Ulrich Neumann
uneumann@graphics.usc.edu
Integrated Media Systems Center
University of Southern California

Suya You
suyay@graphics.usc.edu
Integrated Media Systems Center
University of Southern California

Jinhui Hu
jinhuihu@graphics.usc.edu
Integrated Media Systems Center
University of Southern California

Bolan Jiang
bjiang@graphics.usc.edu
Integrated Media Systems Center
University of Southern California

Ismail Oner Sebe
iosebe@graphics.usc.edu
Integrated Media Systems Center
University of Southern California

Visualizing Reality in an Augmented Virtual Environment

Abstract

An Augmented Virtual Environment (AVE) fuses dynamic imagery with 3D models. An AVE provides a unique approach to visualizing spatial relationships and temporal events that occur in real-world environments. A geometric scene model provides a 3D substrate for the visualization of multiple image sequences gathered by fixed or moving image sensors. The resulting visualization is that of a world-in-miniature that depicts the corresponding real-world scene and dynamic activities. This paper describes the core elements of an AVE system, including static and dynamic model construction, sensor tracking, and image projection for 3D visualization.

1 Introduction

Three-dimensional Virtual Environments (VEs) are used for engineering, training simulations, entertainment, and planning. In many of these cases, the value of the VE is increased if both its geometry and appearance are accurate and realistic analogues of the real world. A further increase in value is obtained if *dynamic* geometry and appearance changes can model moving people and vehicles within the scene.

While many modeling systems facilitate the construction of geometric models for static scene elements such as buildings, terrain, and vegetation, such model creation demands significant skill and time. Commercially available models of the structures and terrain for a city block, for example, can take several weeks to create by many people working together. Furthermore, while photorealistic textured models have many uses, they represent a snapshot of the scene, often intentionally devoid of any people or vehicles that represent the dynamic or temporal aspects of the scene. There is little or no support for dynamic spatiotemporal updates in the current structure of VE models, databases, and rendering systems.

Our goal is to create an enhanced VE model that *augments* current static VE models with representations of the dynamic activities in the corresponding real-world scene. The inclusion of dynamic data makes such AVE models suitable for complex event visualization and comprehension in command-and-control and surveillance applications.

Our approach is to derive the dynamic aspects of the model from real-time video imagery (or other sensor data) gathered from multiple, and possibly moving, sources in the scene.

This paper describes the core components we developed to create an AVE system with specific details on the modeling of static and dynamic scene elements, issues that were summarized or omitted in Neumann & You (2003).

Presence, *Vol. 13, No. 2, April 2004,* 222–233

Before continuing, it is worth considering why an AVE or 3D representation is desirable. The existing alternative is for users to observe separate monitors/windows containing unique imagery streams with a map or 3D model to illustrate the imagery sources' geospatial pose (position and orientation) in the scene. Clearly, as sensors and networks proliferate, a human operator is overwhelmed with this visualization since the separate images provide no integration of the information and no aid to global high-level scene comprehension. The cognitive load to an operator or analyst is only likely to get worse as more computing, sensing, and communications pervade the environment. The human visual system is not capable of fusing and comprehending multiple, independently moving viewpoints of a scene. Our brains have evolved to process images of a 3D world from a single viewpoint. Therefore, a 3D virtual model is the framework we feel is appropriate for human comprehension of the aggregate information captured by sensors.

2 Related Work

There are other recent approaches to the problem of visualization of dynamic events. Several address the problems of multiple-sensor fusion and data analysis for video surveillance and military applications. Camera clusters are used to construct a 3D representation of moving objects in room-size environments (Kanade, Rander, Vedula, and Saito, 1999). The Distributed Interactive Video Array (DIVA) (Hall & Trivedi, 2002) provides a large-scale, redundant cluster of video streams in a 2D map context to observe a remote scene. Spann & Kaufman (2000) developed a system for fusing multiple images with geometry models for battlefield visualization and situational awareness. The VideoFlashlight system developed at Sarnoff Corporation uses video projection visualization (Kumar, Sawhney, Guo, Hsu, & Samarasekera, 2000), with vision-based modeling. The Video Surveillance and Monitoring (VSAM) project, conducted at CMU, Sarnoff Corporation, and other institutions, developed automated video-understanding technologies, presenting a single human operator with selected areas and events of interest over a broad battlefield using a distributed network of active video sensors (Kanade, Collins, Lipton, Burt, & Wixson, 1998).

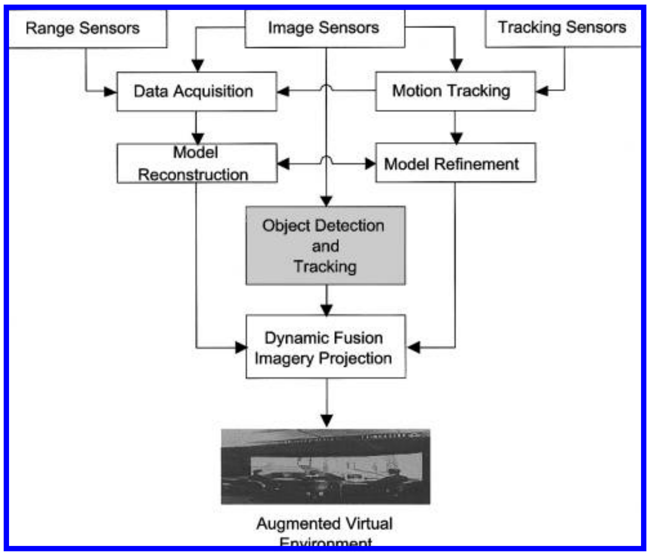

Figure 1. *AVE system components.*

Among these, the AVE approach is unique in its approach to modeling both the static and dynamic elements of a scene and merging them into a single 3D representation that fuses the acquired imagery and geometry.

