3D Video Surveillance with Augmented Virtual Environments

Ismail Oner Sebe, Jinhui Hu, Suya You, Ulrich Neumann

Integrated Media Systems Center
University of Southern California
Los Angeles, CA 90089-0781
{iosebe, jinhuihu, suyay, uneumann}@graphics.usc.edu

ABSTRACT

Recent advances in sensing and computing technologies have inspired a new generation of data analysis and visualization systems for video surveillance applications. We present a novel visualization system for video surveillance based on an Augmented Virtual Environment (AVE) that fuses dynamic imagery with 3D models in a real-time display to help observers comprehend multiple streams of temporal data and imagery from arbitrary views of the scene. This paper focuses on our recent technical extensions to our AVE system, including moving object detection, tracking, and 3D display for effective dynamic event comprehension and situational awareness. Moving objects are detected and tracked in video sequences and visualized as pseudo-3D elements in the AVE scene display in real-time. We show results that illustrate the utility and benefits of these new capabilities.

Categories and Subject Descriptors

I.4.8 [**Scene Analysis**]: Sensor fusion, Tracking; I.3.5 [**Computational Geometry and Object Modeling**]:Modeling packages; Video Surveillance

General Terms

Security, Design.

Keywords

Augmented Reality, Video Surveillance, Object Detection and Tracking

1. INTRODUCTION

Advanced video surveillance technologies and systems are pursued for varied security, law enforcement, and military applications, in addition to civilian environment monitoring, health care, wildlife preserve management, traffic measurement, and catastrophe response management.

In complex environments, it is often impossible for a single sensor to maintain its view of an entire scene since occlusions are common and the sensor has a limited field of regard. To provide an accurate and comprehensive picture of wide area scenarios, a promising solution is to use a large, distributed multi-sensor network to cooperatively interpret the entire scene. Computing and sensing technologies enable multimodal and distributed video sensors and networks to be deployed in commercial or military establishments. However, new problems arise from the wide spread use and proliferation of such sensor resources. Most significant is the human ability (or lack thereof) to successfully fuse and comprehend the information that these sensors can provide.

Typically, in a large surveillance control center, a human operator interacts with dozens (or hundreds) of sensor sources using several separate monitors/windows for visualizing and analyzing the video or data streams (Figure 1). The operator is tasked with integrating the separate 2D images in their cognitive processing of the data to understand the relationships between the images and how the situation changes in the scene (e.g., as people or vehicles in the scene enter and leave the sensor's fields of regard). Although each separate source produces useful data, the human operator is easily overwhelmed with the task of integrating these varied forms of data into a complete global view and understanding of a scene. This problem is exasperated as we scale this scenario to larger networks of video/image/data streams from aerial and ground-based sensors distributed and possibly moving through a whole city.

To cope with the above scenario, we developed an *Augmented Virtual Environment* (AVE) that fuses and displays multiple image streams onto a 3D model substrate that allows arbitrary

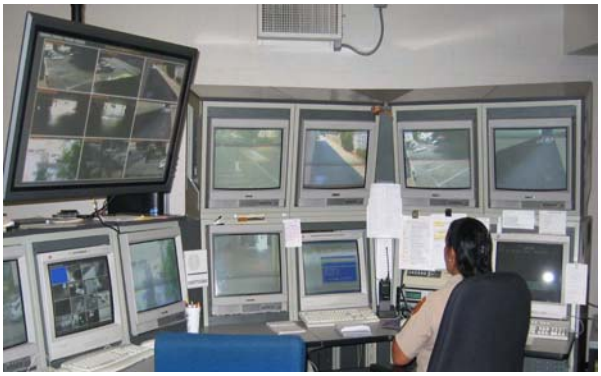

Figure 1 - Separate displays easily lead to users being overwhelmed with data fusion and comprehension of multiple image streams. [USC Public Security Surveillance Center]

IWVS'03, November 7, 2003, Berkeley, California, USA.

viewing [7]. The AVE is a virtual-reality model augmented by multiple video streams to provide a coherent data visualization that enhances scene comprehension, object tracking, and event detection. The presentation all sensor data in one uniform 3D context provides users with a natural browsing and multiresolution view of the spatial and temporal data provided by distributed or moving sensors spanning a wide area.

