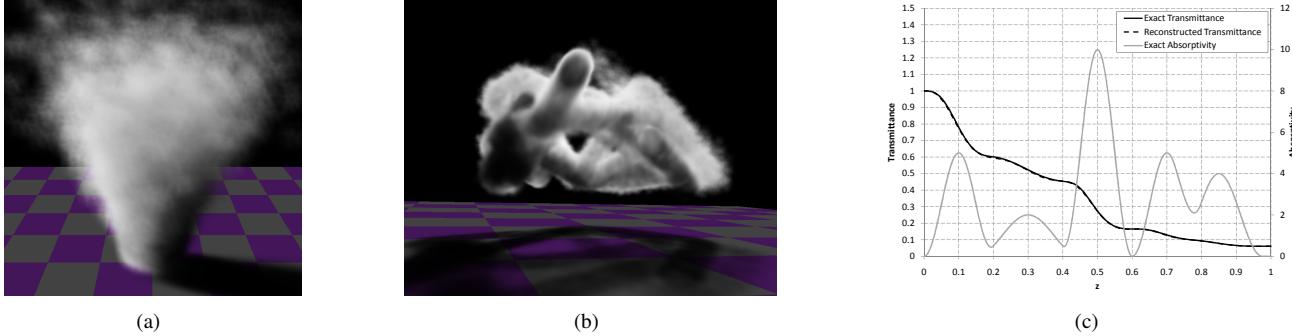


# Fourier Opacity Mapping

Jon Jansen

Louis Bavoil

NVIDIA Corporation



**Figure 1:** (a) A smoke plume rendered using Fourier Opacity Mapping (FOM). (b) A particle system with complex structure rendered using FOM. (c) Reconstructed transmittance using FOM with 15 coefficients. Note how the exact and reconstructed transmittance coincide almost exactly, despite the presence of multiple absorption features throughout the  $z$ -range.

## Abstract

Whilst the Deep Shadow Maps algorithm for accelerating the rendering of volumetric shadows is a fitting solution for offline applications, it can consume an unbounded amount of memory and is difficult to map well to current graphics hardware. For these reasons alternative methods have been proposed for interactive applications based on the idea of accumulating opacity from volume primitives in depth-based buckets or slices. However, these slice-based methods can introduce objectionable and unstable slice-like artefacts due to undersampling of the depth range, and often many slices are required to overcome these artefacts. Further refinements have been proposed to address this, but they constrain the generality of the algorithm, for example by mandating an assumption of uniform opacity or by dropping support for pre-filtering. We introduce a novel algorithm called Fourier Opacity Mapping (FOM) and we show that it is a good choice for rendering artefact-free pre-filtered volumetric shadows in cases where spatial opacity variations are smooth (e.g. smoke, gas and low-opacity hair). We also show how the algorithm can be generalised to other orthonormal bases.

**Keywords:** volumetric, shadows, opacity mapping, smoke, fog, particles, hair

## 1 Introduction

A volumetric shadow (or transmittance) is defined as the amount of light reaching a point in a volume along a light ray traced between the volume point and a light source, assuming that the light source is a point light. As with any shadows, soft shadows from area lights can be rendered by averaging the shadows from multi-

ple point samples on the area light. Scattering is ignored, that is, the light is assumed to travel inside the translucent volume along straight lines. Each translucent primitive that such a light ray hits along its path attenuates the current light visibility by the transmittance  $T_i$  of primitive  $i$  (where  $T = 1 - \text{opacity}$ ).

As with any algorithm based on shadow mapping, we rasterize the translucent primitives of the volume from the point of view of the light. Each pixel from this view corresponds to a light ray being traced from the light center through the volume. Unlike shadow mapping with opaque occluders, we have multiple primitives contributing shadows along a given light ray. Deep Shadow Maps [Lokovic and Veach 2000] is a state-of-the-art algorithm in offline rendering for rendering volumetric shadows. This algorithm captures all the primitives using a linked list per pixel of arbitrary size. The main issue with this algorithm for real-time rendering is the unbounded amount of memory that is required for capturing the primitives, as well as prefiltering the deep shadow map.

For real-time rendering, approximations of deep shadow maps have been proposed [Kim and Neumann 2001; Kniss et al. 2003; Yuksek and Keyser 2008; Sintorn and Assarsson 2008; Sintorn and Assarsson 2009]. These algorithms are based on the idea of discretizing the transmittance function along light rays using predefined depth buckets. They have a fixed memory cost and can run in real time. However, the fact that they discretize the transmittance function into depth buckets causes discontinuities in the rendered shadows along the  $z$  axis of the shadow map frustum. The challenge with this approach is how to limit undersampling artefacts when the density of depth buckets per light ray is not large enough to capture the complexity of the volume.

Inspired by Convolution Shadow Maps (CSM) [Annen et al. 2007; Annen et al. 2008], our algorithm reformulates the absorption function  $\sigma(z)$  as a Fourier series. For rendering shadows from opaque objects, CSMs require many coefficients to represent the sharp step function between lit and occluded. For volumetric shadows, the transmittance function that we want to evaluate is typically much smoother, especially when rendering media such as fog and smoke. Therefore, fewer Fourier coefficients are needed to faithfully represent a typical transmittance function with a truncated Fourier series.

Copyright © 2010 by the Association for Computing Machinery, Inc.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail [permissions@acm.org](mailto:permissions@acm.org).

