

Transmittance Function Mapping

Cyril Delalandre

Pascal Gautron

Jean-Eudes Marvie

{cyril.delalandre, pascal.gautron, jean-eudes.marvie}@technicolor.com

Technicolor Research & Innovation

Guillaume François

guillaume.francois@gmail.com

The Moving Picture Company



(a) Cloud - 100 steps - 8 fps



(b) TinPan Alley - 500 steps - 10 fps



(c) GI-Joe* - 50 steps - 3000 lights - 7min

Figure 1: Our algorithm introduces transmittance function maps for the computation of light scattering within participating media for both real-time using a GeForce GTX480 (a, b) and production rendering (c) using Pixar's RenderMan®.

Abstract

The interaction between light and participating media involves complex physical phenomena including light absorption and scattering. Media such as fog, clouds or smoke feature complex lighting interactions that are intrinsically related to the properties of their constituent particles. As a result, the radiance transmitted by the medium depends on the varying properties on the entire light paths, which generate soft light shafts and opacity variations.

Simulating light scattering in these media usually requires complex offline estimations. Real-time applications are either based on heavy precomputations, limited to homogeneous media or relying on simplistic rendering techniques such as billboards. We propose a generic method for fast estimation of single scattering within participating media. Introducing the concept of *Transmittance Function Maps* and *Uniform Projective Space Sampling*, our method leverages graphics hardware for interactive support of dynamic light sources, viewpoints and participating media. Our method also accounts for the shadows cast from solid objects, providing a full-featured solution for fast rendering of participating media which potentially embrace the entire scene.

1 Introduction

In the course of generating images of virtual worlds closer and closer to reality, simulating translucence is unavoidable as the real world is filled by semi-transparent materials, known as participating media. Ranging from a bright haze to organic materials or heavy smoke, their accurate rendering is an essential step towards realism. The light traversal makes such materials hosts of complex optical phenomena, known as scattering, absorption and emission. This omnipresence in the real world has made realistic rendering of translucency a highly active research area for decades, yielding numerous real-time and offline solutions. The current state-of-the-art techniques are either limited to offline rendering, or, for the sake of

real-time performance, rely on heavy precomputations, approximations or assumptions on material homogeneity.

We introduce a generic method based on ray-marching for fast estimation of single scattering in both homogeneous and heterogeneous materials. The computational cost of such scattering simulation is twofold: the accumulation of scattered contributions along the viewing ray, and the estimation of the reduced light intensity for each of those contributions. In this paper we introduce the concept of *Transmittance Function Map* to represent the medium transmittance in Fourier space, and *Uniform Projective Space Sampling* for fast estimation of single scattering in dynamic participating media. This method requires no precomputation and handles the interaction with other scene components, potentially located within the medium itself. Furthermore, the light sources and viewpoint can be seamlessly moved inside or outside the medium. The scalable nature of ray marching makes our method suitable for many applications, ranging from real-time visualization and video games using graphics hardware to production-quality rendering on offline renderers such as Pixar's RenderMan®.

In the next two sections we describe previous work addressing the rendering of participating media (Section 2) and present key aspects of single scattering computation (Section 3). Section 4 introduces the bases of our rendering technique in the case of homogeneous media. Section 5 presents our solution for factorizing reduced intensity computations in the case of light scattering in heterogeneous media. In Section 6 we discuss practical issues and suggest solutions for improved performance and applicability. Our results are presented in Section 7.

2 Related Work

The literature on efficient simulation of the interactions between light and participating media has been enriched by numerous publications over the last decades. The base theory of radiative transport in participating media has been introduced in [Chandrasekhar 1950] and an extensive survey of rendering techniques for such media is available in [Cerezo et al. 2005]. This Section presents the previous works most closely related to our method, focusing on real-time solutions.

Several methods provide analytic solutions to the single scattering part of the radiative transfer equation, such as [Pegoraro and Parker 2009; Sun et al. 2005]. While effective and providing high interactivity, the underlying equations are built upon the assumption of homogeneous media, and overlook the volumetric shadows effects

*GI-Joe: Rise Of Cobra, images courtesy of Paramount Pictures

due to light occlusion by solid objects. Such occlusions are accounted for in [Wyman and Ramsey 2008], which relies on shadow volumes and ray marching. This method provides high frame rates, but the shadow volume extraction may become problematic in complex scenes.

The real-time rendering of animated heterogeneous materials has been addressed in [Zhou et al. 2008]. Based on a projection of the material into radial basis functions, this method supports multiple scattering and image-based lighting. However, it is based on heavy precomputations, which require the knowledge of the entire animation sequence. Another method is described in [Harris 2005], in which volumetric clouds are represented using dynamically-generated impostors. This technique offers real-time performance, but only provides a coarse approximation of light scattering and generates artifacts upon fast viewpoint movements.

Our *Transmittance Function Mapping* (TFM) technique makes intensive use of projective texturing, which principle has been previously devised in the literature. In particular, the deep shadow maps [Lokovic and Veach 2000] are currently a method of choice for representing transmittance within translucent materials for production rendering. Similar to shadow maps [Williams 1978], the deep shadow maps store several depth records per texel, along with an opacity value. Even using compression, an accurate representation of opacity changes requires the sampling and potential storage of many values, resulting in a large memory footprint. Several methods have been proposed to extend this concept to GPU-based rendering [Kim and Neumann 2001; Kniss et al. 2003], with similar advantages and drawbacks. Some of the deep shadow maps issues have been recently addressed in [Jansen and Bavoil 2010], which introduces *Fourier opacity maps*. Instead of explicitly storing depth and opacity values in each texel, the opacity of a particle cloud is projected into Fourier space using a small set of coefficients. This method is effective and very closely related to our technique, but focuses on opacity storage instead of the computation of actual light scattering. Furthermore, this technique works well in optically thin particle clouds, but tends to generate ringing artifacts in high density media. In this paper, the TFM technique addresses both of those issues. The next Section introduces the basics of the scattering theory underlying the remainder of this paper.