This paper presents our recent extensions to the AVE system, including moving object detection, tracking, and 3D display of dynamic events and objects observed by video cameras. The current system is capable of detecting and tracking moving people and vehicles, and visualizing them as 3D entities in real-time AVE visualizations.

2. RELATED WORK

Several recent systems address the problems of multiple sensor fusion and data analysis for video surveillance and military applications. For example, the Distributed Interactive Video Array (DIVA) developed by Trivedi [10] provides a large-scale, redundant cluster of video streams to observe a remote scene, intending for security and intelligent transportation applications. Spann, et al [9] developed a visualization system by fusing dynamic imagery with geometry model for battlefield visualization and situational awareness. The VideoFlashlight system developed at Sarnoff Corporation allows immersive visualization of urban sites by offline fusion of multiple video streams and 3D site models [4]. The Video Surveillance and Monitoring (VSAM) project, conducted at CMU, Sarnoff Corporation, and other institutions developed automated video understanding technologies, enabling a single human operator to monitor activities over a broad battlefield using a distributed network of active video sensors [13]. The key idea of the VSAM system is to automatically detect and track people, vehicles, and their interactions from multiple video streams, and then insert selected images of interest into an environment map to reduce the operator's cognitive load.

While the above systems offer many useful capabilities, they lack the ability to fuse and display dynamic moving objects as real-time 3D elements in a 3D scene model – a capability that this paper describes as an extension to the AVE system.

3. AVE ARCHITECTURE

An AVE is a novel and comprehensive approach to data fusion, analysis, and visualization that incorporates and presents all the sensors, abstract data, objects, and scenes models within a common context to produce a concise, coherent, and non-conflicting representation for time-space interpretation of real world activity. Video sensors are modeled as "virtual projectors" that have the same imaging parameters as the corresponding sensors. The sensors can be static or dynamic and on different platforms (head-worn, panoramic, aerial, building-mounted, etc.). Sensor data streams can be live or recorded from archives of government or commercial sources. A 3D scene model is created from range sensors and augmented with the images to create a visualization of both the geometry and dynamic imagery in a common 3D context. As with any virtual environment, an AVE allows users to freely move their viewpoints from a "god's-eye view" that visualizes a large region of an environment to a specific area or event of interest, such as a person entering or leaving a building entrance.

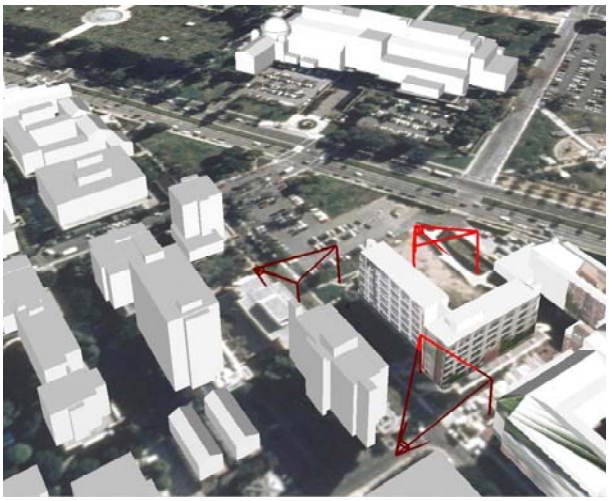

Figure 2 - An AVE system screen snapshot showing three video projections within a campus area.

Figure 2 shows a sample AVE visualization of a USC campus building complex with three video streams projected onto building models. The three video cameras are depicted by their red wireframe viewing frustums to show their current positions and orientations in the world. Note that the visualization viewpoint is completely arbitrary, and the aerial view aids users in fusing and comprehending the multiple camera images and their relationships to the scene. (A static aerial image texture has also been added to the ground to add scene context to the visualization.)

