Haptics in Augmented Reality

James Vallino
Department of Computer Science
Rochester Institute of Technology
jrv@cs.rit.edu

Christopher Brown
Department of Computer Science
University of Rochester
brown@cs.rochester.edu

Abstract

An augmented reality system merges synthetic sensory information into a user's perception of a three-dimensional environment. An important performance goal for an augmented reality system is that the user perceives a single seamless environment. In most augmented reality systems the user views a real world augmented only with visual information and is not provided with a means to interact with the virtual objects. In this paper we describe an augmented reality system that, in addition to visual augmentation, merges synthetic haptic input into the user's perception of the real environment. Our system uses a ***PHANToM***™ *haptic interface device to generate the haptic sensory input in real-time. The system allows user interactions such as moving or lifting a virtual object, and demonstrates interactions between virtual and real objects. Methods to provide proper visual occlusion between real and virtual objects are also described.*

1. The Challenge of Augmented Reality

There has been considerable interest in augmented reality (AR) systems that mix live video from a camera with computer-generated objects registered in a user's three-dimensional environment [1]. Applications of this powerful visualization technique include maintenance tasks [2], surgical planning [3, 4], and new user interfaces [5]. The resulting AR systems allow three-dimensional virtual objects to be visually embedded into a user's perception of the environment.

In a typical AR system, Figure 1, a video camera views the real scene and generates a 2D image of it on the image plane. The user sees an augmented view composed of a synthetic graphics image merged with the image of the real scene. To maintain the illusion that the virtual objects are indeed part of the real world requires a consistent registration of these two images—the major challenge for augmented reality systems.

This registration requirement for creating a high-fidelity augmented reality environment can be stated in terms of the relationships that must be determined and maintained (Figure 1). The object-to-world transform **O**, specifies the position and orientation of a virtual object with respect to the world coordinate system that defines the real scene. The world-to-camera transform **C**, defines the location and orientation of the video camera that views the real scene. Finally, the camera-to-image plane transform **P**, specifies the projection operation the camera performs to create a 2D image of the 3D real scene. Any errors in the determination of these relationships appear to the user as inconsistencies in the appearance and position of the virtual objects in the real scene.

To faithfully create haptic interactions with the virtual objects there is also a registration problem between the real world and the system generating the haptic display. There is a haptic-to-world transform that defines the relationship between the world coordinate system and the coordinate system in which the haptic interface operates. Accurately computing these relationships while maintaining real-time response and a low latency is the primary performance goal for a haptically and visually augmented reality system.

2. Augmenting Reality Using Affine Representations

Only when the relationships between the multiple coordinate systems shown in Figure 1 are known can the synthetic sensory information correctly merge into the user's perception of the real scene. Traditional AR systems approach the problem of computing these transforms using sensing, calibration and measurement to explicitly determine each transform [6]. These systems use sensors to measure the camera's pose with respect to the world coordinate system thus determining the world-to-camera transform, **C**. Quantifying the

---

The support of the National Science Foundation under Grant No. CDA-94-01142 and the Defense Advance Research Projects Agency under Grant No. DAAB07-97-C-J027 is gratefully acknowledged.

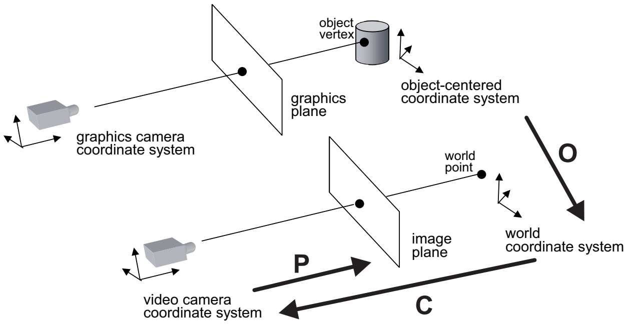


Figure 1 - Augmented reality coordinate systems

camera-to-image transform, **P**, requires knowledge of the camera's intrinsic parameters [7]. The third transform, **O**, is computed by measuring the desired position for the virtual object in the world coordinate system. From this all the necessary transforms are known so that, at least in principle, virtual objects can be rendered and merged correctly with the live video.

