Augmented Virtual Environments (AVE): Dynamic Fusion of Imagery and 3D Models

Ulrich Neumann, Suya You, Jinhui Hu, Bolan Jiang, and JongWeon Lee
Integrated Media Systems Center
University of Southern California
{uneumann | suyay | jinhuihu | bjiang | jonlee}@graphics.usc.edu

Abstract

An Augmented Virtual Environment (AVE) fuses dynamic imagery with 3D models. The AVE provides a unique approach to visualize and comprehend multiple streams of temporal data or images. Models are used as a 3D substrate for the visualization of temporal imagery, providing improved comprehension of scene activities. The core elements of AVE systems include model construction, sensor tracking, real-time video/image acquisition, and dynamic texture projection for 3D visualization. This paper focuses on the integration of these components and the results that illustrate the utility and benefits of the resulting augmented virtual environment.

1. Introduction

Three-dimensional Virtual Environment (VE) models are used for engineering, training simulations, entertainment, tactical planning, and military operations in battlefield environments. In many cases, the value of the VE is increased if both its geometry and appearance are accurate and realistic analogues of the real world.

While current technologies offer useful methods for VE creation and visualization, several significant limitations remain. First, most modeling systems facilitate geometric modeling through the manipulation of either standard geometric primitives, libraries of pre-modeled objects, or manual digitizing of key points. Creating accurate and realistic models of a real environment in this fashion takes enormous effort, skill, and time, resulting in painfully slow evolution of such databases. Commercially available models of a city block, for example, can take several weeks to create by several people working together.

Second, while most modeling systems support texture mapping, they are limited to static texture databases that must be created prior to use. Static textures are usually derived from camera images with known or computed transformations relative to the modeled objects. The creation and management of such texture databases is also time consuming since it includes image capture and the creation of mapping functions for each corresponding image and model patch [9]. Such static texture-maps are inappropriate for applications requiring a dynamic picture of the environment. In time-critical applications, such as military command and control, person or vehicle tracking, facility security, and catastrophe management, a real-time accurate fusion of dynamic imagery and geometric data is desirable. Real time video or other sensor data from multiple and possibly moving sources needs to be mapped onto the VE models. There is little or no support for such dynamic spatio-temporal updates in the current structure of VE models, databases, and rendering systems.

To cope with the aforementioned limitations of static models and visualizations of environments, we introduce the concept of an Augmented Virtual Environment (AVE) for capturing, representing, and visualizing the dynamic spatio-temporal events occurring within a real environment. The AVE is a virtual environment augmented by the fusion of dynamic imagery onto the 3D models.

This paper focuses on the integration of the components needed to create an AVE system and the presentation of results showing the AVE utility and benefits. Due to the system scope, in-depth descriptions of many of the system components are offered in the cited sources.

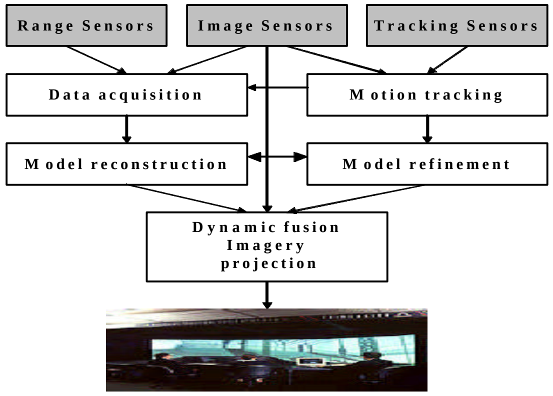

Fig. 1 - AVE system components

2. System overview

Figure 1 depicts the six main components of our AVE system: (1) data acquisition to collect real time geometry and imagery measurements; (2) model reconstruction to obtain a single 3D surface model from the sets of acquired geometric measurements; (3) model refinement to segment structures and extract dominant scene features; (4) tracking to provide image-sensor pose and motion data for registration and data fusion; and (5) data fusion to combine all manner of models, images, video, and data in a coherent visualization to support improved understanding, information extraction, and dynamic scene analysis.

