

Contents

1	Method	1
1.1	Constraints and assumptions	1
1.2	Method overview	2
1.3	Sampling patterns	4
1.4	Parameter acquisition	5
	Bibliography	9

Method

In this chapter, the goal is to solve the problem that we have presented, i.e. rendering translucent materials efficiently using the directional dipole. First of all, we will start this chapter with a list of constraints and assumptions that we will use to devise our method. Secondly, we will give a theoretical justification of our method, deriving a discretization of the rendering equation that can be actually implemented in a GPU environment. Then, we will discuss some possible sampling patterns and how they could possibly improve the results of the final rendering. Then, we will introduce how the actual scattering parameters are acquired in an experimental environment, in order to obtain a plausible result.

1.1 Constraints and assumptions

Quality constraints

1. Close as much as possible to a path traced solution.
2. work with the less amount as possible of provided data, i.e. only the position data and eventually the normals should be provided in order for the method to run. In particular, no uv mapping should be necessary for the method to run.

3. If the quality is not reachable within one frame, converge towards a result in a reasonable amount of time. Techniques should be used to fake the required quality in order to approximate the intermediate result.
4. flexibility: integrable in a engine-ready environment, or adaptable to different shading techniques (forward and deferred rendering)
5. tweakable: the quality / performance tradeoff should be set by a potential artist, and with the fewest number of parameters as possible.

Performance assumptions

1. Maintain a reasonable performance under changing light conditions, deformations and change of parameters, with little or none performance parameters
2. As less as dependent as possible from the geometrical complexity of the model.
3. As less as dependent from the screen resolution.
4. Being real-time, i.e. one frame should take less than 100 ms (10 FPS) to render. The ideal result would be to reach a rendering time of less than 16ms
5. Employ the advantages of the directional dipole model to improve performance.
6. Support a certain number of directional and point lights (up to 3 to 5 pixel lights, as in commercial engines Unity [2012])
7. Require little or no preprocessing.

1.2 Method overview

TODO overview

Going into the mathematical details the idea is to take the integral form of the rendering equation (equation ??):

$$L_o(\mathbf{x}_o, \vec{\omega}_o) = L_e(\mathbf{x}_i, \vec{\omega}_i) + \int_A \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) L_i(\mathbf{x}_i, \vec{\omega}_i) V(\mathbf{x}_i) (\vec{n}_i \cdot \vec{\omega}_i) d\vec{\omega}_i dA_i$$

First of all, we make the assumption of a body that is not emitting light: all the radiance from the body comes from an external source. This assumption can be trivially relaxed and implemented, but to simplify the equation in this chapter we will exclude it from the calculations. Secondly, we limit ourselves to the case of one directional light, treating the case of a point light later as an extension. The directional light direction $\vec{\omega}_l$ and radiance $L_d \delta(\vec{\omega}_d)$.

Under the first assumption, equation ?? becomes:

$$\begin{aligned} L_o^D(\mathbf{x}_o, \vec{\omega}_o) &= \int_A \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) L_d \delta(\vec{\omega}_l) V(\mathbf{x}_i) (\vec{n}_i \cdot \vec{\omega}_i) d\vec{\omega}_i dA_i \\ L_o^D(\mathbf{x}_o, \vec{\omega}_o) &= \int_A S(\mathbf{x}_i, \vec{\omega}_l, \mathbf{x}_o, \vec{\omega}_o) L_d V(\mathbf{x}_i) (\vec{n}_i \cdot \vec{\omega}_l) dA_i \end{aligned}$$

In this way, we remove the internal integral. Then, in order to make a feasible calculation, we need to discretize the other integral as well. We imagine to have a set of N points on the surface. We assume that each one of these points is visible from the light source (so we can get rid of the $V(\mathbf{x}_i)$ term). We will discuss in the implementation section how to make sure that all these points are visible. Each one of these points has an associated area A_i , so that we can write:

$$L_o^D(\mathbf{x}_o, \vec{\omega}_o) = L_d \sum_{i=1}^N S(\mathbf{x}_i, \vec{\omega}_l, \mathbf{x}_o, \vec{\omega}_o) (\vec{n}_i \cdot \vec{\omega}_l) A_i \quad (1.1)$$

Now, instead of using all the points on the surface, we consider only the points within a certain radius r^* from the point \mathbf{x}_o . This is reasonable because because the dominating term in the directional dipole is $e^{-\sigma_{tr}r}$, so after a certain critical radius the contribution from the other points becomes negligible. Assuming the points are distributed uniformly on the circle, we obtain the following area for a point:

$$A_i = \frac{A_c}{N (\vec{n}_i \cdot \vec{\omega}_l)}$$

Where $A_c = \pi(r^*)^2$ is the area of the circle. And, by inserting into equation

