國 立 交 通 大 學

資訊學院資訊科技（IT）產業研發碩士班

碩 士 論 文

整合多攝影機之三維監控系統

A 3-D Surveillance System using Integrated Multiple Cameras

研 究 生：黃永煌

指導教授：陳永昇　教授

中 華 民 國 九 十 九 年 一 月

整合多攝影機之三維監控系統
A 3-D Surveillance System using Integrated Multiple Cameras

研　究　生：黃永煌　　　　Student：Yong-Huang Huang

指導教授：陳永昇　　　　Advisor：Yong-Sheng Chen

國　立　交　通　大　學
資訊學院資訊科技（IT）產業研發碩士班
碩　士　論　文

A Thesis

Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Industrial Technology R & D Master Program on
Computer Science and Engineering

January 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年一月

摘　　要

本論文發展一套視訊監控系統，整合多支攝影機的畫面到三維模型上，並擷取影像中的移動物體加以立體化，呈現一個具空間感且可即時互動的監控畫面。常見的傳統監控系統是利用電視牆來顯示所有攝影機的畫面，監控人員往往很難了解每個畫面之間的空間對應關係，並且容易混淆，在情況緊急時，可能無法做出正確的判斷。透過我們的系統，監控人員可以任意選擇想要觀看的視點，且攝影機的空間關係已融入監控環境的三維模型。如此一來，同時監控多支攝影機不再是一件困難的事。

本系統可以分為兩部分，建立攝影機和三維模型的對應關係，和擷取移動物體加以立體化。首先，我們使用平面補釘模型化(planar patch modeling)建立三維模型，然後在監控場景中放入一些標記，量取標記之間的相對位置關係，再利用單應性矩陣(homography matrix)來表示攝影機畫面和監控場景的對應關係。利用這些資訊建立查詢表(lookup table)，來融合出三維的監控場景。接著，使用背景模型(background modeling)偵測移動物體，並使用告示板(billboarding)技術來立體化顯示此物體的影像。最後我們得到一個三維視覺化的整合監控系統。

Abstract

In conventional surveillance systems, multiple screens are often required for displaying video from multiple cameras and may cause the difficulty of operators to keep track of targets due to the lack of special relationship among the screens. In this thesis, we develop an effective surveillance system with the 3-D environment model that integrates multiple scenes into one single comprehensive view. The system does not require accurate camera calibration and environment model construction with advanced equipments, and can provide a multiscale operating view for showing the status of the surveillance area.

To integrate the monitored area with camera views, the 3-D environment is first manually constructed by planar patch modeling. To map video contents to the corresponding areas of the 3-D model, different homography transformations are estimated for every pairs of image regions in the video contents and corresponding areas in the 3-D model. The planar patches in the 3-D model are automatically divided into different sizes and numbers of the smaller patches are determined by the intensity differences between the image polygons obtained from the homography transformation and texture mapping. Lookup tables are built beforehand for accelerating the coordinate mapping. In monitored scenes, 3-D objects including pedestrians, are projected to the imaging sensor planes through perspective transformation. Therefore, all 3-D objects will appear flattened on the planes in the 3-D model after texture mapping. To overcome this problem, we use billboarding method to model the 3-D moving objects. First, each foreground object extracted from the scene by background modeling is mapped to a billboard, and then vertically aligned to the ground plane. In this way we can adjust the direction of foreground objects according to the viewing direction. The proposed system can provide operators the situational awareness of the monitored site, including activities of the tracking targets through a comprehensive 3-D view.

致　　　謝

在此特別要感謝我的指導教授陳永昇老師，老師的指導和實驗室自由的風氣，讓我學習很多；還有 BSP 所有的夥伴們，乙慈、郁萱、亮亮、發哥、育宏、億婷、慧伶，跟你們相處的日子很快樂、很充實；另外詠成學長在研究方面，給我很多指導協助，小苑和柏志也常常幫助我的實驗;還有每個交大的朋友,你們讓我兩年的碩士生活豐富精彩。另外，感謝我的家人，你們不斷支持我，是我努力的動力；還有我的女友，在我遭遇挫折時，妳的鼓勵讓我可以繼續前進。此論文獻給所有幫助我的朋友和我愛的家人。

Contents

List of Figures

Chapter 1

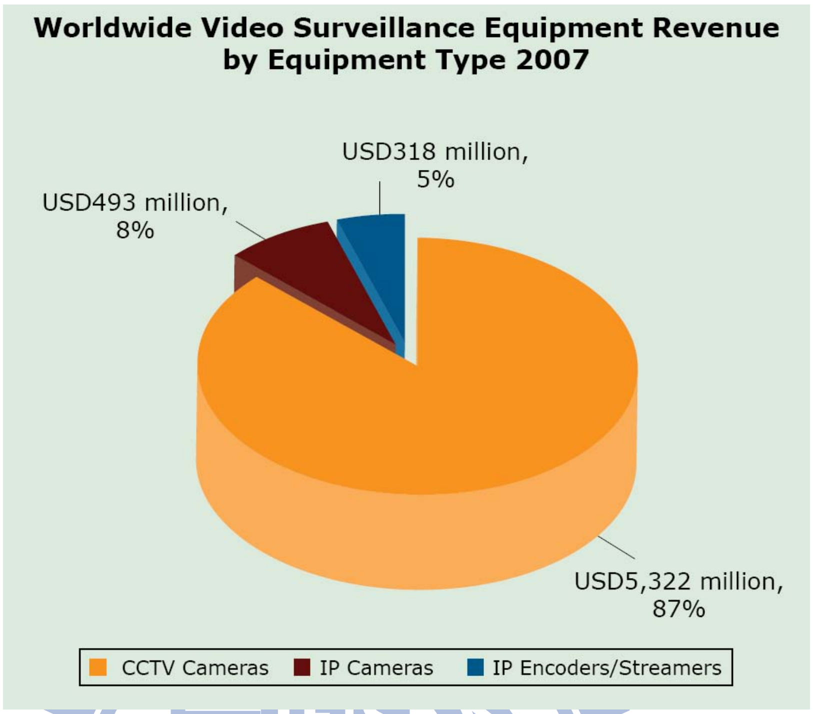
Introduction

1.1 Background

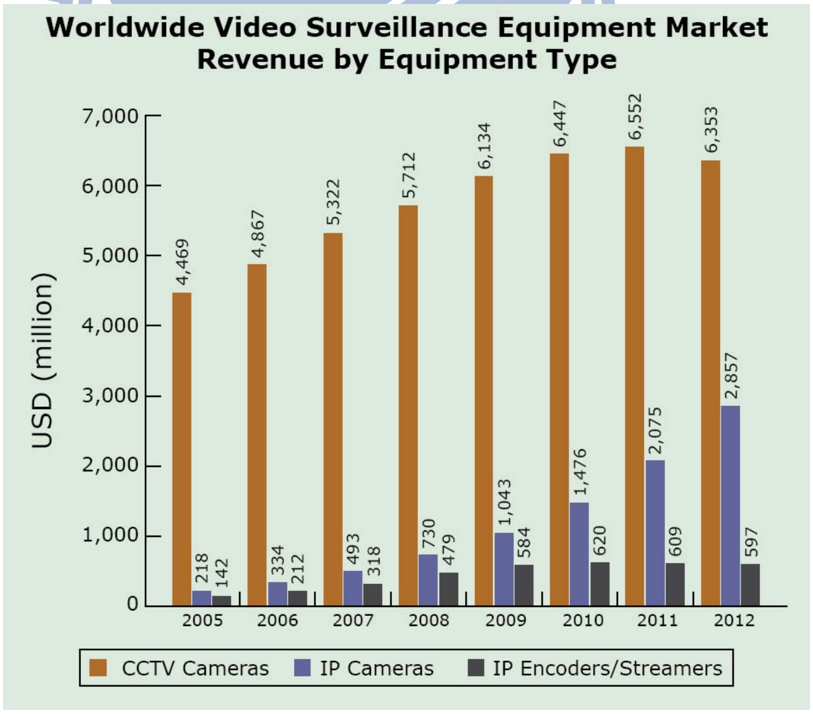
Security is always a critical topic from all aspects. According to the Peace Corps, the number and severity of personal safety and security issues is rising in virtually all countries. By the development of the technology, a wide variety of sensors (video cameras, RFID (Radio Frequency Identification) tags, microphones, infrared badges, fingerprint sensor, etc.) are applied to government organizations, popular buildings, main lines and streets for the purpose of security. In addition, video cameras are extensively used because of their direct and effective visual information. Since the videos from cameras recorded by DVRs (Digital Video Recorders) or NVRs (Network Video Recorders), the image information is the proof of criminal events or the material for data analysis. Recently, video surveillance industry has experienced accelerated growth due to continuously decreasing price and better capability of cameras. The increasing trend of the worldwide video surveillance equipment revenue is shown in Figure 1.1 [18]. Although the CCTV (Closed Circuit Television) cameras are still the most common equipments in video surveillance, key market lately trends to IP (Internet Protocol) cameras. Typical IP-based systems use IP cameras over a LAN (Local Area Network) to any PC (Personal Computer) or server on the network. Since the cameras are IP addressable, they can be effectively accessed from anywhere in the world through wired and wireless options in internet. Consequently, IP cameras are used in many places, and also in our experiment.

Since the areas of surveillance are often wide and the field of view of the camera is limited, multiple cameras are required to cover the whole monitored area. How to combine image data from cameras into the useful and easy understanding information for users is an important research topic. Usually, security guards have two tasks, monitoring and tracking. They frequently scan the videos and try to discover doubtful people or unusual situations. If they discover something suspicious, they may keep track of targets or enlarge the view to figure out what happened and make reactions. In the conventional surveillance system, security guards in the control center watch the area through a monitor wall which is composed of a group of independent screens (Figure 1.2). Such surveillance system has some problems:

1. There are too many screens.



(a)



(b)

Figure 1.1: **The worldwide video surveillance equipment revenue. (Figure source: [18])** (a) The worldwide video surveillance equipment revenue in 2007. (b) The trend of the worldwide video surveillance equipment revenue.

2. It is difficult to connect similar scenes in different screens with corresponding places in the real world.

3. The security guard can be easily confused due to the lack of the spatial relationship between cameras.

For further illustrations, a security guard can only focus on few screens at the same time. It is tiresome to simultaneously gaze between many screens over a long period of time. Besides, the guard may be confused with some scenes , such as stairwells which are much similar in the building. Moreover, a monitored area often covered with several cameras, so a pedestrian showed in these cameras at the same time is hard to track. Also, the spatial relationship between each screen is not intuitive and the guard has a bad interaction with cameras through the monitor wall. Although the images are the useful data for surveillance, the guard is overtook with a large amount of data without synthetic information. In conclusion, the guard needs to integrate the separate images from all cameras in their cognitive processing of the data to understand the connection between the images and how the situation changes in the scene. Security guards might take several days to be familiar with the relationship between cameras and it might be longer for novice or low spatial ability users [20]. Therefore, it is necessary to develop a surveillance system to integrate all scenes into a single comprehensive view for tackling the above problems.

In this thesis, we want to integrate camera views with a virtual 3-D environment. The surveillance system provide a comprehensive view that the guard can monitor multiple scenes simultaneously. The system must be efficient, so the operator can do the immersive interaction with monitored views.

1.2 Survey of video surveillance system

There are many researches on the surveillance system of multiple cameras integration. We will introduce these related works in this section.

1. Contextualized videos: combining videos with environment models to support situational understanding

Figure 1.2: **A conventional surveillance system with multiple screens.**

2. Accurate planar image registration for an integrated video surveillance system
3. Video flashlights: real time rendering of multiple videos for immersive model visualization
4. 3D video surveillance with augmented virtual environments
5. Multi-texture modeling of 3D traffic scenes

Contextualized videos for situational understanding ”Contextualized videos” is defined as a combination of videos with a model of the 3-D environment [21]. It allows observers to see the activities in the videos in their proper locations. The visualization frame work is shown in Figure 1.3. The paper focuses on the layout design [16, 17] and model processing [4, 10]. Associated videos, video billboards, video on fixed planes, video projection and dynamic imagery are explained in layout design. Also, some techniques, such as drawers, rotate-and-shear, landmark, semitransparency and wireframe, are mentioned in model processing. The results of the combination with different visualization techniques is shown in Figure 1.4.

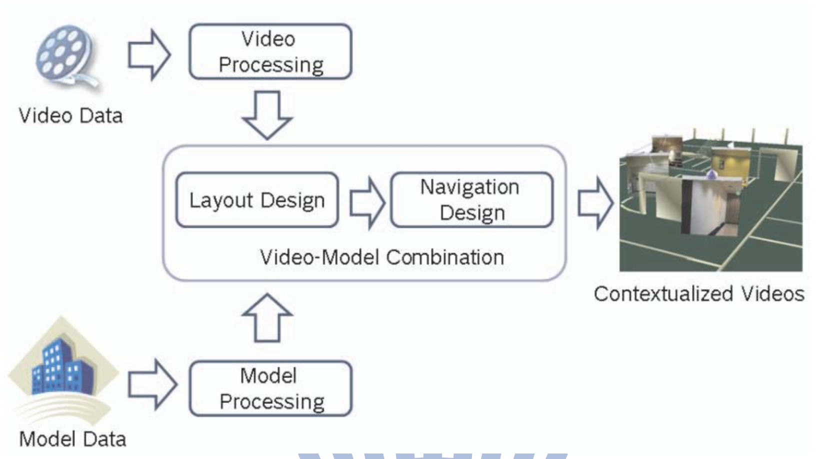


Figure 1.3: **The contextualized video visualization design framework. (Figure source: [21])**

Accurate planar image registration for an surveillance system The 3-D environment of the outdoor surveillance system is constructed from an aerial photograph and a height map of the monitored area (Figure 1.5). The accurate planar image registration is used for precise texture mapping of videos [3]. Images from two cameras stitched on the 3-D model and the operator can move the viewpoint to keep an eye on the moving target with a closer look (Figure 1.6).

Video flashlights: real time rendering for immersive visualization Video projection is applied for real-time mapping of the live videos on the 3-D model like ”video flashlights” [16]. This paper uses an aerial or satellite photograph as the base of the model. The rendering view shows the effect of ”illuminating” the scene with live cameras (Figure 1.7). The system including seven cameras detects moving objects and render them as icons representing people and vehicles (Figure 1.8).

3-D video surveillance with augmented virtual environments An AVE (augmented Virtual Environment) system that fuses and displays multiple image streams onto a 3D model substrate that allows arbitrary viewing [13]. An aerial image of the city is used

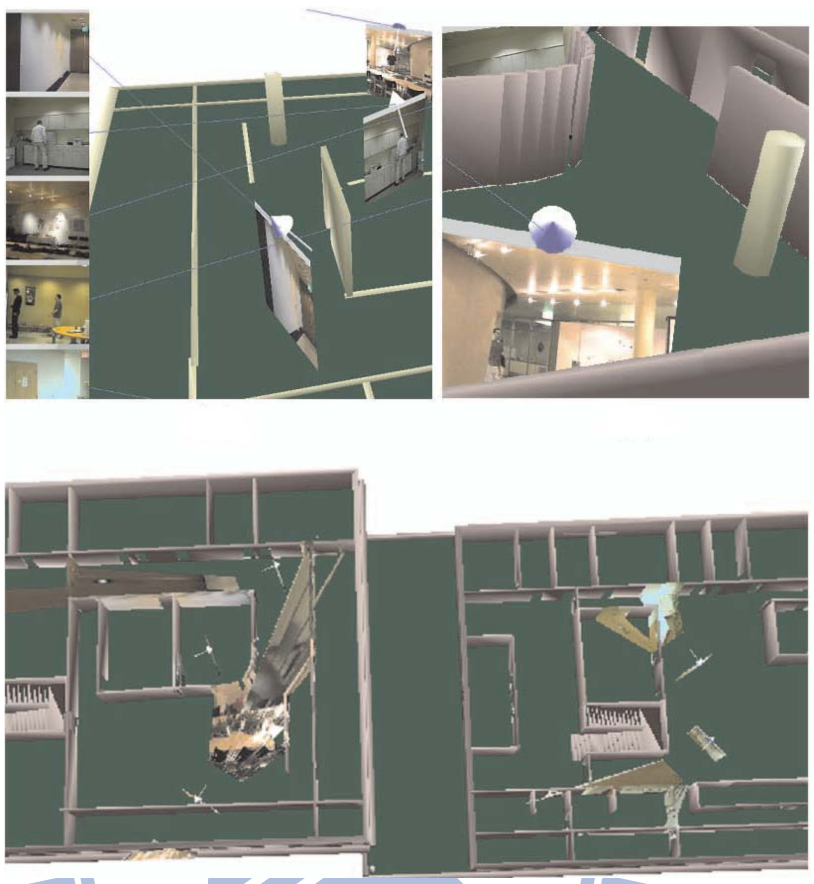


Figure 1.4: **Results of of the combination with different visualization techniques. (Figure source: [21])**

to be the foundation of the 3-D environment (Figure 1.9). The paper utilizes an airborne LiDAR (Light Detection and Ranging) sensor system to collect 3-D geometry samples of a specific environment [15]. The system maps videos onto the 3-D model like ”virtual projectors” and create a dynamic single polygon approximate model of the moving object to visualize it (Figure 1.10) [17].