3 Single Scattering

The interaction between light and participating media is fully described by the radiative transport equation [Chandrasekhar 1950], covering both single and multiple scattering events. In this paper, we focus on single scattering events in non-emissive participating media. Such media are described at each point \mathbf{p} by the following functions:

- The absorption coefficient $\sigma_a(\mathbf{p})$ represents the amount of incoming lighting which gets transformed into other forms of energy, such as heat.
- The scattering coefficient $\sigma_s(\mathbf{p})$ is the amount of incoming lighting scattered around \mathbf{p} .
- The extinction coefficient $\sigma_t(\mathbf{p}) = \sigma_a(\mathbf{p}) + \sigma_s(\mathbf{p})$.
- The phase function $p(\mathbf{p}, \omega_{\text{out}}, \omega_{\text{in}})$ which describes the amount of light scattered at \mathbf{p} from the incoming direction ω_{in} into outgoing direction ω_{out} .

To define single scattering, let us first consider a point \mathbf{p}_n within the medium, and a scattering direction ω_{out} . Given a lighting intensity L_{ri} incoming at \mathbf{p}_n from a direction ω_{in} (Figure 2), the single scattering is:

$$Q(\mathbf{p}_n, \omega_{\text{out}}) = \sigma_s(\mathbf{p}_n) p(\mathbf{p}_n, \omega_{\text{out}}, \omega_{\text{in}}) L_{ri}(\mathbf{p}_n, \omega_{\text{in}}) \quad (1)$$

The contribution of \mathbf{p}_n to the radiance outgoing at \mathbf{p}_{in} is then:

$$L_{\mathbf{p}_n}(\mathbf{p}_{\text{in}}, \omega_{\text{out}}) = Q(\mathbf{p}_n, \omega_{\text{out}}) e^{\int_{\mathbf{p}_n}^{\mathbf{p}_{\text{in}}} -\sigma_t(\mathbf{p}) d\mathbf{p}} \quad (2)$$

The integral part of this equation represents the light attenuation along the path from \mathbf{p}_n to \mathbf{p}_{in} within the medium.

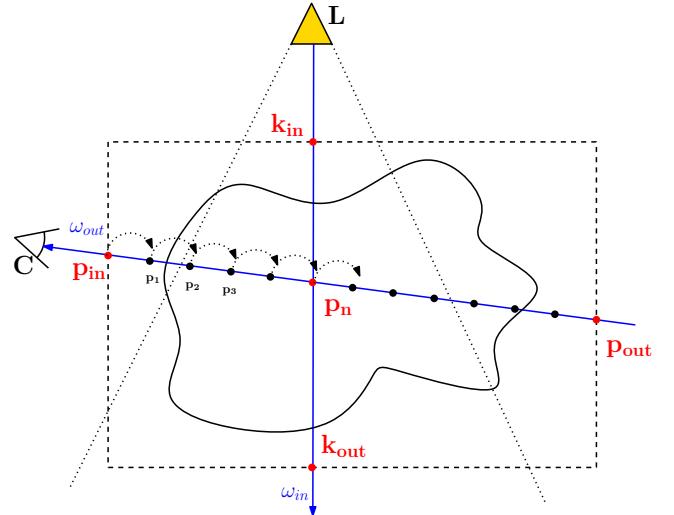


Figure 2: Notations and principle of a classical ray-marching algorithm to compute single scattering inside a participating medium bounded by a box. A ray-marching is performed along the path $[\mathbf{p}_{\text{in}}, \mathbf{p}_{\text{out}}]$; for each sample, a second ray-marching \mathbf{p}_n is performed along $[\mathbf{k}_{\text{in}}, \mathbf{p}_n]$ to compute the light reduced intensity.

As mentioned above, $L_{ri}(\mathbf{p}_n, \omega_{\text{in}})$ describes the lighting intensity incoming at point \mathbf{p}_n from the lighting direction ω_{in} . This value, known as reduced intensity, is determined from the emission properties of the light source L and the extinction coefficient of the medium along the path from L to \mathbf{p}_n :

$$L_{ri}(\mathbf{p}_n, \omega_{\text{in}}) = e^{\int_{\mathbf{p}_n}^{\mathbf{k}_{\text{in}}} -\sigma_t(\mathbf{p}) d\mathbf{p}} \quad (3)$$

The total outgoing radiance at point \mathbf{p}_{in} due to single scattering along a direction ω_{out} is then given by integrating the contributions of each point \mathbf{p} between \mathbf{p}_{in} and \mathbf{p}_{out} :

$$L(\mathbf{p}_{\text{in}}, \omega_{\text{out}}) = \int_{\mathbf{p}_{\text{in}}}^{\mathbf{p}_{\text{out}}} L_{\mathbf{p}}(\mathbf{p}_{\text{in}}, \omega_{\text{out}}) d\mathbf{p} \quad (4)$$

Based on those equations, we first introduce the concept of *Volumetric Shadow Mapping* for fast computation of single scattering simulation in homogeneous media, accounting for volumetric shadows due to solid objects. This homogeneity assumption is then raised using our *Transmittance Function Maps* for fast reduced intensity computation in heterogeneous media.

4 Volumetric Shadow Mapping

We introduce the use of shadow maps and *uniform projective space sampling* for efficient computation of light shafts in homogeneous media due to solid objects or projective textures. Our rendering method is based on deferred shading, and is divided into three main steps: the generation of the shadow map, the gathering of location, and lighting information for deferred shading on solid objects, and the single scattering computation. For the sake of simplicity, only

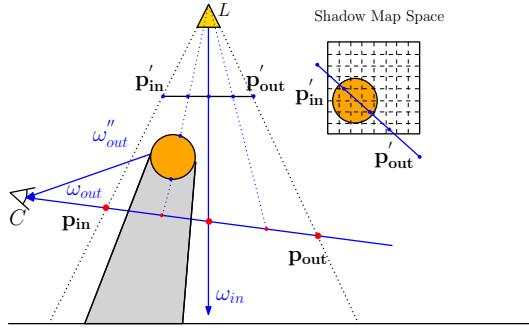


Figure 3: Notations and principle of our algorithm to compute single scattering inside an homogeneous participating medium. We perform a single ray-marching along each viewing ray and use the shadow map to check if the sample is lit.

spot lights and uniform shadow maps [Williams 1978] are considered in this paper, although any other light source type could be used as long as it complies with projective textures and a shadow mapping algorithm.

