Eurographics Symposium on Parallel Graphics and Visualization (2006)
Alan Heirich, Bruno Raffin, and Luis Paulo dos Santos (Editors)

Rendering on Demand

A. Chalmers[†], K. Debattista, V. Sundstedt, P. Longhurst and R. Gillibrand

Computer Science Department, University of Bristol, United Kingdom

Abstract

In order for computer graphics to accurately represent real world environments, it is essential that physically based illumination models are used. However, typical global illumination solutions may take many seconds, even minutes to render a single frame. This precludes their use in any interactive system. In this paper we present Rendering on Demand, a selective physically-based parallel rendering system which enables high-fidelity virtual computer graphics imagery to be rendered at close to interactive rates. By exploiting knowledge of the human visual system we substantially reduce computation costs by rendering only the areas of perceptual importance in high quality. The rest of the scene is rendered at a significantly lower quality without the viewer being aware of the quality difference. This is validated through psychophysical experimentation.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, I.3.2 [Computer Graphics]: Graphics Systems

1. Introduction

The computer graphics industry, and in particular those involved with films, games and virtual reality, continue to demand more realistic computer generated images. In addition, high fidelity imagery is playing an increasingly key role in many applications, for example archaeology, architecture and simulations, see Figure 1, in order to accurately represent real environments on a computer. In such applications, failure to produce images which accurately match reality may in fact lead to, for example, the archaeologists misinterpreting the past [DC01], or the incorrect display of a military vehicle attempting to camouflage in a certain simulated terrain may result in detection of the vehicle in the real battlefield scenario. Only physically-based rendering is capable of achieving the level *quantifiable* realism required by such applications which need to accurately portray reality [MCTG00, UWP05].

Despite the ready availability of modern high performance graphics cards, the complexity of the scenes being modelled 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. Previously parallel processing has been used extensively to try and significantly reduce the overall computational time when rendering high quality images. However, while parallel processing appears to offer almost unlimited performance, enabling many processors to work efficiently together remains a significant challenge. More recently, visual perception techniques have been investigated. Such approaches consider 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.

This paper describes the novel Rendering on Demand system which combines visual perception and parallel processing to selectively guide rendering computations in order to achieve perceptually high fidelity computer graphics in real time.

The rest of the paper is organised as follows. Section 2 presents related work, Section 3 presents our rendering framework. Section 4 discusses our implementation in *Radiance* [LS98]. Section 5 presents performance results and psychophysical validation of our work. Finally we conclude and present future work in Section 6.

[†] alan.chalmers@bris.ac.uk

Figure 1: *High-fidelity rendering: (a) the Temple of Kalabsha [SCM04], (b) and (c) the corridor scene [SDL*05] and (d) the art gallery [LS98].*

2. Related Work

2.1. Visual Attention

There are two general attentional processes, which determine where humans focus their visual attention [Jam90]. These are labelled *bottom-up*, which is an automatic visual stimulus and *top-down*, which is voluntary and focuses on the observer's goal. The *bottom-up* process has been found to be influenced by contrast, size, shape, colour, brightness, orientation, edges and motion. The idea of a *saliency map*, a two-dimensional map encoding the areas that automatically attract our attention in the environment, was introduced in [KU85]. This was later developed into a computer model by [IKN98]. Subsequent work by [MD02] showed, however, that such bottom-up visual attention models do not always predict attention regions in a reliable manner. Perceptual metrics such as Daly's Visible Differences Predictor (VDP) [Dal93] were developed as image-space algorithms that predict the per-pixel probability that observers will detect a difference between two images rendered using different approximations.

The *top-down* approach, highlighted by the visual psychologist [Yar67], demonstrated a strong correlation between a viewer's eye movements to the task at hand. Cater *et al.* showed that conspicuous objects in a scene that would normally attract the viewer's attention in an animation are ignored if they are not relevant to the task at hand [CCL02]. This failure of the human to see unattended items in a scene, is known as *inattentional blindness* [MR98, SC99]. Cater *et al.* also confirmed by using eye-tracking that the effect was indeed caused by inattentional blindness and not peripheral vision. In [CCW03] the concept of *task maps*, which are two-dimensional maps highlighting the task to be performed, was introduced to exploit the *top-down* approach. In [HKG02, PS03] both task maps and bottom-up visual attention were used to guide virtual humans in a complex environment by making some perceptual objects more salient than others and letting the virtual agents pay attention to their surroundings. Recent work has also focused on modeling the influence of task on attention in [NI05].

