Topic 8: Global illumination: Can it be achieved in real-time?

Reg. 4759265

December 1, 2013

Abstract

Rendering highly realistic computer generated imaging can be achieved using techniques of global illumination. These techniques emerged as early as the 1980's and have traditionally been used for offline rendering. However recent developments in multi-core hardware has seen parallel implementations become popular, allowing for possible real-time applications. This report describes the different methods and techniques of implementing global illumination. The techniques such as rasterisation, ray tracing, path tracing and radiosity are compared and the real-time possibilities explored.

Contents

1 Introduction

1.1 Background and Key Issues

Global illumination is a method of rendering realistic computer generating images by simulating light transport between surfaces. Global illumination techniques attempt to model the effects of light in a scene interacting with the objects in the environment to produce realistic lighting. Traditional 3D rendering techniques such as immediate mode rasterisation are only capable of local illumination. Local illumination has no knowledge of the scene as a whole and can only account for rendering light which comes directly from a light source called 'Direct Illumination', (Robart et al., 2008). Global illumination renders direct illumination but also subsequent interactions of the light rays with the scene called 'Indirect Illumination'. Therefore global illumination techniques allow for more realistic scenes that have effects such as dynamic shadows, reflections and refraction, (Mortensen et al., 2008).

Many of these techniques have been around for many years, such as Ray Tracing which was first proposed by Whitted (1980). Global illumination techniques are traditionally used for offline rendering since real-time applications are difficult to achieve due to the large amount of lighting calculations required. Hence real-time applications of global illumination have been limited with the exception of expensive hardware. Now with recent developments in hardware capacities such as improved graphics processing units (GPU's) it is becoming possible to achieve these higher quality graphics in real-time through multi-core parallel implementations.

This report aims to detail and compare the different techniques of global illuminations. The objective is to explain the rasterisation approach and outline various common methods of global illumination such as Ray Tracing, Path Tracing and Radiosity in 2. The details of each technique are described with their advantages/disadvantages and then compared in 3. This report also aims at exploring existing or possible real-time implementations in 4.

2 Methodologies

2.1 Rasterisation

Rasterisation is the traditional rendering method used for most graphical rendering. This process takes the 3D vector geometry of a scene and converts them into pixels to display on a screen. Rasterisation occurs within the graphics pipeline (Figure 1) and is typically carried out by fixed functionally hardware of the GPU. Many graphical standards such as OpenGL and Direct 3D have been built upon the rasterisation technique and have become very popular "advancing in performance, flexibility, and programmability" over the years (Akeley et al., 2002). Rasterisation represents the fastest way of rendering because of its simple direct forward mapping from object data to screen pixels easily rendering in real-time applications.

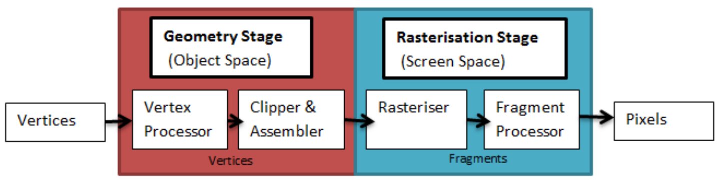

Figure 1: Diagram of the graphics pipeline (Yang et al., 2009)

This mapping is done by a number of steps. First the geometry in the scene is transformed from 3D into the 2D image plane using a projection such as orthographic or perspective. Then the geometry is clipped against the viewing window removing geometry outside of view. Finally in traditional rasterisation, scan conversion is applied to fill in the 2D triangles. It works by finding out which of the pixels are inside a triangle based on the calculation of edge function values. When considered inside the triangle attributes like colour, texture coordinates, direct lighting and shading are applied by interpolating them from the corresponding vertex data (Chang, 2011).

However rasterisation does not achieve global illumination. The approach has no knowledge of the scene as a whole making it impossible to compute indirect methods such as real reflection or refraction (Robart et al., 2008). Therefore the local illumination model is always used, though lots of techniques have been incorporated for increased realism (Akeley et al., 2002).

Yang et al. (2009) describes techniques used to create visual effects of global illumination in traditional rendering. The global illumination effects are calculated in an offline process and the values stored in texture maps called light maps. This process of texture baking together with shadow and screen space methods bring indirect illumination effects into the rendering pipeline in real-time.

