MOBILE ADAPTATIONS FOR A MULTI-USER FRAMEWORK SUPPORTING VIDEO-BASED AVATARS

Peter Quax Tom Jehaes Maarten Wijnants Wim Lamotte
Expertise Center for Digital Media
Hasselt University
Wetenschapspark 2
B-3590 Diepenbeek, Belgium
{peter.quax, tom.jehaes, maarten.wijnants, wim.lamotte}@luc.ac.be

ABSTRACT

The application of Networked Virtual Environment (NVE) technology on mobile devices is an emerging area in research. The marketing potential for these types of applications, when deployed on mobile networks is huge, both for entertainment and professional use. This statement can be verified when observing the popularity of these types of applications on the PC platform. Traditionally, mobile devices were regarded as too limited for NVE-technology deployment in terms of processing power, screen resolution and interaction capabilities. Recent developments have pushed the boundaries and overcome some of these problems. However, it is clear that applications should still adapt to the inherent limitations of the mobile platform. In this paper, we show how our existing NVE framework was adapted for use on mobile platforms. Three major factors are taken into consideration : universal access to NVEs, issues in rendering and optimizations for direct and natural forms of communication between participants. The resulting framework enables connection to an NVE at any time, while remaining compatible with fixed-network users.

KEY WORDS
Networked Virtual Environments, Networking, Rendering, Technology Adaptation.

1 Introduction

Networked Virtual Environment (NVE) technology has been the primary building block of popular applications on the Internet, such as (massively) multiplayer on-line games[6] and on-line communities[13][23]. Traditionally, these applications are run on a powerful PC with a high bandwidth always-on connection to a local area network or the Internet. An emerging trend can be observed towards the deployment of these 'traditional' applications on mobile platforms. Especially entertainment-related software is a prime candidate for such a conversion, witness the growing popularity of portable game consoles featuring wireless connectivity[22][14] and java-enabled mobile phones. Services deployed on mobile networks to allow gamers in various locations around the world to compete have shown to have the potential of becoming a source of revenue[15].

On mobile devices however, comparable hardware to the PC platform is not (yet) readily available. Mobile devices suffer, among others, from limited screen space, processing power and input/output facilities. Some of these problems and limitations may be resolved/diminished in future (e.g. processing power), others are inherent to the compact form factor of the devices (e.g. screen size). When porting NVE technology based applications to mobile platforms, the impact of these factors on the overall user experience and satisfaction should be investigated and the application should be adapted from the 'traditional' version.

In this paper, we will concentrate on three of the factors that have a major impact on the overall appeal of the application, namely universal access, graphical presentation of the application and direct communication between participants.

2 Related Work

Network access to networked virtual environments can be provided using a number of underlying technologies. Some of these include GPRS (for truly mobile users), 802.11 WiFi (near hotspots), WiMax or 3G networks [7]. Our proposed NVE framework is specifically targeted towards WiFi access, but can easily be adapted for use on higher and lower capacity networks, as will be described in this paper. On a higher level, IPv6 is an enabling technology [26]that will facilitate communication among mobile clients, with its extended address space, better multicast support and integrated QoS facilities. It is important to stress that it is not the goal of our research to just have an application running on mobile devices that is not able to connect and/or communicate with other (fixed network) applications and devices. We believe 'backwards-compatibility' to be a very important factor if these mobile applications are to gain popularity.

Communication in networked virtual environments has been evolving from elementary text-based communication (text line chatting) to direct means of communication using speech through VOIP technology. An area of research that has not yet been fully explored is the use of video in NVE applications. The impact of video is especially important in large-scale environments where large numbers of users and large environments are present. Some time ago, our research group has provided some insight in these problems and offered solutions for fixed network clients. We would like to refer the interested reader to [18], [17] and [19] for a more detailed explanation. In this paper, we will explain how this approach can be extended further towards the inclusion of mobile devices.

