A NETWORKED VIRTUAL ENVIRONMENT COMMUNICATIONS MODEL USING PRIORITY UPDATING

Yang-Wai Chow, Ronald Pose and Matthew Regan
School of Computer Science and Software Engineering, Monash University
Clayton, 3800 Victoria, Australia

ABSTRACT

Networked virtual environments have been used in a variety of applications ranging from military training simulations to collaborative teleconferencing, and have recently become increasingly popular in the field of entertainment especially in online gaming. As the size, distribution and number of participants of these virtual world systems increases, so do the demands placed on the limited networking resources. This has presented networked virtual environment system developers with the problem of having to design efficient network communication models in order to reduce the amount of network traffic and bandwidth utilization, while at the same time not compromising on the real-time interactive nature of these large-scale multi-user virtual environments. This paper investigates the implementation of a priority updating network communications model, designed for the purpose of reducing the overall amount of information exchanged through the network.

KEYWORDS

Communications model, network resources, networked virtual environments, online gaming, spatial partitioning, update rates.

1. INTRODUCTION

Networked virtual environments (net-VEs) are real-time simulations of synthetic virtual worlds where multiple users, who might be scattered all over the globe, may interact together through a shared sense of presence, space and time created by the virtual environment system. In order to maintain interactivity between all participants of a net-VE, the shared state of the virtual world in the various distributed end-user systems must be kept reasonably consistent. The net-VE system must present the participants with the illusion that they are all seeing the same things in order for them to interact comfortably with each other. The task of maintaining consistency between participating systems requires the frequent sending of update messages through the network regarding events that are happening in the virtual world. This becomes harder as the size of the virtual world and the number of participants increases, due to limited resources in terms of network bandwidth, computational power and so on.

The increasing popularity of net-VE applications nowadays, especially in the rapidly growing field of online gaming [Kirkke 2003], has lead to the development of very large-scale virtual worlds that create shared environments for thousands of users to interact together. The huge volume of required information transfer between participating systems of such large-scale multi-user virtual environments can place tremendous demands on the bandwidth requirements of a system, which can lead to degradations in the system's performance due to limitations in networking resources. Therefore the challenge presented to net-VE developers is to design efficient network communication models around the limited resources for such large-scale systems.

In dealing with networking constraints, net-VE developers are faced with the problem of having to reduce the amount of data exchanged over the network, whilst not compromising on interactivity among the various users and their surrounding environments. The utilization of resources in a net-VE is directly related to the amount of information that has to be generated, transmitted and received by each host as well as how quickly that information must be delivered by the network, this has been defined as the networked virtual environment information principle [Singhal and Zyda 1999]. Many techniques have therefore been employed by net-VE systems to reduce the amount of information exchanged over the network, while at the same time maintaining reasonable consistency in the shared state of the virtual world.

This paper investigates the implementation of a priority updating network communications model, which is based on a previously designed priority rendering technique. Priority rendering is a rendering technique that was developed with the aim of reducing the overall rendering load [Regan and Pose 1994]. This technique was to be used in a specialized graphics display architecture known as the Address Recalculation Pipeline (ARP), which was designed for the purpose of reducing the end-to-end latency suffered by Head Mounted Display (HMD) virtual reality systems [Regan and Pose 1993]. The concept behind priority rendering is that in the scene that encapsulates a user's head, parts of the scene that are located further away from the user can be updated and rendered at lower rates as compared to parts that are closer to the user. The priority updating technique is applied in this paper for the purposes of network communications, with the intention of reducing the network traffic and bandwidth requirements of a net-VE system by decreasing the overall amount of necessary information exchanged over the network.

2. PREVIOUS WORK

A number of network communication models have been used by various net-VE systems to govern the exchange of shared virtual environment data. These communication strategies often attempt to minimize or reduce the required amount of data transfer. This section provides an overview to some of these techniques as employed by a variety of existing net-VE systems.

