

Snapshot: A rapid technique for driving a selective global illumination renderer

Peter Longhurst
University of Bristol
Merchant Venturers Building
Woodland Road
BS8 1UB, Bristol, UK

Pete.longhurst@bristol.ac.uk

Kurt Debattista
University of Bristol
Merchant Venturers Building
Woodland Road
BS8 1UB, Bristol, UK

debattis@cs.bris.ac.uk

Alan Chalmers
University of Bristol
Merchant Venturers Building
Woodland Road
BS8 1UB, Bristol, UK

alan@cs.bris.ac.uk

ABSTRACT

Even with modern graphics hardware, it is still not possible to achieve high fidelity global illumination renderings of complex scenes in real time. However, as these images are produced for human observers, we may exploit the fact that not everything is perceived when viewing the scene with our eyes. We are drawn to certain salient areas of an image. Taking this into account, it is possible to selectively render parts of an image at high quality and the rest of the scene at lower quality without the user being aware of this difference.

Methods exist for calculating which parts of an image are perceptually important, but generally they rely on having a fully rendered image to process. It is thus only possible to prioritise pixels to speed up the rendering of a frame once that frame has been rendered: an obvious catch. In pre-scripted animated sequences it is indeed possible to use rendered key frames to extract the necessary information, however, the cost of rendering such key frames could be significant and this is not appropriate for any interactive application. This paper presents a high speed OpenGL generated “*Snapshot*” of a frame to generate a saliency map to efficiently drive the selective global illumination rendering of an animated sequence.

Keywords

High Fidelity Graphics, Human Visual Perception, Selective Rendering, Interactive Rendering of Dynamic Scenes.

1. INTRODUCTION

The aim of realistic image synthesis is to produce a high fidelity reproduction that faithfully represents the real scene it is attempting to portray. Such full global illumination solutions can be computationally very costly. Recently, traditional rendering algorithms have been modified in order to selectively spend more time rendering perceptually important pixels at the highest quality, while the remainder of the image, which is not seen by the human viewer, can be rendered at a significantly lower quality [Cater03, Yee01]. Such selective renderers have been

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG 2005 SHORT papers proceedings ISBN 80-903100-9-5
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

shown to significantly improve rendering times without the viewer being aware of the different qualities within the image. A major problem still remains however, and that is: how to rapidly identify the different qualities at which pixels should be rendered?

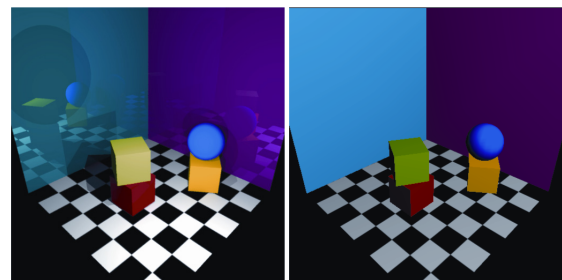


Figure 1: (a) Full global illumination image - 382 seconds (b) OpenGL Snapshot - 2 milliseconds.

The quality to which pixels in an image are to be rendered can be prioritised based on importance criteria. When considering ray tracing, the number of samples needed per pixel can be adjusted according to the saliency of that area of the scene. More rays

should be traced to achieve the desired quality in those salient areas where an individual's attention is likely to be focused. This measure requires knowing information on a per pixel basis, but until the scene is rendered this information is unknown.

An alternative to actually tracing the rays, is to develop methods to extract the necessary information from an interpretation of the scene geometry. OpenGL is a rapid approach that uses modern graphics hardware to draw 3D geometry.

Figure 1 shows a high speed OpenGL image of a scene, which we term a *Snapshot*, (achieved in 2ms) compared with full global illumination solution (382 seconds). This paper investigates how an indication of saliency can be achieved rapidly using such an OpenGL Snapshot.