Multi-texture modeling of 3-D traffic scenes A system for 3-D reconstruction of traffic scenes is presented in [12]. The foreground objects, like cars, are visualized by mapping images to synthetic 3-D wireframe object (Figure 1.11). The rendered view is interpolated from all textures with their individual weight depending on the actual position and viewing direction in the scene. This enables a smooth transition when

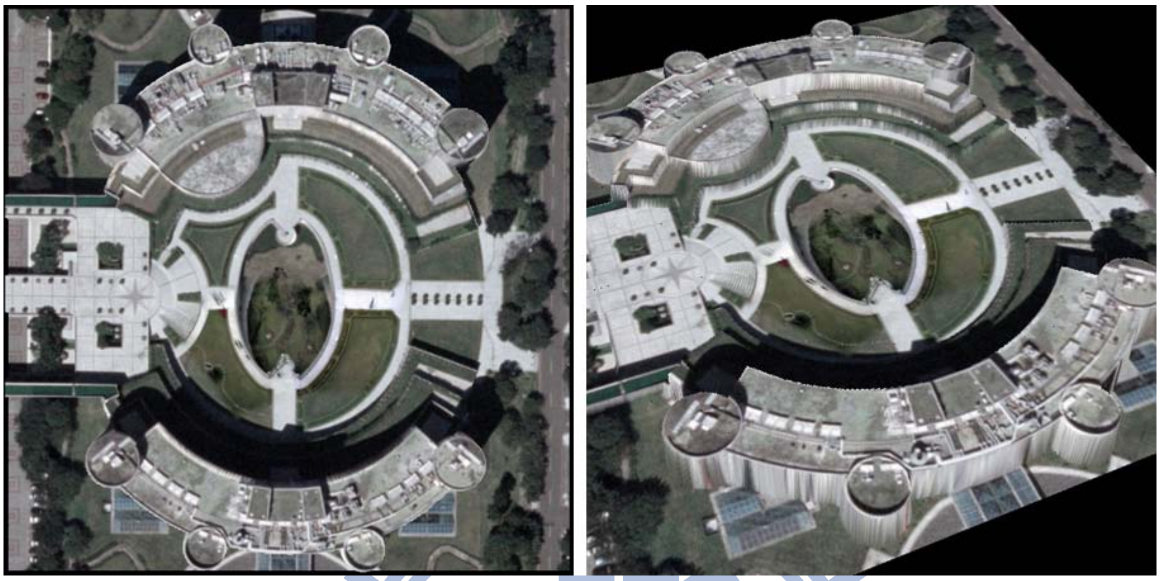


Figure 1.5: **3-D rendering of the whole monitored area. (Figure source: [3])**

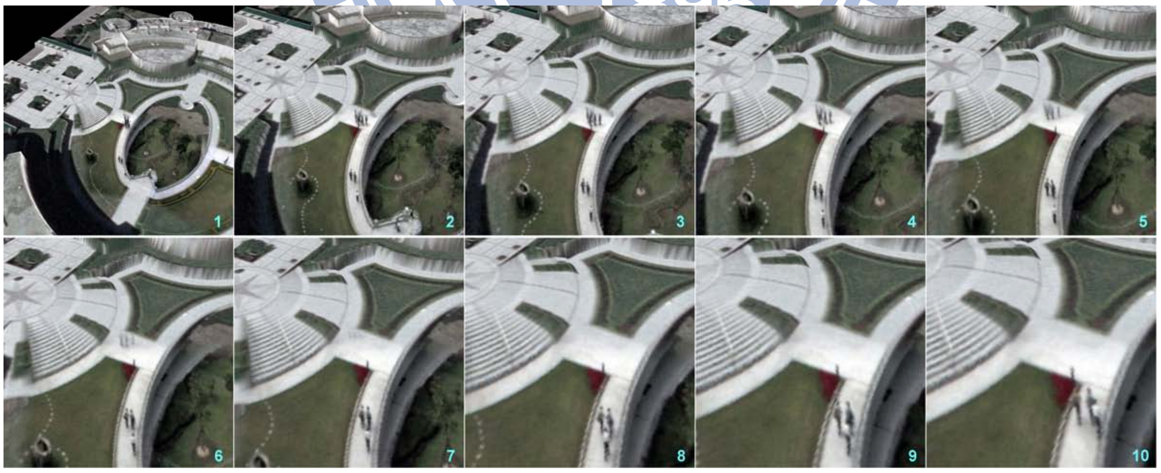


Figure 1.6: **Zoom in the view for specific targets. (Figure source: [3])**

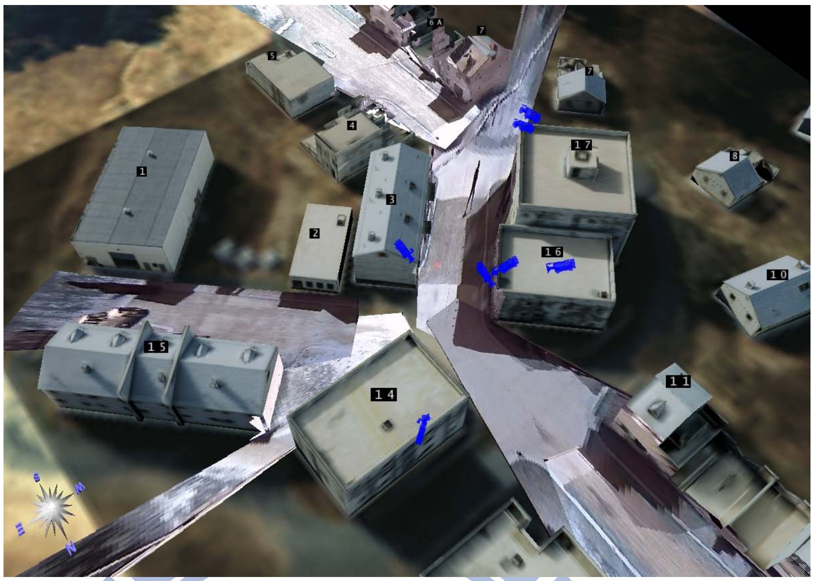


Figure 1.7: **A rendering view of the textured model and flashlight camera textures. (Figure source: [16])**

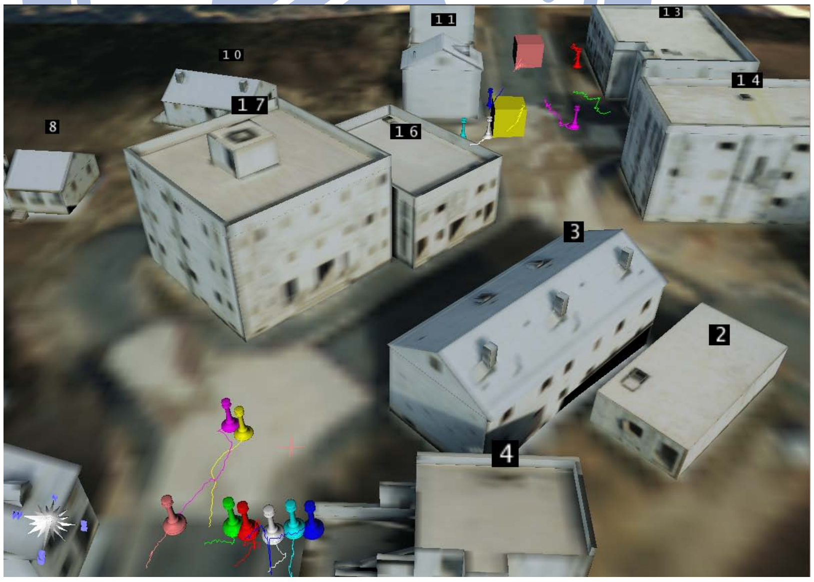


Figure 1.8: **Moving objects are detected and rendered as icons representing people and vehicles in the 3-D model. (Figure source: [16])**

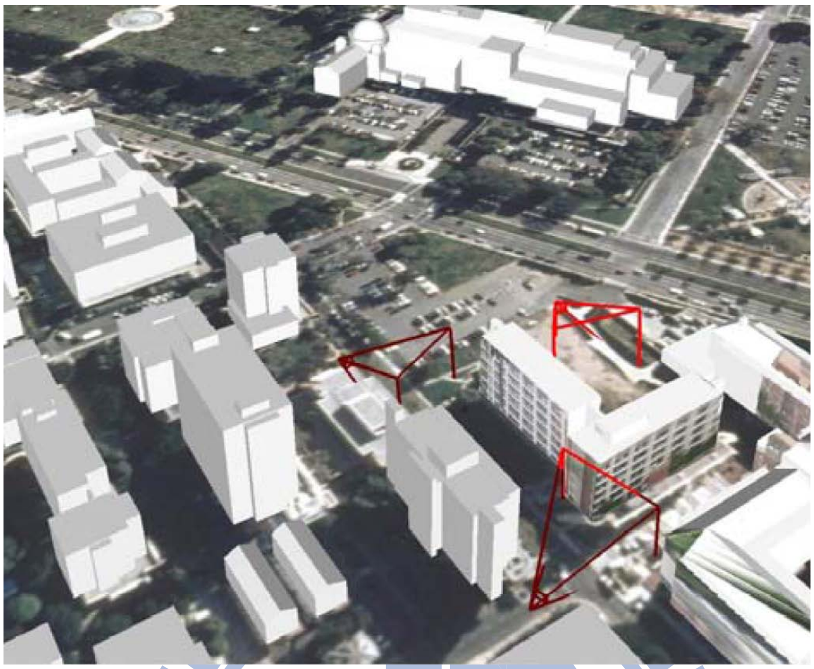
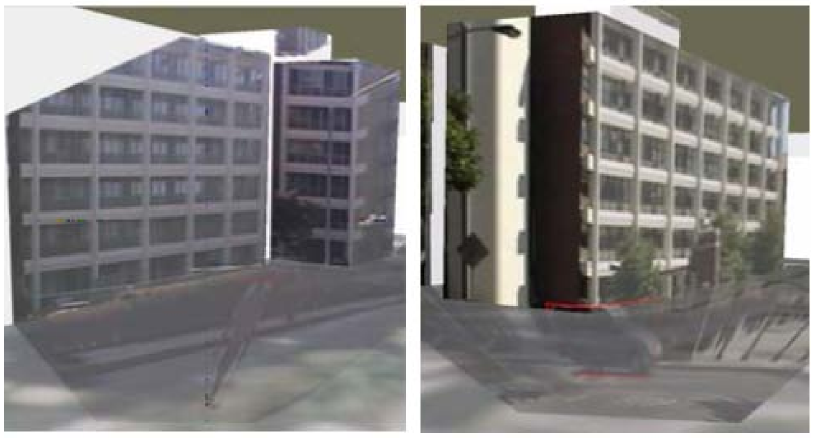


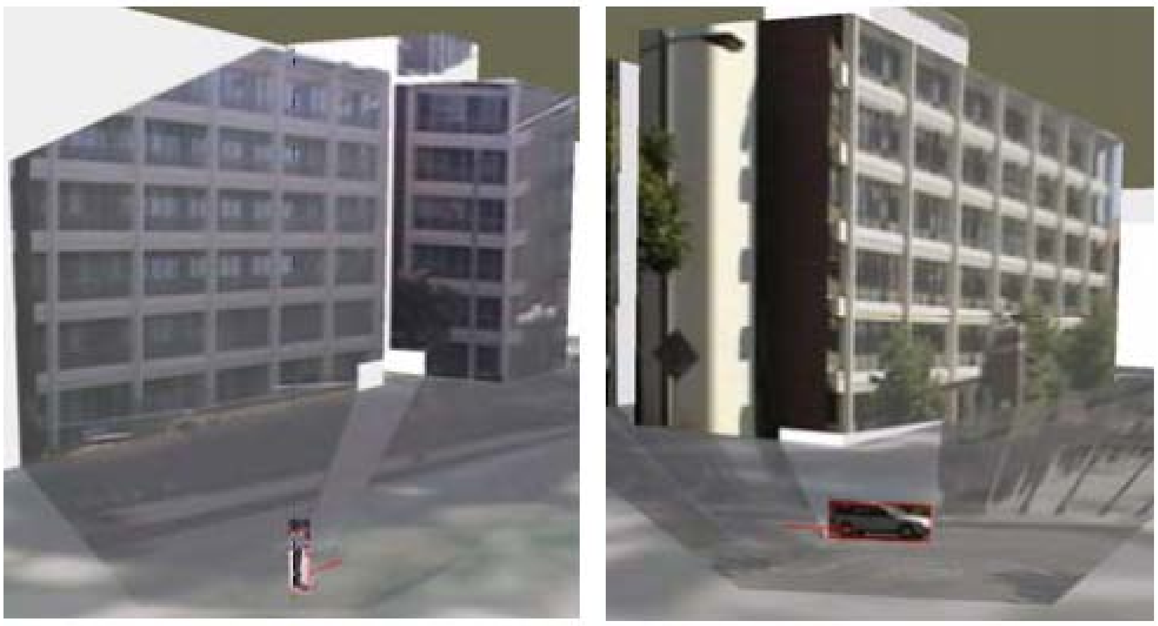
Figure 1.9: **An AVE system screen snapshot. (Figure source: [17])**

navigating the scene (Figure 1.12).

While the above systems offer many useful capabilities, there are still have some problems. Although video projection is an effective way to show the video streams on the 3-D model, precise camera calibration is necessary for projecting the images accurately and the buildings of the 3-D model constructed with advanced equipments or other efforts [8, 14, 19, 24]. Moreover, the 3-D moving objects appear to be distorted if they are not modeled as 3-D objects before [3]. While 3-D objects rendered as icons, real image contents from cameras are abandoned [16]. Also, 3-D reconstruction for 3-D objects contributes the better rendering result of foreground objects but it is difficult to precisely combine 3-D objects with all corresponding 2-D images from different cameras. Moreover, 3-D reconstruction of foreground objects might increases the computer loading [12]. In another way, to insert a polygon at the 3D positions of the foreground objects in the scene for the visualization of foreground objects is much easily applied without heavy loading for the computer. However, the polygons occlude background parts of the scene and create



(a)



(b)

Figure 1.10: **3D visualization for moving objects. (Figure source: [17])** (a) Image projection of moving objects without corresponding models. (b) Dynamic models for moving objects.

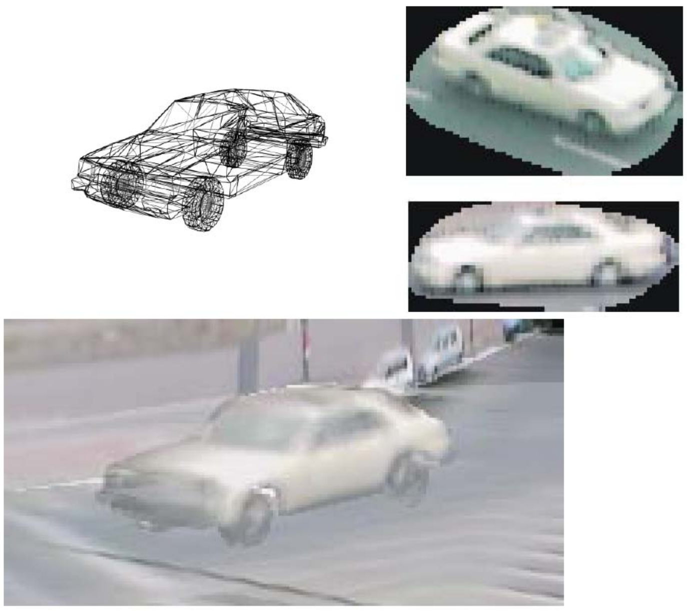


Figure 1.11: **Synthetic 3D object for the vehicles. (Figure source: [12])**

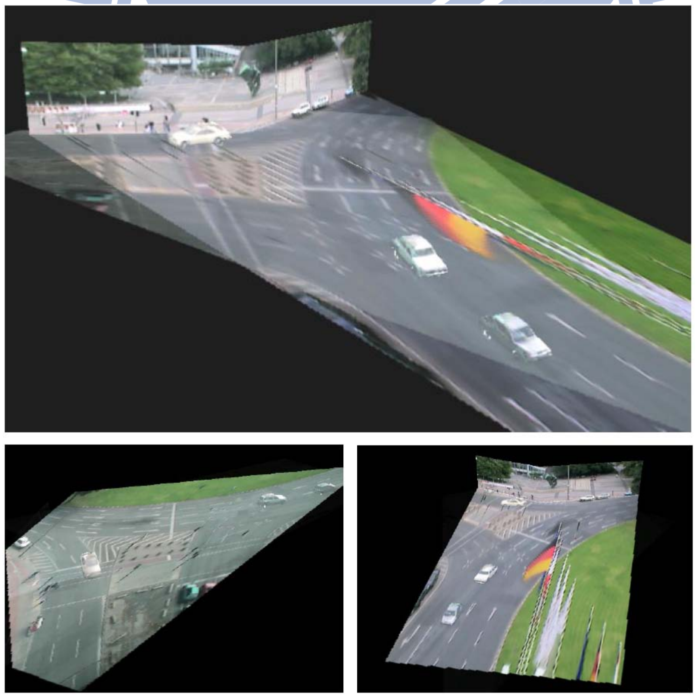


Figure 1.12: **The scene views with reconstructed dynamic objects from different directions. (Figure source: [12])**

texture gaps on buildings in particular [17]. We propose a surveillance system to tackle these problems.

1.3 Thesis overview

The flowchart of the proposed system is shown in Figure 1.13. In this thesis, we develop an effective 3-D surveillance system based on multiple cameras. In the part of system configuration, we use planar patch modeling to build the 3-D environment model. To map video contents to the corresponding areas of the 3-D model, different homography transformations are estimated for every pairs of image regions in the video contents and corresponding areas in the 3-D model. The planar patches in the 3-D model are automatically divided into different sizes and numbers of the smaller patches by our proposed method. Lookup tables are built beforehand for accelerating texture mapping . In monitored scenes, 3-D objects including pedestrians, are projected to the imaging sensor planes through perspective transformation and will be flatten on the planes in the 3-D model after texture mapping. In order to overcome this problem, billboarding method is involved in modeling the 3-D moving objects. The foreground objects are first extracted by background modeling and then are vertically aligned to the ground plane with billboards. In this way we can adjust the direction of foreground objects according to the viewing direction. Finally, operators can monitor a large area through a single comprehensive 3-D view.

1.4 Thesis organization

The remainder of this thesis is as follows. We describe how to integrate camera views into a monitored area in chapter 2. Chapter 3 and chapter 4 show the construction of the proposed system and experiments respectively. Finally, chapter 5 presents our conclusions.

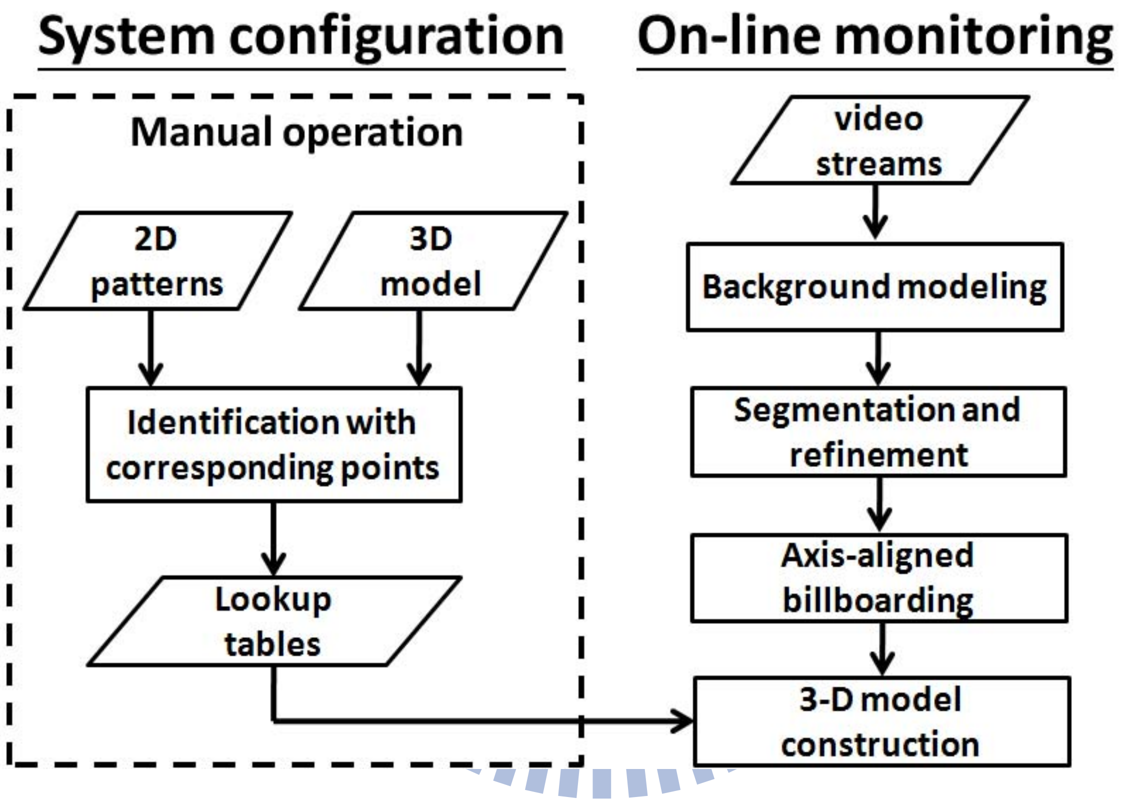


Figure 1.13: **The flowchart of the proposed system.** There are two parts, system configuration and on-line monitoring, in the construction of the system. Before on-line monitoring, we should first build the lookup tables for texture mapping.

Chapter 2

Integration of Camera Views and Monitored Area

The proposed system uses planar patches to model the 3-D environments and then maps video contents to the 3-D model through homography transformation. The planar patches in each scene are divided into smaller patches for better rendering results. Then, the results of the patch registrations are used to build the lookup tables for speeding up texture mapping. Since the cameras is fixed, all manual initialization need to be done only once.

