Integration of Multiple Views for a 3-D Indoor Surveillance System

Yi-Yuan Chen
Identification and Security Technology Center, Industrial Technology Research Institute, Hsinchu county, Taiwan
yiyuan@itri.org.tw

Yung-Huang Huang, Yung-Cheng Cheng, and Yong-Sheng Chen*
Department of Computer Science, National Chiao Tung University Hsinchu city, Taiwan
aocsheep@gmail.com, yccheng.cs94g@nctu.edu.tw, yschen@cs.nctu.edu.tw

Abstract

A conventional surveillance system uses multiple screens to separately display the acquired video streams. It may cause trouble to keep track of the moving targets due to the lack of spatial relationship among the camera views. This paper presents a surveillance system that can integrate multiple video contents into one single comprehensive view. To visualize the monitored area, the proposed system uses planar patches to approximate the 3-D model of the monitored environment and displays the video contents of cameras by applying dynamic texture mapping on the model. Therefore, this system can provide an efficient and practical way for interactive 3-D monitoring of the environment without complicated camera calibration and 3-D model reconstruction procedures. Moreover, the moving objects are extracted through foreground segmentation and then visualized through axis-aligned billboarding. In this way, the proposed system can provide security guards a better situational awareness of the monitored area, including the activities of the tracking targets.
Keywords: Video surveillance system, planar patch modeling, axis-aligned billboarding

1 Introduction

Recently, video surveillance has experienced accelerated growth because of continuously decreasing price and better capability of cameras [1] and has become an important research topic in the general field of security. Since the monitored regions are often wide and the field of views of cameras are limited, multiple cameras are required to cover the whole area. In the conventional surveillance system, security guards in the control center monitor the security area through

Figure 1: A conventional surveillance system with multiple screens.

a screen wall, as shown in Figure 1. It is difficult for the guards to keep track of targets because the spatial relationship between camera views displayed in adjacent screens is not intuitively known. Also, it is tiresome to simultaneously gaze between many screens over a long period of time. Therefore, it is beneficial to develop a surveillance system that can integrate all the videos acquired by the monitoring cameras into a single comprehensive view.

Many researches on integrated video surveillance systems have been proposed in the literature. In an outdoor surveillance system, an aerial or satellite photograph can be used as a reference map and various kinds of measurement equipments can be used to build the 3-D environment [2, 3, 4]. In [2], an accurate planar image registration method was proposed for precise texture mapping of videos on the reference map. Images from two cameras can be stitched on the 3-D model of the monitored area and the operator can move the viewpoint to easily keep tracking the moving target, even across camera views. Video flashlight method [3] was also proposed to apply for real-time mapping of the live videos on the 3-D environment model. In [4], an AVE (augmented virtual environment) system was proposed which can fuse multiple image streams onto a 3-D model substrate and can allow arbitrary viewing directions for visualization. Neumann, et al. utilized an airborne LiDAR (light detection and ranging) sensor system to collect 3-D geometry samples of a specific environment [5]. This system can then map videos onto the 3-D model, like a virtual projector, and can create a dynamic single polygon model to represent a moving object for visualization purpose [6]. Video billboards and video-on-fixed-planes methods project camera views, including foreground objects, onto individual vertical planes in a reference map to visualize the monitored area [7]. In the fixed billboard method, billboards are fixed facing toward specified directions to indicate

the capturing directions of cameras and the locations of billboards indicate the positions of cameras. However, the video displayed on a billboard is difficult to perceive if the angle between the viewing direction and the normal direction of the billboard is too large. In rotating billboard method, when the billboard rotates and faces to the viewpoint of the operator, neither camera orientations nor capturing areas can be preserved.

Although the systems described above offer a certain degree of view integration capabilities, there are still some problems in these systems. For example, video projection method is an effective way to display the video streams on the 3-D model. However, it requires camera calibration [8] with high precision and elaborate 3-D environment model constructed with advanced equipments [9, 10, 11] to accurately project the camera images on the model of the monitored area. Moreover, the 3-D moving objects appear to be distorted if they are not modeled as 3-D objects beforehand [2]. Rendering a polygon at the 3D position of a foreground object in the scene is much easier. However, the polygon may occlude background parts of the scene and may cause texture gaps on buildings [6]. If 3-D objects are rendered as icons in [3], real image appearances of the moving objects are lost. Obviously, it is better to render the moving objects according to their 3-D models, which can be constructed online from camera images. However, this kind of 3-D reconstruction remains essentially difficult and will increase a lot of computational burden.

In this work, we develop a 3-D indoor surveillance system based on the view integration of multiple cameras. We use planar patches to build the 3-D environment model beforehand and then display videos by using dynamic texture mapping on the 3-D model. To obtain the relationship between the camera contents and the 3-D model, homography transformations are estimated for every pair of image regions in the video contents and the corresponding areas in the 3-D model. Before texture mapping, patches are automatically divided into smaller ones with appropriate sizes according to the environment. Lookup tables for the homography transformations are also built for accelerating the coordinate mapping in the video visualization processing. Furthermore, moving objects are segmented from the background and are displayed via axis-aligned billboarding for better 3-D visual effects.

