Ray Tracing

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Nicholas Pochinkov, Aleksandr Prymek, Susanne O'Hare, Tadhg O Se Mc Knight

Supervised by Plamen Stamenov

Introduction & Theory

The Optics of Ray tracing

Light can be represented by rays; these rays of light undergo different phenomena every time they interact with an object. A ray can be reflected, refracted, or transmitted, depending on the type of object it interacts with. Ray tracing is computing the paths and intensities to render a graphic.

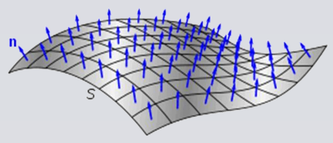


Fig 1.1 Chetvorno, (2019)

(Specular) Reflection of Light: (mirrors and most metals)

The laws of reflection (Young et al., 2015):
- Incident ray, reflected ray, and the normal lie in the same plane
- The angle of incidence is equal to the angle of reflection.

The more complicated the object, the harder it is to compute the surface normals and hence the reflected ray. Fig 1.1 shows normals on a simple surface.

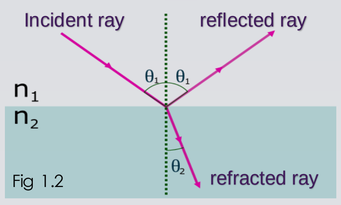


Fig 1.2

Snell's Law: $\frac{Sin\theta_1}{Sin\theta_2} = \frac{n_2}{n_1}$

Fig 1.2 - light ray incident on a surface that both reflects and transmits, showing both laws

Refraction of Light:

A light ray changes direction due to the change in the speed of light travelling through a transparent or translucent object.
the Laws of Refraction are (Young et al., 2015) :
- Incident ray, refracted ray, and the normal lie in the same plane,
- The change in direction can be given by Snell's Law;

Fresnel's Expressions:
(Woan, 2010)

$$R_s = \left|\frac{n_1 cos\theta_i - n_2 cos\theta_t}{n_1 cos\theta_i + n_2 cos\theta_t}\right|^2$$

$$R_p = \left|\frac{n_1 cos\theta_t - n_2 cos\theta_i}{n_1 cos\theta_t + n_2 cos\theta_i}\right|^2$$

$$T_s = 1 - R_s$$

$$T_p = 1 - R_p$$

Fresnel's Equations - (Schlick's approximation):

On hitting a medium, a ray is partially transmitted in (by refraction), and partially reflected. This is described by Fresnel's equations (Fig 1.3). In general, they are polarization-dependent. we almost always deal with unpolarized light, so we can use Schlick's Approximation to save a lot of CPU time.

Schlick's Approximation
(Schlick, 1994)

$$R = R_0 + (1 - R_0)(1 - Cos\theta)^5$$

$$T = 1 - R$$

Where, $R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2$

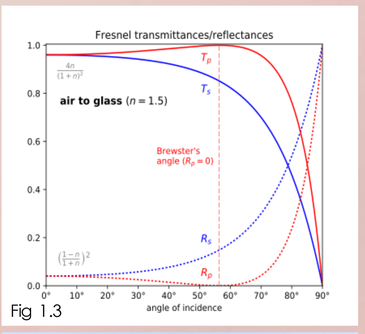


Fig 1.3

Specular and Diffuse reflection:

Diffuse reflection happens at many different angles, not just one. An ideal diffuse reflecting surface has equal luminance when viewed in all directions. Most objects have fine structure, such as cells, crystals or fibres, and, when light is incident on these parts, part of the light will be transmitted through and part will be reflected which promotes diffuse scattering. (Young et al., 2015)

Specular highlights (Phong,1975):

Surfaces (made up of microfacets) in general, are not completely smooth. Specular reflections on the microfacets will have slightly different normals to the flat surface, resulting blurred and spread out light. For white light that is evenly distributed about the normal, you see a bright spot on the object (Fig 1.5).

