EFFICIENT REPRESENTATION AND INTERACTIVE STREAMING OF HIGH-RESOLUTION PANORAMIC VIEWS

Carsten Grünheit, Aljoscha Smolić, and Thomas Wiegand
Heinrich-Hertz-Institute (HHI)
Image Processing Department
Einsteinufer 37, 10587 Berlin, Germany
{gruenheit/smolic/wiegand}@hhi.de

ABSTRACT

A new system for interactive streaming of high-resolution 360° panoramic views over the Internet is presented. The scene is represented very efficiently using MPEG-4 BIFS and displayed at the client using the HHI 3-D MPEG-4 player. The user can navigate interactively through the scene. The navigation decisions are evaluated to trigger the streaming of the data needed to build the visible screen view and to ensure a fluent visualization of the scene. The system layout is designed to support other, more complex photo realistic 3-D environments later on and, thus, will enable the provision of a variety of new, interactive services over the Internet.

1. INTRODUCTION

The Internet has provided us with the possibility to connect and communicate with computers anywhere in the world. New technologies provide increasing transmission bit-rates to the end user. The access to high-speed internetworking for a steadily growing group of users offers the chance to provide services, which rely on the transmission of large amounts of multimedia data.

On the other hand, processing complex data is possible with nowadays personal computers, which offer huge and still increasing processing power of CPUs and graphics adapters. Together with the fact, that storage devices and memory chips became fast, cheap and available in large quantities, we can say that today's computers are well equipped for any kind of even complex multimedia services.

These technological advances have triggered the development of a huge variety of new services in the field of multimedia communication. A particularly popular group of services offers the possibility to access and retrieve visual information about certain places in the world, to virtually visit cities, sights, (even non-terrestrial) landscapes, buildings, museums, etc. A simple media type for this purpose are still images. They are restricted to a predefined viewpoint and have no temporal dimension. Video extends the temporal dimension at the cost of an increased data rate. Another way of providing information about a certain place, building etc. is to model the scenery in a three-dimensional purely virtual environment, e.g. using the Virtual Reality Modeling Language (VRML). The user can navigate through the scene interactively and is free to choose individually preferred views when exploring it. This technology is also used e.g. for network based computer games where users interact not only with the virtual environment, but also with each other.

But just displaying images and videos does not give the user an immersive impression. It does not evoke a feeling of "being part of the scene", above all because of the missing third spatial dimension and the lack of interactivity. On the other hand, purely computer-generated environments are not immersive, either, because scenes generated with reasonable effort do not contain as many details, shadings etc. compared to reality.

In recent years, new technologies for acquisition and visualization of *photo realistic* three-dimensional environments have been developed. Light-field rendering [1] or the concentric mosaics method [2] are examples for the new image-based rendering (IBR) technologies, which treat single images as samples of the plenoptic function [3]. All these technologies have in common that they make use of real image data to create 3-D environments. With each of these methods, immersive impression increases with the amount of image data used. But the more image data are needed to build the scene, the less such a technology seems to be suitable for transmission. Scenes created from video-based rendering (VBR) technologies (e.g. immersive panoramic video [10]) rely on video sequences instead of still images, which further increases the transmission problem.

The popular QuickTimeVR® system [4] is an example for an already available application designed for Internet usage. It allows rotation and zoom within a photo realistic 360° cylindrical or a cubical panoramic view. All information is downloaded completely before display. In order to assure an acceptable reaction time of the system, the visual quality is limited (small images, low resolution, aliasing during display motion).

Our goal is to overcome the transmission problem for 3-D environments and therefore to open up the Internet for new sophisticated IBR and VBR scenarios, which contain very large quantities of data, well beyond sizes that can be downloaded as one piece with reasonable effort.

