Rendering Particles in a Shaft of Light

Taekyu Shin*
University of Maryland Baltimore County

Marc Olano
University of Maryland Baltimore County

D3D10 33.45 fps Vsync off (800x600), R8G8B8A8_UNORM (MS1, Q0)
HARDWARE: ATI Mobility Radeon HD 5730
Press F1 for help

☐ Use Angle (O)ptimization
☐ Use Min & Max (Z)
☒ (R)ender Scene
☒ Render (V)olume Light
☒ Use (P)ost Processing
☐ Render (B)loom

D3D10 50.69 fps Vsync off (800x600), R8G8B8A8_UNORM (MS1, Q0)
HARDWARE: ATI Mobility Radeon HD 5730
Press F1 for help

☐ Use Angle (O)ptimization
☐ Use Min & Max (Z)
☒ (R)ender Scene
☒ Render (V)olume Light
☒ Use (P)ost Processing
☐ Render (B)loom

a b c

Figure 1: *Natural glowy effects attenuating from the light center and shafts of light and naturally jitted in the shafts of light due to particles. (a) Real Picture. (b) Our Implementation. (c) Our Implementation with Lighting Jittered through Particles.*

Abstract

In real life, we often face some lighting situations where glowy effects are produced by light sources where there are many visibly big particles. Using functions in existing shading languages or built-in functions in graphics hardware can generate rather artificial images because it does not consider scattering. Previous works have proposed similar lighting effects, including visualization of shafts of light and atmosphere of the earth. However, although plausible, those lighting effects can look artificial due to the little difference from the those of the real world – particle visualization in the lighting effect. Our method visualizes lighting effects with natural attenuation at the edge in the shafts of light. It also visualizes decently distributed particle effects. Also, our technique can be used in general lighting situation since it is based on comprehensive model on particle distribution. Also, it can easily be extended in such a way that users can manipulate the distribution of particle density. Users can also equalize scattering effect by changing the user input β.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Global Illumination;

Keywords: scattering, multiple scattering, mie theory, particles, global illumination, illumination

1 Introduction

When raining, everything is looks darker than it normally does without raining. This is because of the physical phenomenon, scattering. The scattering of light, especially inelastic scattering, makes color to spread in particular ways based on angles and wavelengths of a photon. Without considering these scattering theories, it may be very difficult to achieve very realistic scenes. These effects can be significant since many graphics applications such as games and animation sometimes need to render mist and fog. Therefore, many computer graphics research works have proposed practical model for single and multiple scattering related to such foggy effects. To achieve realistic looks in the scene, shading should lose its colorfulness. Light intensity is attenuated based on distance, but light is lit through wider space. For light passing through visible participating media in space which has adequate number of particles, we should be able to see different intensity of light based on the density of particles per unit space. This is because the reflection of light occurs due to each particle in the space.

Some of previous research has shown volumetric light effects such as shafts of light, however, it did not show natural attenuating light intensity. Some of previous research has shown that natural light attenuation in a light glow can be achieved with an analytical model; however, it did not achieve realistic looks of particles in the air since the theory basis lies on simplified single scattering models. To render dense fog effects and especially clouds, multiple scattering is most likely a better choice than single scattering because dense fog and clouds exhibit strong multiple light scattering. [Mark Harris 2001] Thus in order to visualize particle effects in such fogs or glows around or in light, we should use multiple scattering. Some achieved many different foggy scenes with multiple scattering; however, to achieve a more realistic-looking scene, it would still be desirable to place particle effects in the scene. Unfortunately, it can be very challenging to render not only naturally attenuating lighting effect and but also particle effects due to scattering given its complexity of the diffuse process in the case of rendering scattering from each particles.

In this paper, we presents a method that can render particle-based effects based on a probability function. Our model is roughly based on the multiple scattering theory and more rigidly based on the single scattering theory. We use light

---

*e-mail: shin7@umbc.edu

intensity at a nearby particle coordinate in the world space to mimic roughly multiple scattering model. It achieves particle effects in the shafts of light although it can be used for any lighting situation. It also achieve natural attenuation effect from light sources by using the existing analytic scattering model.