3. Model acquisition and reconstruction

There are many methods for acquiring real world data for creating scene models [1-8]. An acquisition phase usually collects varied sensor data, often measuring 3D surface-coordinates from range scanners or color data from image sensors. A reconstruction phase then processes the sensor data (resampling, hole-filling, and tessellation) into a consistent form suitable for visualization. Texture-maps of static imagery are also mapped onto geometric models to produce photorealistic visualizations [2, 3, 4].

Our data acquisition was a collaboration with Airborn1 Inc. [27], employing a LiDAR (Light Detection and Ranging) system [27] 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 ensure good coverage. The end result is a cloud of 3D point samples that we project and resample onto a regular grid (~0.5 meter 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 [26] (Figure 2).

Triangle meshes were selected as the final 3D geometric representation since they are easily converted to many other geometric representations; whereas the reverse is not always true [9]; many level-of-detail techniques operate on triangle meshes; photometric information can easily be added to the data in the form of texture projections; and finally, graphics hardware directly supports fast image rendering from meshes.

4. 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 semi-automatically extract and refine building models from LiDAR data. The details of the algorithms are presented in [26]. 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 semi-automatic 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 3 illustrates the results of model refinement; the primitives are automatically fit to the LiDAR data, so user mouse-click accuracy is not critical. Figure 4 shows models we created for the entire University Park area including the USC campus, LA Natural History Museum, Science Museum complex, LA Coliseum, and Sports Arena. Note the presence of curved surfaces and multiple primitives for buildings.

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.

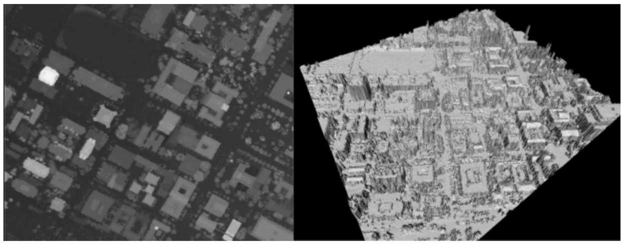

Fig. 2 - LiDAR acquired for USC campus: (left) resampled range image, (right) reconstructed 3D mesh

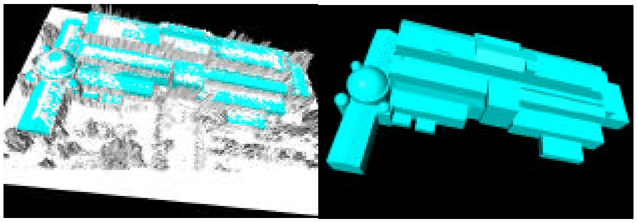

Fig. 3 - Model verification shown (left) by fit primitives embedded in the original LIDAR data, and (right) extracted building model

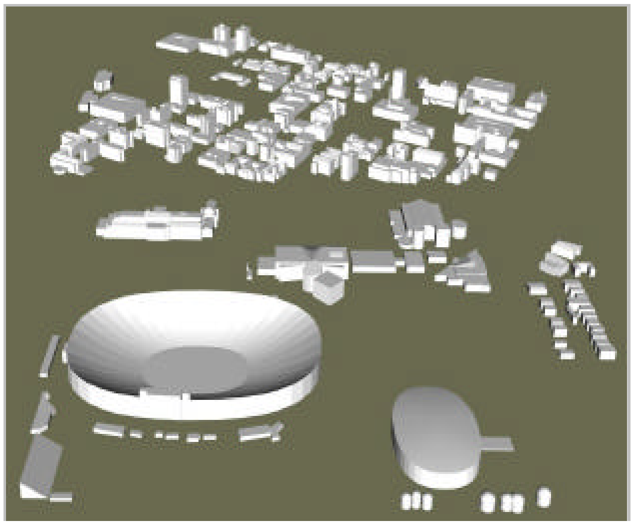

Fig. 4 – University Park models extracted from LiDAR

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 [13, 14]. Methods employing a single tracking sensor have limitations; hybrid systems use multiple sensor measurements to produce more robust results [15].