2. INTERACTIVE STREAMING

A promising approach to introduce high quality image based methods to Internet services is to stream the data from a server interactively. The navigation decisions of the user exploring the scenario trigger the streaming of the image data. Only those data have to be streamed from the server, which contribute to the actually visible part of a scene. The key questions on how to accomplish this task are:

- What kind of representation to choose?
- How to code the data?
- How to stream?
- And how to display and navigate through the environment?

Because of the presumable huge amount of data, there is an inevitable need for high compression. But the applicability of predictive coding schemes is limited, since we want to give the user maximum possible freedom to navigate within a scene. Because the data on the server are pre-encoded, the user's navigation decisions cannot be used when encoding the data. So there is no time line like in video coding or path of movement, which could be exploited for compression. Instead, it is necessary to offer the possibility to access the encoded data as randomly as possible. Furthermore, the necessary resolution is not known in advance. It becomes a dynamic value that depends on the user's currently chosen viewpoint and zoom angle. So there is a need for a scalable coding scheme. Because the network may not provide a throughput that is constantly high enough, the images must be displayed as early as possible and with progressively improving image quality while still transmitting the remaining data.

Another key issue is the delay between the time the data is requested by the client and the time it is received from the server. The roundtrip time for a message in an Internet environment can typically be estimated with about 100 ms. Together with the hardware dependent processing time, this sums up to a significant delay. Hence, if the data are streamed following a user request, no fluent navigation through the scenery is possible. Therefore, interactive streaming demands a pre-fetching strategy, where image data are requested from the server before they actually contribute to the rendered view on the scene.

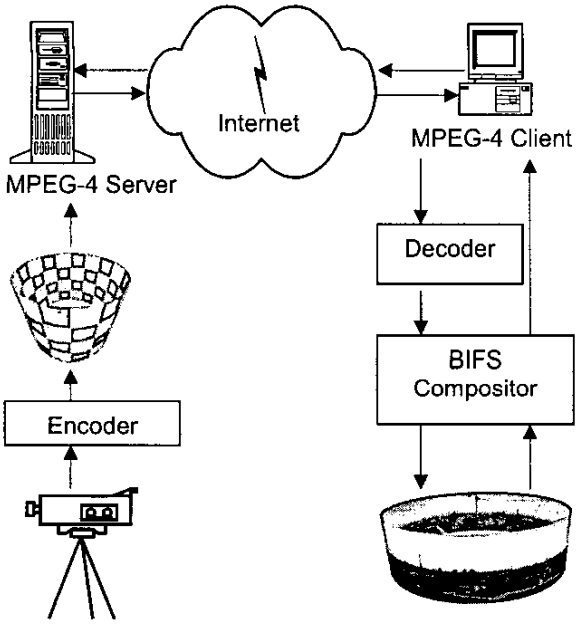

Figure 1: Interactive streaming system

3. STREAMING OF PANORAMIC VIEWS

We have set up a complete interactive streaming system, as shown in Figure 1. As an example of a photo realistic environment containing huge amounts of image data we have chosen 360° high resolution panoramic views acquired by special cameras or by common video cameras in combination with robust global motion estimation and mosaicing tools [5], [6], [7]. Mosaicing denotes the transformation and blending of all images of a considered video sequence into a common image. We developed a scene representation that contains the image and modeling information using MPEG-4 BIFS (BInary Format for Scenes), which is based on VRML. BIFS is designed to represent multimedia content consisting of audio, visual, and object data. It supports data streaming, scene updates and compression. With all these features, it perfectly fits the needs of our system. Furthermore, the representation using a BIFS scene gives us the maximum freedom to extend or change the complete representation easily without changing the whole system.

At the client side, the scene is displayed using the HHI 3-D MPEG-4 player [8]. It allows the user to freely navigate within the scene (e.g. walk, rotate or zoom), even though in our scenario the only completely distortion-free viewpoint is on the cylinder axis with radial viewing direction.

Currently, we use the HHI MPEG-4 client-server architecture [8], which is based on the MPEG Delivery Multimedia Integration Framework (DMIF). DMIF allows to build applications unaware of the delivery technology details, which DMIF hides behind an interface called the DMIF Application Interface (DAI). We use the framework for real time transmission of multimedia content over the Internet.