I3D 2010, Washington, DC, February 19 – 21, 2010.  
© 2010 ACM 978-1-60558-938-1/10/0002 \$10.00

For a given light ray, the Fourier coefficients can be computed exactly (with floating-point precision) in a single rasterization pass by using additive blending. We store the first  $n$  coefficients into a shadow map called a Fourier Opacity Map. Instead of discretizing by projecting the  $\sigma(z)$  function onto a subdivision of depth, we discretize by projecting it onto the cosine and sine functions from a Fourier series. The key advantages of using a truncated Fourier series are that it is more stable under translation and less sensitive to variations of the depth range. Another advantage is that the truncated Fourier series is always smooth (infinitely differentiable actually), whereas slice-based reconstructions have slope discontinuities which translate into visible discontinuities in the shadows. As in the Convolution Shadow Maps algorithm, our approach also has the advantage of allowing rendering soft shadows by prefiltering the Fourier coefficients.

## 2 Related work

The Deep Shadow Maps algorithm [Lokovic and Veach 2000] pre-computes the light visibility function along grid-aligned light rays, generated using rasterization. First, all the shadow-casting primitives (opaque and translucent) are rasterized into a deep shadow map from the point of view of the light source. Each pixel contains a linked list of all the rasterized fragments covering this pixel, sorted in depth order relative to the light. Second, the deep shadow map is preprocessed. For each fragment in the deep shadow map, the transmittance  $T_i$  is computed and a piecewise-linear function is built which interpolates the  $(z_i, T_i)$  control points. The piecewise-linear functions per pixel are then pre-filtered exactly, which expands the number of control points. Finally, the number of control points per pre-filtered polynomial is reduced (compression step) to make the algorithm more practical. Pre-filtering is important to avoid aliasing (especially visible with primitives with  $\geq 50\%$  opacity) and fake subsurface scattering. [Hadwiger et al. 2006] generate and render deep shadow maps on the GPU for volume rendering.

Opacity Shadow Maps [Kim and Neumann 2001] is an approximation to deep shadow maps, in which the depth values of the visibility function per pixel are aligned on a regular grid. Unlike deep shadow maps, they can be generated efficiently on the GPU using Multiple Render Targets [Nguyen and Donnelly 2005] and rendering in depth buckets [Sintorn and Assarsson 2008]. Opacity shadow maps are equivalent to voxelizing the shadow frustum and keeping a density value for each voxel. The only difference is that the density values are pre-accumulated along light rays. Therefore, opacity shadow maps support pre-filtering. Blurring the opacity shadow map in the X, Y, and Z directions softens the shadows and removes layering artefacts. However, a large number of slices must be used along the Z direction to avoid layering artefacts (or loss of detail if filtering in the Z direction). Deep opacity maps [Yuksel and Keyser 2008] are a variation of opacity shadow maps where the depth slices are not aligned on a regular grid but instead relative to the nearest depth layer. Shifting the depth slices based on the nearest layer to the light removes most layering artefacts. Occupancy maps [Sintorn and Assarsson 2009] voxelize the density of hair between the nearest and farthest primitives for each shadow-map pixel. Because the depth slices are not aligned on a grid, pre-filtering deep opacity maps and occupancy maps does not produce soft shadows.

Another approach to rendering volumetric shadows is volume rendering using half-angle slicing [Kniss et al. 2003] [Green 2009]. This approach has the advantage of using less memory than deep opacity map approaches, but is expensive because it requires multiple geometry passes and render-target switches.

## 3 Our algorithm

### 3.1 Continuous formulation

We consider the journey of a ray from a point light source as it passes through the translucent medium. We assume that the usual projection into "shadow space" has already been set up, and that we can identify a ray by its  $(x, y)$  location in shadow space,  $\mathbf{r}$ , and further that a point on the ray is uniquely identified by a  $z$  value in the range  $[0, 1]$  (where 0 represents the closest point to the light source, and 1 the furthest).

Our goal is to determine the transmittance for a given depth,  $d$ , on a given ray,  $\mathbf{r}$ . The transmittance is defined as the ratio of light arriving directly at the given point, compared to the light entering the translucent medium along the ray:

$$T(\mathbf{r}, d) = \frac{I(\mathbf{r}, d)}{I_0(\mathbf{r})}$$

In common with other approaches, we neglect scattering effects: we consider only light arriving directly along a straight path from the light source. For the purposes of brevity and clarity, we will omit  $\mathbf{r}$  from our discussion from now on, but keep in mind that in all cases, "for the given ray,  $\mathbf{r}$ " is implied:

$$T(d) = \frac{I(d)}{I_0}$$

For sufficiently translucent media with constant absorption, Beer's Law can be used as an approximation to  $T(z)$ :

$$T(d) = \exp(-\sigma d)$$

We generalise this to the variable absorption case by replacing the  $\sigma d$  term with the integral of the absorption function along the path of the ray:

$$T(d) = \exp\left(-\int_0^d \sigma(z) dz\right)$$

We now observe that the absorption function  $\sigma(z)$  can be conveniently represented as a Fourier series. Recall that the canonical projection and reconstruction for a Fourier expansion is as follows:

$$\begin{aligned} a_k &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(kt) dt \\ b_k &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(kt) dt \\ f(t) &\approx \frac{a_0}{2} + \sum_{k=1}^n a_k \cos(kt) + \sum_{k=1}^n b_k \sin(kt) \end{aligned}$$

As we are working with respect to  $z$  in the range  $[0, 1]$ , we apply the substitution  $t = \pi(1 - 2z)$ , giving:

$$\begin{aligned} a_k &= 2 \int_0^1 \sigma(z) \cos(2\pi kz - k\pi) dz \\ b_k &= 2 \int_0^1 \sigma(z) \sin(2\pi kz - k\pi) dz \\ \sigma(z) &\approx \frac{a_0}{2} + \sum_{k=1}^n a_k \cos(2\pi kz - k\pi) \\ &\quad + \sum_{k=1}^n b_k \sin(2\pi kz - k\pi) \end{aligned}$$