The SIMNET system, one of the earliest large-scale net-VEs, is a real-time distributed military virtual environment developed within the US department of defense [Calvin et al. 1993; Miller and Thorpe 1995]. The success of this system has in turn lead to the establishing of the Distributed Interactive Simulation (DIS) standard. The fundamental goal of the DIS protocol was to define an infrastructure for linking simulations of multiple types, in other words to achieve interoperability between different systems built for separate purposes [Hofer and Loper 1995]. The DIS network software architecture introduced the use of the Protocol Data Unit (PDU) to regulate the exchange of data within the system. These PDUs contain fixed fields of information used for communicating between participating systems. The most commonly used PDU is the entity state PDU used to carry data such as entity position, orientation, velocity, etc. [Singhal and Zyda 1999].

Dead reckoning algorithms were used in SIMNET to reduce packet traffic. Dead reckoning is a methodology used to predict and extrapolate the behaviors of entities based on their last received state update message, or in simpler terms to fill in the gaps between network update messages. Nine dead reckoning algorithms have been defined by the DIS standard, and the entity state PDUs themselves contain information fields for particular dead reckoning algorithms [Singhal and Zyda 1999]. Dead reckoning techniques have since been developed and used by others for a variety of net-VE applications, as a way of decreasing the amount of necessary message communications across the network whilst still maintaining reasonable consistency [Cai et al. 1999; Capin et al. 1999; Duncan and Gračanin 2003]. By predicting the behaviors of virtual world objects, dead reckoning smoothens the rendering of object movements and therefore has the effect of reducing the impact of network latency on user performance [Pantel and Wolf 2002].

In large-scale virtual environments, the information relevant to an individual user will generally only be local in nature. This has lead to several spatial partitioning and visibility culling techniques that have been adopted by some net-VE systems, in an effort to reduce the system's overall communication costs. The SPLINE system divides the entire virtual world into smaller chunks known as locales [Barrus et al. 1996]. The primary organizing principle behind this system lies in the use of these locales, as everything in the virtual world is either directly or indirectly contained within exactly one locale. A user will also be contained within a locale, and can move from one locale to another. The user will only be sent update messages from the relevant locales which concern that particular user, typically the neighboring locales.

Area-of-Interest (AOI) filtering is another virtual world spatial partitioning technique used in systems such as NPSNET [Macedonia et al. 1995], in order to decrease the amount of packets flowing through the network. In this approach the virtual world is divided into AOIs, where participating systems only establish communication channels and exchange information between other participating systems within the scope of their own AOI. Other net-VE systems like the RING system, decreases message passing by implementing visibility culling. This is a technique where irrelevant update information regarding the state of virtual world objects and other users that are either occluded or lie outside the scope of a user's field of view, are filtered out and not sent to that particular user's system [Funkhouser 1995].

Visibility culling has also been integrated with a scheduling strategy known as priority round-robin scheduling for very large-scale virtual environments [Faisstnauer et al. 2000]. The design of this scheduling scheme is aimed at managing the prioritizing of round-robin update messages sent to participating systems, in a manner that reduces the risk of starvation for update messages competing for networking resources. This approach has been used to reduce the number of messages transmitted over the network by using visibility information in the assigning of update message priorities.

The network communications model implemented in this paper, introduced in Chow et al. [2005] as the priority updating model, not only takes advantage of a virtual world's spatial characteristics but takes virtual world spatial partitioning a step further by allowing objects located in different sections of the virtual world to be updated over the network at different update rates. This approach differs from visibility culling efforts previously described above where occluded or distant objects are completely filtered out from the network update messages. Using this network communications model, distant objects are merely updated less frequently. A more detailed description of this approach is provided in the following section.