2.1 Planar patch modeling

Camera calibration is not an easy work, especially precise calibration. If the calibration has a little inaccuracy, the result of projection might be miss-aligned. Planar patch modeling is done by manual efforts and we can refine the model easily. It is suitable for for both indoor and outdoor environments.

In most surveillance scenes, we can roughly classify objects into two parts, horizontal planes and vertical planes. The horizontal planes, such as hallways and floors, are usually surrounded by doors and wall, which are considered as the part of vertical plane. Both two planes are divided into several planar patches according to the geometry of the scenes. For example, walls in different planes modeled with different planar patches. If the scene consists of simple structures, we can use a bigger patch to model it (Figure 2.1 (a)). If we need an accurate registration for a complex environment, dividing the scene into more patches can obtain a better layout (Figure 2.1 (b)). In the overlapping areas, only one view is chosen by considering the integral layout of the 3-D environment. A benchmark is set in the real world and the distance from each patch to the benchmark manually measured. The coordinate of each point in the model should be defined with unary coordinate system. With these information, we could construct a virtual 3-D environment which is shown in Figure 2.2. In Figure 2.2, different colors of patches indicate textures from different camera views. The following section explains the method that maps the image texture onto this virtual 3-D environment model.

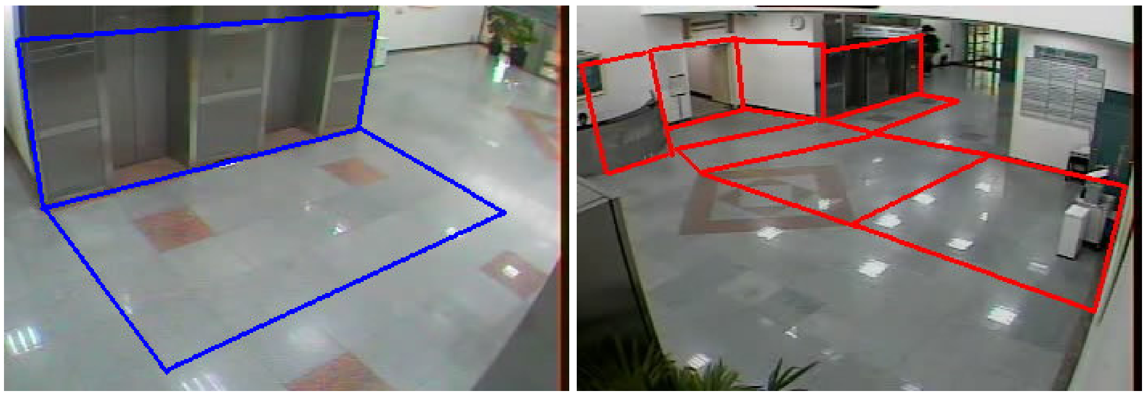

(a) (b)

Figure 2.1: **The scene with planar patches** (a) The scene consists of simple structures (b) The scene consists of complex structures.

2.2 Image registration

In previous section, we divide each scene into several planar patches and the virtual 3-D model is consist of these patches. In order to display the video contents on the model, we estimate the relationships between 2-D images and the 3-D model. For a point on a planar patch, the 3-D coordinates of the point in world coordinate can be seen as 2-D coordinates since one dimension is fixed. Hence, the 3-D model and the 2-D video contents can be mapped through homography, which is a transformation between two planar coordinate systems [6].

A homography matrix **H** represents the relationship between points on two planes:

$$s\mathbf{c_t} = \mathbf{H}\mathbf{c_s} \tag{2.1}$$

where s is a scalar factor which normalizes the homogeneous coordinates, and $\mathbf{c_s}$ and $\mathbf{c_t}$ are a pair of corresponding points in the source and target patches, respectively. Turn equation 2.1 into matrix form:

$$s \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \tag{2.2}$$

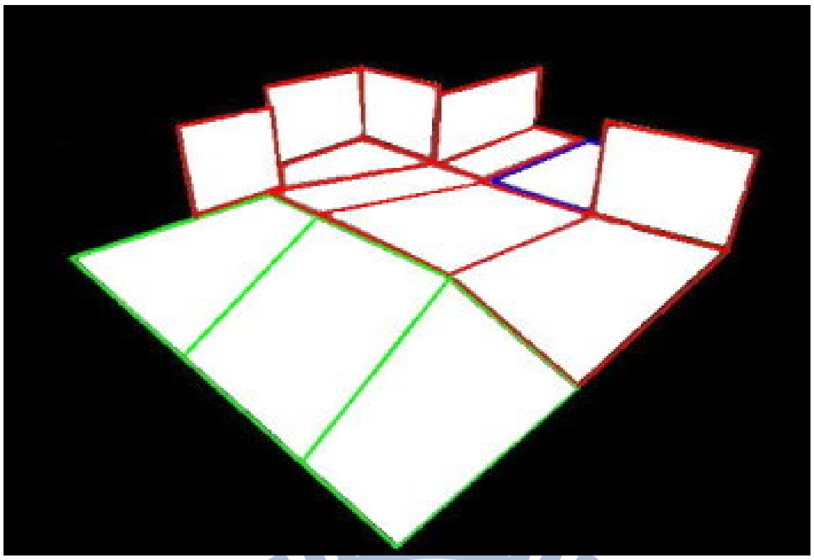


Figure 2.2: **A virtual 3-D environment model for the monitored area.**

We spread the equation 2.2:

$$\begin{cases} s &= x_s h_{31} + y_s h_{32} + 1 \\ su_t &= (x_s h_{11} + x_s h_{12} + h_{13}) \\ sv_t &= (x_s h_{21} + x_s h_{22} + h_{23}) \end{cases} \tag{2.3}$$

and rearrange equation 2.3 into equation 2.4 and equation 2.5:

$$\begin{cases} u_t &= x_s h_{11} + x_s h_{12} + h_{13} - x_s u_t h_{31} - y_s u_t h_{32} \\ v_t &= x_s h_{21} + x_s h_{22} + h_{23} - x_s v_t h_{31} - y_s v_t h_{32} \end{cases} \tag{2.4}$$

$$\begin{bmatrix} x_s & y_s & 1 & 0 & 0 & 0 & -x_s u_t & -y_s u_t \\ 0 & 0 & 0 & x_s & y_s & 1 & -x_s v_t & -y_s v_t \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} u_t \\ v_t \end{bmatrix} \tag{2.5}$$

$$\mathbf{Ah} = \mathbf{b} \tag{2.6}$$

where $\mathbf{A}$ is a $2n \times 8$ matrix, $\mathbf{h}$ is a 8×1 vector, and $\mathbf{b}$ is a $2n \times 1$ vector. Since there are more than four correspondences ($n \geq 4$) where no three correspondences in each patch are collinear, we can estimate $\mathbf{h}$ through the least-squares approach and obtain homography matrix $\mathbf{H}$.

We set $\mathbf{c_s}$ as points of the planar patch in the 3-D model and $\mathbf{c_t}$ as points of the planar patch in the 2-D image and obtain the matrix $\mathbf{H}$ for mapping the 3-D model to the 2-D image. In reverse order, we could also map the 2-D image to the 3-D model through the inverse matrix of $\mathbf{H}$. Now the contents of video can display in the 3-D model by the homography matrix $\mathbf{H}$.

2.3 Rendering refinement

In opengl, four pairs of corresponding points are required to rendering a polygon. Figure 2.3 show the rendering result of the planar patches we built before. There is serious distortion along the diagonal of the polygon because opengl cuts a ploygon into two triangles and renders them by texture mapping. The size of the triangles affects the performance of rendering. Since linear interpolation is used to fill objects with in texture mapping and violates the law of perspective projection, there might be some distortion in rendering layout. For each polygon, the more triangles we use, the better appearance and larger cost we get. If we only use two triangles to render a large polygon, the rendering will be executed fast but the scene distorts a lot. More triangles cause more accurate rendering (Figure 2.4). It is the tradeoff between efficiency and accuracy.

We proposed a procedure that automatically decides the number of triangles. In previous section, the homography matrix has been estimated and it is used to mapping the corresponding coordinates between two planes. Through homography transformation, we obtain the corresponding 3-D location of each pixel which is in the patches on the 2-D image. Texture mapping is applied to render these polygons in the 3-D model by bilinear interpolation. The intensity differences between the image polygons obtained from the homography transformation and texture mapping are defined as the distortion degree of the

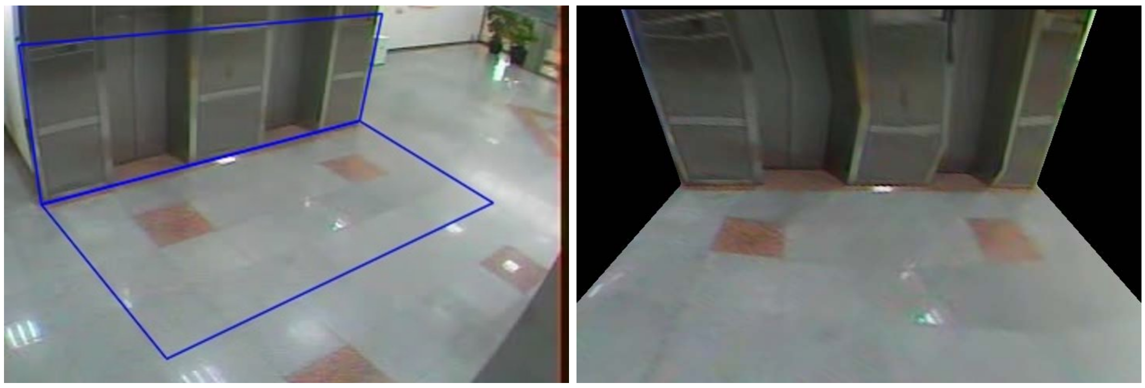
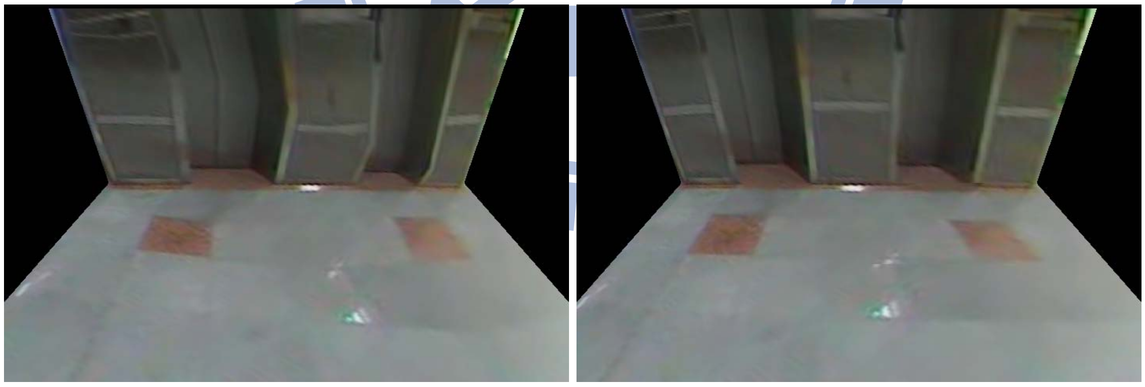


Figure 2.3: **The rendering result of the initial planar patches.** (a) Initial planar patches. (b) Rendering in opengl.



(a) (b)

Figure 2.4: **The comparison of rendering results between different number of triangles.** (a) Fewer triangles in a polygon. (b) More triangles in a polygon.

patch rendering. We use the following equation to estimate the distortion degree of the patch rendering:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{ij} - \tilde{I}_{ij})^2 \tag{2.7}$$

where I is the intensity of the point obtained from homography transformation, $\tilde{I}$ is the intensity of the point obtained from texture mapping, i and j are the coordinate of row and column, respectively, and $m \times n$ is the pixel number of the patch in the 2-D image. MSE (Mean Squared Error) represents the distortion degree of the rendering. In order to have an objective judgement on the distortion degree, we change MSE into PSNR (Peak Signal-to-Noise Ratio):

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{2.8}$$

$$PSNR \geq T \tag{2.9}$$

where MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, MAX_I is 255. Typical values for the PSNR are between 30 and 50 dB, where higher is better. Acceptable values for data loss are considered to be about 20 dB to 25 dB. We could set a threshold T to decide the quality of texture mapping. If the PSNR of the patch is lower than T, we divide it into smaller polygons and repeat the process for each polygon (Figure 2.5). Figure 2.6 shows the result of patch division in camera 3. It is noticeable that PSNR is a mathematical way to model the distortion. Although the value of PSNR is high, there might be some important places broken in the result of rendering. Therefore, we check the result of rendering by our eyes if necessary.

2.4 Lookup table construction

It is time-consuming to calculate mapping coordinates through homography matrices before texture mapping in each cycle. Instead of it, we build static background lookup tables for mapping the 3-D model to the images of video cameras. The image texture can be quickly mapping to 3-D model through static background lookup tables. In reverse

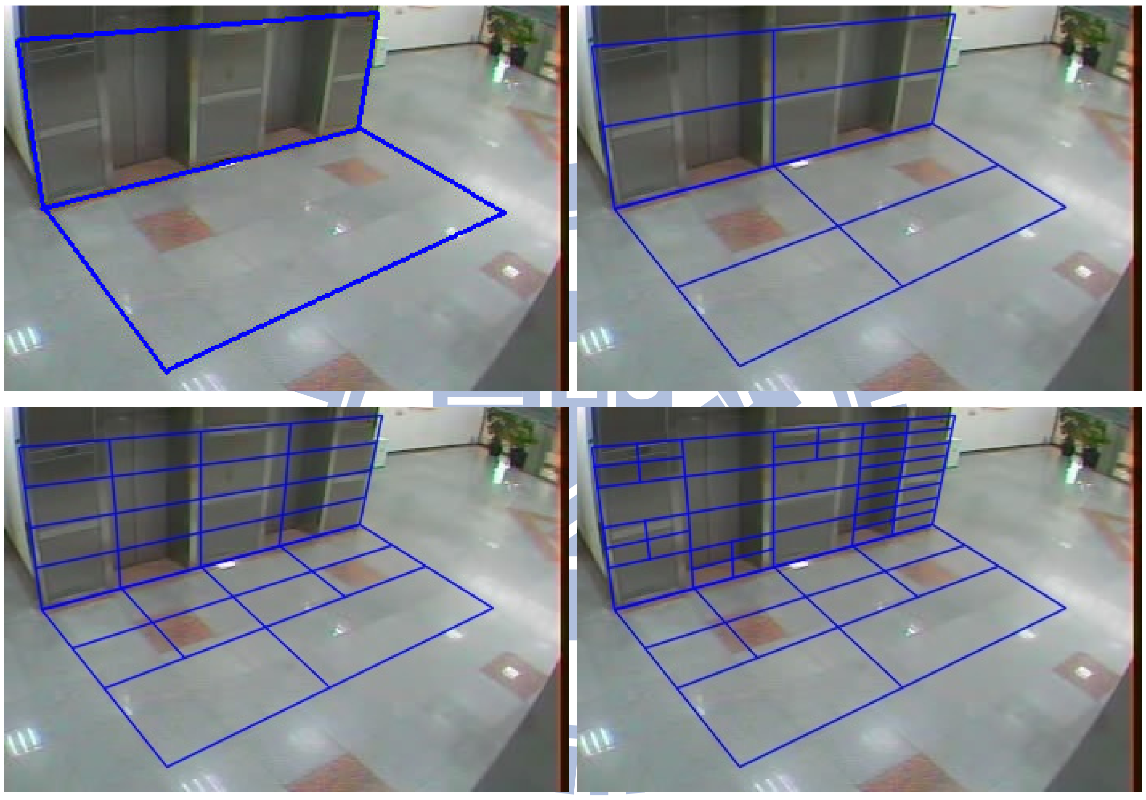


Figure 2.5: **The procedure of patch division.**

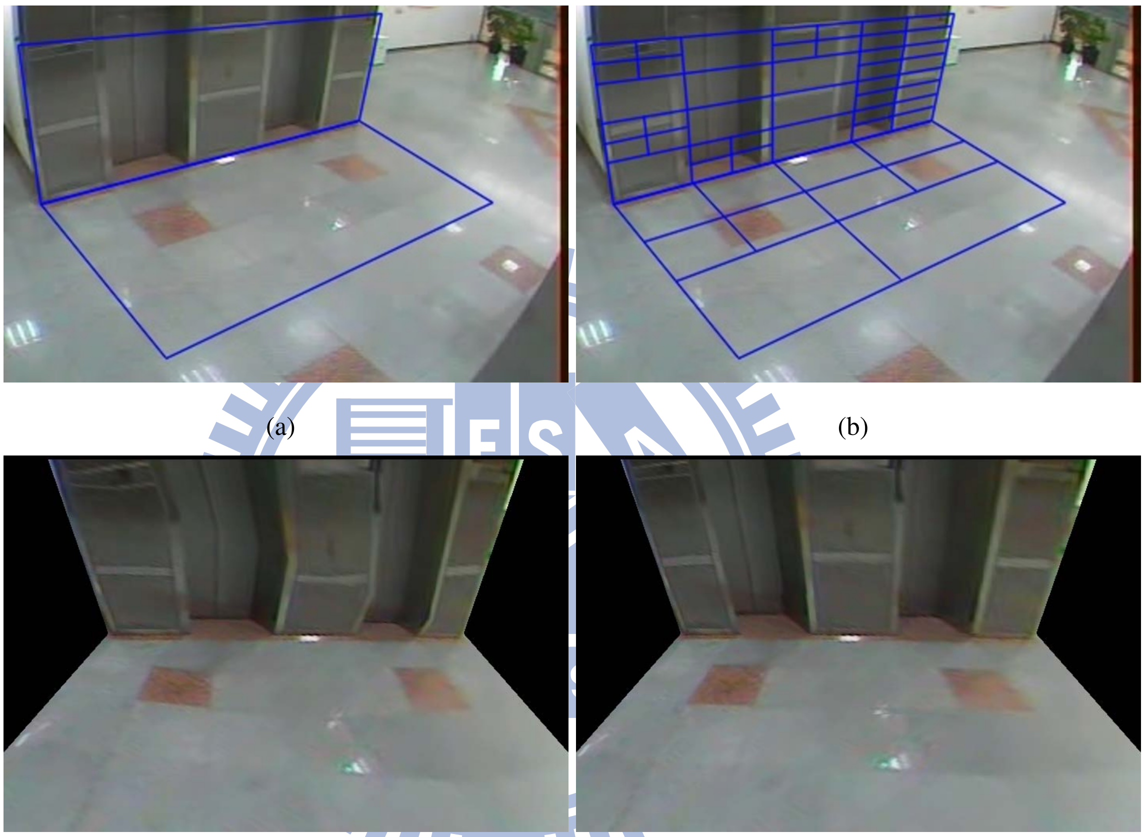


Figure 2.6: **The result of patch division.** (a) Initial patches without patch division. (b) The patches after patch division. (c) The rendering of initial patches. (d) The rendering of the patches after patch division.

