Single Scattering Effects for Computer Games

Venceslas Biri

Single Scattering Effects for Computer Games

Venceslas Biri
University Paris Est Marne-la-Vallée
Gaspard Monge Institute, 5 boulevard Descartes,
F-77454 Marne-la-Valle CEDEX 2
Email : biri@univ-mlv.fr

Abstract— **This paper addresses the rendering of single scattering effects such as glows and shafts of light, along with volumetric shadows induced by shadow casters in the participating media in real-time. Our method is easy to integrate in a video game graphics engine using the shadow volume technique since it requires only a little additional texture memory and is implemented with simple shaders. Realistic images can be produced in real-time for usual graphic scenes and at a high level framerate for complex scenes, allowing changes in the properties of participating medium, animations of objects and even light sources movements.**

Keywords— **Single scattering, Real Time, Hardware rendering**

I. Goals and Contributions

In this paper we show how to render a complete single scattering illumination model. The originality of the model is the introduction of the **indirect single scattering**. Indeed, we will call **direct single scattering** the effect of the first scattering of light along view rays. The **indirect single scattering** is the same effect but along illumination rays of any point of scene.

Our contributions in this subject are :

- **Define an analytic and comprehensive formulation of light scattering along view rays and illumination rays**. Based on a angular formulation of the radiative transfer equation, we present precomputed 2D tables to compute directly these contributions.
- **Integration of the volumetric shadows**. We build a method based on the shadow volume technique and using spatial coherence strategy, allowing the rendering of volumetric shadows in the scene, especially discernable around light sources.
- **Hardware implementation**. Except for the determination of object's silhouette, all the work is done with GPU. We store our precomputed 2D tables in textures and use simple shaders to render the illumination of objects and participating media. The rendering of volumetric shadows also involves an intensive use of the stencil buffer.

Our method is not restricted to isotropic participating media which can be lit by one or several, static or moving, point light sources since no precomputation are done involving either lights or camera. Our technique produces high resolution images and takes into account volumetric shadows, cast by occluders contained in the media. With very few texture memory cost, but using intensively graphics hardware, our method can render images at a high frame rate and is real-time for classical graphics scene. Our method is also easy to implement in traditional graphics engines since it follows the same strategy than the shadow volume algorithm, and use only shaders and textures. Therefore, it is straightforward with our method to obtain animations where objects or even light sources can move.

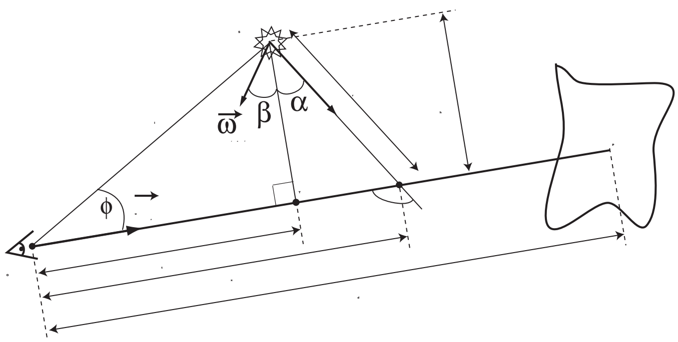

Fig. 2. Notations for our model

II. The single scattering illumination model

In this section, we will present shortly the analytic formulation of the single scattering illumination model. The precise theory can be found in [1].

A. Single scattering along a ray

Considering a view ray immersed in a participating medium, the radiance L observed along $\vec{u_P}$ from a point P can be written (see figure 2) :

$$L_{\vec{u_P}} = L_{dss}(\vec{u_P}) + e^{-k_t OP}L(P) + L_a e^{-k_t OP} + L_{oiss}(\vec{u_P}) \quad (1)$$

where L_{dss} is the direct single scattering and $L(P)$ the radiance of P attenuated by both absorption and out-scattering. The third term represents both emission and multiple scattering inducing a constant radiance L_a along the ray. The last term is the object indirect single scattering.

The radiance of P that can be written :

$$L(P) = L_{iss}(P) + e^{-k_t SP}L_d(P) + L_a e^{-k_t SP} \quad (2)$$

L_{iss} is the indirect single scattering received on point P and $L_d(P)$ an usual direct illumination, like Phong model, that is attenuated by absorption and out-scattering. The third term is once again the contribution of emission and multiple scattering on the illumination ray.