3 AVE Overview

Figure 1 depicts the main components of the AVE system: (1) data acquisition to collect geometry from range and image sensors; (2) model reconstruction to produce static 3D surface models; (3) model refinement to segment structures and extract dominant and newly observed scene features; (4) sensor motion tracking to provide sensor pose and motion data for registration and data fusion; (5) object detection and tracking to facilitate the modeling of dynamic scene elements; and (6) data fusion to combine all the models, images, and video with annotation for a coherent visualization that supports scene comprehension and dynamic scene analysis.

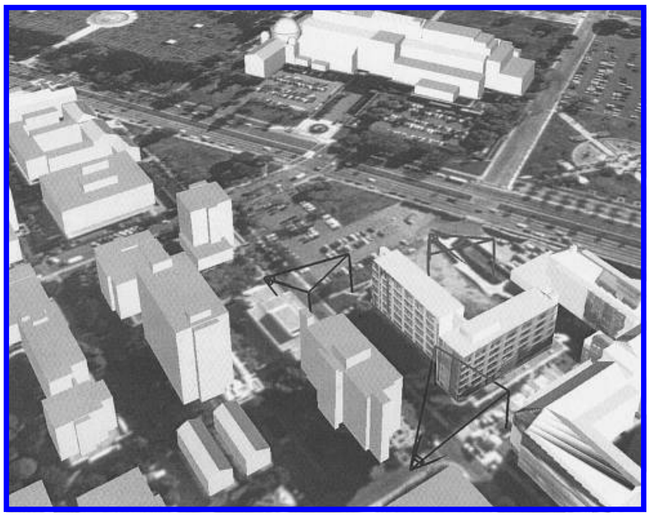

Figure 2. *An AVE image showing three video projections around a campus building complex.*

The AVE system produces visualizations such as those shown in Figure 2. This example has three tracked cameras whose viewing frustums are depicted by the wireframes. (A static texture is projected only onto the ground terrain.) The images from each camera are projected onto the scene model, effectively inverting the camera imaging process. In this example, the cameras view portions of the buildings and surrounding grounds. Note that the building textures visible in Figure 2 arise only from projected video textures. Users observe the dynamic movement of cars and people in the projected textures. Additional cameras could be placed in the scene to increase the area over which dynamic events are visible.

4 Static Scene Modeling

An accurate model of the scene is essential for fusing image data from varied viewpoints to resolve occlusions and create realistic visualizations from varied viewpoints. For efficiency and visualization context, we create a static model off-line so that only the dynamic scene elements have to be modeled at run-time and the entire scene model is visible regardless of sensor activity and coverage.

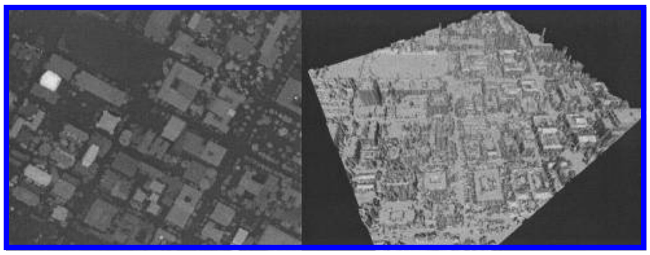

Figure 3. *LiDAR processing produces a resampled range image (left), and a reconstructed 3D mesh (right).*

There are many options for acquiring real-world data and creating scene models (Frueh & Zakhor, 2001; Hsieh, 1996; Gruen & Nevatia, 1998). An acquisition phase usually collects varied sensor data, often measuring 3D surface-coordinates from range scanners or intensity from image sensors. A reconstruction phase then processes the sensor data, performing resampling, hole-filling, and tessellation, to produce a consistent representation suitable for visualization. Texture-maps of static imagery are also mapped onto geometric models to produce photorealistic visualizations (Lee, Jung, & Nevatia, 2002; Debevec, Taylor, & Malik, 1996; Ribarsky, Wasilewski, & Faust, 2002).

Our data acquisition was a collaboration with Airbornel Inc., employing a Light Detection and Ranging system (LiDAR) in an aircraft to quickly collect a 3D point cloud for the University Park area with an accuracy of centimeters in height and sub-meter in ground position (typical). Multiple passes of the aircraft are merged to produce good coverage. The end result is a cloud of 3D point samples that we project and resample onto a regular grid (~0.5 m user-defined resolution) to produce a height field suitable for hole-filling and tessellation. Our reconstruction phase outputs a 3D mesh model in VRML format (Figure 3).

4.1 Model Refinement and Building Extraction

In urban areas, LiDAR provides useful approximations for urban features and buildings. However, resolution limits and measurement noise cause undersampling of building details, and occlusions from landscaping and overhangs lead to data voids in many areas of interest. The models need refinement to improve their utility and visualization value.

