Bandwidth Reduction Techniques for Remote Navigation Systems

Pere-Pau Vázquez and Mateu Sbert

Institut d'Informàtica i Aplicacions, University of Girona,
Campus Montilivi, EPS, E-17071 Girona, Spain
pvazquez@ima.udg.es

Abstract. In this paper we explore a set of techniques to reduce the bandwidth in remote navigation systems. These systems, such as exploration of virtual 3D worlds or remote surgery, usually require higher bandwidth than the common Internet connection available at home.
Our system consists in a client PC equipped with a graphics card, and a remote high-end server, which hosts the remote environment and serves information for several clients. Each time the client needs a frame, the new image is predicted by both the client and the server and the difference with the exact one is sent to the client. To reduce bandwidth we improve the prediction method by exploiting spatial coherence and wiping out correct pixels from the difference image. This way we achieve up to 9:1 reduction ratios without loss of quality. These methods can be applied to head-mounted displays or any remote navigation software.

1 Introduction

Recent advances in computer science have allowed dramatic breakthroughs at the beginning of the new century, such as remote computer-assisted surgery, 3D gaming, or the creation of compelling special effects in films. However, there are still some problems to be solved to obtain attractive applications such as 3D remote exploration or gaming. One of the most important problems comes from the limited bandwidth of common internet connections. Many real time rendering systems require the transmission of images at 15-30 frames per second to guarantee a continuous sensation of movement. Moreover, these frame rates can only be obtained with high-end systems, despite the latest chipset releases of graphics hardware.

In this paper we present some techniques that can help to reduce the necessary bandwidth without any noticeable cost in the final quality. We have taken advantage of Image-Based Rendering methods and used a collaborative scheme between client and server. Reduction ratios of 5:1 in rotations and up to 9:1 in translations over previous systems that also use image compression and view prediction can be achieved. The ratios refer to the amount of information sent through the network. The increase in computing cost is perfectly assumably by the client processor, as the only required operation is a forward warping plus an image update.

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2330, pp. 249–257, 2002.

The remainder of the paper is organized as follows: in Section 2 we review the previous work, in Section 3 we present our techniques for bandwidth reduction, the results obtained are discussed in Section 4, and finally in Section 5 we conclude and analyze possible future work.

2 Previous Work

During the second half of the last decade, a set of new rendering methods, called Image-Based Rendering, have been developed. All of them have in common the use of images in some stage to partly or completely substitute the geometry of the scene [1–3]. The use of precomputed images simplifies the rendering algorithms thus obtaining realistic effects at low cost. Although most of these methods have huge memory requirements and are therefore not suitable for dealing with very complex scenes in a low-end PC, some of these techniques can be used to accelerate rendering or to compensate for network latencies in remote navigation tasks. Mark *et al* [4] use two images generated in a server and warp them together to generate new views at interactive frame rates. Some commercial products such as QuickTime VR [5] offer panoramas over the network which are divided into pieces, so that the client can view them before they are totally downloaded, however, a large portion of the panorama has to be received before it may be viewed. Moreover, these systems only allow for camera rotation and zooming, and it is difficult to extend them to handle translations.

Our approach is a technique to reduce bandwidth which involves image compression and image-based rendering. A cooperative scheme is described in [6], where the server generates a high-quality rendering and a low-quality one and subtracts them, sending the result to the client. The client generates a low-quality image and adds the one sent by the server. This way the author obtains better compression and avoids the low quality of JPEG scheme in edges and smooth shading. On the other hand, this method requires that the geometry is also sent to the client in order to allow it to generate the low-quality image. Biermann *et al* [7] use a similar collaborative scheme. To reduce bandwidth, the client performs a forward warping to predict the new view, the server does the same and also computes the correct view. Then, the predicted view is subtracted from the exact one and the difference view is compressed. This way they achieve reduction ratios of 55% on small movements of the camera. Cohen-Or *et al* [8] use a compression scheme which reduces up to one order of magnitude the bandwidth, compared to a MPEG postrendered sequence with equivalent quality. Their compression scheme requires also that the client has access to the geometric data and exploit spatial coherence in order to improve compression ratios.