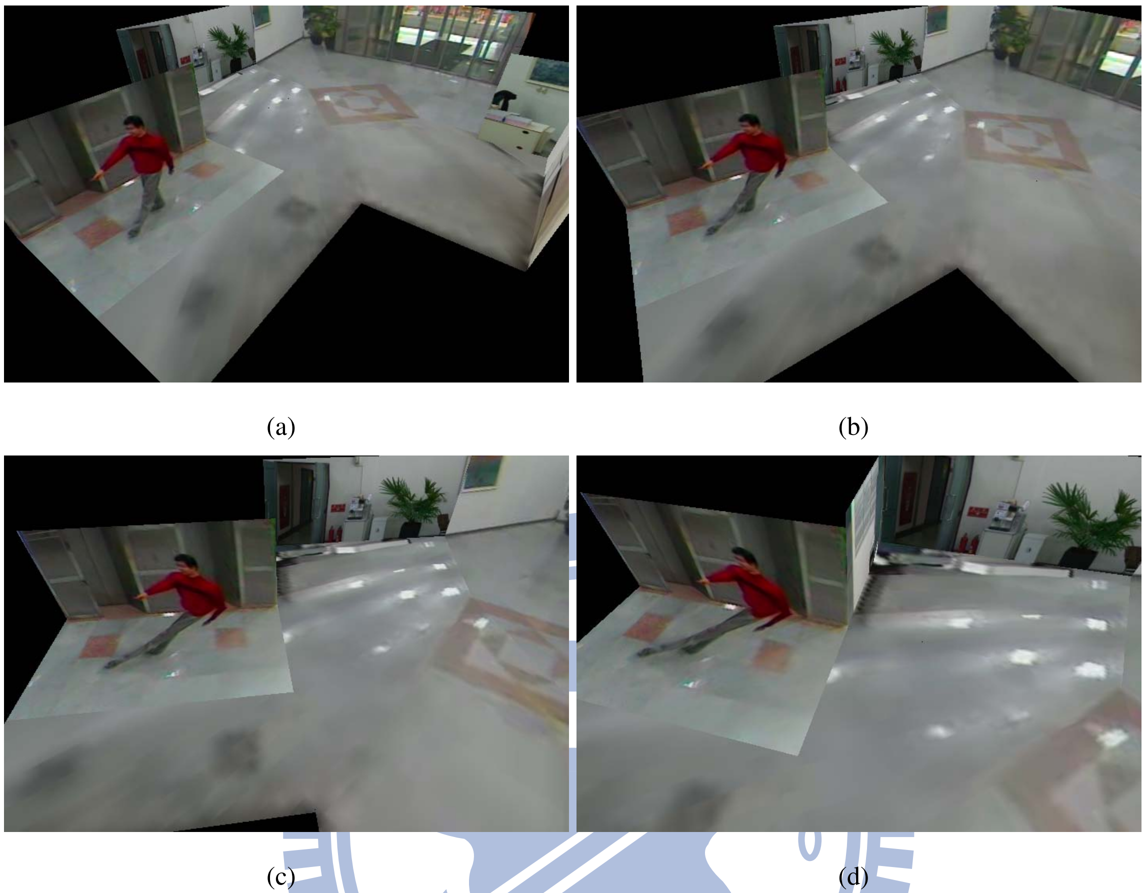


Figure 2.7: **The distortion of foreground objects.** (a) The viewpoint is appropriate to the projection axis of the camera, so the distortion of the person is little. (b)-(d) The viewpoint is far away from the projection axis of the camera, so the distortion of the person is large.

order, we build dynamic foreground lookup tables for mapping the 2-D images to the 3-D model. We can use dynamic foreground lookup tables to obtain locations of foreground objects in 3-D environment. The geometrical relationship of horizontal plane and vertical plane in the scenes is unchanged. The lookup tables just need to be built only one time since the cameras are fixed.

So far, we could display videos on the 3-D model through lookup tables. Since the foreground objects are modeled as planes, we find that foreground objects appear to be distorted (Figure 2.7). We will discuss how to visualize foreground objects in next chapter.

Chapter 3

Real-Time, Interactive, and Immersive Monitoring

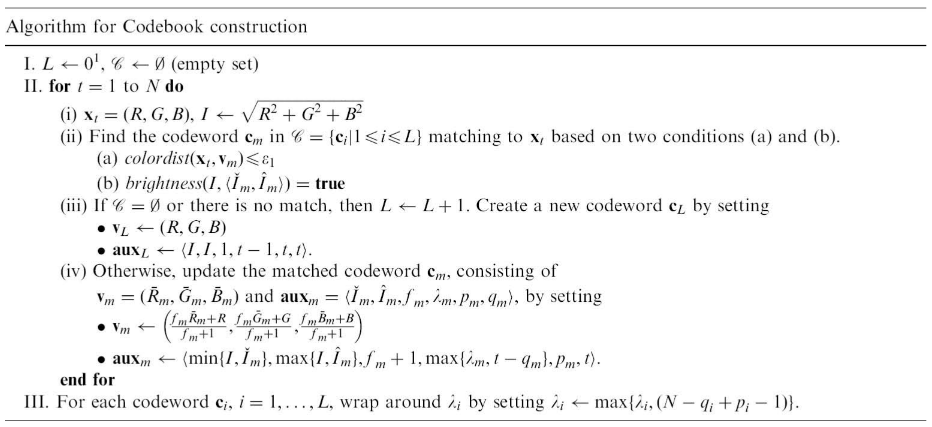
The lookup tables are built beforehand, and they are used to accelerate texture mapping. The images from cameras are mapping onto the 3-D model and the following frames are used as updating texture. Presently we can display the video on the 3-D model. However, the 3-D foreground objects, such as pedestrians, are not modeled before, so they will appear flattened on the floor or wall since they are also regarded as planes. Furthermore, there might be ghost effects in overlapping place of camera views. So we model foreground objects by axis-aligned billboarding to tackle the problem. In the following chapter, we present the proposed method in detail.

3.1 Background modeling

Dynamic objects extraction is a traditional problem in computer vision [11, 23]. There are many researches,such as MOG (Mixture of Gaussian), LBP (Local Binary Pattern) [7] and codebook algorithm [9], in this topic. In our system, effectiveness and real-time are the most important issues we care. Background subtraction is a simple foreground segmentation method but it often fails due to illumination changes, shadow, and reflections. Codebook algorithm is the method we implement which can reduce above problems.

Codebook algorithm builds a codebook consisting of one or more codewords for each pixel in the image. Samples at each pixel are clustered into the set of codewords based on a color distortion metric together with brightness bounds. Figure 3.1 is the construction of codebook.

Codebook algorithm takes a period of time to train its background model. It records the longest interval during the training period that the codeword has not recurred, so it is alright if there are moving objects in the scene during the training time. A pixel on the the tree was sampled to plot its intensity variation over time shown in Figure 3.2. The person who passed through the pixel is removed from the background model because he just appeared in a short time. Besides, the threshold of the decision function that adjusts the pixel is foreground or background is set due to the experimental environment. The pixel values change over time under lighting variation can be modeled as Figure 3.3. The foreground objects are detected by codebook algorithm in our experiments (Figure 3.4.



```
Algorithm for Codebook construction
I. $L \leftarrow 0^1$, $\mathscr{C} \leftarrow \emptyset$ (empty set)
II. for $t = 1$ to $N$ do
    (i) $\mathbf{x}_t = (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$
    (ii) Find the codeword $\mathbf{c}_m$ in $\mathscr{C} = \{\mathbf{c}_i | 1 \leqslant i \leqslant L\}$ matching to $\mathbf{x}_t$ based on two conditions (a) and (b).
        (a) $colordist(\mathbf{x}_t, \mathbf{v}_m) \leqslant \varepsilon_1$
        (b) $brightness(I, \langle \check{I}_m, \hat{I}_m \rangle) = \mathbf{true}$
    (iii) If $\mathscr{C} = \emptyset$ or there is no match, then $L \leftarrow L + 1$. Create a new codeword $\mathbf{c}_L$ by setting
        • $\mathbf{v}_L \leftarrow (R, G, B)$
        • $\mathbf{aux}_L \leftarrow \langle I, I, 1, t-1, t, t \rangle$.
    (iv) Otherwise, update the matched codeword $\mathbf{c}_m$, consisting of
        $\mathbf{v}_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and $\mathbf{aux}_m = \langle \check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m \rangle$, by setting
        • $\mathbf{v}_m \leftarrow \left( \frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m + G}{f_m + 1}, \frac{f_m \bar{B}_m + B}{f_m + 1} \right)$
        • $\mathbf{aux}_m \leftarrow \langle \min\{I, \check{I}_m\}, \max\{I, \hat{I}_m\}, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \rangle$.
    end for
III. For each codeword $\mathbf{c}_i$, $i = 1, \ldots, L$, wrap around $\lambda_i$ by setting $\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i - 1)\}$.
```

Figure 3.1: **The construction of codebook. (Figure source: [9])**

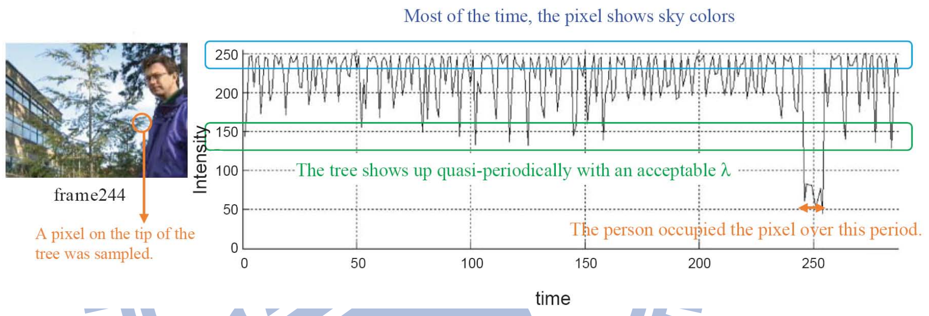


Figure 3.2: **Filtering the foreground objects in training time. (Figure source: [9])**

3.2 Blob segmentation and refinement

A morphological filter is applied to the binary map of the foreground region. After applying opening and closing operators, we reduce the noise and enhance the inner connection of the blob (Figure 3.5). We define the blob which is smaller than a threshold as noise and abandon it. The threshold is set according to the experimental environment.

Sometimes, two people walked too close make two blobs into one bigger blob and the locations of these blobs might have a little inaccuracy to their real locations in the 3-D model. In order to reduce the problem, we use vertical projection histogram to accumulate the foreground pixels and find the peaks and valleys in the bounding box. The region between two valleys with a peak corresponds to a single person. The peak must be above

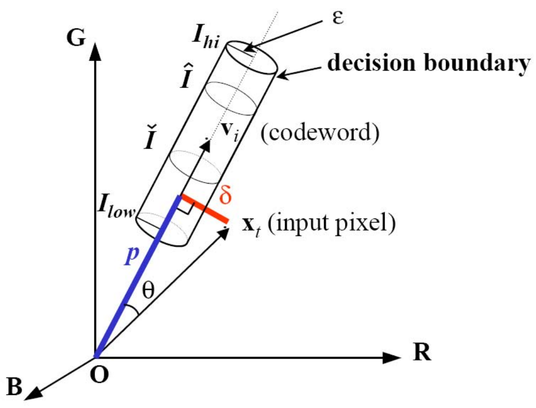


Figure 3.3: **The decision boundary of codebook. (Figure source: [9])**

the peak threshold ($\mathbf{P_T}$) and the valley must be lower than the valley threshold ($\mathbf{V_T}$). The threshold is set according to the experiments. In our experiment, the value of the peak threshold is selected as 70 percent of the height of the foreground region and the value of the valley threshold is selected as the mean of the histogram. By these peaks and valleys, we separate a bigger blob which has less connection between inner blobs into several smaller blobs (Figure 3.6).

Sometimes, the shadows or reflections might make the blob bigger than its real size. Canny edge detection is fast and effective to depict the edges of objects [2]. We apply it in the region of the blob to find the contours of the object detected by background modeling (Figure 3.7). After that, vertical projection histogram and horizontal projection histogram are used to find the tight limit of the bounding box (Figure 3.8). In most case, the boundaries of the polygons approximate the real size of objects through above method.

Although the background modeling and above segmentation methods are used to cut the foreground objects from the scene, there still might be some error of the real size of foreground objects. This makes bad visual effect, like image shaking, when we visualize foreground objects in 3-D model. Kalman filter is applied to smooth the change of the bounding box [22]. The location of the bounding box in 3-D model is according to the bottom-middle point of the bounding box in the 2-D image. The distance between bounding boxs in the current frame and previous frame is the feature for matching bounding boxs.

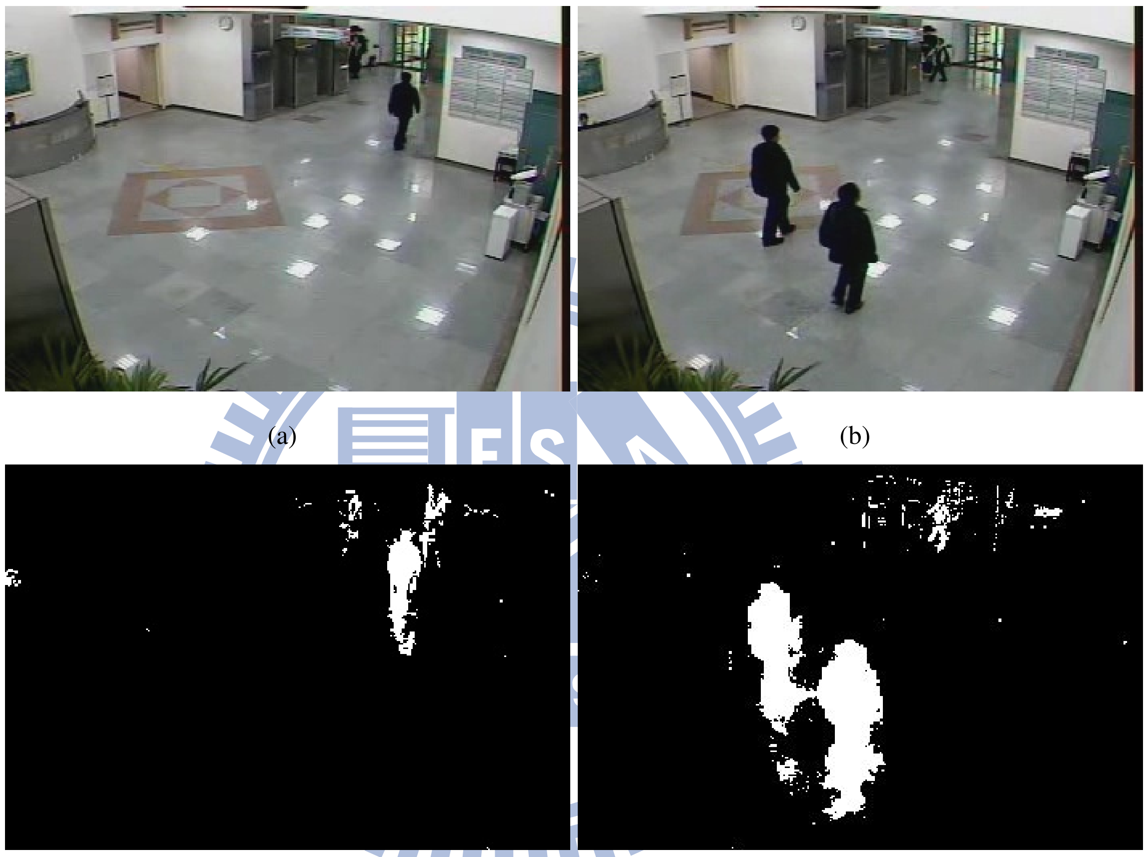


Figure 3.4: **The detection of foreground objects by codebook algorithm.** (a) A frame from the camera. (a) Another frame from the camera. (a) A foreground mask. (a) Another foreground mask.

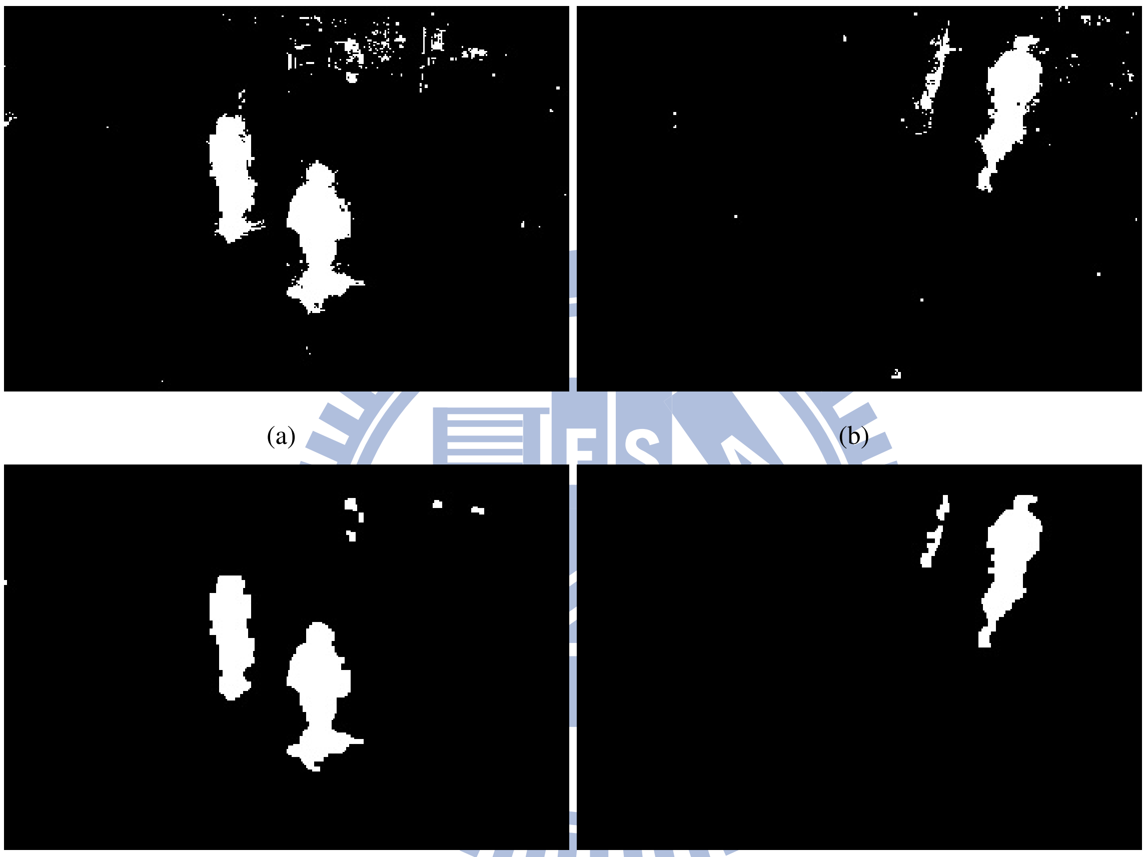
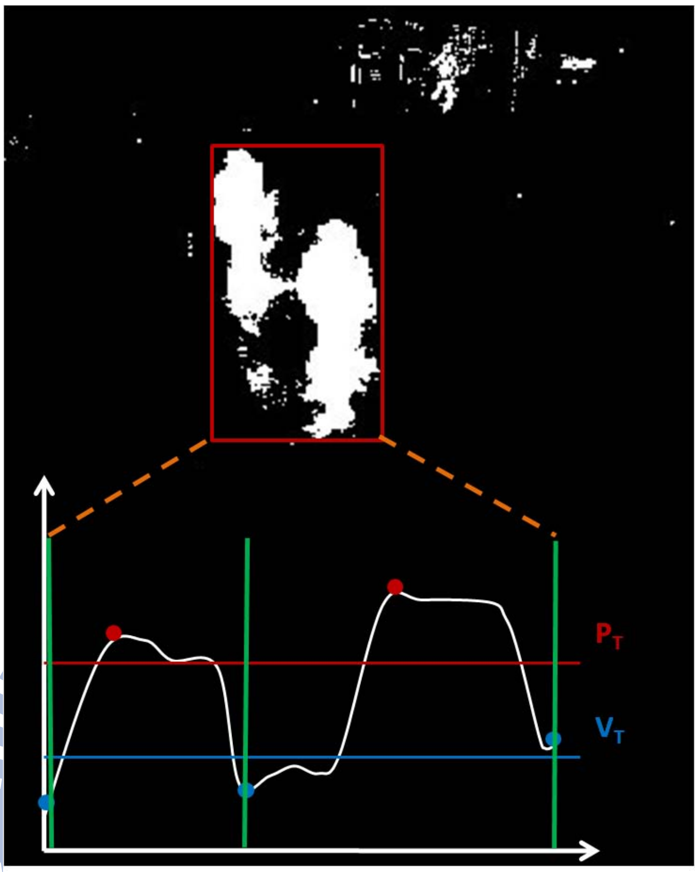
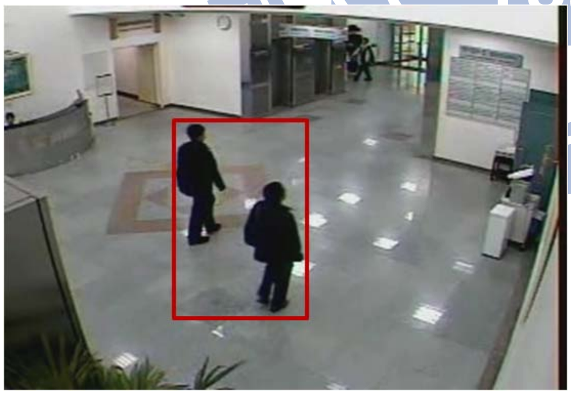

(a) (b) (c) (d)

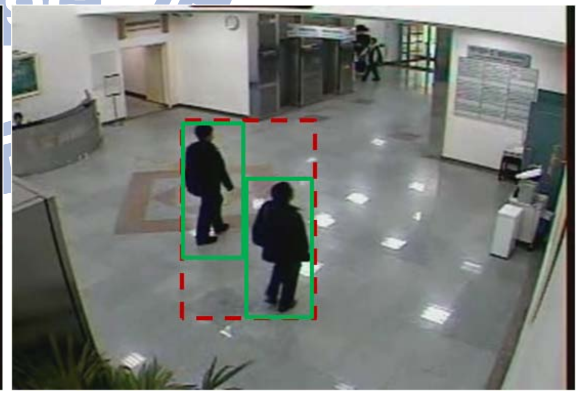
Figure 3.5: **The result after opening and closing.** (a) A foreground mask before opening and closing. (b) Another foreground mask before opening and closing. (c) A foreground mask after opening and closing. (d) Another foreground mask after opening and closing.



