Attention Guided MPEG Compression for Computer Animations

Rafal Mantiuk
Technical University of Szczecin, Poland
rafal.mantiuk@wi.ps.pl

Karol Myszkowski
Max-Planck-Institut für Informatik, Germany
karol@mpi-sb.mpg.de

Sumanta Pattanaik
University of Central Florida, USA
sumant@cs.ucf.edu

Abstract

In this paper we present a framework that aims at delivering high-quality and low-bandwidth 3D animation at real-time rates. In this framework we combine a real-time rendering, MPEG–4 video compression and a model of visual attention. We use the attention model to control quality/bit-rate of MPEG compression across a single frame. OpenGL is used to generate animation sequences in real-time.

CR Categories: I.3.2 [Computer Graphics]: Graphics Systems - Distributed / Network Graphics; I.3.3 [Computer Graphics]: Picture/Image Generation - Viewing Algorithms; I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction techniques.

Keywords: visual attention, video compression, MPEG, streaming, virtual environment

1. Introduction

Modern graphics hardware has brought rich 3D interactive animations to low-end desktops. Yet, graphics hardware is not available to small electronic devices, like cell phones and PDAs. Live video streaming of synthetic 3D imagery can fill this gap and deliver 3D content to mobiles and other small networked devices. Although video can be streamed to the cell phones using existing solutions, its content is limited to pre-encoded sequences and quality is severely limited with available bandwidth. To address those shortcomings we generate video content using real-time 3D rendering and then efficiently compress it using enhanced MPEG compression. One of the biggest technical challenges that must be overcome to build such solutions is combining fast 3D rendering and effective video compression into a single framework capable of running in real-time. Moreover, such framework should produce high-quality and low-bandwidth bit-stream that can be sent, decoded and played on the destination device.

We combined real-time rendering with video compression using a model of visual attention. The model of visual attention simulates a mechanism of human visual system responsible for selection of the salient fragments from the visual information. Based on the prediction of the saliency of the scene objects by the attention model, MPEG encoder controls compression quality. The attention model uses OpenGL generated images for spatial processing, and a 3D model of the scene to locate the salient object in the animation frame.

2. Previous Work

In this chapter we focus on three issues – in the first part we briefly introduce the computational models that simulate function of human brain that is responsible for focusing visual attention. In the second part we give a few examples of applications of the visual attention models to improve rendering, compression or video streaming. In the last section of this chapter we discuss a few solutions for efficient streaming of computer animations.

2.1. Models of Visual Attention

The amount of visual information reaching human visual system (HVS) at any time is huge compared to what it can process. HVS handles this problem by choosing only a manageable size of information at a time. This way information is processed partly “sequentially” and the amount of data that should be analyzed is reduced to manageable size. Visual attention is responsible for selection what data and in which order should be processed.

Yarbus [1967] showed that subjects fixate their attention on similar regions of interest, however, the order in which they fixate may be different. Treisman’s Feature Integration Theory (FIT) [Treisman and Gelade 1980] tries to explain the mechanism that controls those fixations. Triesman suggests that visual signal is decomposed into feature maps, where each feature map is sensitive to a particular impulse, for instance color, orientation, shape or motion. Visual attention is responsible for binding those feature maps into a single phenomenal object, which can be further processed. This approach assumes that attention always focuses on spatial regions. Tipper and Weaver [Tipper and Weaver 1998] argue against this theory and suggest that attention rather fixates on objects instead of regions. Their approach was supported by the results of psychophysical experiments, which showed that focusing attention on two neighboring objects was far more difficult than on a single one.

There have been several attempts to model and computationally simulate human visual attention. In our research we choose a model developed by Itti [2000]. The model, influenced by the Feature Integration Theory, decomposes the image into separate feature maps. Feature maps represent color, intensity and orientation. We used a simplified version of this model in our research. The general architecture of the model is shown in Figure 1. In this figure and in our research, we have not included orientation feature map because of high computational cost of that element and limited influence on saliency prediction. Detailed description of the complete model is given in the original paper [Itti 2000].

