DUKE SUMMER PROGRAM

Understanding
Real-Time Rendering

Xuan Liang

Math of Universe

Paper 1

July 25, 2017

Introduction

As we all know, there are always exquisite videos and images in video games. These images are what we called “CG”, or “computer graphics”. As the computer technology advances rapidly, the computer graphics in games also increases in both their quality and their size. However, since computer memory is not countless, how to hold as more contents as possible in limited capacity is a critical problem that game designers should concern. Real-time rendering, which is different from pre-rendering that were recordings of footage that was previously rendered on different equipment, can produce images or video in a real-time. [1] Rendering in real time not only has higher level of interactivity and flexibility, but also can effectively reduce the capacity that computer graphics need to occupy.

What is rendering

To understand real-time rendering, we should first know about rendering.

Rendering is the most technically complex aspect of 3D production, but it can actually be understood quite easily in the context of an analogy: Much like a film photographer must develop and print his photos before they can be displayed, computer graphics professionals are burdened a similar necessity.

When an artist is working on a 3D scene, the models he manipulates are actually a mathematical representation of points and surfaces (more specifically, vertices and polygons) in three-dimensional space.

The term rendering refers to the calculations performed by a 3D software package’srender engine to translate the scene from a mathematical approximation to a finalized 2D image. During the process, the entire scene’s spatial, textural, and lighting information are combined to determine the color value of each pixel in the flattened image. [2]

Rendering Equation

This is the key academic/theoretical concept in rendering. It serves as the most abstract formal expression of the non-perceptual aspect of rendering. All more complete algorithms can be seen as solutions to particular formulations of this equation.

$$L_{\text{o}}(\mathbf{x}, \omega_{\text{o}}, \lambda, t) = L_e(\mathbf{x}, \omega_{\text{o}}, \lambda, t) + \int_\Omega f_r(\mathbf{x}, \omega_{\text{i}}, \omega_{\text{o}}, \lambda, t)\, L_{\text{i}}(\mathbf{x}, \omega_{\text{i}}, \lambda, t)\,(\omega_{\text{i}} \cdot \mathbf{n})\,\mathrm{d}\,\omega_{\text{i}}$$

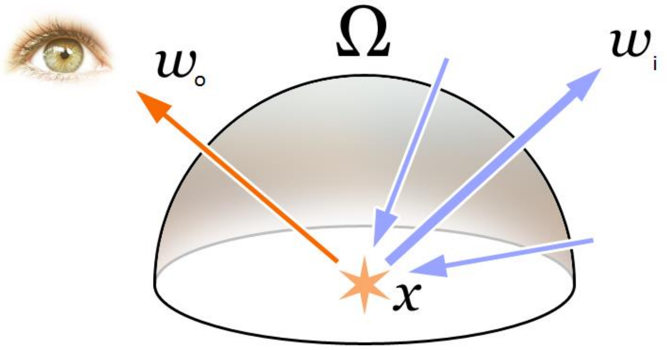

Figure I The rendering equation describes the total amount of light emitted from a point x along a particular viewing direction, given a function for incoming light and a BRDF.

Meaning: at a particular position and direction, the outgoing light (L_o) is the sum of the emitted light (L_e) and the reflected light. The reflected light being the sum of the incoming light (L_i) from all directions, multiplied by the surface reflection and incoming angle. By connecting outward light to inward light, via an interaction point, this equation stands for the whole 'light transport' — all the movement of light — in a scene. [3]

Rendering Techniques

There are three major computational techniques used for most rendering. Each has its own set of advantages and disadvantages, making all three viable options in certain situations.

- **Scanline (or rasterization):**
- Scanline rendering is used when speed is a necessity, which makes it the technique of choice for real-time rendering and interactive graphics. Instead of rendering an image pixel-by-pixel, scanline renderers compute on a polygon by polygon basis. Scanline techniques used in conjunction with precomputed (baked) lighting can achieve speeds of 60 frames per second or better on a high-end graphics card.

