A Microbased Shared Virtual World Prototype

Dr. Gerald Pitts, Mr. Mark Robinson and Mr.Steve Strange
Trinity University
Department of Coomputer Science
715 Stadium Drive
San Antonie, TX 78212

INTRODUCTION

Virtual reality (VR) allows sensory immersion and interaction with a computer-generated environment. The user adopts a physical interface with the computer, through Input/Output devices such as a head-mounted display, data glove, mouse, keyboard, or monitor, to experience an alternate universe. What this means is that the computer generates an environment which, in its ultimate extension becomes indistinguishable from the real world. "Imagine a wraparound television with three-dimensional programs, including three-dimensional sound, and solid objects that you can pick up and manipulate, even feel with your fingers and hands... Imagine that you are the creator as well as the consumer of your artificial experience, with the power to use a gesture or word to remold the world you see and hear and feel. That part is not fiction... three-dimensional computer graphics, input/output devices, computer models that constitute a VR system make it possible, today, to immerse yourself in an artificial world and to reach in and reshape it." (2). Our research's goal was to propose a feasibility experiment in the construction of a networked virtual reality system, making use of current personal computer (PC) technology. The prototype was built using Borland C compiler, running on a IBM 486 33 MHz and a 386 33 MHz. Each game currently is represented as an IPX client on a non-dedicated Novell server. We initially posited the two questions: 1. Is there a need for networked virtual reality?; and 2. In what ways can the technology be made available to the most people possible?

WHAT SHARED VIRTUAL REALITY HOLDS

With more and more people sharing information from remote sites, virtual reality is a perfect marriage between the person to person interface and the person to information interface. Using an earlier example, if two architects designing a building were separated geographically, they could meet in a virtual reality model of the building, walk through it together, and discuss the pros and cons. They could also "summon" other sources of data, utilizing applications and resources native to their own computers. People will be able to interact as if they were in the same room together without the necessity of physical proximity. "Communications has improved because the nuances of spoken language are transmitted instead of the impersonality of Morse code. Shared virtual reality can make the world a smaller place and at the same time provide an unlimited world to explore." (3). Architects can design 3D blueprints of buildings and use a VR simulation to test the ergonomic and structural attributes before construction begins. This will increase safety, accelerate construction, and help prevent costly mistakes in design. Biology students could study human anatomy, stripping away flesh, bone, and muscle to peer at the inner workings of the body, without touching a corpse. "The shift of focus from working with alphanumeric data to working in a three-dimensional representation of that data can alter dramatically both the manner in which we work with computers and our productivity and enjoyment when working with them."(1).

THE PERSONAL COMPUTER SOLUTION

Looking at the promise that virtual reality holds for so many wide-open areas, it seems only logical that its power be made available to as many people as possible. However, the average consumer cannot afford the $15,000 for the a head mounted display, data glove, and other virtual reality equipment typically used for a VR station. For a shared experience, add at least this amount to include another station and networks. The tightly controlled budgets typical in today's academic, business, and government institutions should look closely to insure that such costly hardware is needed. An economical answer to shared virtual reality could be found in the form of personal computers. Current PC technology possess the computational power necessary to generate and maintain realistic,

believable virtual worlds, as demonstrated in PC software such as Ultima Underworld or Alone in the Dark. PC's are affordable and accessible, easily holding the largest percentage of the computer market.

One of the most important factors in supporting PC based virtual environments is that we believe that immersion into a virtual world is a product of the mind. It is precisely because of this mental immersion that pulling away from a good book is so difficult. People do not depend on expensive hardware to immerse the mind. Are video games popular because of HMD's or Gloves? It is the game itself and often the ability to interact with another in the particular electronic world. Sharing a computerized experience greatly enhances immersion into the world because human beings are used to interacting with others on a daily basis. This is an integral part of what we term "reality". An individual does not have to rely on the expensive specialized hardware to have a meaningful virtual reality experience. It was this realization that sparked the construction of our prototype.