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 5). 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 only uses one camera for video acquisition and vision tracking. The stereo stream may be used for detailed 3D building façade reconstruction, in the future.

The differential mode (DGPS) uses two RTK units (base and remote) that communicate via a spread-spectrum radio to perform position calculations to about 2-10 centimeter accuracy at 2-5 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 [16]. This orientation tracker is specified as achieving approximately 1°-3° accuracy, with 150Hz maximum update rate.

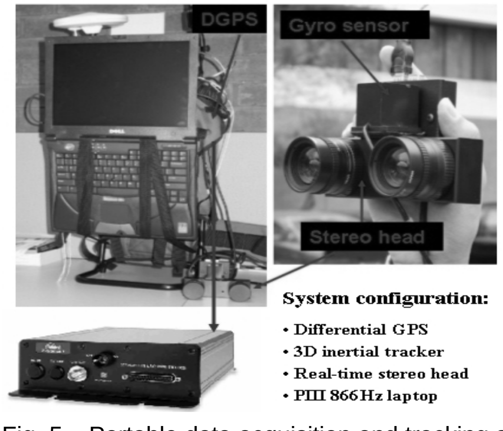

Fig. 5 – Portable data acquisition and tracking system

The tracking and video acquisition systems run in real-time and their data is stored onto hard-disk. We synchronize and resample all these data streams at the 30Hz 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 6 illustrates the typical dynamic registration errors that arise from direct use of the GPS/inertial tracking data to project images onto the 3D model. The 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 center image shows the projected image-texture 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 prior analysis and experiments [15], one degree of orientation angle error results in about 11-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 tracker 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 6 (right).

The vision tracker is a based on our prior work on feature auto-calibration [12], and adapted for this application. The

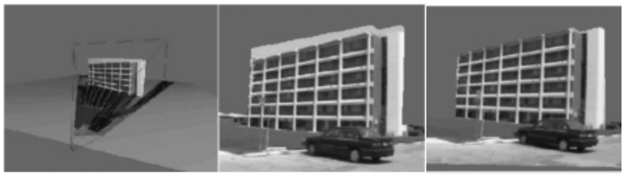

Fig. 6 – Registration errors caused by direct use of the GPS/inertial tracking data to project a video image onto the 3D model (left and center).

Image projection after camera pose is stabilized by vision tracking (right) system employs an Extended Kalman Filter (EKF) framework to extend tracking range from an initial calibrated area to neighboring uncalibrated areas. A key capability of this method is that starting from a known initial estimate of camera pose (obtained from any method or sensors such as a GPS-inertial tracker), the camera pose is continually estimated using naturally occurring 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 man-made environments, and these 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 pose based on both line and point features. We represent the camera state as a 6-dimention 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 [25]. Once calibrated, the camera internal parameters are assumed fixed during a tracking session.

6. Dynamic fusion and imagery projection

The main benefit of the AVE concept is the dynamic fusion of multiple sources of geometric models, images, video, and other sensing information. Most VE and visualization systems provide only static fusion of geometric and photometric information. In [3, 4], for example, Debevec et al created a system for modeling and rendering photo-realistic architectural scenes from photographs. This approach produced 3D models of buildings and used high resolution photographs for texture mapping the models. There is however no provision to accommodate video from cameras moving in the environment.

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 (as discussed in section 5).

Projective texture mapping was introduced in [17]. 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, there are potential pitfalls. Simply applying texture projection produces textures on all surfaces within the frustum of projection, while we only want to texture the surfaces visible to the camera that captured the images. So, visibility information must modulate the projection process [18-21].