In the following of this paper, we will describe how to integrate camera views into a 3-D environment model and how to visualize the moving objects as billboards in Sections 2 and 3, respectively. Then we will present the experimental results in Section 4 and draw the conclusions in Section 5.

2 System configuration

Figure 2 illustrates the flowchart of constructing the proposed surveillance system. The proposed system uses planar patches to model the 3-D environment and then maps video contents to this model through homography transformation. Therefore, camera calibration for the cameras is not required. In the system configuration part, we construct lookup tables for the coordinates transformation of planar patches from the 2-D images acquired from IP cameras deployed in the scene to the 3-D model by specifying corresponding points between the 3-D model and the 2-D images. Since the cameras are fixed, this configuration procedure can be done only once beforehand. Based on the 3-D

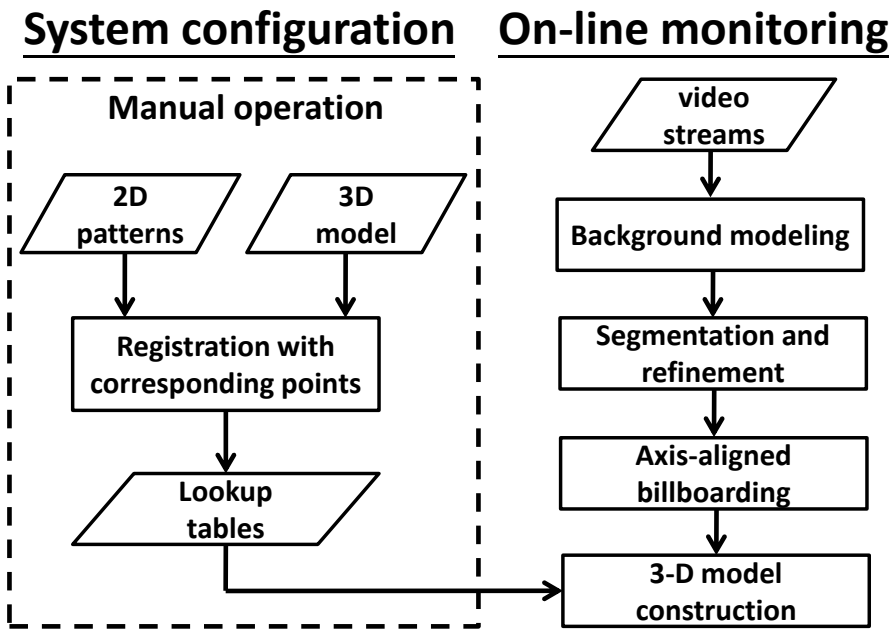


Figure 2: The flowchart and components of the proposed 3-D surveillance system.

model, in the on-line monitoring stage, all videos will be integrated and visualized in a single view in which the foreground objects extracted from images are displayed through billboards. In this way the operator can freely change the viewing direction and the system adjusts the normal direction of foreground objects accordingly.

2.1 Image registration

For a point on a planar object, its coordinates on the plane can be mapped to 2-D image through homography [12], which is a transformation between two planar coordinate systems. A homography matrix $\mathbf{H}$ represents the relationship between points on two planes:

$$s\mathbf{c}_{\mathrm{t}} = \mathbf{H}\mathbf{c}_{\mathrm{s}} \ , \tag{1}$$

where s is a scalar factor and $\mathbf{c}_{\mathrm{s}}$ and $\mathbf{c}_{\mathrm{t}}$ are a pair of corresponding points in the source and target patches, respectively. If there are at least four correspondences where no three correspondences in each patch are collinear, we can estimate the homography matrix $\mathbf{H}$ through the least-squares approach.

We regard $\mathbf{c}_{\mathrm{s}}$ as the coordinate of a point in the 3-D environment model and $\mathbf{c}_{\mathrm{t}}$ as the coordinate of a point in the 2-D image and then calculate the matrix $\mathbf{H}$ to map points from the 3-D model to the image. In the reverse order, we can also map points from the image to the 3-D model.