While Figure 2 shows useful visualizations or the static structures and imagery captured by cameras, moving objects such as people and vehicles are distorted when viewed as simple projections onto

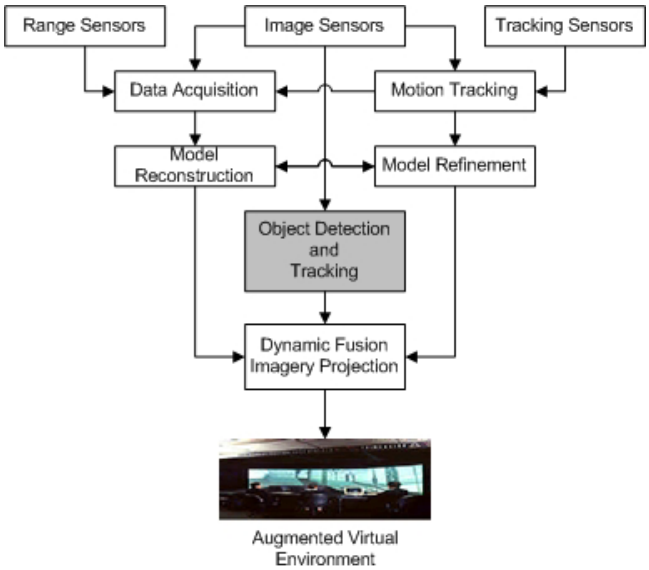

Figure 3 – Components of the Augmented Virtual Environment (AVE) system with dynamic modeling.

the static environment model. This is the main problem addressed in this paper and it is discusssed in more detail in Section 5, as is our dynamic modeling approach to addressing it. Before that, the remainder of Section 3 summarizes the AVE system components and Section 4 describes the tracking methods used to segment moving objects from the static scene.

Figure 3 depicts the main components of the AVE system.

Imagery acquisition: The video acquisition module supports real-time video capture and visualization of concurrent firewire DV video (up to three real-time full resolution DV sources). Note that multiple sets of three cameras can be multiplexed to the three real time inputs to allow for scalable camera numbers at shared update rates. High-resolution static images are also stored and recalled for background textures.

Geometry model acquisition: AVE visualization is a reflection of a real world scene, so we require accurate geometry models of real scenes. An accurate scene model facilitates many tasks including: (1) rapid comprehension of spatio-temporal relationships between sensors, events, and scene features; (2) geometry focus of attention and missing planning; (3) model-based camera calibration and motion tracking; and (4) visibility analysis for sensor planning and control of communication bandwidths. We utilize an airborne LiDAR (Light Detection and Ranging) sensor system to collect 3D geometry samples of a specific environment. We developed a semi-automatic system to extract complex building structures with irregular shapes and surfaces [12].

Sensor tracking and calibration: A key requirement for dynamic video projection in an AVE is to maintain accurate registration between the geometric models and the video textures. We developed a hybrid-sensor tracking method that uses the scene model, natural occurring features, and GPS and INS sensors for dynamic camera tracking in an open outdoor environment [7].

Data fusion and video projection: Real time visible surface and projective texture computation makes use of high performance graphics processors to dynamically "augment" the 3D model with video textures [7]. Using GPU instructions, we achieve real time rendering of three DV-resolution video streams projected onto our USC campus model that has over 200 buildings (Figure 2).

Object detection and tracking: This is a new addition to our system that analyzes video imagery to segment and track moving objects in the scene. The segmented regions are used to determine an approximate model that is dynamically positioned to capture the projection of the segmented imagery, thereby creating a pseudo-3D dynamic model that enhances the visualization of moving objects. This component is detailed in the following two sections.

4. DYNAMIC OBJECT ANALYSIS AND DETECTION

4.1 Previous Work

Dynamic scene analysis and object extraction are traditional problems in computer vision. In the AVE context, we focus on the method of background subtraction for its ease, effectiveness, and real-time performance.

Background subtraction is a simple foreground/background segmentation method suitable for long video sequences [2]. Many variations of this method have been investigated. A major issue is the selection of an appropriate model for the background, since the performance of any approach heavily depends on the accuracy of the background estimation. In particular, this method has the following well-known problems:

1. Illumination changes: Slow changes in the background such as sunrise, sunset, and shadows from moving clouds can create problems.
2. Dynamic background: An inherent problem is the distinction between background and foreground. For example, a car in a parking lot should be classified as foreground during the parking process and background once parked.
3. High-traffic: Estimation of the background model often deteriorates when occluded, as with a road carrying dense traffic.
4. Shadow and inter-reflections: Shadows are often classified as objects. Specular reflections from the windows and smooth surfaces may cause errors in the estimation.
5. Camouflage: Background subtraction often fails when background and foreground have similar color or foreground objects have transparencies.
6. Occlusion: Partial occlusion of the foreground objects by background or other foreground objects cause detection and tracking errors.

