CS6630 Realistic Image Synthesis

Steve Marschner
Fall 2022
Physics based rendering: a brief history

1968 Appel

1980s
• Ray Tracing
• Radiosity
• Microfacet model
• Rendering Equation

1990s
• Heyday of Radiosity
• Major Path Tracing variants emerge

2000s
• Material models
• Volumes, diffusion

2010s
• GPU Ray Tracing
• Denoising
• Path Tracing refinements

2020s
• Differentiable Rendering
• Real-time Path Tracing
Ray Tracing
Appel 1968
Ray Tracing for shadows
Whitted 1980
Recursive ray tracing
Figure 8. 1984.

Cook, Porter, Carpenter 1984
Distribution Ray Tracing
Radiosity
Goral et al. 1984
Radiosity method
Hanrahan et al. 1991
Hierarchical radiosity
Sillion et al. 1991
Nondiffuse radiosity
Path Tracing
Kajiya 1986
The Rendering Equation; path tracing
Figure 10.3: A comparison of bidirectional and standard path tracing. The test scene contains a spotlight, a floor lamp, a table, and a large glass egg. Image (a) was computed with bidirectional path tracing, using the power heuristic to combine the samples for each path length. The image is 500 by 500 with 25 samples per pixel. Image (b) was computed using standard path tracing in the same amount of time (using 56 samples per pixel). The sampling techniques for a particular path length (for example, the top row shows the sampling techniques for paths of length two). The position of each image in its row indicates how the paths were generated: the -th image from the left corresponds to paths with light source vertices (and similarly, the -th image from the right of each row corresponds to paths with eye subpath vertices). Notice that the complete set of sampling techniques is not shown; paths of length are not shown because the light sources are not directly visible, and paths with zero eye or light subpath vertices are not shown because these images are virtually black (i.e. their weighted contributions are very small for this particular scene). Thus, the full set of images (for paths up to length 5) would have one more image on the left and right side of each row, plus an extra row of three images on the top of the pyramid. (Even though these images are not shown, their contributions are...
Veach and Guibas 1997
Markov Chain Monte Carlo (Metropolis Light Transport)
Kelemen et al. 2002
Primary sample space MCMC
Figure 14: Difficult caustic lighting. In this scene, a large portion of the light transport comes from implicit “caustic” paths. (b) Path tracing with 100 paths per pixel produces a very noisy image. (c) MLT, using 10,000 mutations per pixel, gets stuck on some of the caustic paths, producing a splotchy appearance. (d) ERPT using 36 MC samples and 5,000 mutations per pixel. Although some bright spots are visible, they are much less pronounced than in the MLT case. This is so despite the fact that both algorithms use the same mutation strategies. (e) Adding the noise filters to ERPT removes most of the small speckles in the image. (a) A high quality ERPT rendering of the scene using 192 MC samples and 800 mutations per pixel, again using the noise filters. The bottom row images were rendered at a 640×480 resolution in about fifteen minutes, and the top image was rendered at a resolution of 1200×800 in about seven and a half hours on a 3.2 Ghz Pentium 4.

As another example, consider a lens subpath mutation on a path of the form LDDSDE:

In this case, the pixel coordinates of the ray from the eye are changed, and we cast a ray in the new direction. The ray hits a diffuse surface, which is followed by a specular surface. In this case, the outgoing direction from this D vertex is perturbed, and extended through a specular bounce to produce the eye subpath of the form DSDE. This eye subpath is then connected directly to the next vertex in the path, to once again produce a path of the form LDDSDE. Now we apply the density change rules. First, we changed the pixel coordinates, so we apply rule 1. Next, we perturbed the outgoing direction, so we apply rule 2. Then, we extended the path through a specular bounce, so we apply rule 2 again. Finally, we connected two diffuse vertices in the middle of the path, so we apply rule 3.

Cline et al. 2005
“Energy Redistribution” with non-ergodic MCMC
Chakravarty et al. 2017
Recurrent Denoising Autoencoder
Bitterli et al. 2020
Spatiotemporal reservoir sampling

frame times 20-30ms
Real time path tracing — NVIDIA / Omniverse RTX tech demo (2020) (YouTube) (SIGGRAPH presentation)
Real time path tracing — NVIDIA / Omniverse RTX tech demo (2020) (YouTube) (SIGGRAPH presentation)
Two-Pass Methods
Walter et al. 1997 • Jensen 1996
Density estimation (Photon Mapping)
Virtual point lights (Instant Radiosity)

Keller 1997
Walter et al. 2005

LightCuts
Georgiev et al. 2013
Vertex Connection and Merging
Radiative Transport
Drebin et al. 1988
Direct volume rendering
This results in an effective optical depth \( t' = \frac{t}{1-D} \) for very small values of \( D \) for which the approximation was valid. This reduces to the classical result. When \( D \) approaches 1 (i.e., a solid packing of scattering particles), the effective optical depth approaches infinity, as would be expected. Note that this extension is particularly nice in that it only alters the value of the input parameter to the brightness function but does not otherwise alter the properties of that function.