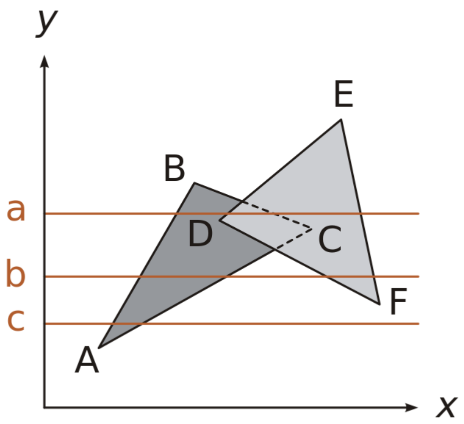

Figure 2 Scan-line algorithm

- **Raytracing:** In raytracing, for every pixel in the scene, one (or more) ray(s) of light are traced from the camera to the nearest 3D object. The light ray is then passed through a set number of "bounces", which can include reflection or refraction depending on the materials in the 3D scene. The color of each pixel is computed algorithmically based on the light ray's interaction with objects in its traced path. Raytracing is capable of greater photorealism than scanline but is exponentially slower.

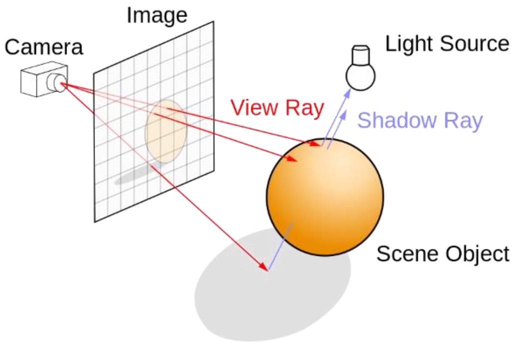

Figure 3 The ray tracing algorithm builds an image by extending rays into a scene.

- **Radiosity:**
- Unlike raytracing, radiosity is calculated independent of the camera, and is surface oriented rather than pixel-by-pixel. The primary function of radiosity is to more accurately simulate surface color by accounting for indirect illumination (bounced diffuse light). Radiosity is typically characterized by soft graduated shadows and color bleeding, where light from brightly colored objects "bleeds" onto nearby surfaces.

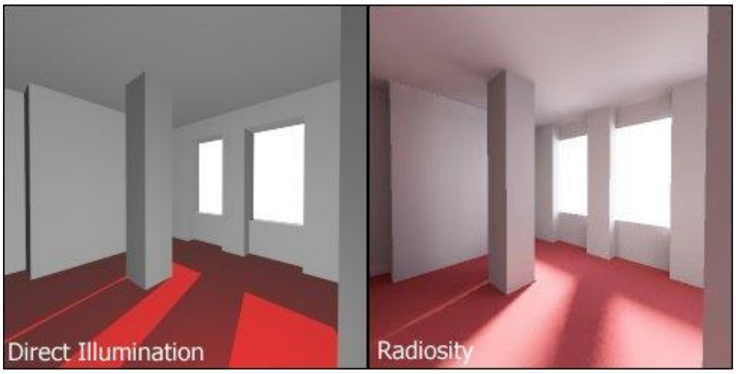

Figure 4 Difference between standard direct illumination without shadow umbra, and radiosity with shadow umbra

- In practice, radiosity and raytracing are often used in conjunction with one another, using the advantages of each system to achieve impressive levels of photorealism.

Rendering Software

Although rendering relies on incredibly sophisticated calculations, today's software provides easy to understand parameters that make it so an artist never needs to deal with the underlying mathematics. A render engine is included with every major 3D software suite, and most of them include material and lighting packages that make it possible to achieve stunning levels of photorealism.

The two most common render engines:

- **Mental Ray** – Packaged with Autodesk Maya. Mental Ray is incredibly versatile, relatively fast, and probably the most competent renderer for character images that need subsurface scattering. Mental ray uses a combination of raytracing and "global illumination" (radiosity).

Work of Mental Ray from Halo 4

- **V-Ray** – You typically see V-Ray used in conjunction with 3DS Max—together the pair is absolutely unrivaled for architectural visualization and environment rendering. Chief advantages of V-Ray over its competitor are its lighting tools and extensive materials library for arch-viz.