We developed techniques to semiautomatically extract and refine building models from LiDAR data. The details of the algorithms are presented in a paper by You, Hu, Neumann, & Fox (2003). LiDAR provides a clear footprint of a building's position and height. This information determines a building's geo-location and is used to segment it from the surrounding terrain. Based on the shape of a building roof (flat roof, slope roof, sphere roof, gable roof, etc.), we classify the building points and fit them to geometric primitives such as cubes, wedges, cylinders, polyhedrons, spheres, or superquadrics. This system is semiautomatic and requires relatively little user interaction to select primitives and key model points. The system automatically does the primitive-fitting and assembly of buildings from multiple primitives. Editing tools allow users to modify the models or obtain a specific representation quickly and accurately.

Figure 4 illustrates the results of model refinement. Primitives are automatically fit to the LiDAR data, so user mouse-click accuracy is not critical. Note the presence of curved surfaces and multiple primitives to represent complex buildings. Figure 5 shows the models we created for the entire University Park area, including the University of Southern California campus, Los Angeles Natural History Museum, Science Museum complex, Los Angeles Coliseum, and Sports Arena.

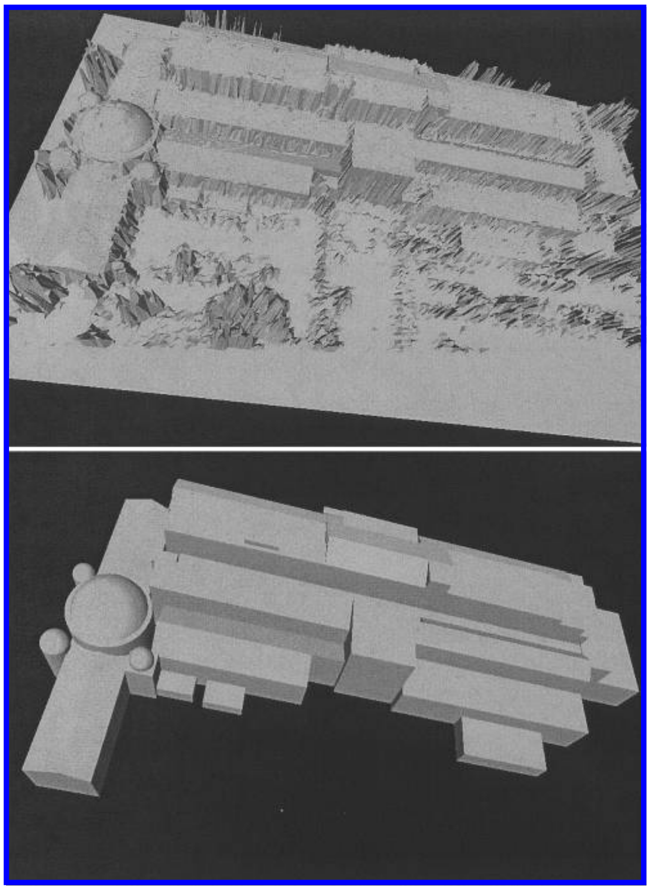

Figure 4. *Model verification shown (top) by primitives embedded in the original LiDAR data, and (bottom) extracted building model showing multiple primitives used for complex structures.*

5 Sensor Tracking

Tracking is vital in the process of data fusion and dynamic visualization. All modeling and imaging sensors must be calibrated to fuse their data into a common 3D context, thereby presenting the observer with a single coherent and evolving view of the complete scene.

Constructing a robust and accurate tracking system for outdoor environments is a challenging problem. A wealth of prior research in sensing technologies deals with motion tracking and registration (Azuma, 1997; Neumann & You, 1999). Methods employing a single tracking sensor have limitations; hybrid systems use multiple sensor measurements to produce more robust results (You, Neumann, & Azuma, 1999).

We developed a hybrid tracking system by integrating vision, GPS, and inertial orientation sensors to track the 6DOF pose of a mobile camera platform (Figure 6). A backpack houses a tracking package consisting of a high-resolution stereo camera head (MEGA-D from Videre Design), differential GPS receiver (Z-Sensor base/mobile from Ashtech), 3DOF inertial sensor (IS300 from Intersense), and a laptop computer. The stereo head has two digital cameras with a FireWire (IEEE 1394) interface. Our current system uses only one camera for video acquisition and vision tracking.