Neither the streaming framework, nor our MPEG-4 player display unit restrict the project to certain image codecs. Currently, JPEG encoding is used to compress the panoramic image data. This codec provides a good compression and should be supported by any system implementing ISO/IEC 14496-1 (MPEG-4 Systems). We are planning to use JPEG-2000 instead, because many of its features we consider to be important for the targeted application [9]. JPEG-2000 offers a competitive low bit rate compression performance and at the same time random access to the coded bit stream on a tile level and on a group of code blocks (precinct)-level. It also offers different ways of scalability of the bit stream. The progression order of the data can be organized with priority to increasing resolution, quality layers, number of components or even spatial position.

4. SCENE REPRESENTATION AND DISPLAY

There is a need to manage the complexity of the scene to be rendered to ensure fast and fluent visualization. This is due to the fact, that any graphics card shows limitations concerning the size of available texture buffers. We developed a technique based on MPEG-4 BIFS nodes to implement this task. The panoramic image is divided into small patches with size being typically less than 512x512 pixels. Each patch is encoded separately (cf. Figure 2). This simple but very efficient method allows the BIFS scene compositor, which is the core module of our player, to load patch textures into the rendered scene and unload them as soon as they do not contribute to the rendered view anymore.

The same mechanism serves as a trigger for initiating requests for the transmission of image data from the server and allows us to implement a mechanism for pre-fetching image data in advance before the user reveals the affected region of the scene.

More precisely, the method is based on the use of the BIFS visibility sensor node. It detects visibility changes of a rectangular box as the user navigates through the scene. Each patch is linked to an enclosing visibility sensor, as illustrated in Figure 3. Each entry of a sensor into the screen view creates an event that is transformed into a request that is sent to the server. On request, the server transmits the assigned patch data to the client. As soon as the sensor's bounding box does not overlap with the visible view on the scene anymore, the BIFS compositor unloads the texture data. Hence, the sum of all patch data influencing the rendering performance is limited by the number of activated visibility sensors.

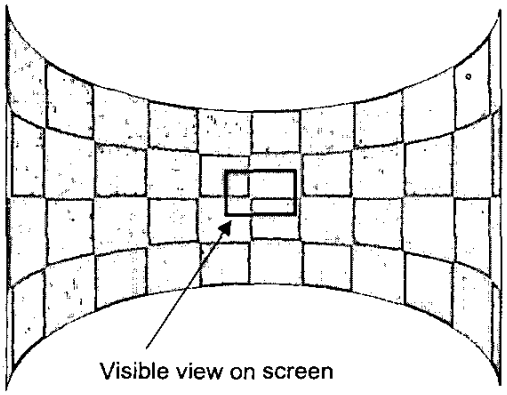

Figure 2: Cylindrical panorama divided into patches

Oversizing the sensors bounding box sizes implements a straightforward method of pre-fetching image data before they actually contribute to the rendered view. For example, in Figure 3, all six patches are requested from the server, even though the two rightmost do not contribute to the screen view. There is a trade-off between oversizing of sensor dimensions and rendering performance: The bigger the sensors are the more patches are loaded into the scene to be rendered. This might reduce rendering performance. But at the same time it ensures that the image data are available when the user reveals a region, even if the navigation speed is fast. The oversize ratio of a patch and it's sensor determines the time available for requesting, receiving and processing the data. Thus, without a scalable coding scheme, it limits the possible navigation speed, if visible artefacts are to be avoided. If the maximum speed is exceeded, the user might reveal parts of patches with no texture to display. These "late" textures pop into their empty positions on the cylinder, as soon as they are available.