4.1 Base Algorithm

In the first step, a shadow map is generated by rendering the distances from the light source to the virtual objects into a buffer. In the second step, the incoming lighting at each visible point of the solid objects of the scene is computed using the shadow map and by leveraging medium homogeneity to solve Equation 3 analytically:

$$L_{ri}(\mathbf{p}_n, \omega_{in}) = e^{-\sigma_t \|\mathbf{p}_n \mathbf{k}_{in}\|} \quad (5)$$

Using the surface reflectance at the visible points the output of this second step is the reflected reduced intensity and the distance from each point to the viewpoint. This can be easily packed into the channels of a floating-point RGBA buffer, in which the alpha channel represents the distance value.

The third step performs the actual single scattering computation based on the above outputs (Algorithm 1). As shown in Figure 3, for each ray starting at C in a direction ω_{out} through a pixel, we first intersect the spot light cone to determine its entry and exit points p_{in} and p_{out} . Those points are then tested against the distance to the nearest solid object to account for potential occlusion, yielding actual entry and exit points. Finally, the ray between p_{in} and p_{out} is sampled to solve Equation 4 numerically: for each sample point \mathbf{p}_n , we compute the light source contribution by fetching the corresponding shadow map. If \mathbf{p}_n is unoccluded, the reduced intensity is computed using Equation 5 and a possible projective texture. The contribution of point \mathbf{p}_n to the single scattering is then added using Equation 4.

While this technique is effective and easily implementable on graphics hardware, two main aspects can be improved: the control on the shadow map sampling, and the appearance of the participating medium outside the light cone. We solve those issues using *projective space sampling*, and *scattered ambient lighting*.

4.2 Projective Space Sampling

The most trivial way of sampling points between the cone entry and exit points consists in picking sample points uniformly spaced between p_{in} and p_{out} . However, as shown in Figure 4, the density of actual shadow sampling points in the shadow map varies significantly depending on the viewing direction. A pathological case is presented in Figure 4(a), in which the sampling density of the

Algorithm 1 Base Volumetric Shadow Mapping

```

Generate Shadow Map
Compute reflected radiance  $L$  and distance information  $d, \forall$  pixel
for all pixels do
    Determine direction  $\omega_{out}$  from  $C$  through the pixel
    Fetch  $L$  and  $d$ , the reflected radiance and distance to the nearest solid object
    Intersect the corresponding ray with the light cone
    if the ray intersects the cone then
        Compute the entry and exit points  $p_{in}$  and  $p_{out}$ 
        if  $d < \|Cp_{in}\|$  then
            return  $L$ 
        end if
        if  $d < \|Cp_{out}\|$  then
             $p_{out} = C + d\omega_{out}$ 
        end if
         $L_{scat} = 0$ 
        for all sample points  $p_n$  along  $p_{in}p_{out}$  do
            Fetch the corresponding shadow map texel
            if  $p_n$  is unoccluded then
                 $L_{scat} +=$  single scattering contribution of  $p_n$ 
            end if
        end for
        Compute  $L_{reduced} = Le^{-\sigma_t \|p_{in}p_{out}\|}$ 
        return  $L_{scat} + L_{reduced}$ 
    else
        return  $L$ 
    end if
end for

```

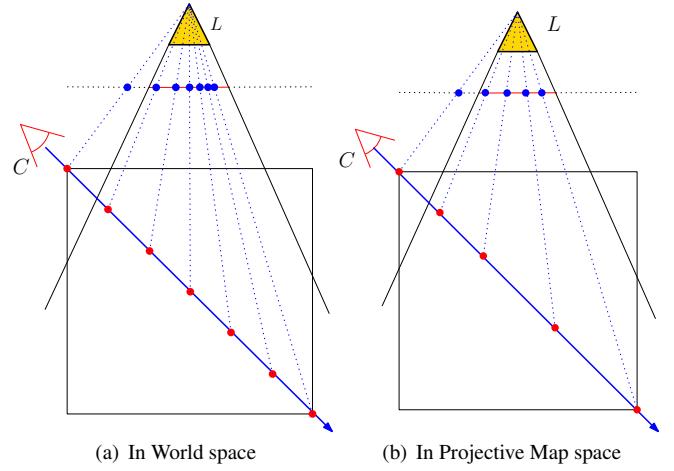


Figure 4: Uniform sampling in world space (a) involves precision issues due to the varying projective map sampling rate along the rays. We overcome this problem by uniform sampling in projective map space (b), providing a constant shadow sampling quality along the entire rays.

shadow map is scarce near the viewpoint, and increases as the distance to the viewpoint gets higher. Consequently, obtaining a satisfying sampling quality near the viewpoint implies an unnecessary oversampling in farther parts of the medium.

To provide an intuitive quality control, we propose to maintain a constant sampling quality over the shadow map using *projective space sampling*: the entry and exit points p_{in} and p_{out} are first projected into the shadow map space, yielding p'_{in} and p'_{out} . Then, projected sample points p'_n are obtained by uniformly subdividing

the segment $\mathbf{p}'_{in}\mathbf{p}'_{out}$. As shown in Figure 4(b), the sample points \mathbf{p}'_n are then projected back in world space to compute their scattering contribution.

This method is particularly useful in scenes containing high frequency details, or when using high resolution projective textures as this directly links the number of samples to the actual sampling rate of the projective texture. As shown in Figure 5, our projective space sampling tends to capture the details of the projective maps with more accuracy than classical sampling in world space, especially when the lighting and viewing directions are nearly collinear.

