(\frac{d'_p}{d_e}\right)\left(\frac{z}{z'}\right) - l_{pe}, \tag{3.12}$$

where z' denotes the zooming factor when the distance between the object and the virtual viewable plane is d'_p.

The distance d'_p is derived as the following:



(a) $\phi_c = 0$



(b) $\phi_c = 10°$

Figure 3-18: Illustration of the distortion by the tilting angle (ϕ_c).

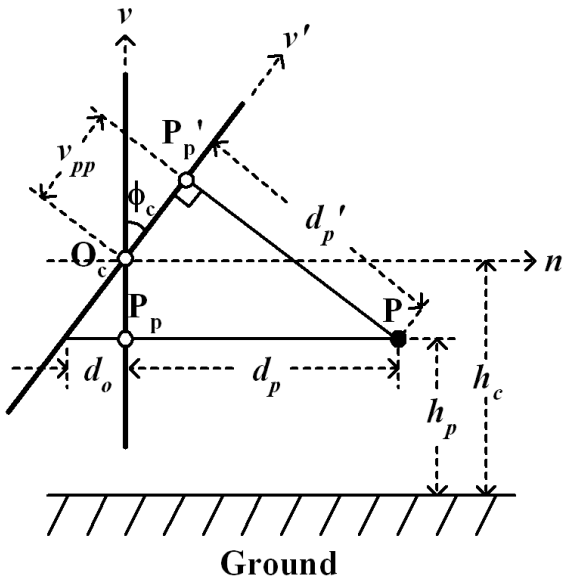


Figure 3-19: Illustration of the effect of tilting angle.

$$d'_p = (d_p + d_o)\ \cos\phi_c, \tag{3.13}$$

where the distance d_o is computed as:

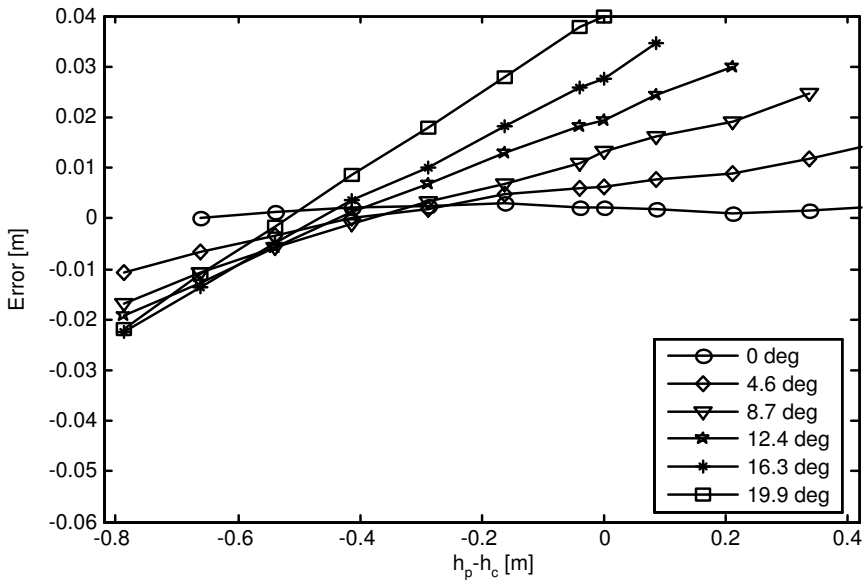
$$d_o = (h_c - h_p)\ \tan^{-1}\phi_c. \tag{3.14}$$

To quantify the localization error caused by tilting angle, we tested the localization error in the simple case. Figure 3-23 shows the setup of experiment where two cameras are placed on the left side for camera 1 and the bottom side for camera 2 in Cartesian coordinate. For simplicity, the panning factor θ_{p1} and θ_{p2} are both zero. We denote the object and the object are placed on P(1.8m, 1.8m) and E(1.5m, 1.5m).

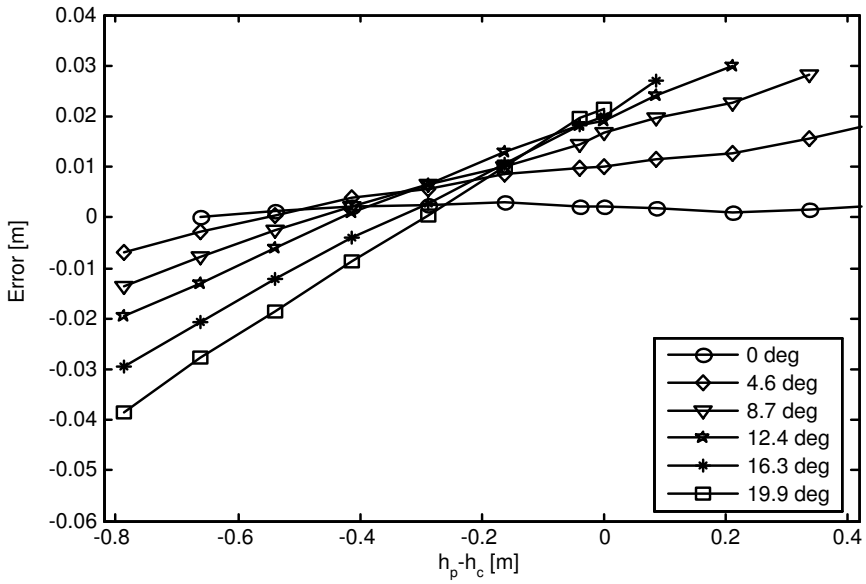
Figure 3-20(a) illustrates the localization error in terms of tilting angle variation. If the tilting angle is zero, the height difference between the camera and the object $(h_p - h_c)$ does not affect the localization result while the higher tilting angle makes the higher localization error. Thus the tilting angle ϕ_c is the reason for localization error. For example, if the height of the object is 0.2m lower than the camera height, the range of localization error is from 0.003 to 0.025m.

Once object height is provided, the localization error is compensated by Equation 3.12. In Figure 3-20(b), we compensated the localization error by denoting the camera height as 1.8m and the object height as 1.6m. The overall error caused by the tilting angle has the error range from 0.003 to 0.011m. If we know the camera height and object height, the error is compensated. Moreover, once the height difference between the object and the camera is unknown, the localization error in high tilting angle, the localization error is obviously improved. Therefore, if we expect the height of the object, the localization error can be successfully compensated.

When the height difference between the object and camera is an unknown value, the compensation for localization caused by tilting angle is difficult. However, if the distance d_p is much longer than the distance d_o, the tilting angle has little effect for the localization error. Figure 3-21 illustrates the localization error in terms of the distance d_p where the tilting angle is 12.4 degree. When the distance d_p increases, the localization error increases but after d_p is 2.7m, the error is saturated. In the worst case, the error rate is 0.01m error per 0.2m height distance. For example, once the camera height difference is 6m, the expected error is about 0.3m. Moreover, when the camera height is 0.2m taller than the object, the error range is from 0.023 to



(a) without compensation for tilting angle error



(b) with compensation for tilting angle error ($h_p - h_c = -0.2m$)

Figure 3-20: Illustration of the localization error in terms of tilting angle variation.

0.04m. Once we assume the object is placed on 0.2m lower than the camera, the compensation reduces the error to the range of 0.006 to 0.024m.

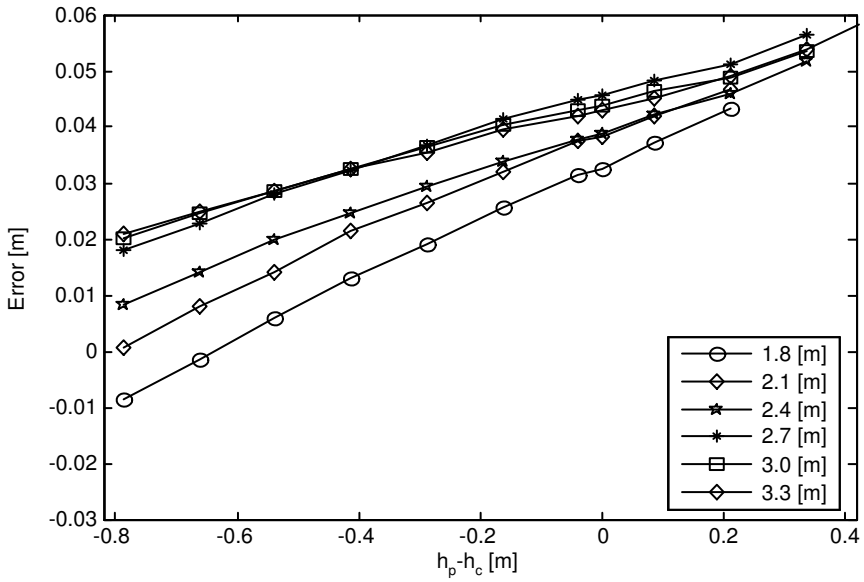
3.4 Analysis and Simulation

3.4.1 Simulation Setup: Basic Illustration

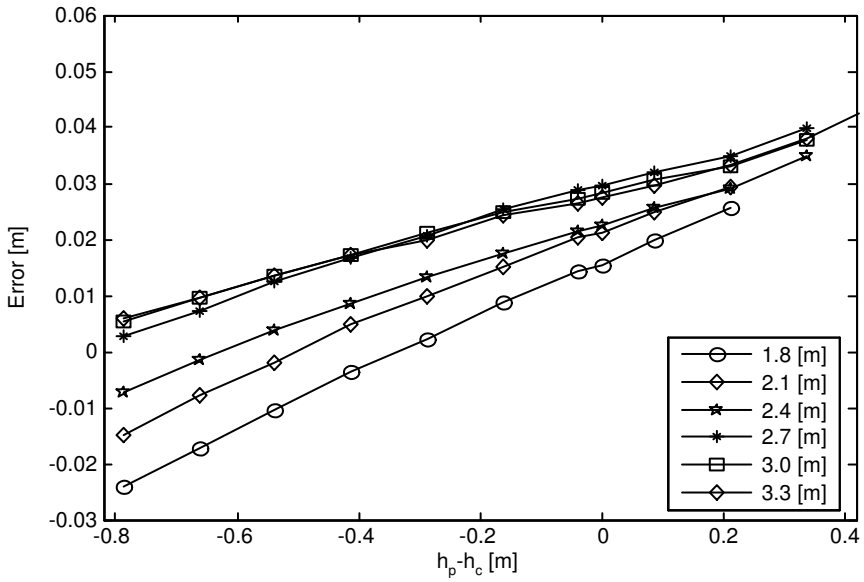
The objective in this simulation ensures the proposed localization algorithm by measuring the localization error in the real case. To show the compensation for camera non-linearity, we chose small space which is closed to the camera. In the case of Figure 3-13, the distortion from camera non-linearity exists in 2.0m inside space. Thus in this simulation, we use 4m x 4m area.

Our target application is a surveillance system where most of target objects are human or vehicle. However, in this simulation, we use a small ball as a target object to simplify the target detection. There are many reasons for localization error caused by detection. For example, the centroid detection of a human is important for reducing localization error since a human is represented as a point. If we use different positions between two camera images, the localization result has some centroid error. Thus in this setup, we use a small ball. Moreover, after taking pictures, we manually search the center of ball. We analyze the localization error in 2-D global coordinate. The object is represented as $P(x_p, y_p)$.

Figure 3-22 shows the displayed images in two cameras where the length l_{p1} and l_{p2} are distances between a reference point E and a real object point P in camera 1



(a) without compensation for tilting angle error



(b) with compensation for tilting angle error ($h_p - h_c = -0.2m$)

Figure 3-21: Illustration of the localization error in terms of the distance d_p ($\phi_c = 12.4$deg).

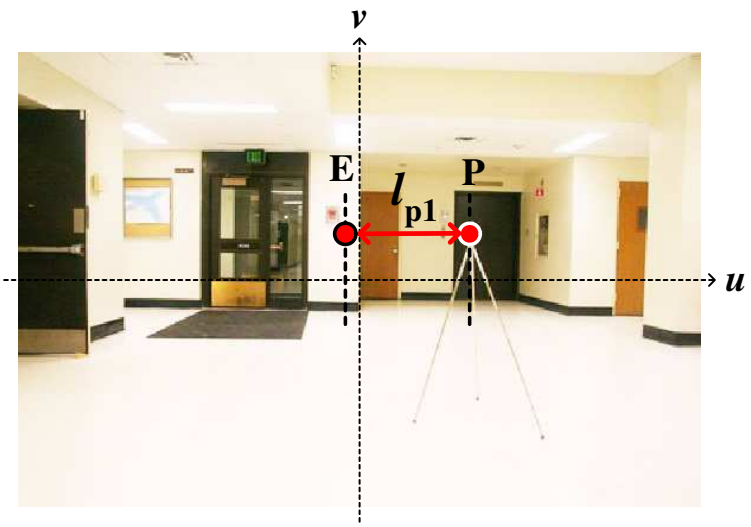
and camera 2, respectively. To explain the test setup, we showed the reference point E in Figure 3-22(a) and (b), but actually the reference point is a virtual point.