The delay between request for and rendering of the patch data is determined by the available bitrate for the elementary streams containing the encoded data and the (hardware dependent) processing time. "Rendering performance" is also a system dependent variable mainly depending on the graphics adapter's frame buffer size, i.e. the memory used to store rendered pixels before they are displayed on the monitor. If not all texture data to be displayed can be placed there, they are redirected to the PC's memory.

Apart from network and hardware requirements, system performance can be influenced by setting the user's freedom to navigate through the scene. A limit for the zoom range, and hence the field of view angle, defines the number of necessary patches and therefore limits the maximum bitrate and amount of texture data to be rendered. Another possibility to reduce the necessary transmission bandwidth is to restrict the maximum navigation speed to a lower value. With these parameters, the system can be optimized for different environments. Another idea is to shape the visibility sensors differently.

Currently, the scene representation is fully implemented using BIFS nodes. Thus, our whole system for interactive streaming of high-resolution panoramic views is fully consistent with the MPEG-4 standard.

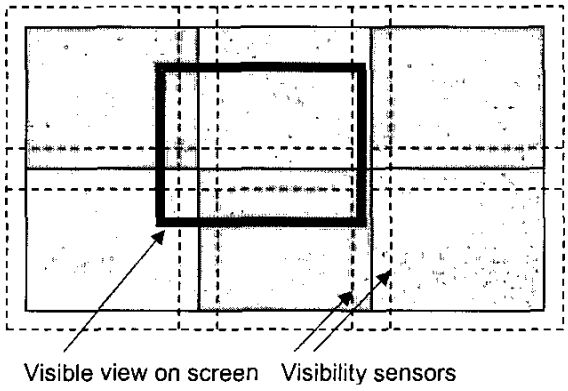

Figure 3: Patches with enclosing visibility sensors

5. EXAMPLES

Our system allows the interactive streaming of panoramic views of arbitrary sizes. We tested our system with images sized up to 13200x2600 pixels requiring a storage space of roughly 100 MB. We are working on images sized about 10000x60000 pixels, which corresponds to 1.7 GB of uncompressed 24-bit RGB data, which is by far too much to be downloaded in advance.

An example for smaller image sized about 3700x400 pixels can be viewed on our project web-page at http://bs.hhi.de/~grunheit/3DStreaming.html.

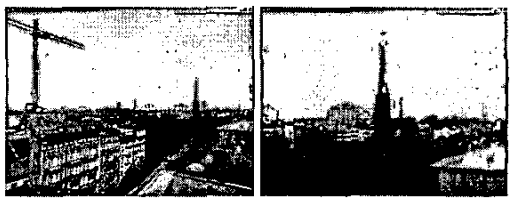

Figure 4: Rendered views from cylindrical panorama

As an example, Figure 4 shows two screenshots of rendered views of a 360° high-resolution mosaic created on the roof of the

HHI building. We rotated and zoomed to get from the view in the first picture to the one in the second. Figure 5 gives an overall view on the whole cylindrical panorama.

In our scene representation a 13200x2600 pixel sized panorama allows a maximum vertical camera angle of 63° without revealing the cylinder edges. With the panorama being divided in 26x6 patches, the angle set to a reasonable value of 28° (i.e. up to 4x4 patches visible at a time), and a visibility sensor oversizing of 40%, it is possible to rotate within the panorama without "late texture" errors at a maximum speed of about 5°/sec with an acceptable rendering performance. These results are obtained assuming an error-free transmission and an available bandwidth of not more than 1 Mbit/sec per elementary stream. With about 4 open streams at a time, the transmission rate is suitable for wideband xDSL connections. Setting the field of view angle and rotation speed to smaller values allows very smooth rotation. Increasing the oversize factor allows for even higher rotation speeds, but at the cost of less smooth motion. Decreasing the factor improves smoothness of rotation at the cost of more "late texture" errors, which can be compensated by a lower rotation speed. The system has been tested on a PC with 2x2 GHz Intel® Xeon processors and a Nvidia® Quadro2 Pro graphics adapter. Using a 1.3 GHz Intel® Pentium 3 Mobile CPU and a Nvidia GeForce 2 Go graphics adapter, comparable results can be achieved at a speed of about 3.3 °/sec.