4.3 Scattered Ambient Lighting

A common artifact of volumetric methods is the lack of coherent scattering and attenuation effects outside the light cone. In the real world, such effects are due to multiple scattering events in the participating medium, and global illumination. Many real-time applications approximate global illumination effects using a simple ambient lighting term, added indiscriminately to each shaded point. As our method only handles single scattering and direct lighting, we propose to extend the principle of ambient lighting to homogeneous participating media. Let us recall Equation 2 which describes the single scattering:

$$L_{\mathbf{p}_n}(\mathbf{p}_{in}, \omega_{out}) = Q(\mathbf{p}_n, \omega_{out}) e^{\int_{\mathbf{p}_n}^{\mathbf{p}_{in}} -\sigma_t(\mathbf{p}) d\mathbf{p}} \quad (6)$$

We introduce a scattered ambient lighting term L_a corresponding to the radiance reaching every point in the medium from every direction. In this context, the integral can be solved analytically over a path from \mathbf{C} to a point \mathbf{p} within the medium:

$$L_{\mathbf{p}, \omega_{out}} = \frac{\sigma_s}{\sigma_t} L_a (1 - e^{-\sigma_t \|\mathbf{Cp}\|}) \quad (7)$$

As this formulation is coherent with the definition of single scattering in [Chandrasekhar 1950], *scattered ambient lighting* can be seamlessly combined with the algorithm described above as shown in Figure 6.

In this Section, we presented our method for fast and efficient single scattering simulation. We leveraged the assumption of homogeneous participating media to compute the reduced intensity of the light source analytically. The next Section introduces *transmittance function maps* to raise this assumption, hence extending our approach to participating media with varying opacities.

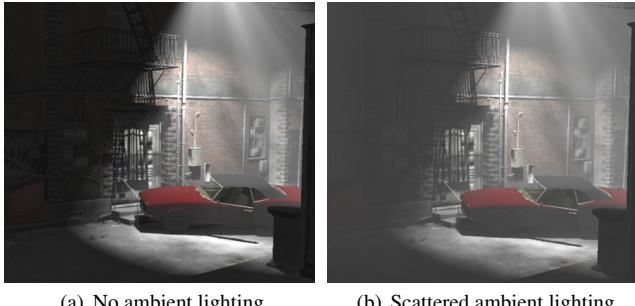


Figure 6: Scattered ambient lighting provides a coherent attenuation outside the light cone.

5 Transmittance Function Mapping

The usual bottleneck of ray-marched scattering simulations in heterogeneous participating media is the computation of the reduced

intensity L_{ri} at each sample point. This reduced intensity can be obtained by ray-marching the medium from the sample towards the light source or using projective texturing techniques such as deep shadow maps [Lokovic and Veach 2000] or Fourier opacity maps [Jansen and Bavoil 2010] for higher performance.

The principle of transmittance function maps directly builds upon the volumetric shadow mapping technique described in the previous section: our aim is to enrich the shadow map with additional information regarding light attenuation along each light ray. To this end, in a way similar to the deep shadow map approach, we sample the medium along light rays. However, instead of explicitly storing a piecewise linear combination of opacity samples, we choose to leverage the continuity and relative smoothness of the transmittance function and project it into a small set of coefficients of a functional basis. Therefore, given a set of basis functions $\{B_j(x)\}_{j \in \mathbb{N}}$, the transmittance function $T(x)$ at a distance x is:

$$T(x) = \sum_j c_j B_j(x) \quad (8)$$

$$c_j = \int T(x) B_j(x) dx \quad (9)$$

Algorithm 2 Transmittance Function Map generation

```

Set the viewpoint as for shadow map generation
for all pixels do
    Set  $c_j = 0 \forall j \in [0..m]$ 
    Determine direction  $\omega_{in}$  from  $\mathbf{C}$  through the pixel
    Fetch  $d$ , the distance to the nearest solid object
    Intersect the ray with the bounding box of the medium
    if the ray intersects the box then
        Compute the entry and exit points  $\mathbf{k}_{in}$  and  $\mathbf{k}_{out}$ 
        if  $d < \|\mathbf{Lk}_{in}\|$  then
            return  $c_j = 0 \forall j \in [0..m]$ 
        end if
        if  $d < \|\mathbf{Lk}_{out}\|$  then
             $\mathbf{k}_{out} = \mathbf{L} + d\omega_{in}$ 
        end if
        for all sample points  $\mathbf{k}_n$  along  $\mathbf{k}_{in}\mathbf{k}_{out}$  do
            Fetch density at  $\mathbf{k}_n$  and compute transmittance  $T_n$ 
             $\delta_K = \|\mathbf{k}_{n-1}\mathbf{k}_n\|$ 
            for all  $j \in [0..n]$  do
                 $c_j + = T_n \cos(\frac{j\pi}{2M}(2x + 1)) \delta_K$ 
            end for
        end for
        end for
        return  $\{c_j\}_{j \in [0..m]}$ 
    else
        return  $c_j = 0 \forall j \in [0..m]$ 
    end if
end for

```

In practice, we solve Equation 9 by marching through the medium and performing numerical integration (Algorithm 2). While the transmittance function could be projected into any functional basis, we chose the Discrete Cosine Transform with regards to its ease of evaluation and to the quality of the reconstructed signal even with a small set of coefficients:

$$B_j(x) = \cos\left(\frac{j\pi}{2M}(2x + 1)\right) \quad (10)$$

This set of coefficients can be simply stored in multiple floating-point render targets right after the shadow map generation.