1.1, we obtain:

$$L_o^D(\mathbf{x}_o, \vec{\omega}_o) = L_d \frac{A_c}{N} \sum_{i=1}^N S(\mathbf{x}_i, \vec{\omega}_l, \mathbf{x}_o, \vec{\omega}_o) \quad (1.2)$$

That is our final approximation for a directional light. For a point light, following the exact same steps, we reach a similar solution. We recall that a point light is defined by an intensity I_p and a source point \mathbf{x}_p :

$$L_o^P(\mathbf{x}_o, \vec{\omega}_o) = I_p \frac{A_c}{N} \sum_{i=1}^N \frac{S(\mathbf{x}_i, \frac{\mathbf{x}_p - \mathbf{x}_i}{\|\mathbf{x}_p - \mathbf{x}_i\|}, \mathbf{x}_o, \vec{\omega}_o)}{\|\mathbf{x}_p - \mathbf{x}_i\|^2} \quad (1.3)$$

And, since the radiance is linearly summable, we can combine the contribution from an arbitrary number of $P_1, P_2 \dots P_p$ point sources and $D_1, D_2 \dots D_d$ directional sources:

$$\begin{aligned} L_o(\mathbf{x}_o, \vec{\omega}_o) &= \\ &= \sum_{k=1}^p L_o^{P_k}(\mathbf{x}_o, \vec{\omega}_o) + \sum_{k=1}^d L_o^{D_k}(\mathbf{x}_o, \vec{\omega}_o) \\ &= \frac{A_c}{N} \left[\sum_{k=1}^p I_p^k \sum_{i=1}^N \frac{S(\mathbf{x}_i, \frac{\mathbf{x}_p^k - \mathbf{x}_i}{\|\mathbf{x}_p^k - \mathbf{x}_i\|}, \mathbf{x}_o, \vec{\omega}_o)}{\|\mathbf{x}_p^k - \mathbf{x}_i\|^2} + \sum_{k=1}^d L_d^k \sum_{i=1}^N S(\mathbf{x}_i, \vec{\omega}_l^k, \mathbf{x}_o, \vec{\omega}_o) \right] \end{aligned} \quad (1.4)$$

1.3 Sampling patterns

As we discussed in the previous section, the BSSRDF function for the directional dipole is dominated by an exponential decay. So, it is more probable to find points that contribute more to the BSSRDF if we take points closer to the evaluation point \mathbf{x}_o . However, our assumption of uniform areas does not hold anymore, so we need to modify the previous equations in order to account for the non-linear sampling.

Assuming to have number generator that can generate numbers on a disc, we can create an exponentially distributed disc by rejection sampling. The probability distribution is:

$$pdf(x) = \sigma_{tr} e^{-\sigma_{tr} x}$$

We will give more detail on the process later. The radius of the point \mathbf{x}_i is

$$r_i = \|\mathbf{x}_o^{proj} - \mathbf{x}_i^{proj}\|$$

Where the two points have been projected on the circle. See figure X for more details. So now we have a new normalization term to include in order to scale back the result. So, we need now to divide by $\exp(-\sigma_{tr} r_i)$ each sample. The new equation for a directional light then becomes:

$$\hat{L}_o^D(\mathbf{x}_o, \vec{\omega}_o) = L_d \frac{A_c}{N} \sum_{i=1}^N S(\mathbf{x}_i, \vec{\omega}_l, \mathbf{x}_o, \vec{\omega}_o) e^{-\sigma_{tr} r_i}$$

The other two equations TODO and TODO change accordingly:

$$\hat{L}_o^P(\mathbf{x}_o, \vec{\omega}_o) = I_p \frac{A_c}{N} \sum_{i=1}^N \frac{S(\mathbf{x}_i, \frac{\mathbf{x}_p - \mathbf{x}_i}{\|\mathbf{x}_p - \mathbf{x}_i\|}, \mathbf{x}_o, \vec{\omega}_o)}{\|\mathbf{x}_p - \mathbf{x}_i\|^2} e^{-\sigma_{tr} r_i}$$

$$\begin{aligned} \hat{L}_o(\mathbf{x}_o, \vec{\omega}_o) &= \\ &= \sum_{k=1}^p L_o^{P_k}(\mathbf{x}_o, \vec{\omega}_o) + \sum_{k=1}^d L_o^{D_k}(\mathbf{x}_o, \vec{\omega}_o) \\ &= \frac{A_c}{N} \left[\sum_{k=1}^p I_p^k \sum_{i=1}^N \frac{S(\mathbf{x}_i, \frac{\mathbf{x}_p^k - \mathbf{x}_i}{\|\mathbf{x}_p^k - \mathbf{x}_i\|}, \mathbf{x}_o, \vec{\omega}_o)}{\|\mathbf{x}_p^k - \mathbf{x}_i\|^2} e^{-\sigma_{tr} r_i} + \sum_{k=1}^d L_d^k \sum_{i=1}^N S(\mathbf{x}_i, \vec{\omega}_l^k, \mathbf{x}_o, \vec{\omega}_o) e^{-\sigma_{tr} r_i} \right] \end{aligned} \quad (1.5)$$