3 Bandwidth Reduction Techniques

In this section we present two bandwidth reduction techniques, a two-level forward warping and a compressing method which avoids sending information cor-

responding to correctly predicted pixels to the client. We use a similar architecture to the one presented in [6], but improving the tasks on the client and on the server side. Figure 1 depicts this architecture and the tasks. The client predicting system is enhanced by adding a simple and low cost two-level of detail forward warping. The server side reduces bandwidth by only transmitting the pixels to be updated, not the whole image (with the aid of a bitmap that encodes the pixels that have to be changed). This way we achieve a reduction of bandwidth of 5 to 1 over the previous method for rotations of up to 30 degrees and a reduction of up to 9 to 1 for translations.

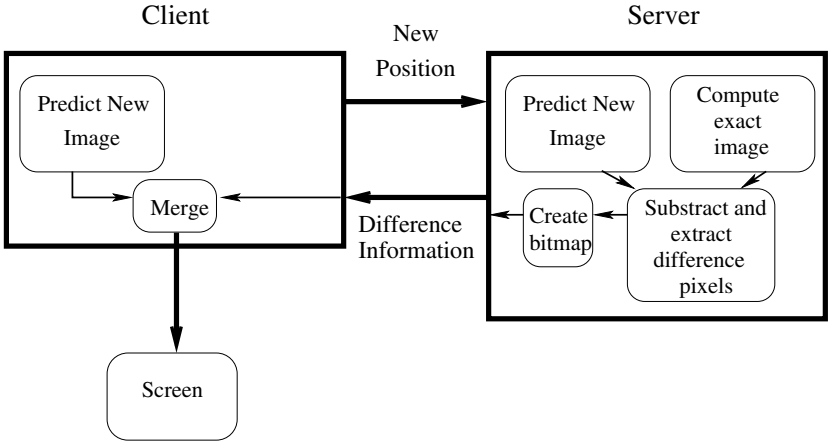

Fig. 1. The client-server remote navigation system. Instead of sending every frame through the network the server only sends the difference pixels, together with a bitmap which indicates the pixels to be modified. The client predicts the new view using a two-level forward warping.

3.1 Two-Level Forward Warping Known Pixels

In a similar way to Biermann *et al* [7] we use a forward warping to predict the next image. However, we will exploit spatial coherence. Some possible techniques are: to create a mip-map of the resulting image (after warping) and fill in the gaps, or to compute correct splat sizes for each point. On the other hand, these techniques can be quite costly and we want a very cheap method. Our system performs a two-level forward warping of the valid pixels in previous image. The first one consists into a reprojection of each point using the OpenGL ([9]) primitive *GL_POINT* assigning to the points a size (*GL_SIZE*) of four. Then, the depth buffer is cleared and the points reprojected again with a size of one. As some of the present gaps should have the same colour as closer pixels, this method reduces the visually unpleasant holes, and correct pixels of the predicted

image increase by a factor of up to a 40% percent, and an average of a 20% percent (bandwidth reduction of 1.25:1 ratio) in rotations. This method is not new in the sense that it can be considered a particular case of splatting, but it is cheap and only uses graphics hardware commonly available in most personal computers. Figure 2*a* depicts the difference in number of wrong pixels for simple and two-level warping strategies for different rotations. Note that the number of wrong pixels in the second case grows slower and the percentage of reduction is almost constant in rotations of up to 30 degrees. We have also tested our method for translation transformations, in this case, as the pixels are reusable longer, the amount of correct pixels grows up to the 68% and an average of 60% (which represents a bandwidth reduction of 2.5 to 1). These results are depicted in Figure 2*b*.

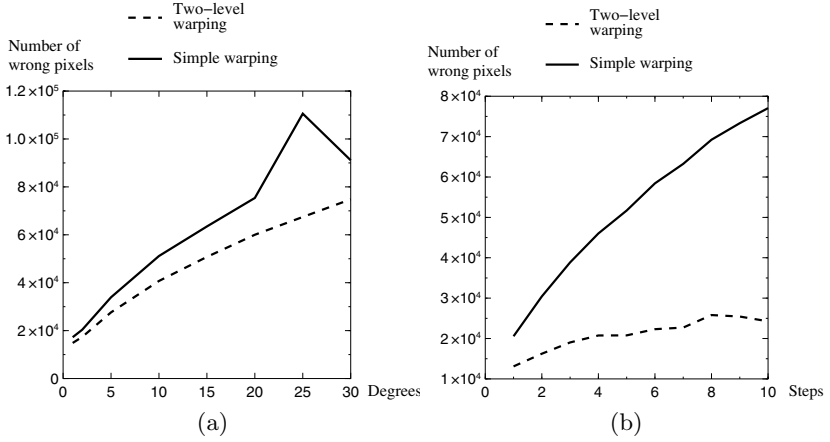

Fig. 2. Comparison of the number of wrong pixels for simple warping and two-level warping in rotations (a) and translations (b). Note that for translations the improvement is higher as the projected colours are correct longer.