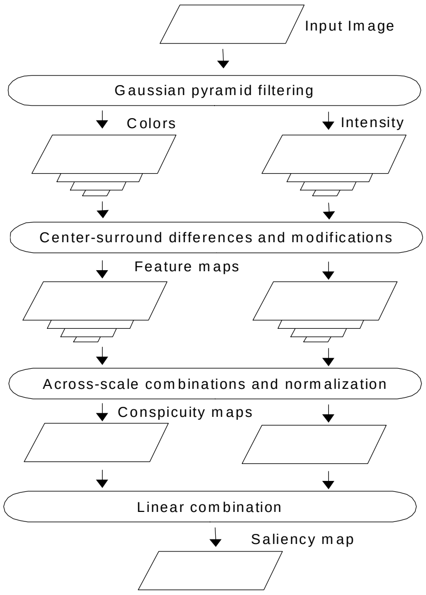

Input Image
Gaussian pyramid filtering
Colors
Intensity
Center-surround differences and modifications
Feature maps
Across-scale combinations and normalization
Conspicuity maps
Linear combination
Saliency map

Figure 1. Architecture of the attention model used in our rendering and video compression framework. This is a simplified version of the model developed by Itti [2000]

2.2. Applications of Attention Models

Since the first computational models of visual attention were proposed, substantial work has been done to find applications of those models in rendering, compression or video transmission.

Yee et al. [2001] used attention model to control accuracy of indirect light computation in RADIANCE. Their solution shows 6x – 9x rendering speedup without visible degradation of quality. However, the attention model used in their solution requires substantial time to compute and is useful only for off-line rendering.

One of the first applications of attention model in real-time rendering was proposed in [Haber et al. 2001]. In their system, view-dependent illumination is updated progressively, so that the objects detected as the most salient are updated in the first order. Real-time performance is achieved by using a simplified version of the Itti's attention model.

Osberger et al. [1998] suggested that efficiency of MPEG encoding could be improved if compression quality of particular macroblocks would vary depending on their perceptual importance. His attention model takes into account both the effect of visual masking and attention. The gain on bit-stream size is between 3% and 10%, depending on animation. Osberger noticed that better compression could be achieved if animation does not contain many moving objects. In this paper we took a similar approach to MPEG compression, however, unlike Osberger's work, we focus on improving compression of synthetic 3D image sequences instead of natural video sequences.

Border and Guillotel [2000] proposed a similar perception driven compression mechanism. They compute Quality Map and use it to vary compression quality between macroblocks and to filter out those spatial frequencies that are not visible for human observer. Filtering spatial frequencies improves compression performance.

Yang et al [1996] presented an interesting application of visual attention in low-bandwidth video conferencing. Their work is based on a simple observation that most of the attention in videoconferencing is focused on faces. They provided hardware and software solution to follow the position of faces in an animation recorded with a camera. They encoded data that was required to animate only faces and then sent it to other end of the wire, thus reducing substantially the bandwidth requirement.

2.3. Streaming of Computer Animations

Attempts have been made to develop efficient video streaming framework to deliver rich and interactive 3D content to low-end desktops. The research focused mainly on reducing size of data transferred across a network.

Levoy in [1995] suggested that a high-end graphics server could send a difference between high-quality, ray-traced images and images rendered using OpenGL. In his approach, a client machine renders OpenGL sequence and reconstructs high quality image by adding the received difference to the frame-buffer. The encoding of the difference sequence between high quality image and low quality OpenGL rendered image sequence is compressed using fewer bits than original high-quality sequence. Though such solution reduces necessary bandwidth, it is not very effective for textured scenes – the textures are either sent over the network or not included in OpenGL rendering. Both cases significantly decrease savings on the bit-rate of the video stream.

Another approach to streaming computer animations involves warping reference images on the client-side [Cohen-Or et al. 1999; Mann and Cohen-Or 1997]. A client, based on geometry information and a reference image, renders several consecutive frames without actually downloading those frames from the server. Unfortunately warping causes artifacts, for example gaps in the areas where data is missing on the reference image. To fill those gaps, only the pixels that cover the gaps are sent to the client every frame. Using such approach, Cohen-Or et al. [1999] reported a reduction of the bit-stream size up to a few percent of a corresponding MPEG-2 stream. However the result did not take into account the transfer of geometry, which had to be available both for the server and the client.

Ilmi Yoon and Ulrich Neuman [2000] used Image Based Rendering to render images on the client side. The main advantage of their solution was that there was no need for geometry data on the client-side. A reference image with depth information was enough to render several consecutive frames of a walk-through animation. Similarly as in the previous approach, only missing or invalid pixels were sent to the client each frame and a reference image was transferred every n-frames.