These problems have been extensively investigated. The illumination change, dynamic background, and high-traffic problems can be solved by an appropriate choice of models for the background. Popular methods include simple averaging or estimation by a per-pixel Gaussian distribution, although these fail to model bi-modal backgrounds.

Ridder [3] suggests a pixel-based Kalman filter for background estimation. Stauffer and Grimson [1] use a per-pixel Gaussian mixture model for the background in order to obtain a robust background image. Harville [2] uses a similar method to model multi-modal backgrounds such as rotating fans, flickering monitors, and waiving trees. Although these background estimation methods address slow illumination changes and multi-modal backgrounds, they require careful estimation of the learning parameters for online processes.

The shadow, inter-reflection, and camouflage problems are difficult to solve with the background subtraction method since the only input is the color information of the pixels.

Model-based approaches [5, 6] can handle occlusions and classify detected objects into categories such as cars and people. These methods use a rough 3D model of the world and the objects that will be tracked. Remagnino and Baumberg [5] use two models, one for cars and one for humans, where the later is a deformable model. Although these methods are not suitable for real-time detection and tracking, the use of a dynamic (deformable) model is similar to our approach.

Another method that can effectively handle occlusions is proposed by Tao et.al [11]. They use a dynamic layer representation for every object that is detected; this representation enables the system to accurately track multiple objects even when they overlap.

4.2 Approach

Our approach uses a background subtraction detection method followed by a pseudo-tracking algorithm. The choice of a relatively simple algorithm is motivated by a need for real-time processing. Figure 4 depicts the main structure of the algorithm.

The background estimation largely determines the performance of the overall system. A variable-length time average dynamically models a single distribution background, although comparable results are obtained using a Gaussian based background model.

The background image is modeled as a temporal pixel average of recent history frames. It is dynamically updated at a selected time interval (from several seconds to several days). We chose an update rate of N=1500 frames (50 seconds) experimentally, and our experiments show that this offers similar performance to a single Gaussian distribution, with lower computation complexity. Unfortunately, averaging a sliding window over every frame requires N-frames of memory, which is impractical. We simply compute a new background update every N-frames as the average of the prior N-frame period. We use this background image until the next update.

Foreground objects are determined using the Euclidian distance criterion between new observations and the background model.

$$\Delta I_i(x,y) = \left\| I_i(x,y) - B_k(x,y) \right\| \quad (1)$$

where $\Delta I_i(x,y)$ is the difference image, and $I_i(x,y)$ is the input image at frame *i*.

Background subtraction is followed by a histogram-based threshold and morphological noise processing. A pixel is considered as a foreground candidate if its difference value is above a pre-defined threshold. Otherwise the pixel is classified as background. The parameters of this step are estimated by taking the quality of the video recorders and the size of the foreground target into consideration. The threshold is estimated offline such that the top ~5% of pixel differences are above threshold and labeled as foreground objects.

A morphological filter is applied to obtain the thresholded mask [14]. The filter is applied to all the possible foreground regions, and only those regions with large enough areas are passed as accepted foreground objects. We filter out objects smaller than 0.1% of the image size as noise. We apply the morphological filter before the connectivity analysis, since the complexity of the connectivity algorithm is proportional to the number of objects and foreground pixels.

From this filtered binary image, a mask image is constructed by using a two-pass 4-neighbors connectivity algorithm [8]. The mask image is used for filtering the input image to derive the foreground object regions.