3. SYSTEM FRAMEWORK

The aim of priority updating is to reduce the overall amount of necessary update information sent through the network, while at the same time not affecting the general performance of users interacting with their surroundings in the shared virtual environment. This is achieved by spatially dividing the entire virtual world into separate equally-sized regions, object updates for these regions can then be sent over the network at different update rates depending on the how close these regions are from the user. Figure 1 illustrates the division of the virtual world into square or hexagon regions. This figure also depicts the how virtual world objects located in the neighboring regions can be updated at different update rates, with respect to the scope of a user's interaction range and viewing range. As a user crosses the boundary from one region into another, regions of relevance to the user should change as appropriate.

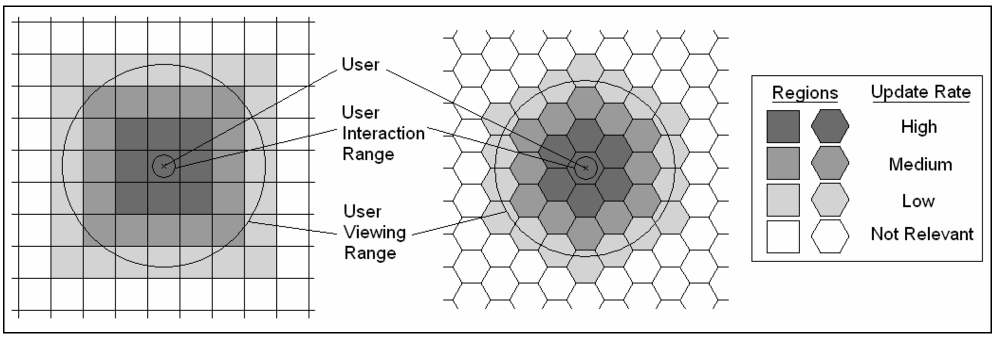

Figure 1. Top-down view of a section of the virtual world partitioned into square or hexagon regions and their relative network update rates with respect to the user.

In a typical net-VE application, the user is generally only allowed to interact with objects or other users within his/her vicinity. In this manner, it is acceptable if distant objects are slightly inaccurately represented on the display, as a user cannot interact with these objects until the user moves closer. Therefore this approach of updating different objects at different update rates is preferred over the alternative method used by some other net-VE systems of not displaying distant objects at all, thereby significantly cutting down the user's viewing range. In priority updating, even if the position or behavior of distant objects might be inaccurate, the simple fact that they are displayed to the user creates a more realistic and satisfying illusion of an extensive viewing range. Moreover as a user moves closer, the message updates for these previously distant objects will be sent at higher rates giving time for the user's system to correct any discrepancies in the objects' display, thus having little or no adverse effects on the user's interaction.

It is important to note that a user's interaction range is different from the user's viewing range, and that the region size should be larger than the user's interaction range. The size of the regions will have to be application specific, because the user's interaction range and viewing range will differ from application to application. The partitioning of the virtual world using other region shapes is also possible, as long as the chosen shape can comfortably cover the entire world leaving no gaps or overlaps. However, care should be taken in the choice of appropriate region shape and size for a particular application, as the size and shape of the regions will have an effect on the computational load required for the processing of region information.

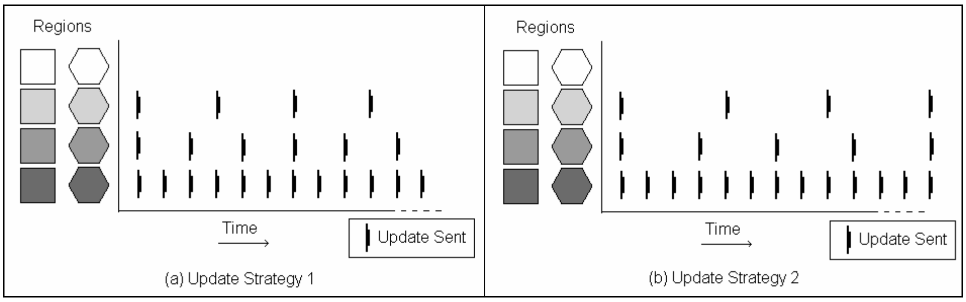

Figure 2. Examples of two different region message update strategies.