The final rendering follows Algorithm 1. However, instead of computing the reduced intensity analytically, we fetch the transmittance

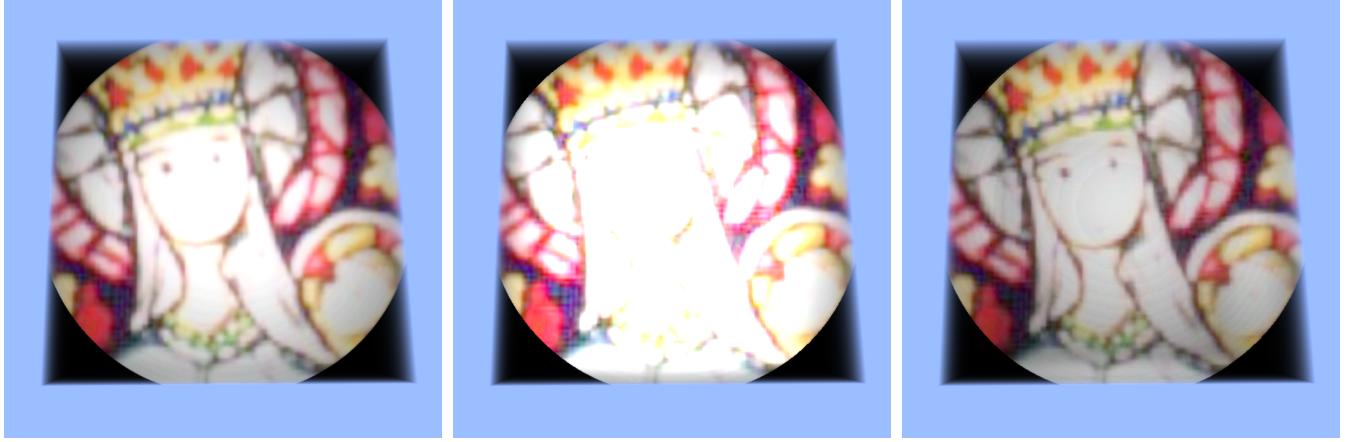


Figure 5: Homogeneous cube lit by a projective texture. The reference image (a) is obtained by a dual ray-marching with 1000 sample per view ray, the image (b) is obtained by a ray-marching in world space with 10 samples, and the image (c) is obtained by a ray-marching in shadow map space with 10 samples.

function map coefficients in the corresponding texel and reconstruct the transmittance at each sample point. This transmittance is then multiplied by the light intensity to obtain the reduced intensity.

This method provides a simple and accurate way of representing the variations of light attenuation in a heterogeneous medium using a small set of projection coefficients in the DCT basis. However, such coefficients have to be computed and stored for each represented wavelength, hence requiring a significant amount of memory space and bandwidth. Also, as pointed out in [Jansen and Bavoil 2010], the representation of transmittance or opacity in a functional basis results in potential ringing artifacts if the medium features high densities. The next Section provides simple solutions to those issues, hence improving the efficiency and genericity of our method.

6 Discussion

6.1 Wavelength Dependence

As described in the previous Section, the transmittance function must be projected and evaluated for each represented wavelength, namely RGB in most contexts. Even using only 4 coefficients per texel, the transmittance function map generation would output 3 floating-point RGBA textures which, in turn, will be sampled during rendering. We propose to reduce the memory footprint of our method by reducing the scope to a subset of participating media, in which the absorption and scattering coefficients are:

$$\sigma_a(\mathbf{p}) = D(\mathbf{p})\sigma_a \quad \text{and} \quad \sigma_s(\mathbf{p}) = D(\mathbf{p})\sigma_s \quad (11)$$

where $D(\mathbf{p})$ is a scalar medium density, and σ_a, σ_s are constant over the entire medium. Basically, the considered participating media is made of a single material with varying density. This is particularly useful in the context of cloud and smoke rendering. Within this context, we reformulate the transmittance function as follows:

$$\begin{aligned} T(\mathbf{k}_n) &= e^{-\int_{\mathbf{k}_{in}}^{\mathbf{k}_n} \sigma_t(\mathbf{k}) d\mathbf{k}} \\ &= (e^{-\int_{\mathbf{k}_{in}}^{\mathbf{k}_n} D(\mathbf{k}) d\mathbf{k}})^{\sigma_t} \end{aligned} \quad (12)$$

Therefore, instead of projecting the entire transmittance function, we only project the wavelength-independent part of the above equation, that is

$$T_{proj}(\mathbf{k}_n) = e^{-\int_{\mathbf{k}_{in}}^{\mathbf{k}_n} D(\mathbf{p}) d\mathbf{p}} \quad (13)$$

The actual, wavelength-dependent transmittance can is then:

$$T(\mathbf{k}_n) = (T_{proj}(\mathbf{k}_n))^{\sigma_t} \quad (14)$$

The advantages of this technique are twofold: on the one hand, the number of required coefficients is reduced, hence reducing the memory space and bandwidth. On the other hand, once the transmittance map is generated, the wavelength-dependent scattering and absorption coefficients can be modified on-the-fly without requiring any update in the transmittance map. This aspect is of particular importance in the context of offline production rendering: avoiding the regeneration of the transmittance map allows the medium designers to get a very fast feedback on changes of scattering properties, hence speeding up the design workflow.

6.2 High Density Medium

The use of basis functions provides a compact and smooth representation of the transmittance function. However, as also pointed out in [Jansen and Bavoil 2010], such methods tend to generate artifacts in the case of density variations in optically thick participating media. The oscillations in the reconstructed signal yield ‘stripping’ effects, as shown in Figure 7. Based on the following observations we introduce a density weighting to overcome this problem:

- artifacts are nearly invisible in optically thin media
- The transmittance values remain within $[0, 1]$
- The transmittance function is a continuous, decreasing function with respect to distance

Based on these observations, our aim is to losslessly smooth out the represented signal, so that artifacts would be less likely to be visible. This is equivalent to decreasing the amount of variations in the represented transmittance, that is, reducing the amplitude of the derivative of the signal. Let us consider a function $f(x)$ meeting the same characteristics as the transmittance function, and make the following assumption:

$$\frac{\partial f(x)^{\frac{1}{\alpha}}}{\partial x} < \frac{\partial f(x)}{\partial x} \quad (15)$$

where $\alpha > 1$ is an arbitrary scalar value. To validate this assumption, we aim at proving that increasing α results in decreasing $\frac{\partial f}{\partial x}$. The derivative of f after exponentiation is:

$$\frac{\partial f(x)^{\frac{1}{\alpha}}}{\partial x} = \frac{1}{\alpha} f(x)^{\frac{1}{\alpha}-1} \frac{\partial f(x)}{\partial x} \quad (16)$$