2. PREVIOUS WORK

When viewing a scene, the human visual system will shift attention around the scene, selecting in turn the available visual information for localisation, identification and understanding of objects in the environment. During this process, more attention is given to salient locations, for example, a red apple in a green tree, and less attention to unimportant regions, so that detail in many parts of the scene can literally go unnoticed [Yarbus67]. Visual perception techniques are increasingly being used to improve the perceptual quality of rendered images, including [Bolin98, Luebke01, and Volevich00]

Previous selective rendering solutions using saliency have used the model of the human visual system proposed by Itti and Koch [Itti00, Itti98, and Koch85]. Some of these techniques have been developed for pre-composed animation sequences to determine areas of perceptual importance within key-framed sections, for example [Myskowski01].

Rendering time for animations has been reduced by saving computational effort on non-salient areas. This works well as salient areas for those frames between key-frames can be quickly interpolated, however, the key frames themselves have to be fully rendered. The drawback of this is that these methods cannot be used for interactive rendering scenarios.

Saliency was also used by Yee et al [Yee01] to accelerate animation renderings with global illumination. As with our approach, they use an initial OpenGL image and apply a model of visual attention to identify conspicuous regions. Yee et al. then construct, for each frame in the animation, a spatiotemporal error tolerance map called the Aleph map from spatiotemporal contrast sensitivity and a low-level saliency map. The Aleph map is then used as a guide to indicate where more rendering effort

should be spent, significantly improving the computational efficiency during the animation. However, such a map takes several seconds to compute.

3. THE SNAPSHOT

We chose OpenGL as a basis for our Snapshot because it is well supported in hardware, fast and cross platform. It is designed to read model data from a Wavefront “.obj” file, an established format which many renderers can read [Alias].

Shadows and Reflections

Figure 1 shows how the simple Snapshot does not contain any of the shadow or reflection information which is present in the scene. Significant potential salient information is thus missing from this simple image. To overcome this, we have included simple shadowing and reflection calculations in Snapshot, as can be seen in Figure 2.

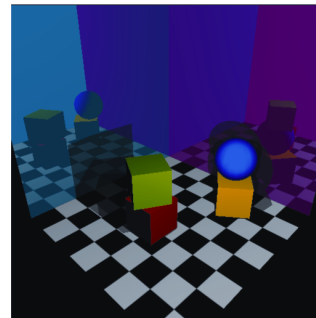


Figure 2: Snapshot with shadowing and reflections.

In OpenGL individual surfaces in a scene are drawn under an approximation of direct illumination. This shading is simply calculated on the vertex normals of a surface in relation to a lighting model, no account is taken for surface occlusions. In order to obtain shadows, manual calculations have to be undertaken to project occluding geometry onto surfaces. A standard method that we choose to do this involves observing the scene from the point of view of the light source. From this view surfaces to be shadowed can be masked off, the rest of the scene can then be traced as shadow into this area [Kilgard99]. The drawback of this technique is that the scene needs to be drawn multiple times depending on the number of objects and lights present. Similarly, reflections require the camera to be moved to a projected position and re-rendering the scene on the mirror plane. To cut the total number of drawings down, we chose to render shadows and reflections solely on large planar surfaces where they will probably be most apparent.

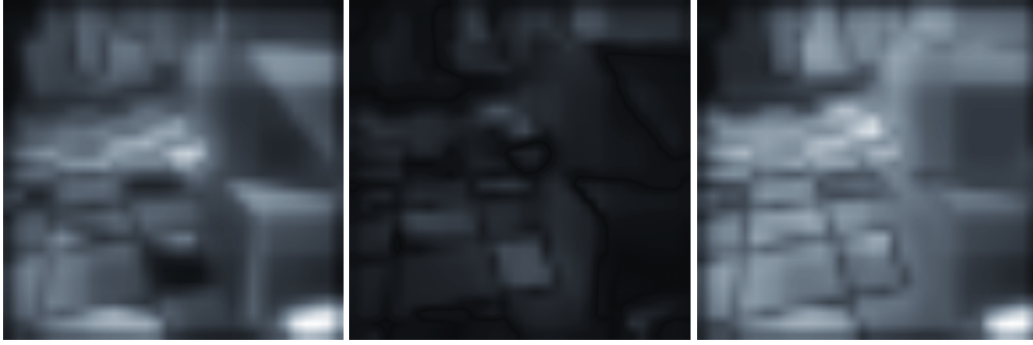


Figure 3: Radiance saliency (left) vs. Snapshot saliency (right) with difference map (centre). (frame #125)