We observe that  $\cos(\theta - k\pi) = (-1)^k \cos(\theta)$  and  $\sin(\theta - k\pi) = (-1)^k \sin(\theta)$ , so setting  $a'_k = (-1)^k a_k$  and  $b'_k = (-1)^k b_k$ , we get the following convenient formulation:

$$a'_k = 2 \int_0^1 \sigma(z) \cos(2\pi kz) dz \quad (1)$$

$$b'_k = 2 \int_0^1 \sigma(z) \sin(2\pi kz) dz \quad (2)$$

$$\sigma(z) \approx \frac{a'_0}{2} + \sum_{k=1}^n a'_k \cos(2\pi kz) + \sum_{k=1}^n b'_k \sin(2\pi kz)$$

We can now derive the integral of  $\sigma(z)$ :

$$\begin{aligned} \int_0^d \sigma(z) dz &\approx \frac{a'_0}{2} d + \sum_{k=1}^n \frac{a'_k}{2\pi k} \sin(2\pi kd) \\ &+ \sum_{k=1}^n \frac{b'_k}{2\pi k} (1 - \cos(2\pi kd)) \end{aligned} \quad (3)$$

An important property of this formulation is that it delivers the correct results at the limits of our z-range:

$$\begin{aligned} T(0) &= 1 \\ T(1) &= \exp \left( - \int_0^1 \sigma(z) dz \right) \end{aligned}$$

### 3.2 Discretization

In the case of real-world rendering applications, we are seldom presented with  $\sigma(z)$  in a form that allows us to readily compute  $a'_k$  and  $b'_k$  using analytical methods. Instead, it is more common for a translucent medium to be represented by a large number of translucent primitives, and for the absorption properties of the medium to be implied by the variable opacity of the primitives.

A useful starting point is to consider the action of a single primitive with opacity  $\alpha_0$  at depth  $d_0$ . The transmittance function for this single primitive case is given by:

$$T(d) = \begin{cases} 1 & \text{if } d < d_0 \\ 1 - \alpha_0 & \text{if } d \geq d_0 \end{cases}$$

We can reformulate this in terms of the generalised Beer's Law using the Dirac delta function:

$$T(d) = \exp \left( \int_0^d \ln(1 - \alpha_0) \delta(z - d_0) dz \right)$$

This leads naturally to the following further generalisation for the action of multiple primitives:

$$T(d) = \exp \left( \int_0^d \sum_i \ln(1 - \alpha_i) \delta(z - d_i) dz \right)$$

i.e.

$$\sigma(z) = - \sum_i \ln(1 - \alpha_i) \delta(z - d_i) \quad (4)$$

Substituting (4) into equations (1) and (2), and making use of the following property of the Dirac delta:

$$\int_{-\infty}^{\infty} f(x) \delta(x - c) dx = f(c)$$

We obtain:

$$a'_k = -2 \sum_i \ln(1 - \alpha_i) \cos(2\pi kd_i) \quad (5)$$

$$b'_k = -2 \sum_i \ln(1 - \alpha_i) \sin(2\pi kd_i) \quad (6)$$

## 4 GPU implementation

Our implementation consists of just two steps:

1. Render the primitives to one or more textures using equations (5) and (6) to generate the coefficients  $a'_k(\mathbf{r})$  and  $b'_k(\mathbf{r})$  (i.e. the "Fourier Opacity Map")
2. Render the primitives to the screen buffer, using equation (3) to calculate the transmittance from the coefficients  $a'_k(\mathbf{r})$  and  $b'_k(\mathbf{r})$ . The coefficients are read from the textures generated in step 1.

### 4.1 Generating the Fourier Opacity Map

No special sorting is required for this step, as all of the primitives contribute to the calculation of the coefficients. Therefore this implementation can be used with arbitrary orderings of primitives, including collections that have already been sorted for some other purpose (e.g. depth-sorted).

We use one or more render targets, depending on the number of coefficients being used in the Fourier series. In our own experiments, we found that 7 coefficients worked well (i.e. the  $a'_0$  DC term + 3 harmonics), hence we used two render targets of 4x16F.

All the render targets are cleared to zero. Then, with additive blending enabled on all channels and with depth testing disabled, we render all the primitives from the point of view of the light source. For each primitive, our pixel shader calculates the following coefficient contributions which are summed into the render targets courtesy of additive blending:

$$\delta a'_{i,k} = -2 \ln(1 - \alpha_i) \cos(2\pi kd_i)$$

$$\delta b'_{i,k} = -2 \ln(1 - \alpha_i) \sin(2\pi kd_i)$$

Note that instead of calculating each  $\sin()$  or  $\cos()$  from scratch, we use the following recurrence relations to compute the higher harmonics:

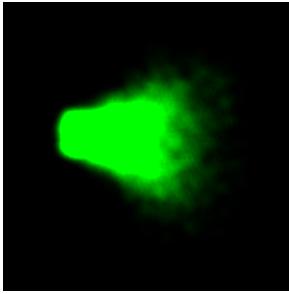
$$\sin((n+1)\theta) = \sin(n\theta) \cos(\theta) + \cos(n\theta) \sin(\theta)$$

$$\cos((n+1)\theta) = \cos(n\theta) \cos(\theta) - \sin(n\theta) \sin(\theta)$$

Figure 2 shows the coefficient map that was generated during the rendering of the image in Figure 1(a).

