

There-Reality: Selective Rendering in High Fidelity Virtual Environments



Alan Chalmers, Kurt Debattista, Georgia Mastoropoulou and Luis Paulo dos Santos

Abstract—There-reality environments are those virtual environments which evoke the same perceptual response from a viewer as if they were actually present, or there, in the real scene being depicted. While it is possible to compute highly accurate representations of real scenes, the computational requirements of such a full physically-based global illumination solution are significant, currently precluding its computation on even a powerful modern PC in reasonable let alone real time. However, to provide a useful tool, there-reality environments need to be interactive. A key factor to consider if we are ever to achieve such “Realism in Real-Time” is that we are computing images for humans to look at. Although the human visual system is very good, it is by no means perfect. By understanding what the human does, or perhaps more importantly, does not see, enables us to save substantial computation effort without any loss of perceptual quality of the resultant image. This paper describes the novel techniques of selective rendering which allow us to direct computational resources to those areas of high perceptual importance while avoiding computing any detail which will not be perceived by the viewer. Such selective rendering methods offer us the real possibility of achieving there-reality environments of complex scenes at interactive rates.

Index Terms—Rehabilitation, virtual human interface, ambient assisted living.

I. INTRODUCTION

Realism in Real-Time, computing physically-based global illumination graphics at interactive rates has been a much sought after goal in computer graphics research for many years. There-Reality environments, virtual environments which are as “real” as if the viewer was “there” in the real scene, are increasingly being considered as a way of providing realistic training in a safe and controlled manner. To provide an adequate training simulation, such environments must be highly realistic, and interactive. The complexity of modelling a real scene is such that, even with the advent of modern, powerful GPUs, it is simply not possible to render high-fidelity

Manuscript Received on December 21, 2006. This work formed part of the Rendering on Demand project, supported by the 3C Research program of convergent technology research for digital media processing and communications.

Alan Chalmers is a full Professor of Visualisation at the new Warwick Digital Laboratory, WMG, University of Warwick, UK, email: alan.chalmers@warwick.ac.uk.

Kurt Debattista is a Research Associate at the Warwick Digital Laboratory, WMG, University of Warwick, UK, email kurt.debattista@gmail.com.

Georgia Mastoropoulou is a former PhD student on the Rendering on Demand project. She now works for the Ministry of Education in Greece, email: G.Mastoropoulou@bristol.ac.uk.

Luis Paulo dos Santos is an Auxiliar Professor at the Department of Computer Science, Universidade do Minho, Portugal, email: psantos@mail.di.uminho.pt.

images of such scenes in reasonable, let alone real-time. If, however, we consider that we are rendering images for humans to view, then we can exploit knowledge of the human visual system (HVS), for although the HVS is good, it is not perfect. When viewing a scene our visual attention is a series of conscious and unconscious processes which enables us to find and focus on relevant parts of the scene quickly and efficiently. Human eyes do not scan a scene in a raster-like fashion, but rather our eyes jump rapidly between features within the scene, so-called saccades.

Visual attention comprises two processes, bottom-up and top-down [James 1890]. The bottom-up process is purely stimulus driven. This is evolutionary, attracting our eyes without volitional control to salient features in a scene, for example movement, which may be a predator lurking in a bush, or the red apple in a green tree, indicating that the fruit is ripe. A saliency map may be calculated for any image, highlighting the perceptually important features [Itti et. al. 1998]. The top-down process, on the other hand, is under volitional control, with the HVS over-riding the bottom-up process and focusing attention on one more objects that are relevant to accomplish a specific visual task, for example, looking for a lost child, or counting the number of occurrences of a certain object. When conducting a task within a scene, even conspicuous objects in a scene that would normally attract the viewer’s attention are ignored if they are not relevant to the task at hand. This is known as Inattentional Blindness [Mack and Rock 1998].

In this paper, we will show how selective rendering techniques, which allow us to direct computational resources to those areas of high perceptual importance while avoiding computing any detail which will not be perceived by the viewer, offer the real possibility of achieving there-reality environments at interactive rates.

The rest of the paper is organized as follows. Section II gives a brief overview of the previous work in selective rendering. Section III highlights an application for which there-reality can play a key role. Three different selective renderers are presented in Section IV, including those exploiting multi-modal effects, and shows and their potential for significantly reducing overall computation time. Finally, Section V presents some conclusions and discusses how selective rendering has the opportunity, perhaps coupled with parallel rendering, to help us reach the goal of .Realism in Real-Time.

II. BACKGROUND

The principles of selective rendering have previously been

applied in computer graphics in three general ways. Adaptive techniques concentrate computational resources on those areas of an image which are deemed currently most important, according to some criteria. Incremental techniques, on the other hand, progressively refine an image until some stopping condition is met. Finally, component based approaches allow the system to control rendering at the component level, for example, the specular or transparent components. These components will be computed only if, for example in a time-constrained system, sufficient time remains. Selective rendering algorithms can thus be defined as those techniques which require a number of rendering quality decisions to be taken and acted upon prior to, or even dynamically during the actual computation of any image or frame of an animation [Debattista 2006].

The work by Clark in the 1970s was amongst the first to consider some form of selective rendering related to level of detail [Clark 1976]. Since then selective rendering for rasterisation approaches have lowered computational costs by reducing the number of geometrical objects which are sent to the rendering pipeline. Much of this recent work is summarized in [Luebke et al. 2002]. Bergman et al. took a different approach. Here the complexity of the shading is selectively refined using a set of heuristics [Bergman et al. 1986].