The visibility calculation needs to be fast in order to support real-time visualization sessions. Fortunately, depth-map shadows [22] offer an approach to fast visibility detection that is supported by many graphics cards, such as NVIDIA's Geforce-3 GPU that supports 24-bit shadow maps.

The depth-map shadows produce a depth map that 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 [22, 23, 24]. This approach requires two-pass processing, one for generating the depth image needed for comparisons, and a second pass for conditional image projection. We implement this approach utilizing graphics hardware that supports SGI OpenGL extensions. A P4 2GHz system achieves real time rendering (26 Hz) of 1280x1024 images with four texture streams projected onto our campus model.

Some implementation details help clarify the projection process. The video imagery and sensor pose streams are loaded into memory and converted to the formats required for texture projection. These streams are synchronized during the data acquisition process so that each frame of the video has a sensor tag associated with it, including the sensor's internal parameters and 6DOF pose. The sensor's internal parameters specify the virtual projector's projection, and the pose data specifies the projector's position and orientation. Each projection is applied sequentially using the model and projection matrix operations. Projection visibility is computed for each sensor's view point. 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 the stencil operations [9]: drawing the outside of the frustum with a stencil increment operation, and the inside with a stencil decrement operation, therefore masking the projection pass to the screen regions within the sensor frustum.

7. 3D Visualization environment

Figure 7 shows a snapshot of our AVE system projecting a video stream onto a building model. To maximize the visualization effectiveness and observe the stages in the process of data fusion, we designed a quad-window display interface; the top-left window shows the 3D geometric model and wireframe camera frustum. The top-right window displays a selected camera's video imagery prior to projection onto the 3D model. These two windows represent the separate geometry and imagery prior to fusion. The lower-left window shows the fusion result viewed from a novel viewpoint, and the lower-right window shows a rendered view from the sensor viewpoint. The four windows are simultaneous views and facilitate a user's interactive control during visualization sessions.

7.1 Visualizing multiple images simultaneously

While a single image stream fused with a 3D scene model has some compelling qualities, the true value of AVE visualizations becomes clearer when one considers multiple simultaneous streams. Humans do not fuse images from disparate viewpoints very well. Given just two or three static images of a building from different viewpoints, we are easily disoriented and have difficulty understanding where the cameras are and where they are looking. This becomes an overwhelming task if we increase the number of cameras (to say 20), with many of them moving within the environment. The AVE approach provides the 3D substrate that we need to tie all the views together, enabling us to understanding the cameras, their images, and their movements.

In many applications involving multiple video images today, a separate screen is used for each camera image. The cognitive load this presents to a user can be overwhelming. Information sensors are likely to increase as computing, sensing, and communications pervade our world. The problem of making all this information digestible and understandable depends on having a coherent presentation that allows a human user to easily understand relationships and switch focus between levels of detail and specific spatial or temporal aspects of the data. The AVE approach facilitates such browsing among multiple video streams since it presents all imagery in a single context, allowing users free selection of their viewpoint.

Figure 8 shows three sensors simultaneously projecting onto an area around a building. The upper-left image shows the fused information from a novel viewpoint. User's can see projected images from the camera's viewpoints in the other three windows.

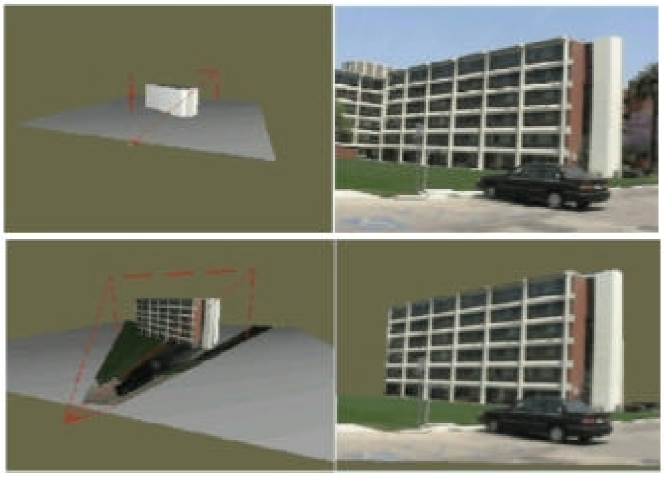

Fig. 7 – Real-time video is projected onto the 3D model to create a dynamic AVE visualization from several viewpoints. The upper right image shows the initial camera images.