### 4.2 Rendering the primitives

We then render the primitives in the normal way, with the usual back-to-front sorting for translucent objects and with the usual alpha-blending setup. We compute texture coordinates for reading the coefficients in the same way as we would compute coordinates for shadow mapping [Williams 1978]. We then apply equation (3) to generate the shadowing term, again using the recurrence relations to skip some of the  $\sin()$  and  $\cos()$  calculations.



**Figure 2:** A Fourier Opacity Mapping coefficient map with 7 coefficients packed into two  $4 \times 16F$  textures  $[R, G, B, A] \leftarrow [X, a_0, a_1, b_1]$  and  $[R, G, B, A] \leftarrow [a_2, b_2, a_3, b_3]$ .

## 5 Generalisation to any orthonormal basis

The analysis we performed for projection onto a Fourier basis can be generalised to work for any orthonormal basis  $f_k(z)$ . The basis coefficient contributions can be calculated in the pixel shader in the same way:

$$\delta a_{i,k} = -\ln(1 - \alpha_i) f_k(d_i)$$

And our transmittance approximation takes on the following generalised form:

$$T(d) \approx \exp \left( -\sum a_k g_k(d) \right)$$

Where:

$$g_k(d) = \int_0^d f_k(z) dz$$

Note that in choosing our basis functions, we need to be able to calculate both  $f_k(z)$  and  $g_k(d)$  efficiently in our shaders. In practice, this means we want to choose basis functions where  $f_k(z)$  and  $g_k(d)$  have analytic forms that can be constructed easily from the usual shader math primitives (i.e. multiplication, addition, sin/cos, exponentiation, logarithm, conditionals etc.).

Reintroducing the implied ray  $\mathbf{r}$  for a moment, we observe that the coefficients  $a_k(\mathbf{r})$  form a map, so we will refer to this generalised map as the "coefficient map".

### 5.1 Projection onto a piecewise constant basis

As an example of our generalisation, slice-based opacity maps [Kim and Neumann 2001] can be reformulated as an orthonormal basis technique using the following piecewise-constant basis functions:

$$f_k(z) = \sqrt{n} \left[ H(z - \frac{k}{n}) - H(z - \frac{k+1}{n}) \right]$$

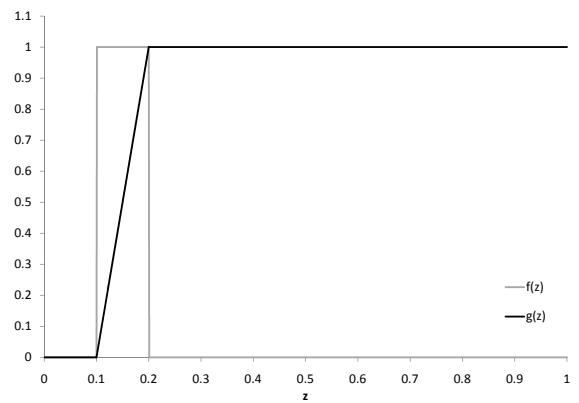
where  $k = 0, 1, \dots, (n-1)$ . The analytic forms for the integrals of  $f_k(z)$  used during reconstruction are given by:

$$g_k(d) = \sqrt{n} \left[ (d - \frac{k}{n}) H(d - \frac{k}{n}) - (d - \frac{k+1}{n}) H(d - \frac{k+1}{n}) \right]$$

In practice, it is convenient to move one of the  $\sqrt{n}$  terms and use the following simplified formulation:

$$f_k(z) = H(z - \frac{k}{n}) - H(z - \frac{k+1}{n})$$

$$g_k(d) = n \left[ (d - \frac{k}{n}) H(d - \frac{k}{n}) - (d - \frac{k+1}{n}) H(d - \frac{k+1}{n}) \right]$$



**Figure 3:** Piecewise constant basis with  $n=10$ ,  $k=1$  –  $f(z)$  is the basis function which we use when rendering the coefficient map,  $g(z)$  is the integral which we use during reconstruction.

## 6 Physical interpretation of pre-filtering

It is useful to understand the physical implications of pre-filtering a coefficient map. Assuming such operations take the form of a convolution, it is sufficient to consider the simpler case of a linear combination between two different rays:

$$a'_k = (1 - \lambda)a_k(\mathbf{r}_0) + \lambda a_k(\mathbf{r}_1)$$

Recalling that:

$$a_k(\mathbf{r}) = \int_0^1 \sigma(\mathbf{r}, z) f_k(z) dz$$

We obtain:

$$a'_k = \int_0^1 [(1 - \lambda)\sigma(\mathbf{r}_0, z) + \lambda\sigma(\mathbf{r}_1, z)] f_k(z) dz$$

i.e. filtering the coefficients is equivalent to filtering the absorption functions:

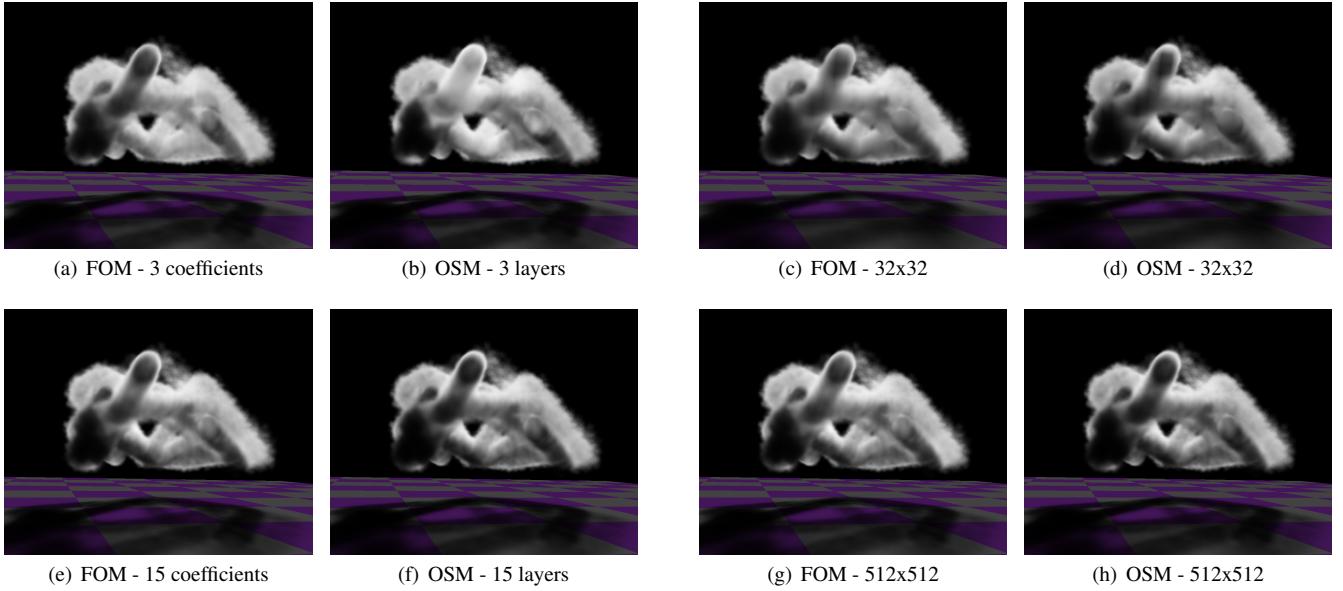
$$\sigma'(z) = (1 - \lambda)\sigma(\mathbf{r}_0, z) + \lambda\sigma(\mathbf{r}_1, z)$$

Recalling that the absorption function represents local density, we may therefore conclude that any convolution filter applied to a coefficient map acts as if the filter were applied to the density of the participating medium. We note that the same physical interpretation attaches to the filtering of Opacity Shadow Maps [Kim and Neumann 2001].

## 7 Discussion of Fourier Opacity Mapping

### 7.1 Stability under translation

Slice-based approaches discretize depth values by partitioning rasterized primitives into depth buckets. For dynamic scenes in which the translucent primitives are moving and the depth buckets may also move, such discrete mappings cause slice-shaped aliasing artefacts (see Figure 5). In contrast, Fourier Opacity Mapping uses a continuous depth mapping and trades discretization artefacts for loss of high-frequency detail and some amount of ringing. When the absorption function is relatively smooth, Fourier Opacity Mapping is able to ensure that the gross shape and z-position of major features in the absorption function are consistently reconstructed (see Figure 1(c)), even under translations (see Figure 6) and (to an extent) scalings in z.



**Figure 4:** Effect of varying the number of coefficients and the resolution of the opacity map, for Fourier Opacity Mapping (FOM) and Opacity Shadow Maps (OSM). Figures a, b, e and f are using a 512x512 opacity map. Figures c, d, g and h are using 15 coefficients per texel.

## 7.2 Outliers

Techniques that rely on an initial pass to narrow the z-range, such as Deep Opacity Maps [Yuksel and Keyser 2008], can have difficulties dealing with outliers correctly (in this context, an outlier is a primitive that is apart from the main group). Such outliers can affect the z-range disproportionately to their actual effect on the transmittance function.

Fourier Opacity Mapping copes well with outliers because it weights their contribution to the transmittance function by primitive density and opacity alone (see Figure 7).

## 7.3 Ringing

Ringing can become more problematic when Fourier Opacity Mapping is applied to thin or high-frequency features with high opacity. We find empirically that for a given number of coefficients, the presence or absence of ringing is determined by a critical feature size. If ringing is present, then the magnitude of the ringing is determined by the overall opacity of the feature (see Figure 8).

For these reasons, we characterise Fourier Opacity Mapping as being most applicable to interactive applications with low opacity and large feature sizes such as clouds of smoke, steam or fog, and low-opacity hair (see Figure 9). In non-interactive applications with a requirement for smaller feature sizes, the onset of ringing can be reduced by using more coefficients.

As we know that the actual transmittance should never exceed unity, we can additionally clamp the reconstructed transmittance to the range [0, 1] in the pixel-/fragment-shader in order to remove some of the over-brightening artefacts that might otherwise occur as a result of ringing (we note that the same solution was used to control ringing in Convolution Shadow Maps [Annen et al. 2007; Annen et al. 2008]). This in turn allows us to tolerate a greater amount of ringing in some situations, thus saving us the computational expense of additional coefficients.

## 7.4 Opaque shadows

We have already seen that the application of Fourier Opacity Mapping to thin high-opacity features can cause ringing, and this would seem to preclude Fourier Opacity Mapping from handling opaque shadows directly. However, opaque shadows can easily be handled separately by traditional opaque shadow-mapping techniques [Williams 1978], with the resulting shadowing terms combined by modulation:

$$T(\mathbf{r}, d) = T_{\text{opaque}}(\mathbf{r}, d)T_{\text{FOM}}(\mathbf{r}, d)$$

## 8 Performance

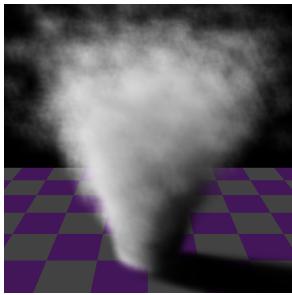
We have tested performance by rendering the 16K-particle plume shown in Figure 1(a) at two different screen resolutions and three different coefficient-map resolutions on a GeForce GTX 260. We have compared Fourier Opacity Mapping with a uniform slice-based approach [Nguyen and Donnelly 2005] (see Figure 4).