Within the field of global illumination, progressive refinement radiosity [Cohen et al. 1988] provided the viewer with an approximate image which was refined over time. An overview of other perceptually driven radiosity algorithms is given in [Prikryl and Purgathofer 1999]. Whitted's original ray tracer included a selective anti-aliasing technique where further rays were only shot within a pixel if the intensity at pixel corners was below a predefined threshold [Whitted 1980]. Since then Mitchell developed adaptive anti-aliasing for ray tracing [Mitchell 1987], Painter and Sloan presented an adaptive progressive refinement ray tracer [Painter and Sloan 1989], and in 1991 Chen et al. presented a progressive re-refinement algorithm for a multipass rendering algorithm combining progressive radiosity, ray tracing and backwards ray tracing, where the individual passes were interruptible [Chen et al. 1991].

In 1998 Myszkowski introduced the visual difference predictor to global illumination [Myszkowski 1998]. Based on Daly's VDP [Daly 1993], Myszkowski's VDP was used to selectively stop rendering in a Monte Carlo path tracer based on the perceptual difference between images at regular intervals. Since this pioneering research, a growing body of work has appeared on using visual perception in computer graphics. Bolin and Meyer used a visual difference predictor both to direct the next set of samples within a stochastic ray tracing framework and as a stopping condition [Bolin and Meyer 1998]. Ramasubramanian et al. [Ramasubramanian et al. 1999], Haber et al. [Haber et al. 2001] and Yee et al. [Yee et al. 2001] all exploited knowledge of the salient parts of a scene to render these at high quality, while the remainder could be computed at a lower quality without the viewer being aware of this quality

difference. Cater et al. [Cater et al. 2002; Cater et al. 2003] and subsequently Sundstedt et al. [Sundstedt et al. 2004] extended this to also consider the visual task a user may be performing in a scene. By only computing at high quality those parts of the scene that the viewer is actually attending to while performing the task, significant computational time was saved with out any loss in perceptual quality being experienced by the user. Recently [Anson et al. 2006] showed how effective selective rendering could be in the presence of participating media. A good overview of perceptually adaptive graphics can be found in [O' Sullivan et al. 2004].

In the case of component based rendering, Wallace et al. presented a multipass algorithm that computed the diffuse component with a rendering pass and used a z-buffer algorithm for view dependent planar reflections [Wallace et al. 1987]. The view independent nature of the indirect diffuse component of global illumination was exploited by Ward et al. to sufficiently populate the innovative irradiance cache, from which other samples could interpolate with significant computational cost savings [Ward et al. 1988]. Other component based rendering have included [Sillion and Puech 1989; Shirley 1990; Heckbert 1990; Chen et al. 1991; Stokes et al. 2004].

In 1998, Slusallek et al. introduced the concept of lighting networks to render scenes based on the users combining the implementations of different rendering algorithms, typically radiosity and ray tracing techniques, into a network [Slusallek et al. 1998]. In 2005, Debattista et al. introduced a component-based selective rendering system in which the quality of every image, and indeed every pixel, was controlled by means of a component regular expression (crex). This crex, inspired by Heckbert's light transport notation [Heckbert 1990], provided a flexible mechanism for controlling which components are rendered and in which order. The crex can thus be used as a strategy for directing the light transport within a scene and also in a progressive rendering framework.

When confronted by multimodal stimuli, humans may not be able to attend to all the stimuli at once. We thus use attention filters to bring what we deem to be important parts of the world around us into awareness. Although a large amount of sensory information can be absorbed simultaneously, the attention paid to one source may be reduced while the brain processes some other sensory stimuli [Broadbent 1958, Carlson 1993]. This is particularly true of the crossmodal interaction between vision and audition, where there can be a small, but significant limitation of attention to one sense, while attention is directed to the other [Massaro and Warner 1977]. An example of this is turning the radio down while we attempt to find a particular street sign. Auditory-visual crossmodal interactions are being increasingly used in virtual reality. For example, Storms found that high-quality sounds coupled with high-quality visual stimuli increase the perceived quality of the visual displays [Storms 1998], while Winkler and Faller showed that both audio and video quality contribute significantly to the perceived audiovisual quality [Winkler and Faller 2005]. More recently Mastoropoulou et al. demonstrated that the presence of

audio could enable the quality of rendering of high-fidelity graphics to be significantly reduced without the viewer being aware of this quality reduction [Mastoropoulou et al 2005].

III. APPLICATIONS

Computer reconstructions of archaeological sites provide a safe and controlled manner in which to investigate hypotheses of the past. However, to avoid the very real danger of “misrepresenting the past”; these reconstructions need to be highly accurate and incorporate all available evidence.

Before the invention of electricity, lighting within ancient environments was provided by sun or moon light, and fire. The fuel used to create a fire directly affects the visual appearance of an environment [Devlin and Chalmers 2001]. Furthermore, fires are dynamic, causing moving shadows, as well as smoke, all of which may have a significant affect on how a site may have been perceived [Sundstedt et al. 2005]. To simulate the perceptual affects of the dynamic nature of ancient lighting, virtual archaeology environments thus need to not only be highly accurate, but also interactive.

One example of the importance of such high-fidelity interactive environments is in the reconstruction of the prehistoric rock carvings of Cap Blanc [Robson Brown et al. 1999]. Overlooking the Beaune valley in the Dordogne, France, the site of Cap Blanc contains dramatic Upper Palaeolithic haut relief carvings. Discovered in 1909 by Raymond Peyrille, the site comprises a frieze of horses, bison and deer, some overlain on other images. The frieze which was carved some 15,000 years ago into the limestone as deeply as 45cms, covers 13m of the wall of the shelter.



Fig. 1. Reconstruction of horse frieze lit by simulated by 55W halogen light as the site is seen today.



Fig. 2. Reconstruction of horse frieze lit by simulated animal tallow candle as the site may have been seen some 15,000 years ago.