Once objects are detected, pseudo-tracking is done by validation. We eliminate spurious new object detections for waiving trees or other foliage by removing objects that are validated for less than a second. Validation is done by correlation matching between detected objects in neighboring frames. A normalized intensity correlation [15] is used as the matching criteria to measure the similarities between two detected regions. We evaluate the sum of squared error of a 10x10 window located at the center of the objects, and choose the minimum error pairs as the best tracked matches. Object regions with an acceptable matching score are assigned a target tag number. The correlation threshold is determined experimentally by examples and by taking the quality of the video cameras and imaging conditions into consideration. An alternative correlation method may use the shape descriptions of the objects, such as matching the elliptic Fourier descriptors of the contours of the objects.

The final outputs of the detection and tracking system are the four-corner coordinates $\{\mathbf{x}_i \mid \mathbf{x}_i = (x_i, y_i), i = 1 \sim 4\}$ bounding the moving object regions in the 2D image plane. These corners are used for modeling the moving objects and estimating their relative 3D positions in the scene. Figure 5 illustrates detecting and tracking moving vehicles and people on the USC campus.

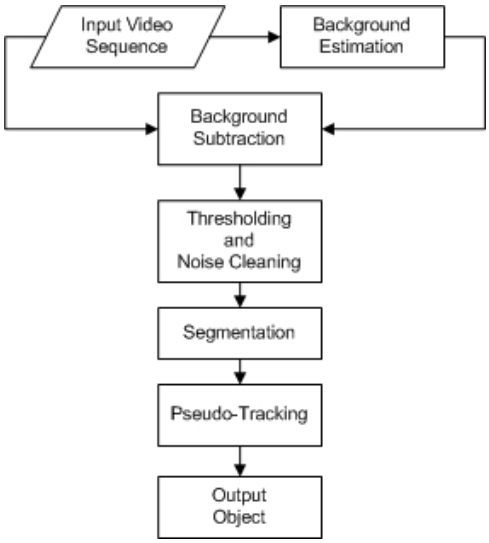

Figure 4 – Detection and Tracking System

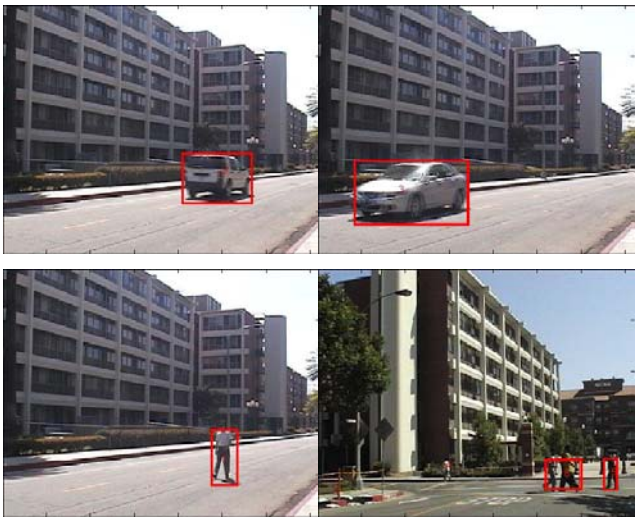

Figure 5 – Results of tracking vehicles and people.

5. 3D VISUALIZATION

The tracked regions need to be placed into the 3D scene model for AVE visualization. We create dynamic models to address this problem. The goal is to resolve the distortion problem that occurs when simply projecting video containing moving objects onto a static model. An aerial camera captures the person/car from a near vertical aspect. The AVE projection of that imagery will look realistic when viewed from other aerial viewpoints, however, if viewed from a ground-level viewpoint, these object look squashed-flat and distorted since the object has no geometric analog in the model and appears as painted on the street surface. Such distortions and lack of geometric fidelity arise from the lack of depth information from the image sensor. Our goal is to increase the fidelity of visualization by inserting approximate models (polygons) at the 3D positions of the objects in the scene – thereby providing a dynamic model surface on which to project the image of the moving object.

As shown in Figure 6, for example, a video sensor captures the dynamic movements of a person or a car in the scene. The video cameras are at ground level so their projections of a person walking or a car moving in the scene appear distorted since there are no models of these moving objects in the 3D scene model. The person and car appears to be "stretched" over the road and part of the building when viewed from a slightly raised viewpoint in Figure 6.