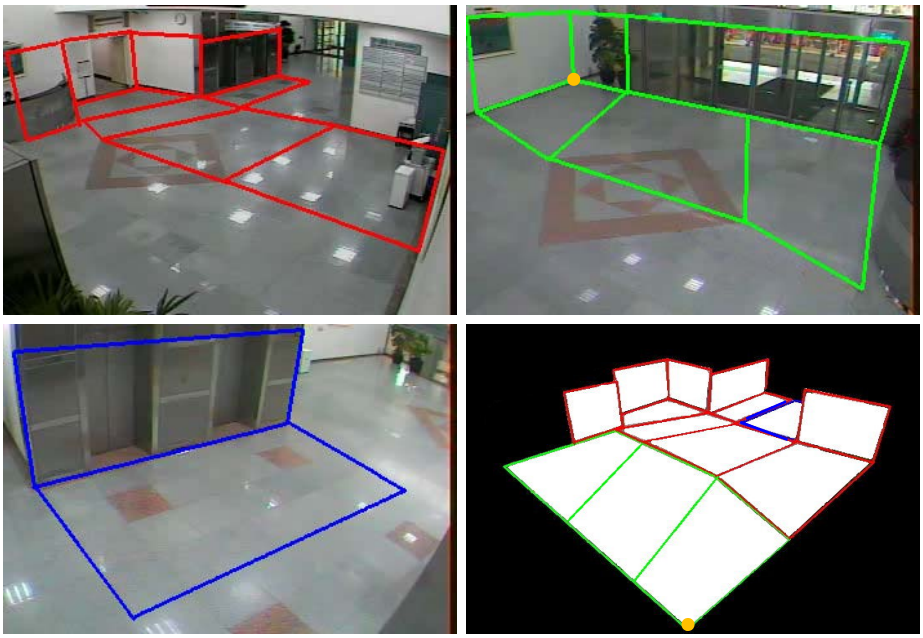


Figure 3: Planar patch modeling for 3-D model construction. Red patches (top-left), green patches (top-right), and blue patches (bottom-left) represent the mapping textures in three cameras. The yellow point is the origin of the 3-D model. The 3-D environment model (bottom-right) is composed of horizontal and vertical patches from these three cameras.

2.2 Planar patch modeling

Precise camera calibration is not an easy job [11]. In the virtual projector methods [3, 6], the texture image will be miss-aligned to the model if the camera calibration or the 3-D model reconstruction has large errors. Alternatively, we develop a method that approximates the 3-D environment model through multiple yet individual planar patches and then renders the image content of every patches to generate a synthesized and integrated view of the monitored scene. In this way we can easily construct a surveillance system with 3-D view of the environment.

Mostly we can model the environment with two basic building components, that is, horizontal planes and vertical planes. The horizontal planes for hallways and floors are usually surrounded by doors and walls, which are modeled as the vertical planes. Both kinds of planes are further divided into several patches according to the geometry of the scenes, as shown in Figure 3. If the scene consists of simple structures, a few large patches can well represent the scene with less rendering costs. On the other hand, more and smaller patches are required to accurately render a complex environment, at the expense of more computational costs.

In the proposed system, the 3-D rendering platform is developed on OpenGL and each patch is divided into triangles before rendering. Since linear interpolation is used to fill triangles with texture images in OpenGL which is not suitable for the perspective projection, distortion appears in the rendering results. One

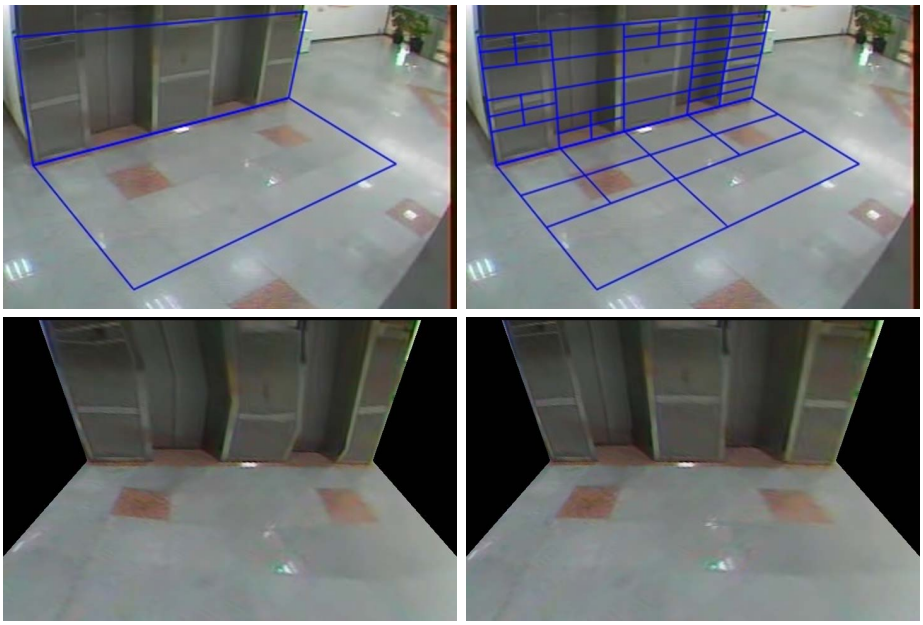


Figure 4: The comparison of rendering layouts between different numbers and sizes of patches. A large distortion occurs if there are fewer patches for rendering (left). More patches make the rendering much better (right).

can use a lot of triangles to reduce this kind of distortion, as shown in Figure 4, it will enlarge the computational burden and therefore not feasible for real-time surveillance systems.

To reach a compromise between visualization accuracy and rendering cost, we propose a procedure that automatically divides each patch into smaller ones and decides suitable sizes of patches for accurate rendering, as shown in the right part of Figure 4. Through homography transformation, we obtain the corresponding 3-D location for each pixel 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 patch rendering. We use the following mean-squared error, MSE, to estimate the amount of distortion when rendering image patches:

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

where I_{ij} is the intensity of the point obtained from homography transformation, $\tilde{I}_{ij}$ is the intensity of the point obtained from texture mapping, i and j are the coordinates of row and column in the image, respectively, and $m \times n$ represents the dimension of the patch in the 2-D image. In order to have an reference scale to quantify the distortion amount, a peak signal-to-noise ratio, PSNR, is calculated by

$$\text{PSNR} \quad = \quad 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) \quad , \tag{3}$$

where MAX_I is the maximum pixel value of the image. Typical values for the PSNR are between 30 and 50 dB and an acceptable value is considered to be about 20 dB to 25 dB in this work. We set a threshold T to determine the quality of texture mapping by evaluating whether PSNR $\geq T$. If the PSNR of the patch is lower than T, the procedure divides it into smaller patches and repeats the process until the PSNR values of every patches are greater than the given threshold T.