2.2 Ray Tracing

Ray tracing is a technique used in generating computer graphic images pioneered by Whitted (1980) which is based upon ray casting by Appel (1968). This method is a per-pixel approach that works similar to the pinhole camera model involving ray casting into each of the pixels of the image. In reality light rays originate from a light source and bounce off surfaces to reach the eye. In ray tracing the rays are traced backwards from a camera position point through the two-dimensional array of pixels into a scene of geometry and lights. Each ray is then tested to see if it intersects with objects in the scene. If an intersection of the ray and geometry occurs, the nearest object is identified and lighting calculations are applied to the point of intersection. These lighting calculations can apply indirect illumination techniques such as reflection and refraction as the scene as a whole can be used with subsequent rays, therefore achieving global illumination. However full global illumination techniques such as "colour bleeding" and inter diffuse reflection are not part of the standard approach.

The process of ray tracing can be seen in Figure 2. To produce the most realistic images ray tracing involves the combination of many lighting techniques together with the material properties of objects. To compute the final pixel colour for each ray the ambient, diffuse and specular values are added up together with returned pixel values of recursive reflected and refracted rays. This uses the same method commonly used in rasterisation called the Phong illumination lighting method (See Equation 1), which can be applied per pixel to get the same direct illumination lighting. The resulting pixel colours of the reflected and refracted rays are returned after a set number of bounces to avoid getting stuck in loops.

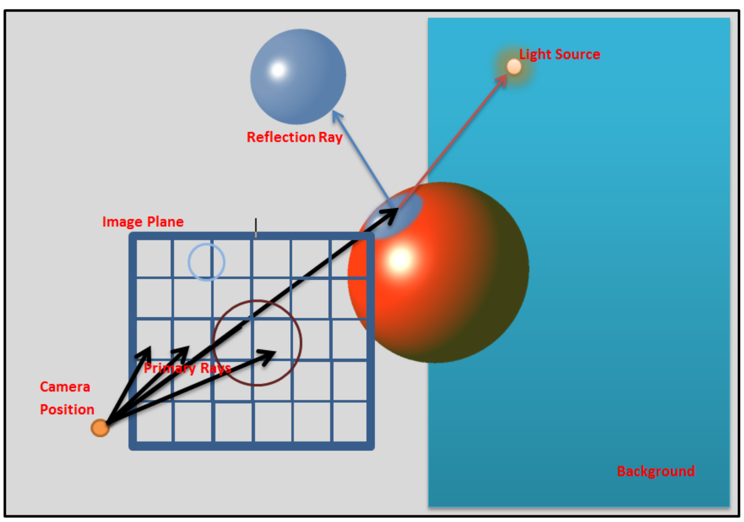

Figure 2: Diagram of the basic ray casting approach

$$Pixel_{RGB} = Ambient_{RGB} + Diffuse_{RGB} + Specular_{RGB} + RayTrace(ReflectedRay)_{RGB} + RayTrace(RefractedRay)_{RGB} \quad (1)$$

2.3 Path Tracing

Path tracing is another rendering method used to achieve global illumination. Path tracing builds upon other methods such as conventional ray tracing or scan-line rendering. It is able to do the physically correct computation of global illumination. The technique works by tracing paths in all directions at a point of intersection usually using a hemisphere approach. However to compute all possible directions is practically impossible, so Monte Carlo based algorithms are used. The Monte Carlo technique is used to randomly sample paths and the results of the sampling of these paths are used as an approximation. The resulting image can be noisy due to the random sampling; however it becomes more accurate in proportion to the square root of samples taken (Xu et al., 2010). Good results can be achieved with many samples but the result will be very slow.

2.4 Radiosity

Radiosity is an alternative global illumination algorithm for rendering computer graphics for producing realistic images. It is a technique that models the light interaction between diffusely reflecting surfaces derived from the field of heat transfer (Goral et al., 1984). Radiosity is calculated independent of the camera and instead is surface oriented with lights represented as surfaces, therefore all surfaces can potentially emit light. The method assumes that all surfaces are ideal diffuse reflectors, accounting for the indirect illumination.

$$
\begin{aligned}
B_j &= E_j + P_j \sum_{i=1}^{N} B_i F_{ij} \\
\text{where } B_j &= \text{Radiosity of surface j} \\
E_j &= \text{Direct energy emission rate of surface j} \\
P_j &= \text{Reflectivity of surface j} \\
B_i &= \text{Radiosity of other surface i} \\
F_{ij} &= \text{Form factor (fraction of energy leaving surface i and arriving at surface j)}
\end{aligned} \tag{2}
$$

The computation of the Radiosity consists in solving the Radiosity equation shown in equation 2. This equation describes the amount of energy emitted or reflected from a surface as the sum of the energy which strikes the surface and the energy in-built to the surface (light source). The equation is constructed from two concepts necessary for modelling the reflection, a form factor and the reflectance of a surface. The form factor between surfaces describes the physical relationship between them as "the fraction of the radiant light energy leaving one particular surface which strikes a second surface" (Goral et al., 1984). This links incident surfaces to any other surfaces in the scene that potentially emit or reflect light. The reflectivity represents the percentage of light energy that strikes the surface will be absorbed. To get the final image the scene is divided into surface patches and the form factors for all are calculated. This process is slow as for all pairs of patches it takes N^2 for N patches, where N is the number of elements (Robart et al., 2008). The final image is then generated with the computed radiosities of the equation.