2 Related Work

Starting from Blinn's work [1982], there have been numerous research papers based on analytic functions using light scattering theories. Those research publications can be categorized based on their characteristics of each solution. Blinn's method is a statistics-based scattering model based on Chandrasekhar's Radiative Transfer Equation(RTE) [1950] Max [1986] modified the scanline and followed [Crow 1977] to visualize shadow. Kajiya [1986] shows well-suited rendering equation for Computer Graphics from well-known radiative heat transfer literature- Thermal Radiation Heat Trarsfer. Goral [1984] describes how irradiance from diffuse surfaces cause other objects to light, such as the color-bleeding effect.

Analytic Model

Some methods are strongly based on analytical functions. These method are usually not interactive due to heavy computation and no use of graphics hardware for parallel computation. Kajiya and Herzen [1984] use single scattering. They separate high and low albedo cases and provide approximation for high albedo with spherical harmonics because of the characteristics of cloud where complex multiple scattering effects commonly occur. This can be a limited model because rendering clouds with single scattering may not be suitable since in clouds occurs complex multiple scattering. Nishita [1987] presented a shading method used for atmospheric effects with single scattering.It provides not only a simplified Rayleigh scattering equation but also a simplified use of Mie scattering theory with simple trigonometric functions. Pattanaik and Mudur [1993] proposed a method that works for the situation where viewers are immersed in the participating media. Its case matches ours. By storing data of energy flux throughout voxel structures and summing up energy while traversing, it computes global illumination effects. This matches our method in a way that it assumes where viewers and the objects are in the same participating media. Another work that assumes that the viewer is in the homogeneous participating media involves [Narasimhan and Nayar 2003]. They proposed a model that works in general for many different participating media situation(haze, fog, mist, rain). The authors first derive multiple scattering scheme model from an isotropic light source. they use the fact that RTE can be used to solve it at any point in the atmosphere. However, multiple scattering model may not be suitable for interactive models due to its complexity. Also, multiple scattering effects through non-homogeneous media was generated in a full monte carlo simulation in [Jensen et al. 2001]. A recent paper [Pegoraro and Parker 2009] presents an analytical model extended from the Airlight model and Radiative Transfer Equation(RTE) in a closed function form. It is applicable in a little dynamic scenes since additional terms based on different variables can be derived from the original scene, and it is not very distinguishable from the reference scene. However, this work is hardly interactive

Interactive Model

Lecocq [2000] presents a rendering model that works in real-time using mathematical approximation of light transport theory through participating media. As Pegoraro [2009] indicates, it can be intractable in some dynamic settings. Therefore, we chose not to use it. Dobashi, Yamato and Nishita [2000] proposed an interactive model for rendering a shaft of light using volumetric structures. By successfully getting Riemann sum, it generates a naturally soft shaft of light. Thus, it is easy to visualize non-uniform density of particles. Although it may also have hole problems and thus require to fill the holes, it can result in a more realistic scene with natural attenuation of light. This technique is ideal for our method since this method uses volume, which allows us to render different distribution of particles. However, it can be prone to low performance based on the number of planes. Therefore, we only use one light plane and compute the value by projecting it onto the plane. Dobashi [2002] developed further their previous method to more accurately render shadows and decrease the dimensions of maps for memory and optimization. However, it does not specifically consider natural attenuation of light shafts based on the distance between the pixel and the center of the light shafts although the rendered light shafts themselves can be inherently soft. Especially this work provides a way of rendering a scene where the Mie scattering theory is observable. Baran [Baran et al. 2010] visualizes shaft of light with natural shadows using linear transformation and pre-filtering. There still may be artifacts. However, the epipolar transformation enables a very plausible shadows in 3D scenes at interactive rates. Chen [Chen et al. 2011] further presented a developed method with min-max data structure, which makes it possible to work very interactively. However, it does not consider natural attenuation of light as the pixel gets far from the center of the light shaft. It does not consider participating media such as variation in density of particles throughout the space, which can result in an unrealistic image. Bo Sun [2005] shows a real-time model that visualizes single-scattering in foggy settings. This physically based but simplified model renders a natural and improved light scattering effects. By combining its model with BRDFs and environment mapping, it extends its technique to general use. Because this can be easily extended to general uses and rather simplified compared to more complex models like [Pegoraro and Parker 2009], we further take this to render natural attenuation in shaft of light.