The methods based on position measurements exhibit errors due to inaccuracies and latencies in position sensing, and errors in the camera calibration parameters. A novel aspect of our augmented reality system (not pursued here) is that it requires no a priori metric information about the intrinsic and extrinsic parameters of the camera, where the user is located in the world, or the position or geometry of objects in the world [8]. In our system we track four features in real-time and define the global affine coordinate system solely from the location of those tracked features in the video image. All relationships are determined in this common affine coordinate system

3. An Haptic Augmented Reality Interface

None of this prior work has included any interaction with the virtual objects except for the visual changes in the augmented reality display whenever the user changes viewpoint. One of the reasons stated by Mine, Brooks, et. al. [9] for the paucity of virtual-environment applications that have left the laboratory setting is the lack of haptic feedback. Previous haptic research is concentrated in the areas of telemanipulation and virtual reality. The work in these areas does not, however, provide insights into the problems of registration with the real scene or interactions between real and virtual objects.

3.1. Haptic technology

Haptic technology provides the user with an ability to experience touch sensations. We emphasize user interaction with a natural-seeming augmented environment requiring the user to have a sense of feeling the virtual objects, touching their surface, and interacting with them in a dynamic fashion. Some work in virtual reality applications demonstrated haptic interfaces. In Project GROPE at the University of North Carolina [10], the user explores molecular docking by manipulating molecules in an immersive virtual environment using a large-scale force-reflexive manipulator. Ziegler, Brandt, et. al. [11] describe a simulator for arthroscopic surgery that uses force feedback in the virtual environment to train surgeons in the procedure. Using the Rutgers Master II force feedback device Dinsmore, Langrana, et. al. [12] built a virtual environment for training physicians to locate and palpate tumor masses.

The work of State, Hirota et. al. [13] shows interaction with virtual objects. There is neither haptic feedback nor dynamic interactions between the virtual and real objects however. Yokokohji, Hollis et. al. [14] demonstrate a haptic interface for an AR system with neither motion of virtual objects nor interaction between virtual and real objects in their system.

To allow realistic interactions with the virtual objects we chose the **PHANToM™** [15]. This device looks similar to a small robot arm with a motor driving each joint. The controller drives the motors to generate the requested force feedback at the end effector, a thimble into which the user inserts a finger. With the supplied GHOST™ library the system defines a world of objects in a haptic scene to be "rendered". Mass can be assigned to the objects so when the user places a finger under an object and lifts, the weight of the object is felt resting on the finger. It is also possible to simulate object surface compliance and texture. Figure 2 shows the system diagram for our haptic AR system.

3.2. Haptic-graphic interaction

In Section 2 we described the method that we use for registering the world and graphics coordinate systems. We accomplish this registration by defining a common

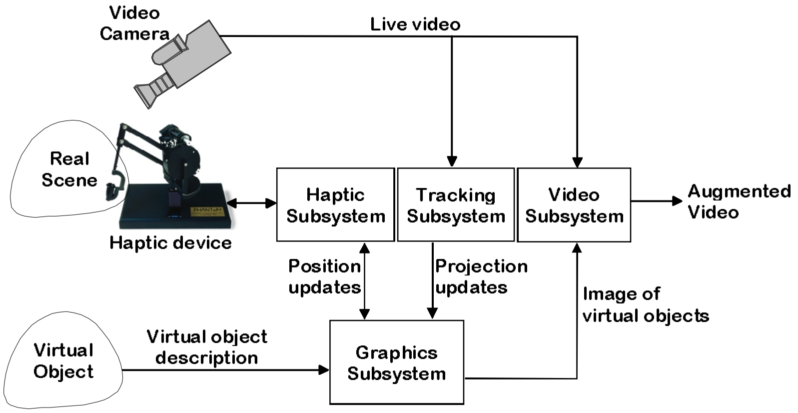


Figure 2 - Components of Augmented Reality System

global affine coordinate system. The system creates a scene graph containing the virtual objects defined in an Euclidean coordinate system. We compute a transform from this coordinate system to the global affine coordinate system as part of the process to place virtual objects in our real scene. The GHOST library allows definition of a haptic scene in a manner analogous to defining the graphic scene used for generating the virtual objects.

Our haptic AR system must establish the relationship between the Phantom's haptic coordinates and the global affine coordinates in which all of the virtual graphic objects are described. Once we establish that relationship it is possible to exchange position information between the haptic and graphic scene so that our system can instruct the Phantom to generate haptic feedback to the user appropriate for the interactions with the virtual objects.