Images of a detailed reconstruction of the frieze were rendered in Radiance [Ward 1994]. Fig. 1 shows the horse illuminated by a simulated 55W incandescent bulb of a

low-power floodlight, which is how visitors view the actual site today, while Fig. 2 shows the horse illuminated by a simulated animal fat tallow candle as it may have been viewed 15,000 years ago. As can be seen the difference between the two images is significant with the candle illumination giving a “warmer glow” to the scene as well as more shadows. However, it is only when the flame is animated that it becomes apparent that perhaps our ancestors, by careful use of 3D structure and the dynamic nature of the flame, were making animations some 15,000 years ago [Chalmers et al. 2000].

So, while we will never know exactly how a site may have looked in the past, by ensuring an interactive environment with the highest fidelity reconstruction possible, we can allow archaeologists and the general public to explore possible hypotheses with confidence.

IV. SELECTIVE RENDERERS

Selective rendering is the technique of directing computational efforts to those parts of the rendering process which are deemed currently most important according to some predefined, or dynamically acquired, information. In this section we present three selective renderers which exploit different approaches to determine where best to direct the computation. In addition, we show how the presence of multimodal stimuli may be exploited to reduce the number of frames that need to be computed a second in order to achieve a perceived interactive environment.

4.1 Detecting Key Objects in a Frame

This dynamic approach enables the selective renderer to detect the presence of predefined key objects, for example an on-screen distractor (OSD), such as a sound-emitting object, in any frame of animation as an integral part of the rendering process. When one or more OSDs are present, these are rendered in high quality, while the remainder of the frame can be rendered at a much lower quality without the viewer being aware of this quality difference [Cater et al. 2003; Mastoropoulou et al. 2005]. This selective renderer can be used for both top-down and bottom-up visual attention processes, that is, whether an object on the screen draws the viewer's attention as part of a task or involuntarily.

The first phase of this selective renderer involves rendering to a base quality level, typically 1 ray per pixel, while locating the OSDs. The second phase renders the image selectively applying a foveal angle gradient decrease in quality around the projected image of the detected OSDs and maintaining quality consistency using a quality buffer, q-buffer. Rays per pixel are used to modulate the rendering quality. Since this selective renderer is primarily created for animations, an animation file maintains a list of all frames with the default rendering quality for each frame and whether any OSD should be considered more salient. Frames that contain distracters are tagged as selective. We also need the option of not having objects as always being distracters, but perhaps being distracters only for certain frames. This is particularly useful for sound emitting

objects which may only be OSDs when they are actually emitting sounds [Mastoropoulou et al. 2005]. Those frames which do not contain any OSDs are rendered at the default high quality.

When rendering the frames tagged as selective in an animation, the selective rendering process can be viewed as a two pass process for each frame. In the first pass, the image is rendered using the traditional rendering method up to the base quality. In this pass the distracter objects are detected through object identification of the primary ray or certain class of secondary rays, as they intersect the destructors' geometry. We term the rays that are allowed to detect the distracters detectable rays. Only certain types of secondary rays are detectable rays, such as pure specular which represent pure reflections of the object. Other secondary rays, for example, indirect diffuse rays, would reflect very little of the true nature of the distracter.

A data structure termed an on-screen distracter list (OSDL) is a queue data structure responsible for storing OSD object data and pixel locations. When the intersection routine for the detectable rays returns the object hit, a distracter object list is consulted and if the object appears on the list, the OSDL adds the detectable object to the list. The first phase ends when the image is rendered entirely to the base quality, at which point the computation of the OSDL is completed for the first phase. The OSDL can be visualized as an on-screen distracter map. OSD maps for a number of scenes are shown in Fig. 3.

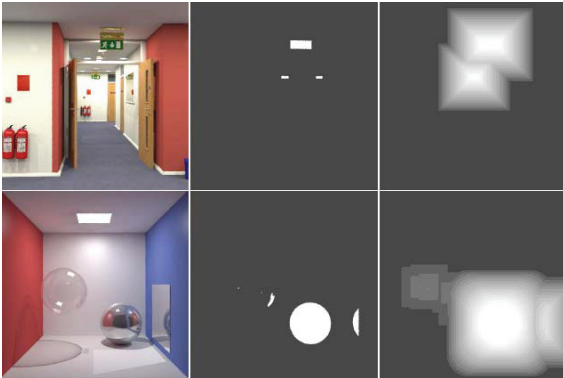


Fig. 3. Selective rendering of key objects. From top to bottom: the Corridor scene and the Cornell Box scene detecting OSDs with modulation for the reflected on-screen distracters. From left to right: the rendered scenes, the OSD maps and the q-buffers. See Color Plate 1.

The second phase introduces another data structure, the quality buffer (q-buffer) which ensures that the correct number of rays is shot for each pixel. The q-buffer is inspired by the z-buffer used for solving depth issues in hidden surface removal of rasterised rendering [Catmull 1978] and is a buffer equal in size to the resolution of the image to be rendered. It stores the quality in rays per pixel of the number of rays rendered up to that point for that given pixel. At the beginning of the second phase all entries in the q-buffer are initialized to the value of the base quality. In the second phase, the OSDL is parsed and for each entry a number of rays intended to be shot is calculated. The ray's hit point is also tested to discover whether or not the pixel is a boundary pixel of the projected

image of the distracter object. If there are no distracter objects at this point, no further action is taken. If the pixel corresponds to an internal point on the distracter's projected image, the difference between the distracter's value and the q-buffer at that pixel is calculated. If this difference is positive a number of primary rays equal to the difference is shot. The q-buffer at that pixel is then set equal to the distracter's quality value. If the difference is negative or zero, no action is taken.

