Realistic Rendering in Real-Time

Alan Chalmers and Kirsten Cater

Department of Computer Science
University of Bristol
Bristol, UK
Alan.Chalmers@bris.ac.uk
Cater@cs.bris.ac.uk

Abstract. The computer graphics industry, and in particular those involved with films, games and virtual reality, continue to demand more realistic computer generated images. Despite the ready availability of modern high performance graphics cards, the complexity of the scenes being modeled and the high fidelity required of the images means that rendering such images is still simply not possible in a reasonable, let alone real-time on a single computer. Two approaches may be considered in order to achieve such realism in real-time: Parallel Processing and Visual Perception. Parallel Processing has a number of computers working together to render a single image, which appears to offer almost unlimited performance, however, enabling many processors to work efficiently together is a significant challenge. Visual Perception, on the other hand, takes into account that it is the human who will ultimately be looking at the resultant images, and while the human eye is good, it is not perfect. Exploiting knowledge of the human visual system can save significant rendering time by simply not computing those parts of a scene that the human will fail to notice. A combination of these two approaches may indeed enable us to achieve realistic rendering in real-time.

Keywords: Parallel processing, task scheduling, demand driven, visual perception, inattentional blindness.

1 Introduction

A major goal in virtual reality environments is to achieve very realistic image synthesis at interactive rates. However, the computation time required is significant, currently precluding such realism in real time. The challenge is thus to achieve higher fidelity graphics for dynamic scenes without simultaneously increasing the computational time required to render the scenes. One approach to address this problem is to use parallel processing [2, 8, 11]. However, such parallel approaches have their own inherent difficulties, such as the efficient management of data across multiple processors and the issues of task scheduling to ensure load balancing, which still inhibits their wide-spread use for large complex environments [2].

The perception of a virtual environment depends on the user and the task that he/she is currently performing in that environment. Visual attention is the process by which we humans select a portion of the available visual information for localisation,

B. Monien and R. Feldmann (Eds.): Euro-Par 2002, LNCS 2400, pp. 21–28.

identification and understanding of objects in an environment. It allows our visual system to process visual input preferentially by shifting attention about an image, giving more attention to salient locations and less attention to unimportant regions. When attention is not focused onto items in a scene they can literally go unnoticed. *Inattentional blindness* is the failure of the human to see unattended items in a scene [4]. It is this inattentional blindness that we may exploit to help produce perceptually high-quality images in reasonable times.

2 Realistic Rendering

The concept of realistic image synthesis centers on generating scenes with an authentic visual appearance. The modeled scene should not only be physically correct but also perceptually equivalent to the real scene it portrays [7].

One of the most popular rendering techniques is ray tracing [4, 10, 14]. In this approach, one or more primary rays are traced, for each pixel of the image, into the scene. If a primary ray hits an object, the light intensity of that object is assigned to the corresponding pixel. Shadows, specular reflections and transparency can be simulated by spawning new rays from the intersection point of the ray and the object, as shown in figure 1. These shadow, reflection and transparency rays are treated in exactly the same way as primary rays, making ray tracing a recursive algorithm.

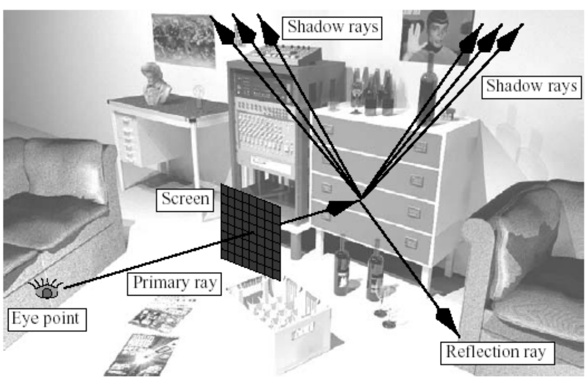

Fig. 1. The ray tracing algorithm, showing shadow and reflection rays, after Reinhard [2].

While most ray tracing algorithms approximate the diffuse lighting component with a constant ambient term, other more advanced systems, in particular the Radiance lighting simulation package [12, 13], accurately computes the diffuse inter-reflections by shooting a large number of undirected rays into the scene, distributed over a hemisphere placed over the intersection point of the ray with the object. Tracing these diffuse rays is also performed recursively. The recursive ray tracing process has to be carried out for each individual pixel separately. A typical image therefore takes at least a million primary rays and a significant multiple of that for shadow, reflection, transparency and diffuse rays. In addition, often more than one ray is traced per pixel (super-sampling) to help overcome aliasing artifacts.