Figure 3-23 shows the experiment setup to measure an actual object. In this experiment, the actual position of the object is calculated from the reference based on the parallel projection model. In Figure 3-23, two cameras are placed on the left side for camera 1 and the bottom side for camera 2 in Cartesian coordinate. Each camera panning factor θ_{p1} and θ_{p2} are both zero.

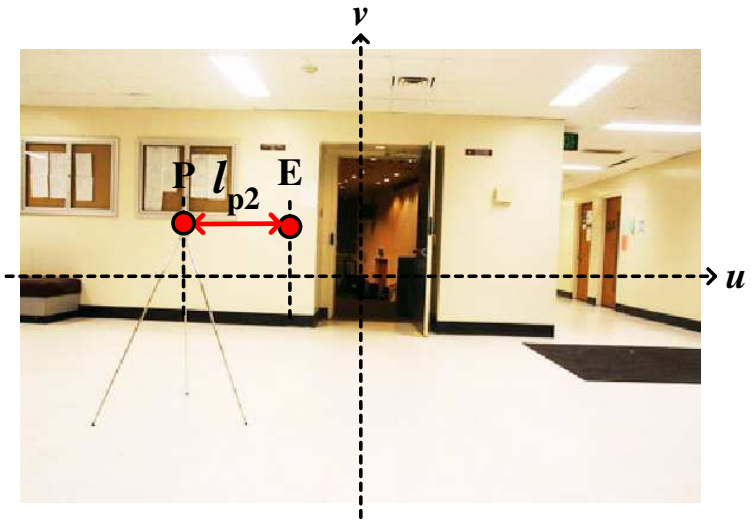
The actual zooming factors are $z_1 = d_{e1}/L_{c1}$ and $z_2 = d_{e2}/L_{c2}$ where z_1' is the zooming factor when the distance between the object plane and the virtual viewable plane is d_{p1} and z_2' is the zooming factor when the distance between the object plane and the virtual viewable plane is d_{p2}. Now, we analyze the localization result and compare the localization error depending on the iteration process called compensation.

3.4.2 Localization Error and Object Tracking Performance

Figure 3-24 shows the error distribution of the algorithm where two cameras are positioned at $O_{c1}(1.8, 0)$ and $O_{c2}(0, 1.8)$. The actual object is located at $P(1.5, 1.5)$. The figures illustrate the amount of localization error as a function of the reference coordinate. Since each camera has limited viewable angles, the reference coordinate located on the outside of viewable angle cannot be considered. Note that the error is minimized when the reference points are close to the actual object point. The localization error can be further reduced with multiple iterations.



(a) camera 1



(b) Camera 2

Figure 3-22: Illustration of two images of camera 1 and camera 2.

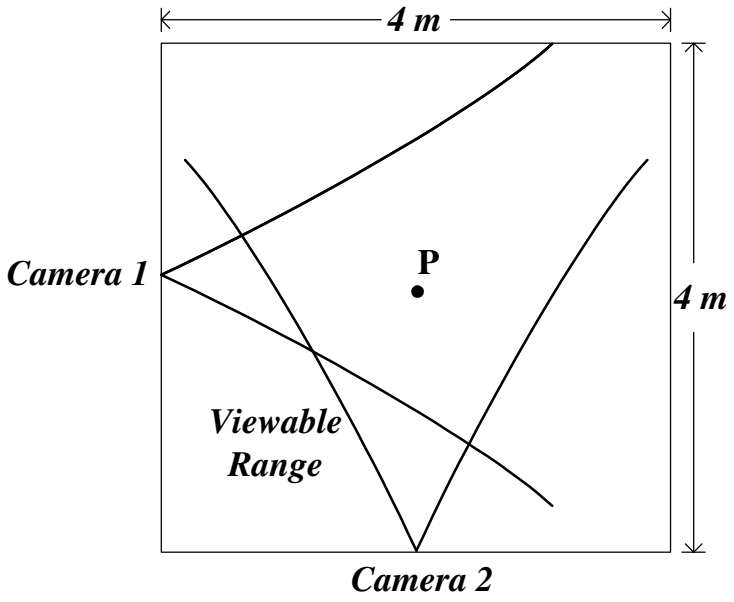


Figure 3-23: Illustration of experimental setup for localizing an actual object.

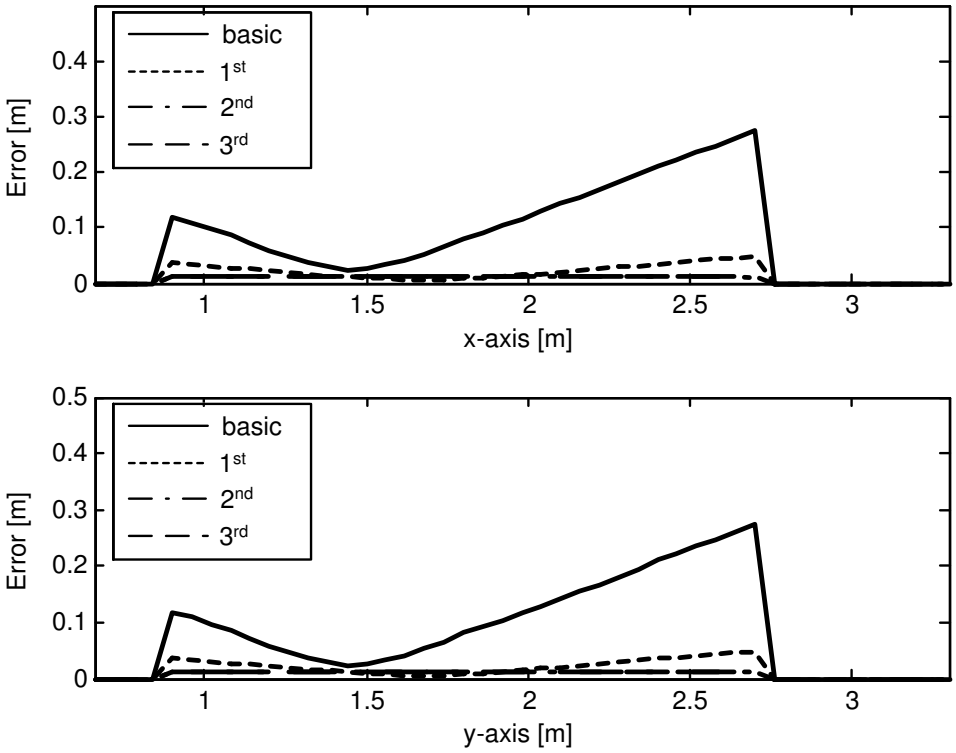


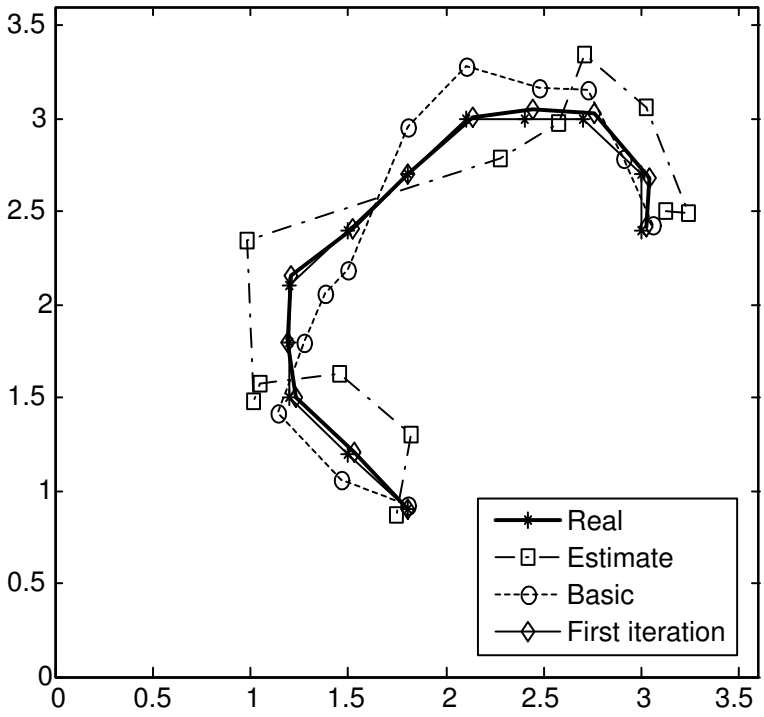
Figure 3-24: Illustration of error comparison based on the number of iterations.

The proposed localization algorithm is also used for a tracking example. In this example, an object moves within a $4m \times 4m$ area and the images are obtained from the real cameras. We first applied the proposed non-iterative localization algorithm with compensation in tracking problems. Each time the object changes coordinates, its corresponding estimation is generated. Figure 3-25(a) illustrates the trajectory result of localization. After the compensation, the tracking performance is improved. Figure 3-25(b) and Figure 3-25(c) illustrate the tracking performance in the x-axis and the y-axis. These figures clearly show that the compensation improves the tracking performance.

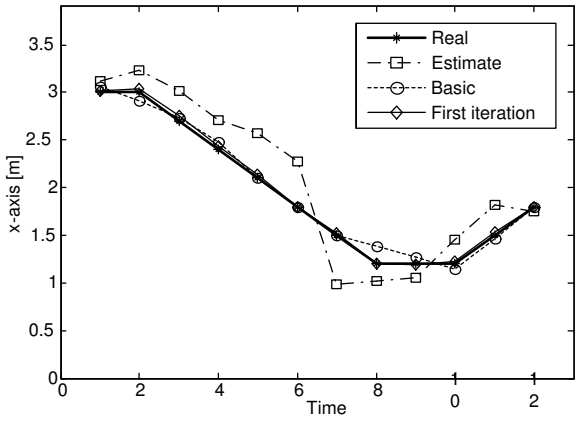
Similarly, the proposed iterative localization algorithm is used in the same tracking example. In this case, only one reference coordinate is used for the entire localization. The chosen estimate is outside of the trajectory as shown in Figure 3-26. This figure illustrates the trajectory result of localization. There is a significant error with the one iteration since the estimated coordinate is not close to the object. Note that the error increases if the object is further away from the estimated coordinate. However, successive iterations eliminated the localization error as shown in the figure.

3.4.3 Application of the algorithms

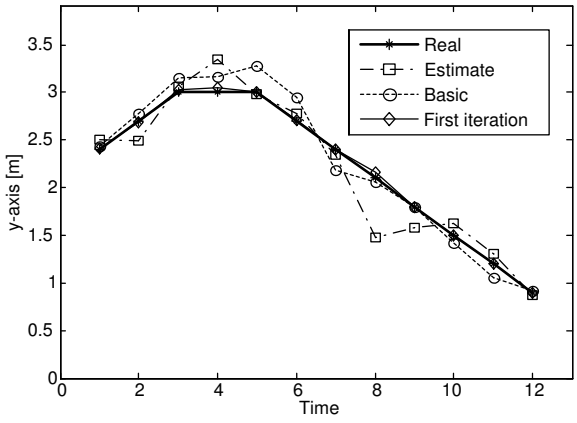
Figure 3-27 shows a tracking environment where the proposed localization algorithm is applied. For illustration, two sequences of images are shown. The coordinate of the center of the room is chosen as the initial reference coordinate. The cameras follow the object during the localization. When the object is detected by individual camera,



(a) Trajectory performance



(b) x-axis performance



(c) y-axis performance

Figure 3-25: Application of the non-iterative localization in tracking a trajectory with rough estimates.