For border pixels, the renderer degrades the quality around the border of the distracter objects within a user-defined radius, usually the foveal angle. This option provides a method of rendering around the foveal angle that is not limited to the size of the object. Each pixel within this radius is cycled through and, for each pixel, the desired quality in rays per pixel is calculated. The q-buffer is then consulted and if the new quality is greater than the corresponding entry in the q-buffer the difference in rays is shot and the q-buffer entry updated in the same method as described above. The q-buffer is necessary to protect the quality of pixels lying within the degradation radius of more than one different objects which might result in more rays than necessary contributing to a given pixel. As the rendering progresses, the image is refined so the projected distracter objects on the image plane's boundaries might change subtly. This algorithm ensures that the new boundaries are updated accordingly by storing the new information onto the OSDL. The q-buffers for various OSD maps and scenes can be seen in Fig. 3.

This selective renderer was implemented as an extension to the light simulation software Radiance [Ward 1994]. A number of user adjustable parameters were added. Firstly, a base quality and a high quality is set for each frame. Secondly, for animations, a frame needs to be tagged as selective or high quality, and if selective, a list of distracter objects for that frame is identified. If a frame is tagged as high quality the entire image is rendered in high quality. This option is used when no distracters are present. For frames that are tagged as selective, the base quality is rendered in the first phase and the selective quality in the second phase, following the algorithm described above. Note that some frames might not have distracter objects and still be marked as selective; useful for when there is a distracter outside the view images, such as an abrupt sound. The results show, for a number of scenes, how the presence of OSDs may be exploited to significantly reduce overall computation time.

TABLE 1: RESULTS FOR SCENES USED WITH THE ON-SCREEN DISTRACTERS SELECTIVE RENDERERS. TIMING IN SECONDS

	Corridor	Cornell
gold	2,454	179
selective	335	50
speedup	7.32	3.58

In Fig. 3, the distracters are: the exit signs for the Corridor scene and the glossy sphere for the Cornell Box. For the corridor scene the rendering parameters do not detect the distracter in the reflections. For the Cornell Box scene the

TABLE 2: THE COMPONENT REGULAR EXPRESSION DESCRIPTION (CREX). †SHADER SPECIFIC COMPONENT.

Character	Description
()	Group one or more component. The latter components in the group execute rays spawned by the former in the group.
<>	Group one or more component. Any spawned ray is never launched within the group but is executed after the group terminates.
{ }	Group one or more component. Group in { } is modulated by an importance map.
<i>k</i> positive integer	Execute last component or group <i>k</i> times.
*	Execute until no more rays spawned.
D	Indirect diffuse.
S	Indirect specular.
C	Indirect glossy.
T	Transmitted glass/dielectric.
R	Reflected glass/dielectric.
M	Mirror.

TABLE 3: CREX BNF. ‡ IMPLEMENTATION SPECIFIC FOR OUR RADIANCE-BASED VERSION.

<crex>	= (<crex>) <crex>> {<crex>} <crex> <component> <crex><component> <crex><mult>
<component>	= D G S T [#] R [#] M [#]
<digit>	= 0 1 2 3 4 5 6 7 8 9
<integer>	= <digit> <digit><integer>
<mult>	= * <integer>

rendering parameters were set to modulate the importance depending on the material of the object in the reflections. The differences can be clearly seen in the q-buffered OSD maps of the Cornell box scene. Rendering parameters were set to default settings with a maximum of 16 rays per pixel, and a base quality of 1 ray per pixel. The irradiance cache was pre-computed. Results, in seconds, to render the images are shown in TABLE 1. The results demonstrate a significant speedup for some of the renderings.

4.2 Selective Component-Based Rendering

Previous work on selective rendering has predominantly determined quality as a function of number of rays traced, since rays per pixel was the only selective variable. The more salient a pixel, the more rays were traced for that pixel as in [Cater et al. 2003; Sundstedt et al. 2004] and most of the selective renderers discussed in the previous sections. In Yee et al.'s work the saliency effected only the indirect diffuse computation, in particular the accuracy of the search radius used in examining previously cached samples inside the irradiance cache [Yee et al. 2001]. Our approach extends this notion by empowering the renderer with the ability to terminate the path of a ray at any point of its execution as dictated by a component regular expression termed the crex [Debatista et al. 2005].

Inspired by Heckbert's light transport notation [Heckbert 1990], the component regular expression, or crex, informs the renderer on the order in which the components are to be rendered. The crex takes the form of a string of characters with each character representing either a component or a special character used for recurrence or grouping, as shown in TABLE 2. The BNF of the syntax is presented in TABLE 3. The alphabetic characters each represent an individual component. The order of the components in the crex dictates the order in which the components are rendered. Components spawn rays to be executed by subsequent components or groups. The integer multiplier is used to execute a component or group *k* times. The * operator is used to execute the previous component or group until no more rays belonging to the recursed components are

spawned. The groups () and <> are used to group components together. When using () the components or groups of components within a group inherit the spawned rays from the previous components within the group. On the other hand when using <> all of the rays spawned within the group will be executed when the <> block terminates. The components within <> can be executed in parallel.



Fig. 4. One set of images from the Corridor scene used for the visual attention experiment: (left) high quality image (HQ) and (right) component-based quality (CBQ). See Color Plate 2.

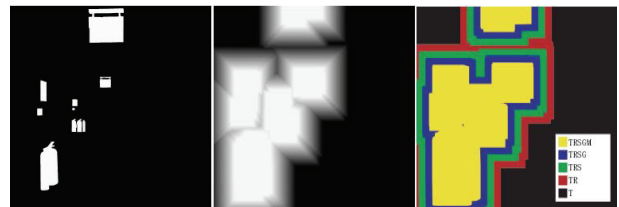


Fig. 5. A visualisation of the importance map: (left) the task map (HQ), (middle) the task map with foveal angle gradient and (right) a colour-coded visualisation of which components of the crex are rendered for each pixel for a crex of T {RSGM}.

