Visualizing Real-time GPS Data with Internet's VRML Worlds

Ismo Rakkolainen, Simo Pulkkinen, Arttu Heinonen
Digital Media Institute
Tampere University of Technology
P.O. BOX 553
33101 Tampere, Finland
INT + 358 3 365 3843
{ira,simo,ahei}@cs.tut.fi

1. ABSTRACT

GPS data tells the position of a GPS user anywhere on the Earth. The position is usually shown on a 2D map. For many applications a 3D representation would improve the usability of the data.

A system is presented to visualize real-time multi-user GPS position data over the Internet in a three-dimensional VRML model of a real world place. The system is the first of its kind. Several mobile units can connect to the VRML world and their movements can be tracked in real-time by multiple viewers. A client-server system passes GPS position data from mobile units on to the viewer application which works on standard web browsers supporting VRML.

1.1 Keywords

GPS, GIS, Java, VRML, Virtual Reality

2. INTRODUCTION

Human senses are adapted to three-dimensional, interactive environments. As processors, displays, storage capacity and networks are improving and getting cheaper, new kinds of user interfaces will become feasible.

3D virtual environments are human-friendly as they try to speak to our senses. Virtual environment will be used as a user interface in, e.g., Helsinki Arena 2000 project [6]. The VRML-modeled city will provide a location-based interface to digital services for a broad audience.

The Virtual Reality Modeling Language (VRML97, VRML) [15] is the present de-facto format for describing 3D environments on the web. VRML97 is also approved as an ISO standard. VRML is supported on standard web browsers like Netscape Navigator and Internet Explorer.

Many VRML worlds model existing places like cities (e.g. [6], [14]) whereas others are purposefully non-realistic, like most "chat worlds". Even if a VRML world represents a real place, there is not usually a connection with the real-time reality. This kind of real-time connection is sometimes desirable and can be done by importing some kind of sensor data from the real world to the virtual VRML world.

One type of sensor data is position data, which can be received from many types of equipment. The scale and accuracy of position tracking varies depending on the methods and devices used. Some methods of different scale are, e.g., magnetic tracking and Global Positioning System (GPS).

GPS is widely used in various applications like car and marine navigation. Usually the position information is presented on 2D maps, which is sufficient for many purposes.

Combining GPS position data to VRML worlds makes virtual worlds more lifelike. A 3D representation of a real world place can make it easier to perceive, e.g., proportions, distances and landforms.

Possible applications of GPS visualization are numerous. Groups of people can communicate and view each other through the 3D-model. Multitude of people can view GPS-equipped sportsmen. Augmented reality enables new types of location-based services. Emergency call sites can be visualized for the rescue staff.

We implemented a generic system, which enables to view position data from real world in a three-dimensional model over the Internet. The system was used to visualize moving entities, which send GPS data of their current position through cellular phones. To test this system we tracked the motion of people with GPS receivers and other equipment walking on our campus area.

3. RELATED WORK

Silicon Graphics Inc. implemented a real-time viewing of America's Cup races in 1995 [16]. The GPS data from the yachts was visualized in 3D graphics. The visualization was done only locally on five press centers with high-end graphics workstations.

Non-real-time GPS survey data has been combined to a VRML model to show a relay race track in California [1]. It was used to visualize the track and plan the race in advance.

Wearable computing is a new emerging field of research. The future PC may be a small, wearable device with fast internet access, 3D-graphics and position tracking capabilities. Many trials have been made to try their possibilities [2], [4], [13].

Earlier we have made a system to monitor a car track race with VRML over the Internet [7], [8]. The accuracy and efficiency was analyzed and the dynamic data stream was very light.

4. SYSTEM OVERVIEW

The system presented is a client-server system. It can be separated into the following main components: mobile unit(s), server and viewer(s) (*cf.* Fig. 1).

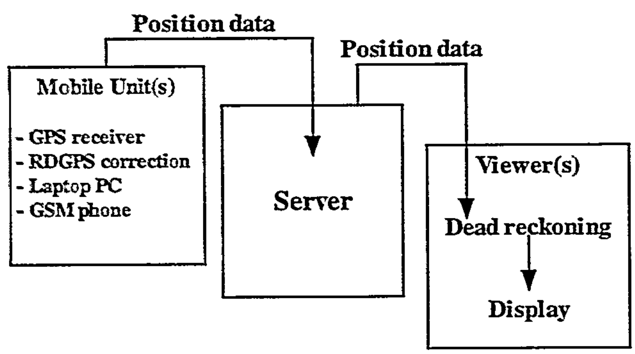

Figure 1: System overview.