2.2. Selective Rendering

We use the term selective rendering to refer to rendering systems that allow a choice of flexible computation within the rendering framework. Techniques that we consider as selective rendering do not necessarily need to exploit visual attention. Examples are level of detail techniques [Cla76, LWC*02] and progressive and adaptive algorithms [CCWG88, PS89, CRMT91]. Although techniques based on visual attention had been developed before, such as Mitchell's [Mit87] adaptive antialiasing sampling for ray tracing, they have become more popular recently. Prikryl and Purgathofer [PP99] provide a good overview of perceptually-driven rendering radiosity algorithms. Myszkowski [Mys98] and Bolin and Meyer [BM98] use visual difference predictors, both to direct the next set of samples within a stochastic ray tracing framework, and as a stopping condition. The main drawback of these approaches is the expense of computing the visual difference predictors many times within the calculation of a single image. Ramasubramanian *et al.* [RPG99] decouples the spatially-dependent saliency component from the luminance-dependent component. Precomputing the former, improved performance substantially for their path tracer. Yee *et al.* [YPG01] exploited a saliency model termed the *Aleph Map* to adjust the search radius accuracy of the interpolation of irradiance cache values. Haber *et al.* [HMYS01] in their real-time renderer used saliency maps and the notion of task objects to identify the most salient objects which were rendered with glossy and specular components. Cater *et al.* [CCW03] selectively rendered the foveal angle around the task related objects of an image at higher quality. In [SDL*05] both task maps and saliency maps are used to vary the number of rays shot per pixel and the specular threshold in a global illumination environment to produce high perceptual quality images at a reduced computational cost.

3. Rendering on Demand Framework

Our rendering framework, which we term Rendering on Demand (RoD), is designed for parallel rendering physically-based high-fidelity still images, animations, and interactive systems for complex scenes in reasonable time [DSPC05]. The overall RoD framework is shown in figure 2.

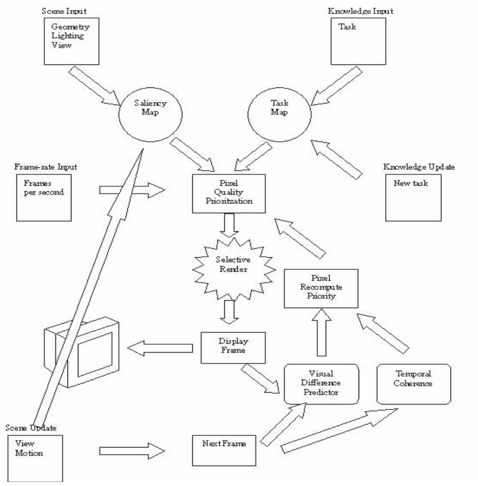

Figure 2: *Overall RoD framework.*

The core of the RoD framework, the parallel selective render, can be represented in two main processing blocks entitled *selective guidance* and *selective rendering*, see Figure 5:

Selective Guidance is responsible for the directives which will drive the selective renderer. Bottom-up visual attention and top-down visual attention models and other metrics such as motion and predicted complexity are handled within this process.

Selective Rendering within our framework is designed to take advantage of parallelism at a number of levels. The selective renderer adjusts parameters, typically per pixel (such as primary rays shot), depending on the selective guidance. At the highest level, the renderer functions over a distributed system. Subsequently, a per-node configuration determines the hardware mappings for the most efficient use of the hardware at that node (GPU, CPUs *etc.*...).

3.1. Importance Map

Our selective guidance system is composed of a map which directs the selective rendering which we term the *importance map* [SDL*05]. The importance map, IM(ω_t,ω_s,op), is a two-dimensional map representing image space. The values within the map dictate where computational resources are best spent to obtain the highest perceptual result, whereby the highest values will result in preferential rendering. The importance map is a combination of a number of maps. We

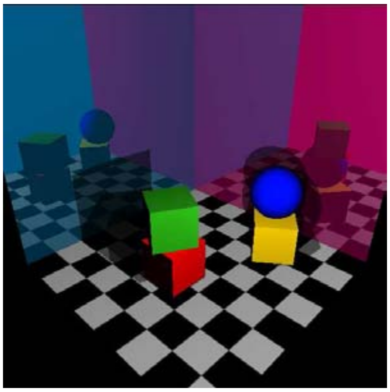

Figure 3: *Snapshot with texture mapping, shadowing and reflections.*

illustrate the system using a *task map* to model the effect of top-down visual attention and a *saliency map* to model bottom-up visual attention, see Figure 4. Further maps such as direct cost, discontinuities [Guo98], sound [MDCT05] and motion could easily be handled by our importance map. The parameters in the importance map allow the user to specify a weighting that defines the relative importance of the task map and the saliency map. The parameter ω_t is a coefficient which is applied to the values in the task map. The other coefficient ω_s is applied to the values in the saliency map, and the two terms are combined through the operator op.