In this form of selective rendering, the quality is controlled by the components rather than the more traditional rays per pixel. The initial rendering stage renders a few of the components and then generates an importance map [Sundstedt



Fig. 6. The Desk and Corridor scenes. From left to right and top to bottom: pre-selective rendering images, the selective guidance, selective component-based. See Color Plate 3.

et al. 2005]. The part of the crex which is grouped in $\{ \}$ is modulated by the value in the importance map for a given pixel. No recursion (*) is allowed in $\{ \}$, but different $\{ \}$ can be separated. The components in $\{ \}$ are ordered by importance such that the first components require a lesser value in the importance map to be rendered. Effectively quality of this selective renderer is dictated by the number of component rays shot. The importance map for the rendered image seen in Fig. 4 can be seen in Fig. 5. The importance map in this case is based on the task only. Fig. 5 shows a colour coded visualisation of how the crex effects the individual pixels for the rendered image. The part of the crex prior to the first $\{ \}$ is used as pre-selective guidance. Further examples of the entire selective rendering pipeline can be seen in Fig. 6. For example, the Desk scene is initially rendered with a crex of TTM from which the importance map is generated and the modulated using the components in the $\{ \}$ part of the full crex of TTM{RSGDTTRSG}.

In order to demonstrate the effectiveness of component-based rendering while exploiting visual attention we ran a task-based psychophysical experiment similar to [Cater et al. 2003] except that in our case the quality is determined by the crex rather than the resolution. The full high quality rendering shown in Fig. 4 took one hour while the selectively rendered image only required half an hour using a crex of T {RSGM}.

TABLE 4: SPEEDUP FOR THE SELECTIVE COMPONENT-BASED RENDERER. TIMING IN SECONDS

	Desk	Corridor
gold	326	712
Cbr	86	163
speedup	3.79	4.37

The following results only use saliency maps. Also, since we are limiting ourselves only to component-based rendering, all rendering is performed with one ray per pixel at a resolution of

512 x 512. We use scenes shown in Fig. 6. For rendering the Desk scene the crex used was TTM{RSGDTTRSG}, while for the Corridor scene the crex was TT{TTMRSGD}2. Results for the component based rendering (cbr), shown in TABLE 4, demonstrate reasonable speedup compared to the full (gold) rendering times.

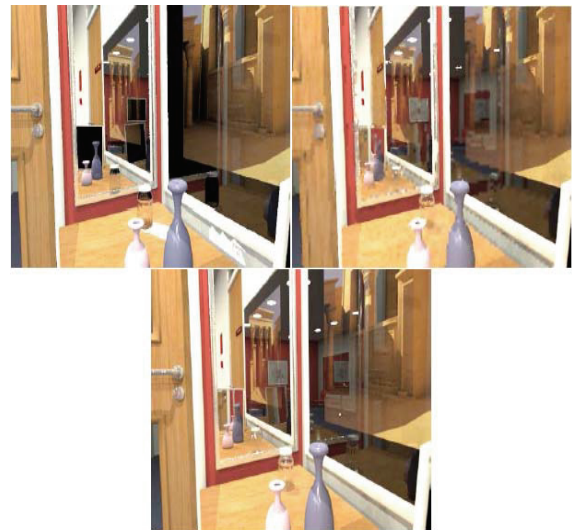


Fig. 7. Comparisons between the time-constrained renderers for the Desk scene. Selective component-based time-constrained rendering (top left) and selective time-constrained rendering (top right). The reference image (below).

We may now compare selective rendering between a traditional rays-per-pixel selective renderer and the selective component-based render using time-constrained rendering systems [Debattista 2006] to fix the time allowed for each renderer to compute. For the traditional selective renderer a maximum of 16 rays per pixel is used, while a fixed 4 fixed rays per pixel is set for the selective component-based renderer. A time constraint of 690 seconds, equivalent to half the time it took to render the reference image, was used. The component-based renderer uses a crex of MT{TRSGD}3.

Resulting images and the reference image are shown in Fig. 7.

The results demonstrate that the component-based time-constrained renderers scale better to lower time constraints since the scheduling and profiling is at a finer grain than that of the traditional time constrained renderer. Further experiments may be needed for conclusive evidence.

4.3 Parallel Selective Rendering

In this section we present a selective parallel rendering framework and demonstrate how it is possible to significantly reduce rendering times by exploiting these approaches to near real-time high-fidelity rendering for complex scenes [Chalmers et al. 2006]. We demonstrate how selective rendering can make use of various hardware in particular, distributed systems and graphics hardware to achieve significant performance improvements.

The selective rendering framework is shown in Fig. 8. The first stage involves generating an image preview using rapid rasterisation rendering. The guidance for the selective rendering is based on an importance map, which can be composed of many different maps. Many of these can be generated using graphics hardware. The final selective rendering stage utilizes parallel computing to improve rendering times.

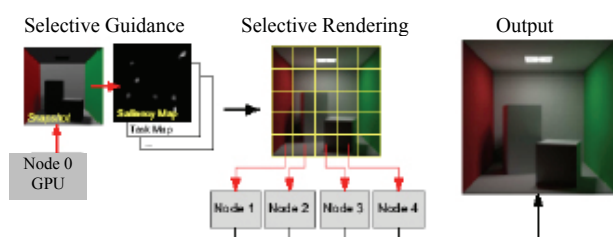


Fig. 8. Rendering on demand framework.

We use the Snapshot technique to produce, using graphics hardware, a rapid image estimate of the scene to be rendered [Longhurst et al. 2005a]. A saliency map may be generated from Snapshot using object space knowledge to identify conspicuity for a number of features: motion similar to the work of [Yee et al. 2001], 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 et al. saliency map [Itti et al. 1998], is composed of three channels: color, intensity and orientation. All of these features are calculated on the GPU for maximum performance, in the order of tens of milliseconds. 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 [Longhurst et al. 2005b; Longhurst et al. 2006]. Fig. 9 shows some scenes rendered using the Snapshot, the generated importance map, which in these cases is only a saliency map, and the final rendering.

