Proceedings of the 13th International Symposium in the Field of Pulp, Paper, Graphics and Packaging, 19-22 June 2007, Zlatibor, Serbia
ISBN: 978-86-7401-242-0

PHOTOREALISTIC INTERACTIVE VISUALIZATION

Zoran G. Kotevski[1], Igor I. Nedelkovski[2], Tome I. Jolevski[3]
Faculty of Technical Sciences - Bitola, Ivo Lola Ribar b.b. 7000 Bitola, R. Macedonia

Abstract – When watching computer monitor or TV screen its obvious that two dimensional picture is seen. With some techniques of visualization this observing can give a feeling of watching three dimensional space. Generation (Rendering) of these two dimensional images requires complex mathematic models for presenting three dimensional objects in space. This paper gives analysis of rendering types and techniques and gives references for future development of software renderers and hardware acceleration, for photorealistic real-time visualization.

I. INTRODUCTION

Rendering is a process of generating an image from a model, by means of computer programs. It's the final process of creating the actual 2D image or animation, from the prepared scene. The model is a description of 3D objects in a strictly defined language or data structure. It contains geometry, viewpoint, texture, lighting and shading information.
Several different, and often specialized, rendering methods have been developed. These methods range from the distinctly non-realistic wireframe rendering, through polygon based, to advanced techniques like scanline rendering, radiocity or ray tracing. Rendering process can take from fragments of a second to days for a single frame, depending on the various factors like complexity of the scene, rendering technique, and hardware capabilities. In general different methods are better suited for either photorealistic rendering, or real-time rendering.
Rendering has uses in computer games, simulators, movies and design visualizations, each employing different balance of features and techniques. As a commercial products, wide variety of renderers are available. Some are integrated into larger modelling and animation packages, some are standalone and some are open source projects.
In the case of 3D graphics, rendering may be done slowly as pre-rendering, or in real-time. Pre-rendering is a computationally intensive process that is typically used for movie creation, while real-time rendering is mostly used in 3D computer games and simulations, which rely on the use of graphics cards with 3D hardware accelerators.

II. RENDERING TECHNIQUES

A lot of research has been done in the field of rendering algorithms, and a number of different techniques are developed. Four of these techniques have emerged as more efficient for light transport modeling:

[1] Rasterization including Scanline Rendering
[2] Ray Casting
[3] Radiocity
[4] Ray Tracing

[1] Zoran G. Kotevski is with the Faculty of Technical Sciences, I.L.Ribar bb, 7000 Bitola, Macedonia, E-mail: zoran.kotevski@uklo.edu.mk
[2] Igor I. Nedelkovski is with the Faculty of Technical Sciences, I.L.Ribar bb, 7000 Bitola, Macedonia, E-mail: igor.nedelkovski@uklo.edu.mk
[3] Tome I. Jolevski is with the Faculty of Technical Sciences, I.L.Ribar bb, 7000 Bitola, Macedonia, E-mail: tome.jolevski@uklo.edu.mk

Each of these techniques employ different algorithms for producing the final picture quality, with different performance. As a rule, techniques for generating photorealistic quality require large amount of computing time. Up to this date, photorealistic rendering in real time is only achieved on super computers, or clusters of computers.

Rasterization and Scanline Rendering

3D rendering works with primitives. In 3D world, triangles and polygons in space are considered to be primitives. If a pixel-by-pixel approach to rendering is impractical or too slow for some tasks, then a primitive-by-primitive approach to rendering may prove useful. One loop through each of the primitives, determines which pixels in the image it affects, and modifies those pixels accordingly. This is called rasterization, and is the rendering method used by most current graphics cards. Rasterization is frequently faster than pixel-by-pixel rendering, and is usually the approach of choice when interactive rendering is required. However, the pixel-by-pixel approach can often produce higher-quality images and is more versatile because it does not depend on as many assumptions about the image as rasterization. Scanline rendering is an algorithm for visible surface determination, in 3D computer graphics, that works on a row-by-row basis rather than a polygon-by-polygon or pixel-by-pixel basis. Scanline rendering as well as rasterization give fast performance but don't offer much image photorealism. Because of their performance these techniques are widely used by most graphics cards for real time rendering in computer games.

Ray Casting

Ray-casting is a technique that transforms a limited form of data into a 3D projection by tracing rays from the viewer's point of view. Ray casting is not a synonym for ray tracing, but can be thought of as an abridged, and significantly faster, version of the ray tracing algorithm. Both are image order algorithms used in computer graphics to render three dimensional scenes to two dimensional screens by following rays of light from the eye of the observer to a light source. Ray casting does not compute the new tangents a ray of light might take after intersecting a surface on its way from the eye to the source of light. This eliminates the possibility of accurately rendering reflections, refractions, or the natural fall off of shadows. However all of these elements can be faked to a degree, by creative use of texture maps or other methods. The high speed of calculation made ray casting a handy method for the rendering in many 3D video games.