(a)

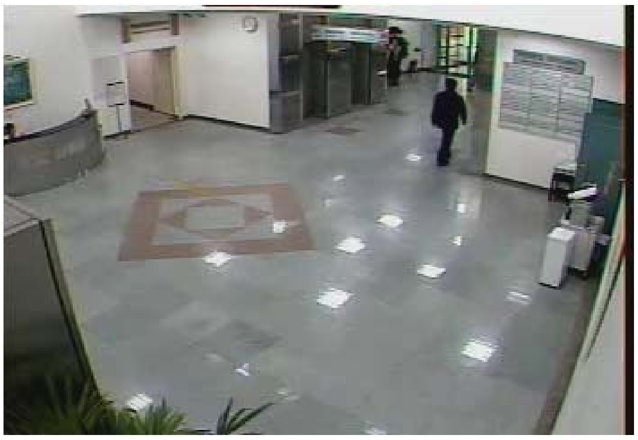


(b)



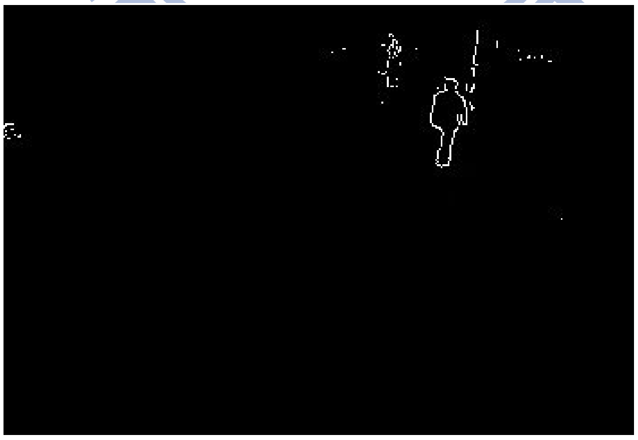
(c)

Figure 3.6: **Blob segmentation through vertical projection histogram.** (a) Vertical projection histogram is used to segment the blobs. (b) The red bounding box before segmentation besieges the two people. (c) A green bounding box after segmentation besiege only one person.



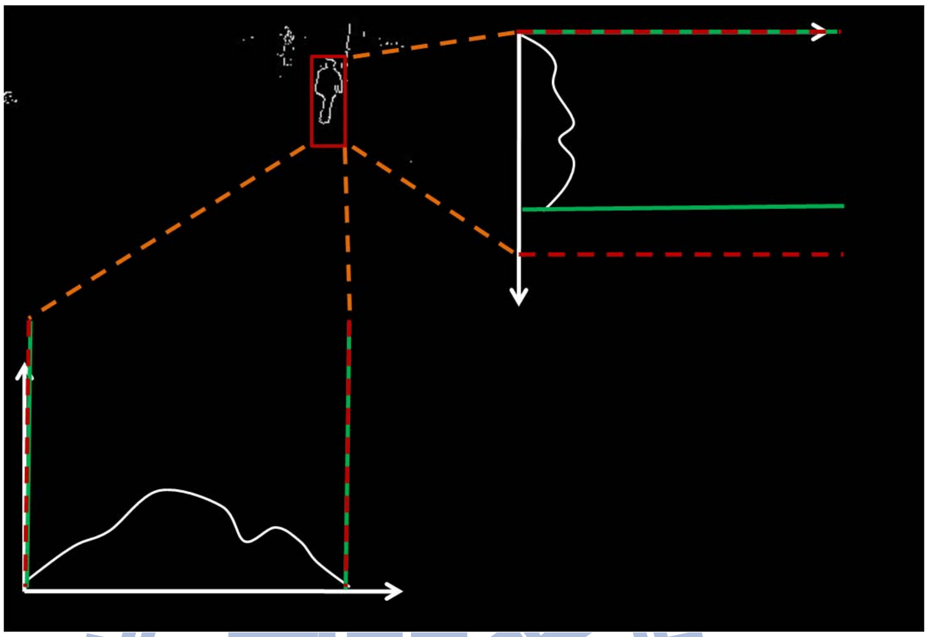
(a)

(b)

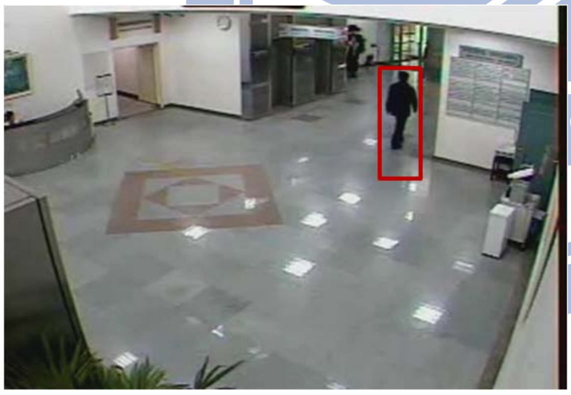


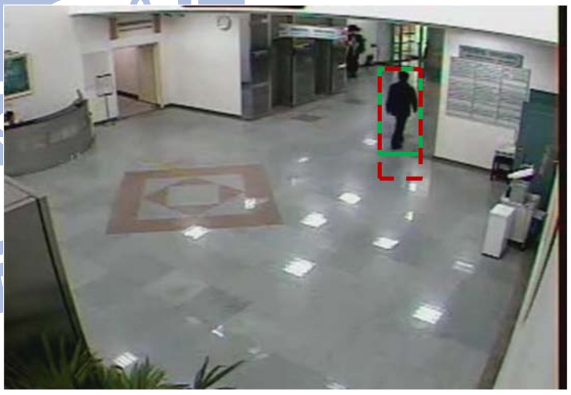
(c)

Figure 3.7: **Canny edge detection.** (a) A image from the camera. (b) The foreground mask from background modeling. (c) The contours of the foreground object.



(a)





(b) (c)

Figure 3.8: **Tight limit of the bounding box through vertical and horizontal projection histogram.** (a) Vertical and horizontal projection histogram are used to obtain the real size of the moving person. (b) The red bounding box besieges the person and his shadow. (c) The green bounding box is more fit for the person than the red bounding box.

We both put the top-left and bottom-right points of the bounding box into Kalman filter to makes the smooth change of height, width and location of the bounding box (Figure 3.9).

3.3 Axis-aligned billboarding

Axis-aligned billboarding is widely used in video game or 3-D mesh model to decrease the computing complexity [5]. When the 2-D image projects to the 3-D model, the vertical objects lie on the floor and are deformed in different viewpoints. Axis-aligned billboarding constructs billboards in the 3-D model for moving objects, like pedestrians, and the billboard always faces to the virtual camera without distortion. The texture of the blobs are mapping onto the billboard for 3-D visualization.

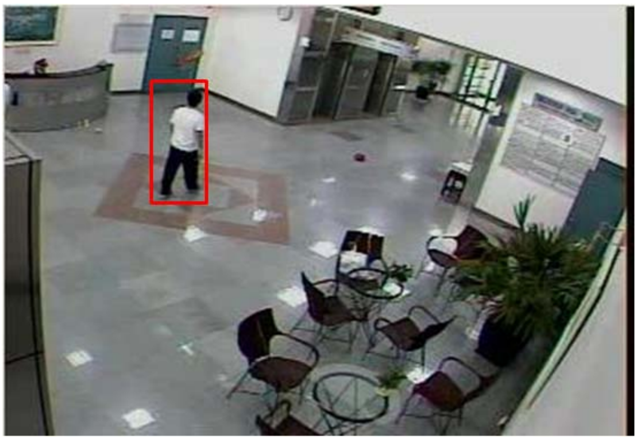
The billboard has three features: location, height and direction. Suppose all objects are always moving on the floor, so billboards are vertically put on the floor of the 3-D model. The location of the billboard in the 3-D model is estimated by mapping the bottom-middle point of the bounding box in the 2-D image through the lookup tables we built in previous chapter (Figure 3.10). The rate between the height of the bounding box and the 3-D model decides the height of the billboard in the 3-D model. The relationship between the direction of a billboard and the viewpoint is defined as Figure 3.11.

The following equations are used to calculate the rotation of the billboard:

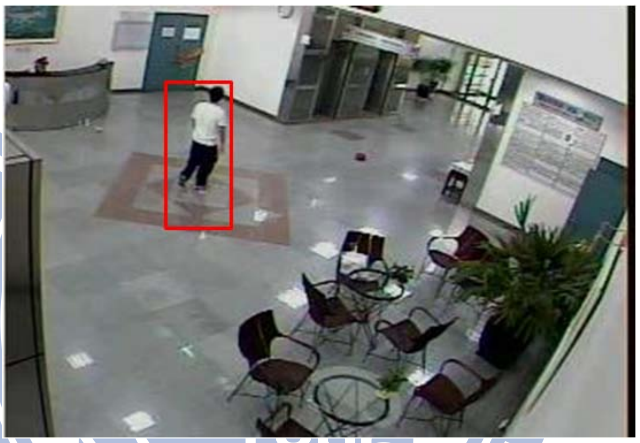
$$\mathbf{Y} = (\mathbf{N_B} \times \mathbf{V_{LE}}) \tag{3.1}$$

$$\Theta = \cos^{-1}(\mathbf{V_{LE}} \cdot \mathbf{N_B}) \tag{3.2}$$

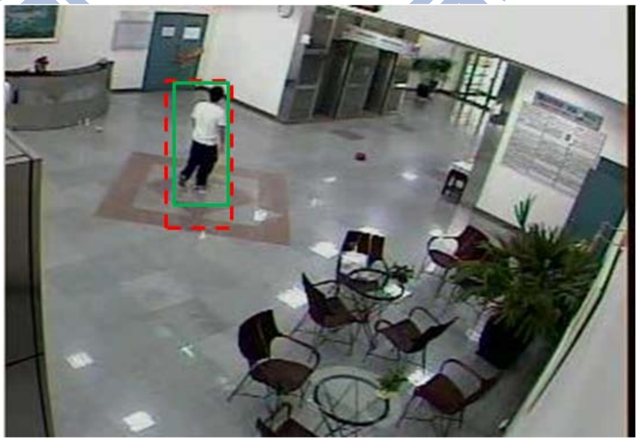
where $\mathbf{L}$ is the location of the billboard, $\mathbf{E}$ is the location that the viewpoint vertically project on the floor, and $\mathbf{V_{LE}}$ is the vector from $\mathbf{L}$ to $\mathbf{E}$. $\mathbf{N_B}$ is the normal vector of the billboard. The rotational axis is defined by $\mathbf{Y}$ and the rotational angle is estimated by Θ. The normal vector of the billboard is changed direction and the billboard is always facing the viewpoint of the operator. Now we can observe the foreground objects in any viewpoint.



(a)



(b)



(c)

Figure 3.9: **Smooth changing of the bounding box through kalman filter.** (a) The image of frame number 50. (b) The image of frame number 51. (c) The image of frame number 51. The red bounding box changes into the green bounding box by kalman filter and it causes stable detection of moving objects.

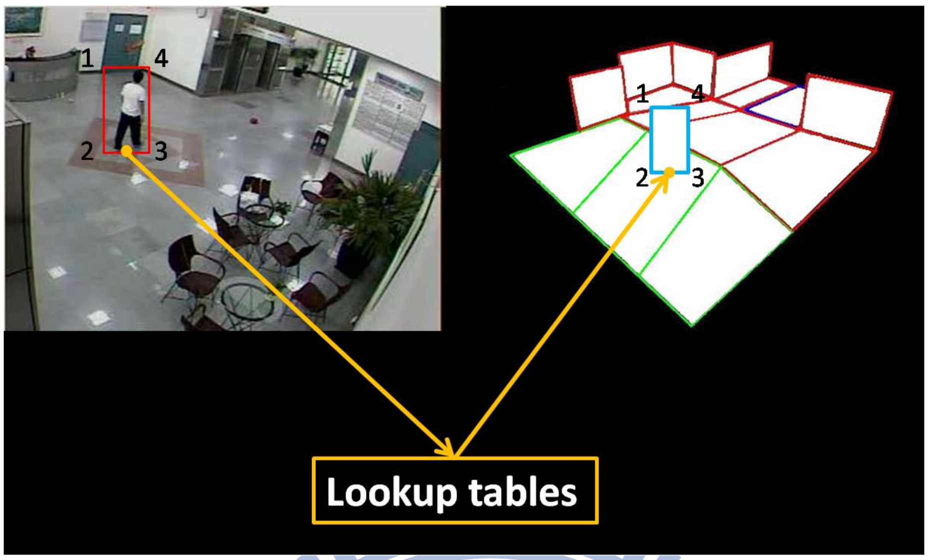


Figure 3.10: **Insert the billboard through lookup tables.**

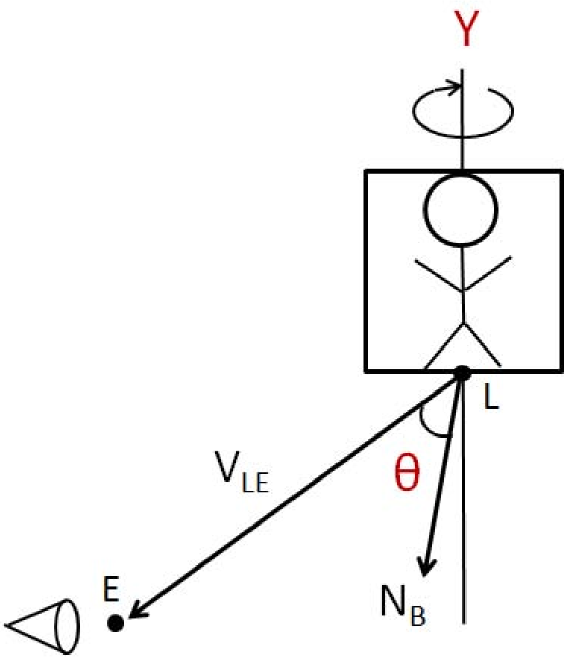


Figure 3.11: **The decision of the billboard direction.**

3.4 3-D model construction

If the field of view of the cameras overlap in the monitor area, a object such as a person could be seen in different camera views at the same time. If we visualize the person in different views simultaneously, this causes ghosting images. To handle the problem, we use 3-D locations of objects to identify the correspondence of objects in different views. When we decide our viewpoint in the model, the rotation angles of corresponding billboards are estimated by billboarding method. And we only visualize the billboard whose rotational angle is smallest in corresponding billboards.

Now we can build the 3-D surveillance system. First, the video contents from cameras are made as textures and we use lookup tables to map textures to the 3-D model. The billboards are also inserted into the 3-D model with proper locations, heights and directions. Then, the foreground objects are visualized by mapping textures with four corners of the bounding box in the 2-D image to four corresponding corners of the polygon in the 3-D model. We find that foreground objects still appear flattened on the ground or walls since we don't remove foreground objects while mapping the whole view to the 3-D model. To overcome the problem, static images without foreground objects from cameras are used to fill the foreground areas in cameras (Figure 3.12). These static images could be updated through the background modeling.

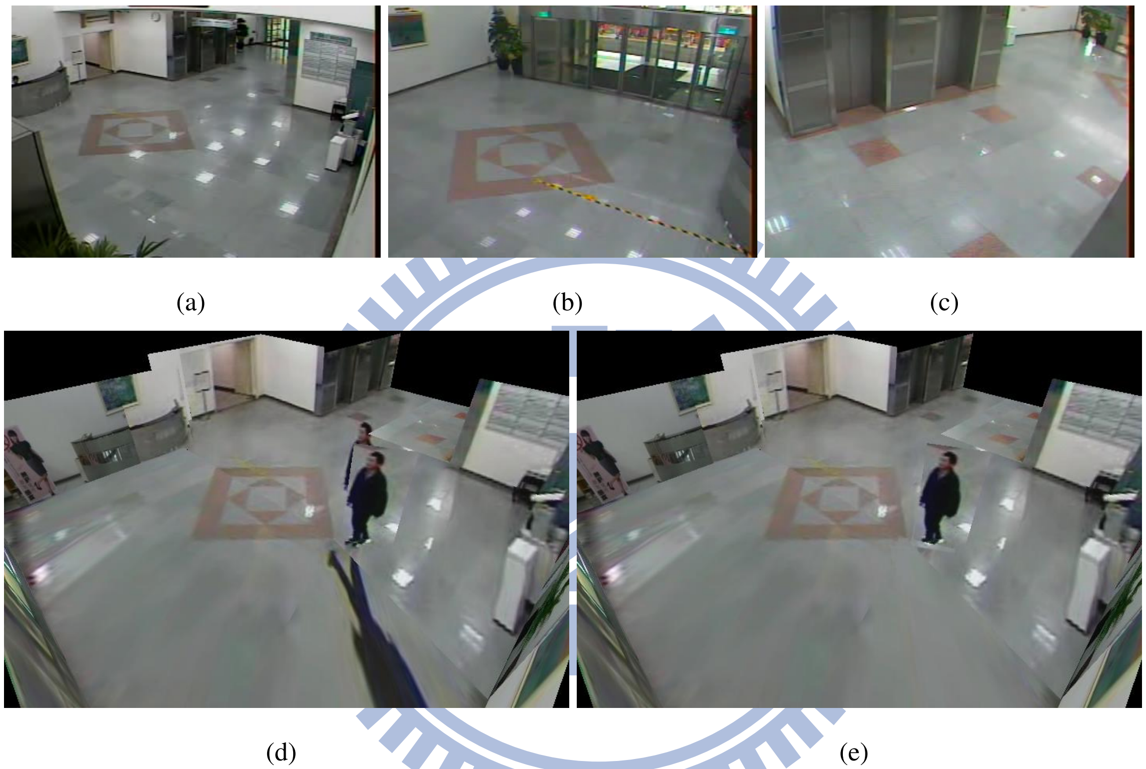


Figure 3.12: **The removal of ghosting effect.** (a) An image without foreground objects from camera 1. (b) An image without foreground objects from camera 2. (c) An image without foreground objects from camera 3. (d) Ghosting effect in overlapping area. (e) The removal of ghosting effect.