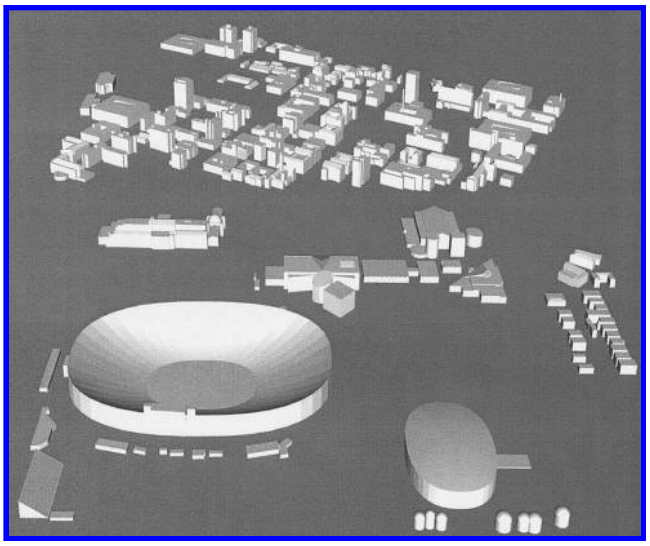

Figure 5. *University Park models extracted from LiDAR.*

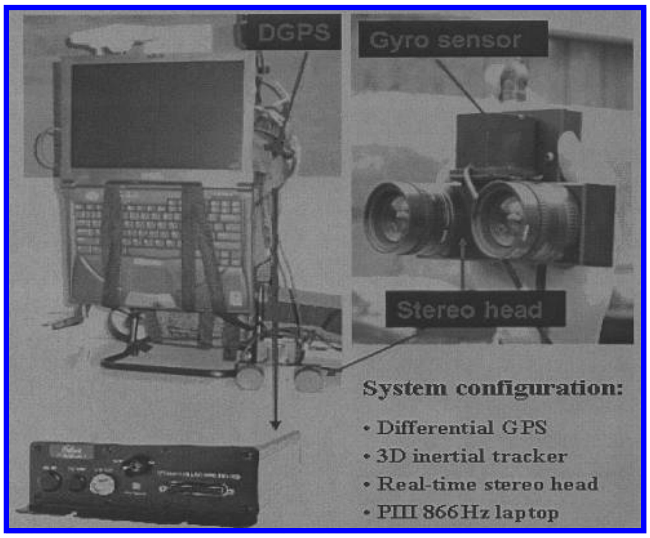

Figure 6. *Portable data acquisition and tracking system.*

The stereo stream may be used in the future for detailed 3D building-façade reconstruction.

The differential mode uses two units (base and remote) that communicate via a spread-spectrum radio to perform position calculations to about 2- to 10-cm accuracy at 2 to 5 updates per second. The inertial sensor is attached to the video camera to report its orientation. This sensor also measures the gravity vector and magnetic north to compensate for gyro drift (Foxlin, 1996). This orientation tracker is specified as achieving an approximate 1- to 3-degree accuracy, with 150 Hz maximum update rate.

The tracking and video acquisition systems run in real time and their data are stored on a laptop computer hard disk. We synchronize and resample all these data streams at the 30 Hz video rate. Each video image has a time stamp and tracking data encoded with it.

5.1 Pose Stabilization with Vision Sensor

Although the GPS–inertial tracking system provides an estimate of the camera pose that is adequate for some applications, its accuracy is inadequate for our AVE-performance expectations. Useful dynamic texture projection in the AVE requires accurate registration between the geometric models and the projected video textures. As cameras move, their images must remain aligned with the 3D models.

Figure 7 illustrates the typical dynamic registration errors that arise from direct use of the GPS–inertial tracking data to project images onto a 3D model. The top left image shows a view of a camera image projected onto a 3D building model. (A wireframe indicates the camera's—and projector's—frustum.) The top right image shows the same projected texture-image rendered from the camera viewpoint. In this example, misalignments between the texture-image and 3D model are apparent; the sky, for example, is erroneously projected onto the upper part of the building model. This misalignment is caused by pose tracking error. In our experiments, one degree of orientation-angle error results in over 10 pixels of alignment error in the image plane, an error that is easily visible and undesirable.

We overcome this problem by using an off-line vision process to stabilize the real-time tracked camera pose. Vision tracking is also helpful for overcoming GPS dropouts or occlusions. The image projection with the corrected pose is shown in Figure 7 (bottom).

The vision tracker is based on our prior work on fea-

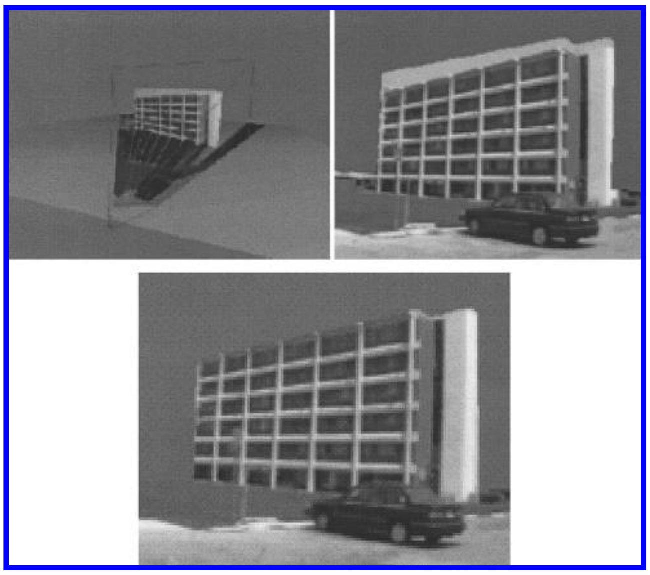

Figure 7. *Registration errors caused by direct use of the GPS-inertial tracking data to project video images onto a 3D model (top row). Image projection after camera pose is stabilized by vision tracking (bottom).*

ture tracking and auto-calibration (Jiang & Neumann, 2001), and adapted for this application. The system employs an Extended Kalman Filter (EKF) framework to simultaneously compute a camera pose from known features while estimating the structures of new (a priori unknown) scene features based on a prediction-correction strategy. Both line and point features in the scene are used for tracking. Straight line segments are prominent features in human-created environments, and can be detected and tracked reliably. In our approach, a line feature is modeled as an infinite 3D line and its observed line segments in different views correspond to different portions of the same line. Point features are also useful for tracking, especially when the user is close to building surfaces, since architectural lines may not be visible.

The EKF estimates a pose based on both line and point features. We represent the camera state as a 6-dimension vector of position, incremental orientation, and their first derivatives. The linear dynamic model is used for state prediction, and both line and point features are used as measurements for the EKF state update. For every new frame, the tracker first predicts the camera pose based on the prediction equation. The model features then are projected onto the image plane based on the prediction, and the discrepancies between projected features and observed features are used to refine the prediction.