Selection of an appropriate operator controls the combination of the two maps in the selective renderer. Our implementation currently uses addition to combine the information from both maps such that all weighted features are preserved. Equal weighting would be IM(0.5,0.5,+) which correspond to our IQ selective rendered animations, see Section 5.1.

3.2. Cost Prediction

The time required to render a complex scene is significantly affected by a number of factors, including image size, geometry, material properties and the quality of the required rendering. The RoD system incorporates a rapid cost prediction scheme which enables the overall computational cost of the high-fidelity global illumination solution to be accurately predicted, in advance of actually rendering the individual frames of an animation. Not only can this cost prediction be used to provide a client with a quote for rendering, prior to any rendering being carried out, but also determining the approximate cost distribution for each voxel of the scene can then be used to calculate an initial data distribution, such that the workload per processor can be better load balanced.

The RoD cost prediction scheme uses detailed knowledge of the scene itself to determine the most appropriate

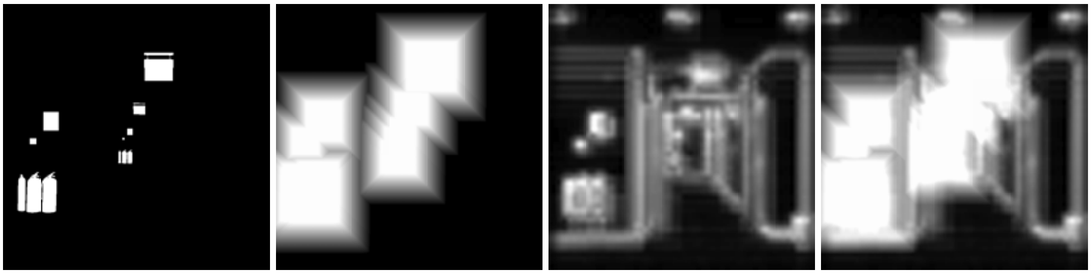

Figure 4: *Generating the importance map from Figure 1 (c): (a) the task objects, (b) the task map with foveal angle, (c) saliency map and (d) the importance map.*

way in which to trace a low number of profiling rays in order to gain some idea of overall rendering costs for any frame [GLDC06].

4. Implementation

While our framework is applicable to any form of high-fidelity rendering, we present an implementation of the framework based on the physically-based lighting simulation system *Radiance* [LS98]. Our implementation is an extension of the *Radiance* renderer, `rpict`. An overview of the steps performed by our implementation is illustrated in Figure 5. We have adapted *Radiance* to our rendering framework, by introducing a selective guidance system based on a rapid image estimate using rasterisation we term *Snapshot* [LDC05]. Furthermore, we have extended *Radiance* to support selective rendering and parallelism. The image filtering tools already existing within *Radiance* were sufficient to satisfy our filtering, tone mapping and interpolation needs.

4.1. Selective Guidance

Our first step corresponds to the selective guidance process. In order to produce the importance map we use a rapid image estimate of the scene description using rasterisation, thus exploiting fast graphics hardware. *Snapshot* produces several different intermediate images. Primarily a rapid rasterisation of the scene is rendered which includes basic lighting, reflection effects and texture mapping, see Figure 3. This result is then further processed, either within programable portions of the GPU, or on the CPU. Shadows and reflections are a key component of any high-fidelity image. To predict the shadow boundaries, we generated shadows in our scenes by projecting shadow maps from the front and back of each light source [Wil78]. Mirror reflections are generated by redrawing the scene's geometry from a reflected camera position onto the mirror plane. Figure 3 illustrates shadows and reflections generated from our system. It is also possible to predict more general reflections and also refractions within *Snapshot* using a technique known as cubic environment mapping [VF94]. To account partially for indirect reflected light, sources are added to the scene as additional light sources with reduced emission. The reflected position of each light source given each planar mirrored surface is calculated. These extra lights are then positioned with their emission components reduced by the reflectivity of the mirror. Furthermore, *Snapshot* implements the same isotropic shaders found in *Radiance* and subsequently in our selective renderer therefore producing results more similar to the final image.

Snapshot is also capable of producing edge maps [LDGC05] and saliency maps [LDC06] from the rapid image estimate. The saliency map generated from *Snapshot* uses object space knowledge to identify conspicuity for a number of features: motion similar to the work of [YPG01], depth and habituation, a novel feature which accounts for the amount of time that an object has been on screen in an animation. Image space saliency, calculated in a similar fashion to the Itti *at al.* saliency map [IKN98], is composed of three channels: colour, intensity and orientation. Figure 6 describes *Snapshot*'s saliency model. All of these features are calculated on the GPU for maximum performance. For further implementation details see [LDC06]. Finally the *Snapshot* is used to locate task related object, tagged at a modelling stage, and thus rapidly allows the importance map to be generated. Figure **??** shows some scenes rendered using the *Snapshot*, the generated importance map (which is only a saliency map since there are no task related objects in these scenes) and the final rendering.