Figure 2 shows examples of two different message update strategies that may be implemented for the different regions (refer to figure 1 for region locations with respect to the user). By using these update strategies, a participant's system will receive update messages for objects in regions closer to the user more frequently, as compared to update messages for objects at a distance. This also means that update messages for all objects within a user's viewing range do not have to be sent over the network upon every single update cycle. Other region update strategies may also be adopted depending on the appropriate frequency of object state updates acceptable or tolerable for a particular application.

The advantage of using this approach is that a user's scope of view is not diminished by culling out the representation of distant objects. These distant objects are merely updated less frequently. The illusion of an extensive viewing range is important in maintaining a user's sense of immersion in the virtual world. A user's sense of immersion is made up of two primary components; presence and co-presence. Presence is defined as the feeling of realistically being and interacting in the alternate reality portrayed by the virtual world, whereas co-presence is the awareness of sharing the environment with other users [Oliveira and Henderson 2003]. Therefore even if slightly inaccurate, the representation of distant objects contributes to the user's awareness of his/her surroundings, leading to a more immersive experience.

4. EXPERIMENTAL IMPLEMENTATION

Experimental simulations of a net-VE system implementing the priority updating framework described earlier were performed. There are two main network communication architectures used in net-VE systems, these are the client/server architecture and the peer-to-peer architecture. The aim of the experiment was to validate the fact that priority updating would decrease the overall amount of data transmitted through the network. Therefore with this goal in mind, using a client/server setup would allow for the monitoring of message update data sent from a centralized server system to the various client systems, rather than having to monitor the collective behavior of a group of individual peer systems in the peer-to-peer architecture. In light of this, the client/server architecture was chosen for the experiments. However, it should be noted that the priority updating framework may also be applied to a peer-to-peer system.

In the experiments, the server was responsible for coordinating the simulation of the large number of virtual world entities as well as the various users. The centralized server system had to transmit update messages to the individual client systems periodically regarding the status of objects in their surrounding environments. Update messages sent by the server over the network contained up-to-date object status information such as object id, object type, current position and current translation vector etc. The client systems merely collected input from the user concerning his/her actions and relayed this information to the server. The client systems also had to display the virtual world to the user. For simplicity's sake, square regions were used for the partitioning of the virtual world. Another advantage of spatially dividing the virtual world into regions is that it is possible to then distribute the processing of different virtual world regions across multiple servers. Different regions can be assigned to different servers in order to share and balance the computational load of managing large-scale virtual environments.