Chapter 4

Experimental Results

4.1 Camera configuration

The experimental environment for the proposed system is located in 52 department at Industrial Technology Research Institute in Taiwan. Three IP cameras are mounted in the lobby of the building in our experiments (Figure 4.1). Figure 4.2 shows the view direction and field of view of cameras with the floor plan of the 52 department. Image views from three IP cameras are shown in Figure 4.3.

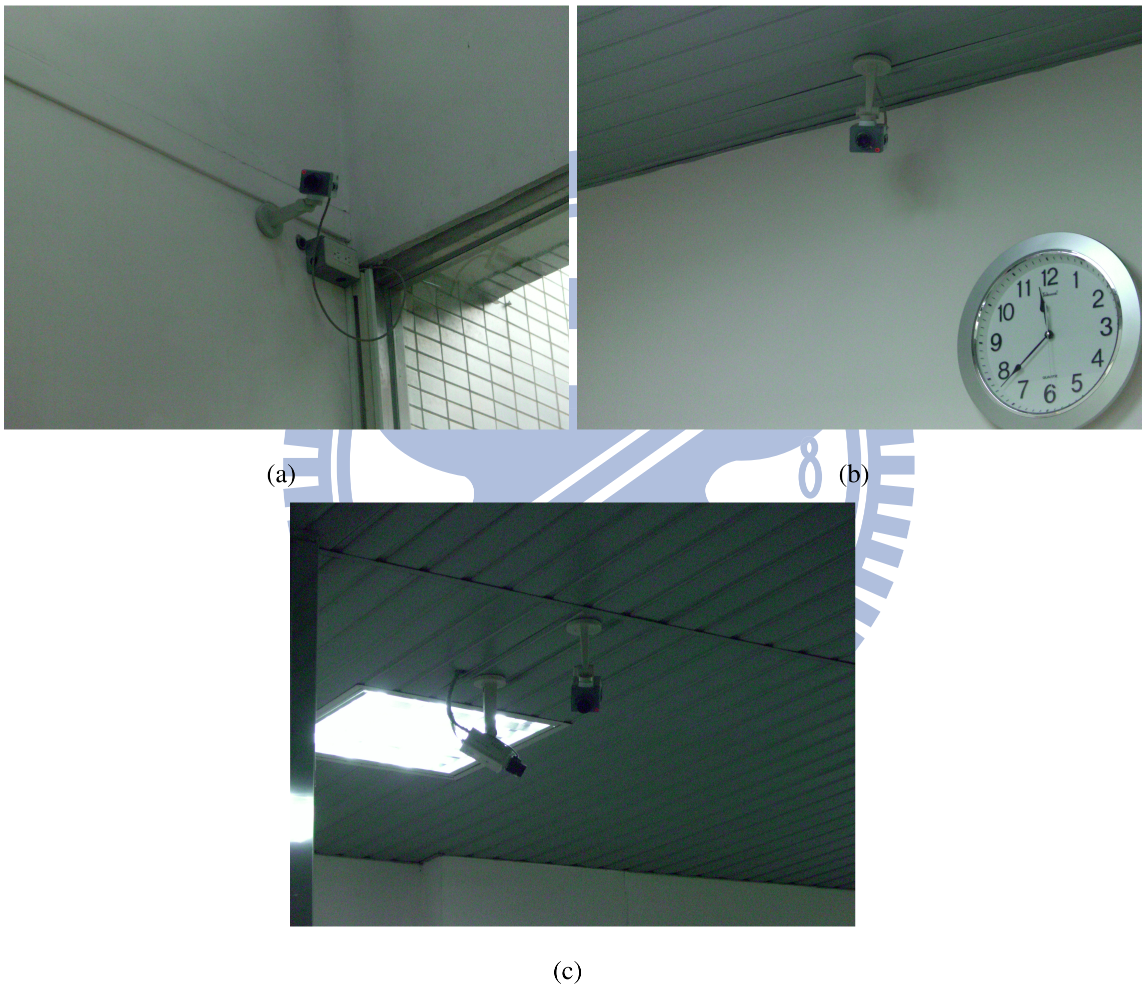
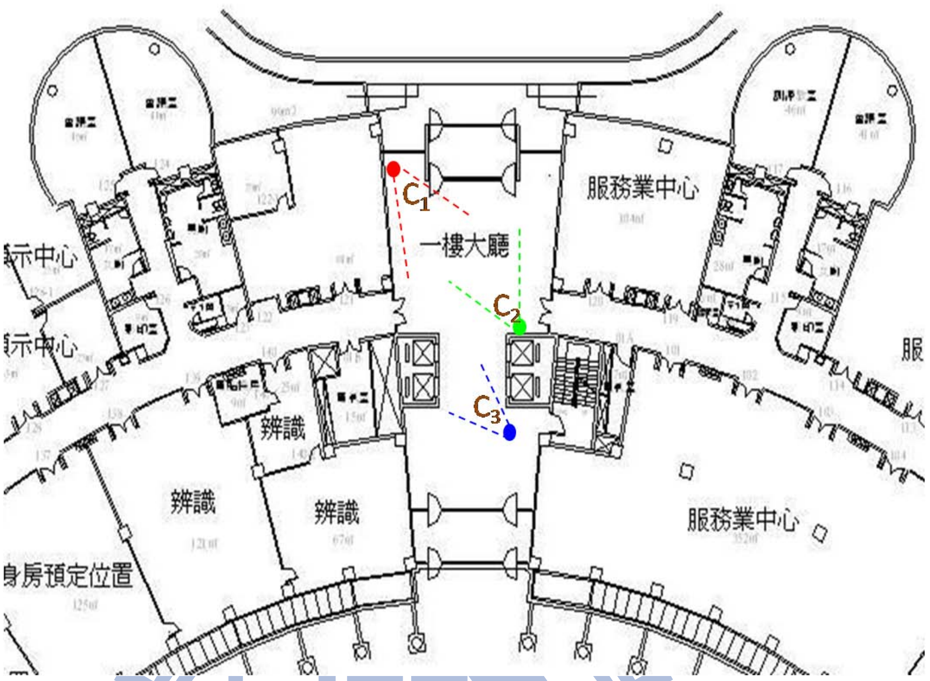
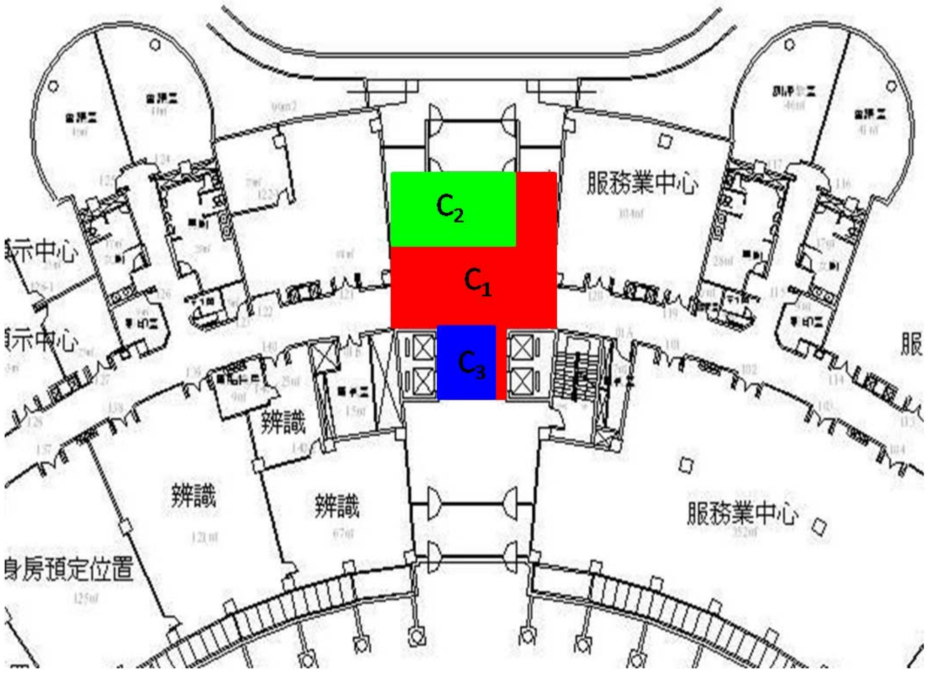


Figure 4.1: **Three IP cameras mounted in the lobby.** (a) Camera 1. (b) Camera 2. (c) Camera 3.

The following procedure is how we construct a 3-D model for the monitored area and map image contexts to the 3-D model. First, we observe the structure of each scene from

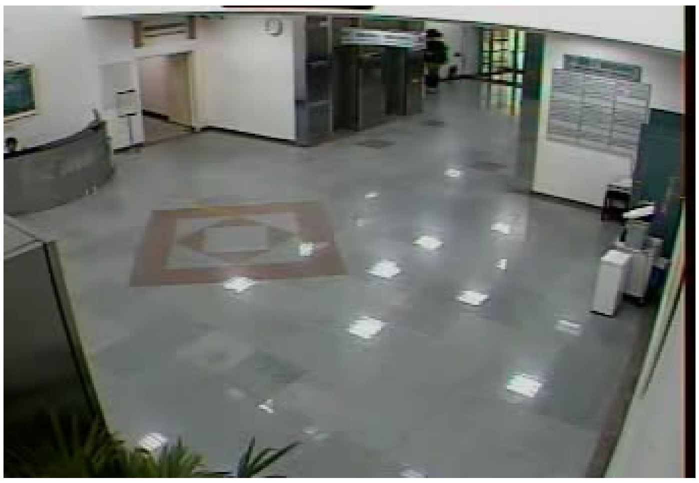


(a)

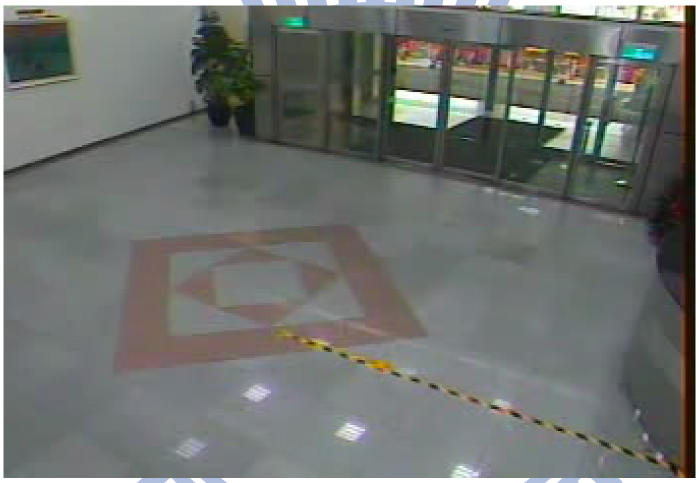


(b)

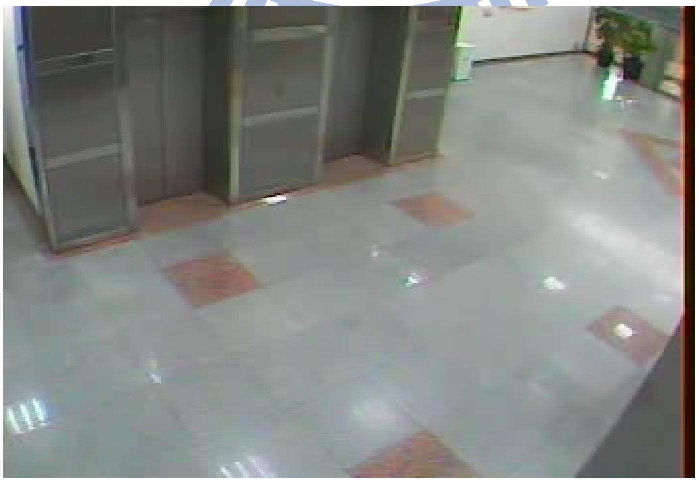
Figure 4.2: **The floor plan of the experimental environment.** (a) The view direction of the cameras. (b) The field of view of the cameras.



(a)



(b)



(c)

Figure 4.3: **Images from three IP cameras.** (a) An image from camera 1. (b) An image from camera 2. (c) An image from camera 3.

cameras. If the structure of the scene is complex, we use more patches to model the scene; otherwise, fewer patches are used to build the environment. Basically, it is sufficient to model a vertical plane, like a wall, by a polygon. In our experiments, the floor of the 3-D model is composed of different images from cameras. In order to make sure of better seams between different views, we use more patches to model the floor. Second, a benchmark (the yellow point in Figure 4.4 (d)) is set in the real world and we measure the relative positions of landmarks to benchmarks for construction of the 3-D model (Figure 4.4). Figure 4.5 shows the scenes which are modeled by planar patch modeling and the corresponding 3-D model, respectively. Finally, the patches are divided automatically into smaller polygons based on the our request for the quality of rendering (Figure 4.6). The lookup tables accelerate the texture mapping and the rendering result of the monitored area is shown as Figure 4.7.

4.2 System setup and refinement

4.2.1 Predefined viewpoints and smooth viewpoint switching

In our experiment, there are three cameras for the surveillance and we set three predefined viewpoints which are approximate to real locations and directions of cameras. When users want to observe someone from one of cameras, they can quickly move to the specific viewpoint. When one viewpoint is translated to another, we apply interpolation to estimate the parameters of the viewpoint. Figure 4.8 shows the smooth switching of the viewpoint.

4.2.2 Static background images patches

Owing to the limit of the field of view of the cameras, there are some ”black areas” in the 3-D model. It means that there are no image contents from cameras mapping onto these areas. In order to obtain the better visual effects, we first take static photos of ”black areas” by a digital still camera (Figure 4.9). Then, we insert the corresponding polygons into the 3-D model for mapping the static photos. Figure 4.10 shows a more complete view of the monitored area.

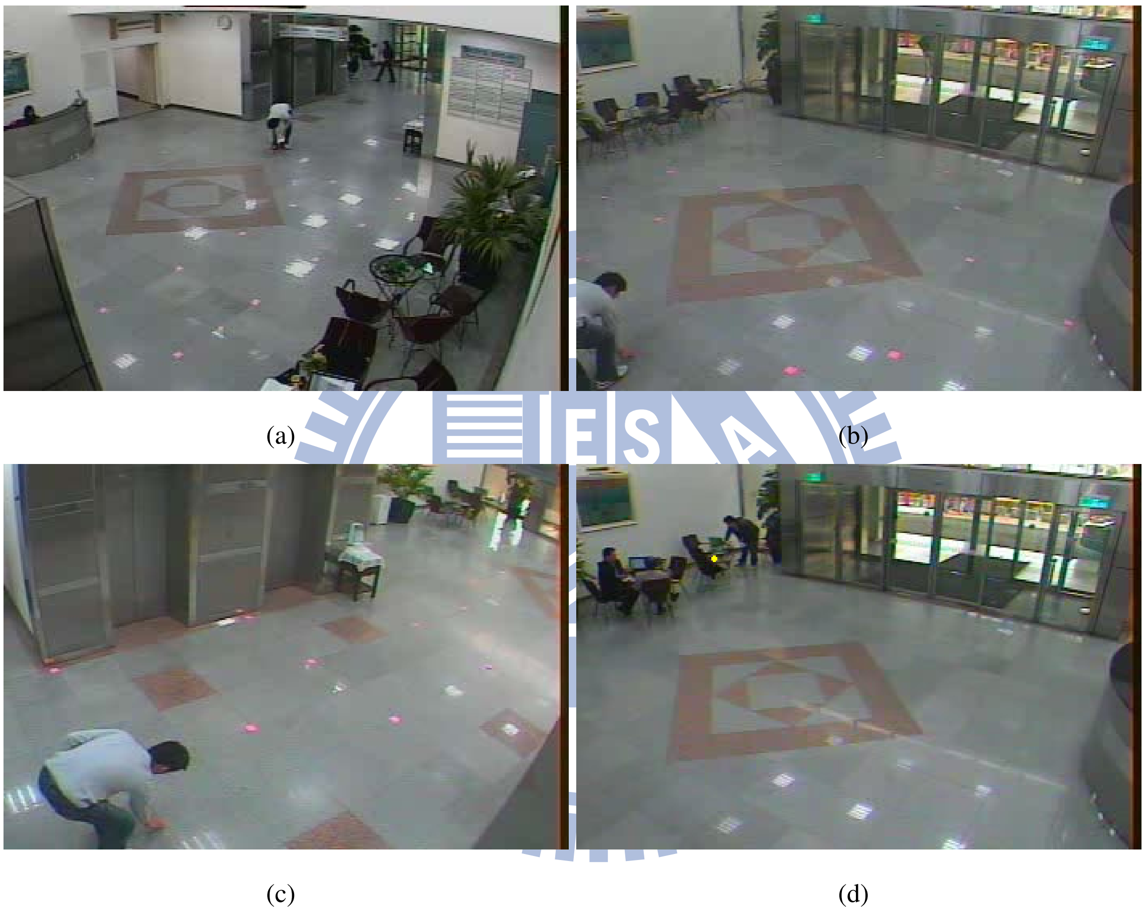
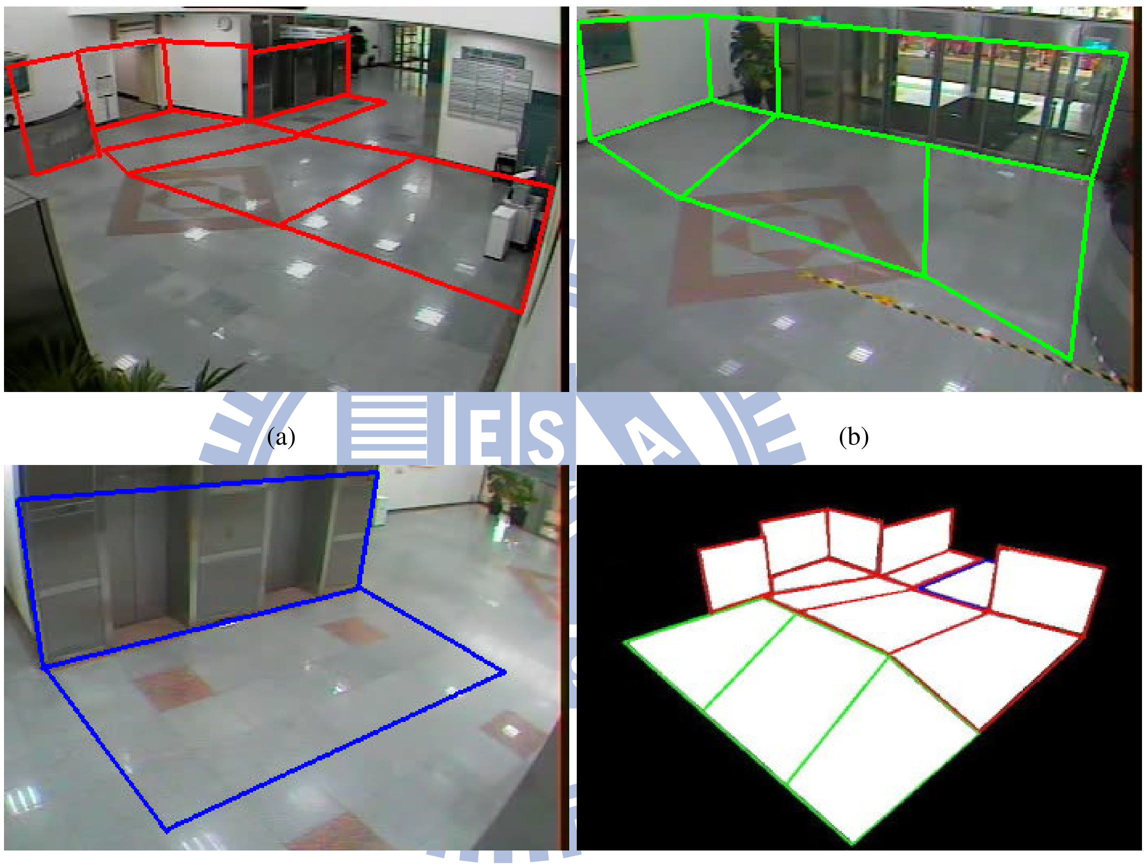


Figure 4.4: **Landmarks in scenes.** (a) Landmarks in scene 1. (b) Landmarks in scene 2. (c) Landmarks in scene 3. (d) The yellow point is the benchmark for in real world.