We integrated the techniques described in this paper and constructed a room-size 3D visualization screen. The system is used for demonstrations of the AVE concept and as a test-bed for algorithms. The display consists of an 8x10 foot screen, back-projected by a sequential-frame stereo video-projector. A 3rdTech ceiling tracker is used to couple the rendering viewpoint to user's head position. A tracker also facilitates mouse-like interactions. The overall system provides the user with a high performance AVE visualization environment.

8. Conclusion

In closing some comments are appropriate on the limitations of our initial AVE system. Clear to everyone we've showed the system to is the inability to properly display objects that are not part of the model. For example, lamp poles, cars, and trees are projected onto the buildings and roads, and they look warped and distorted from other viewpoints. Oddly, however, there seems to be a human ability to adapt to those distortions – perhaps because they are local distortions and since the projector frustums are visible, people seem to have little difficulty or discomfort dealing with those effects. We understand that more rigorous human testing is called for before sweeping or quantifiable benefits can be claimed.

Another issue is the lack of image data in scene areas that the projection moved away from a few moments ago. Even casual users seem to want the image information to be persistent. Also, the sky does not appear in our projections since there is no model for that imagery to project on. Our ongoing work is addressing all of these issues.

Lastly, performance is an issue. The multipass projections and video bandwidth requirements of an AVE make it a