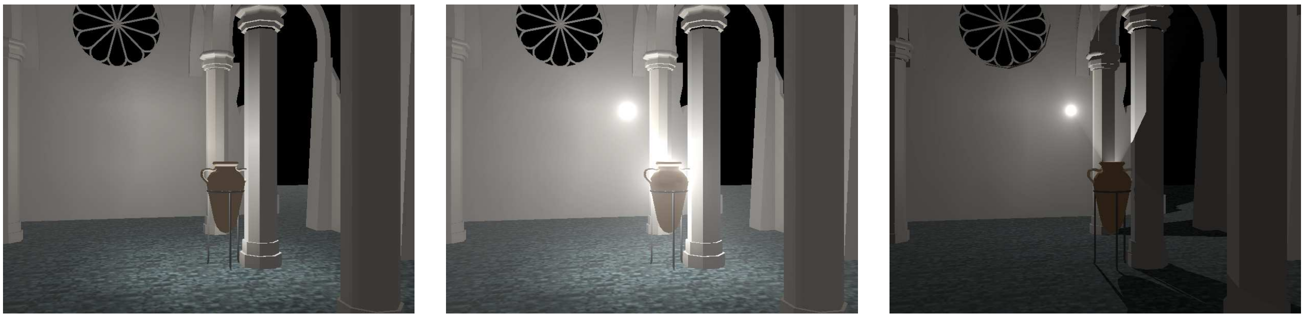

Fig. 1. The same scene lit a. (left) classically, b. (center) with single scattering and c. (right) with single scattering and volumetric shadows. Who could say precisely where are the lights in the left picture ? This is obvious in the right picture

Only L_{dss} and L_{iss} remain unknown. They both are the contribution of the first light scattering along, respectively, a view ray and an illumination ray. To resume we give directly the expressions of these two terms. Developments can be found in [1].

If we denote :

$$\Gamma(\lambda,\gamma) = \int_0^{\gamma} p(\alpha+\frac{\pi}{2})e^{-k_t\lambda\frac{\sin(\alpha)+1}{\cos(\alpha)}}d\alpha \tag{3}$$

the direct single scattering could be written :

$$\begin{aligned} L_{dss}(d,\phi) &= \tfrac{k_t^2\Omega I_S}{4\pi}\tfrac{e^{-k_t t}}{k_t h}\left[\Gamma(k_t h, atan(\frac{d-t}{h}))\right. \\ &\left. -\Gamma(k_t h, atan(\frac{-t}{h}))\right] \end{aligned} \tag{4}$$

Note that the 2D function Γ is purely numerical and only depends on the shape of the phase function. It can therefore be precomputed and stored in a 2D table.

The new single scattering Phong model can be with this equation :

$$L_{iss}(P) = \frac{k_t^2\Omega I_S}{4\pi}\ \frac{k_s}{k_t l}\ \Gamma_P(k_t l) \tag{5}$$

with

$$\Gamma_P(\phi',\lambda) = \int_0^{2\pi}\int_0^{\pi}\int_{\phi'-\frac{\pi}{2}}^{\frac{\pi}{2}} p(\alpha+\frac{\pi}{2})e^{-\lambda\frac{\cos(\alpha-\phi')+1}{\cos(\alpha)}}cos^{\eta}(\phi)d\alpha \tag{6}$$

We want to integrate now the effect of occlusions along any view rays. Equation (4) describes the particular case where the view ray remains totally lit. Using the laplace formula, it is straightforward to see that equation (4) becomes :

$$\begin{aligned} L_{dss}(d,\phi) &= \frac{k_t^2\Omega I_S}{4\pi}e^{-k_t t}[\Gamma(k_t h,\gamma_P) - \Gamma(k_t h,\gamma_B) \\ &+\Gamma(k_t h,\gamma_A) - \Gamma(k_t h,\gamma_O)] \end{aligned} \tag{7}$$

III. Hardware implementation

A. Overview of our method

Our algorithm is easy to implement. We present here the 5 steps of this method and we will precise for each step if the computation is done by CPU or by GPU.

1) (CPU) The silhouettes of every moving shadow caster are computed. If the light source is moving, every silhouette needs to be recomputed.
2) (GPU) Scene is rendered using the conventional polygonal rendering method to obtain the direct illumination and the indirect single scattering. Surface shadows can be obtained using shadow planes algorithms [2], [3]. The stencil buffer now contains lit areas of the scene. An ambient fog is added to take into account both absorption and multiple scattering.
3) (GPU) Scene is rendered once more time and direct single scattering is computed for each vertex of the scene. Depth test is set to equality. Only lit parts of the scene are rendered thanks to the stencil buffer.
4) (CPU) Shadow planes determined by the object's silhouettes are sorted in a back to front order.
5) (GPU) Shadow planes are rendered in that precise order. The depth test function accepts only planes that are closer to the camera. Front facing planes add their contribution when back facing planes subtract them. Stencil function is set to allow fragments if the stencil is equal to 1 for front facing planes and 0 for back facing ones. Front facing planes always decrement the stencil buffer and back facing ones always increment it.

All stages have to be done for each light source. As in [3], a initialization stage is done to obtain the ambiant lighting and the first depth map. Each stage is detailed in [1].

IV. Results

scene	without iss with dss and shadows	with iss, dss and shadows	number of triangles
1	31	30	14 785
2	10.5	10.2	110 014
3	7.6	7	136 266

TABLE I

FPS compared to number of triangles

References

[1] V. Biri, "Real Time Single Scattering Effects," in *Computer Games 06, Best Paper*, 2006, pp. 235–256.

[2] T. Heidman, "Real Shadows Real Time," in *IRIS Universe*, vol. 18, 1991, pp. 28–31.

[3] C. Everitt and M. Kilgard, "Practical and Robust Shadow Volumes," 2003. [Online]. Available: http://developer.nvidia.com/object/robust_shadow_volumes.html