All of the above solutions are restricted mainly to walk-through animations, where frames show only small changes in time. Those methods are inefficient for rapid scene changes or fast camera movement. Although they aim at delivering high quality animations to weak clients, they require certain 2D and 3D capabilities, which are often not available on small electronic devices. The main focus of our work is to provide high quality interactive 3D content on small electronic devices whose rendering capabilities are limited, but have hardware support for MPEG decompression (for example mobile video phones). We use MPEG streaming to encode and to send variety of synthetic image sequences at relatively constant bit-rates.

3. Framework

Our rendering and video compression framework consists of three co-operating subsystems running on the server: Rendering, MPEG-4 Video Compression and Attention Model. The rendering system generates sequence of frames. The frames are compressed to MPEG-4 bit-stream. Standard MPEG uses a fixed quantisation table to quantise the DCT coefficients of the image blocks. Instead of using a fixed quantisation table we use a dynamic table whose quantisation coefficients change depending on the saliency of the frame macroblocks. Such use of dynamic quantisation allows us to aggressively compress the image stream while maintaining the video quality. Our attention model computes the saliency of the image blocks. The three subsystems and the data flow between them are shown in Figure 2.

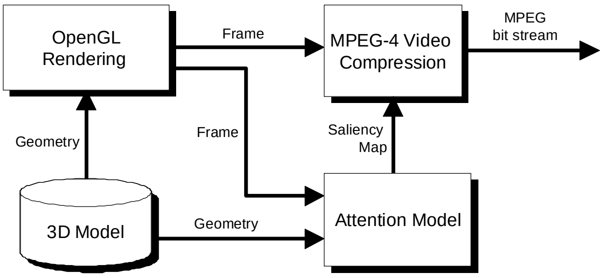


Figure 2. Overview of the rendering and video compression framework

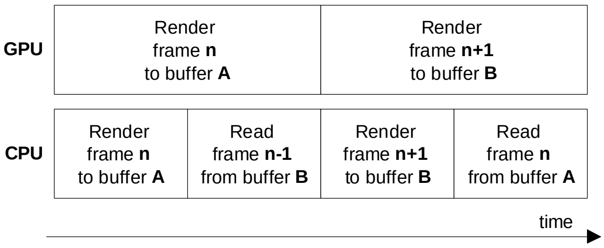
The framework is designed to be suitable for real time applications. Using this framework it will be possible to render, compress and stream video over the Internet and at the same time handle interaction from the user. In this paper we focus on the issues related to improvements in MPEG compression efficiency.

In the following sections we will describe in detail each part of our framework.

3.1. Rendering

OpenGL rendering subsystem is responsible for delivering animation frames to the MPEG-4 encoder and to the attention model. The animation is usually a walkthrough sequence defined on several B-spline interpolated key-frames. OpenGL rendering was chosen to meet real-time requirements.

Each rendered frame is not only displayed on the screen but also compressed to the MPEG-4 stream and analysed by the attention model. Therefore frame buffer must be transferred from graphics memory to the main memory. Such operation turns out to be actually quite ineffective. There are two reasons of that: firstly memory transfer from graphics memory to main memory is several times slower than transfer in opposite direction. Secondly, a processor must wait for OpenGL rendering to finish before glReadPixels operation can be executed. The latter problem can be solved by rendering into two texture buffers (or pixel buffers) instead of a single frame buffer, as suggested in [Zeller 2002]. When using two texture buffers *A* and *B*, a frame *n* is rendered into buffer *A*, then frame *n-1* is read from buffer *B*, without waiting for OpenGL to finish rendering into buffer *A*. In the next step frame *n+1* is rendered into buffer *B*, which has been already read. The order of rendering and buffer reads is shown in Figure 3.



GPU	Render frame **n** to buffer **A**		Render frame **n+1** to buffer **B**	
CPU	Render frame **n** to buffer **A**	Read frame **n-1** from buffer **B**	Render frame **n+1** to buffer **B**	Read frame **n** from buffer **A**

Figure 3. Rendering into two texture buffers for faster glReadPixels() operation.