Our system computes the Phantom to affine transform, $\mathbf{T}_{ap}$, by having the user move the Phantom end effector to four points in the workspace for which affine coordinates are known. The system records the position in the Phantom coordinate system at each point. $\mathbf{T}_{ap}$ is computed by solving:

$$[\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3 \quad \mathbf{a}_4] = \mathbf{T}_{ap}[\mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3 \quad \mathbf{p}_4]$$

where $\mathbf{a}_i$ and $\mathbf{p}_i$ are, respectively, the affine coordinates of the points in the workspace and their corresponding haptic coordinates.

A tight coupling exists between the graphics scene and haptic scene as Figure 3 depicts. At each cycle of operation the system obtains the current Phantom position and transforms it into the global affine coordinate system. This affine point is then transformed into the Euclidean coordinate system of the virtual object using an inverse of $\mathbf{T}_{ap}$. The location of this point on the virtual object determines the appropriate haptic response. The system sends the correct commands to the Phantom to generate the feedback at the user's fingertip, so textures that look different also feel different.

3.3. Virtual-to-virtual haptic interaction

The system allows several different interactions with virtual objects. (MPEG video clips of all the demonstrations described in this paper can be found at http://www.cs.rit.edu/~jrv/research/ar/.) In a typical demonstration, a virtual globe appears in the augmented image. When the globe is stationary (Figure 4) the user

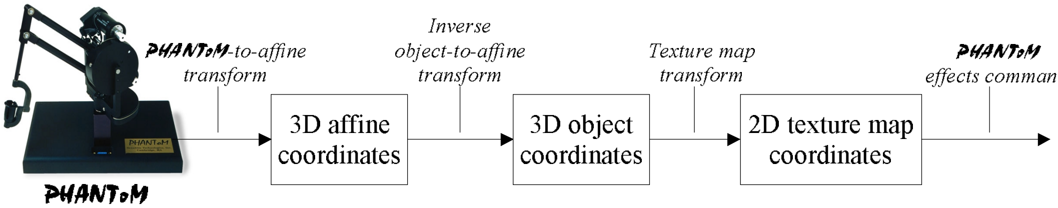


Figure 3 - Phantom-graphics coupling

can move a finger over the surface and receive correctly registered haptic sensations to help distinguish water and land masses. Another demonstration spins the globe on its axis. The system tracks the location of the active point of the Phantom on the globe surface so that haptic feedback changes in response to both user motions and the spinning of the globe.

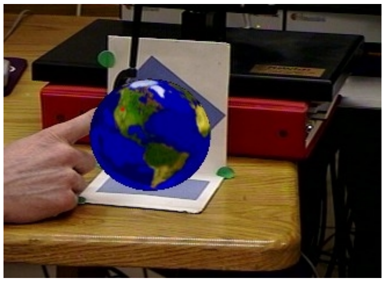


Figure 4 - Haptic interactions with a globe

3.4. Virtual-to-real haptic interactions

The next demonstration is with a cube virtual object. In Figure 6a, the user is moving the cube around by lifting it with a finger. The vertical part of the frame is defined in the haptic scene so that haptic interaction takes place between real and virtual objects. In Figure 6b, the user has just rested the virtual cube on top of the real vertical wall. It remains suspended in that position until moved by the user.

In an augmented view of the scene visual interactions between real and virtual objects must be considered [8]. The virtual camera in the computer graphics correctly handles hidden surface elimination within a virtual object and between virtual objects. The visual interaction between real and virtual objects must be considered. Using the graphics as the foreground element, or key mask, a luminance keyer displays the graphics image at every pixel above the luminance key value. The live video image is shown whenever a graphics pixel has a luminance value less than the key value. If the background regions of the virtual graphics image are rendered with a luminance at or below the key value then the live video is shown in all background regions and the virtual objects will occlude the video image in the areas where a virtual object is rendered.

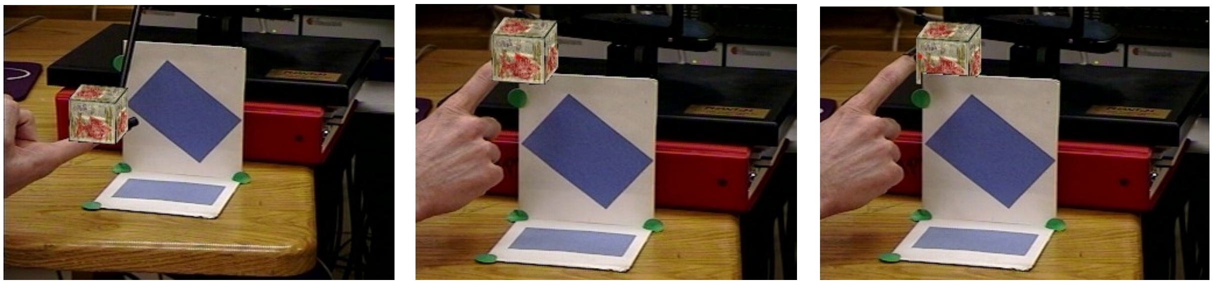
Hidden surface removal does not occur when a real object occludes a virtual one because there is no information about the geometric relationship between these objects [16]. If an affine model of a real object is included as another virtual object and rendered in the background color the keyer correctly resolves the occlusions. In our demonstrations, the vertical part of the frame is defined in the graphics scene as another virtual object. Figure 6c shows the cube disappearing behind the vertical wall after being pushed slightly by the user.