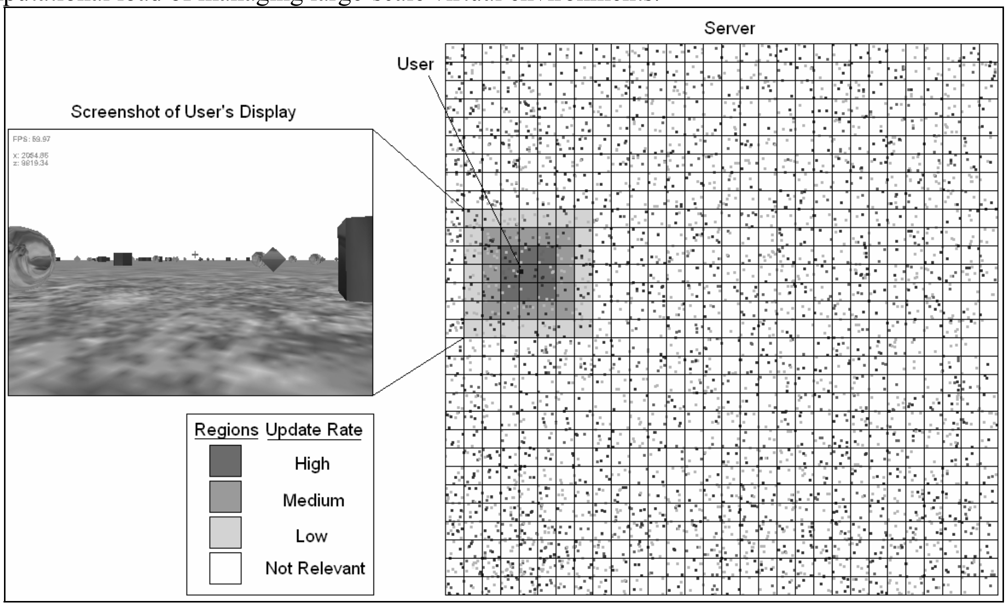

Figure 3. Virtual world regions and the different region update rates for a particular user.

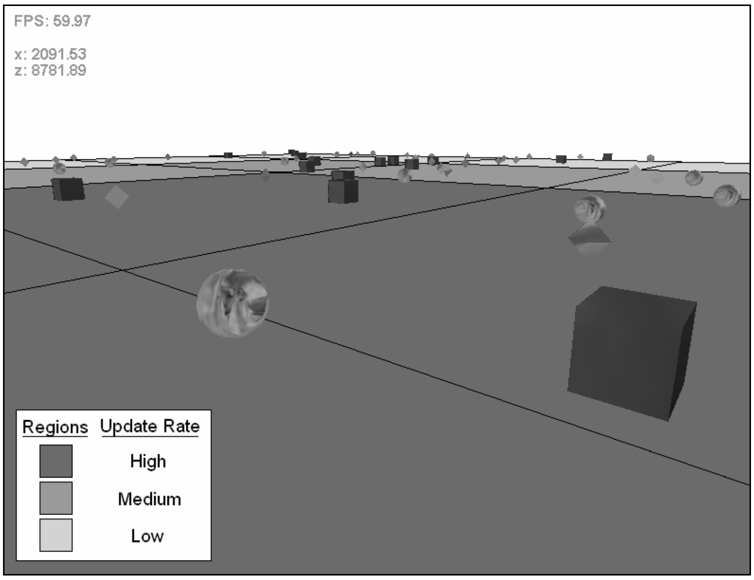

Figure 4. Region update rates from an elevated client's perspective.

Three different methods were used in the simulations for generating and transmitting update messages from the server to the clients. In the first method, update messages were sent over the network for all objects located within a certain viewing radius from the user. Priority updating was implemented for the other two methods, using the two different region message update intervals that were previously shown in figure 2. Using the three different message update methods, measurements were taken of the average number of objects that were updated.

Figure 3 provides an overview of the virtual world simulation. It shows a top-down view of the various entities as coordinated by the server and also the division of the entire virtual world into square regions. Attention is drawn towards one particular user in the simulation. The different update rates of the regions surrounding the user are highlighted, and the screenshot shows what that user was seeing at that point in time. Figure 4 is a depiction of the virtual world regions from an elevated user's perspective, where the different region update rates for the objects are illustrated. From this figure, it can be seen that objects in the regions closer to the user are updated at a high rate while the objects located regions further away are updated at lower rates.

The lower update rates for the distant objects are tolerable because the user can only see these objects but cannot immediately interact with them. In addition, from the user's point of view objects located at a distance away from the user are much smaller in size as compared to close objects. Therefore inaccuracies in the portrayal of these faraway objects will not be very noticeable. The main thing is that through the representation of these distant objects, the user is visually aware of the presence of other users in the shared environment.

5. RESULTS AND DISCUSSION

Figure 5 is a graph of the average number of objects that were updated when using the three different object updating methods. It can be seen from the graph that the use of the priority updating network communications model achieves its intended objective, in that its implementation in a net-VE system considerably reduces the average number of object updates. The amount of data that has to be sent over the network is proportional to the number of objects that are updated. In this respect, the significant reductions in the number of object updates therefore also decreases the quantity of required information exchanged through the network, consequently resulting in reductions to the overall amount of network traffic and bandwidth utilization of the net-VE system.