PROTOTYPE DETAIL

The Reality Forge involves separate computers running individual VR application shells, each configured to suit the computer's own unique hardware. A chief element of the system is its modularity, in the sense of hardware transparency. In other words, if one user had a head-mounted display device, the VR shell could be configured to utilize that piece of equipment, while another user might only have a black and white monitor. This would not change how the users interact in the virtual environment. The core of the VR shell is a virtual reality processor that receives transaction requests from some source (input from the local user or data packets from a remote user), interprets the request, and generates an appropriate response (update the visual image through coordinate transformations, assemble and send a data packet to a remote user, etc.). (See Figure 1). The virtual reality processor directly accesses the object coordinate database in order to translate the three dimensional world to a two dimensional screen and to initiate coordinate updates that will be reflected in all remote databases. The Data I/O Handler controls communication between parts of the shell as well as assembling and disassembling data Packets to be transmitted over the Novell network to other VR shells.

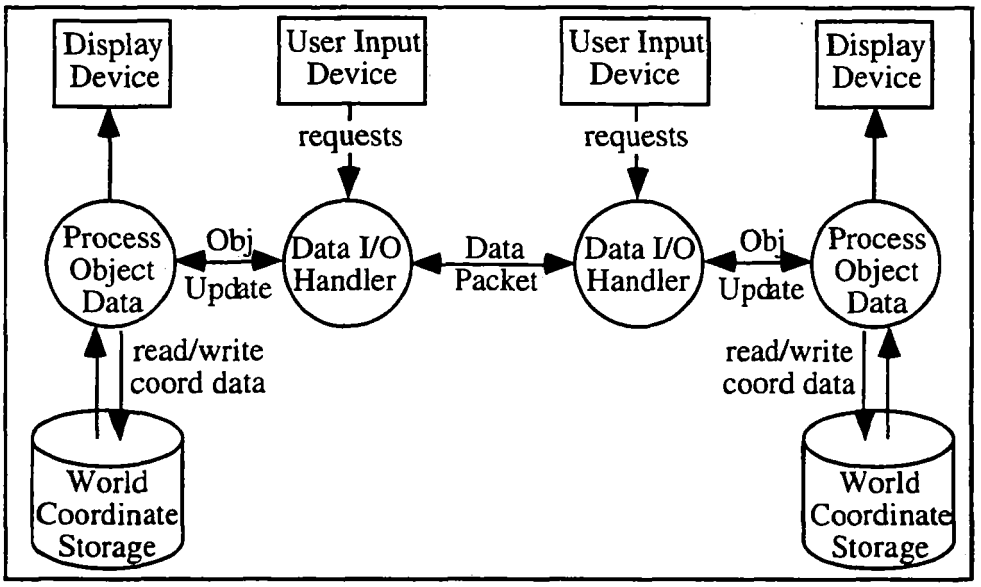

Figure 1 Simplified diagram of The Reality Forge

It was important for our VR system to be as "generic" as possible. There are two reasons for this: 1) Portability of the shell for ease of transference between platforms and uniformity of VR processing techniques. This allows users to upgrade the shell easily as computing power increases. These upgrades will include graphic enhancements (better shading techniques, shadows, etc.), full stereo sounds, and other equipment as the technology evolves (for example, tactile feedback equipment). And 2) Transparency of the VR processor promotes easier application development. (See Figure 2). For example, the implementation of a simple tank game on top of the VR shell took approximately 20 work hours.

Obstacles

It is important to realize that some of the methods employed in this project are not necessarily the most efficient. The program code is the physical manifestation of the actual research, its goal, as stated before: to explore the possibility of a PC-based networked virtual reality.

Graphical Representation of the World

Human beings can see through the reflection of light off objects into the eye. Light rays bouncing off of objects behind the person do not enter the eye and are not seen. Seeing is not so simple for a computer. To simulate human sight, only what a person standing in a real world would see would be rendered by the machine. Objects, especially those which are positioned directly behind the camera wreak havoc with the image if drawn to the screen. To help the computer to draw only what is in this viewing area, objects can be divided into three categories: wholly included, wholly excluded, and straddling. Wholly included objects can be entirely seen from the vantage of the person in question. Wholly excluded are those objects which are entirely out of sight. Straddling objects are those which can be partially seen, such as a long wall view from the middle, looking to one end. The unseen portion of the object must be mathematically clipped away where the object intersects the outside of the viewing area, leaving only wholly included objects. The computer then sorts these objects based upon how far away they are from the person. The most distant objects are drawn first by finding which faces point toward the viewer and filling them in with the specified face color. Closer and closer objects are drawn in this manner until all of the objects have been drawn. Drawing objects from farthest away to nearest gives a realistic rendering where objects in the foreground obscure objects in the background. This method for depth sorting is known as the Painter's algorithm. (4).