4.2. Selective Rendering

The phase corresponding to the selective rendering process within our framework is based on the distributed ray tracing as used by *Radiance*. Ray tracing is traditionally easily extended into a parallel framework, however our approach follows the *Radiance* implementation. Although *Radiance* uses distributed ray tracing to render images, the irradiance cache [WRC88] is used to accelerate the calculation of the indirect diffuse component. As the irradiance cache is a shared data structure, it is non-trivial to parallelise. We use an approach similar to [KMG99].

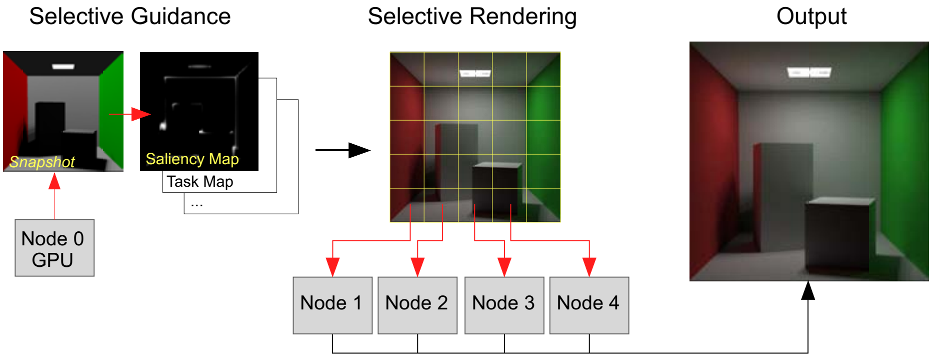

Figure 5: *Implementation framework.*

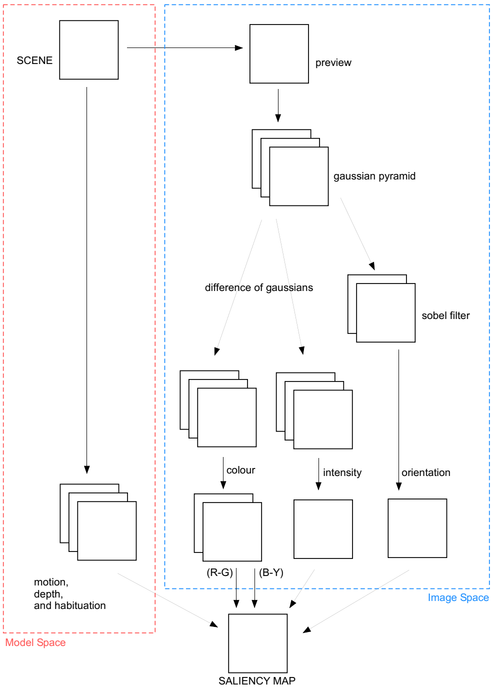

Figure 6: *Saliency model.*

Figure 7: *Test scenes. Scenes represented by image preview (top), importance map consisting of only saliency map (middle) and selectively rendered image (bottom).*

In our implementation we parallelise the rendering using a demand-driven approach, in the form of a master-slave model using the message passing system MPI. The importance map is used by the master to subdivide the workload and by the slaves to decide how many rays to cast per pixel in our virtual environments. The master is responsible for subdividing the image plane into a number of tiles of a given granularity. Each image tile represents a job for a slave to compute. We use the importance map as a simple cost prediction map. Since, at the slave, the importance map dictates the number of rays shot per pixel, the master uses it to improve subdivision by ensuring that each tile contains an equal number of primary rays to be shot. This improves load balancing by ensuring a more even distribution of the workload.

The master farms out the work to all the slaves in the form of the coordinates of the tile to be rendered. The slaves then render the image tiles assigned to them using the importance map to direct the rendering of each pixel. A user-defined parameter indicates the maximum number of rays per pixel and other adaptable parameters. This value is then modulated by the value in the importance map to calculate how many rays are shot for a given pixel. For example, the higher the value within the importance map, the more rays per pixel are shot. When the slave finishes executing the job, it asks for more work from the master until the entire process is completed.