Streaming high-quality video to mobile devices is unfeasible in many application scenarios due to the lack of throughput on mobile networks. A common solution to this problem consists of transcoding the video streams inside the network to a lower bitrate and/or resolution before they reach the mobile user. Usually, this transcoding is performed by a proxy located close to the receiving client [5] [12] [21].

As stated before, the restrictions of the mobile platforms are the main reason why a lot of work on rendering virtual environments on PDA devices has been focussed on remote rendering. In [11] for instance, the authors make use of the Chromium[9] framework to provide a solution for rendering detailed models on less powerful devices. For remote rendering, the PDA is only used as a remote interaction and visualization device , which overcomes some limitations of these devices. Another benefit is that the source data does not need to be transmitted to the mobile device which increases security for applications that handle sensitive data. Another line of development has been focussing on actual 3D rendering on the PDA device, which is also what we set out to achieve in this work. These developments have been inspired by the introduction of several mobile 3D APIs [3][4], rendering engines [1] and the emergence of mobile graphics hardware[2].

In our renderer, we make use of an hybrid geometric/image-based rendering scheme which utilizes geometry simplification based on [8], and the relief-texture mapping technique which was introduced by Oliveira in et al. [16]. The relief textures capture the appearance of the model as seen from each side of the bounding box. During rendering, each visible relief texture is pre-warped into a texture which is subsequently mapped onto the corresponding bounding box quad, resulting in a correct view of the model.

The rest of this paper is organized as follows. Section 3 provides a short introduction to the traditional wired version of our NVE architecture. It also discusses some of the issues in enabling access to the NVE anywhere and anytime. In Section 4, we will provide an overview of the adaptations in rendering that we implemented in order to be able to provide the user with an appropriate view of the environment, depending on the device capabilities. Section 5 discusses the infrastructure needed to provide mobile users with video communication functionality. We will describe the transcoding proxy architecture used to accomplish this. Section 7 provides some conclusions and pointers for future work.

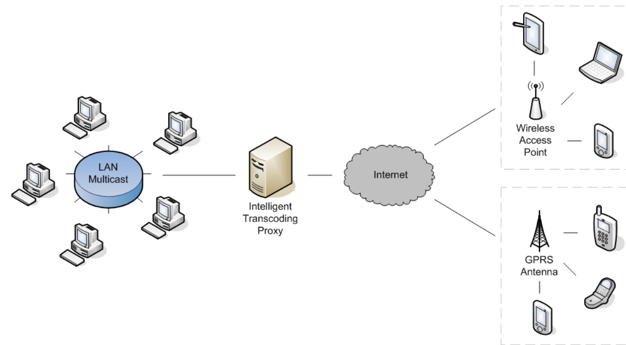


Figure 1. Mobile clients connect to our NVE application through an intelligent proxy.

3 Universal Access to NVEs

As mentioned, we have based the work in this paper on a large-scale NVE application that was developed inhouse[18]. When the clients log in to the NVE server, they first receive a list of available worlds from which they can select the one they want to explore. Upon selection of a world, the server transmits the static world information to the client. Each world is structured as a regular grid of rectangular regions. This spatial subdivision is used to speed up both visibility calculations in the renderer as well as network transmissions.

The framework relies heavily on multicasting to distribute NVE state information such as client position and orientation as well as video and audio data. Each world region maps to a unique multicast address. Clients send their own state information to the multicast address that corresponds to the region they are currently positioned in. To receive data, clients simply subscribe to the multicast addresses of the regions in their current area of interest. In this way they only receive the data they are interested in. Video and audio transmissions are handled similarly.