Collision Detection

Running into things may seem very natural to some people, but collision detection can provide some tricky problems. Pretend we have two cubes floating in a virtual space and we wish to see if they have collided. The accurate way to do this would be to check and see if any of the vertices have intersected any of the faces on the other cube. (cube 1,vertex 1 with face 1,2,3,4,5,6; vertex2 with...cube 2,vertex 1 with face 1,2,3,4,5,6...) This is seventy-two complicated three dimensional geometric checks to see if a collision has occurred each time either moves. A faster method would be to surround each object in the world with a boundary sphere. Each time an object moves the distance between it and other objects are computed. If the distance is less than the sum of the objects' radii, the objects have encroached upon each others' boundary spheres and therefore, have "collided".

Object Data Structure

A suitable data structure for virtual objects is essential to allow for future extension as well as provide efficient access to the object data. As the power of the VR system grows, so does the complexity of the objects in the virtual world. Accessing object data (relative coordinates, etc.) must be quick to provide fast calculations and re-draws. Their is the possibility of data structures needing frequent updating of object responses (a projectile becoming a land mine) as well as fast changes in shared views of the world. Time based movement and rotation ensured objects seen on different speed machines will move at the same rate per second. Objects which are not assigned to represent users may have a prescribed set of responses to certain actions. If a user touches a sphere in a virtual space, the sphere could move away from the user (a transformation change), turn a different color (a change in representation), or both. Object responses are encapsulated within the object's structure as pointers to mathematical time-based functions describing motion. (the projectile shell contains a pointer to a function which describes gravity)

Object Databases

One of the trickiest problems faced in the design of a shared virtual reality system is the physical location of the world database and insuring consistency among all users. Centralization of the object database necessitates each user pulling the world data from a central location, thereby increasing the likelihood of network bottlenecks and

data collisions. A distributed database system would greatly reduce the amount of network traffic but would also pose the difficulty of requiring a data synchronization scheme. The Reality Forge utilizes distributed object databases native to each user's shell with a simple synchronization technique to ensure uniformity of object motion and interaction.

Communication Between Computers

To facilitate the VR systems networkability between different platforms and allow users to take full advantage of their existing hardware, a widely implemented communication protocol must be employed. Netware IPX was initially chosen for its ease of use, but as of February, it was decided to eventually switch to TCP/IP, a widely-used cross-platform protocol. Because this protocol is a connectionless communication service, data exchange between machines was accomplished through the passing of packets. Loss of data packets from queue overflow and overhead of assembly/disassembly compounded the synchronization problem between shells. Therefore, the synchronization algorithm had to be extended to compensate for this to keep the multiple databases "looking" similar.