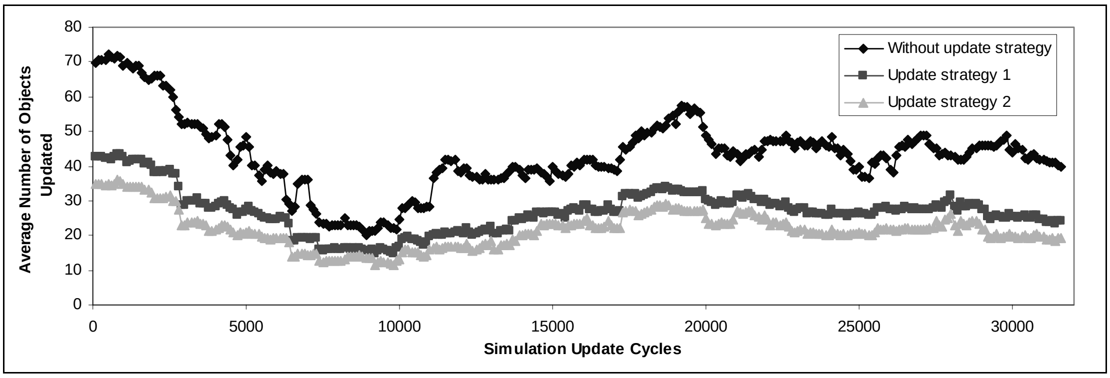

Figure 5. Average number of objects that were updated.

Scalability of a net-VE system is extremely important when designing large-scale systems that are required to host a large number of users. The reductions in the overall network traffic and bandwidth utilization obtained from the use of priority updating, have the potential of making a net-VE system more scalable. This is because more users can connect to the net-VE system without threatening to degrade the system's performance by overwhelming the limited resources available. In addition, the reduction in the amount of information transmitted through the network also decreases the likelihood of causing network congestions, which can further add to network latency by delaying messages from reaching their destinations.

Figure 6 shows a plot of the percentage savings in the average number of object updates, between the two priority updating cases as compared to the case without priority updating. The plot shows that adopting an update strategy where objects in the regions surrounding the user are updated less frequently, yields more savings in the amount of data transfer. However, there is a tradeoff between the overall amount of information transferred and the consistency in the virtual world state as seen by the various users. Therefore, although employing an update strategy with fewer object updates saves on the quantity of data exchanged, such update strategies can only be used for net-VE applications that do not required tight consistency in the representation of the virtual world among the individual user systems.

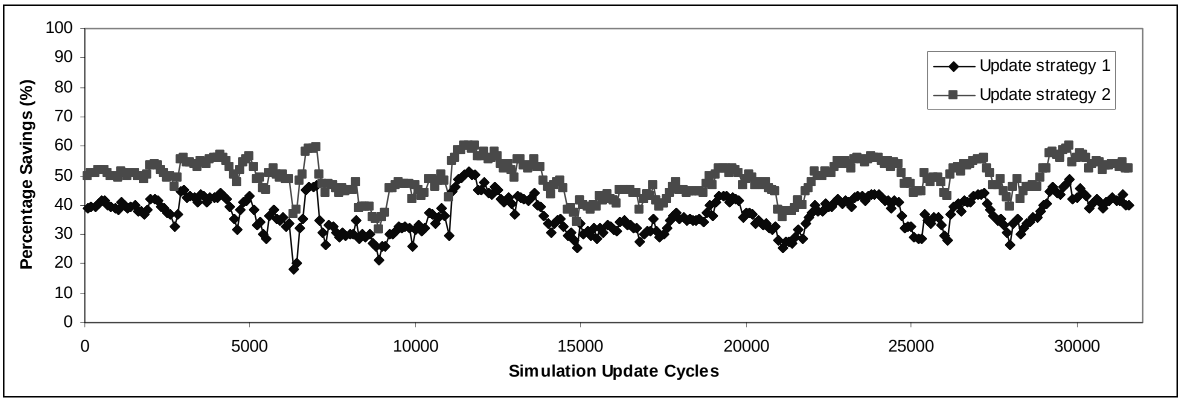

Figure 6. Percentage savings in the average number of objects updated using priority updating strategies.