3.2. Attention Model

The primary role of the Attention Model in our framework is the prediction of saliency of various regions in an animation frame. Attention Model takes a rendered image and the 3D model of the scene as an input and generates a saliency map. Saliency map is a bitmap of the resolution adjusted to the number of MPEG macroblocks. Each 'pixel' value in such bitmap indicates how likely a particular block attracts observer's attention. An example of the saliency map and the input image is shown in Figure 4.

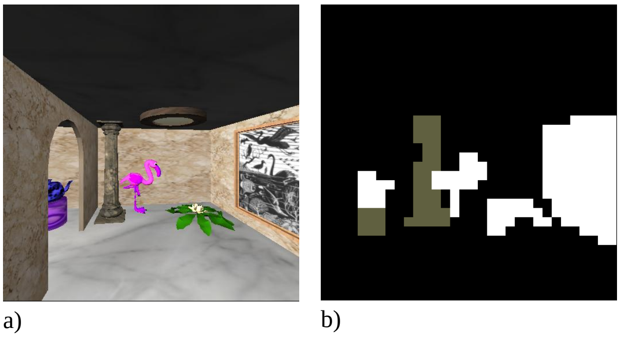


Figure 4. a) Input image and b) Saliency Map. Bright areas indicate objects that are most likely to be attended.

We did not couple our framework with any particular attention model. We rather made it open to any model, which can be 'plugged' to our framework as long as its implementation follows a well defined interface. For our tests we chose two models: Itti's model and a model based on user defined hints. The latter model requires that saliency of objects is predefined by the user in the 3D model of the scene. It is actually a 'faked' model, as it cannot compute saliency automatically. However, it proved to be useful for testing our framework, especially when we wanted to eliminate inaccuracies of attention mechanism from the results.

Before using Itti's attention model we had to adapt it to real-time constraints. Itti's algorithm cannot be executed in real-time on a single processor machine. However, it was shown the algorithm could process 30 frames per second when run on parallel system [Itti 2002]. Instead of limiting our framework to multi-processor systems, we followed [Haber et al. 2001] and removed from the algorithm the most time-consuming part – computing feature maps for orientation. Also the resolution was reduced to match number of macroblocks in MPEG-4 animation. Higher resolution would not improve much the accuracy of the model, but it would have a significant impact on the performance.

Another change to Itti's model was made to improve it's accuracy. Itti's model can be regarded as spatial attention model because attention is assigned to spatial regions on the image, also described as a 'spotlights' of attention. This is in contrary to the recent psychophysical studies, showing that human visual system assigns attention to objects rather than regions (see [Jarmasz 2001] for the discussion on object-based versus spatial attention). We follow those findings and extend Itti's model to pick up the most salient objects instead of regions. Our task is greatly simplified as we can make use of 3D model of the scene. Therefore complex task of image segmentation, necessary in case of natural images, is reduced to simple geometrical projection of 3D objects on the image plane.

3.3. Video Compression

The result of the attention model – saliency map – is used to choose the best ratio of compression quality and bit-stream size across a single frame. The objects on the frame that most likely attract attention are compressed with higher quality than unnoticeable background.

MPEG-4 video compression standard offers a mechanism for adjusting compression quality between macroblocks. One additional bit in the macroblock type identifier is used to indicate that a quantiser scale should change for that macroblock. If such bit is set, two additional bits indicate how much, quantiser scale should change between two macroblocks. Unfortunately such mechanism turned out to be too restrictive and too ineffective for our purpose. A saliency map usually contains rapid changes in the saliency values between two neighboring blocks. MPEG-4 stream can only smoothly change quantiser scale (*qscale*), thus it requires several macroblock to set *qscale* to the appropriate value. As can be seen in Figure 5-a, *qscale* values of the MPEG-4 stream produce a "blurred" image of the actual saliency map predictions. Also many bits are wasted on encoding change in a few consecutive blocks instead of one.

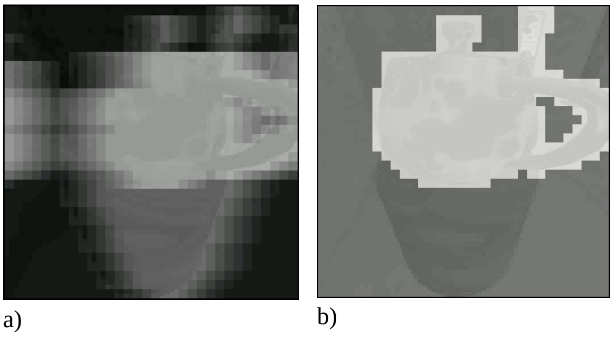


Figure 5. The result of encoding macroblock quantiser scale a) for standard MPEG-4 stream and b) for modified, 2-level encoding. Bright color indicates macroblocks encoded with lower qscale value and higher quality. MPEG-4 macroblock quantiser scale encoding does not allow for rapid change of qscale value, thus the resulting values do not fit well to the saliency map.