We used a VRML model of the university campus area (*cf.* Fig. 2) and conducted the field trials in the campus. The model is accurately modeled (better than one meter) and is based on aerial stereo photographs. The model contains only some 900 polygons but that is enough for our purposes. The visual appearance is greatly improved by texture mapping the model with aerial and other photographs.

Figure 2: The VRML model of Tampere University of Technology campus.

Most of the scene is static and only the light GPS position data needs to be transmitted to the VRML model after the initial downloading.

4.1 Mobile Unit

The mobile unit consists of a GPS receiver (Garmin GPS 75), radio differential correction unit and batteries, a laptop PC and a cellular GSM phone. Figure 3 shows the present prototype of the mobile unit hardware.

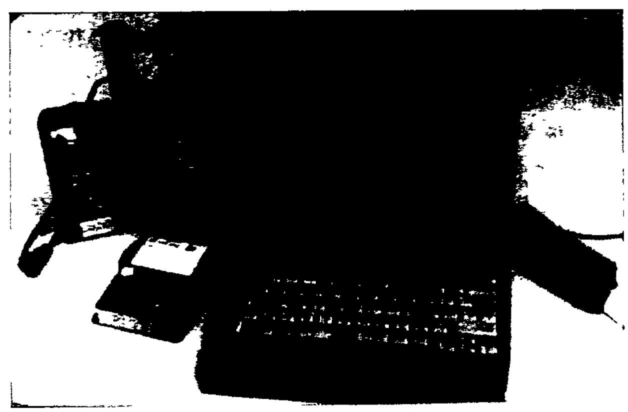

Figure 3: Mobile unit hardware.

A basic GPS receiver has an accuracy of ca. 100 meters, which is too inaccurate for use on the campus area. We used a radio differential correction (RDGPS) with 10 meters accuracy provided by the Finnish Broadcasting Co. A better accuracy (e.g., 2 meters) could easily have gotten at a higher price.

The computer used on the mobile unit was a low-end 486 laptop PC with Windows 95. The communication to the server computer was accomplished with Nokia PCMCIA GSM cellular data card and a cellular GSM phone.

There are also other possibilities to communication like radio modems or satellite phones. We used GSM as it is a reliable, european-wide system. It is also wide-spread (50% of the population in Finland use it), cheap (at lowest 10 cents / min.) and technically easy to deal with.

The GPS receiver sends data in the ASCII format every other second into the COM port of the connected laptop PC. Mobile application is a standalone Java program, which reads the GPS data from the COM port by using beta version of the Java communications API from SUN.

The application filters the relevant coordinates from the GPS data and makes needed calculations to convert coordinates from WGS-84 to Finnish national coordinates, then it sends them to the server computer.

4.2 Server

To show real-time GPS position we need to send the data constantly. The basic VRML97 standard can't use streamed data, so a system is needed to overcome this.

Some trials have been made on streaming VRML, e.g. [17]. There are several activities going on to develop the streaming protocols for VRML and other media types. VRML Consortium's Streaming VRML [18] and DIS-Java-VRML [3] working groups are working on it, and VRML/MPEG-4 liaison [19] working group is also relevant for the issue. Cosmo Software Inc. has also demonstrated streamed animations in VRML with RealNetworks' G2 software [12], so streaming data may become a standard part of VRML browsers in the future. At the moment there is not any standard VRML streaming method available.

In our system we used the *Invent* server [9], [11] which enables sending continuous position data in a user-friendly manner. Invent receives the data from mobile units and sends it with corresponding time and the identity of the unit through the network to viewers.

Invent enables platform independent shared experiences without downloads. It is a client-server software for multi-user VRML worlds (e.g., [10]) originally developed in Tampere University of Technology.

Invent is designed for shared virtual experiences, so it enables mobile users to see each other. This can be used, e.g., for inter-user group communication over a geographic area. Because of its generic implementation, it can easily be expanded for other interaction also.

In order to accommodate users with different hardware, it is possible to provide alternative geometries of varying complexity.

The server program running on a server computer is written in C++ for efficiency and is available for Windows NT, Sun Solaris and SGI IRIX.

