MANCHESTER
1824
The University of Manchester

COMP37111: Advanced Computer Graphics

The Rendering Equation and BRDF

Steve Pettifer

September 2020

Department of Computer Science
The University of Manchester

1 Solving The Rendering Equation

Creating a realistic computer-generated image involves, in one form or another, modelling the physcial properties and interactions between light, materials and the human visual/perceptual system. Whichever way we think about it, what we see in the real world is a result of combination of these things: light from one or more sources interacts with the objects in the environment in complex ways, and eventually some of the light enters our eyes and generates an image on our retinas.

Although coming at the problem from slightly different angles[1], this process was first represented mathematically by Immel, Cohen, and Greenberg (1986) and Kajiya (1986) more or less at the same time. The so-called **rendering equation**[w] takes the form:

$$L_o(\mathbf{x}, \boldsymbol{\omega}, \lambda, t) = L_e(\mathbf{x}, \boldsymbol{\omega}, \lambda, t) + \int_\Omega f_r(\mathbf{x}, \boldsymbol{\omega}', \boldsymbol{\omega}, \lambda, t) L_i(\mathbf{x}, \boldsymbol{\omega}', \lambda, t)(-\boldsymbol{\omega}' \cdot \mathbf{n})\delta\boldsymbol{\omega}' \quad (1)$$

where

- λ is a specific wavelength of light
- t is time
- $L_o(\mathbf{x}, \boldsymbol{\omega}, \lambda, t)$ is the total amount of light of wavelength λ directed outward along direction $\boldsymbol{\omega}$ at time t from a particular position $\mathbf{x}$
- $L_e(\mathbf{x}, \omega, \lambda, t)$ is the emitted light
- $\int_\Omega f_r(\mathbf{x}, \boldsymbol{\omega}', \boldsymbol{\omega}, \lambda, t) L_i(\mathbf{x}, \boldsymbol{\omega}', \lambda, t)(-\boldsymbol{\omega}' \cdot \mathbf{n})\delta\boldsymbol{\omega}'$ is an integral over a hemisphere of indward direction
- $f_r(\mathbf{x}, \boldsymbol{\omega}', \boldsymbol{\omega}, \lambda, t)$ is the **bidirectional reflectance distribution function**[w] (the 'BRDF'—we'll come back to this in a bit)
- $L_i(\mathbf{x}, \boldsymbol{\omega}', \lambda, t)$ is the light of wavelength λ coming inward toward $\mathbf{x}$ from direction $\boldsymbol{\omega}'$ at time t
- $-\boldsymbol{\omega}' \cdot \mathbf{n}$ is the attenuation of inward light due to incident angle

The maths of the rendering equation can seem rather daunting at first, but breaking it down into its component parts, it's quite easy to see what's going on.

Let's ignore the integral part for now, and just look at some of the individual components. The equation focuses on a point on a surface in the scene $\mathbf{x}$, and we're trying to work out what 'colour' this point is so we can project this onto a viewplane and draw a suitably coloured pixel (Figure 2). There are two vectors involved; one of these, $\boldsymbol{\omega}$, represents the direction from point $\mathbf{x}$ towards the viewer's virtual eyepoint. The other, $\boldsymbol{\omega}'$, represents incoming light along a particular direction (eventually we integrate over all $\boldsymbol{\omega}'$ on one side of the surface to give us a hemisphere called Ω). The other two components are easy: λ represents a particular wavelength of light (we can think of that as representing 'colour' for now), and t is time.

If for the moment we skip over the BRDF part, the overall purpose of the rendering equation should be fairly clear: it says something like "the light at a particular point $\mathbf{x}$ is a combination of any light emitted directly from that point, combined with the effects of all the light arriving at that point from all possible directions". The BRDF is what takes into account the fact that light interacts differently with surfaces depending on the angles involved; so light that 'grazes' a surface at a oblique angle contributes differently to the final result to light that hits a surface head-on (Figure 1); but we'll come back to that later.

[1] Its very hard to write this stuff without geometric puns; please just take them as read from now on

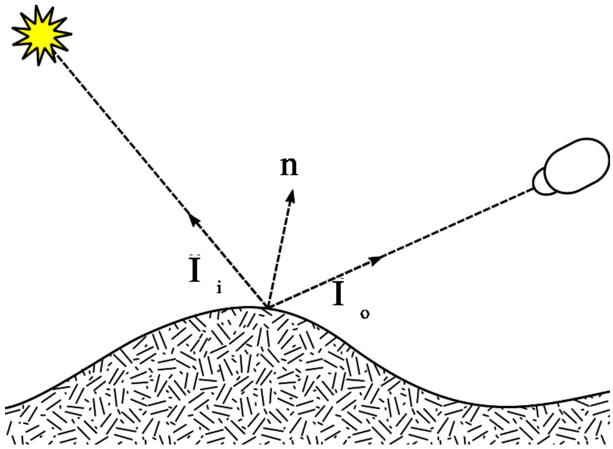

Figure 1: In a simple case of reflection, the angle between incident light and the surface normal is equal to the angle between the reflected ray and the surface normal. Image by **Meekohi**[w].

To generate a realistic looking scene then, we need to solve the rendering equation for that scene, taking into account the light sources, the different materials involved, and the position of the viewer. The rendering equation has many nice mathematical properties in this regard: first, it is mathematically *linear*, consisting only of additions and multiplications (there are no 'to the power of's involved, which computationally is a good thing). Its also *spatially homogeneous*, in that it can be applied to all points in a scene regardless of their position or orientation. This means that you can refactor and rearrange the equation relatively easily to give computationally tractable implementations.