The appropriate balance in this tradeoff has to be ascertained by a particular net-VE system's developers. A number of methods have been used to evaluate inconsistency in virtual environments, such as 'phase difference' [Lui et al. 1999] or 'drift distance' [Diot and Gautier 1999] between objects. A time-space inconsistency metric has been defined by [Zhou et al. 2004] in order to quantify the level of inconsistency in a distributed virtual environment system. The advantage of their approach is that the time-space consistency properties of the system in question can be estimated based on some characteristic parameters of the net-VE, without the actual execution of the real system. In this respect, this virtual world consistency estimation is especially useful for the system developers in the design phases of a particular net-VE system.

In view of the fact that in priority updating, an end-user system will only receive update messages regarding distant objects at certain intervals, the use of efficient dead reckoning algorithms in conjunction with this approach is essential in order to smoothen out the rendering of object movements. The prediction and extrapolation of object behaviors is even more important when factoring in network characteristics such as network jitter, data corruption and packet losses, which can widen the gap between update messages.

Dead reckoning will also prevent objects from 'teleporting' from one location to another, as the sporadic status updates for these objects are received. The human visual perception is less sensitive to details of objects located at a distance away from the user's point of view, as discussed in Chow et al. [2005]. Even if there are large discrepancies between the predicted behavior and actual behavior of distant objects, these errors can be corrected before the objects are within interaction range. This is because the user's system will begins to receive more frequent update messages concerning these objects as the user moves closer to them. In this manner, the user's interaction with the virtual world object and other participants will not be compromised.

6. CONCLUSION

This paper has shown that the implementation of the priority updating approach in the network communication architecture of large-scale multi-user networked virtual environment systems, can lead to significant savings in the overall communication costs of the system. This update strategy has the advantage of transmitting less update information over the network, whilst still maintaining reasonable consistency in the shared state of the virtual world and not compromising on the interactivity between users and their surrounding environments. The reduction in the overall communication costs also improves the scalability of a net-VE system, by allowing for more participants to be able to immerse themselves in the shared virtual environment without degrading the system's performance.

The limitation of this priority updating strategy is that it sacrifices accuracy in the representation of distant virtual world objects, for the gain of requiring less data exchange. This approach is therefore only suitable for virtual environment applications where tight consistency between the virtual world's shared state among the various users is not essential for the usability of the virtual environment. Very large-scale networked virtual environment applications such as in Massively Multiplayer Online Role Playing Games (MMORPG) can potentially greatly benefit from the use of this network communication model. These virtual environments have to cater for thousands of users, but on the other hand do not require perfect consistency for user interaction and satisfaction.

ACKNOWLEDGEMENT

Textures used in the display of the experimental virtual world simulations presented in this paper, were taken from Paul Bourke's texture library at: (http://astronomy.swin.edu.au/~pbourke/texture/).

REFERENCES

Barrus, J.W., Waters, R.C. and Anderson, D.B., 1996. Locales: Supporting Large Multiuser Virtual Environments. *IEEE Computer Graphics and Applications*, Vol. 16, No. 6, pp. 50-57.

Cai, W., Lee, F.B.S. and Chen, L., 1999. An Auto-adaptive Dead Reckoning Algorithm for Distributed Interactive Simulation. *Proceedings of the 13th Workshop on Parallel and Distributed Simulation*, Atlanta, Georgia, United States, pp. 82-89.

Calvin, J., Dickens, A., Gaines, B., Metzger, P., Miller, D. and Owen, D., 1993. The SIMNET Virtual World Architecture. *Proceedings of the Virtual Reality Annual International Symposium (IEEE VRAIS '93)*, Seattle, United States, pp. 450-455.