3.5. Foreground detection for improved visual occlusion

The images in Figure 4 and Figure 6 show one strong limitation of haptic interactions in this augmented reality system. This limitation is that the Phantom provides a very compelling sense of touching virtual objects but the visual image in the augmented display is not as compelling. The reason for this is that proper visual occlusions are not occurring between the virtual objects and the user's hand. In these examples, the virtual objects always occlude the user's hand even when the hand is interacting with a front surface of the object. A technique to ameliorate this problem is to create a red marker, defined as an object in the global affine coordinate system, representing the active point of the Phantom end effector. It facilitates the user's interaction with the virtual objects by being a substitute for the cues that visual occlusion normally provides.

Section 3.4 describes the method used for proper occlusion between real and virtual objects. To apply that idea to the haptic subsystem requires the Phantom device to be defined as a graphic object with actual joint angles monitored during operation controlling its configuration. However, this still does not provide for occlusion of virtual objects by the user's finger and hand.

A more general approach is to use a technique for foreground detection that has previously been applied to detecting humans moving through rooms. [17]. The technique initially analyzes the "background" workspace scene over a specified number of video frames and computes mean and covariance statistics on the YUV values of the video signal. The assumption is that the real scene contains objects that will always be present while the system is operating. When the system is running, any YUV pixel value found to be statistically different than the background color is marked as a foreground block. The foreground regions identify areas of the image where occlusion by a real object should take place. The system uses a 90% confidence level for the statistical test. Two methods have been implemented. The first gathers mean and covariance statistics on all three individual components in the YUV color space. A second method, which is more robust to shadow effects, computes statistics on only the UV chrominance components normalized by the luminance Y. All statistically different regions are deemed to represent real



(a) (b) (c)

Figure 6 - Interactions with a virtual cube

objects that have entered the workspace at the depth of the active point of the Phantom end effector.

To get the proper visual interactions we texture map the foreground plane with the detected foreground information. Figure 5a shows the foreground plane with detected areas of foreground activity marked in the regions with projections of virtual graphic objects. These are the only regions of interest. We want the virtual graphics image to key the live video into the final augmented image in any foreground region where the foreground mask plane is the virtual object closest to the camera in affine space. In the texture map applied to the foreground plane we render a detected foreground block as an opaque block below the luminance key value. In areas where we are not doing foreground detection or have detected no foreground activity we need the foreground plane to have no effect on the final virtual image or be transparent. The augmented image (Figure 5b) shows the user's finger properly occluding the virtual object that it hides.

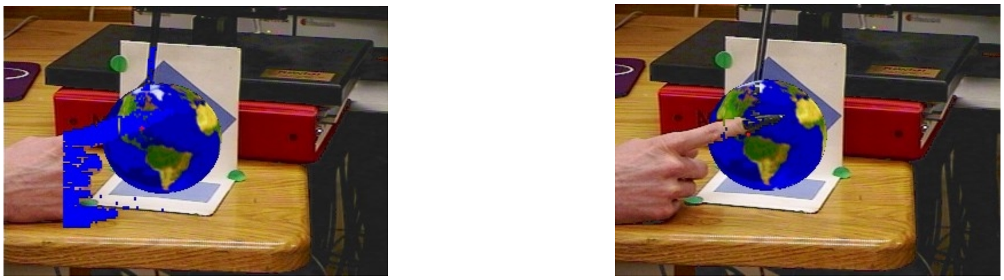
4. Discussion

The Phantom provides a very compelling sense of touching virtual objects especially when there is no conflict with the visual cues. Color background detection and masking of the video is a promising way to eliminate conflicts when proper occlusions between the real Phantom and the virtual objects do not occur.