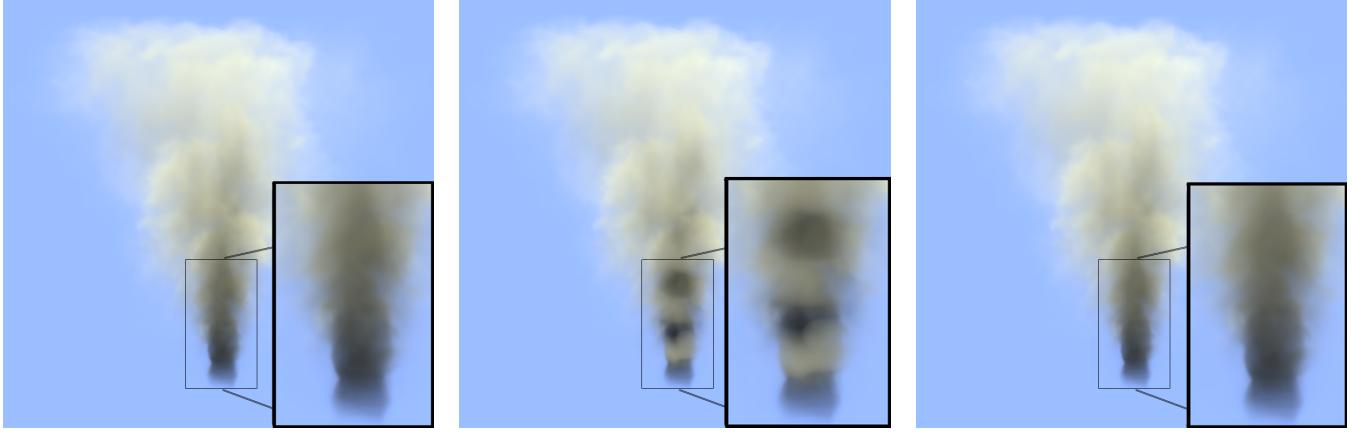


Figure 7: Compared to a reference solution (a), density weighting (DW) drastically reduces the noticeability of ringing artifacts in dense media.

The relationship between this derivative and α can be expressed by its derivative with respect to α . If the hypothesis is true, the derivative below is negative.

$$\frac{\partial^2 f(x)}{\partial \alpha \partial x} = \frac{\partial}{\partial \alpha} \left(\frac{(f(x)^{\frac{1}{\alpha}} \frac{\partial f(x)}{\partial x})(\alpha + \ln(f(x)))}{f(x)\alpha^3} \right) \leq 0 \quad (17)$$

The function $f(x)$ being positive and decreasing, we deduce $f(x)^{\frac{1}{\alpha}} \geq 0$ and $\frac{\partial f(x)}{\partial x} \leq 0 \forall x$. Consequently, $f(x)^{\frac{1}{\alpha}} \frac{\partial f(x)}{\partial x} \leq 0$ and $f(x)\alpha^3 \geq 0$.

Therefore, to validate the hypothesis we must have:

$$\alpha + \ln(f(x)) \geq 0 \quad \forall x \Leftrightarrow \max f(x) \geq e^{-\alpha} \quad (18)$$

In our context $f(x)$ is the transmittance function. In optically thick media, the transmittance function satisfies the above equation on most of its domain, making an exponentiation with a high α particularly useful. Conversely, optically thinner media would benefit from lower values of the factor α .

Based on this fact, we insert this user-defined density weighting α in our computations by rewriting the transmittance function as:

$$T^w(\mathbf{k}_n) = e^{-\int_{\mathbf{k}_{in}}^{\mathbf{k}_n} \sigma_t \frac{D(S)}{\alpha} dS} \quad (19)$$

The actual function is then obtained directly using the exponential property:

$$T(\mathbf{k}_n) = (T^w(\mathbf{k}_n))^{\alpha} \quad (20)$$

As shown in Figure 7(c), this factor allows us to significantly reduce the artifacts even in the case of low albedo media such as smoke.

7 Results

The techniques presented in the previous sections describe a full featured solution for fast rendering both homogeneous and heterogeneous participating media. This section illustrates some results obtained using an Intel Xeon 3.6GHz and a nVidia GeForce GTX480 GPU for real-time rendering using OpenGL, and Pixar's RenderMan® for production-quality rendering.

In order to compare objectively our images to these references, we use the SSIM (*Structural SIMilarity*) measure as defined by [Wang et al. 2004]. The SSIM scores range from -1 to 1, 1 being only reachable while comparing two identical images.

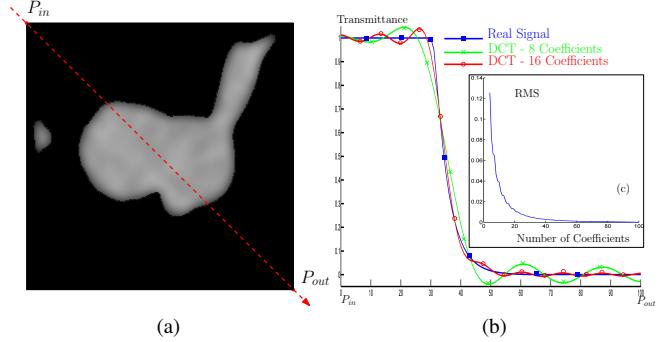


Figure 8: The transmittance function along a path in a medium (a) is projected into a set of coefficients (b). Note that the reconstruction error drops rapidly when increasing the number of coefficients.