In this parallel selective renderer, the selective rendering process uses parallel processing to speed up the rendering even further. Also the quality is not only modulated by rays per pixel but also by a separate component, the irradiance cache search

radius [Yee et al. 2001], which improves rendering times substantially for rendering scenes without a pre-computed irradiance cache. The importance map is used by the master process to subdivide the workload and by the slave processes to decide how to modulate the rendering parameters.

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. The importance map is used 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 improvement is of 2% to 4% in terms of computation time when compared to a fixed tile demand driven approach. Although the computational requirements of each individual ray may differ, the demand driven approach together with the subdivision map alleviates the problem significantly.

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 tile assigned to them using the importance map to direct the rendering of each pixel. The sole selective variable in this case is rays per pixel. When the slave finishes executing the job, it asks for more work from the master until the entire process is completed.



Fig. 9. The Corridor scene (top) and Tables scene (bottom). Left to right: the rapid image estimate, the saliency map and the selectively rendered final image. See Color palte 4

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 [Ward et al. 1988] 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 parallelize. As with the previous selective renderers, this parallel approach was implemented in Radiance. Distributed computation over a network of workstations was performed using the MPI message passing protocol.

In addition, the irradiance cache was parallelized using the component-based parallel irradiance cache [Debattista et al.

2006], which substantially improves parallel performance by subdividing the indirect diffuse computation from the rest of the components. Results rendered on a cluster of 16 2.4 GHz processors with 3GB RAM are presented in TABLE 5. As can be seen the speedup is significant overall compared to the traditional uniprocessor computation.

4.4 Crossmodal Effects

Finally we show how cross modal effects, such as the presence of sound, can influence how a viewer perceives a scene. Computing 25 frame per second (fps) to maintain interactive rates requires significant resources. Experiments conducted in [Mastropoulou et al. 2005] showed that in the presence of audio stimuli, and more specifically sound effects, viewers consistently failed to notice variations in the motion smoothness between walkthrough animations displayed at different rates. These same differences were immediately obvious when no sound was present, Fig. 10.

TABLE 5: TIMINGS IN SECONDS FOR THE RENDERING ON DEMAND SELECTIVE RENDERER. TSU FOR TOTAL SPEED UP RESULTING FROM BOTH PARALLELISM AND SELECTIVE RENDERING.

	Cornell Box		Tables		Corridor	
	Time	Tsu	Time	Tsu	Time	Tsu
gold	450	1	4,057	1	4,500	1
Selective	130	3.46	1,793	2.26	1,859	2.42
2	62	7.26	832	4.88	882	5.10
4	34	13.24	457	8.88	451	9.98
8	17	26.47	249	16.29	234	19.23
16	11	40.91	149	27.23	141	31.91

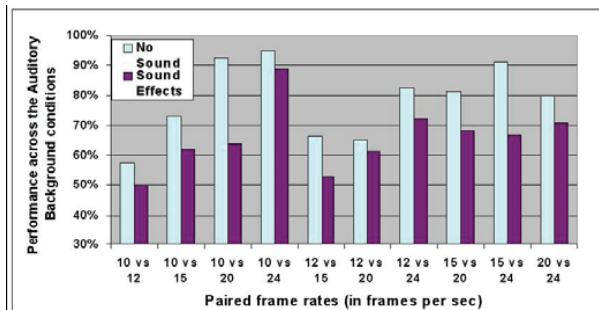


Fig. 10. The performance of all Subjects across the “No Sound” and “Sound Effect” conditions, separately for each frame rate combination.

Participants watched pairs of computer generated walkthroughs with varying audiovisual content and we re asked which of the two walkthroughs had better visual quality. A 2-Alternative Forced-Choice method was used. Each animated sequence was rendered at 5 different frame rates: 24 fps, 20 fps, 15 fps, 12 fps and 10 fps, giving 25 frame rate combinations. Forty subjects were tested, half of whom saw the animations with the sound effects and the other half in silence.

As Fig. 10 clearly shows, audio stimuli do indeed attract a part of the viewers’ attention away from the visuals and as a result, the observers find it more difficult to distinguish smoothness variations between audiovisual composites displayed at different rates, than between silent animations. This effect is even more pronounced with viewers who are not familiar with animated computer graphics [Mastropoulou

2005].

These results have significant benefit for any selective rendering system. When sound effects are present, to achieve perceptually interactive display rates, even for users familiar with computer graphics, fewer frames need to be computed per second than the traditional 25 fps.

V. CONCLUSIONS

There-reality environments provide simulations of reality as close as is possible to the real scene being depicted. To provide a meaningful system to a user, these environments should not only be highly realistic, but also allow a user to investigate the environments interactively. A key factor in striving for “Realism in Real-Time” is to realize that we are computing high-fidelity images for humans, and while the human visual system is very good it is not perfect. Exploiting knowledge of the human visual system enables us to selectively render only parts of a scene at the highest quality, and the remainder of the scene at a significantly lower quality, and thus much less computational cost, without the viewer being aware of this difference in quality. In this paper we have discussed a number of selective rendering approaches. As we can see, when selective rendering is combined with parallel processing we are able to achieve significant speedups on a modest number of processors, for example a speed up of over 40 on 16 processors for the Cornell box scene. Coupled with the exploitation of multimodal interactions in the virtual environment, we will be able to compute even less while maintaining perceived interactive frame rates. Future work will look at combining the different selective rendering methods into a single multimodal selective rendering framework capable of adopting the most appropriate approach depending on the scenes being rendered. Such a system will offer the real possibility of achieving realism in real-time.

ACKNOWLEDGMENT

The work reported in this paper has formed part of the Rendering on Demand (RoD) project within the 3C Research program whose funding and support is gratefully acknowledged.