Although a lot of previous methods decently visualize volumetric scattering light effects, such light effects as shafts of light and glows around light sources often need more complex attenuation functions due to its complexity of light transport in aerosol.

3 Technical Approach

Our model is roughly based on multiple-scattering theories. To add the particle effect, we add a probability density function to define how many particles lie throughout space. We base our scattering of light on the probability function so that users can manipulate the intensity of scattering effects per unit space. Our model assumes that scattering direction from each particle is isotropic; therefore, we pre-define an equally distributed direction texture that determines particle coordinates deterministically from each rendering coordinate in order to add the particle effect. This ensures that particle effects from light scattering stays consistent.

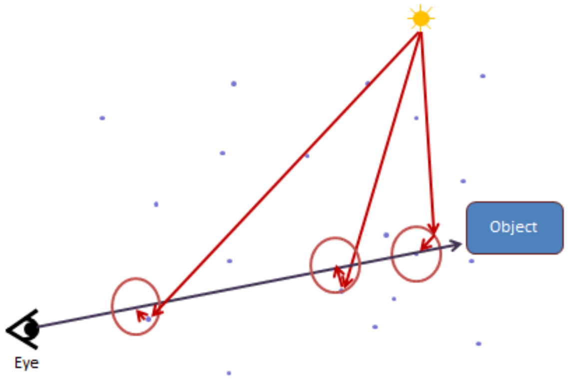


Figure 2: *Simplified diagram that represents our approach. It does ray-marching and it sums up scattered light intensity through the view vector. It only picks a particle coordinate to compute scattered light within some distance from a point on the view vector.*

Instead of using the rendered coordinate, we use the computed particle coordinate within the unit space to compute light intensity in order to emphasize the particle effect in the shafts of light. Figure 2 represents our method in a simple form.

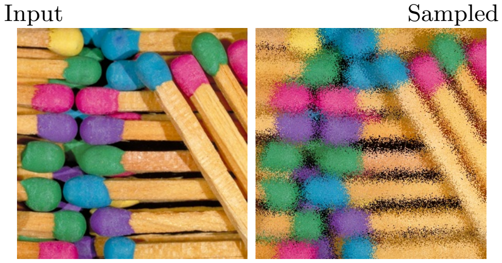

Input
Sampled

Figure 3: *The left image is the input and the right image is from one sample taken based on gaussian distribution where σ is 5.*

4 Implementation

Our implementation is based on [NVidia 2008 (accessed July 1, 2011)a], which is very similar to but simpler than [Dobashi et al. 2000]. We took the sample code from [NVidia 2008 (accessed July 1, 2011)b] and built our model on it. To provide the general structure of the base implementation. We show the general steps to render a pixel as follow.

```
1. Initialize light shadow texture.
2. Initialize a ray at the eye position.
3. WHILE until ray hits object.
    3.1 Get a sample $I_o$ from the current point.
    3.2 Get a nearby particle coordinate.
    3.3 Compute $I_s$ from the nearby particle.
    3.4 Get the light intensity at the desired point.
    3.5 Repeat 3.1 - 3.2
    3.6 Ray forwarding.
4. Return the sum.
```