(a) (b) (c) (d)

Figure 4.5: **The results of planar patch modeling.** (a) The patches of the scene 1. (b) The patches of the scene 2. (c) The patches of the scene 3. (d) The patches of the 3-D model.

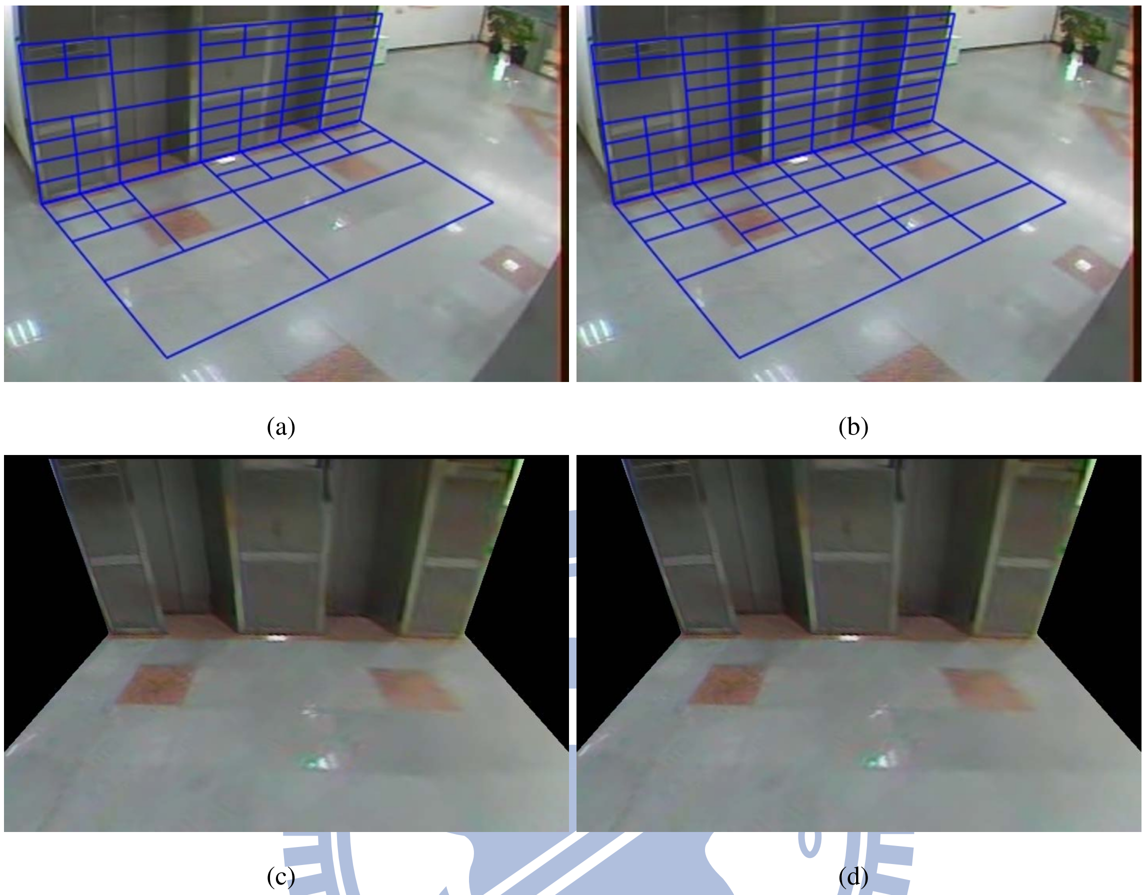


Figure 4.6: **The results of automatic patch division.** (a) Patch division in PSNR 25. (b) Patch division in PSNR 30. (c) The rendering result in PSNR 25. (d) The rendering result in PSNR 30.

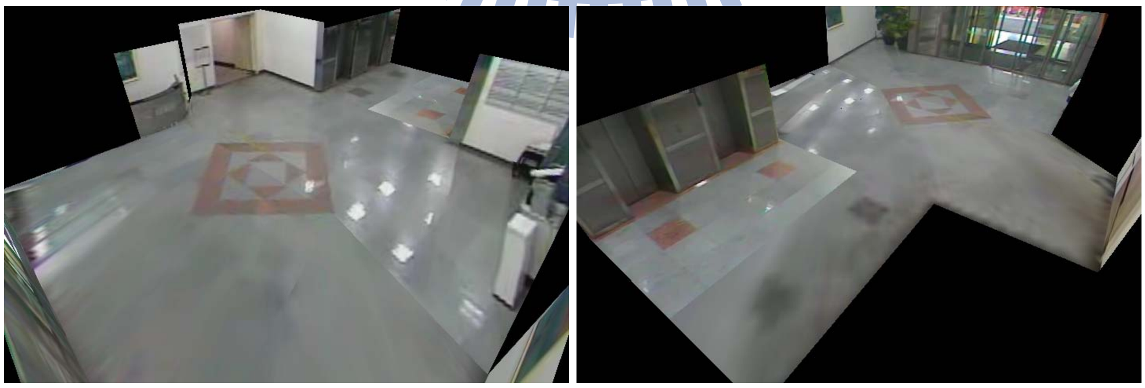


Figure 4.7: **The rendering of the 3-D environment.**

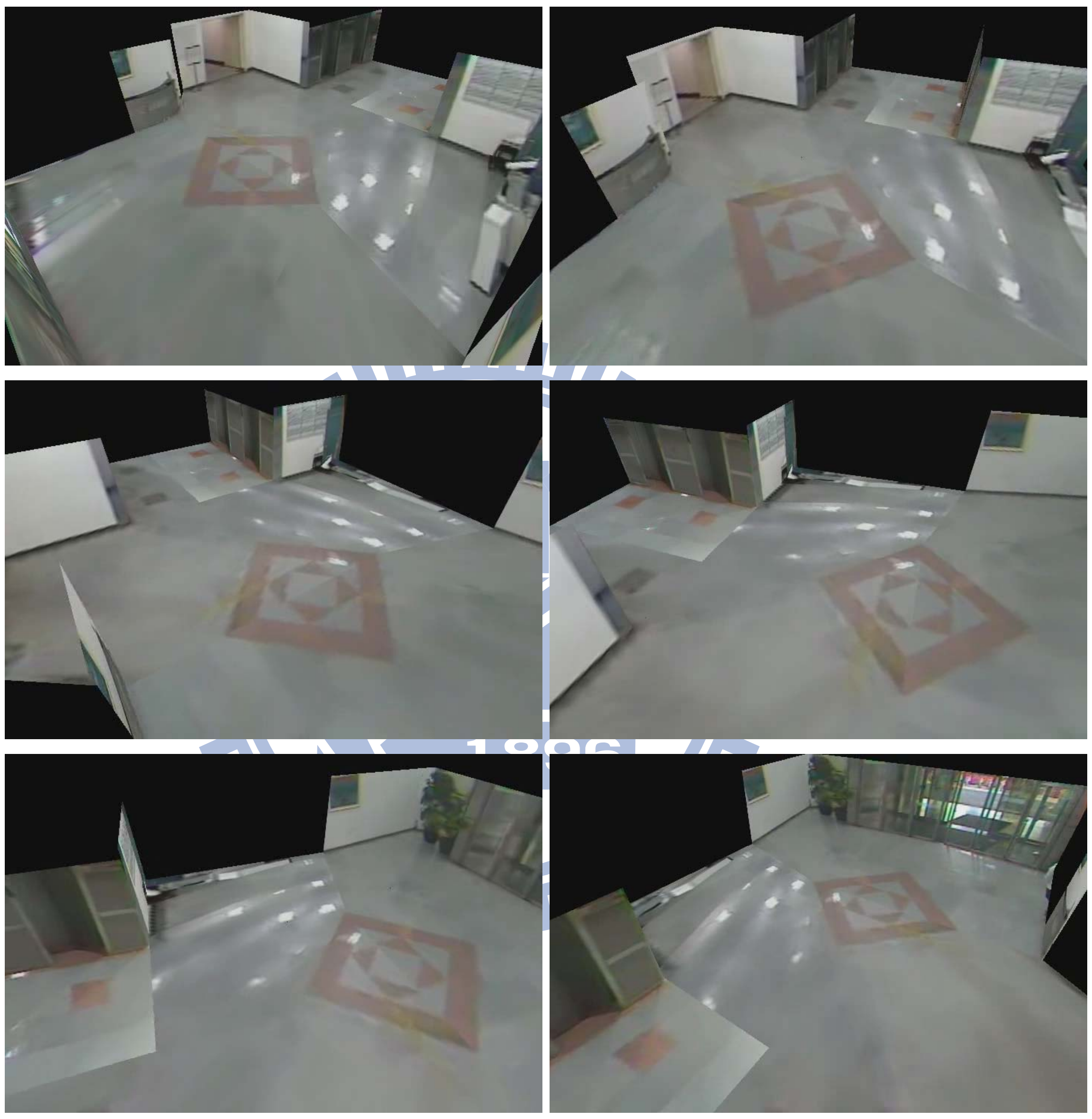
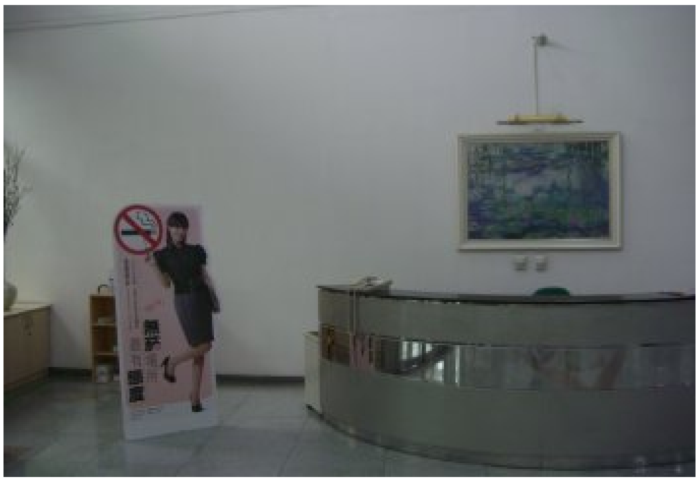
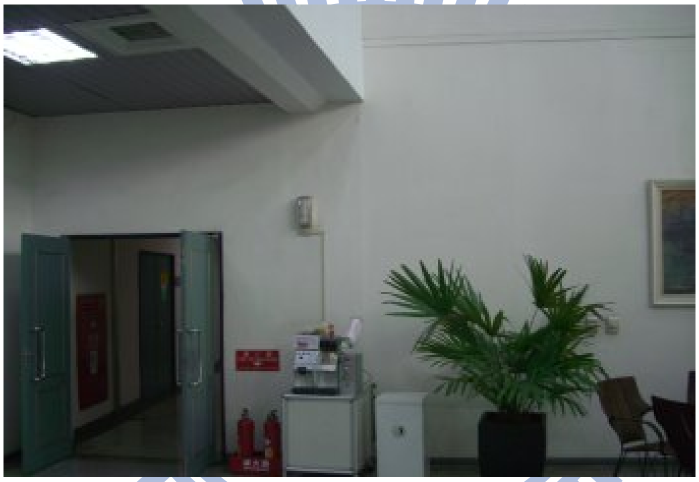


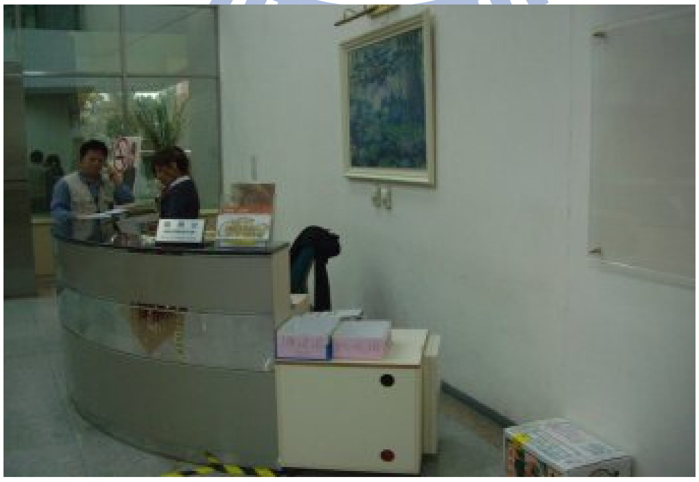
Figure 4.8: **The results of smooth viewpoint switching.**



(a)



(b)



(c)

Figure 4.9: **The static photos from a digital still camera.**

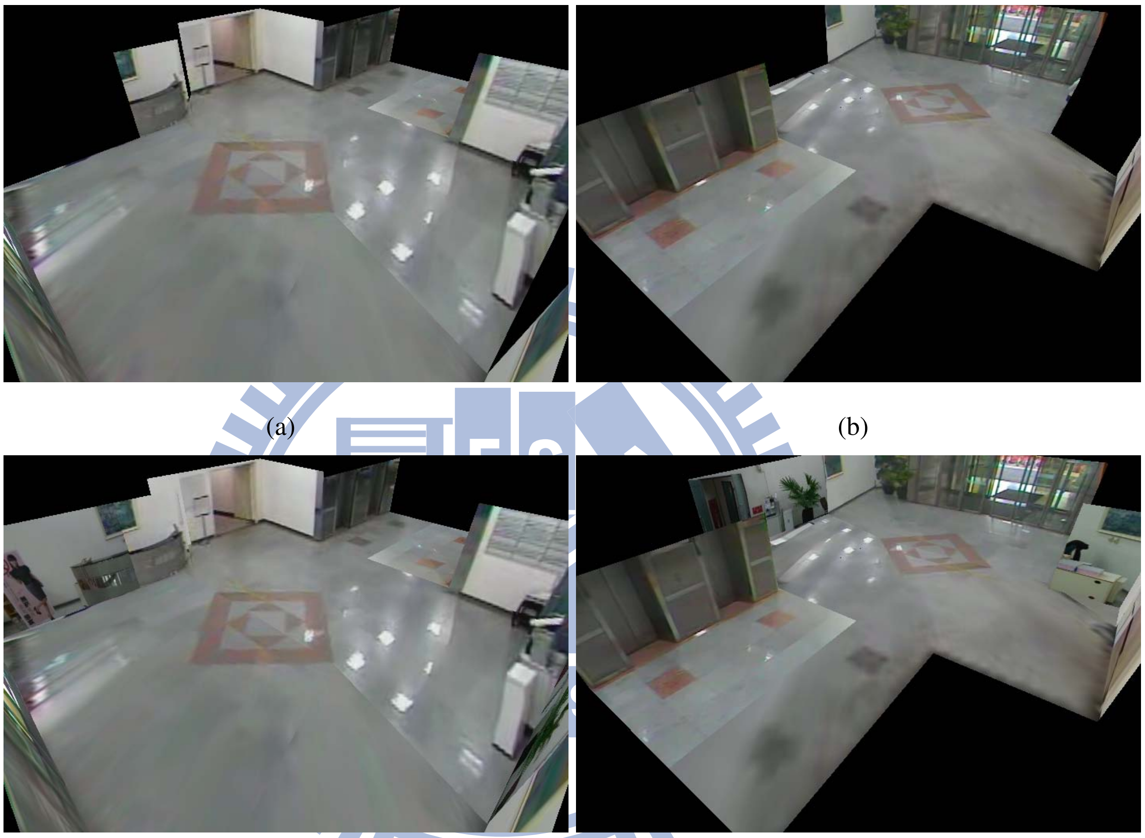


Figure 4.10: **The 3-D model with static background images patches.** (a) One view without static background images patches. (b) Another view without static background images patches. (c) One view with static background images patches. (d) Another view with static background images patches.

4.2.3 Removal of dynamic background objects

In our monitored area, there are some dynamic background objects, such as electric doors and elevator doors, will be detected by background modeling while they are opened or closed. Since the dynamic background objects have been modeled in background construction, we don't want to model them again by billboards. If we do so, monitors could be confused and can't focus on other moving objects. The ground mask is a mask setting the region of interesting that cuts floor areas from the environment and it is helpful to filter the dynamic background objects. If the blob locates in floor areas, we think it is a moving foreground object and keep it; otherwise, the blob is a dynamic background object and we abandon it (Figure 4.11).

4.3 Results of the 3-D surveillance system

4.3.1 Immersive monitoring

When security guards find something wrong or unusual in monitored area, they might want to zoom in the view to obtain the details of the object. If there is a PTZ (Pan-Tile-Zoom) camera in monitored area, security guards can operate it to enlarge the view. Instead of a real PTZ camera, we simulate the function of PTZ by moving the viewpoint in the 3-D model. Security guards can observe the whole monitored area in any viewpoint. The results are shown in Figure 4.12.