Coeff map resolution	1024x1024	512x512	256x256
FOM Coeff rendering (ms)	15.5	4.3	1.7
FOM Scene rendering (ms)	14.0	12.3	11.7
OSM Coeff rendering (ms)	10.1	3.0	1.5
OSM Scene rendering (ms)	13.3	12.2	11.7

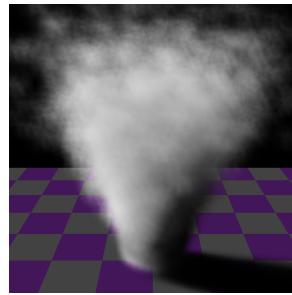
**Table 1:** 1680x1050 Fourier Opacity Mapping (FOM) with 7 coefficients and Opacity Shadow Maps (OSM) with 7 coefficients.

Coeff map resolution	1024x1024	512x512	256x256
FOM Coeff rendering (ms)	15.9	4.2	1.6
FOM Scene rendering (ms)	10.1	7.4	7.3
OSM Coeff rendering (ms)	11.1	2.8	1.5
OSM Scene rendering (ms)	9.9	7.3	7.3

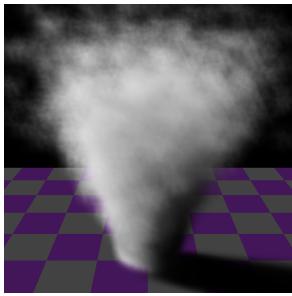
**Table 2:** 1280x800 Fourier Opacity Mapping (FOM) with 7 coefficients and Opacity Shadow Maps (OSM) with 7 coefficients.



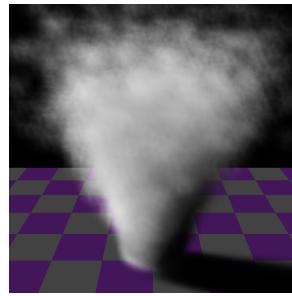
(a) FOM - baseline



(b) OSM - baseline



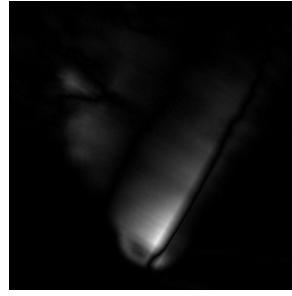
(c) FOM - with depth bounds inflated by 10%



(d) OSM - with depth bounds inflated by 10%



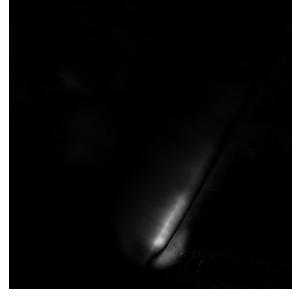
(e) FOM - difference [5(a)- 5(c)]



(f) OSM - difference [5(b)- 5(d)]

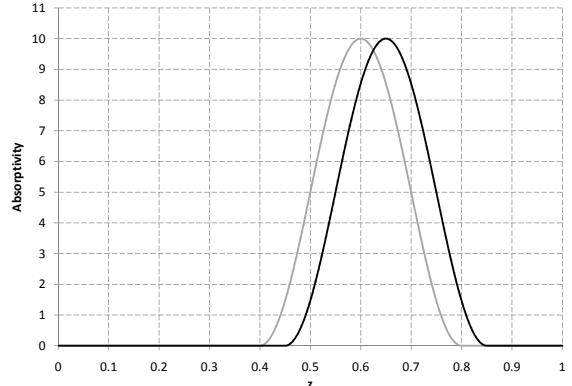


(g) FOM - relative difference [5(e)/5(a)]

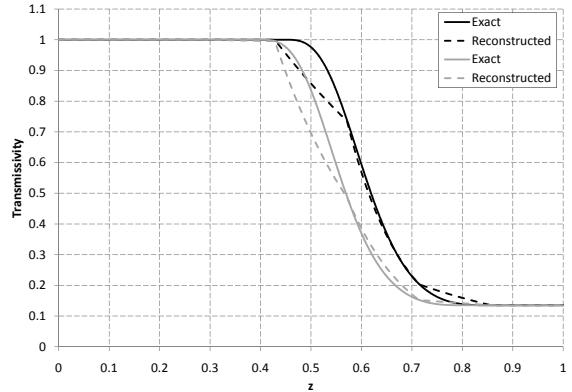


(h) OSM - relative difference [5(f)/5(b)]

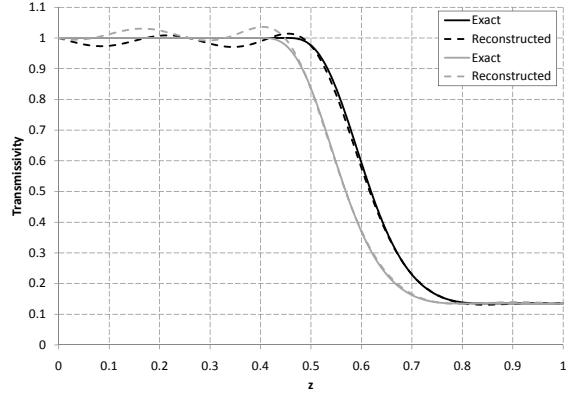
**Figure 5:** Effect of depth bounds discontinuity on smoke plume shadow rendering (7 coefficients in all cases), for Fourier Opacity Mapping (FOM) and Opacity Shadow Maps (OSM).



(a) A representative absorption function that might be encountered during the simulation of a plume of smoke or steam by a particle system, together with a representative translation of the same absorption function.