Our camera is calibrated by using the method described in Zhang (2000). Once calibrated, the camera-internal parameters are assumed fixed during a tracking session.

6 Dynamic Object Analysis

A new addition to our system 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 model that enhances the visualization of moving objects.

Dynamic scene analysis and object extraction are traditional problems in computer vision. For AVE requirements, background subtraction is used, due to its effectiveness and efficiency. Many variations of this approach have been investigated. A major issue is the selection of an appropriate model for the background. Popular methods include simple averaging or estimation with a per-pixel Gaussian distribution or multimodal distributions. Ridder, Munklet, and Kirchner (1995) suggest a pixel-based Kalman filter for background estimation. Stauffer and Grimson (1999) use a per-pixel Gaussian mixture model for the background in order to obtain a robust background image. Mixtures of Gaussians can model multimodal backgrounds such as rotating fans, flickering monitors, and waving trees. Although these background-estimation methods address slow illumination changes and nonstatic backgrounds, they require careful estimation of the learning parameters for online processes.

Model-based approaches (Koller & Malik, 1994) 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

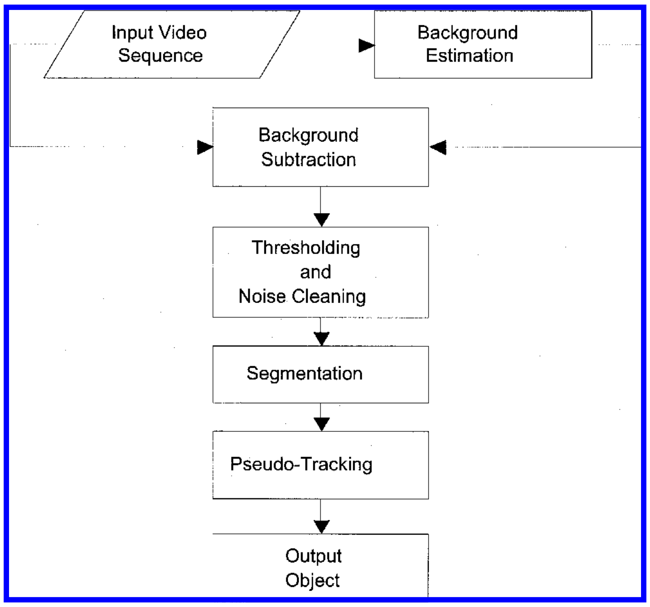

Figure 8. *Detection and tracking of moving objects.*

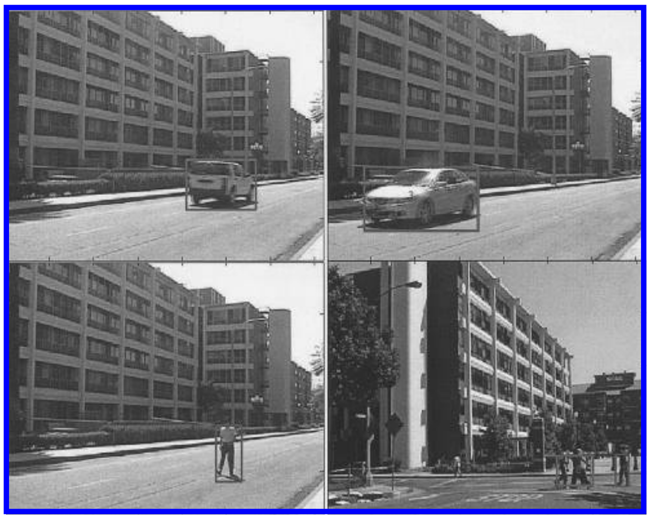

Figure 9. *Examples of tracked vehicles and people.*

be tracked. Remagnino et al. (1997) have two models, one for cars and one for humans, where the latter is a deformable model. Although these methods are not suitable for real-time detection and tracking, the use of a dynamic model is similar to our approach. Occlusions can also be managed with a dynamic layer representation for every detected object (Tao, Sawhney, & Kumar, 2000). This representation enables the system to track multiple objects, even when they overlap.

Our approach is a background-subtraction detection method followed by a pseudotracking algorithm. The choice of a relatively simple algorithm is motivated by a need for real-time processing. Figure 8 depicts the main components 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. Our experiments show that this offers performance that is similar to that of a single Gaussian distribution, with lower computation complexity.

Background subtraction is followed by a histogram-based threshold and morphological noise processing. The parameters of this step are roughly estimated by taking the quality of the video recorders and the size of the image into consideration. The threshold is set such that only the top 5% differences are labeled as objects. Objects smaller than 0.1% of the image size are eliminated as noise. Objects are segmented using a two-pass 4-connected component analysis for speed and simplicity.

Once objects are detected, pseudotracking is done by validation. We eliminate objects with validation of less than one second. This eliminates spurious new object detections for waving trees or other foliage. Validation is done by correlation matching between detected objects in neighboring frames. A correlation above a fixed threshold is accepted and the object is assigned a tag number.

The outputs of the detection and tracking system are the four-corner coordinates bounding the moving object regions in the 2D image plane. Figure 9 illustrates the results of applying this approach to track a moving vehicle and people around the University of Southern California campus.