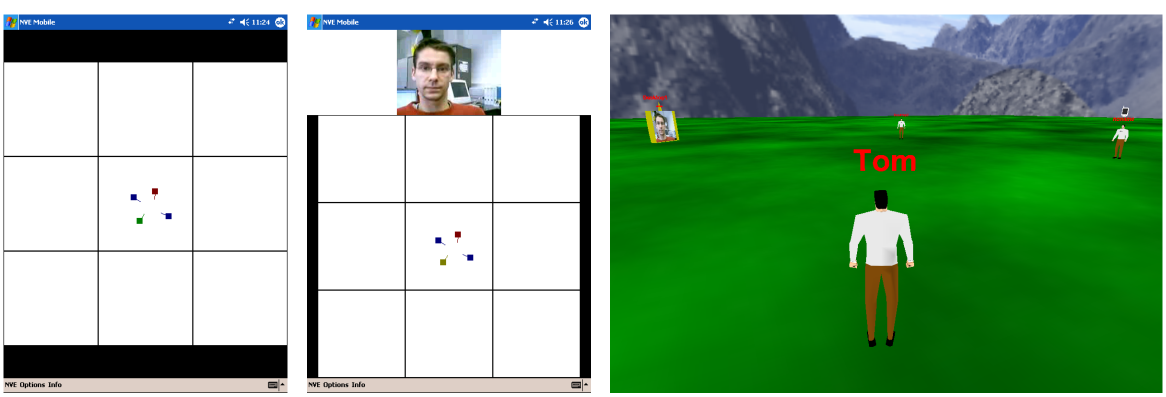
Applications targeting mobile devices should be available on a wide range of devices and networks in order to be commercially viable. Access to the environment should be transparent to the user, whatever underlying network technology is being used. The application should dynamically adapt to the changes in e.g. bandwidth availability , while providing a maximal quality of experience. Universal access to NVE's was recently described in another publication[20], and can be implemented in several ways. In case a WLAN connection is available that is directly connected to a LAN, it will usually suffice to just forward the data. However, when a direct connection between the mobile network and a LAN is not available, we propose the introduction of proxies to act as intermediate parties to do some translation of protocols. Using IPv6, which was designed to accommodate a much larger range of addresses, multicasting for mobile clients will also become easier to implement. The architecture of mobile clients connected to the NVE trough the proxy is depicted in figure 1.

(a) (b) (c)

Figure 2. Rendering the NVE on a mobile device. (a) screenshot from the mobile client. (b) viewing one client video stream. (c) the same view from the desktop application.

(a) (b) (c)

Figure 3. Hybrid rendering. (a) using no IBR. (b) using mixed IBR/geometry. (c) using only IBR. Note the difference in frame-rate.



(a) (b) (c)

Figure 4. 2D overview of the environment. (a) 2D only on mobile device. (b) combined with video display on mobile device. (c) corresponding 3D view on PC.

4 Rendering on mobile devices

Our mobile NVE extensions target users on the road using a PDA or SmartPhone or similar device. We want to give them the opportunity to stay in touch with the NVE using a 3D interface comparable to the one they are used to on their desktop systems. An example of this can be seen in figure 2. The two images on the left display screenshots of our client application running on a PDA while the image on the right shows the corresponding view from the desktop client. As can be seen, both the environment and the avatars are represented similarly on both platforms.

We have also provided mobile users with the possibility to view the video streams transmitted by other users of the NVE. To indicate which clients are sending out video, we have placed a multi-colored 3D arrow above the client avatar. The mobile user can indicate he wants to view a client video stream by tapping on the corresponding avatar with the stylus. The 3D arrow of the selected video avatar will then start rotating, indicating that the currently visible video originates from that client. The next section will go into further detail on the efficient distribution of these video streams. Note that in our current implementation only one specific video stream can be viewed at a time.

4.1 Hybrid representations

Clients connected to our NVE system request and receive environment data such as world layout and model data from the game server. To improve both model transmission and rendering, our NVE application features representation optimizations that have been specifically designed to enable the client application to present and render a sufficiently good initial view of the virtual world as quickly as possible. We have achieved this by using a hybrid representation approach consisting of both several levels of geometry simplifications and image-based representations of the models that are used in the environment. Further details of this hybrid approach can be found in [10].