In Figure 3 we can see an example of rotation. In this case the camera rotates ten degrees. Figure 3*a* has been created by only warping the pixels of the previous image, while 3*b* has been created with our two-level forward warping, the amount of difference pixels has been reduced in a 20 percent. Moreover, image 3*b* can be used if latency is high as it does not present as many noticeable artifacts as 3*a* does. Figures 3*c* and 3*d* show the difference images which have to be sent through the network.

Figure 4 shows the images resulting from a translation of five steps in world coordinates. Note that the amount of reused pixels is higher than in rotations and thus our two-level warping performs better than simply warping known points. Moreover, the predicted image could be used in case of high network latency. Figure 4*a* shows the image obtained with simple warping and 4*b* with

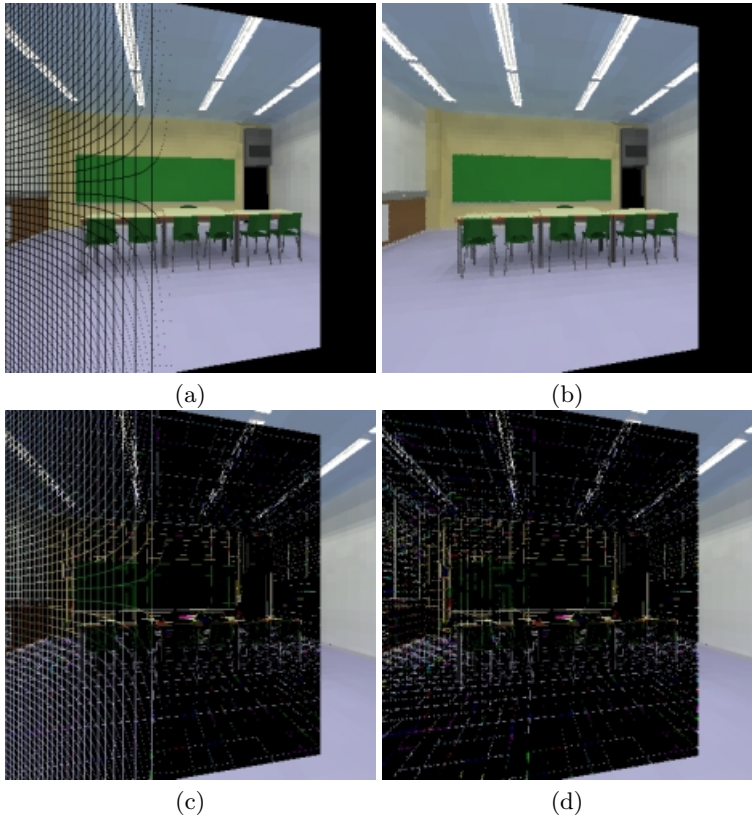

Fig. 3. Predicted images for a rotation of ten degrees. Figure *(a)* has been created using a simple forward warping and Figure *(b)* was created using a 2-level forward warping. The number of invalid pixels decreases a 20%. Figures *(c)* and *(d)* show the difference images respectively.

our two-level warping. Figures4c and 4d show the respective difference images. We obtain an improvement of up to a 68% and the average is about the 60%.