Despite the enormous amount of computation that is required for ray tracing a single image, this rendering technique is actually well suited to parallel processing as the computation of one pixel is completely independent of any other pixel. Furthermore, as the scene data used during the computation is read, but not modified, there is no need for consistency checking and thus the scene data could be duplicated over every available processor. As such parallel ray tracing has often been referred to as an *embarrassingly parallel problem*.

However, in reality, the scenes we wish to model for our virtual environments are far too complex to enable the data to be duplicated at each processor. This is especially true if, rather than computing a single image of a scene, we wish to navigate through the entire environment. It should be noted, however, that if a shared memory machine is available, the scene does not have to be distributed over a number of processors, nor does data have to be duplicated. As such, parallel ray tracing on shared memory architectures is most certainly a viable approach and has led to implementations that may render complex scenery at interactive rates [8]. However, such shared memory architectures are not easily scalable and thus here we shall consider realistic rendering on the more scalable distributed memory parallel systems.

3 Parallel Processing

The goal of parallel processing remains the solution of a given complex problem more rapidly, or to enable the solution of a problem that would otherwise be impracticable by a single processor [1]. The efficient solution of a problem on a parallel system requires the computational ability of the processors to be fully utilized. Any processor that is not busy performing useful computation is degrading the overall system performance. Careful task scheduling is essential to ensure that all processors are kept busy while there is still work to be done.

The demand driven computational model of parallel processing has been shown to be very effective for parallel rendering [2, 9]. In the demand driven approach for parallel ray tracing, work is allocated to processors *dynamically* as they become idle, with processors no longer bound to any particular portion of pixels. Having produced the result for one pixel, the processors demand the next pixel to compute from some work supplier process.

This approach facilitates dynamic load balancing when there is no prior knowledge as to the complexity of the different parts of the problem domain. Optimum load balancing is still dependent on all the processors completing the last of the work at the same time. An unbalanced solution may still result if a processor is allocated a complex part of the domain towards the end of the solution. This processor may then still be busy well after all the other processors have completed computation on the remainder of the pixels and are now idle as there is no further work to do. To reduce the likelihood of this situation it is important that the computationally complex portions of the domain, the so called *hot spots*, are allocated to processors early on in the solution process. Although there is no *a priori* knowledge as to the exact computational effort associated with any pixel, nevertheless, any insight as to possible hot spot areas, such as knowledge of the computational effort for computing previous pixels, should be exploited. The order in which tasks are supplied to the processors can thus have a significant influence on the overall system performance.

4 Visual Perception

Advances in image synthesis techniques allow us to simulate the distribution of light energy in a scene with great precision. Unfortunately, this does not ensure that the displayed image will have a high fidelity visual appearance. Reasons for this include the limited dynamic range of displays, any residual shortcomings of the rendering process, and the restricted time for processing. Conversely, the human visual system has strong limitations, and ignoring these leads to an over specification of accuracy beyond what can be seen on a given display system [1]. The human eye is “good”, but not “that good”. By exploiting inherent properties of the human visual system we may be able to avoid significant computational expense *without* affecting the perceptual quality of the resultant image or animation.

4.1 Inattentional Blindness

In 1967, Yarbus [15] showed that the choice of task that the user is performing when looking at an image is important in helping us predict the eye-gaze pattern of the viewer. It is precisely this knowledge of the expected eye-gaze pattern that will allow us to reduce the rendered quality of objects outside the area of interest without affecting the viewer’s overall perception of the quality of the rendering.

In human vision, two general processes, called bottom-up and top-down, determine where humans locate their visual attention [4]. The bottom-up process is purely stimulus driven, for example a candle burning in a dark room; a red ball amongst a large number of blue balls; or the lips and eyes of a human face as they are the most mobile and expressive elements of the face. In all these cases, the visual stimulus captures attention automatically without volitional control. The top-down process, on the other hand, is directed by a voluntary control process that focusses attention on one or more objects, which are relevant to the observer’s goal when studying the scene. In this case, the attention normally drawn due to conspicuous aspects in a scene may be deliberately ignored by the visual system because of irrelevance to the goal at hand. This is “inattentional blindness” which we may exploit to significantly reduce the computational effort required to render the virtual environment.