Reflected Light Sources

The biggest problem with the aforementioned approximations is that there is no account taken for indirect illumination. In a scene with many reflective surfaces or mirrors a significant contribution to the lighting of an object will come indirectly from reflections of light. To account partially for this reflected light, sources are added to the scene as additional light sources with reduced emission. The reflected position of each light source given each mirrored surface is calculated. These extra lights are then positioned with their emission components reduced by the reflectivity of the mirror. Each additional light will simply directly illuminate each surface in the scene. This extra contribution is proportionally greatest on surfaces which face none of the primary light sources.

4. SALIENCY COMPARISONS

In order to investigate the potential use of Snapshot in selective rendering we determined the saliency of the frames. For this we used the Itti and Koch method implemented in the iLab toolkit [Ilab]. We generated a saliency map for every Snapshot frame and for each Radiance rendered image. The absolute pixel difference was found and averaged across the image. This gave a measure of saliency difference between the images. Figure 3 shows the error for one frame of the animation. In these images the brightest areas represent the areas of greatest saliency. The Radiance image shows how, when properly calculated, indirect illumination adds salient information. Figure 4 shows how the error changed through the animation. The lower line on the graph demonstrates that adding shadows and reflections decreases this error. For the most part, the lines follow a similar path; however, for the complex part of animation it is clear that the Snapshot without shadows and reflections produces a significantly worse result.

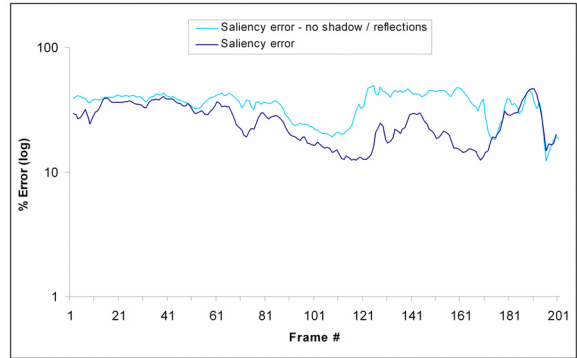


Figure 4: Saliency error during course of animation.

Table 1 shows numerically the saliency error for our two example frames. As previously shown in Figure 4, the error in saliency is always less when approximated shadows and reflections are added. In areas consisting of many such artefacts this difference is maximised, for the example complex frame the error almost halves. When the entire animation is considered this difference is not so great, but it is still significant. When there are no shadows or reflections the average error is 35%, this drops to 26% when these are added. A standard statistical analysis shows the difference of these means to be highly significant (the probability that the sets are then same < 0.001).

	<i>Error</i>
SF: no Shadows/Reflections/Texture	43%
SF: Textures	39%
SF: Textures + Shadows	37%
CF: no Shadows/Reflections/Texture	48%
CF: Textures	37%
CF: Textures + Shadows + Reflections	19%
Animation Average	26%
Animation Avg no shadows/reflections	35%

Table 1: Percentage error: Global illumination vs. Snapshot saliency.

Comparison of Saliency Map Prioritisation Based Rendering

To check the validity of using a saliency map based on our scene estimation to generate a full global illumination render we used our selective renderer, SharpEye, which allows us to manipulate the number of rays traced per pixel. For this experiment, we computed a reference image in which every pixel was dictated by the saliency map shooting a maximum of 9 rays per pixel. Further images were based on the generated saliency maps: from the full global illumination solution, the simple Snapshot and the more complex Snapshot.

The full global illumination solution took over 7 minutes to compute, whereas the selective rendered image using the saliency map generated from the full solution took only 2 minutes 11 seconds. Of particular interest is the selective rendering based on the saliency maps for the simple Snapshot took 2 minutes and 33 seconds while the Snapshot with shadows and reflections took 2 minutes 6 seconds, even faster than saliency map from the full solution. The VDP perceptual error (for the frame considered) for all selective renderings is less than 0.5% [Daly93].