Our model can be represented as:

$$I = I_o + (1 - (1 - \bar{d})^{0.4}) * I_s. \tag{1}$$

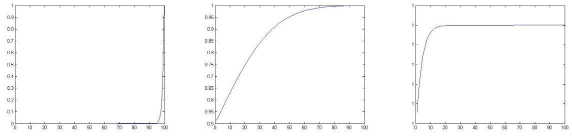
I_o is the intensity from the ray-marching point every time it forwards its ray. $\bar{d}$ is the particle density value in the unit space. I_s is a value that represents scattered intensity value from nearby particle points. Although it may be physically correct to use $\frac{e^{-\bar{d}}}{distance^2}$ in order to attenuate the I_s term. [S.N. Pattanaik 1993] Instead, we use $(1-\bar{d}^{0.4})$ term because this term results in a more obvious result. Through the *step 3.2 and 3.4*, our method computes I_s. We explain further the details of the computation below.

Determining Particle Density per unit space

We first determine the particle coordinate every ray-forwarding during the ray-marching. We use a common log-normal distribution cumulative function as our particle density since it is often used to express particle density distribution in aerosol. Based on the function, we determine the particle density and further particle positions in a unit space throughout the whole space and thus we can render a different scene with different particle density distribution per unit space.

$$\bar{d} = cdf(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{x}^{-\infty} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \tag{2}$$

p is a log-normal probability density function. Its default mean $\mu = 0.5$, standard deviation $\sigma = 0.5$. μ and σ can be modified in order to achieve different results. To expedite our algorithm, we make pre-computed textures from the probability function defined. (See Figure 4)
One alternative to this probability function is to use the gabor noise function. [Lagae et al. 2009] We implemented in Matlab and feed the data manually.


(a) a (b) b (c) c

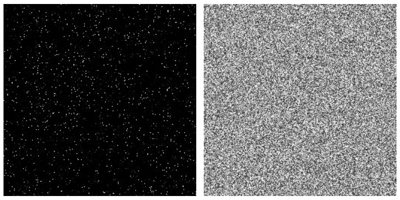

(d) Output A (e) Output B (f) Output C

Figure 4: *On the first row, it is the graph of the log-normal distribution function with μ=2, σ=0.1 and μ=0, σ=0.1 and μ=2, σ=0.1 respectively. In the second row, each image is the corresponding texture to the probability function above respectively.*

Computation of Scattering Intensity

To achieve natural attenuation and particle effects in light glows, we incorporate the surface radiance model to compute I_s, which defines scattering effects. In order to do it, we determine where the neighbor particle lies based on the probability density value from above.

$$D_i = k_c * (1 - \bar{d}) \tag{3}$$

$$C_i = R_i + D_{R_i} * direction \tag{4}$$

direction is pre-defined and saved in a texture. Since our model assumes scattering angle is isotropic, we use randomly equally distributed directions. D_{R_i} is distance between the ray marching point i and a particle point in the unit space for the ray R. k_c is a constant, which can affect the range of scattering effect. We set it to 7. C_i is the particle coordinate in the unit space. R_i is a R^3 vector for the current ray position during ray-marching.

$$\beta = ((1 - \bar{d}) + c_1) * c_2 \tag{5}$$

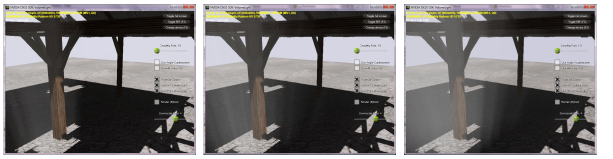
I_s is computed by using the surface radiance model from [Sun et al. 2005] where we set β as the equation above c_1 is a constant term to ensure that $(1 - \bar{d})$ is not set to 0. Experimentally, we set c_1 to 0.05. c_2 is also a constant term. Also experimentally, it is 0.03.