4.3 Viewer Client

Viewer client is embedded in an HTML page. It consists of an embedded VRML model and a Java applet. Client-side parts of the Invent software are written in Java for platform independence and ease of use. To start the viewer client, the user just needs to go to the HTML page. No download or installations are needed and the user even doesn't have to know he is using the Invent software. This is not the case with most multi-user software.

On initial loading of the HTML page, only the static part of the VRML model and the viewer client are loaded. Then viewer client connects to the Invent server and waits for the connected mobile units. When a mobile unit is connected, viewer client adds dynamically a 3D representation of the moving entity (an avatar) to the static VRML world loaded earlier. This is accomplished by using the External Authoring Interface (EAI) for communication between the VRML browser and the viewer client Java applet.

Also different viewpoints are created for each avatar. This makes it possible to follow motion of the avatar(s) from a distance and angle defined by the user. It can also give certain kind of feeling of telepresence for the user when looking the environment from the avatar's perspective.

As mobile unit(s) moves in the real world, the corresponding GPS coordinates are sent through Invent server to the viewer client(s) in real-time. The real position of the mobile unit is sent as often as possible to the client. The GPS equipment we used gives a new position every other second. Relative to the accuracy and speed of the mobile unit this is usually enough.

If the mobile unit just sends GPS data, others can monitor its movements. If the mobile user also receives data from the server, he can also see what is happening around him. This may be useful, e.g., for some 3D group communication when the users are scattered over a large area.

When real position data is not available (between two consecutive GPS updates), a dead reckoning algorithm is used to estimate mobile unit(s) position and orientation.

Dead reckoning calculates new coordinates according to the last known position and speed. The calculated values are interpolations between the real ones (from GPS device) thus resulting as a smoother movement of avatar(s). Finally the calculated or real coordinates are passed from viewer applet to the VRML browser which renders the scene.

5. RESULTS

5.1 Visualization

The visualization of the real-time GPS data with the VRML world succeeded well. With the low polygon count, the model runs rather smoothly even on a 166 MHz MMX Pentium with 32MB RAM. With a new PC and 3D-graphics card the rendering performance is excellent. A demo (without real-time persons walking and the server software) can be found on http://www.dmi.tut.fi/netve/GPSdemo.

The present version of VRML can't handle very big worlds smoothly. GeoVRML [5] working group addresses these problems.

Avatars are also a problem. Every mobile user creates his own avatar to the world, which require hundreds of polygons each. This in turn requires 3D rendering power. With dozens of avatars the rendering gets slow.

If only few mobile objects need to be monitored by a big audience which don't have avatars, the rendering problem is avoided.

5.2 Data Transfer

The system is able to manage simultaneously multiple mobile units and viewer clients well. We couldn't test the system with many mobile units, though.

The data transfer rate to and from the server can be gotten from the equation:

$$M\,(136 + 8V)\ \text{bits/s},$$

where M is the number of mobile units and V is the number of viewer connections.

Theoretically, for a 33,600 bps modem, 10 mobile units can be viewed by over 350 users. For a 9600 bps GSM modem, about 25 mobile users can see each others' movements in real-time.

5.3 GPS Problems

The input frequency from GPS device was adequate to give real-time motion in a VRML world. Dead reckoning algorithm smoothed movements eliminating awkward jumps in the actual GPS data.

The inaccuracy of the basic GPS device (ca. 100 meters) was one of the main concerns. By using RDGPS correction we were able to partly overcome this problem. This gave us 10 meters accuracy, which was adequate for our purposes. At a higher price even a better nationwide accuracy (2 meters) could have been gotten. By constructing proprietary systems (local differential correction), even better accuracy (a few centimeters) can be gotten over local areas.

The accuracy of satellite navigation will improve as new hybrid GNSS satellite navigation systems will become widely available. Also the distortion of the GPS signal by the US military may be withdrawn in the future.

5.4 Hardware

Mobile unit hardware that we used in this test version was quite inconvenient to use because of it's size, weight and lack of integration. Using modern components the size and weight of the mobile unit can significantly be reduced. All the different devices could be integrated into pocket size. Some of this integration has already been done in commercially available products, e.g., Nokia Communicator 9110 (weighting only 249 g) which has a palmtop computer and a GSM phone in the same unit. Also most cellular phones will be equipped with GPS by the 2001.

It is difficult with present PC computers to render smoothly very detailed or huge areas. The situation will improve constantly as more powerful hardware (next-generation 3D graphics accelerators, CPUs etc.) comes available at low consumer prices.