4.2 Experiment

The effectiveness of inattentional blindness in reducing overall computational complexity was illustrated by asking a group of users were asked to perform a specific task: to watch two animations and in each of the animations, count the number of pencils that appeared in a mug on a table in a room as he/she moved on a fixed path through four such rooms. In order to count the pencils, the users needed to perform a smooth pursuit eye movement tracking the mug in one room until they have successfully counted the number of pencils in that mug and then perform an eye saccade to the mug in the next room. The task was further complicated and thus retain the viewer’s attention, by each mug also containing a number of spurious paintbrushes. The study involved three rendered animations of an identical fly through of four rooms. The only difference being the quality to which the individual animations had been rendered. The three qualities of animation were:

- High Quality(HQ): Entire animation rendered at the highest quality.
- Low Quality(LQ): Entire animation rendered at a low quality with no anti-aliasing.
- Circle Quality(CQ): Low Quality Picture with high quality rendering in the visual angle of the fovea (2 degrees) centered around the pencils, shown by the inner green circle in figure 2. The high quality is blended to the low quality at 4.1 degrees visual angle (the outer red circle in figure 2) [6].

Fig. 2: Visual angle covered by the fovea for mugs in the first two rooms at 2 degrees (smaller circles) and 4.1 degrees (large circles).

Each frame for the high quality animation took on average 18 minutes 53 seconds to render on a Intel Pentium 4 1GHz Processor, while the frames for the low quality animation were each rendered on average in only 3 minute 21 seconds.

A total of 160 subjects were studied which each subject seeing two animations of 30 seconds each displayed at 15 frames per second. Fifty percent of the subjects were asked to count the pencils in the mug while the remaining 50% were simply asked to watch the animations. To minimise experimental bias the choice of condition to be run was randomised and for each, 8 were run in the morning and 8 in the afternoon. Subjects had a variety of experience with computer graphics and all exhibited at least average corrected vision in testing. A count down was shown to prepare the viewers that the animation was about to start followed immediately by a black image with a white mug giving the location of the first mug. This ensured that the viewers focused their attention immediately on the first mug and thus did not have to look around the scene to find it.

On completion of the experiment, each participant was asked to fill in a detailed questionnaire. This questionnaire asked for some personal details, including age, occupation, sex and level of computer graphics knowledge. The participants were then asked detailed questions about the objects in the rooms, their colour, location and quality of rendering. These objects were selected so that questions were asked about objects both near the foveal visual angle (located about the mug with pencils) and in the periphery. They were specifically asked not to guess, but rather state “don’t remember” when they had failed to notice some details.

4.3 Results

Figure 3 shows the overall results of the experiment. Obviously the participants did not notice any difference in the rendering quality between the two HQ animations (they were the same). Of interest is the fact that, in the CQ + HQ experiment, 95% of the viewers performing the task consistently failed to notice any difference between the high quality rendered animation and the low quality animations where the area around the mug was rendered to a high quality. Surprisingly 25% of the viewers in the HQ+LQ condition and 18% in the LQ+HQ case were so engaged in the task that they completely failed to notice any difference in the quality between these very different qualities of animation.

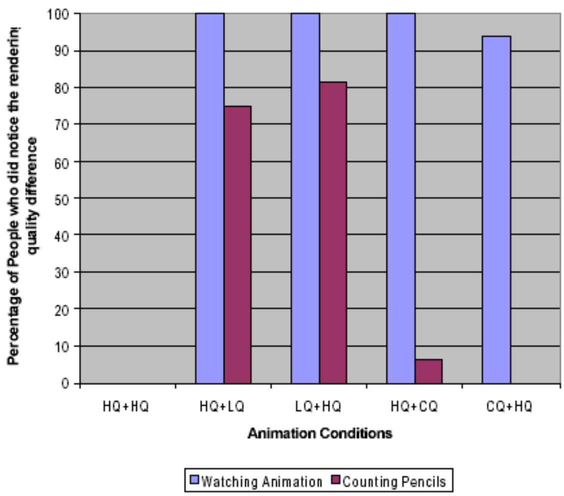

Fig. 3. Experimental results for the two tasks: Counting the pencils and simply watching the animations.