5. CONCLUSIONS

Selectively rendering a global illumination computation can significantly reduce the time taken to render a scene without affecting the perceived quality of the resultant image. In the tests conducted in this paper, we have found that, in many cases, a simple OpenGL Snapshot does closely match an equivalent global illumination image in terms of saliency. The addition of approximate shadows and reflections to the Snapshot, although adding to the Snapshot computation time (from 2ms to 14ms), significantly increased the saliency correspondence between the Snapshot and the full global illumination solution. This Snapshot was then able to successfully drive the selective renderer, achieving a high perceptual fidelity between the selective rendered images and the full solutions.

Future work will use eye-tracking to further verify the possibility of using Snapshot as a means of driving a selective global illumination renderer. By examining these eyetracking results in conjunction with the saliency maps we hope to establish where the most significant flaws are in using OpenGL as a fast preliminary renderer. We will also consider more complicated scenes.

6. REFERENCES

- [Alias] Alias wavefront: <http://www.aw.sgi.com/>.
- [Bolin98] M. R. Bolin and G. W. Meyer. A perceptually based adaptive sampling algorithm. *Computer Graphics*, 32(Annual Conference Series):299-309, 1998.
- [Cater03] K. Cater, A. Chalmers, and G. Ward. Detail to attention: Exploiting visual tasks for selective rendering. *Eurographics Rendering Symposium*, (Annual Conference Series), 270-280, 2003.
- [Daly93] S. Daly. The visible differences predictor: An algorithm for the assessment of image fidelity. In A. B. Watson, editor, *Digital Images and Human Vision*, pages 179-206, Cambridge, MA, 1993. MIT Press.
- [Haines86] E. A. Haines and D. P. Greenberg. The light buffer: a shadow testing accelerator. *IEEE Computer Graphics and Applications*, 6(9):6-16, 1986.
- [Ilab] ilab neuromorphic vision c++ toolkit: <http://ilab.usc.edu/toolkit/>.
- [Itti00] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 10(10-12):1489-1506, 2000.
- [Itti98] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. In *Pattern Analysis and Machine Intelligence*, volume 20, pages 1254-1259, 1998.
- [Kilgard99] M. Kilgard. Creating reflections and shadows using stencil buffers. In *GDC 99*, 1999.
- [Koch85] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Human Neurobiology*, volume 4, pages 219-227, 1985.
- [Lee85] M. E. Lee, R. A. Redner, and S. P. Uzelton. Statistically optimized sampling for distributed ray tracing. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 61-68. ACM Press, 1985.
- [Luebke01] D. Luebke and B. Hallen. Perceptually driven simplification for interactive rendering. *Rendering Techniques*, 2001.
- [Myskowski01] K. Myskowski, T. Tawara, H. Akamine, and H. Seidel. Perception-guided global illumination solution for animation rendering. *SIGGRAPH 2001 Conference Proceedings*, pages 221-230, 2001.
- [Volevich00] V. Volevich, K. Myskowski, A. Khodulev, and E. Kopylov. Measuring and using the visual differences predictor to improve performance of progressive global illumination computation. In *Transactions on Graphics*, 19:122-161, 2000.
- [Ward98] G. Ward and R. A. Shakespeare. *Rendering with Radiance*. Morgan Kaufmann Publishers, 1998.
- [Weghorst84] H. Weghorst, G. Hooper, and D. P. Greenberg. Improved computational methods for ray tracing. In *ACM Transactions on Graphics*, pages 52-69. ACM Press, 1984.
- [Yarbus67] A. Yarbus. Eye movements during perception of complex objects. In *Eye Movements and Vision*, pages 171-196, 1967.
- [Yee01] H. Yee, S. Pattanaik, and D. P. Greenberg. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. In *ACM Transactions on Graphics*, pages 39-65. ACM Press, 2001.