Translating the pure mathematical representation into sensible code, however, isn't trivial: the equation, after all, includes the integration of an infinite number of incoming rays of light (all the ω' bits), and describes the effect an infinitely small point on a surface—and of course computers aren't inherently very good at dealing with an infinite number of infinitely small or infinitely large things! So all the different approaches that we'll look at in this course are approximations to 'solving the rendering equation', all of which make their own assumptions about the nature of the scene and the behaviour of light and its interaction with different materials. We'll explore the pros and cons of these different compromise approaches, but its worth keeping in mind the ideal of the rendering equation whilst we look at them.

2 The Bidirectional Reflectance Distribution Function

Let's return to the impressively-named Bidirectional Reflectance Distribution Function, the BRDF. First, we'll get the maths out of the way. The BRDF was first defined by Fred Nicodemus (Nicodemus, 1965). It's modern definition is usually given as

$$f_r(\omega_i, \omega_o) = \frac{\delta L_r(\omega_o)}{\delta E_i(\omega_i)} = \frac{\delta L_r(\omega_o)}{E_i(\omega_i)\cos\theta_i \delta\omega_i} \tag{2}$$

In this equation, L is the **radiance**[w], E is the **irradiance**[w], which are both properties describing how much light energy comes from a surface (this is a huge over-simplification, but it'll do for now). The angle between the vector representing the incoming light ω_i and the surface normal n is given by θ_i. One thing to notice here, is the $\cos\theta_i$ part of the denominator, which tells us is that the BRDF is affected by the angle between the incoming light and the surface normal. When the incoming light and the surface normal are aligned and the angle between

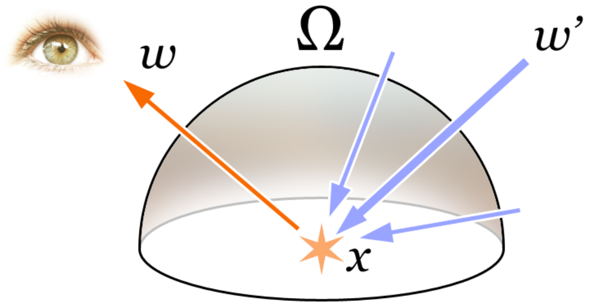

Figure 2: The rendering equation states that the outgoing illumination along direction ω from a point x is a function of any illumination coming from point x itself, combined with the effects on point x of all the light arriving at that point from all directions forming a hemisphere centred on that point. Image by **Timrb**[W].

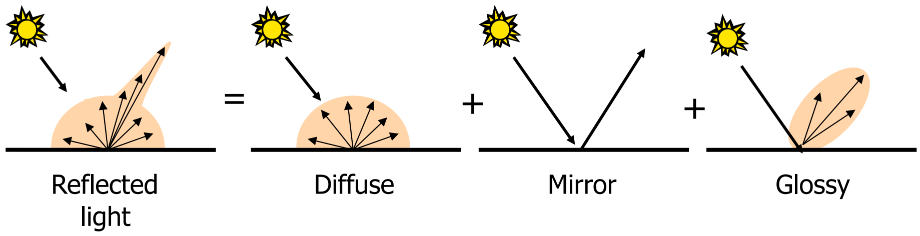

Figure 3: Light reflected from a surface is usually a combination of diffuse, perfect reflection and glossy effects. For most surfaces this means that some light is scattered randomly (the diffuse component), but that there can be a 'hotspot' of light pointing in a specific direction relative to the angle of the incoming light.

them is therefore zero, then this component will be at its maximum value; when they are perpendicular, it'll be at a minimum. Without going into the maths or the physics in any more depth, it's not hard to see that the BRDF encapsulates an effect that we see all around us all the time: objects look different depending on the angle at which we look at them, and the direction from which they are lit. We'll see this concept approximated many times during this course.

If you can't convince yourself that this is true by just looking around you, then try this thought experiment. Look at Figure 3—you've seen it before in the second year course unit when we discussed local illumination models—and recall that the light you see reflected from a surface depends on how 'matt' or 'shiny' that surface is. Matt surfaces tend to scatter any incoming light in random directions more-or-less equally; highly polished shiny surface tend to 'bounce' the light out along a particular direction; and in the real world most surfaces do a bit of both so you get some scattering with a 'hotspot' of bright light ('specular highlights'). It's fairly obvious that if you move the light point that the hotspot moves accordingly as the angle of 'bounce' changes; and it's not a huge leap of imagination to realise that whether or not you see that hotspot depends on the angle at which you're looking at the surface. If you're looking directly into the hotspot, you'll get a lot of light coming your way, and if you're looking at it from any other angle, you'll get less of that light directed towards you. So in summary: what you see is a combination of the position of the light, the nature of the material, and the position of the viewer.

References

Immel, David S., Michael F. Cohen, and Donald P. Greenberg (1986). "A radiosity method for non-diffuse environments". In: *SIGGRAPH Comput. Graph.* 20.4, pp. 133–142. ISSN: 0097-8930. DOI: 10.1145/15886.15901.

Kajiya, James T. (1986). "The rendering equation". In: *SIGGRAPH Comput. Graph.* 20.4, pp. 143–150. ISSN: 0097-8930. DOI: 10.1145/15886.15902.

Nicodemus, Fred E. (1965). "Directional Reflectance and Emissivity of an Opaque Surface". In: *Appl. Opt.* 4.7, pp. 767–773. DOI: 10.1364/AO.4.000767.