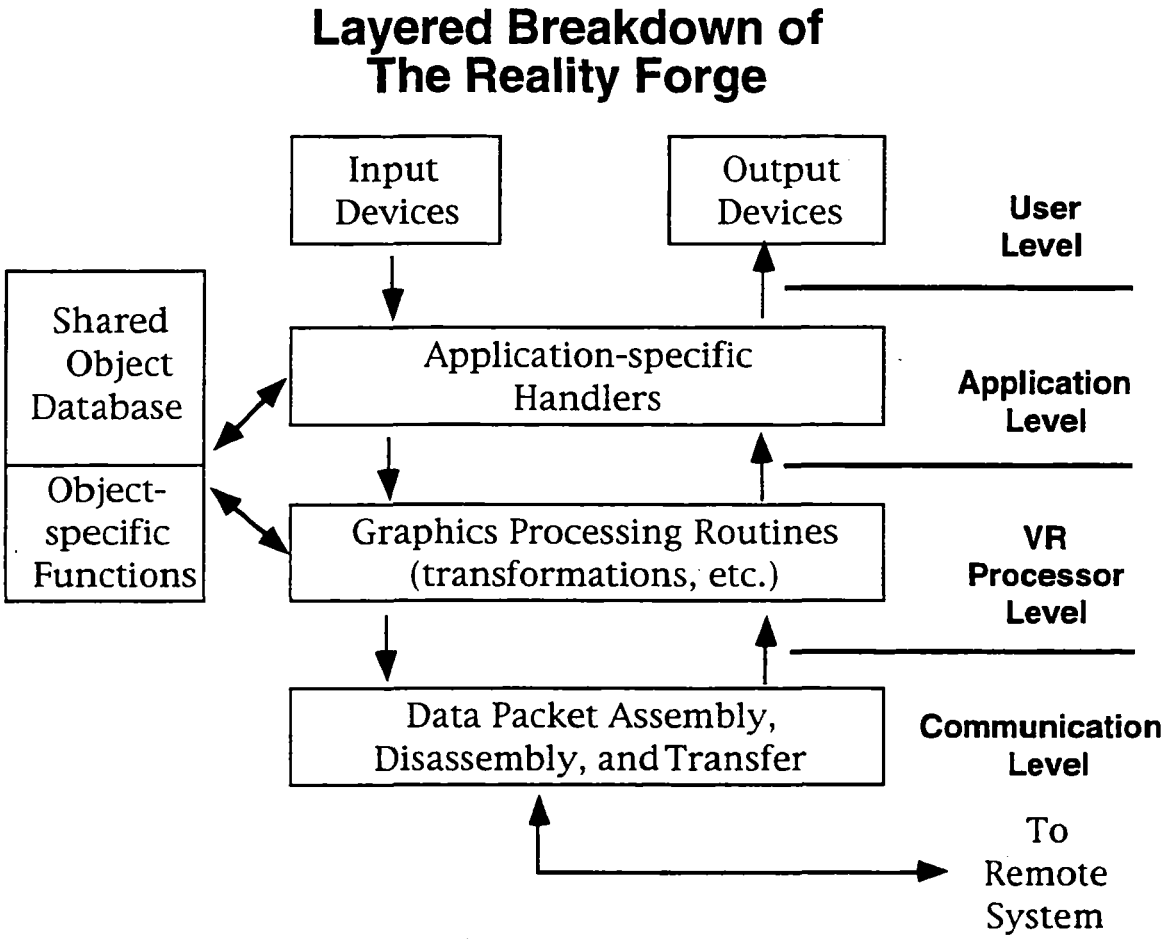

Figure 2 The Structure of The Reality Forge

Multi-user Interaction

When multiple, independent users interact in a shared virtual world, conflicting actions can be generated. For example, if one user wishes to "grab" an object at the same time another user tries to do the same, then a conflict arises and some solution must be determined. As the number of users increase, so does the likelihood of action conflicts. We have employed a simple conflict resolution routine where one of the shells is designated as an

arbitrator. A VR shell will react to a user request for movement or possession of an object only after attaining permission from the arbitrator. Permission is granted to the request received first.

CONCLUSION

If virtual reality is a human/computer interface, then networked virtual reality serves as a human/human interface, providing an incredible interpersonal communication tool. This shared virtual reality presents problems far beyond those associated with just the linking of machines. This project has been an exploration into the difficulties involved with such a task, as well as the "ground up" development of a PC-based VR system (dubbed The Reality Forge). Networked virtual reality will reach its full potential in the very near future and in order for it to achieve maximum accessibility, it must be geared toward the resources of the current average user. Armed with the results of this experiment, we believe that this is certainly an attainable goal. Keeping virtual reality financially exclusive will do nothing but dwarf interest and investment into it. In order to avoid this crippling mistake, networked virtual reality must be built around technology that is both affordable and accessible.

BIBLIOGRAPHY

1. Aukstakalnis, Steve. *Silicon Mirage.* Berkley, California: Teachtit Press, 1992.

2. Rheingold, Howard. *Virtual Reality.* New York, New York: Simon & Schuster, 1990.

3. Teixeira, Kevin. Kevin Pimentel. *Virtual Reality: Through the New Looking Glass.* New York: McGraw-Hill, 1993.

4. Van Dam, A. J.D. Foley. *Fundamentals of Interactive Computer Graphics.* Reading, Massachusetts: Addison-Wesley, 1984.