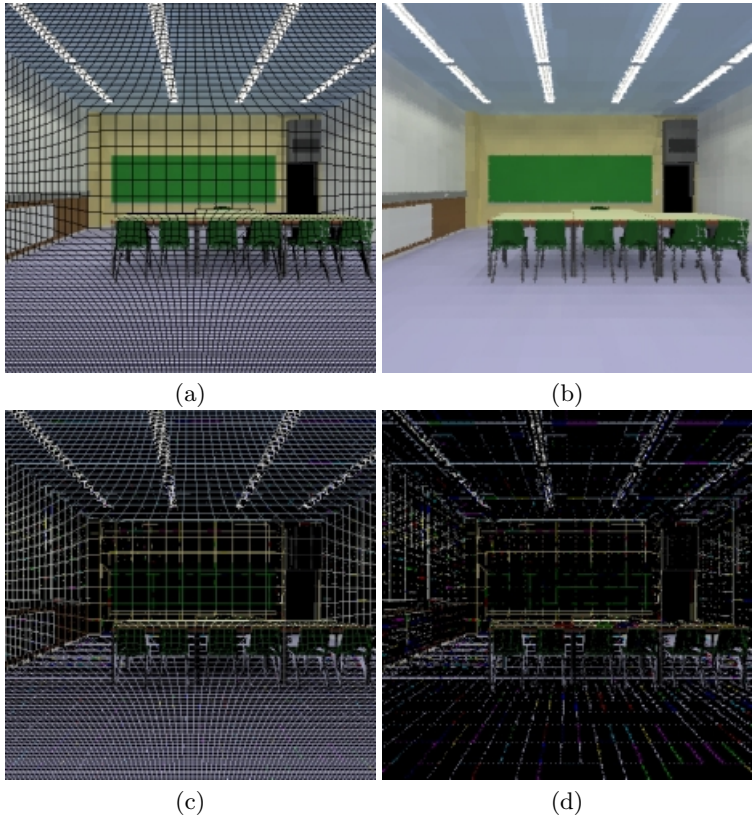

Fig. 4. Predicted images for a translation of 5 steps. Figure *(a)* has been created using a simple forward warping and Figure *(b)* was created using a 2-level forward warping. The number of incorrect pixels decreases a 60%. Figures *(c)* and *(d)* show the difference images respectively.

3.2 View Compression

Up to now we have seen how to reduce the amount of information to be sent through the network thanks to the use of spatial coherence. The previous method in [7] sent the total difference image, although compressed, and achieved reductions of up to 55% with small movements of the camera. However, with this method, some pixels that do not contribute to the final image are sent through the network. Our purpose in this paper is not to study new compression techniques, but we are going to take advantage of the fact that, as some of the

information will not be needed, such as the pixels which are not different, we can wipe them out, and only send through the network the different ones. These pixels can also be compressed. With this method we can obtain up to 8:1 reduction ratios for small movements and an average of 5-6:1 reduction over the JPEG compressed difference image.

In order to do it we have to inform the client which pixels have to be updated. This can be achieved by creating a bitmap containing values of 1 for the pixels which have to be changed and zeroes elsewhere. The size of the bitmap will be 1/24th of the total image, as each pixel colour is usually encoded with three bytes and a single bit is enough to determine if it has to be changed. Thus, if the difference pixels are less than 23/24 of the total (about 96%) pixels in the image, we will save memory. In our tests we have found that the number of *valid pixels* is far higher than this 4 per cent for rotations of more than 30 degrees. In order to perform warping at every frame, the depth values of the updated pixels should also be transmitted. With our system this poses no problem, as the same bitmap may also be used to encode the Z values to be sent and thus we will have equivalent savings respect to the previous methods, which send these depth values. Our method achieves good results with no loss of quality because the number of pixels that can be reused is in general higher than a 4% and thus, our masking method can save memory. A lossy scheme could be competitive with ours, but our method has the advantage that we are not losing quality. Moreover, we could further reduce the amount of information to be sent by also using a lossy method [10] to compress the resulting data, which is smaller than the original one if the number of pixels to reuse is higher than a 4%.

4 Results

We have tested our method with different scenes and different movements of the camera, rotations and translations. The comparison between the size of information with our method and JPEG compressed difference images is depicted in Figure 5*a*. We can see that for the classroom scene, we achieve bandwidth reductions of up to 8:1 for small rotations, and the average reduction stays high for wide movements: 5:1 for rotations of 30 degrees.

We have also tested translations. As we have already mentioned, translations show a better behavior due to the fact that the pixels can be reused longer. With our method we obtain improvements of up to 89.5% (ratio of 9.58:1) and an average of 85% of savings. In Figure 5*b* we see the reduction ratio for movements of 1 to 10 steps in the viewing direction.