The parallel irradiance cache relies on each of the slaves computing irradiance cache values and storing them in an outgoing buffer. Whenever the buffer reaches a user-defined threshold it is broadcast to all other slaves. In this approach each slave is allowed to broadcast to every other slave. In order to maximise computation, each slave has a separate communicator process which listens for incoming irradiance cache samples. Whenever a set of samples is received, the communicator process stores the data in a shared memory area, where the computation process can collect it and insert it onto the irradiance cache. This approach removes the contention on the centralised node but can still run into scalability problems.

5. Results

In this section we demonstrate the effectiveness of the framework. Primarily we validate our importance map framework by means of a task-based experiment within a virtual environment while showing uniprocessor speedup results. Secondly, we show results from our GPU implementation of the saliency map. Finally, we present results for visual attention combined with our parallel rendering approach.

5.1. Validating Selective Rendering using the Importance Map

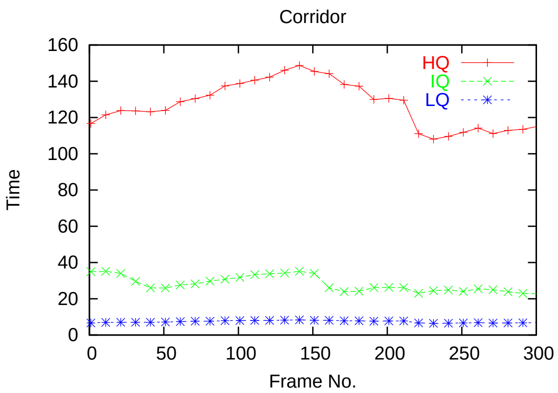

Figure 8: *Corridor scene animation results. HQ for high quality. IQ using our framework and LQ using low quality.*

We ran an experiment that demonstrates the potential of selective rendering using visual attention [SDL*05]. Two studies were conducted and with 16 participants in each group, varying in age from 18-39. The scene used in our experiment is an office corridor which contains several items related to fire safety, see Figure 1 (b) and (c). Before beginning the experiment, the subjects read a sheet of instructions on the procedure of the particular task they were to perform. Both groups were asked to play the role of a fire security officer with the task of counting the total number of fire safety items. A pre-study was run to confirm that the observer would have enough time to perform the task. Each of the participants was also shown an example of what kind of fire safety items the scene could contain.

Two different walk-throughs of the corridor scene were used in the experiment. To avoid familiarity effects that might influence the scan path of the observers, both the animations were rendered with different views and the location of the objects changed within each scene. Each animation contained the same number of task-related objects (15) but not the same number of non-task related objects. The same camera path was used for both animations. The order in which the subjects saw their two stimuli was also altered to avoid bias.

Half the subjects were shown a high quality (HQ) animation and a selectively rendered animation for the importance map quality (IQ). The HQ version was rendered with full global illumination. A HQ stimuli was always shown in order to ascertain if the participants could distinguish between this and a lower quality stimuli. The IQ animation were selectively rendered using a specular threshold value between 0.01 for HQ areas and 1 for low quality (LQ) areas. The maximum number of rays per pixel was set to 16. In LQ areas only 1 ray per pixel was cast. Rendering the entire frame to the same detail as the high quality areas takes on average 4.5 times longer than our optimized method using the importance map. Results for the entire animation are shown in Figure 8. The other half of the subjects were shown a HQ and a LQ image. The LQ version was rendered using the lowest values of 1 ray per pixel and a specular threshold of 1. Each animation was 17 seconds long, including a countdown before the animation started. Having watched both animations, the participants were asked which of the two they thought had the worst rendering quality (two-alternative forced-choice).

For each pair, a result of 50% correct selection is the unbiased ideal. This is the statistically expected result when no differences between the higher and lower quality stimuli were perceived. We compared all stimuli pairings to an expected 50/50 data to ascertain whether the viewers perceived the difference in quality ($df = 1$, critical value 3.84 at 0.05 level of significance). The statistical analysis of the results confirmed that for HQ/IQ the difference in proportions is not significant ($\chi^2 = 0.25$, $p > 0.05$). From this we can conclude that there is no relationship across the categories and the two animations were perceived as the same quality. In the HQ/LQ condition there is a significant difference ($\chi^2 = 4$, $p < 0.05$). From this we can conclude that people did notice a difference between the HQ and LQ animations.

5.2. Importance map generation

We demonstrate results for the *Snaphsot* image preview and saliency map generation for the scenes in Figure 7. Table 1 shows the results including scene complexity. Scene complexity (geometry and light sources) obviously has an impact on the results *Snapshot* due to the rasterisation nature. These results could be improved using level of detail techniques and object culling. The timings for the saliency map generation or more or less constant at 0.03 of a second which means that they can be used in real time. Figure 9 demonstrates the advantages of generating the saliency map on the GPU as opposed to the CPU at varying resolutions.