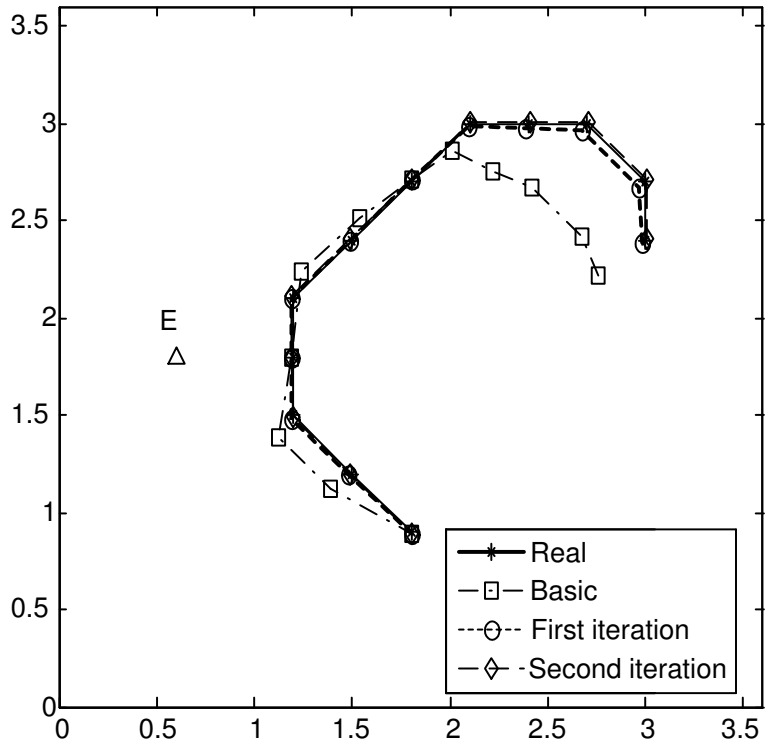
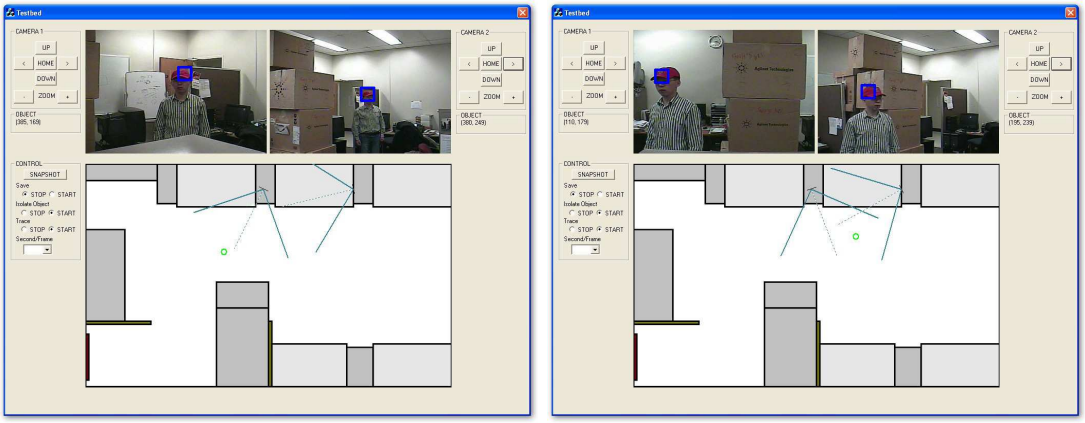


Figure 3-26: Application of the iterative localization with single estimate.

the coordinate of the camera images are combined for actual coordinate. The actual coordinate is shown in the tracking environment. In the experiment, cameras are following the object through panning.

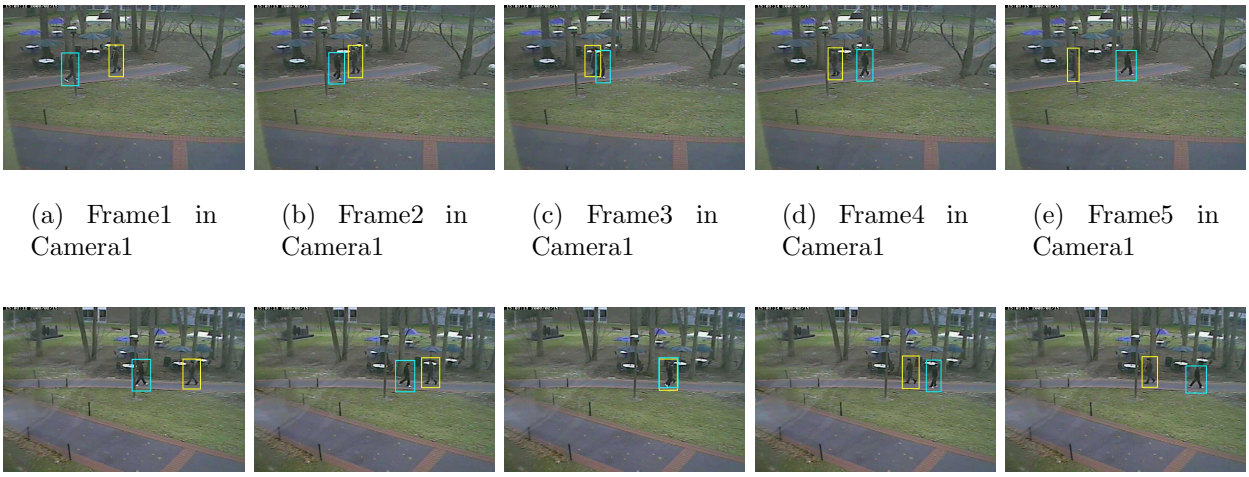
Figure 3-28 illustrates detection objects in outdoor environment where two objects are used for evaluating the proposed localization algorithm. Both cameras are placed on the same side and the panning angles in camera 1 and 2 are 3^o and 34^o, respectively. This setup is almost the worst case since two cameras are parallel and shows the objects in close angles.

Figure 3-29 illustrates two objects trajectory in an outdoor environment. Since the method is computationally simple, the total computation time is proportional to the the number of objects, which is not a significant with respect to overall computation.



(a) Snapshot 1 (b) Snapshot 2

Figure 3-27: The snapshots of the tracking environment based on the proposed localization algorithm. Human face is used to localize a person. The circle represents the actual coordinate of the person within the room.



(a) Frame1 in Camera1 (b) Frame2 in Camera1 (c) Frame3 in Camera1 (d) Frame4 in Camera1 (e) Frame5 in Camera1

(f) Frame1 in Camera2 (g) Frame2 in Camera2 (h) Frame3 in Camera2 (i) Frame4 in Camera2 (j) Frame5 in Camera2

Figure 3-28: Illustration of detection results for people localization in an outdoor environment.

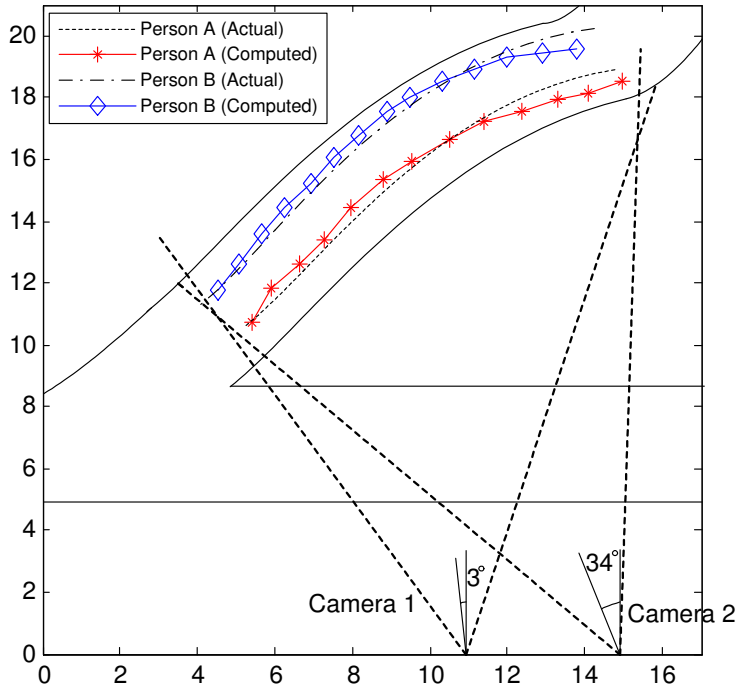


Figure 3-29: Illustration of two objects trajectory in an outdoor environment.

Chapter 4

Human Body and Face Detection Approach for Tracking and Localization with Multiple Cameras

4.1 Introduction

In this chapter, we propose a robust human body and face joint detection method with multiple cameras, which enables to collaborate one another in order to alleviate the single camera limitation such as occlusion, body overlapping without face detection, and face recognition inability due to low resolution. The original detection called by primary detection searches body and face information, and draw a rectan-

gular for each in a view of camera. Through the primary detection in a multiple camera environment, it is possible to localize each person in a global coordinate [19]. The global localization and tracking enable us to monitor all-around human movement without disturbances such as overlapping, occlusion and limited viewable range. Thus, the global information reversely assists the primary detection suffering from the above single camera limitations. The assistant detection through global localization and tracking is called as a secondary detection. That is, the secondary detection is performed when the primary detection from one of cameras is failed. Furthermore, we may easily support the camera panning and zooming factors through the global information; thus, the detection algorithm supports the dynamic camera change.

The rest of this chapter is organized as follows. Section 4.2 introduces application system model and briefly describes problem description in multiple-camera environment. Section 4.3 illustrates primary and secondary human and body joint detection. In Section 4.4, we present analysis and simulation results.

4.2 Problem Description

4.2.1 Application System Model

Global Localization

Figure 4-1 illustrates the application system model. The one of target application goals is to obtain each detected human position for monitoring in a global coordinate. Each global position can be simply concerned into a 2-D tracking problem on a ground

plane [17] [35]. In order to localize each detected human in a global coordinate, it is necessary to have more than one views obtained from different cameras [19] [68]. In Figure 4-1, person A and B are in an observable area from two different cameras while person C is in the area from one camera. In the case of person A and B, the camera 1 and 2 are sufficiently achieving global localization. On the other hand, person C needs one of two cameras 1 or 2 to control panning toward the person C for the global localization. Similarly, the cameras may need to control zooming as well as panning through the global localization. Hence, the application system model provides global position for each detected person as well as support zooming and panning features.

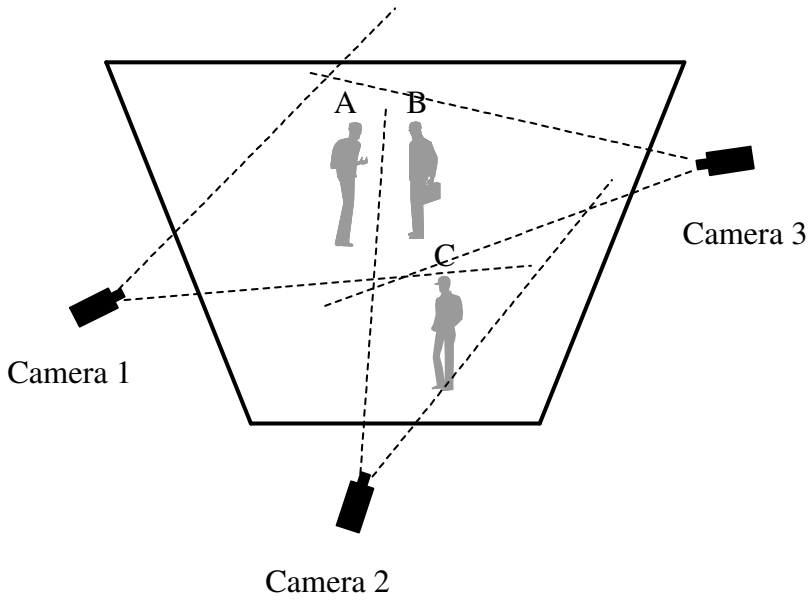


Figure 4-1: Illustration of the application system model.

Global and Local Position Interaction

Figure 4-2 illustrates the overall processing model for human body and face joint detection in a multiple camera environment. In addition to the primary and secondary

detection, two sorts of tracking approaches are incorporated in the system: local and global tracking. The local tracking is corresponding to the detected human trajectory through each camera monitor while the global tracking is corresponding to the trajectory in a global coordinate supporting a real position. The global detection and tracking reversely provide valuable information to local detection and tracking, especially when the local performance is degraded or failed. For example, several overlapped persons are viewed as single body by one of multiple cameras. For the further processing, the overlapped body needs to be splitted. Note that we jointly use a face information in order to reduce the overlapping cases as much as possible. Hence, we consider the overlapping problem when both a body and a face are overlapped altogether. The global detection and tracking provide the secondary detection and can distinguish different persons when even a body and a face are both overlapped. That is, through the interactions between global positions and local positions, the surveillance performance is enhanced.

4.2.2 Multiple-Camera Detection

So far, we introduced the overall application system model with multiple cameras. In addition, we briefly addressed the single camera limitations, and discussed the motivation of multiple cameras. In this part, several examples of single camera limitation are illustrated in detail, and the advantages of multiple camera detection are presented.