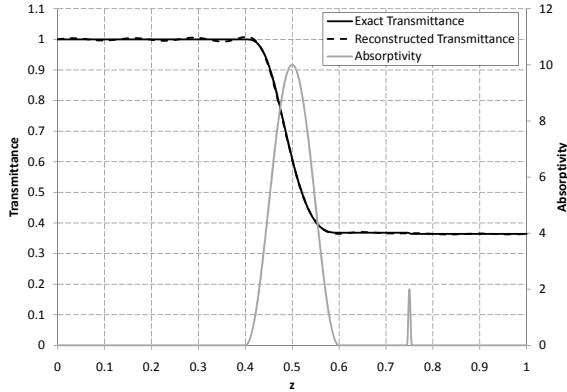


(b) Comparison of exact vs reconstructed transmittance obtained from absorption functions in 6(a), where reconstruction was performed by projection onto a piecewise-constant basis.



(c) Comparison of exact vs reconstructed transmittance obtained from absorption functions in 6(a), where reconstruction was performed by projection onto a Fourier series.

**Figure 6:** Comparison of stability under translation. Note how the Fourier-basis reconstruction accurately preserves the location and shape of the major (and perceptually important) light-to-dark transition, but also note how this is at the expense of some ringing elsewhere in the reconstruction. In this case (and in many such cases) we find that this tradeoff is beneficial because the ringing is introduced in a part of the  $z$ -range that is either unoccupied or perceptually unimportant.



**Figure 7:** A representative absorption function with an outlying feature, together with the exact transmittance function and the reconstruction using Fourier Opacity Mapping - the outlier has negligible effect on the reconstruction.

Timings are available in Tables 1 and 2. Although the performance of Fourier Opacity Mapping is comparable to Opacity Shadow Maps during the main scene rendering step, Fourier Opacity Mapping is at a disadvantage by as much as 1.5x during the coefficient rendering pass.

However, we also note that the performance of this pass is approximately proportional to the resolution chosen for the coefficient map, so in practice we can achieve acceptable performance by careful resolution management. We are helped in this by the smooth low-frequency nature of the subject matter, and also by the smoothing action of any pre-filtering step.

## 9 Tinting and color

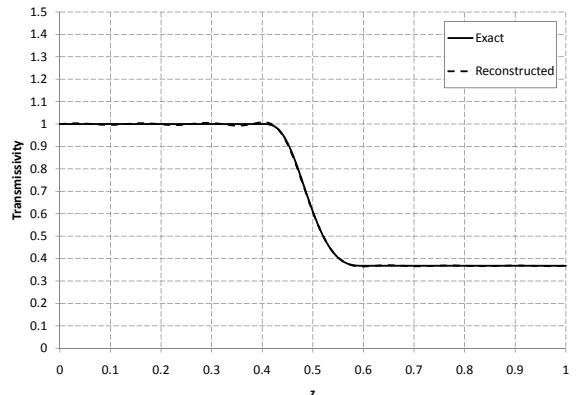
We have assumed so far that the opacity information associated with a primitive is single-valued, however Fourier Opacity Mapping extends trivially to the case where spectral opacity information is provided in the form of an RGB triple by triplication of coefficients.

## 10 Conclusion

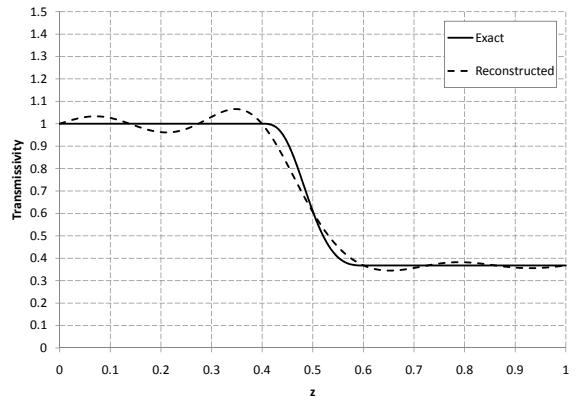
We have presented Fourier Opacity Mapping, and we have shown how it can be used to render artefact-free pre-filtered volumetric shadows using a compact two-step algorithm that offers comparable performance to existing interactive techniques. We have shown how our algorithm is derived from firm physical principles, and we have also shown how the algorithm can be generalised to other orthonormal bases, such as piecewise-constant. The algorithm has shown to be robust enough to be used in the shipping game Batman: Arkham Asylum, for shadowing smoke particle systems (see Figure 10). By way of future work, we would like to investigate other methods based on the inner-product, including other orthonormal bases, and also whether it is possible to take account of scattering effects within our framework.

## Acknowledgements

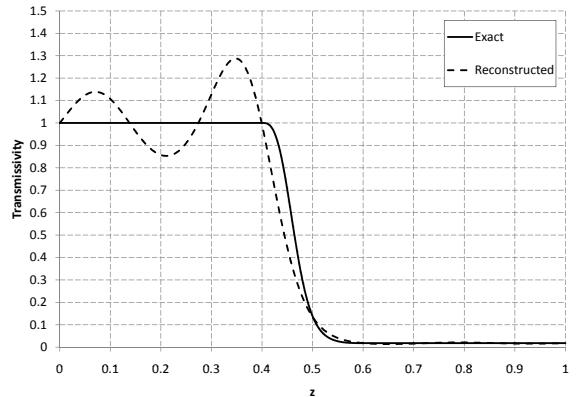
We would like to thank Jonathan Cohen, Peter Shirley and Simon Green for their enthusiasm and helpful suggestions, and Miguel Sainz for supporting us unstintingly in seeing this work to fruition.



(a) A thin feature reconstructed using Fourier Opacity Mapping with 15 coefficients.