6.1 Dynamic Object Modeling

Moving objects have no model in the static 3D scene model constructed for AVE visualization. We create a dynamic model for moving objects to address the

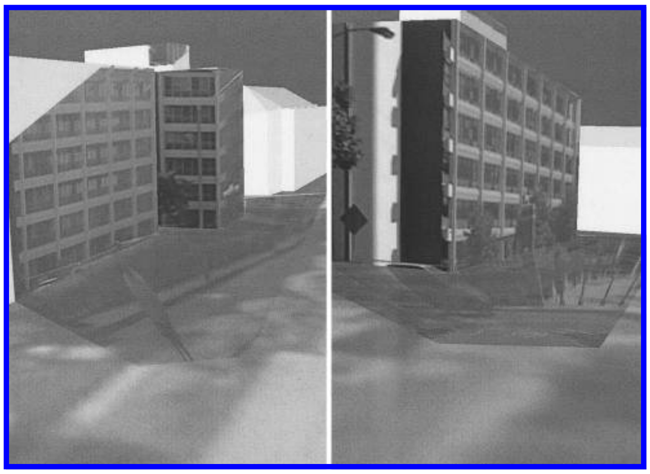

Figure 10. *Image projection of a person (left) and car (right) without dynamic models produces distorted presentation.*

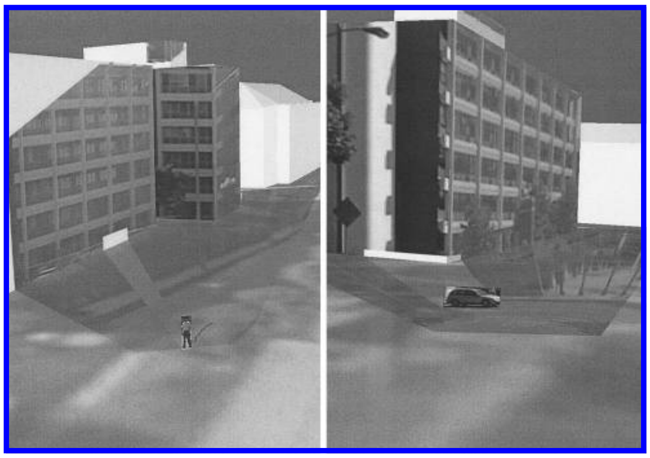

Figure 11. *Dynamic models automatically oriented and positioned at the 3D positions of the person and the car greatly improve the comprehension of video textures projected onto the model.*

distortion that occurs when simply projecting a video containing moving objects onto a static model. To illustrate this problem, consider an aerial camera that captures a moving person from a near-vertical aspect. The AVE projection of that imagery looks realistic when viewed from nearby aerial viewpoints; however, if viewed from a ground-level viewpoint, the person appears to be squashed and distorted since the imagery of the person is simply projected onto the flat street surface. We increase the fidelity of visualization by inserting approximate models (polygons, cuboids, or cylinders) at the 3D positions of the moving objects in the scene, thereby providing a dynamic model surface on which to project the images of moving objects.

We use the observation that tracked objects (people and vehicles) rest on the ground. The midpoint coordinate of the lower edge of the bounding box of a tracked object defines its contact point on the ground. A ray from the camera viewpoint through the contact point in the image is intersected with the ground model to determine the 3D position for a dynamic model of the moving object.

As shown in Figure 10, for example, a video sensor captures the dynamic movements of a person or a car in the scene. The video cameras are near ground level so their projections of a person or a car appear distorted without dynamic 3D scene models. The person and car appear to be "smeared" over the road and part of the building when viewed from the raised viewpoints in Figure 10. To reduce these distortions in AVE visualization, we add dynamic models in Figure 11. The paths of the walking person and the moving car are depicted by a red line to show their current positions and orientations in the 3D world. The video texture is projected onto the model as before and the visualization viewpoint is the same as for Figure 10. Users can freely navigate around the complete model.

There are others ways to acquire a dynamic 3D model of objects. Stereo methods are possible, but computing and camera costs are an issue, so single-camera solutions are preferable.

7 Dynamic Fusion and Imagery Projection

Traditional texture maps in VE's require that portions of each texture image are, a priori, associated with, and mapped onto, patches of the geometric model(s) before visualization can begin. In contrast to this fixed image-to-model association, an AVE system must associate texture images with the sensor and its pose within the model. The mapping between the model surfaces and imagery is computed dynamically as a result of texture projection during the rendering process. Changing the sensor pose automatically changes the mapping function. To implement the projection process, we need (1) a model of perspective image projection; (2) a strategy to handle the problem of visibility and occlusion; and (3) an accurate sensor model including camera parameters, projection geometry, and pose.

Projective texture mapping was introduced by Segal, Korobkin, Van Widenfelt, Foran, & Haeberli (1992). Although it was originally proposed only for shadows and lighting effects, it became extremely useful in many areas of computer graphics, image-based rendering, and visualization. While texture projection is a powerful approach to integrating dynamic imagery with 3D models, it produces textures on all surfaces within the frustum of projection. We wish to texture only the surfaces visible to the camera that captured the images, so visibility information must modulate the projection process (Teller & Seouin, 1991; Greene, Kass, & Miller, 1993).

The visibility calculation needs to be fast in order to support real-time visualization sessions. Fortunately, depth-map shadows (Reeves, Salesin, & Cook, 1987) offer an approach that is supported by many graphics cards, such as NVIDIA's Geforce GPUs that support 24-bit shadow maps. The depth-map facilitates a comparison of a projected depth value against the range component of a texture coordinate to determine if the surface point is visible or hidden from the sensor (Haeberli & Segal, 1993; Heidrich & Seidel, 1999). This approach requires two-pass processing, one pass for generating the depth image needed for comparisons, and a second pass for conditional image projection. We implement this approach utilizing hardware that supports SGI OpenGL extensions.