7.1 Volumetric Shadow Mapping

Our approach offers real-time performance, as shown in Figure 9. The images were rendered at a resolution of 1280×720 using 1024^2 shadow maps and projective textures. The capture of projective texture and shadow details is achieved using 200 marching steps per pixel for the final rendering. The same algorithm has been implemented as a RenderMan shader for rendering the underwater city of the movie GI-Joe: Rise of Cobra, featuring ~ 3000 light sources: our method seamlessly supports an arbitrary number of light sources, provided the light cones are sampled for each traversing ray. In this case, depending on the geometry and lighting complexity, the number of marching steps could be reduced to ~ 50 , hence allowing the scene designers to insert additional light sources by reducing the per-light computational costs.

7.2 Transmittance Function Mapping

The Transmittance Function Mapping method offers real-time performance on 128^3 volumetric data described by density values (Figure 1(b)), and interactive performance on higher definition volumes (Figure 1(a)). As shown in the accompanying video, the rendered volumes can be arbitrarily animated as our method does not require any precomputation. In the remainder of this section, we compare our method with reference images obtained using a brute force ray marching algorithm to estimate the transmittance func-



Figure 9: Volumetric Shadow Mapping - 200 steps - 30 fps

Smoke - 128 ³ voxels - Figures 10 (a-d)						
	FPS	SSIM		PSNR		
		DW	No DW	DW	No DW	
TSM 4 coeffs	15.10	99.73%	87.12%	54.52	17.64	
TSM 8 coeffs	14.78	99.74%	95.49%	58.58	25.50	
TSM 16 coeffs	12.18	99.74%	95.35%	60.02	25.60	
Bunny - 256 ³ voxels - Figures 10 (e-h)						
	FPS	SSIM		PSNR		
		DW	No DW	DW	No DW	
TSM 4 coeffs	14.41	98.68%	79.77%	35.61	16.74	
TSM 8 coeffs	14.20	99.03%	82.57%	43.27	15.07	
TSM 16 coeffs	13.98	99.05%	77.55%	46.19	14.30	

Table 1: Objective comparison between reference images and the TFM technique with and without Density weighting (DW = 10) using 100 samples/ray and a 1024² Transmittance Function Map.

tion. Note that closely related methods such as [Jansen and Bavoil 2010; Lokovic and Veach 2000] aim at representing the *opacity* of a medium, and generally overlook the effects of light scattering. As the results obtained with such methods are intrinsically different from ours, we do not compare them with our approach.

7.2.1 Number of coefficients

The number of projection coefficients for the transmittance function has a significant impact on the quality of the reconstruction (Figure 8). An insufficient number of coefficients yields an overly smooth reconstruction, hence missing higher frequency details in the volume. As shown in Table 1, our *Transmittance Function Mapping* technique provides satisfying results, both in subjective and objective terms using the SSIM and PSNR measures. In our tests, the images generated using a Transmittance Function Map containing at least 8 coefficients per texel are objectively almost indistinguishable from the reference solution. Also, while a lower number of coefficients yields images objectively different from the ground truth, such images remain visually plausible (Figure 10(b,f)). Com-

bined to the inherent scalability of the ray marching algorithm, this allows our algorithm to be effective on a wide range of graphics hardware by adjusting the marching steps and the number of coefficients. Also, the Bunny volume features sharp density changes, as well as large zones with constant density. Even though this volume could be considered a pathological case for our technique, Figure 10 and Table 1 show that differences with the reference images remain very small.

8 Conclusion

Interactive simulation of complex participating media is a complex matter with no simple solutions. We proposed a full-featured approach to single scattering in both homogeneous and heterogeneous media using volumetric shadow mapping and attenuation function maps. Based on projective texturing and a Fourier representation of transmittance data, our method does not rely on any assumptions on the medium or on the lighting and viewing conditions. As this paper describes a scalable approach to the single scattering simulation problem, the introduced solution finds its use in both real-time and production rendering, making it suitable for a wide range of applications such as video games, asset previewing, or final post-production rendering. Future work will particularly consider the extension to the simulation of multiple scattering events in Fourier space based on the transmittance function map. Also, we would like to widen the concept of transmittance function maps to image-based lighting for enhanced realism.



Figure 11: Cloud 600×200×400 - 7fps

References

- CEREZO, E., PEREZ, F., PUEYO, X., SERON, F., AND SILLION, F. 2005. A survey on participating media rendering techniques. *The Visual Computer* 21, 5, 303–328.
- CHANDRASEKHAR, S. 1950. Radiative transfer. Clarendon Press, Oxford.
- DELALANDRE, C., GAUTRON, P., MARVIE, J.-E., AND FRANÇOIS, G. Single scattering in heterogenous participating media. In *ACM SIGGRAPH 2010 Talks*, SIGGRAPH '10.
- GAUTRON, P., MARVIE, J.-E., AND FRANÇOIS, G. 2009. Volumetric shadow mapping. In *SIGGRAPH 2009 talks*.
- HARRIS, M. J. 2005. Real-time cloud simulation and rendering. In *SIGGRAPH 2005 Courses*.
- JANSEN, J., AND BAVOIL, L. 2010. Fourier opacity mapping. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, 165–172.