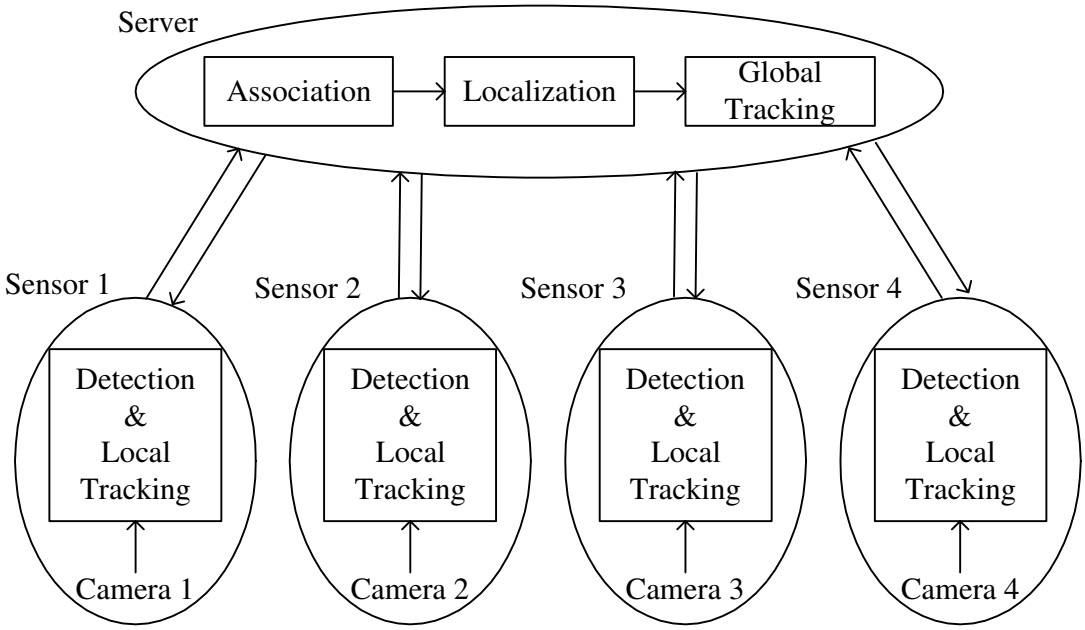
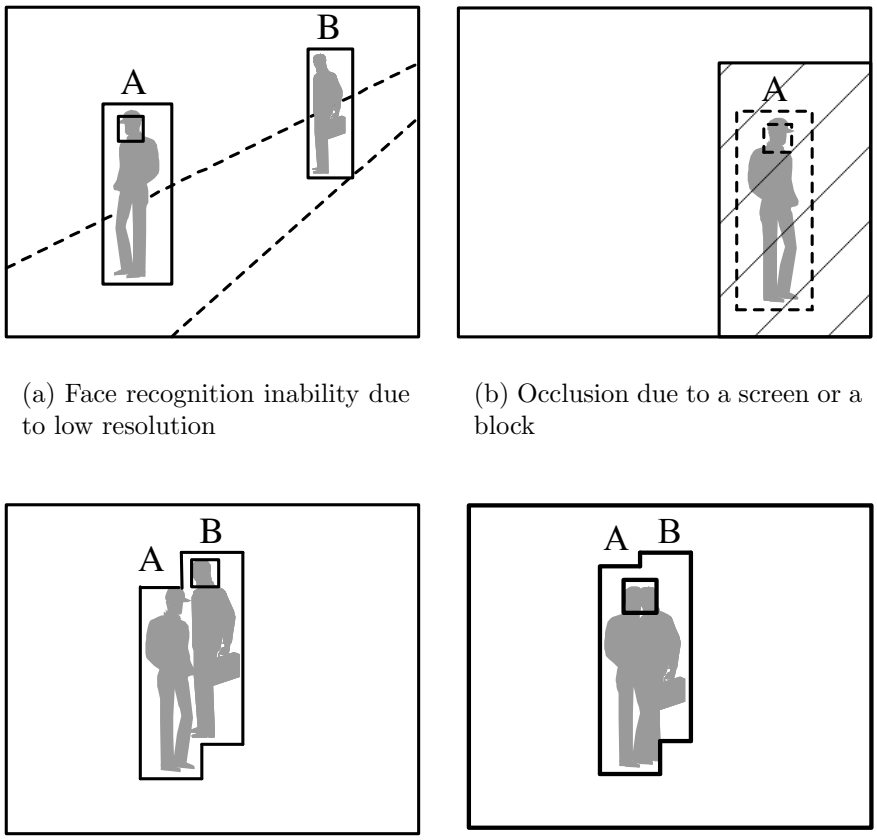


Figure 4-2: Illustration of the overall processing model for human body and face joint detection in a multiple camera environment.

Limitation of Single Camera Detection

Figure 4-3 illustrates examples of single camera detection limitation: face recognition inability due to low resolution, occlusion, body overlapping without face detection and face overlapping. In Figure 4-3(a), a person B face is not detected due to the low resolution of the face image. Figure 4-3(b) illustrates that a person A is moving behind a screen or a block. As long as the person A is behind the screen, it is impossible for the camera to detect the person A. Figure 4-3(c) and 4-3(d) illustrates overlapping examples. In both cases, it is recognized as only one person based on single body and face detection. The body overlapping without face detection is resulted from person A face detection failure while the face overlapping is resulted from more severe overlapping close to a complete unity.



(a) Face recognition inability due to low resolution

(b) Occlusion due to a screen or a block

(c) Body overlapping w/o face detection

(d) Face overlapping

Figure 4-3: Examples of single camera detection limitations.

Advantage of Multiple-Camera Detection

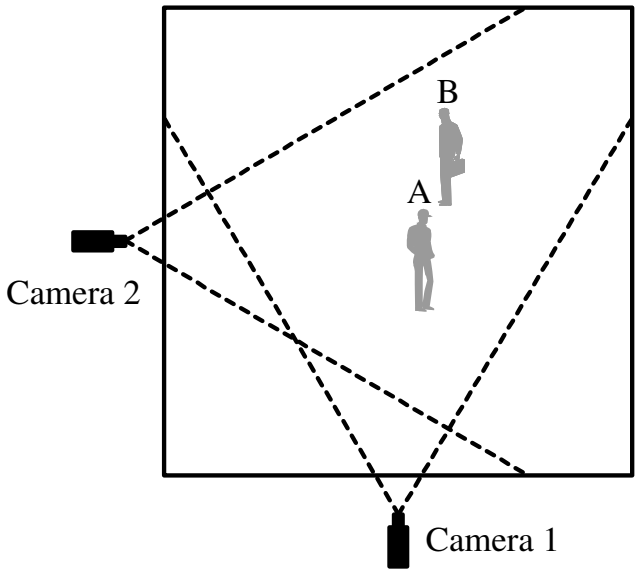
Figure 4-4 illustrates one of examples which multiple cameras have an advantage by alleviating the single camera limitations. As shown in Figure 4-4 (a), two persons A and B are moving around in an observable area. Although two cameras 1 and 2 are capturing images and detecting human bodies and faces at the same time, the detection results are different due to the different camera perspective as shown in Figure 4-4 (b). The two persons are overlapped and viewed as one human through a camera 1 while the two persons are distinct with separate detections through Camera 2. Since Camera 2 holds the complete detection information, the overlapping problem in the camera 1 can be alleviated through the secondary detection supported by a global information. In this chapter, our contribution is that how the secondary detection is handled in order to alleviate the single camera limitations.

4.3 Body and Face Joint Detection Algorithm

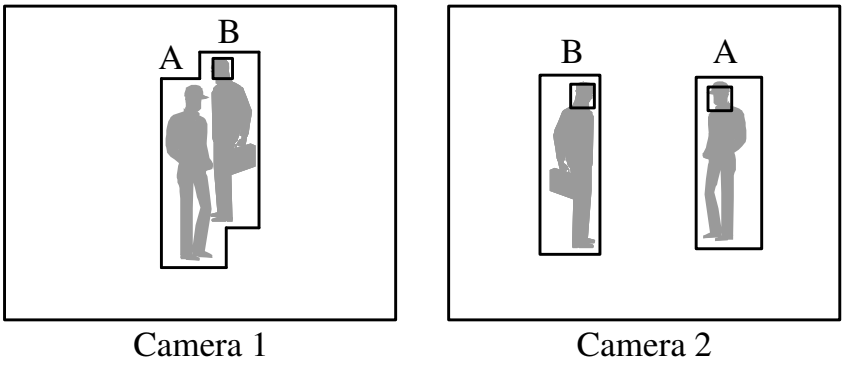
4.3.1 Single-Camera Body and Face Joint Detection

Primary Detection with Single Camera

Human detection starts from motion detection which provides body information of human. Conventional approaches for human body detection include temporal difference, background subtraction, and optical flow [14] [69]. One of the most successful approaches is the background subtraction which detects moving regions in an image by taking the difference between a current image and a reference background image [14].



(a) Two persons A and B are moving around in an observable, and two cameras detecting them.



(b) Each camera has different detection result due to the different camera perspective.

Figure 4-4: Illustration of an example which multiple cameras have an advantage by alleviating the single camera limitations.

The captured image from a camera is simply subtracted with the reference image. If the color difference in a pixel is greater than a noise-based threshold, the each pixel is represented as a moving object and becomes a human body candidate. [3]. The main advantage of the background substraction for a human detection is that it is much less affected by the geometry or photometry of a scene [3].

Since a camera supports panning and zooming in our application, the background substraction requires automatic recovering and updating in a dynamic scene. We present a procedure for the background update which supports panning and zooming operation. Figure 4-5 shows the process for gathering motion information from a current image and a background image. Once the camera is panning or zooming, the human in the background is also recognized as a background and updated as a part of the background image as shown in Figure 4-6(a). Then, a ghost motion image shown in 4-6(c) exists. Since the ghost image does not have any motion information, it should be eliminated through the background update as shown in 4-6(d). The background update is replacing the part of ghost image to a new background after a few frames. Thus, the background update requires a few frames for recovery. Note that the global information provided by tracking and localization using multiple cameras can improve the procedure of background update. We will discuss the approach in the part of a multiple camera detection.

Together with the body detection, the face detection identifies a human and other moving objects. In order to isolate a human face, the color based approach has been actively discussed in [70] [71] [72] and [73]. The color based approach has a sufficient capability for the face detection since the color information is one of main

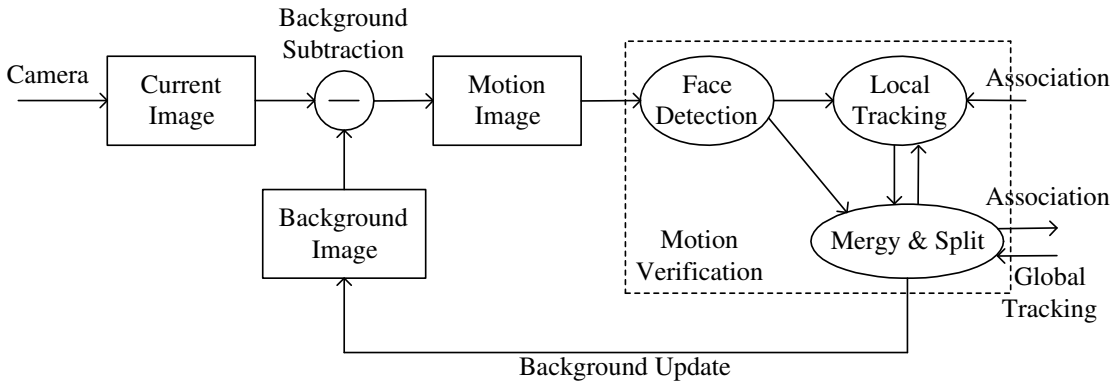
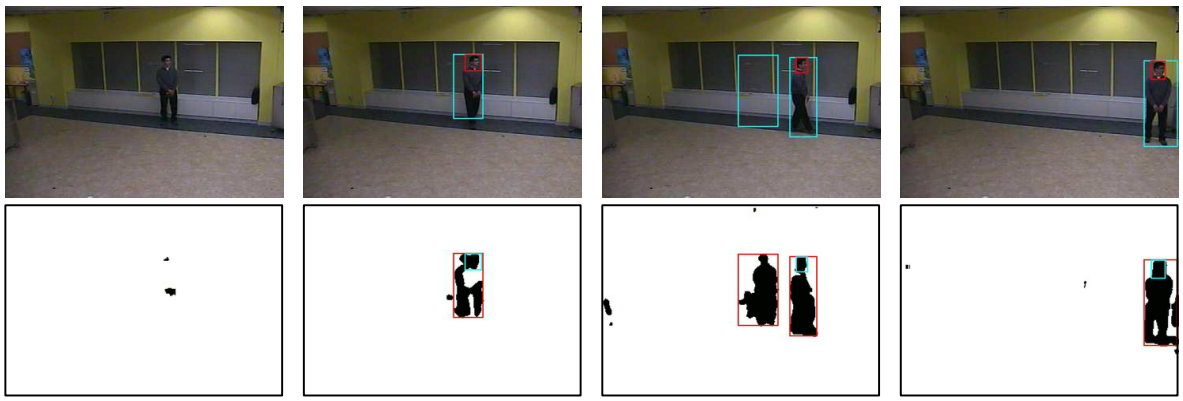


Figure 4-5: Illustration of the processing for gathering motion information from a current image and an background image.