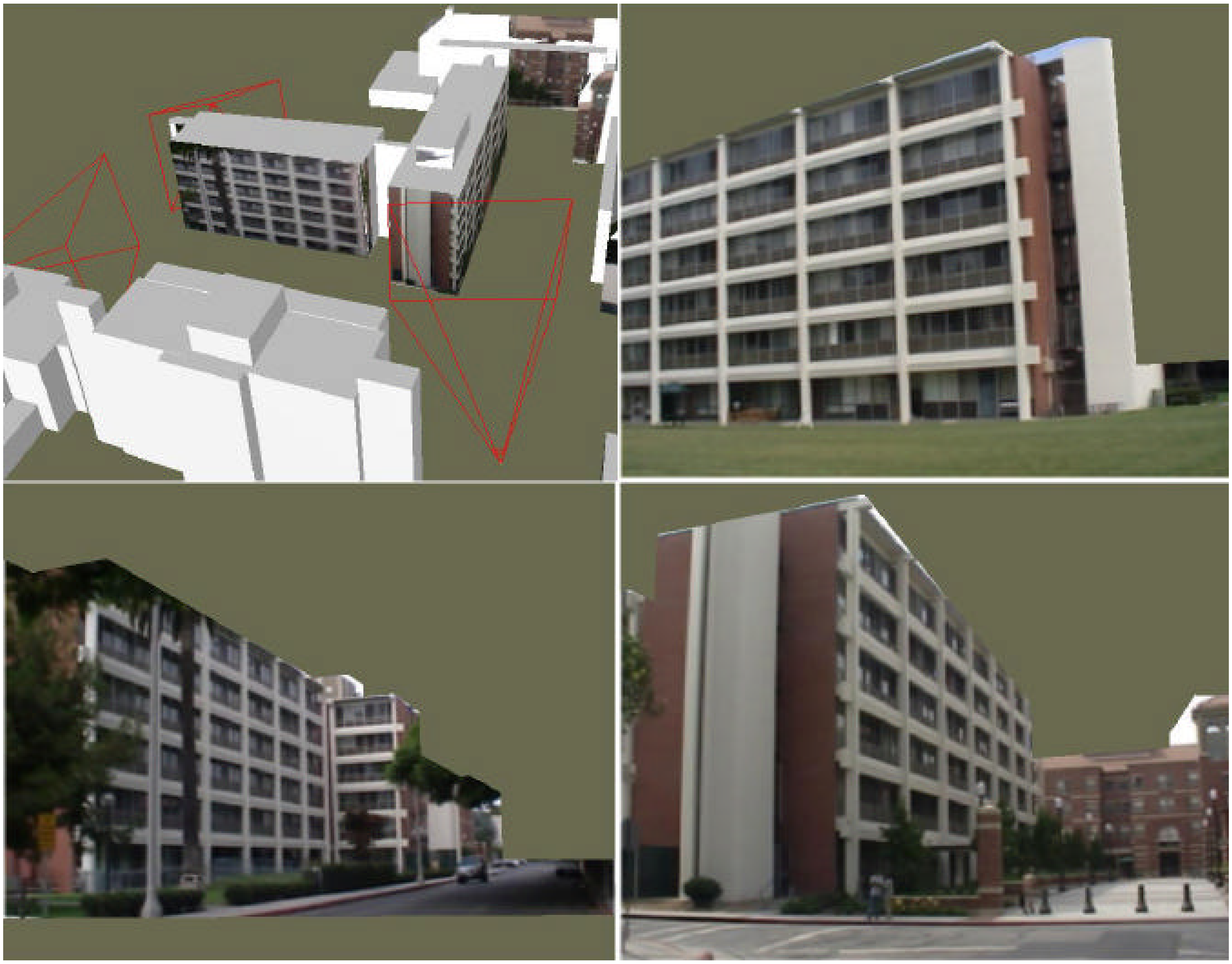

Fig. 8 – An AVE view of three video projections within one campus area. The top-left window shows a novel view and the three remaining windows show rendered views from the three camera (projection) viewpoints.

compute and data-intensive system that, if scaled up to modest levels of 20 video streams, would stress any compute and graphics systems available today.

We presented our methodologies and novel prototype of an augmented virtual environment (AVE) that supports dynamic fusion of imagery with 3D models. The core techniques we developed and integrated include model reconstruction, model refinement, building extraction, sensor tracking, real-time video/image acquisition, and dynamic texture projection for 3D visualization. We presented methodologies for the rapid creation of realistic geometric models from LiDAR data and the projection of dynamic imagery from multiple image sources. We developed hybrid 6DOF tracking system with integrated GPS, inertial, and vision tracking technologies and applied these for dynamic data fusion and sensor registration.

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 spatio-temporal events and changes within a real environment.

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 our MURI team: Avideh Zakhor at UC Berkeley, William Ribarsky at Geogia 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 Airborn1 Inc for providing us with the LiDAR data. Our thanks also go to HP, NVIDIA, Intel, and Microsoft for their support.