In short, the main benefit of using this hybrid system is that the image-based representations consume less disk storage and application memory and are faster to send over the network than their geometrical counterparts. Furthermore, they can be used to increase the rendering framerate, especially if they are used for representing objects with low scene priority such as for instance distant objects. Our LoD selection algorithm determines which model representation should be used for each object in the environment, based on the current framerate.

We have incorporated this same strategy in our mobile renderer. Figure 3 shows a screenshot of our PDA client in which the user is navigating a world that incorporates 100 objects. Each object is represented by one of 3 different models. All 3 models also have an image-based representation, which is used to render the objects that are distant. The screenshots show that as more image-based models are used, the framerate increases from approximately 5 for full geometry rendering to 11.5 frames per second when using only image-based rendering. In practice, we only use mixed mode rendering, because when viewed up close the image-based models do not present sufficient detail, as can be seen in figure 3 if we look at the flamingo in the front.

4.2 2D interface

While a 3D interface is very nice to use, not all mobile devices will be capable of displaying the 3D environment at a sufficiently interactive framerate. For that reason, we have also implemented a simple 2D interface, as can be seen in figure 4. Again, we have provided the possibility to view the video stream send out by one of the other clients. We have used different colors to indicate whether or not a client provides a video stream and to indicate which client's video stream is currently being displayed. For instance, a video client is indicated in green, while the selected video client is colored brown. Other clients are colored blue and the client's own avatar is indicated in red. This 2D interface can be used on a wider range of devices because it is very lightweight and automatically scales to the resolution of the device output screen.

5 Video Communication

One of the main challenges we encountered when porting our multi-user NVE framework to mobile devices was to provide support for video streaming on these devices. Mobile devices have capabilities and constraints that are very different from those of standard desktop PCs. In the context of video streaming, the two main limiting factors of mobile devices are their limited display size and relatively slow processor. Both these factors imply that video streams sent to a mobile device ideally should have a rather modest spatial resolution. First of all, it is pointless to send high resolution video streams to mobile devices since these devices will normally not be able to display them at their original resolution. This is especially true for our framework, given that the mobile device in this case needs to divide its available display size over the virtual environment and incoming video streams, meaning that even less screen space can be dedicated to displaying received video. Secondly, lower resolution encoded videos generally require slightly less processor time to decode than their high resolution counterparts, which is a significant advantage due to the limited processing power of most mobile devices.

Closely related to the porting issues introduced by the constraints of mobile devices are the issues associated with the connection schemes of these devices. In contrast to desktop PCs, mobile devices are typically connected through an unreliable, low throughput wireless link. As a result, mobile users normally have less downstream bandwidth at their disposal than desktop users. This means application designers very deliberately need to distribute the available client bandwidth over the different types of

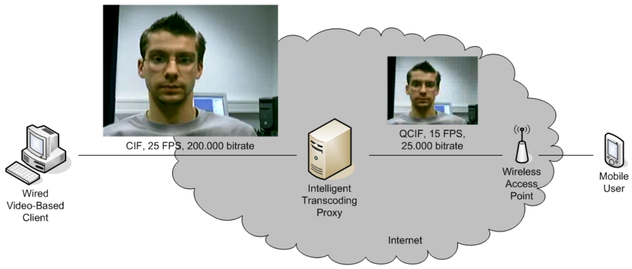


Figure 5. The proxy transcodes incoming video streams on-the-fly to the quality requested by the destined mobile client.

streams used by their application. Limited downstream bandwidth however also implies that it is unacceptable to allocate bandwidth to high-quality multimedia streams if these streams cannot be displayed efficiently on the receiving client device due to its constraints and restrictions. If we translate these observations to our NVE framework, video streams sent out by video-based avatars and destined for mobile users should have a relatively low bitrate and should have a framerate the receiving client device is capable of decoding in real-time.