(a) New background after panning

(b) Starting split

(c) Splitted motions

(d) Background update

Figure 4-6: Illustration of the recovering and updating an background image after panning.

face features. Moreover, the approach is simple and fast [70] [72]. On the other hand, in the color based face detection, the problem is to detect false ones whose color information is a similar skin [71]. Thus, the color based face detection requires one additional step which is filtering out false detection for the verification such as naive bayes classifier [70] [71] [72] [73]. Figure 4-7 shows the body and face detection in an overlapped example. Even though the body is overlapped, the non-overlapped face can distinguish the different persons, and split the body images.

(a) Motion image (b) Detected image

Figure 4-7: Illustration of the body and face joint detection in an overlapped case.

Local Tracking in Single Camera

Besides the detection issue, a local tracking is also one of the critical parts for a robust surveillance system. The local tracking has an advantage of a detection recovery by predicting the next possible position. Figure 4-8 (a) shows that a person is detected by the body and face joint detection. Once he is turning back as shown in 4-8 (b), a face detection may fail due to the rotation which eliminates face characteristics.

However, the local tracking predicts the next head and body position, and recovers the missing face detection.

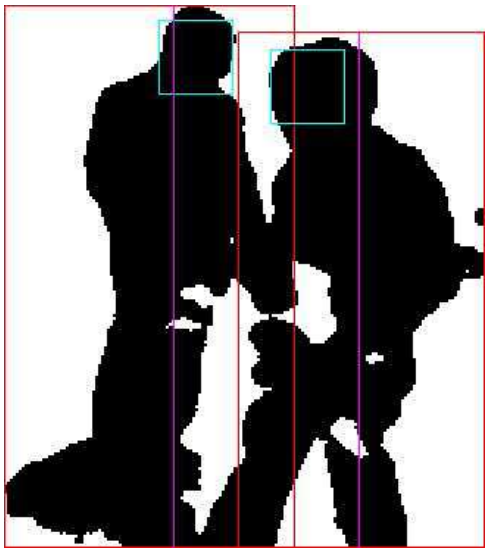
(a) Body and face joint detection (b) Local tracking

Figure 4-8: Illustration of the local tracking in a single camera.

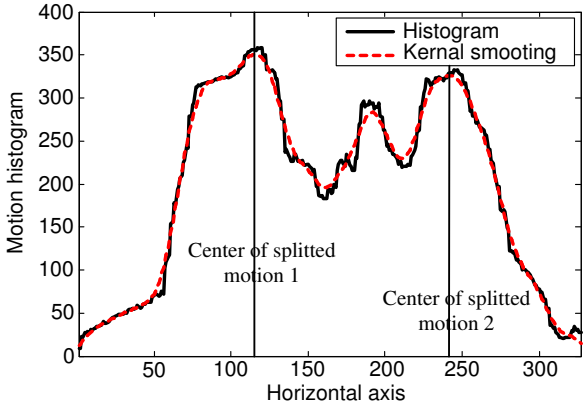
The local tracking supports the background updating as well. Once a person stops the movement, the image of person is updated as a background. However, the local tracking is capable of managing the trajectory history. Thus, once a person is detected in the new background, the view of the person is not updated as background image.

Furthermore, the local tracking algorithm supports motion splitting. In [2], the vertical projection histogram of the silhouettes is used to split a group of people. In the body and face joint detection, if a motion has the same number of faces, the locations of faces are used for motion splitting. However, if the motion has less number of faces, the silhouette-based splitting is used. Figure 4-9 illustrates the motion splitting when two persons are overlapped.

Note that the local tracking uses the position and velocity of a person which are



(a) Splitted motion image



(b) Histogram of motion silhouette

Figure 4-9: Illustration of the motion splitting where the motion is splitted into two persons.

estimated from previous detection. However, since a camera has a narrow view of camera, the performance of local tracking is also limited. Therefore, we use secondary detection provided by a multiple camera environment.

4.3.2 Multiple Camera Body and Face Joint Detection

Case of a Multiple Camera Detection

Figure 4-10 illustrates the complexity of real human movements in a multiple camera environment. In this scenario, a secondary detection is necessary due to each single camera limitations. For example, the person A is moving away from a camera 1 whose face detection may fail. In the case of person B and C, they are close to one another which may suffer from an overlapping problem with respect to a camera 1. The person D is moving behind a screen with respect to a camera 1. The person E and F are moving in and out of each camera observable range.

Figure 4-11 illustrates the secondary detection in cameras 1 and 2. After the primary detection based on the body and face joint detection, the view of camera 1 detects three separate motion information denoted as M_1, M_2, and M_3. The detected motion M_1 and M_2 accompany each face while the other detected motion M_3 does not. Since a global localization sustains each human position, the global position supports each local camera detection by transferring the points to a vertical line in local detection and tracking. The number of vertical line is the same as the number of people in a view of each camera. As shown in Figure 4-11 (b), the transferred lines G_1, G_2, G_3, and G_4 inform the local cameras to support the secondary detection:

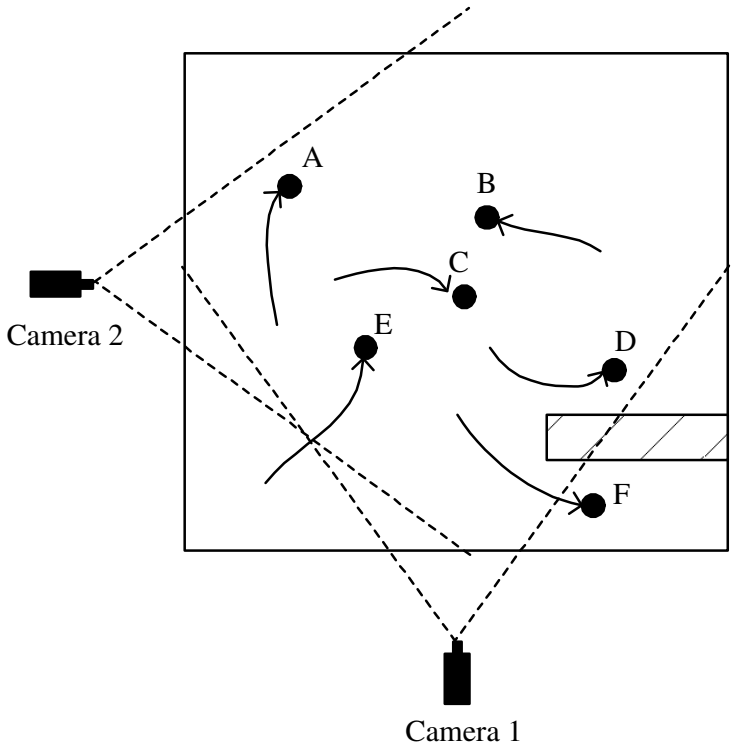


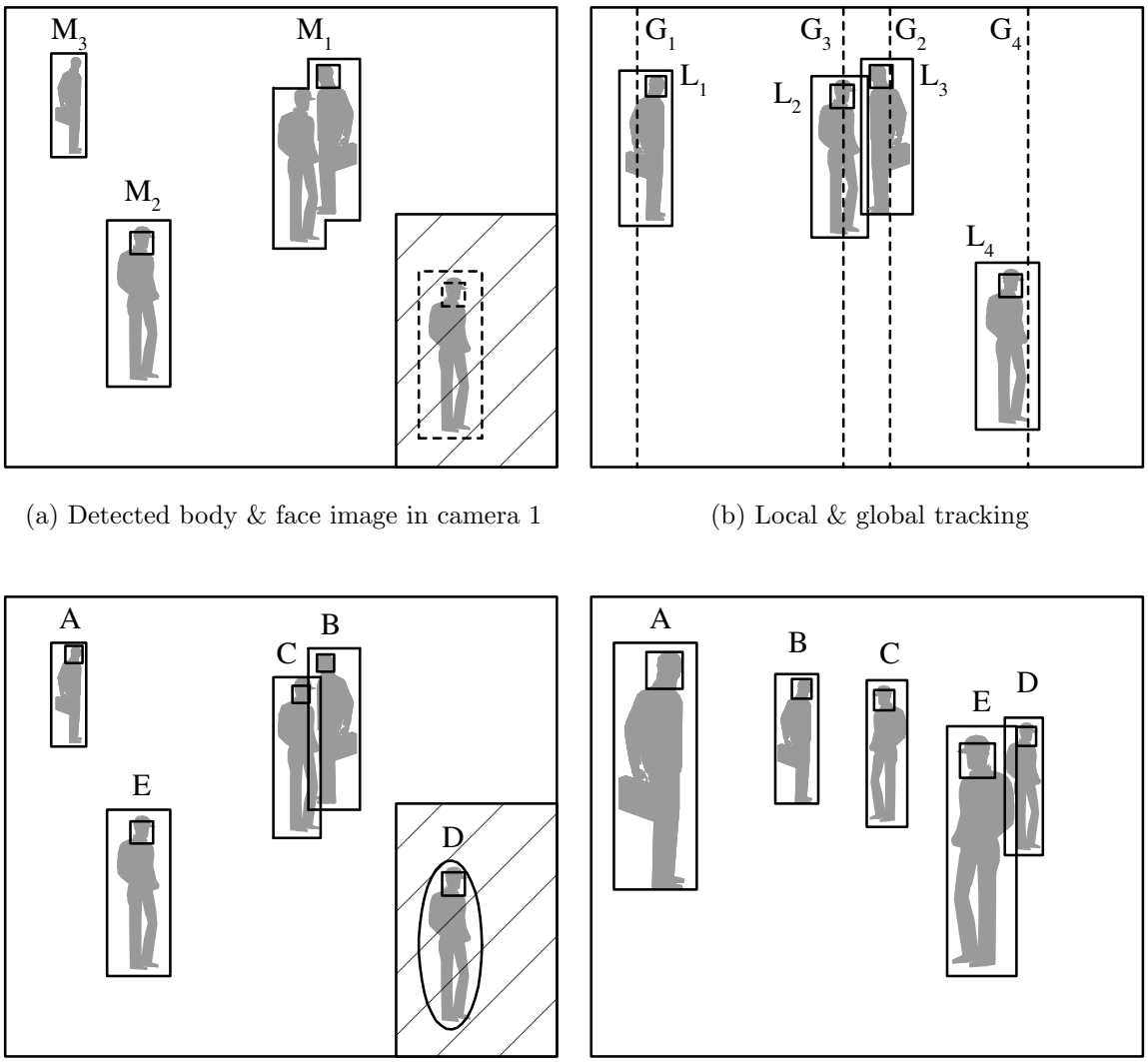
Figure 4-10: An example: complexity in real human movements in a multiple camera environment.

the person A recovers a face detection, the person B and C are splitted, and person D is detected even behind a screen. Note the motion information M_2. Since the transferred vertical line is based on the global information in the previous frame, the M_2 recognizs a new person who is corresponding to the person E. In addition, through a global tracking, the global information is capable to predict that the person F is moving out the viewable area of both cameras. Figure 4-11(c) and (d) illustrate the view of camera 1 and 2 after the secondary detection where the occluded person is displayed by a circle. Thus, the secondary detection based on multiple cameras improves the body and face joint detection.

Transferred Vertical Lines Through Global Localization

The transferred vertical line is one of important factors which links between local and global information. Figure 4-12 illustrates the global position transferring to a view of camera where the global position is presented as a vertical line. u- and v- axis represent the camera image plane [19]. u_{ps} denotes the distance between the center and the vertical line for representing a position of person.

The unit of u_{ps} is the number of pixels. Once $O_c(x_c, y_c)$ and $P(x_p, y_p)$ denote the origin of virtual viewable plane and the object position on object plane, the distance u_{pp} represents the position of the person in virtual viewable plane and is equal to $u_{pp} = (x_p - x_c)\cos\theta_c + (y_p - y_c)\sin\theta_c$ where θ_c denotes the camera angle [19]. The actual distance between the object plane and virtual viewable plane is expressed as $d_p = \sqrt{(x_p - x_{pp})^2 + (y_p - y_{pp})^2}$ where $x_{pp} = x_c + u_{pp}\cos\theta_c$ and $y_{pp} = y_c + u_{pp}\cos\sin_c$. The camera plane length L_s is a given constant and zoom factor z is derived from



(a) Detected body & face image in camera 1

(b) Local & global tracking

(c) View of camera 1

(d) View of Camera 2

Figure 4-11: Illustration of the secondary detection in camera 1 and 2.