We found out that much better fit between quantiser scale and Saliency Map can be achieved if qscale can change only between two values. To realize this, we made two modifications in the standard MPEG-4 stream: firstly, a picture header contains two values denoting the high and low quality quantiser scales (instead of one *qscale* value). Secondly, there is no need for encoding value of *qscale* change. A single bit in the macroblock header determines whether *qscale* should switch between high and low quality value. Not only did such method of encoding give better fit between *qscale* values and the saliency map, but it also saved substantial number of bits. We measured that in case of standard MPEG-4 encoding 5.5% of total bit-stream size was used for the encoding of *qscale* changes between macroblocks. Whereas, our modification uses only 0.4% of the bit-stream size.

Our MPEG-4 encoder is based on an open source *FFmpeg* package[1]. The package offers optimized encoding and decoding for several video formats, including basic profile of MPEG-4. Our extensions to *FFmpeg* package included macroblock quantiser scale encoding – both MPEG-4 compliant and the new method, optimized for our framework and described in this section.

[1] FFmpeg package for video encoding, decoding and streaming can be found at http://ffmpeg.sourceforge.net/.

4. Results

To measure overall performance of the framework we compared quality of walk-through animation sequence encoded using standard MPEG-4 compression and our enhanced, attention guided encoder. The animation was recoded at several different bit-rates using both encoders. Then we made subjective test to compare quality of animations encoded using both methods. Five subjects were asked to decide which one of the two animations they had seen looked better. Each pair of animations was encoded using two different methods. Order in which the animations were shown was random. The subjects rated a pair of animations A and B using grades: (2) A is much better then B, (1) A is better than B, (0) A is the same as B, (-1) A is worse than B and (-2) A is much worse than B.

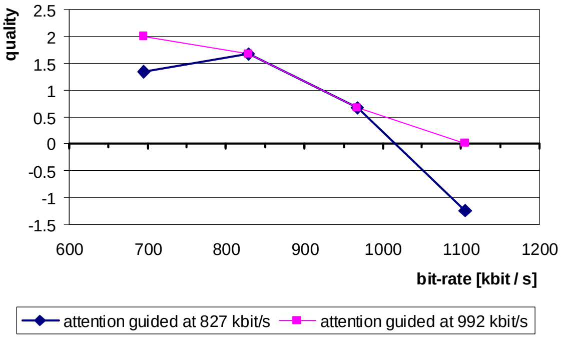


Figure 6. Comparison of an attention guided compression and a standard MPEG-4 stream. Four different bit-rates of MPEG-4 were placed along horizontal axis; two bit-rates of a stream encoded using the new method were distinguished with diamond and square marks. Higher “quality” values indicate that the animation encoded using the attention guided compression looked better in subjects’ opinion, than MPEG-4 animation encoded at particular bit-rate (x-axis).

The chart in Figure 6 show the results of our tests. Both lines correspond to subjective comparison of quality of the attention guided encoder versus the standard MPEG-4 encoder. For example square point at 700 kbit/s indicate that attention guided encoded animation at 992 kbit/s was judged to be of much better quality (2) than MPEG-4 encoded animation at 700 kbit/s.

The diamond point at 700 kbit/s could imply that quality of the MPEG-4 stream at 700 kbit/s was better than the same stream at 820 kbit/s, what should not happen in normal circumstances. This discrepancy came from the fact that the subjects could not often discern the difference between two animations. The differences were subtle so judgment was often difficult and erroneous.

We can conclude from the chart that the animation recorded using the attention guided encoder at 827 kbit/s was roughly of the same quality as MPEG-4 stream of 1020 kbit/s and to achieve the quality of MPEG-4 stream at 1100 kbit/s, the proposed attention guided encoder required 992 kbit/s. This gives reduction of bit-stream size between 10% and 19%. However, it is important to notice that the reduction of size greatly depends on accuracy of an attention model and content of an animation. The walk-through animation sequence used for the tests was adjusted for the maximum accuracy of the attention model.