5 Method Justification

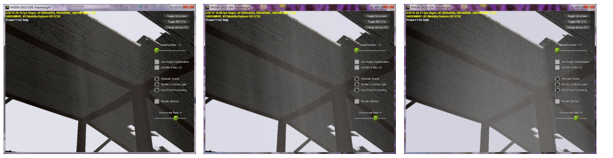
Non-rigid Multiple Scattering Model Our method mimics multi-scattering model by projecting single-scattering from different particles acting as a luminaire. However, it is roughly based on multiple scattering since it does not compute scattering effects from every particle in the space to render one pixel. However, it well suits the characteristics of foggy effects when used to visualize it. That is because very far particles have very minor effects in rendering since attenuation due to distance is significant. Thus, we take the most significant particle to consider, the firs and second closest particles. The picture from [Grundland and Dodgson 2007] well shows this fact.(Please see Figure 3) Secondly, in order to visualize particle effects in shafts of light, it is important to have natural noise based on analytic models. This method results in natural jitter in physically plausible ways. Last, it ensures natural attenuation of shafts of light. The intensity at the edge of shafts of light is affected by dark neighbor particles, leading to the natural attenuation at the edge of shafts of light.

Attenuation effect : Many existing techniques visualize shafts of light without jittery effects. Light glow in the scene can be very obvious and it might not be ideal looks given the fact that there is natural attenuation based on distance in real life. Our sampling scheme from neighboring particles can help to achieve this attenuation since each pixel is affected by its neighbors and neighbor pixels are likely to be un-lit near the edge of light glow

Jittering Light Effects : When rendering particles in a light glow, the attenuation of glow should not be necessarily completely smooth. Our neighbor sampling scheme can accomplish this goal.



(a) Result A (b) Result B (c) Result C



(d) Result A2 (e) Result B2 (f) Result C2

Figure 5: *In the first row, the result A is created with the probability function of μ=2, σ=0.1, whose graph is 4(a). The result B with μ=0, σ=0.1, whose graph is 4(b). The result C with μ=2, σ=0.1, whose graph is 4(c). In the second row, each image is the corresponding scne for the probability function. In these images, please note some particle effects differing from different particle density inputs.*

6 Result

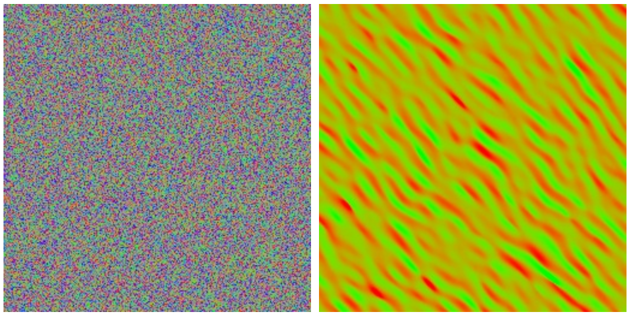
We show the result scenes with different probability function inputs. (Please see Figure 5) Note that when there are dense particles per unit space, shafts of light are scattered throughout the space, amplifying the volume of the bright space. (Compare 5(a) and 5(c)) Also note that the attenuation at the edge of the shafts of light is smoother in dense participating media.

In the images, because our method takes a lot of samples through ray-marching, it tends to average every pixel, attenuating visible particle visualization. When the shafts of light is thin, less samples are taken through ray-marching and thus the particle effects are more obvious. Therefore, we show the same scene from a different angle. (See Figure 5(d), 5(e) and 5(f)) Notice that 5(f) has more jittered lighting than 5(d).

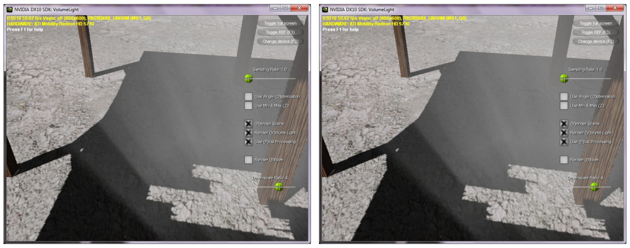
We also used the gabor noise as an input to our *directions*, which was previously computed as equally distributed random vales. The result does not vary noticeably. (Please see Figure 6.) This is because tested directions from the gabor noise is 2 dimensions. Therefore, we suggest an idea that can solve the problem in the Future Work section. Also, it is very possible that our method be extended to general lighting situation since our method is analytically based on the scattering theory and the particle density function.

