# Bringing Transmittance Function Maps to the Screen: Wrath of the Titans and Prometheus

Jose Esteve\* MPC Jamie Portsmouth<sup>†</sup> MPC Pascal Gautron<sup>‡</sup> Technicolor

Jean-Colas Prunier<sup>§</sup> MPC Jean-Eudes Marvie<sup>¶</sup> Technicolor

Cyril Delalandre<sup>||</sup> Technicolor



**Figure 1:** Scenes from Wrath of the Titans (a) and Prometheus (b) involve extremely dense volumetric objects. We adapt the Transmittance Function Mapping algorithm for high quality interactive previsualization and tuning of those media. Images: (a)  $(\bigcirc 2012$  Warner Bros. Entertainment, (b)  $(\bigcirc 2012 \ 20^{th}$  Century Fox.

# Abstract

Participating media are an unavoidable part of todays visual effects. The computation of compelling lighting effects within clouds or smoke remains challenging, both in terms of memory occupancy and computational power. Also, the fine tuning and layout of production-quality scenes requires efficient techniques for fast previsualization of the results. The Transmittance Function Maps provide an efficient solution for real-time previsualization of relatively wispy media such as clouds. However, this technique cannot support the extremely high densities encountered within the pyroclastic clouds of Wrath of the Titans, or in the sandstorm of Prometheus. We propose an adaptation of the Transmittance Function Mapping technique for the interactive previsualization of extremely dense, production-quality participating media. Based on a dual ray marching approach, our technique provides significant quality improvements while preserving real-time performance.

# 1 Introduction

One of the challenges of digital visual effects is the generation of compelling sequences involving massive, highly detailed volumetric elements. Besides the costs of simulation, the accurate estimation of light transport within such participating media is crucial for producing lifelike digital images. This estimation is generally expensive, both in terms of memory and rendering time. Therefore, the interactive tuning and arrangement of volumetric objects is also of primary importance to avoid numerous overnight test renders.

Several approaches address the problem of real-time rendering of participating media by introducing drastic assumptions, limitations or precomputations. However, in the context of production rendering the images produced interactively must be reasonably consistent with the final renders and not require excessive precomputations. Building upon the existing Transmittance Function Mapping technique [Delalandre et al. 2011], we adapted this algorithm for interactive rendering of the Wrath of the Titans' pyroclastic clouds and Prometheus' sand storms.

# 2 Background

This section provides technical background on scattering computation, and discusses related previous work on lighting simulation in participating media.

#### 2.1 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. For performance reasons we focus on single scattering in the remainder of this section. A medium is described at each point  $\mathbf{p}$  by the following functions:

- The absorption coefficient  $\sigma_a(\mathbf{p})$  represents the fraction of incoming lighting which gets transformed into other forms of energy, such as heat.
- The scattering coefficient σ<sub>s</sub>(**p**) is the fraction of incoming lighting scattered at **p** in other directions.
- The extinction coefficient σ<sub>t</sub>(**p**) = σ<sub>a</sub>(**p**) + σ<sub>s</sub>(**p**).
- The phase function p(p, ω<sub>out</sub>, ω<sub>in</sub>) describes the amount of light scattered at p from the incoming direction ω<sub>in</sub> into outgoing direction ω<sub>out</sub>.

To define single scattering, let us first consider a point  $\mathbf{p_n}$  within the medium, and a scattering direction  $\omega_{out}$ . Given a lighting inten-

<sup>\*</sup>e-mail:jose-e@moving-picture.com

<sup>&</sup>lt;sup>†</sup>e-mail:jamie-p@moving-picture.com

<sup>&</sup>lt;sup>‡</sup>e-mail:pascal.gautron@technicolor.com

<sup>&</sup>lt;sup>§</sup>e-mail: jean-cp@moving-picture.com

<sup>¶</sup>e-mail:jean-eudes.marvie@technicolor.com

e-mail: cyril.delalandre@technicolor.com

sity  $L_{ri}$  incoming at  $\mathbf{p_n}$  from a direction  $\omega_{in}$  (Figure 2), the single scattering contribution to the radiance from a differential segment around  $\mathbf{p_n}$  is:

$$Q(\mathbf{p}_{\mathbf{n}},\omega_{\mathbf{out}}) = \sigma_s(\mathbf{p}_{\mathbf{n}})p(\mathbf{p}_{\mathbf{n}},\omega_{\mathbf{out}},\omega_{\mathbf{in}})L_{ri}(\mathbf{p}_{\mathbf{n}},\omega_{\mathbf{in}}).$$
 (1)