3 On-line monitoring

The proposed system displays the videos on the 3-D model. However, the 3-D foreground objects such as pedestrians are projected to image frame and become 2-D objects. They will appear flattened on the floor or wall since the system displays them on planar patches. Furthermore, there might be ghosting effects when 3-D objects are in the overlapping areas of different camera views. We need to tackle this problem by separating and rendering 3-D foreground objects in addition to the background environment.

3.1 Background modeling

Dynamic object extraction is a conventional problem in computer vision [13, 14]. The more accurate positions and shapes of objects are, the more reliable identification and tracking results can be obtained. Background subtraction is a simple foreground segmentation method, but it often fails due to illumination changes, shadows, and reflections. In this work we adopt the codebook algorithm to reduce these problems [15]. For each pixel in the image, codebook algorithm builds a codebook containing one or more codewords by clustering the training samples based on a color distortion metric and brightness bounds. During the training period, it records the longest interval in which the codeword does not appear. Therefore, this method can tolerate moving objects in the scene during the training period. The threshold of the decision function determining whether the pixel is foreground or background is adjusted according to the monitored environment.

3.2 Segmentation and refinement

In foreground extraction, it is difficult to separate one people to another when these two people are close to each other. Therefore, the extracted positions of moving targets will be inaccurate. 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. A region between two valleys around a peak indicates a single person, as shown in Figure 5. The peak must be greater than a specified peak threshold, $\mathrm{P_T}$, and the valley must be lower than another specified valley threshold, $\mathrm{V_T}$. The thresholds are set according to the monitored environment. In our experiment, the value of the peak threshold is selected as 70% of the height of the foreground region and the value of the valley threshold is chosen as the mean of the histogram. With these peaks and valleys detected, we can separate a large blob into several smaller ones which are more likely to contain a single moving target in each blob.

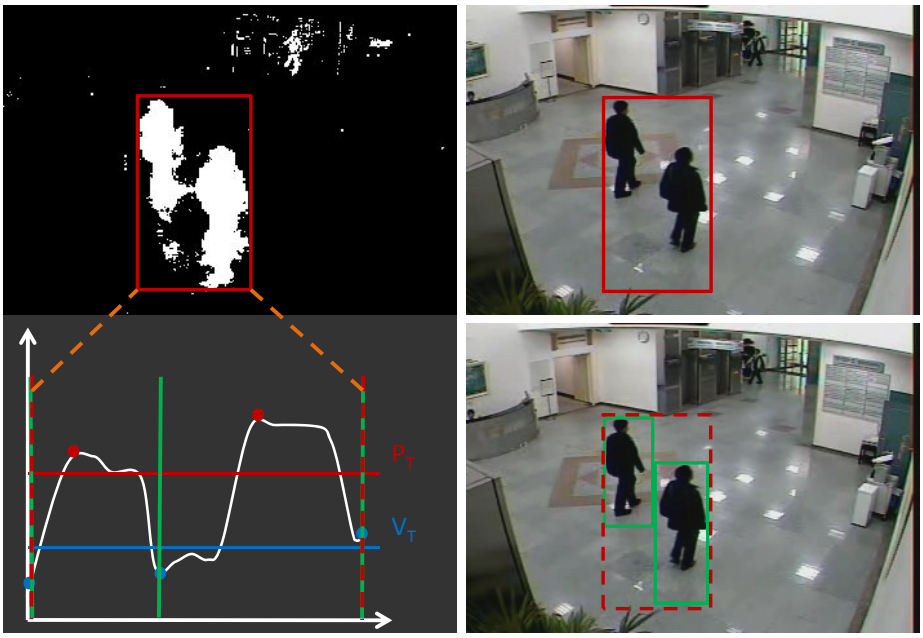


Figure 5: Blob partition through vertical projection histogram. The binary maps of the different foreground objects may sometimes overlap each other. To solve this problem, we find the peaks and valleys in vertical projection histogram to segment the blobs (left). The blob with red bounding box (top-right) can then be divided into two blobs with green bounding box (bottom-right).

The shadows and reflections may also cause problems for object segmentation and tracking because they may enlarge the blobs than their actual sizes. The major difficulty in separating the shadows or reflections from a targeted object is due to the huge diversities of the physical property of floor, directions of light sources, and the additive noises in the indoor environment. To reduce this effect, we apply Canny edge detection algorithm [16] in the region of the blob to find the contours of an object detected by background modeling. Then vertical and horizontal projection histograms are calculated to find the better boundary of the bounding box, as shown in Figure 6.

For each acquired image frame, sensor noises and lighting variation will affect the intensities of the static background region and thus reduce the accuracy of foreground extraction. This problem will cause the fluctuations of the detected boundary of foreground objects and result in bad visual effects, such as shaking of foregrounds, when we render the foreground objects on billboards. In this work, therefore, we stabilize the changes of the bounding box with the Kalman filter [17] in terms of its size and position, as shown in Figure 7.