References

1. C. Fruh and A. Zakhor, “3D Model Generation for Cites Using Aerial Photographs and Ground Level Laser Scans”, *IEEE Conference on Computer Vision and Pattern Recognition*”, 2001.
2. S. C. Lee, S. K. Jung and R. Nevatia, “Automatic Integration of Façade Textures into 3D Building Modelings with Projective Geometry Based Line Clustering”, *EUROGRAPHIC’02*, 2002.
3. P. E. Debevec, C.J.Taylor and J. Malik, “Modeling and Rendering Architecture from Photograph: A Hybrid Geometry-and Image-based Approach”, *SIGGRAPH’ 96*, pp. 11-20, 1996.
4. P .E. Debevec, Y. Yu, and G.D. Borshukov, “Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping”, *9th Eurographics Workshop on Rendering*, pp. 105-116, 1998.
5. Y. Hsieh, “SiteCity: A semi-automated Site Modeling System”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 499-506, 1996.
6. D. Liebowitz, A.Criminisi and A. Zisserman, “Creating Architectural Modeling From Images”, *EUROGRAPHI*C’99, pp. 39-50, 1999.
7. William Ribarsky, Tony Wasilewski, and Nickolas Faust, “From Urban Terrain Models to Visible Cities,” *IEEE Computer Graphics & Applications*, Vol. 22, No. 4, 2002.
8. A Gruen and R. Nevatia (editors), “Special Issue on Automatic Building Extraction from Aerial Images”, *Computer Vision and Image Understanding*, November 1998.
9. Foley, J. D., Van Dam, A., Feiner, S. K. and Hughes, J. F., “Computer Graphics: principles and practice”, Addison-Wesley, Reading, Massachusetts, 1990.
10. M.J.Zyda and D.R.Pratt, “NPSNET: A 3D Virtual Simulator for Virtual World Exploration and Experience”, *Tomorrow’s Realities Galley, Visual Processings of SIGGRAPH91*, 1991.
11. Cruz-Neira, C., Sandin, D.J. and DeFanti, T.A, “Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE”, *SIGGRAPH’93*, 1993.
12. B. Jiang, U. Neumann, “Extendible Tracking by Line Auto-Calibration,” *International Symposium on Augmented Reality*, pp.97-103, New York, October 2001.
13. R. Azuma, “A Survey of Augmented Reality”, *Presence: Teleoperators and Virtual Environments.* Vol. 6, No.4, pp. 355-385, 1997.
14. Ulrich N. and Suya Y., "Natural Feature Tracking for Augmented-Reality", *IEEE Transactions on Multimedia*. Vol. 1. No.1, 1999.
15. Suya Y., Ulrich N. and Ronald A., "Orientation Tracking for Outdoor Augmented Reality Registration", *IEEE Computer Graphics & Applications*, Vol. 19, No. 6, Nov. 1999.
16. E. Foxlin. “Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter”, *IEEE Virtual Reality Annual International Symposium*, pp. 184-194, 1996.
17. Segal, M., Korobkin, C., Van Widenfelt, R., Foran, J. and Haeberli, P., “Fast shadows and lighting effects using texture mapping”, *SIGGRAPH ’92*, pp. 249–252, 1992.
18. Teller, S. J. and Seouin, C. H. “Visibility preprocessing for interactive walkthroughs”, *SIGGRAPH ’91*, pp. 61–69, 1991
19. B. Curless, and M. Levoy, “A volumetric method for building complex models from range images”, *SIGGRAPH’96*, pp. 303-312, 1996.
20. N. Greene, M. Kass, and G. Miller, Hierarchical Z-Buffer Visibility. *SIGGRAPH’ 93*, 231-238, 1993.
21. Mark, W. R., Mcmillan, L. and Bishop, G. Post-rendering 3D warping. *ACM Symposium on Interactive 3D Graphics*, pp. 7–16, 1997.
22. William Reeves, David Salesin and Robert Cook, “Rendering Antialiased Shadows with Depth Maps”, *Computer Graphics*, Volume 21, Number 4, July 1987.
23. Paul Haeberli and Mark Segal, “Texture mapping as a fundamental drawing primitive”, in Michael F. Cohen, Claude Puech, and Francois Sillion, editors, *Fourth Eurographics Workshop on Rendering*, pp. 259-266, 1993.
24. W. Heidrich and H.-P. Seidel, “Realistic, Hardware-accellerated Shading and Lighting”, *SIGGRAPH '99*, 1999
25. Z. Zhang, “A flexible new technique for camera calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No.11, pp. 1330-1334, 2000.
26. J. Hu, S. You, and U. Neumann, “Extract Complex Buildings from LIDAR Data”, Tech. Report, pending number, Computer Science Department, University of Southern California, 2002.
27. http://www.airborne1.com.