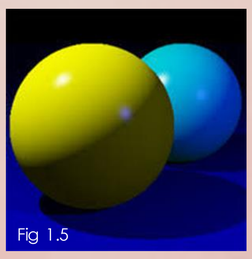


Fig 1.5

intensity of the highlight is given by (Phong reflection model):
$k_{spec} = (\hat{R} \cdot \hat{V})^n$, $\hat{R}$ - reflection, $\hat{V}$ - viewpoint vector, n – smoothness

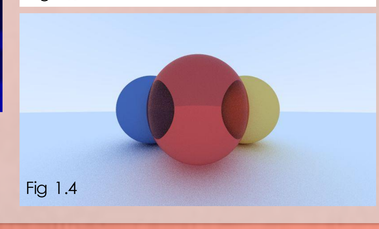


Fig 1.4

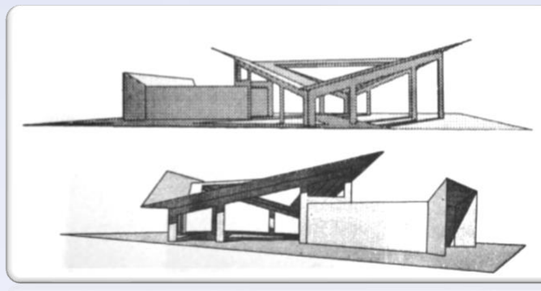
Timeline of History & Future

1532

The Idea of ray tracing has been around from as early as the 16th Century. The German painter and theorist Albert Durer is credited with its invention.

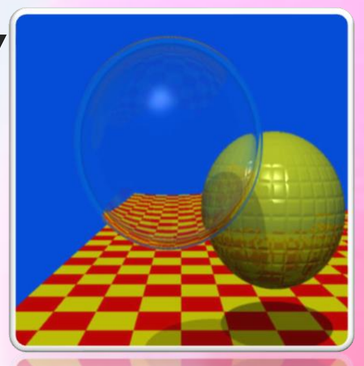
1969

Ray tracing was first developed by Appel [APPE68], this technique was first described by Arthur Appel in 1969 by a paper entitled "Some Techniques for Shading Machine Renderings of Solids", and by Goldstein and Nagel [MAGI68; GOLD71]. Appel used a sparse grid of rays to determine shading, including whether a point was in shadow. Goldstein and Nagel originally used their algorithm to simulate the trajectories of ballistic projectiles and nuclear particles and only later applied it to graphics. Appel ray traced shadows while Goldstein and Nagel evaluated Boolean set operations.



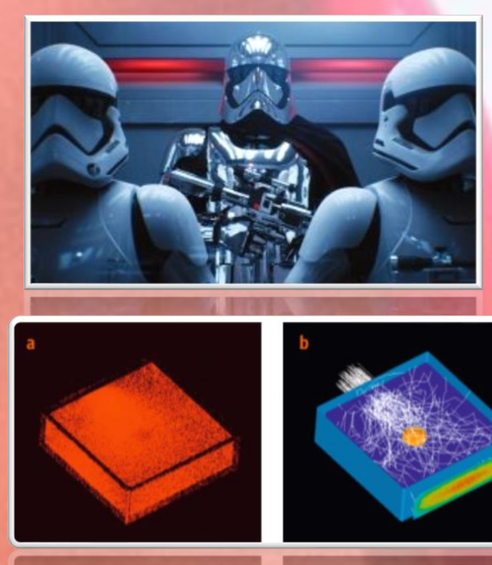
1980

Ray tracing was extended by Whitted [WHIT80] and Kay [KAY79a] to handle specular reflection and refraction. While previous algorithms determined the ray colour without recursively tracing more rays, Whitted's algorithm traced out the three rays of light formed after a light ray hits an object; the reflected ray, the refracted ray and the shadow. His algorithm allowed more realistic images to be made and played a huge role in the ongoing advancements of ray tracing.



2000s-Present Day

Whitted's paper was published in 1980 yet, due to the high computational cost, ray tracing did not become popular in film and television effects until the 1990s and early 2000s. Ray tracing also has applications in the medical world. Ray tracing software can assist and speed up the development of medical devices, such as those used in photodynamic therapy (a way to treat tumors using a photosynthezier), by simulating the propagation of light in biological tissue. A modern ray-tracing method, called Monte-Carlo ray tracing, can also generate high quality ultrasound images of embryos.



The above image (a) shows a model of biological tissue which contains a tumour, shown in yellow. Image b shows a ray-traced model, with ray paths, the irradiance distribution of the transmitted light is displayed on the detector surface. The commercial models, such as the model which created this ray tracing, called ASAP, has many advantages such as the comprehensive built in analysis tools.