To produce a more realistic AVE visualization, we create a dynamic single polygon approximate model of moving objects. Three parameters are needed to define the polygon: position, orientation, and size. In Figure 7, point A is the optical center of the camera, the red box in the image plane is the tracking box that bounds the moving object. Point B is the mid point of the bottom edge of the tracking box. From points A and B, we compute a line AB (or Vector **v**) and compute its intersection with the ground (shown as point C). By assuming moving objects rest on the ground, point C is taken as the 3D position of the object and its model. The orientation of the model is denoted by vector **n**, which is in the same vertical plane as **v**, and set as perpendicular to the up vector. The 3D position (C) and orientation (**n**) define a plane. The vertices of the dynamic model polygon are computed as the projections of the detected 2D bounding box corners onto the plane, determining the polygon size.

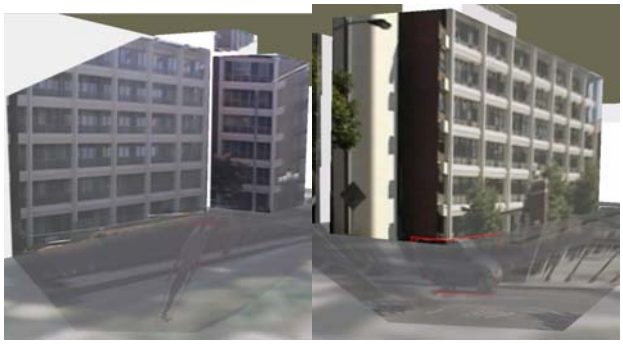

Figure 6 – Image projection of moving objects without corresponding models result in distorted presentation. Note the person and the car projected onto ground and building surfaces appears flat and distorted.

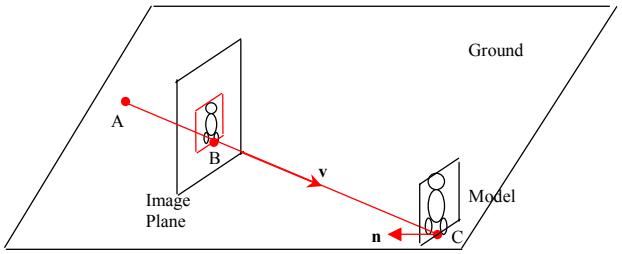

Figure 7 – Parameters for dynamic model position, orientation, and size are derived from a tracked object to define a dynamic model polygon.

When the texture is projected onto the scene including the dynamic model, the tracked object will be projected onto its model polygon in the correct 3D scene position. Figure 8 shows the result of applying this approach to the scene shown in Figure 6. The paths of the walking person and the moving car are depicted by a red line to show their 3D paths in the scene. The video texture projection process is unaltered and the visualization viewpoint remains completely arbitrary.

There are others ways to acquire dynamic 3D model of objects. Stereo methods are possible, but cost is an issue and video surveillance systems usually do not ensure that all areas are observable by multiple cameras, so single camera solutions are preferable.

An integrated prototype system has been developed to demonstrate the AVE capabilities for video surveillance and other applications. The current implementation achieves real time (~25Hz) visualizations on a 2.2GHz Pentium-4 workstation, supporting three live firewire video streams and high-resolution aerial photograph projections onto our entire USC campus model containing over 200 buildings (Figure 2).

6. CONCLUSION

This paper presents a novel visualization system for video surveillance based on tracking and 3D display of moving objects in an Augmented Virtual Environment (AVE). The tracking and dynamic modeling extensions provide increased fidelity and range of viewpoints for visualizations including moving objects. Any detection and tracking system could be used for similar effect. Our implementation is an adaptive background-subtraction

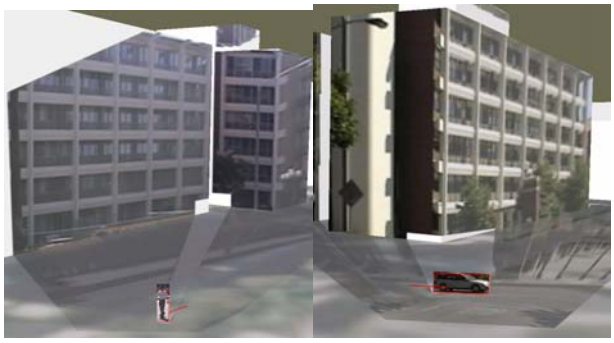

Figure 8 – Dynamic models automatically oriented and positioned at the 3D positions of the person and the car greatly improve the display of video textures projected onto the model.

method combined with a pseudo-tracking algorithm. The outputs of the tracking algorithm are used to create dynamic pseudo-3D models suitable for real-time video texture projection.