The contribution of  $\mathbf{p_n}$  to the radiance outgoing at  $\mathbf{p_{in}}$  is then:

$$L_{\mathbf{p}_{\mathbf{n}}}(\mathbf{p}_{\mathbf{i}\mathbf{n}},\omega_{\mathbf{out}}) = Q(\mathbf{p}_{\mathbf{n}},\omega_{\mathbf{out}})e^{\int_{\mathbf{p}_{\mathbf{n}}}^{\mathbf{p}_{\mathbf{i}\mathbf{n}}} -\sigma_{t}(\mathbf{p})d\mathbf{p}}.$$
 (2)

The integral exponent in this equation accounts for the light attenuation along the path from  $p_n$  to  $p_{in}$  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  $[\mathbf{p_{in}}, \mathbf{p_{out}}]$ ; for each sample  $\mathbf{p_n}$  a second ray-marching is performed along  $[\mathbf{k_{in}}, \mathbf{p_n}]$  to compute the light reduced intensity.

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

$$L_{ri}(\mathbf{p_n},\omega_{\mathbf{in}}) = IT(\mathbf{p_n},\mathbf{k_{in}}) \tag{3}$$

where  $\mathbf{k_{in}}$  is the intersection of the light ray with the bounding box of the medium and  $T(\mathbf{p_n}, \mathbf{k_{in}})$  is the transmittance of the medium along the path between  $\mathbf{p_n}$  and  $\mathbf{k_{in}}$ ):

$$T(\mathbf{p}_{\mathbf{n}}, \mathbf{k}_{\mathbf{in}}) = e^{-\int_{\mathbf{p}_{\mathbf{n}}}^{\mathbf{k}_{\mathbf{in}}} \sigma_t(\mathbf{p}) d\mathbf{p}}.$$
 (4)

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

$$L(\mathbf{p_{in}},\omega_{out}) = \int_{\mathbf{p_{in}}}^{\mathbf{p_{out}}} L_{\mathbf{p}}(\mathbf{p_{in}},\omega_{out}) \mathrm{d}\mathbf{p}$$
(5)

#### 2.2 Related Work

Research towards efficient rendering techniques for participating media has produced numerous valuable contributions, some of which are described in [Cerezo et al. 2005; Engel et al. 2006; Wrenninge et al. 2011; Jönsson et al. 2012].

The usual bottleneck of ray-marched single 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 brute force ray-marching the medium from the sample towards the light source or, more efficiently, using projective texturing techniques.

For production rendering, Deep Shadow Maps [Lokovic and Veach 2000] remains the method of choice for representing transmittance functions. By storing a linked list of all the volume samples along light rays, they produce close-to-ideal shadowing effects. The sample lists can be efficiently quantized and filtered for higher performance. However this approach inherently requires the storage of an unpredictable amount of samples per pixel depending on the depth complexity of the scene. While not problematic in production renderers, real-time implementations of Deep Shadow Maps on graphics hardware typically introduce restrictions such as a fixed list size [Kim and Neumann 2001; Kniss et al. 2003; Johnson et al. 2009; Yang et al. 2010], potentially yielding visually disturbing artifacts especially in highly heterogeneous media. Rendering production-quality volumes also introduces challenges in terms of real-time storage, sorting, quantization and filtering of samples.

Conversely, the methods targeting real-time applications typically involve limitations such as homogeneous media [Mitchell 2005; Wyman and Ramsey 2008], heavy precomputations [Zhou et al. 2008] or highly approximate scattering simulation [Jansen and Bavoil 2010].

While the state of the art of volume visualization techniques contains numerous efficient approaches, the Transmittance Function Maps [Delalandre et al. 2011] provide a simple solution for interactive computation of single scattering in production-quality participating media. We describe this technique in the next section, before detailing the adaptations performed to address the specific issues encountered during the production of Wrath of the Titans and Prometheus.

### 3 Transmittance Function Maps

#### 3.1 Principle

The principle of transmittance function maps directly builds upon the shadow mapping technique [Williams 1978]: the 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, the medium is sampled along light rays. However, instead of explicitly storing a piecewise linear combination of opacity samples, this method leverages the continuity and relative smoothness of the transmittance function to 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:

$$T(x) = \sum_{j=0}^{\infty} c_j b_j(x), \qquad (6)$$

$$c_j = \int T(x)b_j(x)\mathrm{d}x. \tag{7}$$

In practice, Equation 7 is solved by marching through the medium and performing numerical integration (Algorithm 1). While the transmittance function could be projected into any functional basis, Transmittance Function Maps can be efficiently implemented Algorithm 1 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 L 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 an opaque object occludes the medium, stop the generation if  $d < ||\mathbf{Lk_{in}}||$  then return  $c_i = 0 \forall j \in \{0..m\}$ end if // If an opaque object lies within the medium, adjust the end point if  $d < \|\mathbf{lk}_{out}\|$  then  $\mathbf{k_{out}} = \mathbf{l} + d\omega_{in}$ end if  $D = \|\mathbf{k_{in}k_{out}}\|$ for all sample points  $k_n$  along  $k_{in}k_{out}$  do Fetch density at  $\mathbf{k_n}$  and compute transmittance  $T_n$  $\delta_k = \|\mathbf{k_{n-1}k_n}\|$ for all  $j \in \{0..n\}$  do  $c_j = c_j + T_n b_j(x) \delta_k$ end for end for **return**  $\{c_j\}_{j \in \{0..m\}}$ else **return**  $c_i = 0 \ \forall i \in \{0..m\}$ end if end for

using a decomposition into Fourier Cosine Series, in which the basis functions are:

$$b_j(x) = \frac{2}{D} \cos\left(\frac{j\pi x}{D}\right) \tag{8}$$

where D is the integration range, *ie* the length of the light path within the medium. The set of coefficients can be simply stored into multiple floating-point render targets right during the generation of the map.

The final rendering is performed by marching through the medium along camera rays. For each point the amount of incoming light  $L_{ri}$  is obtained by fetching the coefficients stored in the corresponding texel and evaluating the basis functions.

Our implementation of Transmittance Function Mapping offers real-time performance on production-quality volumetric data described by density values (Figure 3). The absence of precomputations allow for interactive tuning and layout of volumes.

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. However, as also 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.

# 4 Adaptation for Production

The production of Wrath of the Titans and Prometheus involved numerous volumetric effects (Figure 1), many of them including volcanic eruptions and sand storms. As opposed to clouds and smoke



**Figure 3:** *Production cloud (600×200×400) rendered in 12.6ms using Transmittance Function Maps.* 

in which particle densities tend to remain relatively low and within a limited range, these natural phenomena tend to result in locally extreme densities. In terms of the transmittance function, the gradual decrease usually encountered in clouds is shrunk into a steep ramp function (Figure 4).



**Figure 4:** Overall behavior of a transmittance function along a ray through a dense volume. Light enters the volume bounding box at  $x_i$ , and first hits the volume at  $x_s$ . The transmittance begins dropping at  $x_s$ , and falls below threshold at  $x_t$ . The transmittance continues to drop until the light leaves the volume at  $x_e$ , and then remains constant until the ray leaves the bounding box at  $x_0$ .

Transmittance Function Maps are based on the representation of transmittance using a relatively small set of smooth basis functions. While sufficient on relatively wispy media, the rapid transmittance decrease in dense volumes cannot be accurately encoded within this band-limited representation, even using the exponentiation technique of [Delalandre et al. 2011]. Visually, this inaccuracy translates into missing shadow features and lighting oscillations.

Based on this observation we modify the Transmittance Function Maps algorithm to meet the production requirements (Figure 5).

#### 4.1 Locally Dense Media

In uniformly dense participating media light only penetrates a thin outer layer before being absorbed. The general scattering problem then becomes analogous to subsurface scattering, for which highly efficient real-time techniques exist such as [François et al. 2008]. However, as shown in Figure 6, even extremely dense pyroclastic clouds can also be partly translucent, preventing a systematic use of subsurface scattering. Based on those observations we propose a simple method for adaptive determination of the depth of the illuminated layer depending on the local density of the considered media. Once determined, this depth is used to adjust the representation range of Transmittance Function Maps, hence improving the representation quality.

#### 4.2 Adaptive Sampling

Dealing with a finite set of projection coefficients imposes a limit on the maximum frequency in the reconstructed transmittance. The quality of the reconstruction therefore improves when high frequencies in the original function are avoided. As the density variations of the volumes cannot be modified, we propose to reduce the range over which the transmittance is encoded.

We demonstrate the effect of restricting the encoding region of the transmittance curves in Figure 7, using 6 Fourier series coefficients for the reconstructions. The projection of the entire transmittance function results in strong oscillations. Using an integration range tightly bounding the relevant transmittance variations, allows for a lossless smoothing of the original function, hence improving the



(a) Original TFM

(b) Adapted TFM

**Figure 5:** Real-time rendering of a 256<sup>3</sup> pyroclastic ash cloud with blackbody emission, rendered with transmittance encoding using 6 Fourier Series coefficients. The original Transmittance Function Mapping algorithm (a) produces softer illumination which lacks the high contrast of the rim light coming from the right side. Our adaptive algorithm (b) enhances the lighting features in the denser parts of the medium, while keeping the original quality in wispier areas.