5. Conclusion

In our research we built and examined the framework of interoperating subsystems of rendering, video compression and attention model for delivering a high-quality, low-bandwidth 3D animation at real-time rate. Our findings from the subjective tests are as follows:

- Compression efficiency improved when quality/bit-rate ratio was adjusted locally based on the output from the attention model.
- The attention model made a better prediction of the salient regions when supplemented with the 3D model of the scene.

Following two steps allowed us to efficiently implement our framework.

- Rendering to pixel buffers allowed improved transfer between GPU and CPU memory.
- Our simplified MPEG macro-block quantiser scale coding provided us with a significant gain in compression efficiency.

In our future work, we would like to compare performance of the framework when different attention models are used. We would like to make use of the predictions made by an attention model not only to improve compression, but also to improve the rendering subsystem, for example to adjust Level of Detail of the rendered objects. Finally, we would like to apply our framework to an end-to-end live streaming application.

Acknowledgements

This work was partially supported by Army-STRICOM.

References

BORDES, P. AND GUILLOTEL, P. 2000. Perceptually Adapted MPEG Video Encoding. *Human Vision and Electronic Imaging*, pages 168-175.

COHEN-OR, D., MANN Y. AND FLEISHMAN, S. 1999. Deep Compression for Streaming Texture Intensive Animations. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH 1999, 261-267.

ITTI, L. 2000. *Models of Bottom-Up and Top-Down Visual Attention*. PhD thesis, California Institute of Technology.

ITTI, L. Jan 2002. Real-Time High-Performance Attention Focusing in Outdoors Color Video Streams. In: *Proc. SPIE Human Vision and Electronic Imaging VII* (HVEI'02), San Jose, CA, (B. Rogowitz, T. N. Pappas Ed.), pages 235-243.

JARMASZ, J. December 2001. Towards the Integration of Perceptual Organization and Visual Attention: The Inferential Attentional Allocation Model.Carleton University. Cognitive Science Technical Report 2001-08. URL: http://www.carleton.ca/iis/TechReports.

HABER, J., MYSZKOWSKI, K., YAMAUCHI, H. AND SEIDEL, H-P. Perceptually Guided Corrective Splatting. 2001. In *EuroGraphics* (Manchester, UK, September 4-7 2001), pages 142-152.

LEVOY, M. 1995. Polygon-assisted JPEG and MPEG compression of synthetic images. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, SIGGRAPH 1995. 21–28.

MANN, Y. AND COHEN-OR, D. 1997. Selective Pixel Transmission for Navigating in Remote Virtual Environments. *EuroGraphics '97*, Volume 16, (1997), Number 3, 201-206.

OSBERGER, W., MAEDER, A.J. AND BERGMANN, N. January 1998. A Perceptually Based Quantization Technique for MPEG Encoding. *Proceedings SPIE 3299 - Human Vision and Electronic Imaging III*, San Jose, California, USA, 26-29.

TIPPER, S.P. AND WEAVER, B. 1998. The medium of attention: Location-based, object-based, or scene-based? In R. D. Wright (Ed.), Visual Attention (77-107). Oxford, NY: Oxford University Press.

TREISMAN, A. AND GELADE, G. 1980. *A* feature-integration theory of attention..*Cognitive Psychology*, 12, pages 97-136.

YANG, J., WU, L. AND WAIBEL, A. June 1996. Focus of Attention in Video Conferencing. Technical Report, CMU-CS-96-150, School of Computer Science, Carnegie Mellon University. URL: http://reports-archive.adm.cs.cmu.edu/cs1996.html.

YARBUS, A. L. 1967. Eye Movements and Vision. Plenum Press, New York.

YEE, H., PATTANAIK, S. N. AND GREENBERG, D. P. 2001. Spatio-Temporal Sensitivity and Visual Attention in Dynamic Environments. *ACM Transactions on Computer Graphics*, Vol. 20(1), pages 39-65.

YOON, I. AND NEUMANN, U. 2000. Web-Based Remote Rendering with IBRAC. *EuroGraphics 2000*. Volume 19 (2000), Number 3.

ZELLER, C. 2002. Balancing the Graphics Pipeline for Optimal Performance. GDC2002 Tutorial. URL: http://developer.nvidia.com/docs/IO/3564/ATT/PipelinePerformance.pdf