Our own ray-tracer

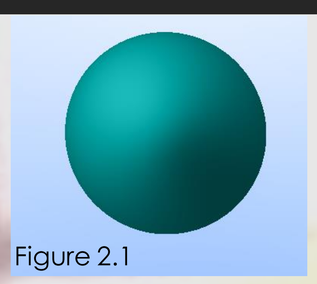


Figure 2.1

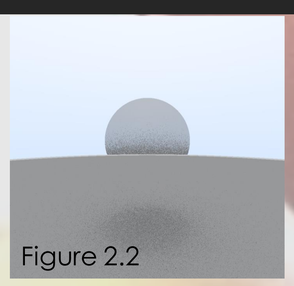


Figure 2.2

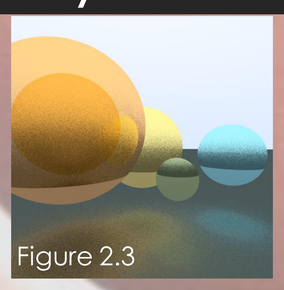


Figure 2.3

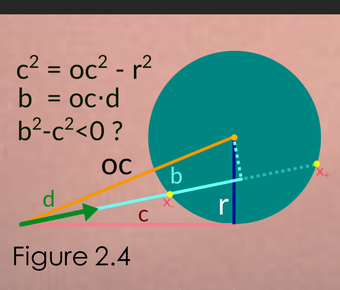


Figure 2.4

To gain a better understanding of the techniques used to ray-trace images, we developed a simple ray-tracing application in C++. Our application allows spheres, the simplest shape for ray tracing, and works by casting rays from the observer at the direction of the objects. If the ray hits, the ray bounces off and a recursive function is used to sees if it hits anything else. This happens until the ray no longer hits anything. To manipulate the lighting, one can redefine a function for the surrounding lighting, and for each of the objects we can change the reflation properties of the spheres. Fig 2.1 shows my first attempt, and Fig 2.2 & 2.3 Show how errors in the hit algorithm can give strange effects. The hit algorithm used in the end is shown in Fig 2.4, with extra code to find the normal and reflection.
The Code goes as such [Shirley, 2018]:

```
Create Spheres
For each pixel:
  Send a Ray (ideally multiple but not here)
    Check for each sphere if is a nearest hit
    If ray hits, send scattered reflected ray
      return color of the hit due to the ray
    else
      return the surrounding color
    Gamma Correct the color value
    Output pixel values (to .ppm)
```

Our final ray-traced image is shown to in Fig 2.5.

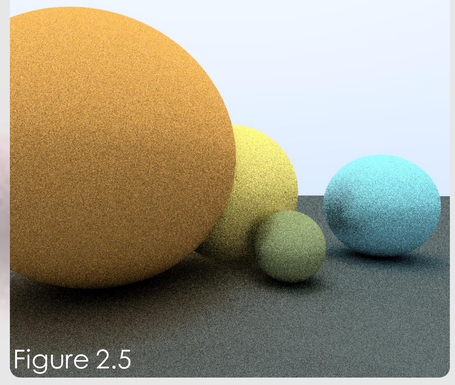


Figure 2.5

The future of Ray Tracing

The release of Turing architecture based-GeForce™RTX-20 series GPUs, with dedicated ray tracing processors, has made real-time ray tracing possible. Usually, ray tracing is only implemented in raster. Full vector implementation is only possible in less demanding games like Minecraft (Fig 4.1). With ray tracing support the announced for the new PS5 and Xbox Two, AMD is expected to release its own version of RTX. The Fig 4.2 and Fig 4.3 are from the RTX real-time ray tracing demo of Battlefield V.

As the PC hardware improves, more complicated rendering techniques will be usable in real-time, such as *photon mapping*, an extension of normal ray tracing which greatly enhances caustic & indirect illumination. Fig 4.4 shows caustic illumination using this.

The photon mapping process can be divided into three steps:
1) Photons are emitted from the light source and traced through the scene, and the information is stored where they intersect.
2) Photons are stored for quick access. (eg: using *kd*-trees or the faster [Gupte 2011]). Usually, two photon maps are created for a scene: a caustic one & a global one for soft indirect illumination.
3) Lastly, the image is rendered and specular reflections and incident illumination on the surfaces are calculated.