3.3 Axis-aligned billboarding

In visualization, axis-aligned billboarding [18] constructs billboards in the 3-D model for moving objects, such as pedestrians, and the billboard always faces to the viewpoint of the user. The billboard has three properties, that is, location,

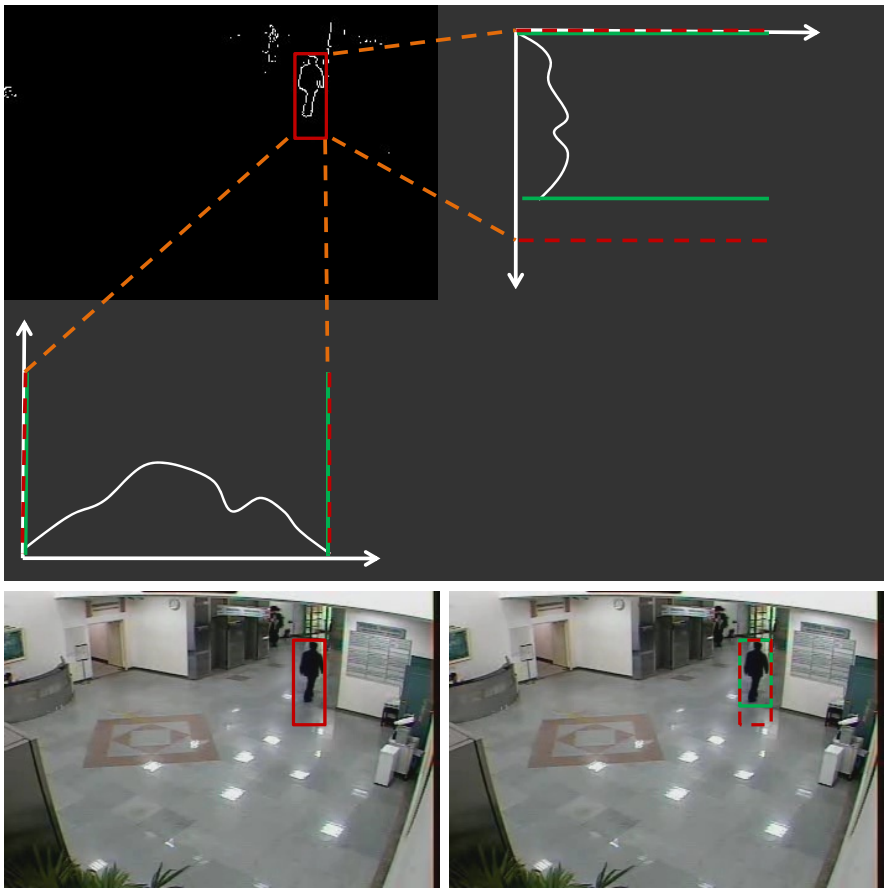


Figure 6: Refinement of the bounding box through vertical and horizontal projection histograms. The edges of the moving target are first detected. Vertical and horizontal projection histograms are used to restrict the width and height, respectively, of the bounding box (top). As a result, the red bounding box contains the target and the shadow (bottom-left) can be pruned into the green bounding box, which is more fitted to the moving target (bottom-right).

height, and direction. By assuming that all the foreground objects are always moving on the floor, the billboards can be aligned to be perpendicular to the floor in the 3-D model. The 3-D location of the billboard is estimated by mapping the bottom-middle point of the foreground bounding box in the 2-D image through the lookup tables, as shown in Figure 8(a). The ratio between the height of the bounding box and the 3-D model determines the height of the billboard in the 3-D model. The relationship between the direction of a billboard and the viewpoint is defined as shown in Figure 8(b).

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

$$\mathrm{Y} = (\mathbf{n} \times \mathbf{v}) \ , \tag{4}$$

$$\theta = \cos^{-1}(\mathbf{v} \cdot \mathbf{n}) \ , \tag{5}$$

where **v** is the vector from the location of the billboard, L, to the location E projected vertically from the viewpoint to the floor, **n** is the normal vector of the billboard, Y is the rotation axis, and θ is the estimated rotation angle. The normal vector of the billboard is parallel to the vector **v** and the billboard is always facing toward the viewpoint of the operator.

3.4 Video content integration

If the fields of camera views are overlapped, objects in these overlapping areas are seen by multiple cameras. In this case, there might be ghosting effects when we simultaneously display videos from these cameras. To deal with this problem, we use 3-D locations of moving objects to identify the correspondence of objects in different views. When the operator chooses a viewpoint, the rotation angles of the corresponding billboards are estimated by the method presented above and the system only render the billboard whose rotation angle is the smallest among all of the corresponding billboards, as shown in Figure 9.

3.5 Automatic change of viewpoint

The proposed surveillance system provides target tracking feature by determining and automatic switching the viewpoints. Before rendering, several viewpoints are specified in advance to be close to the locations of the cameras. During the viewpoint switching from one to another, the viewpoint gradually marches from the starting point to the destination one for smooth view transition.

The switching criterion is defined as the number of blobs found in the specific areas. First, we divide the floor area into several parts and associate them to each camera, as shown in Figure 10. When people move in the scene, the viewpoint is switched automatically to the predefined viewpoint of the area containing more foreground objects. We also make the billboard transparent by setting the alpha value of textures, so the foreground objects appear with fitting shapes, as shown in Figure 11.