4.3.2 Transparent billboard

In above results, the foreground objects visualized by billboarding. We can find that there are some unnecessary background textures mapping on the billboard. In order to remove unnecessary background textures, we change the color model from RGB to RGBA. The alpha value affects the degree of transparency. The foreground mask from background modeling is applied to set the alpha value of textures. Therefore, the foreground objects appear with much fitting shapes. The results are shown in Figure 4.13. Under the premise of accurate detection for foreground objects, the transparent billboards show the better

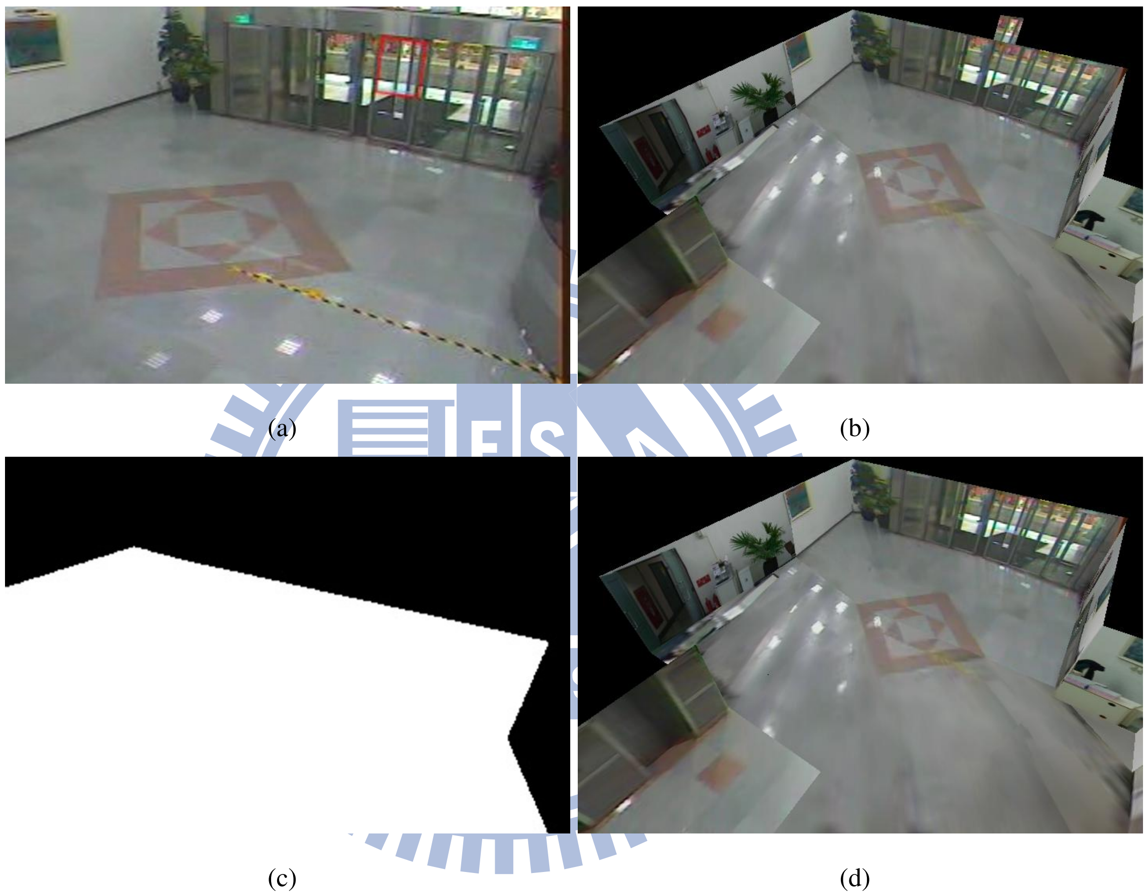


Figure 4.11: **Dynamic background objects removing.** (a) The electric door is detected by background modeling. (b) The unnecessary billboard of the electric door. (c) The ground mask of camera 2. (d) The billboard of the electric door has been removed.

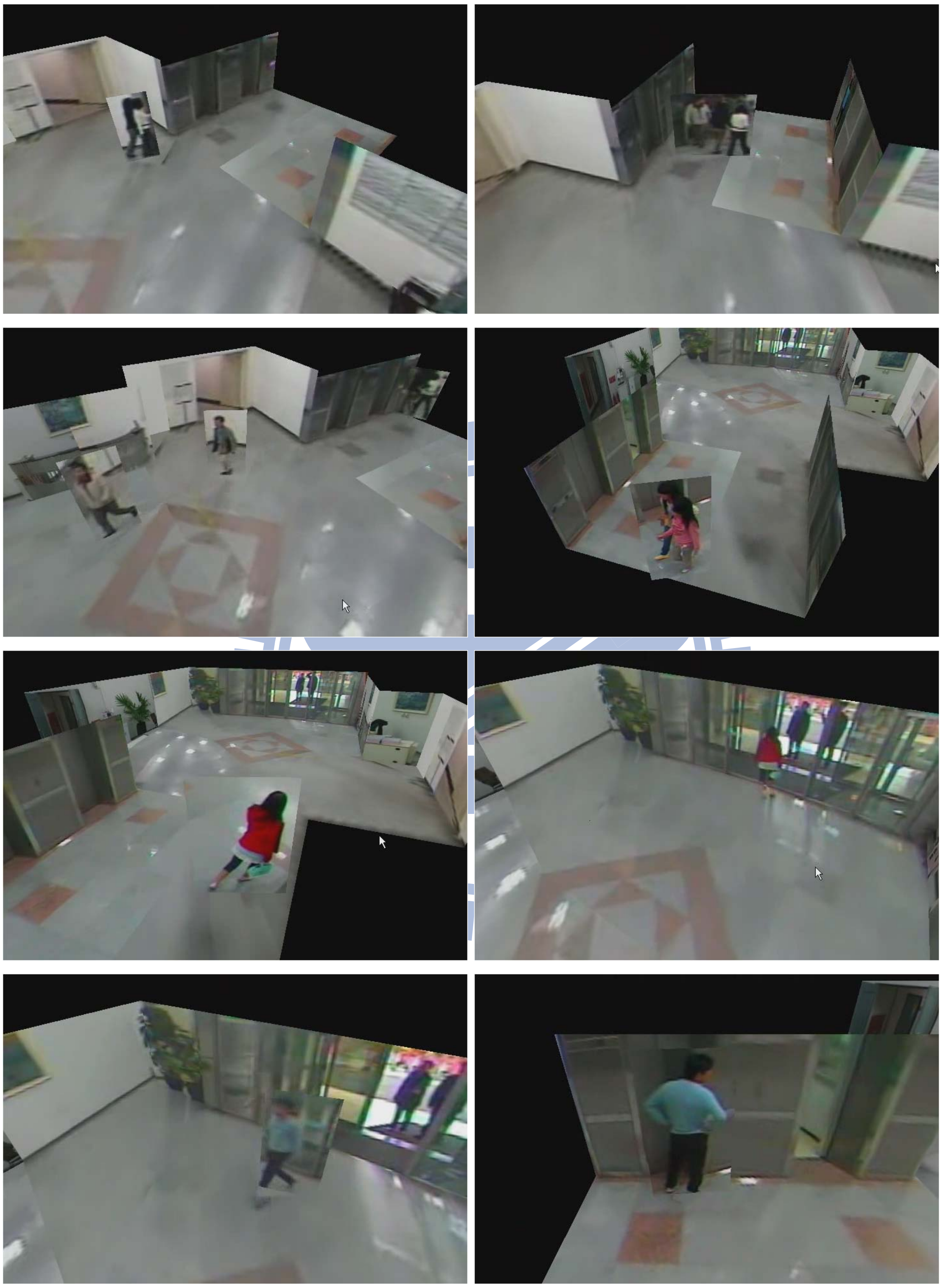


Figure 4.12: **Results of the immersive monitoring.** We can observe pedestrians with any viewpoint in the monitored area. The billboards are always facing to the viewpoint.

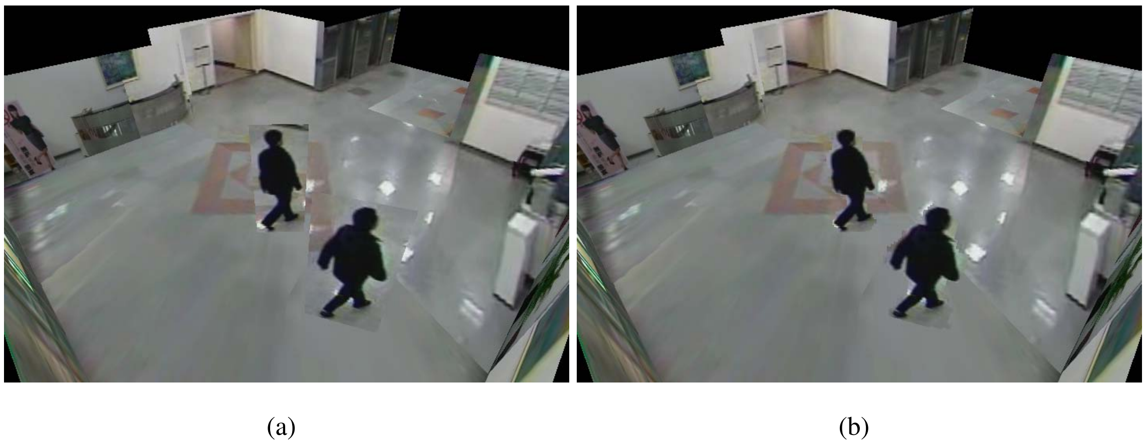


Figure 4.13: **The result of the transparent billboard.** (a) Using the normal billboard. (b) Using the transparent billboard.

appearance of foreground objects. Otherwise, the foreground objects are displayed with broken shapes.

4.3.3 Auto-tracking of targets

We implement the function of auto-tracking of targets by switching the viewpoint. The switching criterion is defined as comparing the number of blobs in specific areas. First, we divide the floor area into several parts and distribute them to each camera (Figure 4.14). When people move into the scene, the virtual camera auto-switch to the predefined viewpoint of the area which has more people in it (Figure 4.15). In order to avoid over frequently switching, there is a cool time between every switching. In this case, we also make the billboard transparent by setting the alpha value of texture.

4.4 System performance

The proposed system are implemented in C/C++ language, compiled by Microsoft Visual Studio 2005 compiler and executable on WindowsXP platform. Opengl and opencv are also applied to our system [1]. The system performance is estimated by timer counting the total execution time of processing a large number of frames from three IP cameras. We

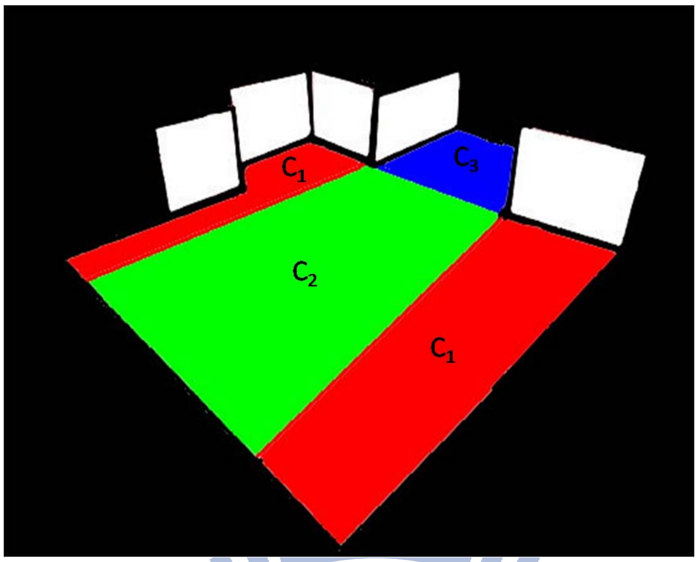


Figure 4.14: **The scope of each camera.** We divide the floor area depending on the field of view of the cameras. Count the number of people in each part for auto-switching the virtual camera.

construct the proposed surveillance system on the PC with Intel Core Quad Q9550 processor, 2G bytes ram, and nVidia GeForce 9800GT graphic card. Three IP cameras with 352 $\times$ 240 resolution are connected to the PC by Internet. When the system is in the situation of the low loading, like fewer foreground objects in the scene, the frame rate of the system is about 30 frames per second. If there are more foreground objects in the scene, the frame rate of the system is about 20 frames per second. Some programming skills, such as multi-threading and GPU programming, can further improve the performance of the surveillance system.

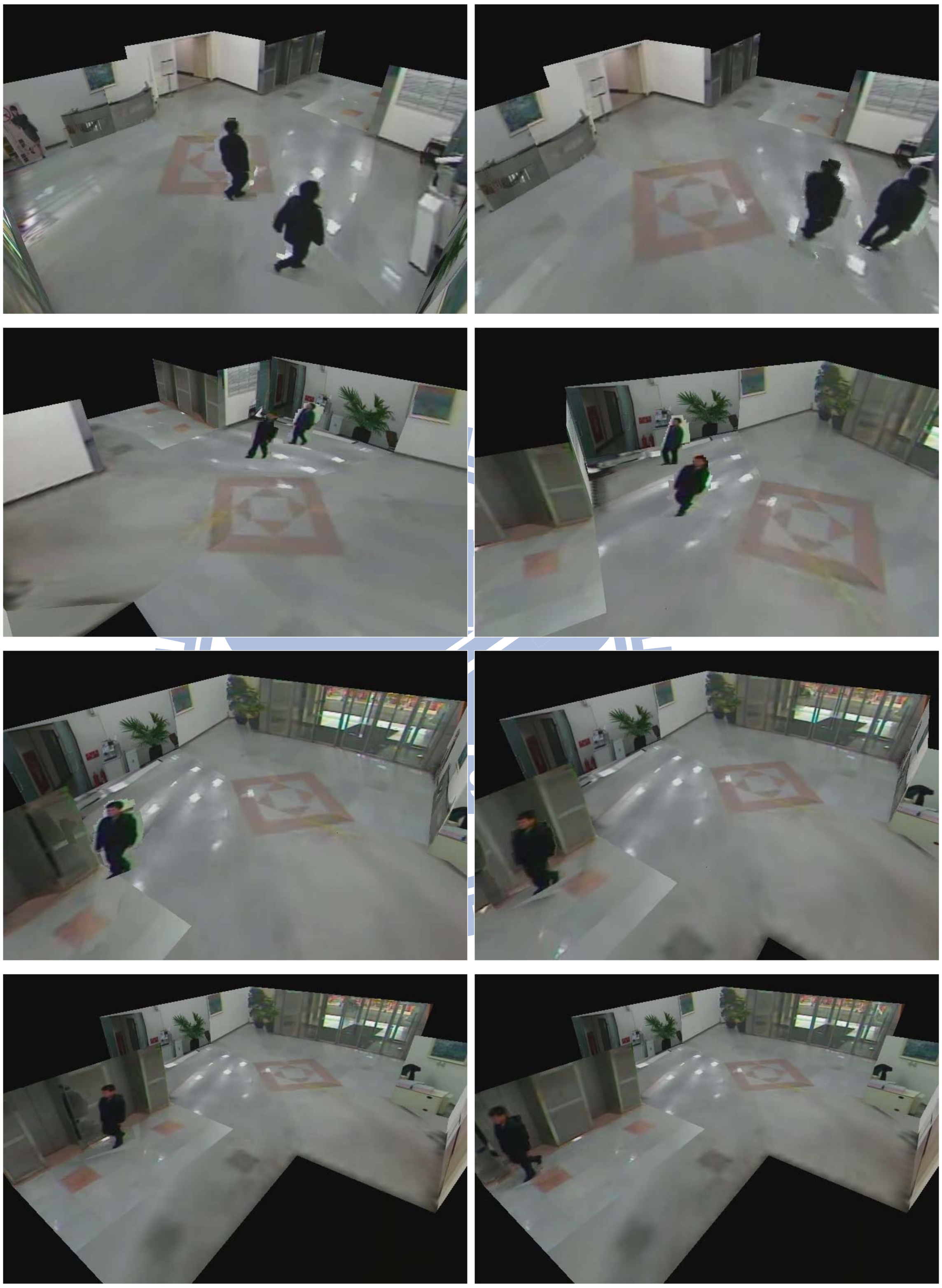


Figure 4.15: **Auto-tracking of targets.** The viewpoint of the operator auto-switch according the number of people in the monitored area. Two people in the view walk across the lobby and the viewpoint keeps track of targets.

Chapter 5

Conclusions

In this work, we present a 3-D surveillance system using integrated multiple cameras. In initialization, we apply planar patch modeling for the construction of 3-D environment model. The scenes from cameras are divided into several patches according to their structures and the polygons are inserted into the corresponding positions of the 3-D model. The homography transformations are estimated for relationships between image regions of the video contents and the corresponding areas of the 3-D model. Through homography transformations, we can map videos onto the 3-D model. Besides, automatic patch division causes the better rendering layout and lookup tables are built for accelerating the texture mapping. When the system runs on-line, the foreground objects are first extracted from the scene by background modeling. Then, vertical and horizontal projection histograms are applied to acquire the precise segmentation of the foreground objects. The locations of the foreground objects in the 3-D model are obtained through lookup tables and corresponding billboards are inserted into the model for 3-D visualization. In our experiments, we use three methods described below to improve the visual effects of the system. Smooth view switching is achieved by the interpolation of viewpoint parameters and the areas of blind spot of the cameras are filled with static photos for better layout of the 3-D environment. Also, ground masks can be used to remove the dynamic background objects specified beforehand. Finally, we construct a system that provides a single comprehensive view for the all monitored areas to facilitate tracking moving targets through its interactive control and immersive visualization. The process is real-time and it is suitable for both indoor and outdoor environments.

In the future, we plan to enhance the background modeling and tracking algorithms. Enhanced background modeling and tracking algorithms for precisely segmenting moving objects lead to better visual effect in the 3-D environment.

Bibliography

[1] Opengl tutorial. `http://www.swiftless.com/tutorials/opengl/opengltuts.html`.

[2] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[3] Yung-Cheng Cheng, Kai-Ying Lin, Yong-Sheng Chen, Jenn-Hwan Tarng, Chii-Yah Yuan, and Chen-Ying Kao. Accurate planar image registration for an integrated video surveillance system. In *Computational Intelligence for Visual Intelligence, 2009. CIVI '09. IEEE Workshop on*, pages 37–43, 30 2009-April 2 2009.

[4] Niklas Elmqvist and Philippas Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, 2008.

[5] A. Fernandes. Billboarding tutorial. Website, 2005. `http://www.lighthouse3d.com/opengl/billboarding`.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[7] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):657–662, April 2006.

[8] Jinhui Hu, Suya You, and U. Neumann. Approaches to large-scale urban modeling. *Computer Graphics and Applications, IEEE*, 23(6):62–69, Nov.-Dec. 2003.

[9] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, and Larry Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172 – 185, 2005. Special Issue on Video Object Processing.

[10] Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 53, Washington, DC, USA, 2003. IEEE Computer Society.

[11] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, 2006.

[12] K. Mueller, A. Smolic, M. Droese, P. Voigt, and T. Wienand. Multi-texture modeling of 3d traffic scenes. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 1, pages I–657–60 vol.1, July 2003.

[13] Ulrich Neumann, Suya You, Jinhui Hu, Bolan Jiang, and JongWeon Lee. Augmented virtual environments (ave): Dynamic fusion of imagery and 3d models. In *VR '03: Proceedings of the IEEE Virtual Reality 2003*, page 61, Washington, DC, USA, 2003. IEEE Computer Society.

[14] S. Noronha and R. Nevatia. Detection and modeling of buildings from multiple aerial images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(5):501–518, May 2001.

[15] U. Neumann S. You, J. Hu and P. Fox. Urban site modeling from lidar. In *Second International Workshop on Computer Graphics and Geometric Modeling CGGM'2003*, May 2003.

[16] H. S. Sawhney, A. Arpa, R. Kumar, S. Samarasekera, M. Aggarwal, S. Hsu, D. Nister, and K. Hanna. Video flashlights: real time rendering of multiple videos for immersive model visualization. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 157–168, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[17] Ismail Oner Sebe, Jinhui Hu, Suya You, and Ulrich Neumann. 3d video surveillance with augmented virtual environments. In *IWVS '03: First ACM SIGMM international workshop on Video surveillance*, pages 107–112, New York, NY, USA, 2003. ACM.

[18] Rick Sizemore. Internet protocol/networked video surveillance market: Equipment, technology and semiconductors. Technical report, April 2008.

[19] Seth J. Teller, Matthew E. Antone, Zachary Bodnar, Michael Bosse, Satyan R. Coorg, Manish Jethwa, and Neel Master. Calibrated, registered images of an extended urban area. *International Journal of Computer Vision*, 53(1):93–107, 2003.

[20] M.C. Velez, D. Silver, and M. Tremaine. Understanding visualization through spatial ability differences. In *Visualization, 2005. VIS 05. IEEE*, pages 511–518, Oct. 2005.

[21] Yi Wang, David M. Krum, Enylton M. Coelho, and Doug A. Bowman. Contextualized videos: Combining videos with environment models to support situational understanding. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1568–1575, 2007.

[22] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina (Chapel Hill), Chapel Hill, NC, July 2006.

[23] Dengsheng Zhang and Guojun Lu. Segmentation of moving objects in image sequence: A review. *Circuits, Systems, and Signal Processing*, 20(2):143–183, March 2001.

[24] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000.