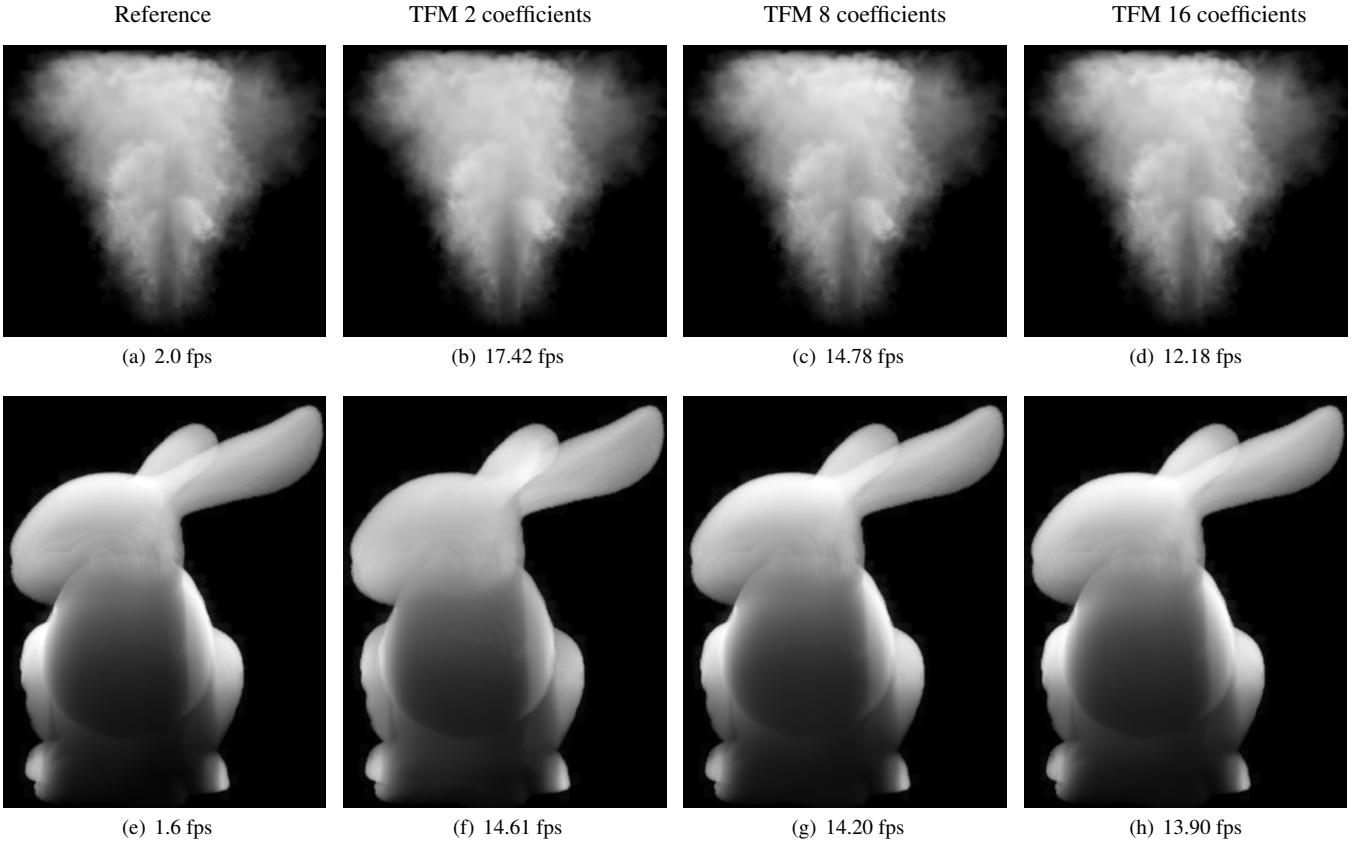


Figure 10: Quality assessment between a dual ray-marching algorithm using 100 steps/view ray with 100 reduced intensity steps/sample point and our algorithm using 100 steps/view ray and a 1024×1024 Transmittance Function Map with 2, 8 and 16 DCT coefficients.

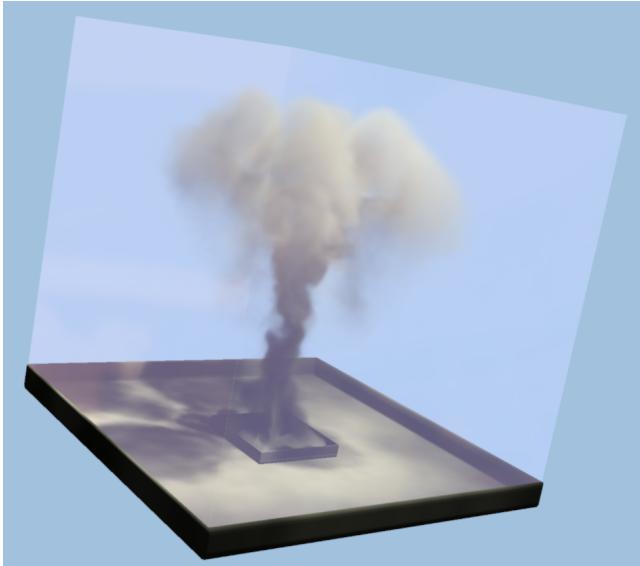


Figure 12: Dynamic Smoke $600 \times 200 \times 400$ - 25fps

KIM, T.-Y., AND NEUMANN, U. 2001. Opacity shadow maps. In *Proceedings of the Eurographics Workshop on Rendering*, 177–182.

- KNISSL, J., PREMOZE, S., HANSEN, C., SHIRLEY, P., AND MCPHERSON, A. 2003. A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics* 9, 2, 150–162.
- LOKOVIC, T., AND VEAUCH, E. 2000. Deep shadow maps. In *Proceedings of SIGGRAPH*, 385–392.
- PEGORARO, V., AND PARKER, S. 2009. An analytical solution to single scattering in homogeneous participating media. *Proceedings of Eurographics* 28, 2, 329–335.
- SUN, B., RAMAMOORTHI, R., NARASIMHAN, S. G., AND NAYAR, S. K. 2005. A practical analytic single scattering model for real time rendering. *Proceedings of SIGGRAPH* 24, 3, 1040–1049.
- WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4, 600–612.
- WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In *Proceedings of SIGGRAPH* 12, 3, 270–274.
- WYMAN, C., AND RAMSEY, S. 2008. Interactive volumetric shadows in participating media with single-scattering. *Proceedings of IEEE Symposium on Interactive Ray Tracing*, 87–92.
- ZHOU, K., REN, Z., LIN, S., BAO, H., GUO, B., AND SHUM, H.-Y. 2008. Real-time smoke rendering using compensated ray marching. In *Proceedings of SIGGRAPH*, 1–12.