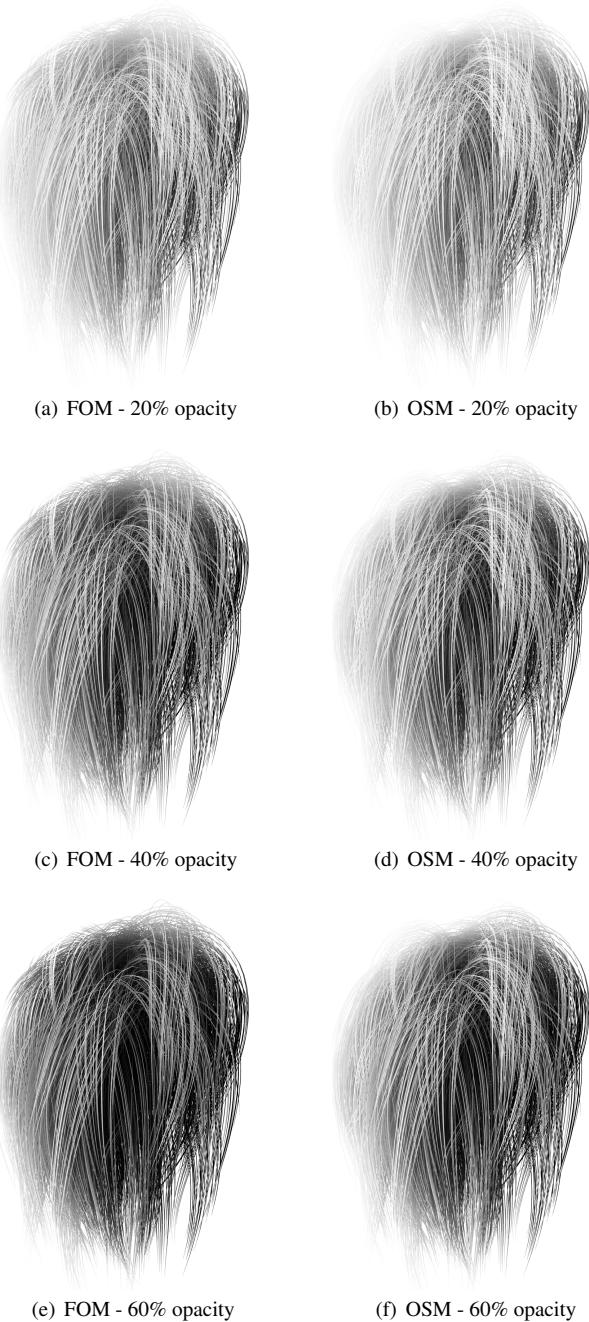


(b) The same thin feature reconstructed using Fourier Opacity Mapping with 7 coefficients - note that a small amount of ringing has become apparent.



(c) A 4x more opaque thin feature reconstructed using Fourier Opacity Mapping with 7 coefficients - note how the magnitude of the ringing has increased.

**Figure 8:** How ringing in Fourier Opacity Mapping is affected by number of coefficients and feature opacity.



**Figure 9:** Comparison of Fourier Opacity Mapping (FOM) and Opacity Shadow Maps (OSM) on hair. For both FOM and OSM, the opacity map had a 512x512 resolution and was pre-filtered with a 33x33 box filter to reduce aliasing and soften the shadows. The OSM images are using 32 slices and the FOM images are using 5 coefficients per texel. This hair model is a built-in example hair system from Maya®Hair 8.5.)



**Figure 10:** Screenshot from the game “Batman: Arkham Asylum”, courtesy of Rocksteady Studios and Square Enix.

## References

- ANNEN, T., MERTENS, T., BEKAERT, P., SEIDEL, H.-P., AND KAUTZ, J. 2007. Convolution shadow maps. In *Rendering Techniques 2007: Eurographics Symposium on Rendering*, Eurographics, J. Kautz and S. Pattanaik, Eds., vol. 18, 51–60.
- ANNEN, T., DONG, Z., MERTENS, T., BEKAERT, P., SEIDEL, H.-P., AND KAUTZ, J. 2008. Real-time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph.* 27, 3, 1–8.
- GREEN, S., 2009. DirectX11 effects. GDC 2009.
- HADWIGER, M., KRATZ, A., SIGG, C., AND BÜHLER, K. 2006. Gpu-accelerated deep shadow maps for direct volume rendering. In *Proceedings of the 21st ACM SIGGRAPH/Eurographics symposium on Graphics hardware*.
- HALÉN, H. 2007. *Real-Time Hair Simulation and Visualisation for Games*. Master’s thesis, Lund University.
- KIM, T.-Y., AND NEUMANN, U. 2001. Opacity shadow maps. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, June 25-27, 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* 2003, 150–162.
- LOKOVIC, T., AND VEACH, E. 2000. Deep shadow maps. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 385–392.
- NGUYEN, H., AND DONNELLY, W. 2005. Real-time rendering and animation of realistic hair in ‘nalu’. In *GPU Gems 2*. 361–380.
- SINTORN, E., AND ASSARSSON, U. 2008. Real-time approximate sorting for self shadowing and transparency in hair rendering. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*, February 15-17, 2008, Redwood City, California.
- SINTORN, E., AND ASSARSSON, U. 2009. Hair self shadowing and transparency depth ordering using occupancy maps. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, February 27-March 01, 2009, Boston, Massachusetts.
- WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In *Proceedings of SIGGRAPH 1978*, vol. 12, 270274.
- YUKSEL, C., AND KEYSER, J. 2008. Deep opacity maps. In *Computer Graphics Forum* 27(2): 675-680, (Proceedings of EUROGRAPHICS 2008).