The graph shows that at a resolution of 512×512 our approach is approximately two orders of magnitude faster than the GPU-based approach. a situation which will be further accentuated in the future since graphics hardware has had a trend of advancing faster than other components of a computer [OLG*05].

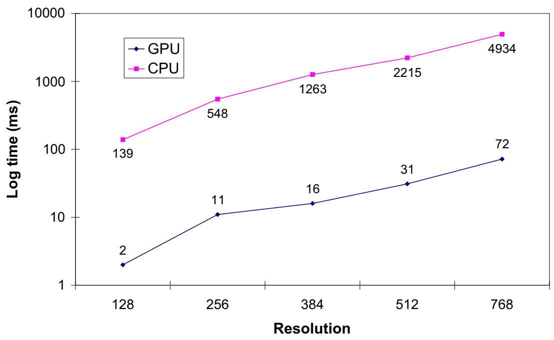

Figure 9: *Linear image filtering: Nvidia 6600GT GPU vs. P4 3.4Ghz CPU.*

5.3. Selective Rendering in Parallel

We demonstrate the combination of selective guidance and parallel selective rendering by showing timings of rendering a 90 frame animation in the Kalabsha scene with four different settings [DSPC05], see Figure 1 (a). All frames were rendered at a resolution of 500×500 with a maximum of five rays per pixel. The first rendered sequence used the plain uniprocessor version, representing the traditional rendering method within *Radiance*. In this sequence all the pixels in each frame were rendered with five rays per pixel. The second sequence exploited visual attention by rendering only the salient parts of the scene at a higher quality, based on a saliency map of that scene. In this case the importance map was based only on the saliency map. The importance map dictated how many rays per pixel were shot up to a maximum of five. The third rendered sequence used the parallel version on sixteen processors to render the sequence without visual attention. The final sequence used both parallelism on sixteen processors and the saliency maps for visual attention. We used the distributed version of the parallel irradiance cache for both of the sequences that were rendered in parallel.

Results, presented in Figure 10, clearly demonstrate the effectiveness of our approach. For all the rendered sequences, the first few frames of the animations were initially quite expensive. This was due to the irradiance cache being empty. In subsequent frames as the irradiance cache became more populated the timings became more homogenous. The uniprocessor (uni SM) saliency version gained a 3 time speedup over the standard uniprocessor version (uni). The parallel (16) version was around 13 times faster than the traditional version. The combined saliency and parallelism (16 SM) approach was around 37 times faster than the traditional rendering.

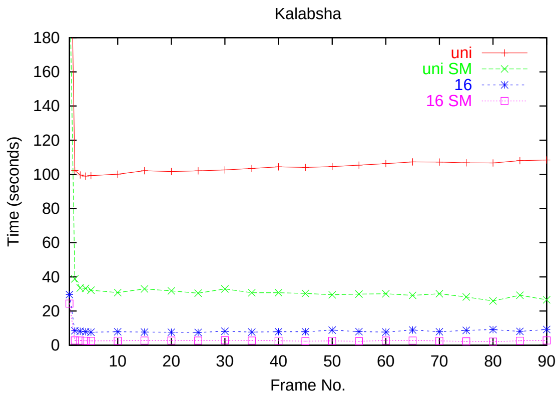

Figure 10: *The Kalabsha scene exploiting bottom-up visual attention in parallel.*

6. Conclusion and Future Work

Rendering on Demand is a framework for producing perceptually realistic images in real time, the core of which is a parallel selective renderer. In this paper we have shown that it is possible to significantly reduce the computational cost for high-fidelity rendering of virtual environments by using selective rendering. By knowing the user's focus of attention in the virtual environment we are able to direct our computational resources to render those areas, which are a small fraction of the total environment, at very high quality. The rest of the environment can be rendered at a much lower quality and, as the psychophysical experiments have shown, the users are simply unable to notice the difference between a selectively rendered solution and a high quality image. As we have also shown, parallel processing can be used to reduce the time to render the virtual environment even further. We are thus able to achieve perceptually high-fidelity virtual reality at close to interactive rates.

	Corridor	Cornell Box	Kitchen
Number of triangles	150,000	40	757,000
Lights	32	5	1
Preview image time (ms)	4,500	31	210
Saliency map generation time (ms)	36	31	36
Total time (ms)	4,536	62	246

Table 1: *Timing results for test scenes.*

We are, however, still some way from achieving true interactivity for complex virtual environments, especially those which involve participating media [SGGC05]. Future work will consider including audio and motion in the virtual environment to reduce even further the amount of high-fidelity rendering required as the input of additional sensory information lowers the amount of cognitive power that the brain has to process the visual information. Furthermore, we will exploit the temporal and spatial coherence between frames of an animation, to minimise the number and quality of the pixels we actually have to render.