4 Experiment results

We developed the proposed surveillance system on a PC with Intel Core Quad Q9550 processor, 2GB RAM, and one nVidia GeForce 9800GT graphic card. Three IP cameras with 352 $\times$ 240 pixels resolution are connected to the PC through Internet. The frame rate of the system is about 25 frames per second.

Owing to the limit of the fields of view of the cameras, there are some areas in the 3-D model without textures. In order to obtain a more complete view of the monitored area, we took photos of these areas beforehand for static texture mapping of the 3-D environment model. In the monitored area, automated doors and elevators are specified as background objects, albeit their images do change when the doors open or close. These areas will be modeled in background construction and not be visualized by billboards. The proposed system uses a ground mask to indicate the region of interest. Only the moving objects located in the indicated areas are considered as moving foreground objects, as shown in Figure 12.

The experimental results shown in Figure 13 demonstrate that the viewpoint can be able to be chosen arbitrarily in the system and operators can track targets with a closer view or any viewing direction by moving the virtual camera. Moreover, the moving objects are always facing the virtual camera by billboarding and the operators can easily perceive the spatial information of the foreground objects from any viewpoint.

Each billboard visualizes a moving object by displaying the image within its bounding box. However, the bounding box is a rectangle and contains the true foreground object with irregular shape as well as some background areas. To further improve the visual effects, we utilize the alpha channel of the color model for each billboard and set the background areas in the billboard to be transparent, as shown in Figure 14. In this way, the moving objects fit better with the static background without image conflicts of background areas from both dynamic videos and static background images.

5 Conclusions

In this work we have developed an integrated video surveillance system that can provide a single comprehensive view for the monitored areas to facilitate tracking moving targets through its interactive control and immersive visualization. We utilize planar patches for 3-D environment model construction. The scenes from cameras are divided into several patches according to their structures and the numbers and sizes of patches are automatically determined for compromising between the rendering effects and efficiency. To integrate video contents, homography transformations are estimated for relationships between image regions of the video contents and the corresponding areas of the 3-D model. In visualization, the foreground objects are first extracted from the scene by background modeling. Then, vertical and horizontal projection histograms are applied to the extracted objects to generate the precise segmentations of the foreground objects, which are displayed on billboards.

Acknowledgements

This work was supported in part by the Ministry of Economics Affairs, Taiwan, under Grants 98-EC-17-A-02-S1-032 and 98-EC-17-A-02-S2-0047, and Industrial Technology Research Institute, Taiwan, under Grant 9365C51200.

References

[1] R. Sizemore, "Internet protocol/networked video surveillance market: Equipment, technology and semiconductors," Tech. Rep., 2008.

[2] Y. Cheng, K. Lin, Y. Chen, J. Tarng, C. Yuan, and C. Kao, "Accurate planar image registration for an integrated video surveillance system," in *IEEE Workshop on Computational Intelligence for Visual Intelligence*, 2009, pp. 37–43.

[3] 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*, 2002, pp. 157–168.

[4] U. Neumann, S. You, J. Hu, B. Jiang, and J. Lee, "Augmented virtual environments (ave): dynamic fusion of imagery and 3-d models," in *IEEE Virtual Reality*, 2003, pp. 61–67.

[5] S. You, J. Hu, U. Neumann, and P. Fox, "Urban site modeling from lidar," in *ICCSA'03: Proceedings of the 2003 international conference on Computational science and its applications*, 2003, pp. 579–588.

[6] I. O. Sebe, J. Hu, S. You, and U. Neumann, "3-d video surveillance with augmented virtual environments," in *IWVS '03: First ACM SIGMM international workshop on Video surveillance*, 2003, pp. 107–112.

[7] Y. Wang, D. M. Krum, E. M. Coelho, and D. A. Bowman, "Contextualized videos: Combining videos with environment models to support situational understanding," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1568–1575, 2007.

[8] Z. Zhang and Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, 2000.

[9] J. Hu, S. You, and U. Neumann, "Approaches to large-scale urban modeling," *IEEE Computer Graphics and Applications*, vol. 23, no. 6, pp. 62–69, 2003.

[10] S. Noronha and R. Nevatia, "Detection and modeling of buildings from multiple aerial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 501–518, 2001.

[11] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master, "Calibrated, registered images of an extended urban area," *International Journal of Computer Vision*, vol. 53, no. 1, pp. 93–107, 2003.

[12] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.

[13] D. Zhang and G. Lu, "Segmentation of moving objects in image sequence: A review," in *Circuits, Systems, and Signal Processing*, 2001, pp. 143–183.