6. CONCLUSION AND OUTLOOK

We have presented a new system, that allows to provide high-resolution panoramic views over the Internet. Our interactive streaming system offers the opportunity to navigate through the scenery without downloading all the image data from the server in advance. Instead, only data contributing to the visible screen view and additional data to ensure a fluent navigation through the scene are transmitted on request. The system is fully compliant with the MPEG-4 standard.

Our next step will be the integration of JPEG-2000 to implement scalability to the system and to improve random access to the coded image data. Further effort will be spent on finding more sophisticated pre-fetching strategies. Problems with transmission errors and unreliable network performance need to be targeted.

Our interactive streaming system for high-resolution panoramic views is intended to mark the starting point on our way to build efficient streaming systems for complex photo-realistic three-dimensional environments created by image based rendering technologies. This will be the basis for the provision of a variety of new interactive services over the Internet.

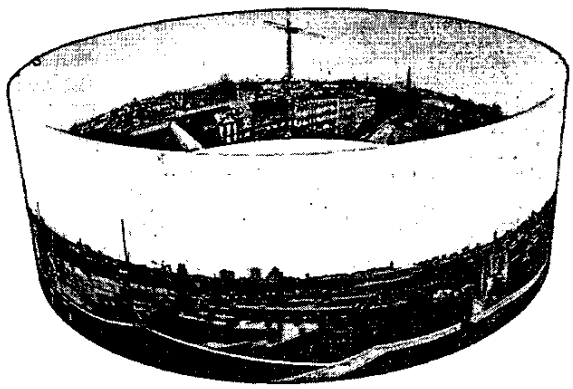

Figure 5: Cylindrical panoramic view

REFERENCES

[1] M. Levoy and P. Hanrahan, "Light Field Rendering", Proc. ACM SIGGRAPH, pp. 31-42, August 1996.

[2] H.Y. Shum and L.W. He, "Rendering with Concentric Mosaics", Proc. ACM SIGGRAPH, pp. 299-306, August 1999.

[3] E.H. Adelson and J. Bergen, "The plenoptic function and the elements of early vision", In Computational Models of Visual Processing, pp. 3-20, MIT Press, Cambridge, MA, 1991.

[4] S.E. Chen, "QuickTime VR – An Image-Based Approach to Virtual Environment Navigation", Proc. ACM SIGGRAPH, pp. 29-38, August 1995.

[5] A. Smolic and J.-R. Ohm, "Robust Global Motion Estimation Using a Simplified M-Estimator Approach", Proc. ICIP2000, IEEE International Conference on Image Processing, Vancouver, Canada, September 2000.

[6] A. Smolic, "Robust Generation of 360° Panoramic Views from Consumer Video Sequences", to appear Proc. VIPromCom2002, IEEE International Symposium on Video/Image Processing and Multimedia Communications, Zadar, Croatia, June 16.-19. 2002.

[7] A. Smolic and T. Wiegand, "High-Resolution Video Mosaicing", Proc. ICIP2001, IEEE International Conference on Image Processing, Thessaloniki, Greece, October 2001.

[8] A. Smolic, Y. Guo, J. Guether and T. Selinger, "Demonstration of Streaming of MPEG-4 3-D Scenes with Live Video", ISO/IEC JTC1/SC29/WG11, MPEG01/M7811, Pattaya, Thailand, December 2001.

[9] A. Skodras, C. Christopoulos, T. Ebrahimi, "The JPEG 2000 Still Image Compression Standard", IEEE Signal Processing Magazine, pp. 36-58, September 2001.

[10] Thomas Pintaric, Ulrich Neumann, Albert Rizzo, "Immersive Panoramic Video", Proceedings of the 8th ACM International Conference on Multimedia, pp. 493.494, October 2000.