3 Technique Comparison

The algorithms and techniques described in Section 2 each attempt to render realistic computer graphic images through different ways trying to solve the rendering equation. Each method has differing pros and cons relating to the quality and speed of rendering resulting images which is examined in Table 1.

Table 1: Comparing the advantages and disadvantages of various rendering techniques

Rendering	**Advantages**	**Disadvantages**
Rasterisation	Rasterisation is the most commonly used method of rendering with hardware GPU's specifically designed for the its process in the graphics pipeline. It is a fast method of rendering which easily achieves real-time performance.	Rasterisation has no global illumination support and is incapable of indirect illumination without workarounds.
Ray Tracing	Ray tracing naturally allows for direct illumination methods such as Reflections, Refractions and Shadows to be implemented. It is a realistic simulation of lighting. The algorithm can be easily parallelised using multiple processing units.	Ray tracing is not necessarily photo-realistic and does not model inter-diffuse reflection, therefore unable to fully implement the rendering equation (Robart et al., 2008). Hard Shadows are produced from the direct cut off of light vectors from exact positions of lights. Performance is slow because calculations must be performed for every screen pixel. Aliasing is a problem due to tracing a finite number of rays through a discrete pixel grid.

Path Tracing	Path tracing includes the same benefits of the algorithm being used such as ray tracing. However path tracing allows for inter-diffuse reflections which enables effects such as soft shadows, ambient occlusion and colour bleeding.	Impossible to implement rays in all directions so samples must be used which can create noisy images. It takes some time to sample enough times to produce good quality images. Additionally some light effects such as caustics from point lights are difficult or near impossible to simulate.
Radiosity	Radiosity has realistic shading and is able to better capture the 'global ambience' of an indoor scene, accounting for the interaction between diffuse surfaces for colour bleeding, soft shadows and ambient occlusion. The process is viewer independent and is possible to be pre-computed for the environment.	Radiosity has no specular component due to only using diffuse inter-reflection and the calculation of the Radiosity equation for all patches is very slow. It is also unable to represent sudden changes in visibility such as hard shadows.

Rasterisation is the best performing algorithm but lacks the global illumination of the other algorithms without workarounds or hacks. Path tracing is the closest to simulating the rendering equation giving realistic rendering, however it is also the slowest to create a good quality result. The algorithms increase in global illumination quality descending Table 1, however the computation to generate each image generally increases. There are also many solutions to the disadvantages of these algorithms, many combining the above methods to get higher quality results. Ways to improve the computation times and real-time implementations are explained in Section 4.

4 Real-Time Implementations of Global Illumination

Rasterisation is widely used in real-time for computer graphics, however it does not provide global illumination without additional techniques and workarounds. It can be slow with these additions, however graphics hardware is made to accelerate it. “Rasterization excels at splicing many different algorithms together (that’s a strength, not a weakness) to achieve a desired result that may not match physical reality” (Akeley et al., 2002). Akeley et al. (2002) states that rasterisation is good at producing real time results, also able to produce reference locality, good cache behaviour and best at tolerating long latencies from complicated graphics models. Yang et al. (2009) shows how rasterisation together with shader methods are widely used in the computer games industry for real-time rendering.

Ray tracing has been used in computer graphics for many years. However it is more commonly used in offline rendering where a high quality image is generated, but takes a long time, typically several hours to produce. For example ray tracing used in modelling packages such as 3DS Max (V-Ray) and Maya (Mental Ray) when rendering an image of the scene. Until recently real-time applications of ray tracing has yet to become possible, with the exception for large expensive hardware. This is due to the slow nature of the ray tracing calculation, performing the complex lighting equation for every pixel of the screen and checking for intersections with all objects in the scene.

Recent progression with GPUs means they have become increasingly programmable and multi core. GPU’s “will most likely support some form of ray-tracing in the near future” states Akeley et al. (2002). Ray tracing is easily accelerated with the GPU due to the independent nature of the rays cast. Each ray can be calculated in parallel using the many cores of the GPU to greatly speed up the computation time. Additionally the application of Spatial Partitioning can be applied to reduce the computation required for intersection calculations. Spatial Partitioning works by grouping multiple objects together into Bounding Volume Hierarchies (BVH), so that only a few regions of the scene’s geometry need to be intersection checked describes Glassner (1993).