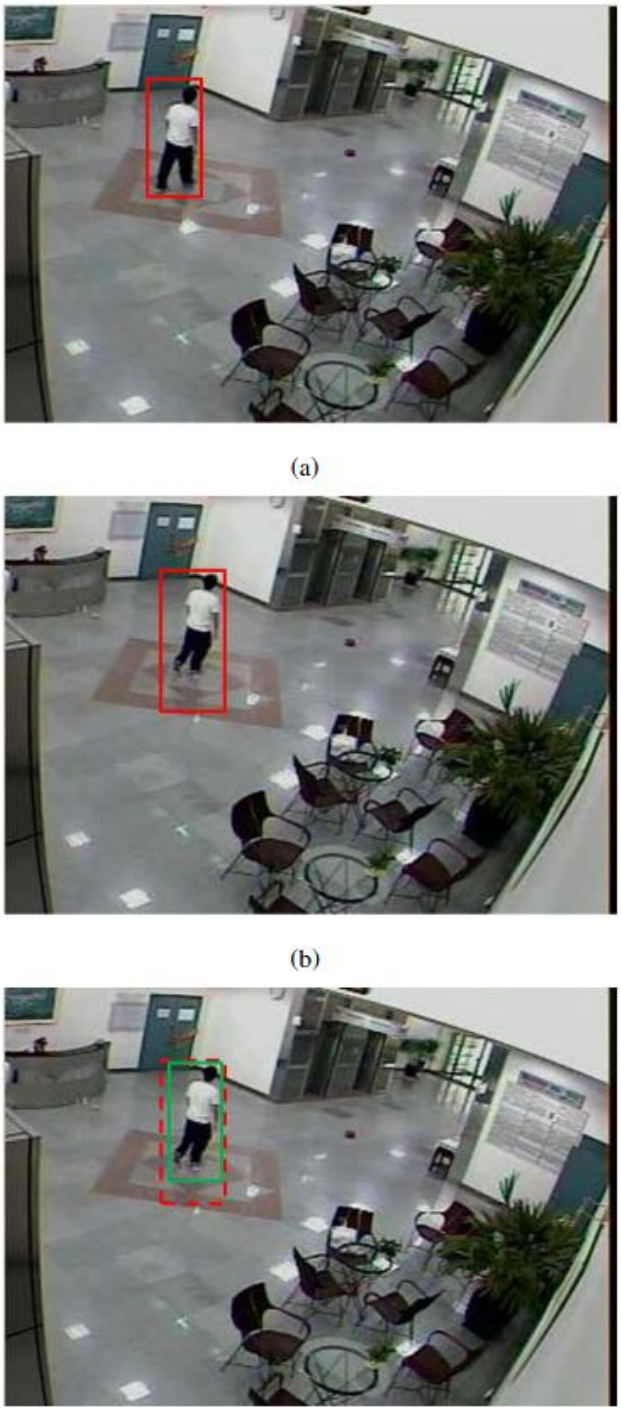
[14] T. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[15] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[16] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[17] G. Welch and G. Bishop, "An introduction to the kalman filter," Tech. Rep., 1995.

[18] A. Fernandes, "Billboarding tutorial," 2005. [Online]. Available: http://www.lighthouse3d.com/opengl/billboarding/

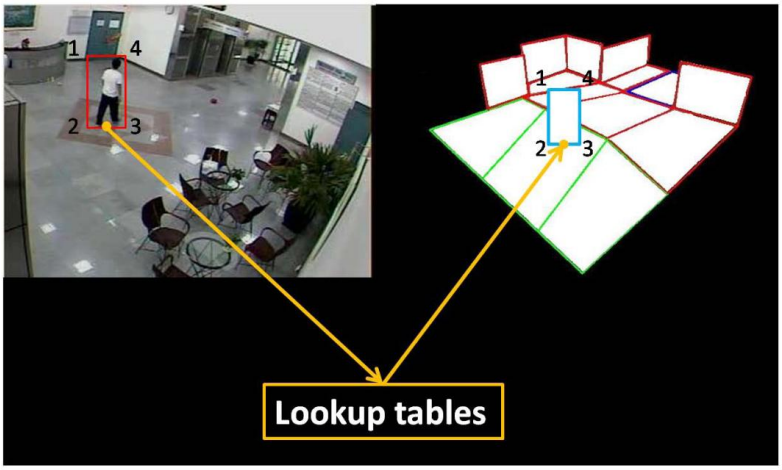


(a)

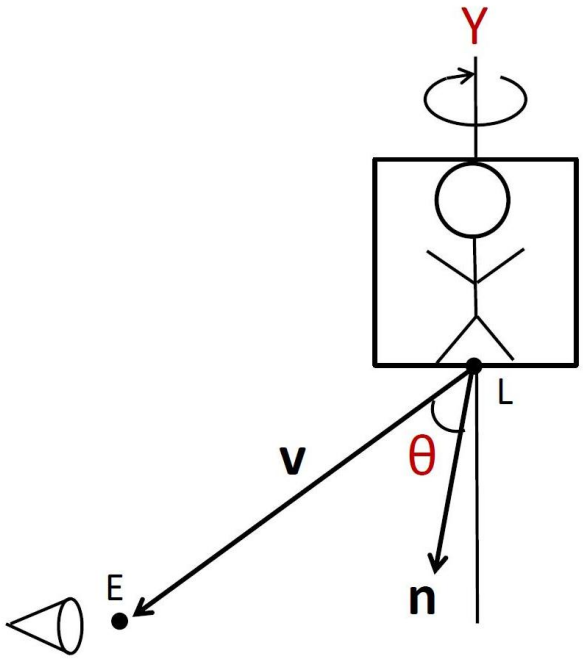
(b)

(c)

Figure 7: Smooth tracking of the bounding box through the Kalman filter. From consecutive images (a) to (c), the red bounding boxes are stabilized into green bounding boxes by the Kalman filter.