REFERENCES

- [1] O. Anson, V. Sundstedt, D. Gutierrez and A. Chalmers. Proceedings of the *Symposium on Applied Perception in Graphics and Visualization*, in APGV 2006, *ACM SIGGRAPH*, 2006.
- [2] L. Bergman, H. Fuchs, E. Grant and S. Spach. Image rendering by adaptive refinement, In *SIGGRAPH’86*, ACM Press, pp. 29-37, 1986.
- [3] D. Broadbent. *Perception and Communication*, London: Pergamon Press, 1958.
- [4] N. R. Carlson. *Psychology, The Science of Behavior*, Needham Heights, MA: Allyn and Bacon, 1993.
- [5] M. R. Bolin and G. W. Meyer. A perceptually based adaptive sampling algorithm, In *SIGGRAPH’98*, ACM Press, pp. 299-309, 1998.
- [6] K. Cater, A. Chalmers and P. Ledda. Selective quality rendering by exploiting human inattention blindness: looking but not seeing, In *Proceedings of the ACM symposium on Virtual reality software and*

- technology, ACM Press, pp. 17-24, 2002.
- [7] K. Cater, A. Chalmers and G. Ward. Detail to Attention: Exploiting Visual Tasks for Selective Rendering. In Proceedings of the *Eurographics Symposium on Rendering*, pp. 270-280, 2003.
- [8] E. Catmull. A hidden-surface algorithm with anti-aliasing. In *SIGGRAPH '78*, ACM Press, pp. 6-11, 1978.
- [9] A. Chalmers, K. Debattista, R. Gillibrand, P. Longhurst and V. Sundstedt. Rendering on demand. In *EGPGV2006 - 6th Eurographics Symposium on Parallel Graphics Visualization*, Eurographics, pp. 9-18, 2006.
- [10] A. Chalmers, C. Green and M. Hall. Firelight: Graphics and Archaeology, *SIGGRAPH 2000 Electronic Theatre*, 2000.
- [11] S. E. Chen, H. E. Rushmeier, G. Miller and D. Turner. A progressive multi-pass method for global illumination. In *SIGGRAPH'91*, ACM Press, pp. 165-174, 1991. J. H. CLARK. Hierarchical geometric models for visible surface algorithms, *Commun. ACM* 19, vol. 10, pp. 547-554, 1976.
- [12] M. F. Cohen, S. E. Chen, J. R. Wallace and D. P. Greenberg. A progressive re_ement approach to fast radiosity image generation. In *SIGGRAPH'88*, ACM Press, pp. 75-84, 1988.
- [13] S. Daly. The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In *Digital Images and Human Vision*, A. B. Watson, MIT Press, Cambridge, MA, pp. 179-206, 1993.
- [14] K. Debattista, V. Sundstedt, L. Santos and A. Chalmers. Selective component-based rendering. In Proceedings of *GRAPHITE 2005*, ACM SIGGRAPH, 2005.
- [15] K. Debattista, L. P. Santos and A. Chalmers. Accelerating the irradiance cache through parallel component based rendering. In *EGPGV2006-6th Eurographics Symposium on Parallel Graphics Visualization*, Eurographics, pp. 27-34, 2006.
- [16] K. Debattista. Selective Rendering for High-Fidelity Graphics, PhD Thesis, University of Bristol, 2006.
- [17] K. Devlin and A. Chalmers. Realistic Visualization of the Pompeii Frescoes}, *AFRIGRAPH 2001*, ACM SIGGRAPH, pp. 43-47, 2001.
- [18] J. Haber, K. Myszkowski, H. Yamauchi and H. P. Seidel. Perceptually guided corrective splatting. In *Computer Graphics Forum*, vol. 20, pp. 142-152, 2001.
- [19] P. S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In *SIGGRAPH'90*, ACM Press, pp. 145-154, 1990.
- [20] L. Itti, C. Koch and E. Niebur. A model of Saliency-Based Visual Attention for Rapid Scene Analysis. In *Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254-1259, 1998.
- [21] W. James. A saliency-based search mechanism for overt and covert shifts of visual attention. In *Principles of Psychology*, 1890.
- [22] P. Longhurst, K. Debattista and A. Chalmers. Snapshot: A rapid technique for driving a selective global illumination renderer. In *WSCG 2005 SHORT papers proceedings*, 2005.
- [23] P. Longhurst, K. Debattista, R. Gillibrand and A. Chalmers. Analytic anti-aliasing for selective high fidelity rendering. In Proceedings of *SIBGRAP I, IEE Computer Society Press*, pp. 359-366, 2005.
- [24] P. Longhurst, K. Debattista and A. Chalmers. A GPU based saliency map for high-fidelity selective rendering. In Proceedings of *AFRIGRAPH 2006*, ACM SIGGRAPH, pp. 21-29, 2006.
- [25] D. Luebke, B. Watson, J. D. Cohen, M. Reddy and A. Varshney. Level of Detail for 3D Graphics, Elsevier Science Inc., 2002.
- [26] A. Mack and I. Rock. Inattentional Blindness. MIT Press, 1998.
- [27] D. W. Massaro and D. S. Warner. Dividing attention between auditory and visual perception. *Perception & Psychophysics* 21, pp. 569-574, 1977.
- [28] G. Mastoropoulou, K. Debattista, A. Chalmers and T. Troscianko. Auditory bias of visual attention for perceptually-guided selective rendering of animations. In Proceedings of *GRAPHITE 2005*, ACM SIGGRAPH, 2005.
- [29] G. Mastoropoulou, K. Debattista, A. Chalmers and T. Troscianko. The influence of sound effects on the perceived smoothness of rendered animations. *APGV 2005: Second Symposium on Applied Perception in Graphics and Visualization*, ACM SIGGRAPH, 2005.
- [30] D. P. Mitchell. Generating antialiased images at low sampling densities. In *SIGGRAPH'87*, ACM Press, 65-72, 1987.
- [31] K. Myszkowski. The Visible Differences Predictor: Applications to global illumination problems. In *Eurographics Workshop on Rendering*, pp. 223-236, 1998.
- [32] C. O' Sullivan, S. Howlett, R. McDonnell, Y. Morvan and K. O' Conor. Perceptually adaptive graphics. In *Eurographics State of the Art Reports*, 2004.
- [33] J. Painter and K. Sloan. Antialiased ray tracing by adaptive progressive re-ement. In *SIGGRAPH'89*, ACM Press, pp. 281-288, 1989.
- [34] J. Prikryl and W. Purgathofer. Overview of perceptually-driven radiosity methods. Tech. Rep. TR-186-2-99-26, Vienna, Austria, 1999.
- [35] M. Ramasubramanian, S. Pattanaik and D. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *SIGGRAPH'99*, ACM Press/Addison-Wesley Publishing Co., pp. 73-82, 1999.
- [36] K. Robson Brown, A. Chalmers, T. Saigol, C. Green and F. D'Errico. An automated laser scan survey of the Upper Palaeolithic rock shelter of Cap Blanc. *Journal of Archaeological Science*, vol. 28, pp. 283-289, 2001.
- [37] P. Shirley. A ray tracing method for illumination calculation in diffuse-specular scenes. In Proceedings of *Graphics Interface '90*, Canadian Information Processing Society, Toronto, Ontario, 205-12, 1990.
- [38] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. In *SIGGRAPH'89*, ACM Press, pp. 335-344, 1989.
- [39] P. Slusallek, M. Stamminger, W. Heidrich, J. C. Popp and H. P. Seidel. Composite lighting simulations with lighting network. *IEEE Computer Graphics and Applications* 18, vol. 2, pp. 21-31, 1998.
- [40] W. A. Stokes, J. A. Ferwerda, B. Walter and D. P. Greenberg. Perceptual illumination components: a new approach to efficient, high quality global illumination rendering. *ACM Trans. Graph.* vol. 23, no. 3, pp. 742-749, 2004.
- [41] R. L. Storms. Auditory-Visual Cross-Modal Perception Phenomena. PhD thesis, Naval Postgraduate School, Monterey, California, 1998.
- [42] V. Sundstedt, A. Chalmers, K. Cater and K. Debattista. Top-down visual attention for efficient rendering of task related scenes. In *Vision, Modeling and Visualization*, 2004.
- [43] V. Sundstedt, D. Gutierrez, F. Gomez and A. Chalmers. Participating Media for High-Fidelity Cultural Heritage. *VAST 2005 - Symposium on Virtual Reality, Archaeology and Cultural Heritage*, Eurographics, 2005.
- [44] V. Sundstedt, K. Debattista, P. Longhurst, A. Chalmers and T. Troscianko. Visual attention for efficient high-fidelity graphics. In *Spring Conference on Computer Graphics (SCCG 2005)*.
- [45] J. R. Wallace, M. F. Cohen and D. P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. In *SIGGRAPH'87*, ACM Press, pp. 311-320, 1987.
- [46] G. J. Ward, F. M. Rubinstein and R. D. Clear. A ray tracing solution for diffuse interreflection. In *SIGGRAPH'88*, ACM Press, pp. 85-92, 1988.
- [47] G. J. Ward. The radiance lighting simulation and rendering system. In *SIGGRAPH'94*, ACM Press, pp. 459-472, 1994.
- [48] T. Whitted. An improved illumination model for shaded display. *Commun. ACM* 23, 6, pp. 343-349, 1980.
- [49] S. Winkler and C. Faller. Audiovisual quality evaluation of low-bitrate video. In *SPIE/IS&T Human Vision and Electronic Imaging*, vol. 5666, pp. 139-148, 2005.
- [50] H. Yee, S. Pattanaik and D. Greenberg. Spatiotemporal sensitivity and Visual Attention for efficient rendering of dynamic Environments. In *ACM Transactions on Computer Graphics*, vol. 20, pp. 39-65, 2001.