Radiocity

Radiosity is a global illumination algorithm used in 3D computer graphics rendering. Radiosity is an application of the finite element method to solving the rendering equation for scenes with purely diffuse surfaces. Typical radiosity methods only account for paths which leave a light source and are reflected diffusely number of times before hitting the eye. As a rendering method, radiosity produces good photorealistic images.

a) b)

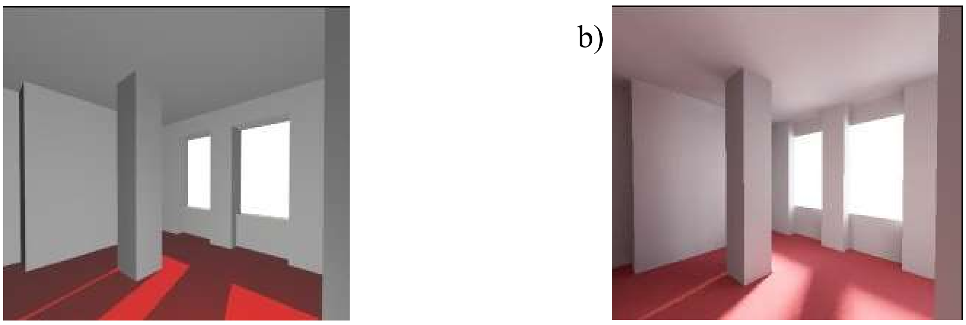


Fig. 1. Same scene rendered with direct illumination (a), and radiocity (b)

Ray Tracing

Ray Tracing is a specific rendering algorithm in 3D computer graphics. Ray Tracing produces mathematically modeled visualizations of programmed scenes using a technique that follows rays from the view point outward, rather than originating at the source of light. In comparisment with scanline rendering or ray casting, it produces similar results, but facilitates more advanced optical effects, such as accurate simulations of light reflection and refraction, and remains to be efficient enough to be of practical use when high quality output is required. Ray Tracing is the dominant method for rendering photorealistic scenes. Examples of typical Ray Tracers are POV-Ray and Ray Shade.

III. SOFTWARE RENDERING

Software rendering refers to a rendering process that is unaided by any specialized hardware, such as a Graphics Processing Unit. Software rendering takes place entirely on the CPU. Software rendering with the CPU has the main advantage that it is not restricted to the capabilities of graphics hardware. Early versions of computer games used software renderers for creating interactive visualization. But with the development of graphics processing hardware, hardware rendering almost completely took over, at least in the game programming industry. For real-time rendering the focus is on performance. With the development of graphics acceleration hardware, games and other applications started using hardware Application Programming Interfaces like DirectX[4] and OpenGL[5]

IV. HARDWARE ACCELERATED REALTIME RENDERING

A Graphics Processing Unit or GPU is a dedicated graphics rendering device for a personal computer, workstation, or game console. Modern GPUs are very efficient at manipulating and displaying computer graphics, and their highly parallel structure makes them more effective than typical Central Processing Unit (CPU) for a range of complex algorithms. A GPU implements number of graphics primitive operations in a way that makes running them much faster than drawing directly to the screen with the host CPU. Modern GPUs also have support for 3D computer graphics, and typically include digital video-related functions as well.

Recent developments in GPUs include support for programmable shaders which can manipulate vertices and textures with many of the same operations supported by CPUs.

V. FUTURE DEVELOPMENT

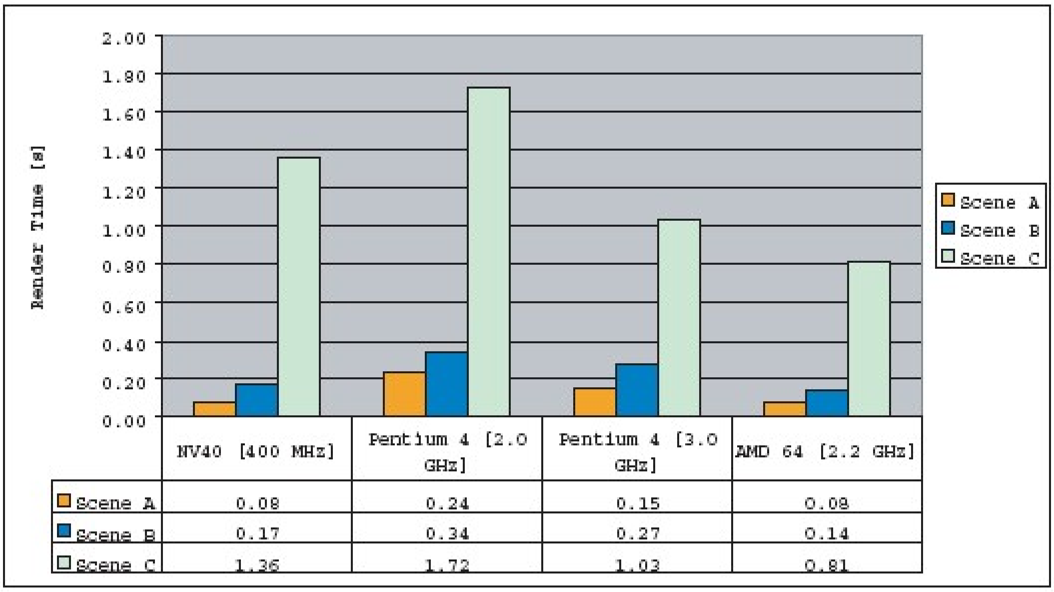
While offline global illumination algorithms like Ray Tracing or Radiocity are able to produce very realistic images, these effects are often absent in interactive applications. Recent researches seek to bridge the gap between realism and interactivity, and take complex light transport effects, like soft shadows, into account.