Another idea is utilizing ray tracing to improve the sound design as Mark Cerny suggested, by treating low-frequency sound waves (base)as rays it is possible to model them using light tracing principles. The main problem is that scale of sound wavelength can reach up to a couple of meters. However, exact ray-tracing for the low-frequencies is almost never necessary.

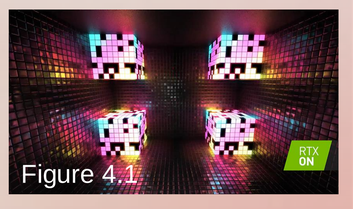


Figure 4.1

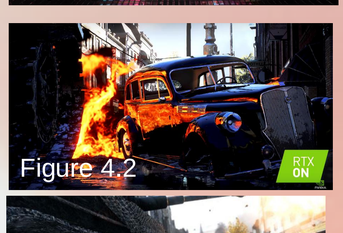


Figure 4.2

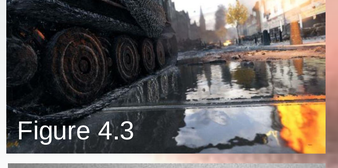


Figure 4.3

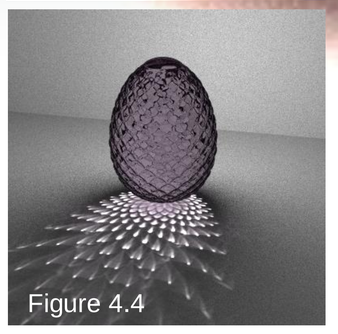


Figure 4.4

References

Wikipedia. 2019. Albrecht Durer. [ONLINE] Available at: https://en.wikipedia.org/wiki/Albrecht_Dürer. [Accessed 7 November 2019].
Foley, J., van Dam, A., Feiner, S. and Hughes, J., 1994. Introduction to Computer Graphics. 1st ed. California: Addison-Wesley Publishing Company.
Foley, J., 1997. Computer Graphics: Principles and Practice. 2nd ed. The United States of America: Addison and Wesley Publishing Company.
Breault. 2005. Ray Tracing in Medical Applications. [ONLINE] Available at: http://www.breault.com/sites/default/files/knowledge_base/ma_lp_001_raytracing_medicine_english.pdf. [Accessed 7 November 2019]
Mattausch, O., Goksel,O, 2016. Monte-Carlo Ray-Tracing for Realistic Interactive Ultrasound Simulation. Postgraduate. Switzerland: University of Zurich.
Polygon. 2019. Real-time ray tracing, and what it means for video games, explained. [ONLINE] Available at: https://www.polygon.com/e3/2019/6/10/18410733/ray-tracing-xbox-scarlett-ps5-amd-navi-nvidia-rtx. [Accessed 7 November 2019].
Young, H., Freedman, R., Ford, A., and Young, H. (2015). *University Physics.*
Woan, G. (2010). *The Cambridge handbook of physics formulas*. Cambridge University press.
Schlick, C. (1994). *An inexpensive BRDF Model for Physically-based Rendering*. CGF.
Phong, Bui Tuong (1975). *Illumination for computer generated pictures*. Communications of ACM 18
Peter Shirley (2018) Ray Tracing in One Weekend
Reedbeta, (2005). *English Wikipedia.*
Encyclopaedia Britannica, Inc, (2006)
Sbergjohansen, (2017). *Wikipedia*.
Chetvorno, (2019). *Wikipedia.*
Alan Bradley (2019) What is ray tracing, and is it really the future of gaming? GamesRadar+ https://www.gamesradar.com/what-is-ray-tracing-and-is-it-really-the-future-of-gaming/
Fig 4.1 https://www.nvidia.com/en-us/geforce/news/minecraft-rtx-trailer-screenshots-partnership/
Fig 4.2 https://ie.ign.com/articles/2019/10/10/what-is-ray-tracing-and-should-you-care
Shadow6ix, (2018), *Last Tiger* [ONLINE]. Available at: https://www.nvidia.com/en-gb/geforce/shot-with-geforce/?m=V3siaSI6IjI3MTQ4IiwicMKDIjAifQ%3D%3D [Accessed 7 November 2019].
Luke Wilimitis, (2019), *photon mapping i* [ONLINE]. Available at: https://wilimitis.github.io/portfolio/2019-01-16/ [Accessed 7 November 2019].

IOP Institute of Physics
Juno Practitioner

Athena SWAN Bronze Award