Section 2 gave a brief overview of the uncalibrated, relative coordinate technique our system uses for registering the coordinate systems. This method is an attractive alternative to methods that require position measurement and/or camera calibration. It has limitations for the visual augmentation though [8]. There are implications on the haptic side also. As mentioned in Section 2 the global affine coordinate system is a non-Euclidean coordinate system. The technique we adopted computes the dynamics in the Euclidean reference frame of the Phantom and then transforms the haptic objects into the global affine coordinate system to determine the position for rendering their visual counterparts. Whether the dynamics can be computed directly in the global affine coordinate system is a question for future research.

5. Conclusions

Since its inception computer graphics has been interactive in nature. Augmented reality systems have been interactive only to the extent that the user could move about the workspace and be a passive viewer of the visually augmented scene. We have implemented an augmented reality system that incorporates a real-time



(a) (b)

Figure 5 - Foreground detection for visual occlusion

haptic interface device, thus adding touch as a second modality of synthetic sensory information augmenting the user's perception of a real scene. The user can realistically interact with a virtual object. These interactions include feeling the surface of the object, feeling the weight and dynamic forces of the object and moving the object within the workspace in a variety of manners. Future work aimed at decreasing system latency, better handling occlusions by real objects and scaling up the system will improve the performance of this augmented reality interface.

6. References

[1] R. T. Azuma, "A Survey of Augmented Reality," *Presence*, vol. 6, pp. 355-385, 1997.

[2] T. P. Caudell and D. Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes," presented at Proc. Hawaii Int. Conf. System Sciences, 1992.

[3] W. E. L. Grimson, G. J. Ettinger, S. J. White, P. L. Gleason, T. Lozano-Perez, W. M. W. III, and R. Kikinis, "Evaluating and Validating an Automated Registration System for Enhanced Reality Visualization in Surgery," presented at Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine '95, Nice, France, 1995.

[4] A. State, M. A. Livingston, W. F. Garrett, G. Hirota, M. C. Whitton, E. D. Pisano, and H. Fuchs, "Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies," presented at Proc. SIGGRAPH '96, 1996.

[5] P. Wellner, "Interacting with paper on the digitaldesk," *Comm. of the ACM*, vol. 36, pp. 86-95, 1993.

[6] M. Tuceryan, D. S. Greer, R. T. Whitaker, D. E. Breen, C. Crampton, E. Rose, and K. H. Ahlers, "Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 255-273, 1995.

[7] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Transactions of Robotics and Automation*, vol. RA-3, pp. 323-344, 1987.

[8] K. N. Kutulakos and J. R. Vallino, "Calibration-Free Augmented Reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, pp. 1-20, 1998.

[9] M. R. Mine, F. P. Brooks, and C. H. Sequin, "Moving objects in space: exploiting proprioception in virtual-environment interaction," presented at Proceedings of 24th International Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, 1997.

[10] F. P. Brooks, M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick, "Project GROPE - Haptic Displays for Scientific Visualization," *Computer Graphics*, vol. 24, pp. 177-185, 1990.

[11] R. Ziegler, C. Brandt, C. Kunstmann, W. Müller, and H. Werkhäuser, "Haptic Display of the VR Arthroscopy Training Simulator," presented at Proceedings SPIE Vol. 3012: Stereoscopic Displays and Virtual Reality Systems IV, San Jose, CA, USA, 1997.

[12] M. Dinsmore, N. Langrana, G. Burdea, and J. Ladeji, "Virtual Reality Training Simulation for Palpation of Subsurface Tumors," presented at Proceedings of the IEEE 1997 Virtual Reality Annual International Symposium, Albuquerque, NM, USA, 1997.

[13] A. State, G. Hirota, D. T. Chen, W. F. Garrett, and M. A. Livingston, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," presented at Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, 1996.

[14] Y. Yokokohji, R. L. Hollis, and T. Kanade, ""What You can See Is What You can Feel" - Development of a Visual/Haptic Interface to Virtual Environments," presented at Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium, Santa Clara, CA, USA, 1996.

[15] J. K. Salisbury and M. A. Srinivasan, "Phantom-based haptic interaction with virtual objects," *IEEE Computer Graphics and Applications*, vol. 17, pp. 6-10, 1997.

[16] M. M. Wloka and B. G. Anderson, "Resolving Occlusion in Augmented Reality," presented at Proceedings 1995 Symposium on Interactive 3D Graphics, Monterey, 1995.

[17] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.