7 Limitation

We use I_o and multiply it by a constant. However, when there are no dense particles, I_o should be amplified. Thus, it is ideal to multiply it by some variable determined by scattering theories. Well amplified I_o would result in a more natural shafts of light. Another limitation is that the probability function is not computed on the fly. It causes rather an inconvenient user interface and can be difficult for end-users to figure out how to use this implementation. The attenuation of I_s, although physics-based, is rather abrupt. It should be naturally attenuated. Also, it is a very important to visualize specular effects where there are dense particles. However, it is missing in our implementation. Due to that, it



(a) Equi-distributed directions (b) Directions from gabor noise



(c) Output (d) Output

Figure 6: *Comparison between the equally distributed direction and gabor noise direction with strong anisotropicity. In the first row, 6(a) is the equ-distributed function and 6(b) is the gabor noise. In the second row, 6(c) is the equ-directions image and 6(d) are the corresponding images to the gabor noise directions.*

cannot generate natural specularity through dense particles. (See Figure 7)

Figure 7: *Notice the natural specular reflections from particles. Our method cannot visualize this since it is lacking the specular term.*

8 Future Work

There are some situations where we can see the Mie scattering theory other than the Rayleigh scattering theory. Most of previous research did not incorporate the Mie scattering theory rigidly. To render the natural bluish color-effects to visualize sufficient big particles with respect to visible range of wavelengths of light, the Mie scattering theory can come in handy. Nishita's work [1987] provides a simplified model to render the Mie scattering theory. Dobashi's work [2002] incorporated the Mie scattering theory to visualize the bluish and redish atmosphere of the Earth. It can be an active research area to incorporate the Mie scattering theory properly.

It is also possible to consider using the data of particle density extracted from the Fraunhofer approximation. [Vargas-Ubera J. 2001] (See appendix for the details.) We use a probability function; however, it is uncertain that it exactly matches the normal particle density distribution in aerosol. The data of the particle distribution and sizes from Fraunhofer approximation can be used to better visualize the particle effects in light glow.

It is also very possible to add anisotropic scattering effect in the implementation by using the 3-dimensions gabor noise. We tested the gabor noise to determine the scattering directions. Our result with 2-dimensions gabor noise is rather dissatisfactory. We believe that it can be improved further by using a 3-dimension gabor noise, which can be easily extended from the implementation provided on the gabor noise website. It should be used to directly determine the coordinates of each partcle in the world space rather than let it determine the directions of scattering.

A The Fraunhofer approximation in the air

Using the data in Range of validity of the Fraunhofer approximation in the estimation of particle size distributions from light diffraction, we can generate a probability function that defines BRDF and the constant T_{sp} in the airlight model from Bosun's method [2005]. T_{sp} is defined as

$$T_{sp} = Particle\, density \times \frac{1}{distance} \tag{6}$$

where $distance$ is the distance between particle position and the rendered coordinate

B Particle Radiance Model

We incorporate the surface radiance model from [Sun et al. 2005]

$$L_p = I_0 k_d [\frac{e^{-T_{sp}}}{D_{sp}^2} cos(\theta_s) + \beta^2 \frac{G_0(T_{sp}, \theta_s)}{2\pi T_{sp}}] + \tag{7}$$

$$I_0 k_s [\frac{e^{-T_{sp}}}{D_{sp}^2} cos^n(\theta_s^{'}) + \beta^2 \frac{G_n(T_{sp}, \theta_s)}{2\pi T_{sp}}] \tag{8}$$