Figure 6: The participating media used in production contain both low (left) and extreme densities (right). This diversity prevents the systematic use of approaches based on subsurface scattering, while being challenging for more general real-time volume rendering techniques such as Transmittance Function Maps.

quality of the reconstructed signal. However, the integration range depends on the medium properties as well as the lighting direction. Furthermore, for the sake of interactivity we cannot consider introducing a precomputation pass on each medium.

Based on these observations we introduce a dual ray marching approach for the generation of this adaptive Transmittance Function Map (Algorithm 2). For each light ray the entire medium is first ray marched as described in [Delalandre et al. 2011]: for each step the transmittance is evaluated and projected into the functional basis. During this ray marching we compare the transmittance to a user-defined threshold. If the transmittance remains above the threshold on the entire light path, the medium is considered moderately dense and does not undergo further operations.

If the transmittance drops below the user-defined threshold this first ray marching is stopped and the bounds of transmittance variations stored. A second ray marching is then performed within those bounds to finely sample and project the transmittance function. In our implementation we used 100 samples for each ray marching.

In a worst case scenario the transmittance threshold is reached after marching through the entire medium. The ray marching is then restarted from scratch, resulting in a severe performance hit. However, in practice the transmittance either decays rapidly (dense zones), or decreases slowly without reaching the threshold (wispy parts of the media). Therefore, the overhead of our method is negligible.

Algorithm 2 Dual Adaptive Ray Marching
$\mathbf{k}_{\min} = \mathbf{k}_{in}$
$\mathbf{k}_{\max} = \mathbf{k}_{out}$
isDenseMedium = false
// Initial ray marching through the entire medium
for all sample points $\mathbf{k}_{n}$ along $\mathbf{k}_{in}\mathbf{k}_{out}$ do
Fetch density $d_n$ at $\mathbf{k}_n$ and compute transmittance $T_n$
// Store the first point with nonzero density
if $\mathbf{k}_{\min} = \mathbf{k}_{in}$ and $d_n > 0$ then
$k_{min} = k_n$
end if
// Store the last point before $T_n$ drops below the threshold
if $T_n < T_{th}$ then
$\mathbf{k_{max}} = \mathbf{k_{n-1}}$
isDenseMedium = true
Break the marching
end if
Compute the sample contribution as in Algorithm 1
end for
// If needed, refined second marching
if isDenseMedium then
for all refined sample points $\mathbf{k_m}$ along $\mathbf{k_{min}k_{max}}$ do
Fetch density $d_m$ at $\mathbf{k_m}$ and compute transmittance $T_m$
Compute the sample contribution to the coefficients
end for
end if
Store the coefficients along with $k_{min}$ and $k_{max}$

#### 4.3 Adjustable Number of Coefficients

As mentioned above the quality of the function reconstructed from Transmittance Function Maps is very sensitive to the number of coefficients. A small number of coefficients preserve interactive performance at the detriment of quality, while a higher number closes the gap towards the reference solution. Therefore, in our implementation the number of coefficients can be adapted depending on



**Figure 7:** Our technique reduces the integration range of the Transmittance Function Map. A representation of the transmittance along the entire light ray through the medium introduces oscillations and loss of detail (top). Our transmittance falloff detection allows for a tight adaptation of the projection boundaries, hence improving the representation quality (bottom).



**Figure 8:** Adaptive Transmittance Function Maps have been integrated into the Maya viewport for interactive setup of volumetric scenes.

the tasks performed by the artists: while general tuning can be performed in real-time with few coefficients, higher quality rendering is triggered when working on fine volume tuning and lighting design. In practice the real-time mode requires only 6 coefficients, while 14 are used for high quality preview.

# 5 Application: Wrath of the Titans and Prometheus

We integrated this solution into the Moving Picture Company (MPC) volumetric toolset for interactive setup of the volumetric elements of Wrath of the Titans and Prometheus (Figure 8). The presented images have been generated using a Nvidia QuadroFX 1800 graphics processor, at an average frame rate of 20 fps using 6 coefficients.

Our algorithm automatically detects an optimal path length and adapts the transmittance function map representation accordingly. Our method being based on [Delalandre et al. 2011], the scattering properties of the volumes can be adjusted in real-time at no extra cost. Our solution has been used during the production, and providing preview results consistent with the final images rendered using Pixar's RenderMan (Figure 9).