5.2 Shadowing Effect

The scattering function was derived from considering the volume of two cylinders for entering and exiting rays of light. At that time, it was mentioned that there was a small overlap between the cylinders \( V_{in} \) and \( V_{out} \) which was neglected. This overlap actually becomes quite significant when \( L = E (p = p_0) \). The two cylinders, in fact, coincide and the entire volume is erroneously counted twice. This geometrical situation will yield a brighter observed intensity than that predicted by the simple model. The correct value will be produced by counting only...
Jensen and Christensen 1998
Volumetric photon mapping
Figure 7: The Cornell box, Cars, and Lighthouse scenes. Render times are shown as (minutes:seconds). For both the fixed and adaptive gathering approaches our method produces noise-free results while conventional photon mapping suffers from significant noise, especially around distant light sources.

Jarosz et al. 2008
Beam Radiance Estimate

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.
Pauly et al. 2000
Metropolis in volumes
Scattering Models
Figure 6a shows the color of the two components of Basilio's glass vase for all lighting conditions. The lighting is generated by the diffuse and specular reflectance of the outer surface of the vase with nonplastic and plastic materials, respectively. The material functions were calculated with the number of LEDs used and the slope of the specular component for all glass materials. The material function was obtained using the procedure of Cook and Torrance 1981. Calculating the pattern of the outer surface of the glass material is easy to simulate the bidirectional distribution of the light sources. The slope of the material is used to simulate the different normal vector of the material, and it is dominated by the energy shift. The outer surface of the glass material can be used to simulate the glass material or any other glass material. The pattern of the outer surface of the glass material can be simulated using the microfacet reflection models.
Walter et al. 2007
Microfacet transmission model
Xiao D. He et al. 1991
Comprehensive physical (wave) model for light reflection
Stam 1999
Fourier-based diffraction model
Belcour et al. 2017
Microfacet iridescence model
Jakob et al. 2014
Layered surface model
Jakob et al. 2010
Anisotropic volume media
Diffusion and Translucency
Stam 1995
Diffusion for light transport
Figure 9: A simulation of subsurface scattering in a marble bust. The marble bust is illuminated from behind and rendered using: (a) the BRDF approximation, in 2 minutes; (b) the BSSRDF approximation, in 5 minutes; and (c) a full Monte Carlo simulation, in 1250 minutes. Notice how the BSSRDF model matches the appearance of the Monte Carlo simulation, yet is significantly faster. The images in (d–f) show the different components of the BSSRDF: (d) single scattering term, (e) diffusion term, and (f) Fresnel term.

Highly scattering materials (such as milk and skin).

A particularly interesting aspect of the BSSRDF simulation is that it is able to capture the smooth appearance of the marble surface. In comparison, the BRDF simulation gives a very hard appearance where even tiny bumps on the surface are visible (this is a classic problem in realistic image synthesis where objects often look hard and unreal).

For the marble we used synthetic scattering and absorption coefficients, since we wanted to test the difficult case when the average scattering albedo is 0.5 (here the contribution from diffusion and single scattering is approximately the same). Figure 9 demonstrates how the sum of both single scattering and the diffusion term is necessary to match the Monte Carlo simulation.

Figure 10 contains three renderings of milk. The first rendering uses a diffuse reflection model; the others use the BSSRDF model and our measurements for skim milk and whole milk. Notice how the diffuse milk looks unreal and too opaque compared to the BSSRDF images, even though multiple scattering dominates and the radiant exitance due to subsurface scattering is very diffuse. It is interesting that the BSSRDF simulations are capable of capturing the subtle details in the appearance of milk, making the milk look more bluish at the front and more reddish at the back. This is due to Rayleigh scattering that causes shorter wavelengths of light to be scattered more than longer wavelengths.

Skin is a material that is particularly difficult to render using methods that simulate subsurface scattering by sampling ray paths through the material. This is due to the fact that skin is highly scattering (typical albedo is 0.95) and also very anisotropic (typical average cosine of the scattering angle is 0.85). Both of these properties mean that the average number of scattering events of a photon is very high (often more than 100). In addition, skin is very translucent, and it cannot be rendered correctly using a BRDF (see Figure 11). A complete skin model requires multiple layers, but a reasonable approximation can be obtained using just one layer. In Figure 11 we have rendered a simple face model using the BSSRDF and our measured values for skin (skin1). Here we also used the Henyey-Greenstein phase function \( g = 0.85 \) as the estimated mean cosine of the scattering angle. The skin measurements are from an arm (which is likely more translucent than skin on the face), but the overall appearance is still realistic considering the lack of spatial variation (texture). The BSSRDF gives the skin a soft appearance, and it renders the color bleeding in the shadow region below the nose. Here, the absorption by blood is particularly noticeable as the light that scatters deep in the skin is redder. For this simulation the diffusion term is much larger than the single scattering term. This means that skin reflects light fairly diffusely, but also that internal color bleeding is an important factor. The BRDF image was rendered in 7 minutes, the BSSRDF image was rendered in 17 minutes.

6 Conclusion and Future Work

In this paper we have presented a new practical BSSRDF model for computer graphics. The model combines a dipole diffusion approximation with an accurate single scattering computation. We have shown how the model can be used to measure the scattering properties of translucent materials, and how the measured values can be used to reproduce the results of the measurements as well as synthetic renderings. We evaluate the BSSRDF by sampling the incoming light over the surface, and we demonstrate how this technique is capable of capturing the soft and smooth appearance of translucent materials.

In the future we plan to extend the model to multiple layers as well as include support for efficient global illumination.
d’Eon and Irving 2011
Advanced diffusion models
Differentiable rendering
Tzu-Mao Li et al. 2018
Differentiable ray tracing
Cheng Zhang et al. 2020
Path-space differentiable rendering
Radiative backpropagation

Nimier-David et al. 2020