We modified the model by taking out the specularity term and substituting the β as described in the Implementation section.

$$L_p = I_0 k_d [\frac{e^{-T_{sp}}}{D_{sp}^2} cos(\theta_s) + (((1 - \bar{d}) + c_1) * c_2)^2 \frac{G_0(T_{sp}, \theta_s)}{2\pi T_{sp}} \tag{9}$$

References

Baran, I., Chen, J., Ragan-Kelley, J., Durand, F., and Lehtinen, J. 2010. A hierarchical volumetric shadow algorithm for single scattering. *ACM Trans. Graph. 29* (December), 178:1–178:10.

Blinn, J. F. 1982. Light reflection functions for simulation of clouds and dusty surfaces. In *Proceedings of the 9th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '82, 21–29.

Chandrasekhar. 1950. *Radiative Transfer*.

Chen, J., Baran, I., Durand, F., and Jarosz, W. 2011. Real-time volumetric shadows using 1d min-max mipmaps. In *Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '11, 39–46 PAGE@7.

Crow, F. C. 1977. Shadow algorithms for computer graphics. *SIGGRAPH Comput. Graph. 11* (July), 242–248.

Dobashi, Y., Yamamoto, T., and Nishita, T. 2000. Interactive rendering method for displaying shafts of light. In *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on*, 31 –435.

Dobashi, Y., Yamamoto, T., and Nishita, T. 2002. Interactive rendering of atmospheric scattering effects using graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, HWWS '02, 99–107.

Goral, C. M., Torrance, K. E., Greenberg, D. P., and Battaile, B. 1984. Modeling the interaction of light between diffuse surfaces. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '84, 213–222.

Grundland, M., and Dodgson, N. A. 2007. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recognition 40*, 11, 2891–2896.

Jensen, H. W., Marschner, S. R., Levoy, M., and Hanrahan, P. 2001. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 511–518.

Kajiya, J. T., and Von Herzen, B. P. 1984. Ray tracing volume densities. *SIGGRAPH Comput. Graph. 18* (January), 165–174.

Kajiya, J. T. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '86, 143–150.

Lagae, A., Lefebvre, S., Drettakis, G., and Dutré, P. 2009. Procedural noise using sparse gabor convolution. *ACM Trans. Graph. 28* (July), 54:1–54:10.

Lecocq, P., Michelin, S., Arques, D., and Kemeny, A. 2000. Mathematical approximation for real-time lighting rendering through participating media. In *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on.*

Mark Harris, A. L. 2001. Real-time cloud rendering. *Eurographics 20*, 76–84.

Max, N. L. 1986. Atmospheric illumination and shadows. *SIGGRAPH Comput. Graph. 20* (August), 117–124.

Narasimhan, S. G., and Nayar, S. K. 2003. Shedding light on the weather. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 1*, 665.

Nishita, T., Miyawaki, Y., and Nakamae, E. 1987. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *SIGGRAPH Comput. Graph. 21* (August), 303–310.

NVidia. 2008 (accessed July 1, 2011). *Volume Light.*

NVidia. 2008 (accessed July 1, 2011). *Volume Light Sample.*

Pegoraro, V., and Parker, S. G. 2009. An Analytical Solution to Single Scattering in Homogeneous Participating Media. *Computer Graphics Forum (Proceedings of the 30th Eurographics Conference) 28*, 2, 329–335.

S.N. Pattanaik, S., 1993. Computation of global illumination in a participating medium by monte carlo simulation, July, September.

Sun, B., , Ramamoorthi, R., Narasimhan, S., and Nayar, S. 2005. A practical analytic single scattering model for real time rendering. vol. 24, 1040–1049.

Vargas-Ubera J., Gale D., F.-A. J. 2001. Range of validity of the fraunhofer approximation in the estimation of particle size distributions from light diffraction. In *SPIE*, SPIE, vol. 4419, 435–438.