6. FUTURE WORK

The system could be used, e.g., to broadcast sport events over the Internet and telepresence applications like guided interactive tours over the Internet.

Development in VRML streaming technologies will eventually allow more complex and larger worlds. This could help especially with generating huge geographic models of real world places (immersive 3D maps).

The system can also be used for 3D CSCW (Computer Supported Cooperative Work) applications. Mobile user will be able to view his surroundings in 3D and locate his mates through the virtual window. Networked cooperative 3D applications will become possible. Augmented reality equipment and other new user interfaces can also be used for some purposes.

7. CONCLUSIONS

The system described enables to transmit and view real occurrences very lightly and in real-time. It is the first multi-user system to visualize real-time GPS position data with VRML over the Internet. It also enables location-based 3D group communication.

Also we presented a way to send the GPS data stream to clients. Our system enables networked applications that previously were possible only locally with high-end workstations. The results are acceptable even with a low-end Pentium PC.

The future developments in computer hardware will make powerful 3D-graphics, light mobile networked computer systems and new user interfaces usual. They can benefit from the user's position data and visualize location-based information.

The applications of visualizing the real world occurrences with a VRML model are numerous. The need for GIS, spatial data and position data will grow and VRML is a tool for visualizing them over internet. New types of applications in internet using spatial data will emerge in the near future.

8. ACKNOWLEDGMENTS

This research was funded by Tekes - Technology Development Center of Finland. We also wish to thank prof. Pauli Kuosmanen for his helpful comments.

9. REFERENCES

[1] Abernathy, M., Shaw, S., Integrating Geographic Information in VRML Models. *Proceedings of the VRML'98 Symposium*, February 16-19, 1998, Monterey, CA, USA.

[2] Billinghurst, M., Weghorst, S., Furness, T., Wearable Computing for Three Dimensional CDCW. *Proceedings of the International Symposium on Wearable Computing.* October 13-14, 1997, Cambridge, MA, USA.

[3] The DIS-Java-VRML Working Group, *http://www.stl.nps.navy.mil/dis-java-vrml/.*

[4] Feiner, S., MacIntyre, B., Höllerer, T., A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. *Proceedings of the International Symposium on Wearable Computing.* October 13-14, 1997, Cambridge, MA, USA.

[5] The GeoVRML Working Group, *http://www.ai.sri.com/geovrml/.*

[6] The Helsinki Arena 2000 project, Helsinki Telephone Co. *http://www.helsinkiarena2000.fi/.*

[7] Hoppenot ,Y., Broadcasting a Modeled Reality with VRML, *M.Sc. Thesis*, Tampere University of Technology, September 1997.

[8] Hoppenot, Y., Rakkolainen, I., Real Virtuality - Bringing Position Data into VRML Worlds. *Proceedings of the Second Tampere International Conference on Machine Automation*, September 15-18, 1998, Tampere, Finland.

[9] Husberg, F., Blomqvist, M., Visala, K., Niskanen, M., Rakkolainen, I., Shared Platform Independent VRML worlds. *Proceedings of the ISMCR'97 Conference*, June 3-4, 1997, Tampere, Finland.

[10] The Interactive Virtual World of MTV3 Internet, *http://www.mtv3.fi/3d/english.html.*

[11] The Invent software, *http://www.mediaclick.fi/products/invent/documents/invent.html.*

[12] The RealNetworks' G2 software, *http://www.realnetworks.com/g2/overview.html.*

[13] Smailagic, A., Martin, R., Metronaut: A Wearable Computer with Sensing and Global Communication Capabilities. *Proceedings of the International Symposium on Wearable Computing.* October 13-14, 1997, Cambridge, MA, USA.

[14] The Virtual Los Angeles project. *http://www.aud.ucla.edu/~bill/UST.htm.*

[15] The Virtual Reality Modeling Language Specification, http://www.vrml.org/Specifications/VRML97/.

[16] Virtual Reality, The Ultimate Commentator's Tool, Iris online, Silicon Graphics Inc. http://www.uni-karlsruhe.de/~SGI/IOL/IRIS_Online_April_95.html#2.

[17] The VRML Dream Co. *http://www.vrmldream.com/.*

[18] The VRML Streaming Working Group, *http://www.vrml.org/WorkingGroups/vrml-streams/.*

[19] The VRML/MPEG-4 Liaison Working Group, *http://www.vrml.org/WorkingGroups/vrml-mpeg4/.*