(a)



(b)

Figure 8: (a) Anchoring the billboards through the lookup tables. (b) Orientation determination of the axis-aligned billboarding. L is the location of the billboard, E is the location projected vertically from the viewpoint to the floor, and **v** is the vector from L to E. The normal vector **n** of the billboard is rotated according to the location of the viewpoint. Y is the rotation axis and θ is the rotation angle.

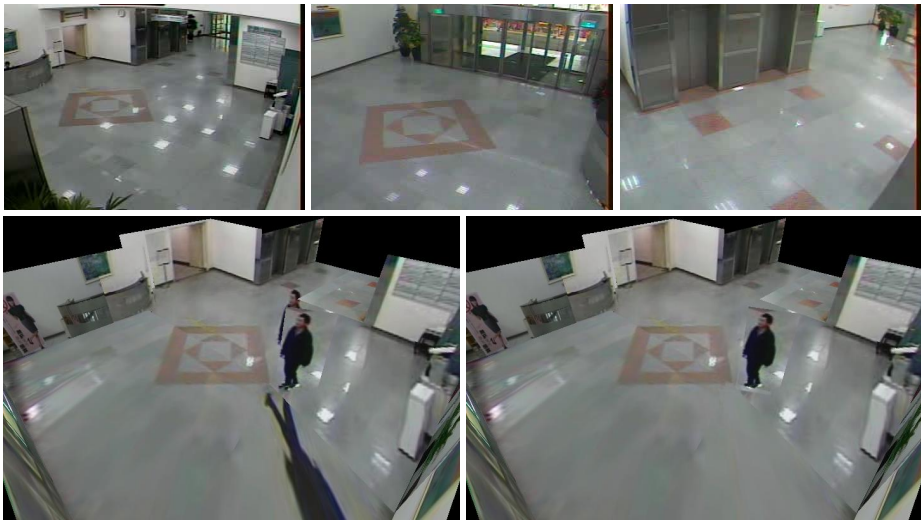


Figure 9: Removal of the ghosting effects. When we render the foreground object from one view, the object may appear in another view and thus cause the ghosting effect (bottom-left). Static background images without foreground objects are used to fill the area of the foreground objects (top). Ghosting effects are then removed and static background images can be update by background modeling.

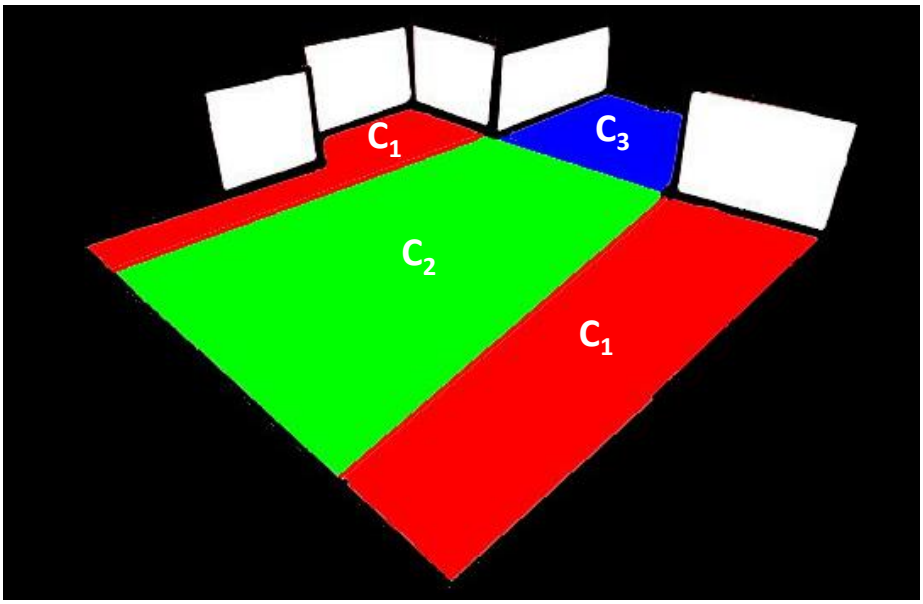


Figure 10: Determination of viewpoint switch. We divide the floor area depending on the fields of view of the cameras and associate each area to one of the viewpoint close to a camera. The viewpoint is switched automatically to the predefined viewpoint of the area containing more foreground objects.

Figure 11: Automatic switching the viewpoint for tracking targets. People walk in the lobby and the viewpoint of the operator automatically switches to keep track of the targets.

Figure 12: Dynamic background removal by ground mask. There is an automated door in the scene (top-left) and it is visualized by a billboard (top-right). A mask covered the floor (bottom-left) is used to decide whether to visualize the foreground or not. With the mask, we can remove unnecessary billboards (bottom-right).

Figure 13: Immersive monitoring at arbitrary viewpoint. We can zoom out the viewpoint to monitor the whole surveillance area or zoom in the viewpoint to focus on a particular place.

(a) (b)

Figure 14: The image conflicts of background areas between the dynamic videos on the billboards and the static background images, as shown in (a), can be removed by setting the background regions in the billboards to be transparent, as shown in (b).