Work of V-Ray from CONAN EXILES

What is Real-Time Rendering

Real-time rendering is a graphics rendering technique — almost exclusive to video games — that helps load graphics. With real-time rendering, frames are loaded instantaneously to create images that simulate actual movement; this also allows video game makers to create interactive worlds. For the rendering to be considered real-time, it must be 15 frames per second (FPS) or faster. The main piece of hardware taxed is the video card, and some older video cards may be unable to keep up with such rendering. In contrast to real-time rendering is pre-rendering, which is sometimes used in video games.

When someone plays a three-dimensional (3D) game, it should appear to the player that his or her character is running through an environment or performing some action fluidly. In reality, there is nothing fluid about the movements; the video game is just loading images so quickly that it appears fluid, like in animation. Aside from creating movements that are more realistic, this has another advantage: making games interactive. Pre-rendered games, which were made during the early days of video games, were only interactive to a point; some objects could be moved, but the video game's world as a whole was static. Games that use real-time rendering can have very complex interactions.

Officially, real-time rendering speed must be 15 FPS or faster. If the graphics cannot load this quickly, then the user will notice obvious loading problems and the game will often lag. Aside from lagging, another problem that a player might notice is that certain images in the game, such as a character's outfit or background elements, will load very slowly.

To play videos games with real-time rendering, the computer must have a powerful video card, because this is the main hardware taxed during rendering. The video card is responsible for loading and generating all the graphics in a computer, and it must work very hard to ensure the rendering loads quickly. A computer with a weak video card may be unable to generate the images properly, or the rendering may slow down to below real-time speeds.

Properties of Real-Time Rendering

- **Comparing pre-rendering in frames:** On the quality in frames, pre-rendering is now better than real-time rendering. Also, pre-rendering often used global illumination, which leads to a fact that images produced by pre-rendering looks more authentic than images made by real-time rendering.
- **Computer Memory:** Real-time rendering does not occupy computer memory, and it is an indivisible part of the game. Wherever pre-rendering computer graphics need to exist individually in hard drive.

- **Interactivity:** Because it is impossible to predict exactly how a player will interact with the game environment, images must be rendered in "real-time" as the action unfolds.
- **Speed Matters:** In order for motion to appear fluid, a minimum of 18 - 20 frames per second must be rendered to the screen. Anything less than this and action will appear choppy.

- **The methods:** Real-time rendering is drastically improved by dedicated graphics hardware (GPUs), and by pre-compiling as much information as possible. A great deal of a game environment's lighting information is pre-computed and "baked" directly into the environment's texture files to improve render speed.

Conclusion

In the 1900s, nobody could imagine we can get news from all the world through a small box. Just like no one believed we could almost produce a real world in animation just 30 years ago. New technology has the ability to bring human mind into a whole new level. Maybe real-time rendering will also become useless in the future, but currently, I still believe that this technique will take the place of pre-rendering because of the appearance of Graphic Processing Unit (GPU) and Motion capture. Besides, Real-time rendering can be good strategy in movie world. Since real-time rendering can show videos to directors clearly, the process of editing and finalizing can be easier. Thus, what we can expect is that real-time rendering will break the boundary between video game world and Hollywood as well as change the world.

References

[1] “Pre-rendering” From Wikipedia, the free encyclopedia edited on 11 April 2016

[2] “What is Rendering?” by Justin Slick Updated February 04, 2017

[3] “Rendering (computer graphics)” From Wikipedia, the free encyclopedia edited on 2 July 2017

[4] “3D Rendering | What You Need to Know” by Dax Ayala Published on May 25, 2017

[5] “The Prospects of Real-time Rendering of Animation Technology” by Lu Yan Published on July 05,2012

[6] “what-is-real-time-rendering” Written by Alex Newth Last modified on 18 June 2017

[7] “, Real – Time Graphics Architectures” By Akeley, Kurt, and Pat Hanrahan Course CS448A Notes, Fall 2001

Figure 1 by Timrb from https://commons.wikimedia.org/wiki/File:Rendering_eq.png

Figure 2 by Vierge Marie from https://commons.wikimedia.org/wiki/File:Scan-line_algorithm.svg

Figure 3 by Henrik from https://commons.wikimedia.org/wiki/File:Rendering_eq.png

Figure 4 by Hugo Elias from https://commons.wikimedia.org/wiki/File:Radiosity_Comparison.jpg