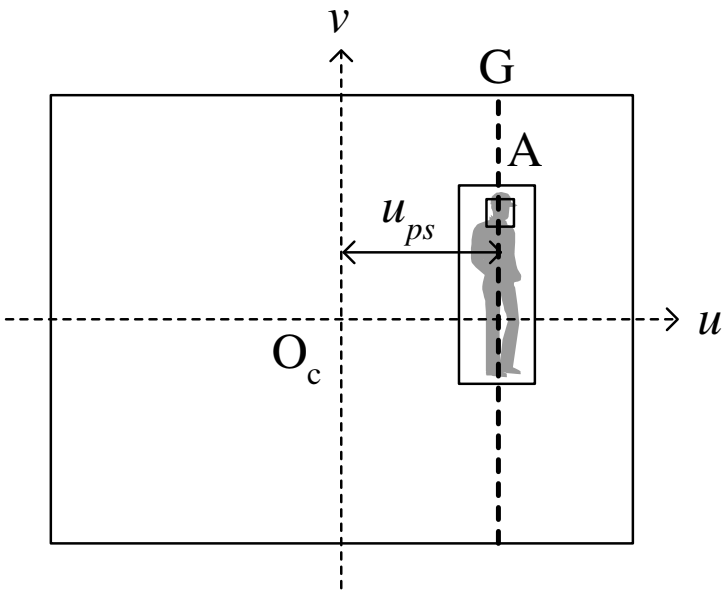


Figure 4-12: Illustration of the global position transferring to a view of camera where the global position is presented as a vertical line.

zooming table using the distance d_p. Thus the position of vertical line is represented as follows:

$$u_{ps} = u_{pp}\left(\frac{zL_s}{d_p}\right). \tag{4.1}$$

Secondary Detection Supporting Overlapping, Occlusion and Panning/Zooming

Figure 4-13 illustrates the secondary detection using global information where the global tracking is incorporated to split the overlapped persons. L_1 and L_2 represent the local positions of the person A and B provided by the local tracking in a single view of camera. Once two persons are overlapped, the local tracking cannot separate them by itself. The global tracking provides the position of person A and B which are shown by two vertical lines denoted G_1 and G_2. Therefore, the secondary detection

from global information provides more accurate location by alleviating the overlapping problem.

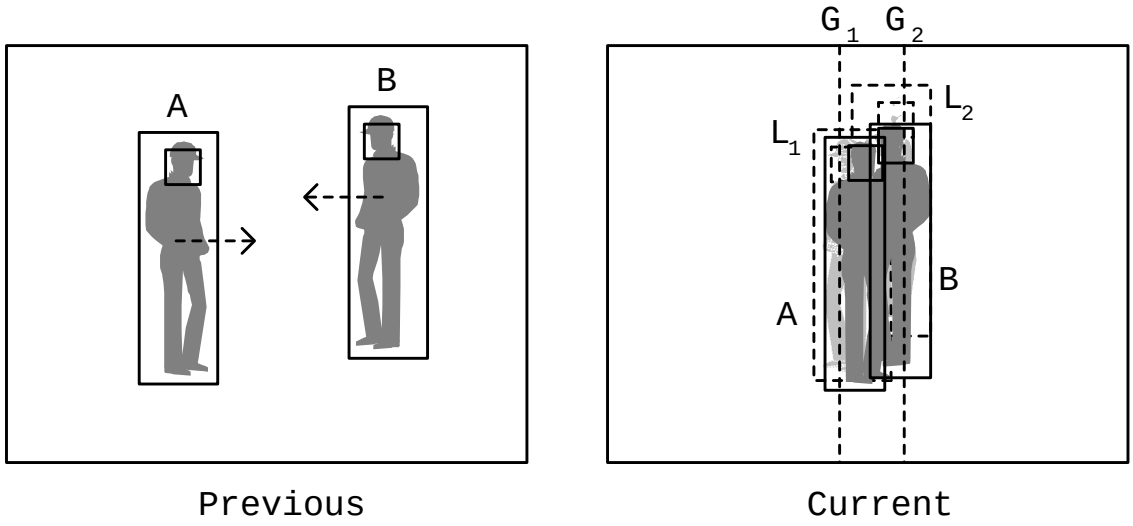


Figure 4-13: Illustration of the secondary detection using global information where global tracking is incorporated to split overlapped persons.

Figure 4-14 illustrates occlusion detection where the body and face joint detection incorporates the global information of each position. Since the person is occluded by a screen, camera 1 cannot detect the person. However, the global information indicates that the person is positioned behind the screen.

In the single camera detection, once a camera has some movement such as a panning or a zooming, the background image is initialized by the first incoming image after the camera movement. Then if the person is moving in the view of camera, the person makes two motions; the first location and a new motion of the person. Since one of motions does not have any movement, the motion image is updated as background. Thus, after the panning or the zooming, a few frames are necessary to update the background image. However, in multiple camera cases, we have a global

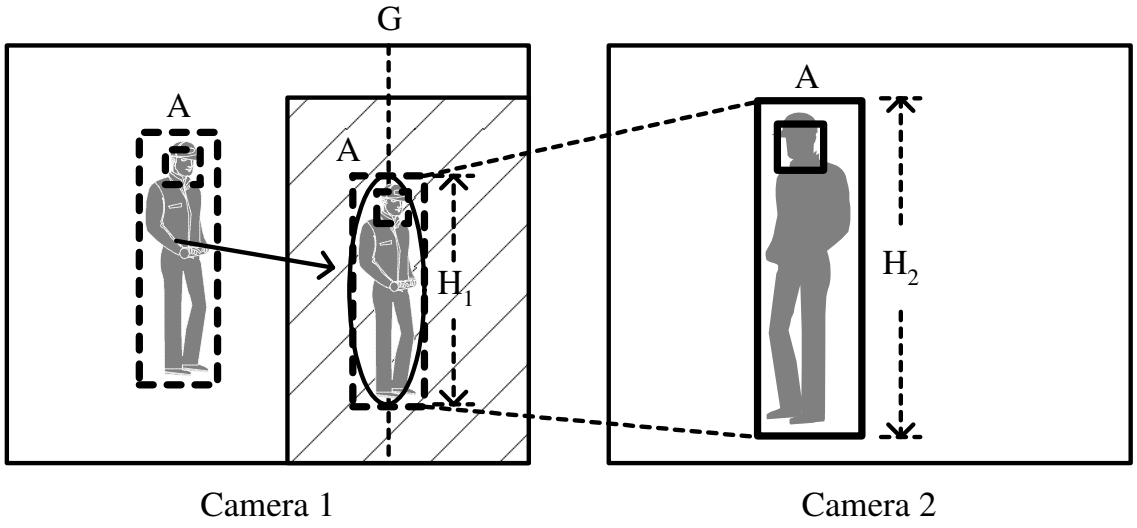
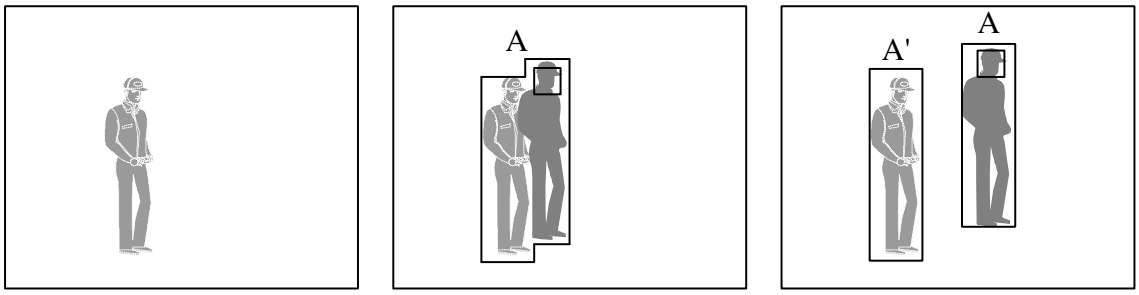


Figure 4-14: Illustration of the occlusion detection where human body and face joint detection achieves using local tracking, global tracking through cameras collaboration.

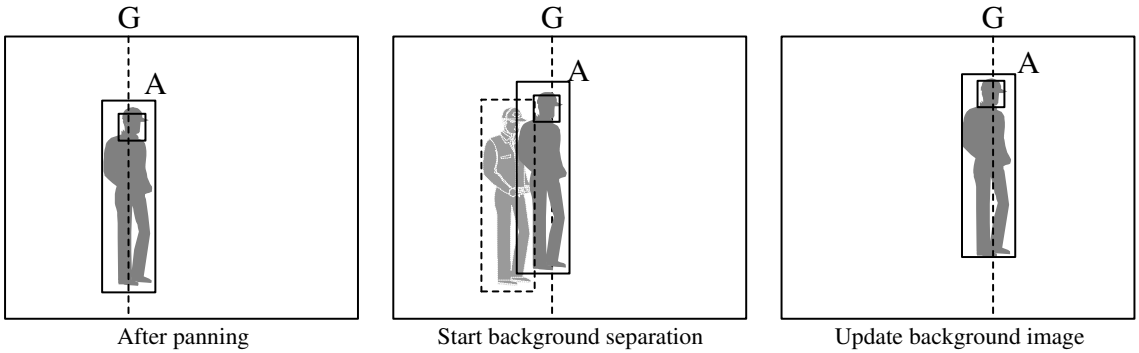
position of each person. After camera movement, the global position of the person is kept providing by the global tracking. Thus, we redraw the motion of the person using the global information. In a multiple camera environment, since the position of person is globally projected to the view of each camera, the real person is easily separated. Therefore, the global information improves the procedure of background update. Figure 4-15 illustrates the background update after panning operation.

4.4 Algorithm Verification

Figure 4-16(a) and 4-16(b) compare the primary and the secondary detections with respect to Camera 1. In the primary detection from Camera 1, two separate bodies M_1 and M_2 are detected, and each body accompanies one detected face as shown in Figure 4-16(a). In the scenario, globally localized positions obtained in a previous



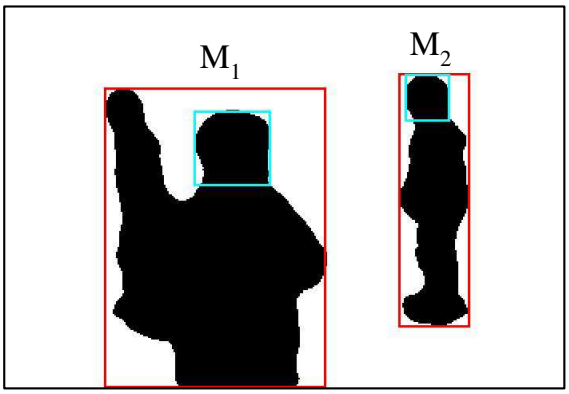
(a) Primary detection



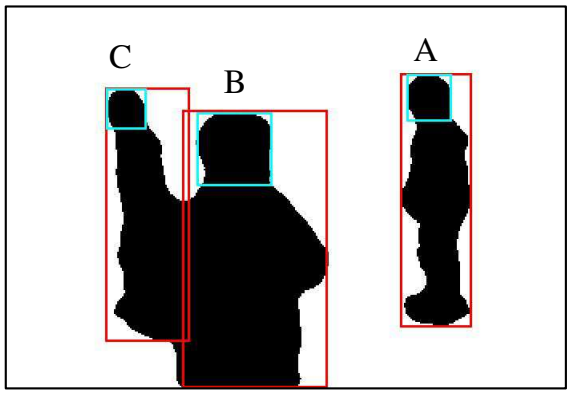
(b) Secondary detection

Figure 4-15: Illustration of the background update after panning.

frame are transferred to Camera 1, and the motion M_1 is splitted by two persons as shown in Figure 4-16(c). Similarly, Camera 2 is supported by the secondary detection as shown in Figure 4-16(d).



(a) Primary detection



(b) Secondary detection

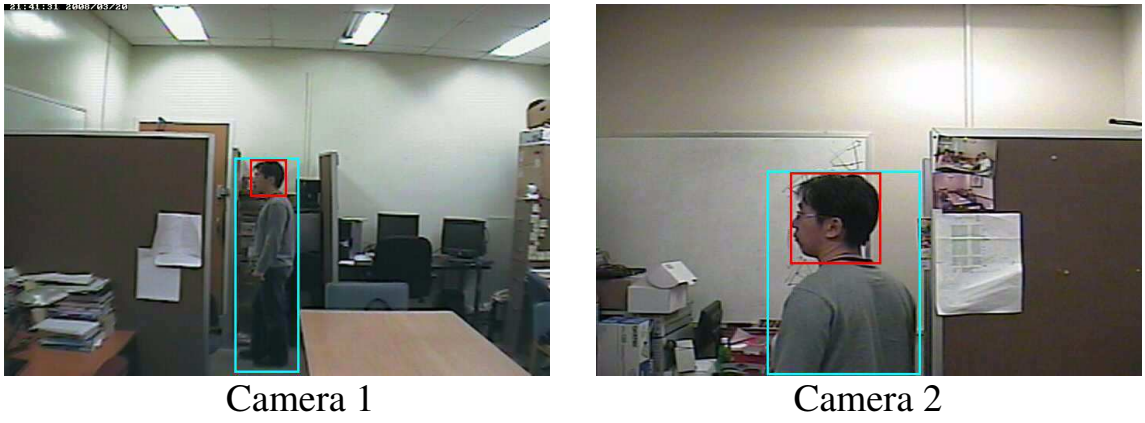
(c) View of Camera 1

(d) View of Camera 2

Figure 4-16: Illustration of the secondary detection alleviating an overlapping problem.