(a) Interactive Previsualization

(b) Pixar's RenderMan

Figure 9: The interactive results provided by our technique (a) are consistent with the images obtained using Pixar's RenderMan and Deep Shadow Maps (b).



Figure 10: A single shot in Wrath of the Titans features many kinds of participating media, ranging from pyroclastic clouds to simpler smoke. Image ©2012 Warner Bros. Entertainment.

The pyroclastic clouds and explosions of Wrath of the Titans (Figures 1a and 10) have been setup using our approach, allowing the artists to finely tune the parameters of each volume without resorting to costly production renders. The production workflow for the sandstorm (Figure 1b) and explosions of Prometheus also benefited from our approach for fast adjustment and setup.

# 6 Conclusion

The Transmittance Function Maps algorithm can be used satisfactorily to render moderately dense participating media such as clouds and smoke. The core of this technique is based on the assumption of smooth transmittance functions. However, many volumetric assets used in production contain locally extreme densities, resulting in highly visible artifacts. We propose an adaptive sampling scheme for improved rendering. The transmittance function along light rays is first analyzed during the Transmittance Function Maps generation. If the medium is determined as extremely dense, our adaptive sampling scheme tightly bounds the relevant variations of transmittance within the medium to compute an enhanced representation of the transmittance function.

Our technique was successfully implemented in the MPC Volumetric Toolset and embedded within Autodesk's Maya viewport. Our technique particularly found its use for the tuning of the volumetric elements used in Wrath of the Titans and Prometheus.

While providing promising results and significant workflow optimization, the described technique considers the simultaneous rendering of a small number of participating media. Future work will consider extensions for the visualization of massive volumetric datasets using out-of-core techniques and innovative solutions for light scattering.

### References

- CEREZO, E., FRANCISCO, S., PEREZ, F., SILLION, F., AND PUEYO, X. 2005. A survey on participating media rendering techniques. *Visual Computer 21*, 5, 303–328.
- CHANDRASEKHAR, S. 1950. Radiative transfer. Clarendon Press, Oxford.
- DELALANDRE, C., GAUTRON, P., MARVIE, J.-E., AND FRANÇOIS, G. 2011. Transmittance function mapping. In *Proceedings of the 13D Symposium*, 31–38.
- ENGEL, K., HADWIGER, M., KNISS, J., REZK-SALAMA, C., AND WEISKOPF, D. 2006. *Real-time Volume Graphics*. AK Peters.
- FRANÇOIS, G., PATTANAIK, S., BOUATOUCH, K., AND BRE-TON, G. 2008. Subsurface texture mapping. *IEEE Comput. Graph. Appl.* 28, 1, 34–42.
- JANSEN, J., AND BAVOIL, L. 2010. Fourier opacity mapping. In Proceedings of the I3D Symposium, 165–172.
- JOHNSON, G. S., HUNT, W. A., HUX, A., MARK, W. R., BURNS, C. A., AND JUNKINS, S. 2009. Soft irregular shadow mapping: fast, high-quality, and robust soft shadows. In *Proceedings of the I3D Symposium*, 57–66.
- JÖNSSON, D., SUNDÉN, E., YNNERMAN, A., AND ROPINSKI, T. 2012. Interactive Volume Rendering with Volumetric Illumination. In *Eurographics STAR program*.
- KIM, T.-Y., AND NEUMANN, U. 2001. Opacity shadow maps. In *Proceedings of the Eurographics Workshop on Rendering*, 177–182.
- KNISS, 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 VEACH, E. 2000. Deep shadow maps. In *Proceedings of SIGGRAPH*, 385–392.
- MITCHELL, J. L. 2005. *ShaderX3: Light Shaft Rendering.* Charles River Media, 573–588.
- WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In Proceedings of SIGGRAPH 12, 3, 270–274.
- WRENNINGE, M., ZAFAR, N. B., HARDING, O., GRAHAM, G., TESSENDORF, J., GRANT, V., CLINTON, A., AND BOUTHORS, A. 2011. Production volume rendering. SIGGRAPH Courses.
- WYMAN, C., AND RAMSEY, S. 2008. Interactive volumetric shadows in participating media with single-scattering. In *Proceedings* of *IEEE Symposium on Interactive Ray Tracing*, 87–92.
- YANG, J. C., HENSLEY, J., GRN, H., AND THIBIEROZ, N. 2010. Real-time concurrent linked list construction on the GPU. *Computer Graphics Forum* 29, 4, 1297–1304.
- 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.