Each projection is applied sequentially using the model and the projection matrix operations. Projection visibility is computed for each sensor's viewpoint. Occluded model surfaces either keep their original colors or blend with other projections, depending on the application and user preferences. The projection also has to be clipped to the sensor's viewing frustum specified by the sensor parameters. This is implemented by using stencil operations, therefore masking the projection pass to the screen regions within the sensor frustum.

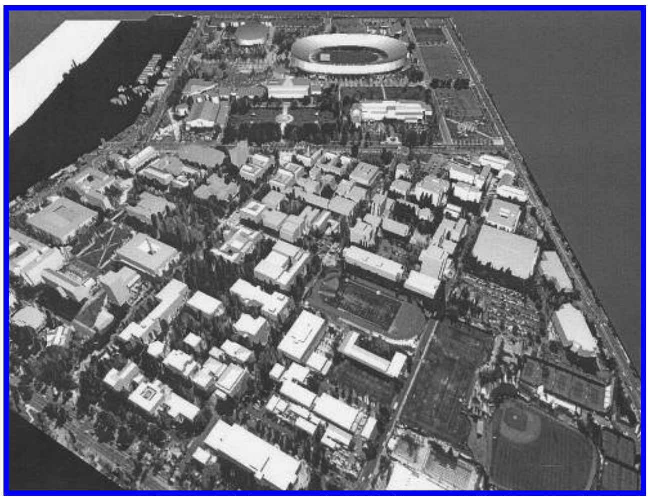

Figure 12. *Campus-area buildings embedded in terrain and vegetation models and in aerial ground texture.*

8 3D Visualization

The current AVE implementation achieves real time (~25 Hz) visualizations on a 2.2 GHz Pentium-4 workstation, supporting three live firewire video streams and high-resolution aerial photograph projections onto our entire University of Southern California campus model containing over 200 building models embedded in LiDAR terrain and vegetation models (Figure 12).

Figure 13 shows the current AVE quad-window presentation. The top-left window is the user's browsing window. It shows a user-controlled view of the full 3D geometric model and all camera data. Wireframe frustums are added to illustrate the camera positions and poses in the scene. Users typically fly around the scene using this window and select regions (or sensors) of interest. The other three windows display three selected camera viewpoints. These views can be locked to the selected camera (sensor) pose, making the images very similar to those obtained by the cameras. In addition to the video and model geometry, annotation that has

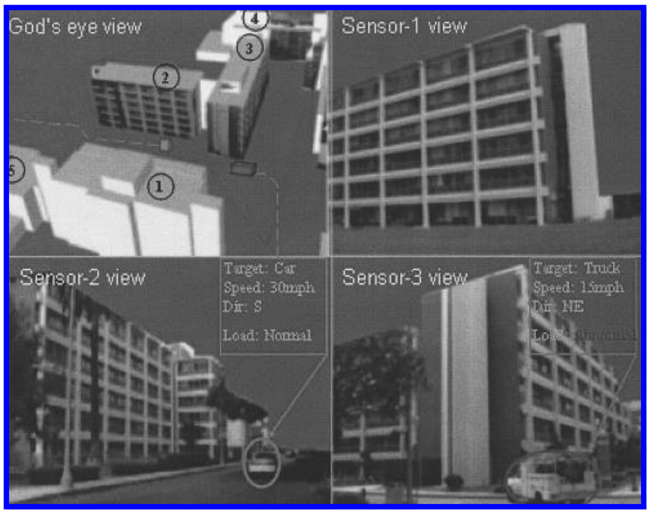

Figure 13. *Quad-window AVE display.*

added to the scene model by manual or automatic analysis appears in all windows. The four windows are simultaneous views and facilitate a user's interactive control during visualization sessions.

Our display system consists of an 8 × 10-foot screen, back-projected by a sequential-frame stereo video projector. A 3rdTech ceiling tracker is used to couple the rendering viewpoint to the user's head position. A hand tracker also facilitates mouse-like interactions. The overall system provides the user with a high performance AVE visualization environment.

9 Conclusion

Some comments are appropriate on the limitations of our AVE system. In addition to moving objects, other scene objects do not appear properly in the static model. Such objects include lamp poles mailboxes, and vegetation (trees, shrubs, etc.). These issues define much of the future work that is modeling related.

Another issue is the lack of image data in scene areas that the projection moved away from a few moments previously. Even casual users seem to want the image information to be persistent. Texture retention is difficult and largely unexplored in a real-time context. Also, the sky does not appear in our projections since there is no model for that imagery to project on. Our ongoing work on texture issues will need to address these issues.

Lastly, performance is an issue. The multipass projections and video bandwidth requirements of an AVE make it a computer- and data-intensive system that, if scaled up to modest levels of 20 or 50 concurrent video streams, would overwhelm any computer and graphics system available today. Methods are needed for efficiently managing the data selection and movement within a system.

We presented our methodologies and novel prototype of an AVE that supports dynamic fusion of imagery with 3D models. The core techniques we developed and integrated include model construction, sensor tracking, object tracking, dynamic modeling, and dynamic texture projection for 3D visualization.

We described implementation issues relating to the integration of the components and demonstrated the feasibility of an AVE that has the capability to capture, represent, and provide visualizations of dynamic spatiotemporal events and changes within a real environment.