References

[BM98] BOLIN M. R., MEYER G. W.: A perceptually based adaptive sampling algorithm. In *SIGGRAPH '98* (1998), ACM Press, pp. 299–309.

[CCL02] CATER K., CHALMERS A., LEDDA P.: Selective quality rendering by exploiting human inattentional blindness: looking but not seeing. In *Proceedings of the ACM symposium on Virtual reality software and technology* (2002), ACM Press, pp. 17–24.

[CCW03] CATER K., CHALMERS A., WARD G.: Detail to Attention: Exploiting Visual Tasks for Selective Rendering. In *Proceedings of the Eurographics Symposium on Rendering* (2003), pp. 270–280.

[CCWG88] COHEN M. F., CHEN S. E., WALLACE J. R., GREENBERG D. P.: A progressive refinement approach to fast radiosity image generation. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques* (1988), ACM Press, pp. 75–84.

[Cla76] CLARK J. H.: Hierarchical geometric models for visible surface algorithms. *Commun. ACM 19*, 10 (1976), 547–554.

[CRMT91] CHEN S. E., RUSHMEIER H. E., MILLER G., TURNER D.: A progressive multi-pass method for global illumination. In *SIGGRAPH '91* (1991), ACM Press, pp. 165–174.

[Dal93] DALY S.: The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In *Digital Images and Human Vision* (1993), A.B. Watson, MIT Press, Cambridge, MA, pp. 179–206.

[DC01] DEVLIN K., CHALMERS A.: Realistic visualisation of the pompeii frescoes. In *AFRIGRAPH 2001* (November 2001), Chalmers A., Lalioti V., (Eds.), ACM SIGGRAPH, pp. 43–47.

[DSPC05] DEBATTISTA K., SUNDSTEDT V., PEREIRA F., CHALMERS A.: Selective parallel rendering for high-fidelity graphics. In *Theory and Practice of Computer Graphics* (June 2005).

[GLDC06] GILLIBRAND R., LONGHURST P., DEBATTISTA K., CHALMERS A.: Cost prediction for global illumination using a fast rasterised scene previewing. In *Proceedings of AFRIGRAPH 2006, ACM SIGGRAPH* (2006), pp. 41–48.

[Guo98] GUO B.: Progressive radiance evaluation using directional coherence maps. In *SIGGRAPH '98* (New York, NY, USA, 1998), ACM Press, pp. 255–266.

[HKG02] HILL, JR. R. W., KIM Y., GRATCH J.: Anticipating where to look: predicting the movements of mobile agents in complex terrain. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems* (2002), ACM Press, pp. 821–827.

[HMYS01] HABER J., MYSZKOWSKI K., YAMAUCHI H., SEIDEL H.-P.: Perceptually guided corrective splatting. In *Computer Graphics Forum* (2001), vol. 20, pp. 142–152.

[IKN98] ITTI L., KOCH C., NIEBUR E.: A model of Saliency-Based Visual Attention for Rapid Scene Analysis. In *Pattern Analysis and Machine Intelligence* (1998), vol. 20, pp. 1254–1259.

[Jam90] JAMES W.: A saliency-based search mechanism for overt and covert shifts of visual attention. In *Principles of Psychology* (1890).

[KMG99] KOHOLKA R., MAYER H., GOLLER A.: Mpi-parallelized radiance on sgi cow and smp. In *ParNum '99: Proceedings of the 4th International ACPC Conference Including Special Tracks on Parallel Numerics and Parallel Computing in Image Processing, Video Processing, and Multimedia* (1999), Springer-Verlag, pp. 549–558.

[KU85] KOCH C., ULLMAN S.: Shifts in selective visual attention: towards the underlying neural circuitry. In *Human Neurobiology* (1985), vol. 4, pp. 219–227.

[LDC05] LONGHURST P., DEBATTISTA K., CHALMERS A.: Snapshot: A rapid technique for driving a selective global illumination renderer. In *WSCG 2005 SHORT papers proceedings* (2005).

[LDC06] LONGHURST P., DEBATTISTA K., CHALMERS A.: A gpu based saliency map for high-fidelity selective rendering. In *Proceedings of AFRIGRAPH 2006, ACM SIGGRAPH* (2006), pp. 21–29.

[LDGC05] LONGHURST P., DEBATTISTA K., GILLIBRAND R., CHALMERS A.: Analytic antialiasing for selective high fidelity rendering. In *Proceedings of SIBGRAPI, IEE Computer Society Press* (2005), pp. 359–366.