Based on the above described requirements, we decided to connect mobile users to wired framework users through an intelligent proxy equipped with transcoding functionality. This resulted in the network setup shown in figure 1. The intelligent proxy acts as a unicast-to-multicast and multicast-to-unicast gateway for its connected clients. This means mobile clients on the one hand can unicast their positional and state information to the proxy, which will subsequently disseminate this information to all interested wired clients by sending it to the correct multicast group. On the other hand, the intelligent proxy subscribes to multicast groups on behalf of connected mobile clients, and subsequently unicasts the relevant information distributed in these communication channels to them. Furthermore, mobile users can also indicate at which quality they want to receive video streams. If the requested video quality does not match the quality of the video stream as sent out by the selected video-based client, the intelligent proxy transcodes the original video stream to the requested format before forwarding it to the mobile client. This is illustrated in figure 5.

A general discussion of the implementation of our intelligent proxy can be found in [24], while the proxy's video transcoding functionality is described in full detail in [25].

6 Test Results and Demonstration

In order to clarify some of the issues described in this paper, we have recorded a demonstration video and some supporting visual material, which is available from the following URL:

`http://research.edm.uhasselt.be/nve/IMSA2005/`

7 Conclusions and future work

We have shown how our existing PC-based Networked Virtual Environment framework was adapted for use on mobile devices. We have identified three major points of interest, namely universal access to the NVE, graphical representation of the environment and natural means of personal communication through the use of video.

Regarding universal access, we have introduced a proxy architecture that enables remote clients to connect to the virtual environment that is running on a multicast-enabled LAN. This enables users of e.g. public hotspots to remain in touch with everything that is going on in the virtual environment. We have adapted the rendering to be able to provide the mobile user with a view on the environment that is suited for his/her type of device. When the device is powerful enough for local 3D rendering, our hybrid representation method can be used to display large amounts of simultaneously connected users and/or static objects, while at the same time keeping the framerate at an acceptable level. In case a device is not equipped with 3D hardware and/or limited screen space, a 2D view is used to represent the environment. Video communication between clients is made possible through intelligent transcoding proxies that enable on-the-fly conversion between various video formats and qualities, to specifically suit the device and network the user is connected with.

The presented framework will, in the near future, be extended with the ability for session mobility, through which a session can be transferred from a fixed-network PC to a mobile device and back. We will also be looking at extending the features of video communication, especially the encoding of video on the mobile device itself. There are also tests underway to quantify the scalability of the application, when run on a mobile device.

Acknowledgments

Part of this research was funded by the IBBT (Interdisciplinary institute for BroadBand Technology), IWT of Flanders project MAVIC (number 040541) and the Flemish Government. The authors would like to thank all members of the NVE research group at EDM for their support in developing the technology as presented in this paper.

References

[1] Ideaworks3d segundo3d(tm) mobile 3d graphics technology. World Wide Web, `http://www.ideaworks3d.com/`.

[2] Nvidia handheld graphics solutions. World Wide Web, `http://www.nvidia.com/page/handheld.html`.

[3] Opengl es. World Wide Web, http://www.khronos.org/opengles/.

[4] Sun, java3d. World Wide Web, http://java.sun.com/products/java-media/3D/.

[5] Elan Amir, Steve McCanne, and Hui Zhang. An Application Level Video Gateway. In *Proc. 3rd ACM International Conference on Multimedia*, pages 255–265, San Francisco, California, November 1995.

[6] Blizzard. *World of Warcraft*. World Wide Web, http://www.worldofwarcraft.com.

[7] Eurescom. *A roadmap for mobile and wireless systems beyond 3G*. World Wide Web, http://www.eurescom.de/message/messageMar2002/beyond3g.asp.

[8] Hugues Hoppe. Progressive meshes. *Computer Graphics (ACM SIGGRAPH '96 Proceedings)*, 30(Annual Conference Series):99–108, 1996.