1.4 Parameter acquisition

When rendering translucent materials, it is important that we have the right scattering properties, in order to match the appearance of real world objects. The scattering parameters may be tweaked by the artist and set up manually,

but this is a long process since the the scattering properties are not directly related to material appearance. In order to avoid this problems, the scattering parameters are measured from samples taken from real world objects. In this section, we will give an overview of two methods used to estimate the scattering parameters.

The first method was presented alongside the standard dipole model by Jensen et al. [2001]. The measurement apparatus consists of a series of lenses that focus the light on the sample. The light power Φ is measured by calibrating the sensor with a spectralon sample. A picture of the sample is then acquired at different exposure, in order to build an high dynamic range image. This is necessary since the scattering decays exponentially, so a high range is needed to have meaningful measurements. The measured data are then fitted to diffusion theory in order to obtain the scattering coefficients. Due to the nature of the measurement, it is not possible to measure the mean cosine g of the material, but only the reduced scattering coefficient $\sigma'_s = \sigma_s(1 - g)$ and the absorption coefficient σ_a . This measurement model uses the diffusion approximation to work, so it shares the same limitations: it is valid only for materials where $\sigma_a \ll \sigma_s$.

The second method, proposed by Narasimhan et al. [2006] proposes a method to measure the scattering coefficient by dilution. The assumption is that water does not interfere with the scattering properties of the materials dissolved within it for small distances (less than 50 cm). Naturally, the material needs then to be already in a liquid form, or to be a powder that can be easily dissolved in water. The setup of the experiment is a box full of water with a camera and an area light. High dynamic range picture of the material dissolved in water are then taken, and the scattering coefficients can be measured with a low error. Various measurements at different concentrations are needed in order to get an effective measurement of the coefficients, but then the coefficients can be extrapolated for any concentration.

Some of the scattering properties measured thanks to this method are reported in table 1.1. This coefficients will be used throughout the report when referencing to a specific material.

Material	Absorption, σ_a			Scattering, σ_s			Mean cosine, g			η	Source
	R	G	B	R	G	B	R	G	B		
Apple	0.0030	0.0034	0.0046	2.29	2.39	1.97	-	-	-	1.3	J
Ketchup	0.061	0.97	1.45	0.18	0.07	0.03	-	-	-	1.3	J
Marble	0.0021	0.0041	0.0071	2.19	2.62	3.00	-	-	-	1.5	J
Potato	0.0024	0.0090	0.12	0.68	0.70	0.55	-	-	-	1.3	J
Whole milk	0.0011	0.0024	0.014	2.55	3.21	3.77	-	-	-	1.3	J
Coffee	0.1669	0.2287	0.3078	0.2707	0.2828	0.297	0.907	0.896	0.88	1.3	N
Soy milk	0.0001	0.0005	0.0034	0.2433	0.2714	0.4563	0.873	0.858	0.832	1.3	N
Wine (merlot)	0.7586	1.6429	1.9196	0.0053	0	0	0.974	0	0	1.3	N
Beer (Budweiser)	0.1449	0.3141	0.7286	0.0037	0.0069	0.0074	0.917	0.956	0.982	1.3	N
White grapefruit juice	0.0096	0.0131	0.0395	0.3513	0.3669	0.5237	0.548	0.545	0.565	1.3	N

Table 1.1: Scattering material parameters estimated using different methods. For the source field, J materials come from Jensen et al. [2001], while N materials come from Narasimhan et al. [2006]. Note that the materials measured with the technique proposed in Jensen et al. [2001] are without the g coefficient.

Bibliography

Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 511–518, New York, NY, USA, 2001. ACM. URL <http://doi.acm.org/10.1145/383259.383319>.

Srinivasa G. Narasimhan, Mohit Gupta, Craig Donner, Ravi Ramamoorthi, Shree K. Nayar, and Henrik Wann Jensen. Acquiring scattering properties of participating media by dilution. *ACM Trans. Graph.*, 25(3):1003–1012, July 2006. URL <http://doi.acm.org/10.1145/1141911.1141986>.

Unity. *Unity Manual: Forward Rendering Path Details*, 2012. URL <http://docs.unity3d.com/Manual/RenderTech-ForwardRendering.html>.