[LS98] LARSON G. W., SHAKESPEARE R.: *Rendering with radiance: the art and science of lighting visualization*. Morgan Kaufmann Publishers Inc., 1998.

[LWC*02] LUEBKE D., WATSON B., COHEN J. D., REDDY M., VARSHNEY A.: *Level of Detail for 3D Graphics*. Elsevier Science Inc., 2002.

[MCTG00] MCNAMARA A., CHALMERS A., TROSCIANKO T., GILCHRIST I.: Comparing Real and Synthetic Scenes using Human Judgements of Lightness. In *Proceedings of the Eurographics Workshop in Brno* (2000).

[MD02] MARMITT G., DUCHOWSKI A.: Modeling Visual Attention in VR: Measuring the Accuracy of Predicted Scanpaths. In *Eurographics 2002, Short Presentations* (2002), pp. 217–226.

[MDCT05] MASTOROPOULOU G., DEBATTISTA K., CHALMERS A., TROSCIANKO T.: Auditory bias of visual attention for perceptually-guided selective rendering of animations. In *Proceedings of GRAPHITE 2005, ACM SIGGRAPH* (2005).

[Mit87] MITCHELL D. P.: Generating antialiased images at low sampling densities. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), ACM Press, pp. 65–72.

[MR98] MACK A., ROCK I.: *Inattentional Blindness*. MIT Press, 1998.

[Mys98] MYSZKOWSKI K.: The Visible Differences Predictor: Applications to global illumination problems. In *Eurographics Workshop on Rendering* (1998), pp. 223–236.

[NI05] NAVALPAKKAM V., ITTI L.: Modeling the influence of task on attention. *Vision Research 45*, 2 (2005), 205–231.

[OLG*05] OWENS J. D., LUEBKE D., GOVINDARAJU N., HARRIS M., KRÜGER J., LEFOHN A. E., PURCELL T. J.: A survey of general-purpose computation on graphics hardware. In *Eurographics 2005, State of the Art Reports* (Aug. 2005), pp. 21–51.

[PP99] PRIKRYL J., PURGATHOFER W.: *Overview of Perceptually-Driven Radiosity Methods*. Tech. Rep. TR-186-2-99-26, Vienna, Austria, 1999.

[PS89] PAINTER J., SLOAN K.: Antialiased ray tracing by adaptive progressive refinement. In *SIGGRAPH '89* (1989), ACM Press, pp. 281–288.

[PS03] PETERS C., SULLIVAN C. O.: Bottom-up visual attention for virtual human animation. In *Computer Animation for Social Agents* (2003).

[RPG99] RAMASUBRAMANIAN M., PATTANAIK S. N., GREENBERG D. P.: A perceptually based physical error metric for realistic image synthesis. In *SIGGRAPH '99* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 73–82.

[SC99] SIMONS D., CHABRIS C.: Gorillas in our midst: Sustained inattentional blindness for dynamic events. In *Perception* (1999), vol. 28, pp. 1059–1074.

[SCM04] SUNDSTEDT V., CHALMERS A., MARTINEZ P.: High fidelity reconstruction of the ancient egyptian temple of kalabsha. In *AFRIGRAPH 2004* (November 2004), ACM SIGGRAPH.

[SDL*05] SUNDSTEDT V., DEBATTISTA K., LONGHURST P., CHALMERS A., TROSCIANKO T.: Visual attention for efficient high-fidelity graphics. In *Spring Conference on Computer Graphics (SCCG 2005)* (May 2005).

[SGGC05] SUNDSTEDT V., GUTIERREZ D., GOMEZ F., CHALMERS A.: Participating media for high-fidelity cultural heritage. In *Proceedings of VAST 2005, Eurographics* (2005).

[UWP05] ULBRICHT C., WILKIE A., PURGATHOFER W.: Verification of physically based rendering algorithms. In *State of the Art Reports, Eurographics 05* (2005), pp. 95–112.

[VF94] VOORHIES D., FORAN J.: Reflection vector shading hardware. In *SIGGRAPH '94* (New York, NY, USA, 1994), ACM Press, pp. 163–166.

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *SIGGRAPH '78* (1978), ACM Press, pp. 270–274.

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *SIGGRAPH '88* (1988), ACM Press, pp. 85–92.

[Yar67] YARBUS A.: Eye movements during perception of complex objects. In *Eye Movements and Vision* (1967), pp. 171–196.

[YPG01] YEE H., PATTANAIK S., GREENBERG D.: Spatiotemporal sensitivity and Visual Attention for efficient rendering of dynamic Environments. In *ACM Transactions on Computer Graphics* (2001), vol. 20, pp. 39–65.