Figure 4-17 illustrates the occlusion detection. Since the person is already detected by both cameras 1 and 2 as shown in Figure 4-17(a), a global localization is supported for the secondary detection. Once the person is occluded by a screen, the global position is transferred to Camera 1, and the occluded person can be displayed with

a circle as shown in Figure 4-17(b).

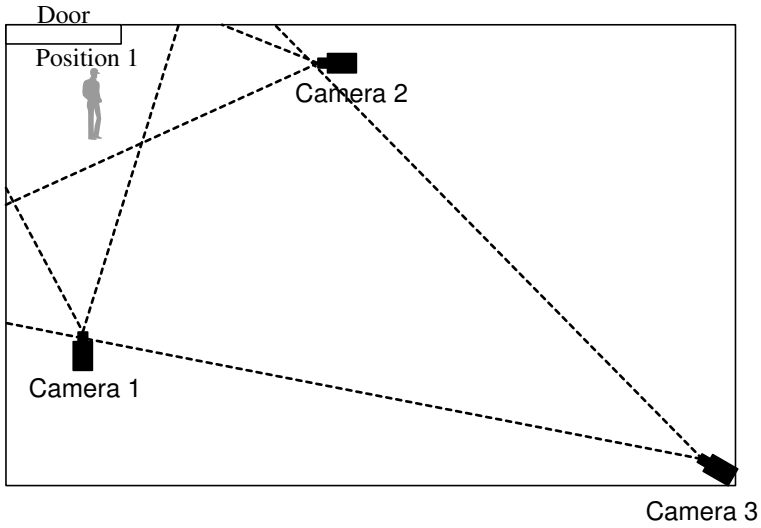


(a) Before occlusion

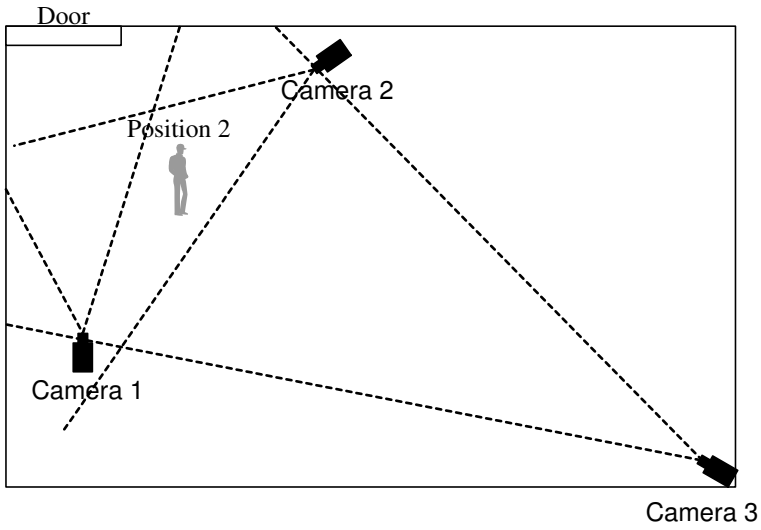
(b) After occlusion

Figure 4-17: Illustration of the secondary detection alleviating an occlusion problem.

Figure 4-18 illustrates the room layout to simulate the face recognition inability due to low resolution where a person enters through a door and then moves from position 1 and position 2. In the view of camera 3, the face detection is not accomplished due to a low resolution. Given the door position an origin (0,0), each Camera 1, 2 and 3 is positioned at (0.6m, 5m), (5m, 0.6m) and (9m, 6.5m), respectively. The overall room size is 9.2m×7m.



(a) Position 1



(b) Position 2

Figure 4-18: Illustration of the room layout to simulate the face recognition inability due to a low resolution.

As shown in Figure 4-19, three cameras are detecting a person moving from the position 1 to 2. From the snapshot in a position 1, the person face is detected by only two cameras 1 and 2 while the body is detected by all three cameras 1, 2 and 3. In the scenario, the secondary detection uses the global position supporting face detection for a camera 3. Now, the person is moving to the position 2 which is covered by only camera 3. Since more than one camera needs to monitor the person for the global localization, one of cameras 1 or 2 needs to operate panning. In our simulation, Camera 2 is panning 30^o to the left direction, and finally two cameras 2 and 3 monitor the person. Since the secondary detection dynamically recovers the background image, Camera 2 easily detects and tracks the person. Through this simulation, we verified the cameras collaboration in order to alleviate the face recognition inability due to low resolution.

Figure 4-20 illustrates another layout to verify the performance of global trajectory in multiple cameras. The cameras 1 and 2 are positioned at (3.63m, 0m) and (0m, 2.97m), respectively. The overall room size is 5.5m×7m rectangle area. The dotted line represents the viewable range of each camera. Since our localization method uses more than one view of camera, the solution space is limited by the overlapped area of both cameras.

Figure 4-21 illustrates the trajectory of three people based on the body and face joint detection. In Figure 3-25(a), the view of Camera 1 detects only one motion or body. The body information includes three overlapped people. On the other hand, the view of Camera 2 detects three distinct separated motions or bodies. In the scenario, the global information of the three distinct people from Camera 2 supports

Camera1

Camera 2

Camera 3

(a) Snapshot in position 1

Camera1

Camera 2

Camera 3

(b) Snapshot in position 2

Figure 4-19: The secondary detection alleviating the problem with face recognition inability due to low resolution.

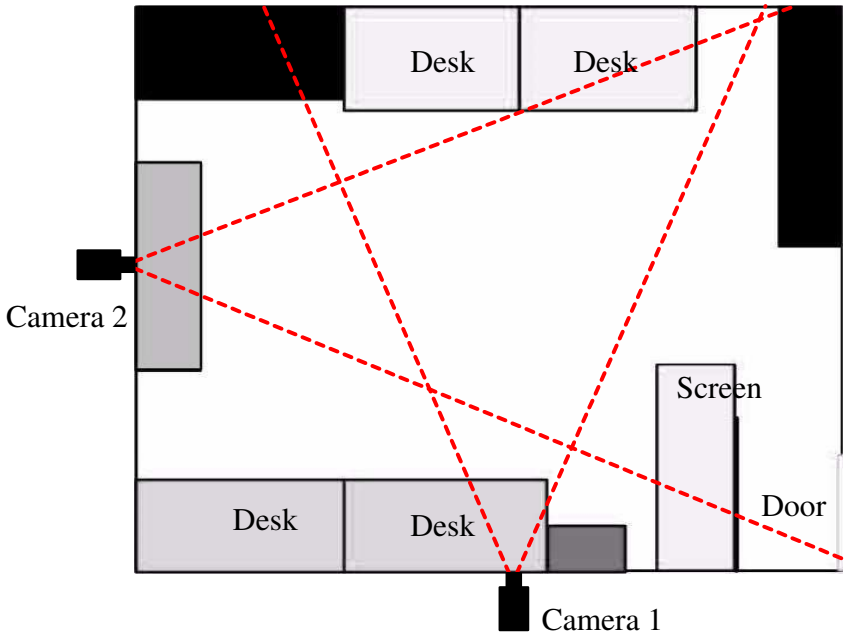
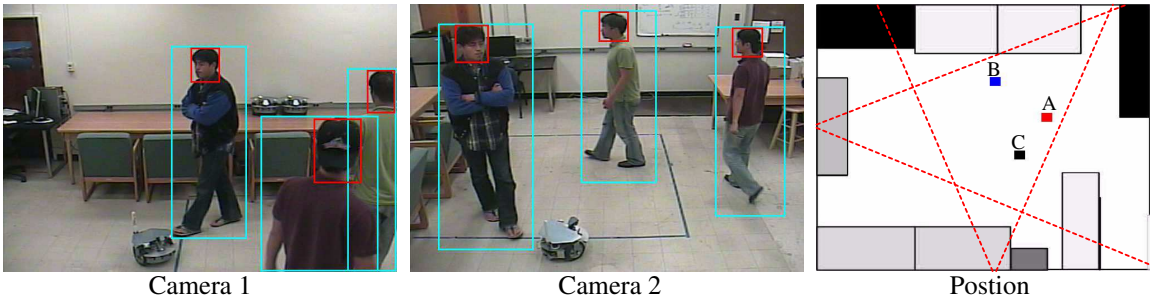


Figure 4-20: Illustration of the room layout to verify the performance of a global trajectory in a multiple camera environment.

the secondary detection to Camera 1. Similarly, Figure 3-25(b) and (c) present robust detection and global localization through the camera collaboration.

Camera 1 Camera 2 Postion

(a) Snapshot 1



Camera 1 Camera 2 Postion

(b) Snapshot 2

Camera 1 Camera 2 Postion

(c) Snapshot 3

Figure 4-21: Illustration of the secondary detection supporting a global localization.

Chapter 5

Future Research and Conclusion

5.1 Conclusion

The dissertation has presented the robust object detection and localization methods for real-time autonomous surveillance applications. First, the spectral characterization for efficient image detection using hyperspectral processing techniques has presented. We proposed an algorithm to reduce complexity and improve the library by using effective band selection and library refinement. The effective bands are heuristically selected for processing based on the contribution coefficient defined in this dissertation. The complexity of the proposed algorithm has been estimated in TMS320C6713 DSP. This approach has reduced the computation complexity. We have shown that for effective detection, only a small number of bands is needed.

Next, an accurate and effective object localization algorithm with visual images from unreliable estimate coordinates has proposed. In order to simplify the modeling of visual localization, the parallel projection model is presented where simple geom-

etry is used in computation. The algorithm minimizes the localization error through iterative approach with relatively low computational complexity. Non-linearity distortion of the digital image devices is compensated during the iterative approach. The effectiveness of the proposed algorithm in object position localization as well as tracking is illustrated. The proposed algorithm can be effectively applied in many tracking applications where visual imaging devices are used.

Finally, we have shown the robust human detection method in a multiple camera environment. Through the additional perspectives from multiple cameras, each camera sufficiently collaborates one another with additional information. The global localization from the multiple cameras based human detection enable us to monitor human movements in a global coordinate, and reversely assists the original detection which suffers from the single camera limitations. Furthermore, we showed that our proposed application supports the camera panning and zooming through the global localization with the dynamic image. We have shown that the proposed method is to significantly alleviate the single camera based detection limitation by performing simulations in a variety of multiple camera environments.

5.2 Future research

The proposed hyperspectral processing algorithm significantly reduces the computational complexity for processing, since the algorithm allows to use minimum number of spectral bands. However, the computation complexity of the proposed algorithm is still higher than conventional detection methods. Moreover, the hyperspectral

processing suffers from spatial resolution (i.e. image size). Once minimum bands are used, the complexity from spatial resolution becomes more significant than spectral complexity. Therefore, in the future work, we will present a FPGA implementation of hyperspectral algorithm which optimizes resource usage and satisfies high speed operation.

The human body and face joint detection provides the flexibility for complex detection problem. However, the computational complexity of the detection is critical for real-time processing in multiple camera environment. Moreover, the body and face joint detection has two types of detection: primary and secondary detections. The detections collaborate using local and global views to reinforce the object detection. However, the collaboration requires data dependency which limits the scalable system design. Besides in the aspect of object detection, a series of images have data redundancy. If a detected object is moving in an image, the detection of the object may not be required. Since the video stream as a general image source supports the motion information of two consecutive images, the data redundancy in a series of images can be minimized. In the future work, we present data partitioning to reduce computational complexity for object detection and tracking application. In addition we will propose data reduction strategy for improving overall processing time.

Bibliography

[1] J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding: CVIU*, vol. 73. no. 3, pp. 428-440, 1999.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: Real-Time Surveillance of People and Their Activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830. 2000.

[3] T. E. Boult, R. J. Micheals, X. Gao, and M. Eckmann, "Into the woods: visual surveillance of non-cooperative and camouflaged targets in complex outdoor settings," *Proc. IEEE, IEEE Press*, vol. 89, no. 10, pp. 1382-1402, 2001.

[4] A. Bakhtari, M. D. Naish, M. Eskandari, E. A. Croft and B. Benhabib, "Active-Vision based Multisensor Surveillance - An Implementation," *IEEE Trans. on Systems, Man and Cybernetic - Part C : Application and Riviews*, vol. 36, no. 5, pp. 668-680, Sep. 2006.