Implementations of real-time ray tracing include work using parallel methods on the GPU’s of AMD and NVidia hardware. These hardware companies have produced API’s to allow computations to be performed on their graphics hardware called OpenCL (AMD) and CUDA (NVidia). OptiX (Parker et al., 2010) is NVidia’s implementation of ray tracing running in parallel with the CUDA API and optimised through accelerated data structures. Real-time ray tracing has also been performed in the computer games field with implementations of the Quake series and Wolfenstein: Ray Traced (Pohl, 2012).

Path tracing produces very realistic images, however the process is very computationally expensive. Since the process is based upon methods such as ray tracing it benefits from the same acceleration methods. Path tracing is traditionally used similar to ray tracing in producing highly realistic offline images. Many modelling software such as Blender contain path tracers built in. Path tracing is also often used to generate reference images for testing quality of the other rendering algorithms.

Real-time implementations of path tracing have similarly been developed to take advantage of GPU hardware. NVidia's OptiX also supports real-time rendering with the path tracing method. Bikker (2010) implemented a real-time path tracing application called Brigade for the application of gaming. However results can still be grainy due to the Monte Carlo sampling techniques often used.

Radiosity can be very computationally expensive when evaluating the Radiosity equation for all patches in a scene. The quadratic computation time means that the calculation is not traditionally performed in real-time. However Radiosity can be performed in real-time by first using a offline Radiosity approach (Yang et al., 2009). The equations can be pre-computed to generate a light map for the scene. This therefore reduces the computation at run-time to a simple texture mapping of patches. This means current desktop computers with standard graphics acceleration hardware can benefit from the lighting of Radiosity. Due to this Radiosity has been included in many computer games and can be used in modelling software such as 3DS Max.

5 Conclusion

Global illumination can produce realistic computer graphics images. The main global illumination algorithms have been described with their methodologies. Each have different complexities and lighting advantages. Some methods are more frequently used than others, but often these algorithms are used together to produce high quality results.

Rasterisation is increasingly being expanded with additions to provide global illumination effects. Ray tracing is becoming more plausible with the improved multi-core hardware available today. It will likely be used increasingly in rendering high quality real-time scenes and we will possibly see path tracing start to become used. However large scenes and high frame rates are still currently unable to be performed. Radiosity continues to be used for realistic rendering, however it will likely continue to be pre-computed for real-time applications.

Global illumination can therefore be performed in real-time, especially with increasing hardware capacities. Real-time global illumination methods will certainly become more common and possibly integrated into future hardware. There is certainly a lot of future work and research to be explored in this field of computing. Future work into additional areas of lighting and into improving the real-time performance are key to making global illumination more mainstream in computer graphics rending.

References

Akeley, K., Kirk, D., Seiler, L., Slusallek, P., and Grantham, B. (2002). When will Raytracing replace Rasterization? In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, SIGGRAPH '02, pages 86–87, New York, NY, USA. ACM.

Appel, A. (1968). Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, AFIPS '68 (Spring), pages 37–45, New York, NY, USA. ACM.

Bikker, J. (2010). Brigade: Real-time path tracing. `http://igad.nhtv.nl/~bikker/`.

Chang, Y.-N. (2011). An efficient hardware-software codesign of 3D rasterization module. In *Consumer Electronics - Berlin (ICCE-Berlin), 2011 IEEE International Conference on*, pages 15–16.

Glassner, A. (1993). *An Introduction To Ray Tracing*. Academic Press Limited.

Goral, C. M., Torrance, K. E., Greenberg, D. P., and Battaile, B. (1984). Modeling the interaction of light between diffuse surfaces. *SIGGRAPH Comput. Graph.*, 18(3):213–222.

Mortensen, J., Yu, I., Khanna, P., Tecchia, F., Spanlang, B., Marino, G., and Slater, M. (2008). Real-time global illumination for VR applications. *Computer Graphics and Applications, IEEE*, 28(6):56–64.

Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., and Stich, M. (2010). OptiX: a general purpose ray tracing engine. *ACM Trans. Graph.*, 29:66:1–66:13.

Pohl, D. (2012). Quake Wars gets ray traced. `http://software.intel.com/en-us/articles/quake-wars-gets-ray-traced/`.

Robart, M., Hill, S., and Tanguy, E. (2008). Global illumination for advanced computer graphics. In *Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on*, pages 1–2.

Whitted, T. (1980). An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349.

Xu, Q., Sbert, M., Feixas, M., and Scopigno, R. (2010). A new refinement criterion for adaptive sampling in path tracing. In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pages 1556–1561.

Yang, X., Yip, M., and Xu, X. (2009). Visual effects in computer games. *Computer*, 42(7):48–56.