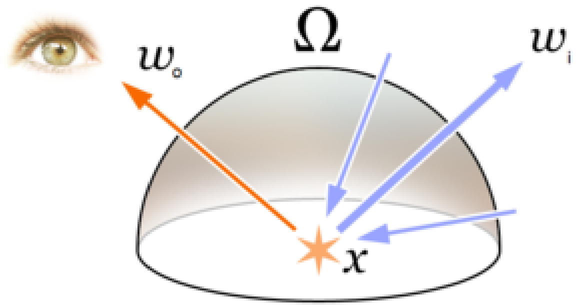
WIKIPEDIA

Rendering equation

In computer graphics, the **rendering equation** is an integral equation in which the equilibrium radiance leaving a point is given as the sum of emitted plus reflected radiance under a geometric optics approximation. It was simultaneously introduced into computer graphics by David Immel et al.[1] and James Kajiya[2] in 1986. The various realistic rendering techniques in computer graphics attempt to solve this equation.



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.

The physical basis for the rendering equation is the law of conservation of energy. Assuming that L denotes radiance, we have that at each 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 itself is the sum from all directions of the incoming light (L_i) multiplied by the surface reflection and cosine of the incident angle.

Contents

Equation form

The rendering equation may be written in the form

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

where

- $L_{\text{o}}(\mathbf{x},\,\omega_{\text{o}},\,\lambda,\,t)$ is the total spectral radiance of wavelength λ directed outward along direction ω_{o} at time t, from a particular position $\mathbf{x}$
- $\mathbf{x}$ is the location in space
- ω_{o} is the direction of the outgoing light
- λ is a particular wavelength of light
- t is time
- $L_e(\mathbf{x},\,\omega_{\text{o}},\,\lambda,\,t)$ is emitted spectral radiance
- $\int_\Omega \dots\,\operatorname d\omega_{\text{i}}$ is an integral over Ω
- Ω is the unit hemisphere centered around $\mathbf{n}$ containing all possible values for ω_{i}

- $f_r(\mathbf{x}, \omega_\mathrm{i}, \omega_\mathrm{o}, \lambda, t)$ is the bidirectional reflectance distribution function, the proportion of light reflected from ω_i to ω_o at position $\mathbf{x}$, time t, and at wavelength λ
- ω_i is the negative direction of the incoming light
- $L_\mathrm{i}(\mathbf{x}, \omega_\mathrm{i}, \lambda, t)$ is spectral radiance of wavelength λ coming inward toward $\mathbf{x}$ from direction ω_i at time t
- $\mathbf{n}$ is the surface normal at $\mathbf{x}$
- $\omega_\mathrm{i} \cdot \mathbf{n}$ is the weakening factor of outward irradiance due to incident angle, as the light flux is smeared across a surface whose area is larger than the projected area perpendicular to the ray. This is often written as $\cos\theta_i$.

Two noteworthy features are: its linearity—it is composed only of multiplications and additions, and its spatial homogeneity—it is the same in all positions and orientations. These mean a wide range of factorings and rearrangements of the equation are possible. It is a Fredholm integral equation of the second kind, similar to those that arise in quantum field theory.[3]

Note this equation's spectral and time dependence — L_o may be sampled at or integrated over sections of the visible spectrum to obtain, for example, a trichromatic color sample. A pixel value for a single frame in an animation may be obtained by fixing t; motion blur can be produced by averaging L_o over some given time interval (by integrating over the time interval and dividing by the length of the interval).[4]

Note that a solution to the rendering equation is the function L_o. The function L_i is related to L_o via a ray-tracing operation: The incoming radiance from some direction at one point is the outgoing radiance at some other point in the opposite direction.

Applications

Solving the rendering equation for any given scene is the primary challenge in realistic rendering. One approach to solving the equation is based on finite element methods, leading to the radiosity algorithm. Another approach using Monte Carlo methods has led to many different algorithms including path tracing, photon mapping, and Metropolis light transport, among others.