As future work, we pursue model-based tracking and classification of moving objects. Furthermore, enhanced tracking and classification methods will lead to better 3D models of the detected objects and thereby further enhance the fidelity of video projections.

Lastly, without modifying the video projection process, the dynamic models occlude background parts of the scene, creating texture gaps on buildings in particular. This problem may be addressed by developing learning or synthesis methods for the occluded areas.

7. ACKNOWLEDGMENTS

This work was supported by a Multidisciplinary University Research Initiative (MRUI) on "Next Generation 4-D Distributed Modeling and Visualization", and in part by the U.S. Naval Research Office (ONR). We thank the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, for their support and facilities. We acknowledge Todd Stannet at Airborn1 Inc. for providing us with the USC campus LiDAR data. We thank Kyoungchin Seo, Bolan Jiang, and Hongming Tu at the CGIT laboratory at USC for their help in system implementations.

8. REFERENCES

[1] Stauffer, C., Grimson, W.E.L. "Adaptive background mixture models for real-time tracking" CVPR 99, Vol. 2, pp. 246-252, June 1999

[2] Harville, M., Gordon, G., Woodfill, J. "Foreground Segmentation using Adaptive Mixture Models in Color and Depth". Proceedings of the IEEE Workshop on Detection and Recognition of Events in Video, July 2001

[3] Ridder, C., Munklet, O., Kirchner H. "Adaptive background estimation and foreground detection using Kalman-filtering". ICRAM'95, pp. 193-199, 1995

[4] Kumar R. Sawhney H.S. Guo Y. Hsu S. Samarasekera . "3D manipulation of motion imagery", ICIP2000, September 2000

[5] Remagnino R., Baumberg A., Grove T., Hogg D., Tan T., Worrall A., Baker K. "An integrated traffic and pedestrian Model-based Vision System". Proceedings of BMVC97, volume 2, 8-11th September, pp 380-389, 1997

[6] Koller, W., Malik J. "Robust multiple car tracking with occlusion reasoning". In Proc. Third European Conference on Computer Vision, May 2-6, 1994, pp. 189-196, LNCS 800, 1994.

[7] Neumann U., You S., Hu J., Jiang B., and Lee J. "Augmented Virtual Environments (AVE): Dynamic Fusion of Imagery and 3D Models" VR03, March 2003

[8] Gonzales, R.C. Woods, R.E. Digital Image Processing. Addison-Wesley, 2nd edition (September 1993), 41-4

[9] Spann J.R. Kaufman K.S. "Photogrammetry using 3D graphics and projective textures", IAPRS 2000, Vol. 33

[10] Hall B. Trivedi M. "A novel graphical interface and context aware map for incident detection and monitoring", 9th World Congress on Intelligent Transport Systems, October, 2002.

[11] Tao H. Sawhney H.S. Kumar R. "Dynamic layer representation with applications to tracking", in Proc. IEEE conf. on Computer Vision and Pattern Recognition, CVPR 2000, June 2000, vol 2, pp 134-141

[12] S. You, J. Hu, U. Neumann, and P. Fox, Urban Site Modeling From LiDAR, Second International Workshop on Computer Graphics and Geometric Modeling CGGM'2003, May 2003.

[13] T. Kanade, R. Collins, A. Lipton, P. Burt and L. Wixson, "Advances in cooperative multi-sensor video surveillance", Proc. of DARPA Image Understanding Workshop, Vol. 1, pp. 3-24, 1998

[14] R. Jain, R. Kasturi, B.G. Schunck, Machine Vision, McGraw-Hill International Edition, 1995, pp 76

[15] U. Neumann and S. You, "Natural Feature Tracking for Augmented-Reality", *IEEE Transactions on Multimedia*. Vol. 1. No.1, pp. 53-64, March 1999