[9] Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D. Kirchner, and James T. Klosowski. Chromium: a stream-processing framework for interactive rendering on clusters. *ACM Trans. Graph.*, 21(3):693–702, 2002.

[10] Tom Jehaes, Peter Quax, Patrick Monsieurs, and Wim Lamotte. Hybrid representations to improve both streaming and rendering of dynamic networked virtual environments. In *Proceedings of the 2004 International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI2004)*, pages 26–32. ACM Press, 2004.

[11] Fabrizio Lamberti, Claudio Zunino, Andrea Sanna, Antonino Fiume, and Marco Maniezzo. An accelerated remote graphics architecture for pdas. In *Proceeding of the eighth international conference on 3D Web technology*, pages 55–ff. ACM Press, 2003.

[12] Zhijun Lei and Nicolas Georganas. Video Transcoding Gateway for Wireless Video Access. In *Proc. IEEE Canadian Conference on Electrical and Computing Engineering (CCECE'03)*, Montreal, May 2003.

[13] Linden Labs. *SecondLife*. World Wide Web, http://www.secondlife.com.

[14] Nintendo. *Nintendo DS game station*. World Wide Web, http://www.nintendo.com/systemsds.

[15] Nokia. *N-Gage Arena*. World Wide Web, http://arena.n-gage.com.

[16] Manuel M. Oliveira, Gary Bishop, and David McAllister. Relief texture mapping. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 359–368. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[17] Peter Quax, Chris Flerackers, Tom Jehaes, and Wim Lamotte. Scalable transmission of avatar video streams in virtual environments. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME 2004)*. IEEE, 2004.

[18] Peter Quax, Tom Jehaes, Pieter Jorissen, and Wim Lamotte. A multi-user framework supporting video-based avatars. In *Proceedings of the 2nd workshop on Network and system support for games*, pages 137–147. ACM Press, 2003.

[19] Peter Quax, Patrick Monsieurs, Tom Jehaes, and Wim Lamotte. Performance evaluation of client-side video stream quality selection using autonomous avatars. In *Proceedings of the 2004 International Conference on Advances in Computer Entertainment Technology (ACE 2004)*, pages 251–256. ACM Press, 2004.

[20] Peter Quax, Maarten Wijnants, Tom Jehaes, and Wim Lamotte. Bridging the gap between fixed and mobile access to an audio/video-enabled large-scale nve. In *Proceedings of the IASTED International Conference on Web Technologies, Applications and Services (WTAS)*, 2005.

[21] Bo Shen, Sung-Ju Lee, and Sujoy Basu. Caching Strategies in Transcoding-enabled Proxy Systems for Streaming Media Distribution Networks. *IEEE Transactions on Multimedia, Special Issue on Streaming Media*, 6(2):375–386, April 2004.

[22] Sony Computer Entertainment. *Playstation Portable*. World Wide Web, http://www.us.playstation.com/psp.aspx.

[23] There Inc. *There*. World Wide Web, http://www.there.com.

[24] Maarten Wijnants, Patrick Monsieurs, and Wim Lamotte. Improving the User Quality of Experience by Incorporating Intelligent Proxies in the Network. Submitted to the 1st International Conference on Multimedia Services Access Networks (MSAN'05), June 2005.

[25] Maarten Wijnants, Patrick Monsieurs, Peter Quax, and Wim Lamotte. Exploiting transcoding to increase the user quality of experience in networked applications. In *Proceedings of International Workshop on Advanced Architectures and Algorithms for Internet DElivery and Applications (AAA-IDEA'05)*, 2005.

[26] Wolfgang Fritsche and Florian Heissenhuber. *Mobile Ipv6. Mobility Support for the Next Generation Internet*. White paper on Ipv6 Forum, http://www.ipv6forum.com/navbar/papers/MobileIPv6_Whitepaper.pdf.