[5] N. X. Dao, B-J. You, S-R. Oh and Y. J. Choi, "Simple Visual Self-Localization for Indoor Mobile Robots using Single Video Camera," *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3767-3772, 2004.

[6] D. N. Zotkin, D. Ramani and L. S. Davis, "Joint Audio-Visual Tracking Using Particle Filters," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 11, pp. 1154-1164, 2002.

[7] T. Boggs and R. B. Gomez, "Fast Hyperspectral Data Processing Methods," *SPIE AeroSense 2001 Conference Proceeding*, pp. 74-78, 16-20 April 2001.

[8] Darren B. Ward, Eric A. Lehmann and Robert C. Williamson, "Particle Filtering Algorithms for Tracking an Acoustic Source in a Reverberant Environment," *IEEE Trans. on Speech and Audio Processing*, vol. 11, no. 6, pp. 826-836, Nov. 2003.

[9] Huang Lee and Hamid Aghajan, "Collaborative node Localization in Surveillance Networks using Opportunistic Target Observations," *ACM international workshop on Video surveillance and sensor networks*, pp. 9-18, 2006.

[10] O. Yakimenko, I.Kaminer and W. Lentz, "A Three Point Algorithm for Attitude and Range Determination using Vision," *Proceedings of the American Control Conference*, pp 1705-1709, June 2000.

[11] H. Tsutsue, J. Miura and Y. Shirai, "Optical Flow-Based Person Tracking by Multiple Cameras," *Proceeding of IEEE International Conference on Multisensor Fusion and Intergration for Intelligent Systems*, pp 91-96, 2001.

[12] Vincent Lepetit and Pascal Fua, "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey," *Foundation and Thrends in Computer Graphics and Vision*, 2005.

[13] Zhengyou Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, Nov. 2000.

[14] R. T. Collins, A. J. Lipton, H. Fujiyoshi and T. Kanade, "Algorithms for Cooperative Multisensor Surveillance," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1456- 1477, 2001.

[15] N. Doulamis, A. Doulamis and S. Kollias, "Efficient Content-Based Retrieval of Humans from Video Databases," *ICCV*, pp. 89-95, 1999.

[16] S. Piva, L. Comes, M. Asadi and C. S. Regazzoni, "Grouped-People Splitting Based on Face Detection and Body Proportion Constraints," *AVSS*, pp. 24-28, 2006.

[17] S. Khan and M. Shah, "Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1355-1360, 2003.

[18] Mei Han, Amit Sethi, Wei Hua and Yihong Gong, "A detection-based multiple object tracking method," *In Proceedings of ICIP'2004*, pp. 3065-3068, 2004.

[19] J. Lee, K.-S. Park, S. Hong and W. D. Cho, "bject Tracking Based on RFID Coverage and Visual Compensation in Wireless Sensor Network," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1052-1058, May 2007.

[20] G. A. Shaw and H. K. burke, "Spectral Imaging for Remote Sensing," *Lincoln Laboratory Journal*, vol. 14, num 1, pp. 3-28, 2003.

[21] Richard B. Gomez and Ambrose J. Lewis, "On-Board Processing for Spectral Remote Sensing," *ISPRS Special Session Future Intelligent Earth Observing Satellites (FIEOS)*, 2002.

[22] Sek M. Chai, Antonio Gentile, Wilfredo E. Lugo-Beauchamp, Javier Fonseca, J.L. Cruz-Rivera and D.S. Wills, "Forcal-plane processing architectures for real-time hyperspectral image processing," *Applied Optics*, vol. 39, no. 5, pp. 835-849, Feb. 2000.

[23] S. M. C. Nascimento, F. Ferreira and D. H. Foster, "Statistics of spatial cone-excitation ratios in natural scenes," *Journal of the Optical Society of America A*, vol. 19, no. 8, pp. 1484-1490, Aug. 2002.

[24] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," *Prectice Hall, second edition*, pp. 567-635, 2002.

[25] W. H. Bakker and K. S. Schmidt, "Hyperspectral Edge Filtering for Measuring Homogeneity of Surface Cover Types," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 56, pp. 246-256, 2002.

[26] M. L. Nischan, R. M. Joseph, J. C. Libby and J. P. Kerekes, "Active Spectral Imaging," *Lincoln Laboratory Journal*, vol. 14, num 1, pp. 131-144, 2003.

[27] M. K. Griffin and H. K. Burke, "Compensation of Hyperspectral Data for Atmospheric Effect," *Lincoln Laboratory Journal*, vol. 14, num. 1, pp. 29-54, 2003.

[28] G. P. Abousleman, M. W. Marcelline and B. R. Hunt, "Hyperspectral image compression using entropy-constrained predictive trellis coded quantizatin," *IEEE trans. Image Processing*, vol. 6. pp. 566-573, Apr. 1997.

[29] Nirmal Keshava, "Distance Metrics and Band Selection in Hyperspectral Processing With Applications to Material Identification and Spectral Librararies," *IEEE Trans. on Geoscience and Remote Sensing*, vow. 42, no. 7, pp. 1552-1565, July 2004.

[30] Peter Bajcsy and Peter Groves, "Methodology for Hyperspectral Band Selection," *Photogrammetric Engineering and Remote Sensing journal*, vol. 70, number 7, pp. 793-802, July 2004.

[31] S. Kumar, J. Ghosh and M. M. Clawford, "Best-Bases Feature Extraction Algorithms for Classification of Hyperspectral Data," *IEEE Trans. on Geoscience and Remote Sensing*, vol.39, no.7, July 2001.

[32] G. Girouard, A. Bannari, A. Harti and A. Desrochers, "Validated Spectral Angle Mapper Algorithm for Geological Mapping: Comparative Study Between Quickbird and Landsat-tm," *XXth ISPRS Congress*, pp. 599-605, July 2004.

[33] Rulph Chassaing, "Digital Signal Processing and Applications with the C6713 and C6416DSK," *wiley interscience*, 2005.

[34] Texas Instrument, "Datasheet of TMS320C6713B," *Available: http://focus.ti.com/lit/ds/symlink/tms320c6713b.pdf*, Nov. 2005.

[35] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth and L. Van Gool, "Color-based object tracking in multiple-camera environments," *Proceedings of the DAGM03*, pp. 591-599, 2003.

[36] H. Jin and G. Qian, "Robust Multi-Camera 3D People Tracking with Partial Occlusion Handling," *ICASSP 2007*, pp. 909-912, 2007.

[37] J. Berclaz, F. Fleuret and P. Fua, "Robust people tracking with global trajectory optimizaition," *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[38] O. Javed, S. Khan, Z. Rasheed and M. Shah, "Camera handoff: tracking in multiple uncalibrated stationary cameras," *IEEE Workshop on Human Motion*, pp. 113-118, 2000.

[39] V. Ayala, J. B. Hayet, F. Lerasle and M. Devy, "Visual Localization of a Mobile Robot in indoor Environments using Planar Landmarks," *Proceeding of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 275-280, 2000.

[40] K. Nickel, T. Gehrig, R. Stiefelhagen and J. McMonough, "A Joint Particle Filter for Audio-visual Speaker Tracking," *International Conference on Multimodel Interfaces*, pp 61-68, 2005.

[41] Paul E. Debevec, "Modeling and Rendering Architecture from Photographs," *University of California at Berkeley Computer Science Division, Berkeley CA*, 1996.

[42] M. Watannabe and S. K. Nayar, "Telecentric optics for computational vision," *ECCV96*, pp. 439-451, 1996

[43] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradig for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communicaions of the ACM*, vol 24, num. 6, pp. 381-395, 1981.

[44] C. Geyer and K. Daniilidis, "Omnidirectional Video," *The Visual Computer*, vol. 19, pp. 405-416, Oct. 2003.

[45] S. Spors, R. Rabenstein and N. Strobel, "A Multi-Sensor Object Localization System," *Proceedings of the Vision Modeling and Visualization Conference 2001*, pp. 19-26, 2001.

[46] Sylvain Bougnoux, "From Projective to Euclidean Space under Any Practical Situation, a Criticism of Self-calibration," *Proceedings of the Sixth International Conference on Computer Vision*, pp. 790-796, 1998.

[47] R.-K. Lenz and R.-Y. 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*, vol. 10, no. 5, pp. 713-720, Sep. 1988.

[48] 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.

[49] Fengjun Lv, Tao Zhao and Ramakant Nevatia, "Camera Calibration from Video of a Walking Human," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 28, no.9, pp 1513-1518, Sep. 2006.

[50] O.D. Faugeras, Q.-T. Luong and S.J. Maybank, "Camera Self-Calibration: Theory and Experiments," *European Conference on Computer Vision*, pp. 321-334, 1992.

[51] A. Zisserman, P. A. Beardsley and I. D. Reid, "Metric Calibration of a Stereo Rig," *IEEE Workshop on Representations of Visual Scenes*, pp.93-100, 1995.

[52] E. Horster, R. Lienhart, W. Kellermann and J.-Y. Bouguet, "Calibration of visual sensors and actuators in distributed computing platforms," *Proceedings of the third ACM international workshop on Video surveillance and sensor networks*, pp. 19-28, 2005.

[53] Peter Sturm and S. J. Maybank, "On plane-based camera calibration: A general algorithm," *IEEE Conf. Computer Vision and Pattern Recognition*, 1999.

[54] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, "A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry," *Artifical Intelligence Journal*, 87-119, Oct. 1995.

[55] Qurban Memon and Sohaib Khan, "Camera Calibration and Three-dimensional World Reconstruction of Stereo-vision using Neural Networks," *International Journal of Systems Science*, vol. 32, num. 9, pp. 1155-1159, 2001.

[56] R. Cipolla, T. W. Drummond and D. Robertson, "Camera Calibration from Vanishing Points in Images of Architectural Scenes," *Proc. British Machine Vision Conference*, vol.2 pp. 382-391, Sep. 1999.

[57] P. A. Beardsley, A. Zisserman and W. Murray, "Sequential Updating of Projective and Affine Structure from Motion," *International Journal of Computer Vision*, vol. 23, no. 3, pp. 235-259, 1997.

[58] O. Faugeras, "Stratification of three-dimensional vision: projective, affine, and metric representations: errata," *Journal of Optical Society of America*, vol. 12, no. 3, pp 465-484, 1995.

[59] T. Moons, L. Van Gool, M. Proesmans and E. Pauwels, "Affine Reconstruction from Perspective Image Pairs with a Relative Object Camera Translation in Between," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 77-83, 1996.

[60] M. Pollefeys and L. V. Gool, "A Statified Approach to Metric Self-Calibration," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 407-412, June 1997.

[61] P.A. Beardsley and A. Zisserman. "Affine calibration of mobile vehicles," *Europe-China workshop on Geometrical Modelling and Invariants for Computer Vision*, Xi'an, China, 1995.

[62] J. J. Koenderink and A. J. van Doorn, "Affine structure from motion," *Journal of Optical Society of America*, vol. 8, no. 2, pp. 377-385, 1991.

[63] P. Sturm and L. Quan, "Affine Stero Calibration," *Proc. CAIP'95*, pp. 838-843, 1995.

[64] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: a Factorization Method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137-154, 1992.

[65] C. J. Poelman and T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery," *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206-218, 1997.

[66] http://www.usa.canon.com/

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

[68] J. M. Rehg, M. Loughlin and K. Waters, "Vision for a Smart Kiosk," *Computer Vision and Pattern Recognition*, pp. 690-696, 1997

[69] W. Hu, T. Tan, L. Wang and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Trans. Systems, Man, and Cybernetics, Part C:Applications and Reviews*, vol. 34, no. 3, pp. 334-352, 2004.

[70] Jie Yang and Alex Waibel, "A Real-Time Face Tracker," *IEEE Proc. of the 3rd Wokshop on Applications of Computer Vision*, pp. 142-147, 1996.

[71] 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, pp. 696-706, 2002.

[72] D. Nguyen, D. Halupka, P. Aarabi and A. Sheikholeslami, "Real-time face detection and lip feature extraction using field-programmable gate arrays," *IEEE Trans. Systems, Man and Cybernetics*, vol. 36, no. 4, pp. 902-912, Aug. 2006.

[73] Kah Kay Sung and Tomaso Poggio, "Example Based Learning for View-Based Human Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 170-177, Jan 1998.

[74] Rahul Swaminathan, Michael D. Grossberg and Shree K. Nayar, "A Perspective on Distortions," *IEEE Conf. Computer Vision and Pattern Recognition*, pp.594-602, 2003.

[75] R. Y. Tsai, "A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology using Off-the-shelf TV Cameras and Lenses," *IEEE J. Robotics Automat.* , pp. 323-344, vol. RA-3, no. 4 1987.