5 Conclusions and Future Work

We have presented two techniques for bandwidth reduction in remote navigation systems. These techniques can be applied to any system which has a limited bandwidth. Our method consists in two parts, one concerning the client side, and another one which uses the information available on the server side. In the first

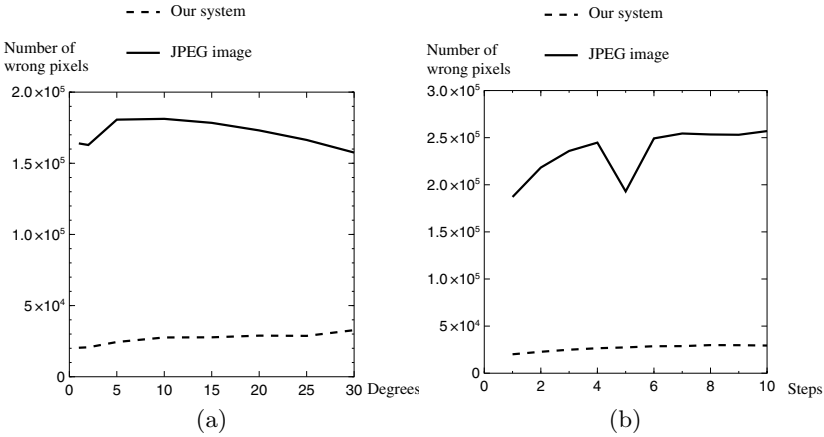

Fig. 5. Comparison of results. With our method we achieve an improvement of up to a 80% on rotations, that is, theoretically we can send 5 times more information with the same bandwidth. For translations improvements are better, up to a 85% of savings.

one a two-level forward warping is performed in order to exploit spatial coherence. The second task consists in reducing the information to be transmitted by only sending the pixels to be updated, with the aid of a bitmap that encodes their positions. This way we achieve up to a 9.5 to 1 reduction ratios. The average ratio is about 5:1 for wide movements of even 30 degrees. Our contribution is the combination of forward warping (adding a previous step changing pixel sizes to exploit spatial coherence) and the use of masking tailored to reduce the amount of information sent through the network. In spite of their simplicity, these methods lead to very good bandwidth reduction ratios.

In the future we want to apply our method to dynamically varying environments. In this case the server should send information about the variation in the visible pixels (such as a vector field, combined with the bitmap that indicates the pixels to be moved) to obtain temporally valid views in the client side.

Acknowledgments

This work has been partially supported by BR98/1003 grant of Universitat de Girona, SIMULGEN ESPRIT project #35772, and TIC 2001-2416-C03-01 of the Spanish Government.

References

1. Marc Levoy and Pat Hanrahan. Light field rendering. In *Computer Graphics Proceedings (Proc. SIGGRAPH '96)*, pages 31–42, August 1996.
2. Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Computer Graphics Proceedings (Proc. SIGGRAPH '96)*, pages 43–54, 1996.

3. L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Proc. of SIGGRAPH 95*, pages 39–46, August 1995.
4. L. McMillan W. R. Mark and G. Bishop. Post-rendering 3d warping. In *Proc. of 1997 Symposium on Interactive 3D Graphics*, pages 7–16, New York, April 1997. ACM Press.
5. Shenchang Eric Chen. Quicktime vr - an image-based approach to virtual environment navigation. In *Computer Graphics Proceedings (Proc. SIGGRAPH '95)*, pages 29–38, 1995.
6. Marc Levoy. Polygon-assisted jpeg and mpeg compression of synthetic scenes. In *Computer Graphics Proceedings (Proc. SIGGRAPH '95)*, pages 21–28, August 1995.
7. H. Biermann, A. Hertzmann, J. Meyer, and K. Perlin. Stateless remote environment navigation with view compression. Technical Report TR1999-784, Media Research Laboratory, New York University, New York, NY, 1999.
8. Daniel Cohen-Or, Yair Mann, and Shachar Fleishman. Deep compression for streaming texture intensive animations. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, Annual Conference Series, pages 261–268, Los Angeles, 1999. ACM Siggraph, Addison Wesley Longman.
9. Silicon Graphics Inc. Open gl web page, 2001. Specification document, available from http://www.opengl.org.
10. Hee Cheol Yun, Brian K. Guenter, and Russell M. Mersereau. Lossless compression of computer-generated animation frames. *ACM Transactions on Graphics*, 16(4):359–396, October 1997. ISSN 0730-0301.