---

[4] Microsoft DirectX is a collection of APIs (Application Programming Interfaces) for handling tasks related to multimedia, especially game programming, on Microsoft platforms. One portion of it, Direct3D, competes against OpenGL. It is widely used in the development of computer games for Microsoft Windows. DirectX is also used among other software production industries, most notably among the engineering sector because of its ability to quickly render high-quality 3D graphics using the latest 3D graphics hardware.

[5] OpenGL (Open Graphics Library) is a standard specification defining a cross-language cross-platform API for writing applications that produce 3D and 2D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives. OpenGL is widely used in CAD, virtual reality, scientific visualization, information visualization, flight simulation and video game development.

The main area of research for photorealistic interactive visualizations is Real-Time Ray Tracing. There have been some efforts to develop Ray Tracing methods using GPUs, but as fig. 2 shows, current GPUs are not very efficient in accelerating Ray Tracing to Real-Time visualizations, for personal computers.



	NV40 [400 MHz]	Pentium 4 [2.0 GHz]	Pentium 4 [3.0 GHz]	AMD 64 [2.2 GHz]
Scene A	0.08	0.24	0.15	0.08
Scene B	0.17	0.34	0.27	0.14
Scene C	1.36	1.72	1.03	0.81

Fig. 2. Rendering times for different scenes - GPU vs. CPUs

One of the latest area of research is the development of the OpenRT project which includes highly optimized software core for Ray Tracing, along with an OpenGL like API, in order to offer an alternative for the current approach for interactive 3D graphics. The goal of the "OpenRT - Real Time Ray Tracing Project" is to develop ray tracing to the point where it offers an alternative to the current rasterization based approach for interactive 3D graphics. Therefore the project consists of several parts: a highly optimized ray tracing core, the OpenRT-API which is similar to OpenGL and many applications ranging from dynamically animated massive models and global illumination, via high quality prototype visualization to computer games.
Huge breakthrough in Ray Tracing announced Saarland University. That is experimental Ray Processing Unit, and it is designed to accelerate some of the intensive computations of Ray Tracing, up to 100 times.

VI. Conclusion

Rendering as a process, plays one of the main roles in the production of computer games, movies, animation and computer simulations. The key element of future rendering development is developing a technique for photorealistic real time rendering.
The first thing that can be concluded from the analysis of the rendering techniques is that the best photorealistic image quality can be achieved with the Ray Tracing technique.
Second thing is that the current GPUs still haven't got the power to calculate complex ray tracing algorithms in real time, especially when intensive scenes are rendered. Current GPUs don't have much RAM and there are hardware specific limitations when rendering high resolution meshes with a lot of textured objects. However, if the current trend of enhancing graphics cards continues the same way like past years, future generations of GPUs will make real time ray tracing in computer games possible.

In near future, can be expected to emerge programmable ray processing units that will make real-time rendering possible on personal computers

VII. References

http://en.wikipedia.org/wiki/3D_rendering
http://computer.howstuffworks.com/3dgraphics.htm
https://renderman.pixar.com/
http://usa.autodesk.com/adsk/servlet/index?id=6837573&siteID=123112
http://www.sve.man.ac.uk/mvc/Research/Previous/icam/whatis/
http://graphics.cs.ucf.edu/GPUassistedGI/index.php
http://www.gamedev.net/reference/programming/features/realtimerender/
http://en.wikipedia.org/wiki/Rendering_(computer_graphics)
http://www.realtimerendering.com/
http://www.opengl.org/about/overview/
http://www.lightworks-user.com/full_pr.php?prid=78
http://www.icem.com/downloads.asp?PageId=128
http://www-306.ibm.com/software/applications/plm/catiav5/prods/rt1/
http://www.microsoft.com/downloads/details.aspx?familyid=2da43d38-db71-4c1b-bc6a-9b6652cd92a3&displaylang=en
http://www.bitpipe.com/detail/PROD/1017065072_239.html
http://www.windowsmarketplace.com/details.aspx?itemid=2849272
http://www.csee.umbc.edu/~olano/s2005c37/proposal.html
http://www.zermatt.se
http://www.clockworkcoders.com/oglsl/rt/
http://web.cs.wpi.edu/~emmanuel/research/realtime_RT/
http://www.cgl.uwaterloo.ca/~gmoreno/streamray.html
Ray Tracing on GPU – Martin Christen, Diploma Thesis, University of Applied Sciences, Basel