Capin, T.K., Esmerado, J. and Thalmann, D., 1999. A Dead-reckoning Technique for Streaming Virtual Human Animation. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 3, pp. 411-414.

Chow, Y.W., Pose, R. and Regan, M., 2005. A Message Update Model for a Large-scale Multi-user Networked Virtual Environment System. *Accepted to Multimedia Cyberscape Journal (MMCJ)*.

Chow, Y.W., Pose, R. and Regan, M., 2005. Design Issues in Human Visual Perception Experiments on Region Warping. *Submitted to IADIS International Conference on Applied Computing 2005*.

Diot, C. and Gautier, L., 1999. A Distributed Architecture for Multiplayer Interactive Applications on the Internet. *IEEE Network*, Vol. 13, No. 4, pp. 6-15.

Duncan, T.P., and Gračanin, D., 2003. Pre-reckoning Algorithm for Distributed Virtual Environment. *Proceedings of the 2003 Winter Simulation Conference*, Vol. 2, pp. 1086-1093.

Faisstnauer, C., Schmalstieg, D. and Purgathofer, W., 2000. Priority Scheduling for Networked Virtual Environments. *IEEE Computer Graphics and Applications*, Vol. 20, No. 6, pp. 66-75.

Funkhouser, T.A., 1995. RING: A Client-Server System for Multi-User Virtual Environments. *Proceedings of SIGGRAPH 95, in Computer Graphics, Annual Conference Series 95*, Monterey, California, United States, pp. 85-92.

Hofer, C.R. and Loper, M.L., 1995. DIS Today. *Proceedings of the IEEE,* Vol. 83, No. 8, pp. 1124-1137.

Kirkke, J., 2003. South Korea Beats the World in Broadband Gaming. *IEEE Multimedia*, Vol. 10, No. 2, pp. 12-14.

Lui, J.C.S., So, O.K.Y. and Tam, T.S., 1999. Deriving Communication Sub-Graph and Optimal Synchronizing Interval for a Distributed Virtual Environment System. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, Vol. 2, pp. 375-361.

Macedonia, M.R., Zyda, M.J., Pratt, D.R., Brutzman, D.P. and Barham, P.T., 1995. Exploiting Reality with Multicast Groups. *IEEE Computer Graphics and Applications*, Vol. 15, No. 3, pp. 38-45.

Miller, D.C. and Thorpe, J.A., 1995. SIMNET: The Advent of Simulator Networking. *Proceedings of the IEEE,* Vol. 83, No. 8, pp. 1114-1123.

Oliveira, M. and Henderson, T., 2003. What Online Gamers Really Think of the Internet? *Proceedings of the 2nd Workshop on Network and System Support for Games (NETGAMES '03),* Redwood City, California, United States, pp. 185-193.

Pantel, L. and Wolf, L.C., 2002. On the Suitability of Dead Reckoning Schemes for Games. *Proceedings of the 1st Workshop on Network and System Support for Games (NETGAMES '02)*, Braunschweig, Germany, pp. 79-84.

Regan, M. and Pose, R., 1993. An Interactive Graphics Display Architecture. *Proceedings of the Virtual Reality Annual International Symposium (IEEE VRAIS '93)*, Seattle, United States, pp. 293-299.

Regan, M. and Pose, R., 1994. Priority Rendering with a Virtual Reality Address Recalculation Pipeline. Proceedings of SIGGRAPH 94, in Computer Graphics, Annual Conference Series 94, Orlando, Florida, United States, pp. 155-162.

Singhal, S. and Zyda, M., 1999. Networked Virtual Environments: Design and Implementation. Addison-Wesley.

Zhou, S., Cai, W., Lee, B.S. and Turner, S.J., 2004. Time-Space Consistency in Large-Scale Distributed Virtual Environments. *ACM Transactions on Modeling and Computer Simulations*, Vol. 14, No. 1, pp. 31-47.