Acknowledgments

This work was supported by a Multidisciplinary University Research Initiative (MURI) 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 our MURI team: Avideh Zakhor at UC Berkeley, William Ribarsky at Georgia Tech, Suresh Lodha at UC Santa Cruz, and Pramod Varshney at Syracuse University for their participation and contributions to the MURI project. We thank Todd Stennett at Airborne1 for providing us with the LiDAR data. We also thank Olympus Inc. for their support of this research.

References

Airborne1 Inc. http://www.airborne1.com

Azuma, R. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments, 6*(4), 355–385.

Debevec, P. E., Taylor, C. J., & Malik, J. (1998). Modeling and rendering architecture from photographs. A hybrid geometry and image-based approach. *Proceedings of SIGGRAPH 1996,* 11–20, New Orleans.

Foxlin, E. (1996). Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. *Proceedings of IEEE Virtual Reality Annual International Symposium,* 184–194.

Frueh, C., & Zakhor, A. (2001). 3D model generation for cities using aerial photographs and ground level laser scans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE CS Press, 2*(2): 31–38.

Greene, N., Kass, M., & Miller, G. (1993). Hierarchical Z-buffer visibility. *Proceedings of SIGGRAPH 1993,* 231–238.

Gruen, A., & Nevatia, R. (Eds.), (1998). *Special Issue on Automatic Building Extraction from Aerial Images, Computer Vision and Image Understanding.*

Haeberli, P., & Segal, M. (1993). Texture mapping as a fundamental drawing primitive. In M. F. Cohen, C. Puech, & E. Sillion (Eds.), *Fourth Eurographics Workshop on Rendering,* 259–266.

Hall, B., & Trivedi, M. (2002). A novel graphical interface and context aware map for incident detection and monitoring. *Proceedings of the 9th World Congress on Intelligent Transport Systems.*

Heidrich, W., & Seidel, H.-P. (1999). Realistic, hardware-accelerated shading and lighting. *Proceedings of SIGGRAPH 1999.*

Hsieh, Y. (1996). SiteCity: A semi-automated Site Modeling System. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 499–506.

Jiang, B., & Neumann, U. (2001). Extendible tracking by line auto-calibration. *Proceedings of the International Symposium on Augmented Reality,* 97–103.

Kanade, T., Collins, R., Lipton, A., Burt, P., & Wixson, L. (1998). Advances in cooperative multi-sensor video surveillance. *Proceedings of DARPA Image Understanding Workshop, 1,* 3–24.

Kanade, T., Rander, S., Vedula, S., & Saito, H. (1999). *Mixed reality, merging real and virtual worlds.* In Yuichi Ohta and Hideyuki Tamura (Eds.) (pp. 41–57). New York: Springer Verlag.

Koller, W., & Malik, J. (1994). Robust multiple car tracking with occlusion reasoning. *Proceedings of the 3rd European Conference on Computer Vision,* 189–196.

Kumar, R., Sawhney, H. S., Guo, Y., Hsu, S., & Samarasekera, S. (2000). 3D manipulation of motion imagery. *Proceedings of the International Conference on Image Processing* (ICIP 2000), Vancouver.

Lee, S. C., Jung, S. K., & Nevatia, R. (2002). Automatic integration of façade textures into 3D building modelings with projective geometry based line clustering. *EUROGRAPHIC 2002, 21*(3): 511–519.

Neumann, U., & You, S. (1999). Natural feature tracking for augmented-reality. *IEEE Transactions on Multimedia, 1*(1), 53–64.

Neumann, U., & You, S. (2003). Augmented Virtual Environments (AVE): Dynamic Fusion of Imagery and 3D Models. *Proceedings of IEEE Virtual Reality 2003,* 61–67.

Reeves, W., Salesin D., & Cook, R. (1987). Rendering anti-aliased shadows with depth maps. *Computer Graphics, 21*(4): 283–291.

Remagnino, R., Baumberg, A., Grove, T., Hogg, D., Tan, T., Worrall, A., & Baker, K. (1997). An integrated traffic and pedestrian model-based vision system. *Proceedings of BMVC 1997, 2,* 380–389.

Ribarsky, W., Wasilewski, T., & Faust, N. (2002). From urban terrain models to visible cities. *IEEE Computer Graphics & Applications, 22*(4): 10–15.

Ridder, C., Munklet, O., & Kirchner, H. (1995). Adaptive background estimation and foreground detection using Kalman filtering. *Proceedings of ICRAM 1995,* 193–199.

Segal, M., Korobkin, C., Van Widenfelt, R., Foran, J., & Haeberli, P. (1992). Fast shadows and lighting effects using texture mapping. *Proceedings of SIGGRAPH 1992,* 249–252.

Spann, J. R., & Kaufman, K. S. (2000). Photogrammetry using 3D graphics and projective textures. *IAPRS 2000,* 33.

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1999), 2,* 246–252.

Tao, H., Sawhney, H. S., & Kumar, R. (2000). Dynamic layer representation with applications to tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000), 2,* 134–141.

Teller, S. J., & Seouin, C. H. (1991). Visibility preprocessing for interactive walkthroughs. *Proceedings of SIGGRAPH 1991,* 61–69.

You, S., Neumann, U., & Azuma, R. (1999). Orientation tracking for outdoor augmented reality registration. *IEEE Computer Graphics & Applications, 19*(6), 36–42.

You, S., Hu, J., Neumann, U., & Fox, P. (2003). Urban site modeling from LiDAR. *Second International Workshop on Computer Graphics and Geometric Modeling (CGGM 2003),* 579–588.

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