Alan Chalmers is a Professor at the Warwick Digital Laboratory, University of Warwick. He has an MSc with distinction from Rhodes University, 1985 and a PhD from University of Bristol, 1991. He has published over 130 papers in journals and international conferences on high-fidelity graphics and parallel rendering. He is Honorary President of Afrigraph and a former Vice President of ACM SIGGRAPH. His research goal is "Realism in Real-Time", obtaining physically-based realistic images at interactive rates through a combination of parallel processing and visual perception techniques.



Kurt Debattista is a Research Associate at the Warwick Digital Laboratory of the University of Warwick. He has recently obtained his PhD from the University of Bristol. He has a BSc in Mathematics and Computer Science and an MSc with distinction in Computer Science from the University of Malta, where he also worked as an assistant lecturer. His research interests are high-fidelity graphics, particularly selective rendering, parallel computing and systems software.



conferences.

Luis Paulo Peixoto dos Santos, is an Auxiliar Professor at the Departement of Computer Science, Universidade do Minho, Portugal, where he lectures Computer Architecture, Parallel Processing and Physically Based Lighting and Rendering. He got his PhD in 2001 entitled "Scheduling in Parallel Systems".. His main research interests include High Fidelity Interactive Rendering and Parallel Processing; having published about a dozen papers on specialized



Georgia Mastoropoulou works for the Ministry of Education in Greece. She has recently obtained her PhD from the University of Bristol. She has a BSc/MSc in Computer Science and Informatics from the University of Patras, Greece and an MSc with distinction in Advanced Computer Science from the University of Bristol. Her research interests are intersensory, and in particular auditory-visual, interactions for perceptually-adaptive techniques in computer graphics.