Limitations

Although the equation is very general, it does not capture every aspect of light reflection. Some missing aspects include the following:

- Transmission, which occurs when light is transmitted through the surface, such as when it hits a glass object or a water surface,
- Subsurface scattering, where the spatial locations for incoming and departing light are different. Surfaces rendered without accounting for subsurface scattering may appear unnaturally opaque — however, it is not necessary to account for this if transmission is included in the equation, since that will effectively include also light scattered under the surface,
- Polarization, where different light polarizations will sometimes have different reflection distributions, for example when light bounces at a water surface,
- Phosphorescence, which occurs when light or other electromagnetic radiation is absorbed at one moment in time and emitted at a later moment in time, usually with a longer wavelength (unless the absorbed electromagnetic radiation is very intense),
- Interference, where the wave properties of light are exhibited,
- Fluorescence, where the absorbed and emitted light have **different** wavelengths,
- Non-linear effects, where very intense light can increase the energy level of an electron with more energy than that of a single photon (this can occur if the electron is hit by two photons at the same time), and emission of light with higher frequency than the frequency of the light that hit the surface suddenly becomes possible, and
- Relativistic Doppler effect, where light that bounces on an object that is moving in a very high speed will get its wavelength changed; if the light bounces at an object that is moving towards it, the impact will compress the photons, so the wavelength will become shorter and the light will be blueshifted and the photons will be packed more closely so the photon flux will be increased; if it bounces at an object that is moving away from it, it will be redshifted and the photons will be packed more sparsely so the photon flux will be decreased.

For scenes that are either not composed of simple surfaces in a vacuum or for which the travel time for light is an important factor, researchers have generalized the rendering equation to produce a *volume rendering equation*[5] suitable for volume rendering and a *transient rendering equation*[6] for use with data from a time-of-flight camera

References

1. Immel, David S.; Cohen, Michael F.; Greenberg, Donald P. (1986), "A radiosity method for non-diffuse environments" (http://www0.cs.ucl.ac.uk/research/vr/Projects/VLF/vlfpapers/multi-pass_hybrid/Immel_D_S__A_Radiosity_Method_or_Non-Diffuse_Environments.pdf) (PDF), *Siggraph 1986*: 133, doi:10.1145/15922.15901 (https://doi.org/10.1145%2F15922.15901), ISBN 0-89791-196-2
2. Kajiya, James T. (1986), "The rendering equation" (http://www.cse.chalmers.se/edu/year/2011/course/TDA361/2007/rend_eq.pdf) (PDF), *Siggraph 1986*: 143, doi:10.1145/15922.15902 (https://doi.org/10.1145%2F15922.15902), ISBN 0-89791-196-2
3. Watt, Alan; Watt, Mark (1992). "12.2.1 The path tracing solution to the rendering equation". *Advanced Animation and Rendering Techniques: Theory and Practice*. Addison-Wesley Professional. p. 293. ISBN 978-0-201-54412-1.
4. Owen, Scott (September 5, 1999). "Reflection: Theory and Mathematical Formulation" (http://www.siggraph.org/education/materials/HyperGraph/illumin/reflect2.htm). Retrieved 2008-06-22.
5. Kajiya, James T.; Von Herzen, Brian P. (1984), "Ray tracing volume densities", *Siggraph 1984*, **18** (3): 165, doi:10.1145/964965.808594 (https://doi.org/10.1145%2F964965.808594)
6. Smith, Adam M.; Skorupski, James; Davis, James (2008). *Transient Rendering* (http://classes.soe.ucsc.edu/cmps290b/Fall07/TransientRendering/ucsc-soe-08-26.pdf) (PDF) (Technical report). UC Santa Cruz. UCSC-SOE-08-26.

External links

- Lecture notes from Stanford University course CS 348B, *Computer Graphics: Image Synthesis Techniques*