Furthermore, having performed the task of counting the pencils, the vast majority of participants were simply unable to recall the correct colour of the mug (90%) which was in the foveal angle and even less the correct colour of the carpet (95%) which was outside this angle. The inattentional blindness was even higher for “less obvious” objects, especially those outside the foveal angle. Overall the participants who simply watched the animations were able to recall far more detail of the scenes, although the generic nature of the task given to them precluded a number from recalling such details as the colour of specfic objects, for example 47.5% could not recall the correct colour of the mug and 53.8% the correct colour of the carpet.

5 Conclusions

The results presented demonstrate that inattentional blindness may in fact be exploited to significantly reduce the rendered quality of a large portion of a scene without having any affect on the viewer’s perception of the scene. This knowledge will enable us to prioritize the order, and the quality level of the tasks that are assigned to the processors in our parallel system. Those few pixels in the visual angle of the fovea (2 degrees) centered around the pencils, shown by the green inner circle in figure 2 should be rendered first and to a high quality, the quality can then be blended to the low quality at 4.1 degrees visual angle (the red outer circle in figure 2).

Perhaps we were too cautious in our study of inattentional blindness. Future work will consider whether in fact we even need to ray trace some of the pixels outside the foveal angle. It could be that the user's focus on the task is such that he/she may fail to notice the colour of many of the pixels outside this angle and that these could simply be assigned an arbitrary neutral colour, or interpolated from a few computed sample pixels.

Visual perception, and in particular inattentional blindness does depend on knowledge of the task being performed. For many applications, for example games and simulators, such knowledge exists offering the real potential of combining parallel processing and visual perception approaches to achieve "perceptually realistic" rendering in real-time.

References

1. Cater K., Chalmers AG. and Dalton C. 2001 Change blindess with varying rendering fidelity: looking but not seeing, Sketch SIGGRAPH 2001, Conference Abstracts and Applications.
2. Chalmers A., Davis T. and Reinhard E. Practical Parallel Rendering, AKPeters, to appear 2002.
3. Glassner A.S. , editor. *An Introduction to Ray Tracing*. Academic Press, San Diego, 1989.
4. James W. 1890 Principles of Psychology, New York: Holt.
5. Mack A. and Rock I. 1998 Inattentional Blindness, Massachusetts Institute of Technology Press.
6. McConkie GW. and Loschky LC. 1997 Human Performance with a Gaze-Linked Multi-Resolutional Display". ARL Federated Laboratory Advanced Displays and Interactive Displays Consortium, Advanced Displays and Interactive Displays First Annual Symposium, 25-34.
7. McNamara, A., Chalmers, A., Troscianko, T. and Reinhard, E., "Fidelity of Graphics Reconstructions: A Psychophysical Investigation". *Proceedings of the 9th Eurographics Workshop on Rendering* (June 1998) Springer Verlag, pp. 237 - 246.
8. Parker S., Martin W., Sloan P.-P., Shirley P., Smits B., and Hansen C. Interactive ray tracing. In *Symposium on Interactive 3D Computer Graphics*, April 1999.
9. Reinhard E., Chalmers A., and Jansen FW. Overview of parallel photo-realistic graphics. In *Eurographics STAR – State of the Art Report*, pages 1–25, August-September 1998.
10. Shirley P. *Realistic Ray Tracing*. A K Peters, Natick, Massachusetts, 2000.
11. I. Wald, P. Slusallek, C. Benthin, and M. Wagner. Interactive rendering with coherent ray tracing. *Computer Graphics Forum*, 20(3):153–164, 2001.
12. Ward GJ, Rubinstein FM., and Clear RD. A ray tracing solution for diffuse interreflection. *ACM Computer Graphics*, 22(4):85–92, August 1988.

13. Ward Larson GJ. and Shakespeare RA. *Rendering with Radiance*. Morgan Kaufmann Publishers, 1998.
14. Whitted T. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.
15. Yarbus AL.1967 Eye movements during perception of complex objects. In L. A. Riggs, Ed., Eye Movements and Vision, Plenum Press, New York, chapter VII, pp. 171-196.