A Programmable System for Artistic Volumetric Lighting

Derek Nowrouzezahrai[1] Jared Johnson[2] Andrew Selle[2] Dylan Lacewell[2] Michael Kaschalk[2] Wojciech Jarosz[1]

[1]Disney Research Zürich [2]Walt Disney Animation Studios

Figure 1: *Our system was used to author artistic volumetric effects for the movie Tangled. Our technique's ability to produce curving light beams is used to match the organic artistic style of the film.*

Abstract

We present a method for generating art-directable volumetric effects, ranging from physically-accurate to non-physical results. Our system mimics the way experienced artists think about volumetric effects by using an intuitive lighting primitive, and decoupling the *modeling* and *shading* of this primitive. To accomplish this, we generalize the physically-based photon beams method to allow arbitrarily programmable simulation and shading phases. This provides an intuitive design space for artists to rapidly explore a wide range of physically-based as well as plausible, but exaggerated, volumetric effects. We integrate our approach into a real-world production pipeline and couple our volumetric effects to surface shading.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and realism—Color, shading, shadowing, and texture

Keywords: Lighting design, artist control, participating media

Links: DL PDF

1 Introduction

Light scattering in participating media is responsible for many natural phenomena. Simulating the evolution of volumetric media over time, as well as the complex light transport within it, are difficult problems in animation (e.g. [Fedkiw et al. 2001; Hong et al. 2007]) and rendering [Jensen and Christensen 1998; Jarosz et al. 2011]. Recent advances have made it feasible to incorporate a wider range of such effects in feature animation production; however, most of this work has focused on accelerating computation and increasing accuracy. Given physically accurate techniques, manipulating physical parameters to attain a target look is a challenging process. Previous work addresses this problem by investigating art-directable control of light transport for surface reflectance (e.g. [Kerr et al. 2010]). However, artistic authoring and manipulation of *volumetric* lighting remains an unsolved problem.

Guiding accurate fluid animation using arbitrary source terms, while maintaining a principled framework, has been previously explored [McNamara et al. 2004; Treuille et al. 2003]. Similarly, our framework allows for programmatic and art-directed injection of source terms into physically-based volumetric light transport.

While physically accurate and art-directable rendering have seemingly conflicting goals, recent efforts to incorporate physically-based rendering into production have shown great potential [Tabellion and Lamorlette 2004; Křivánek et al. 2010]. Such techniques are seeing increased adoption because they provide complex and subtle lighting which would otherwise take extensive manual manipulation to replicate with ad-hoc techniques. Unfortunately, physically accurate rendering is often not sufficiently expressive for the caricatured nature of animated films: though physically-based rendering may provide a great starting point, the challenge then becomes introducing controls necessary to obtain a desired artistic vision. We carefully combine these two areas and choose to generalize an existing physically-based approach for rendering volumetric lighting to art-directable shading and simulation (Section 4).

We present a system for generating target stylizations of volume effects, mimicking the way professional artists hand draw these effects. We base our approach on *photon beams* [Jarosz et al. 2011], which provide physically-based rendering of participating media. We make the following contributions while generalizing photon beams to allow for artistic control of volumetric effects:

- We observe that manipulating physical parameters of participating media results in unintuitive changes to the final image. To address this we derive physically-based scattering properties to match a user-specified target appearance, providing an intuitive space for appearance modeling of participating media.
- To allow for *non-physical* effects, we generalize both the photon generation and radiance estimation stages of the photon beams method. We replace each stage with a procedural, programmable component. While each component could implement the physically-based approach, this provides enough programmatic flexibility for artist-driven volumetric effects.

(c) ACM, 2011. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM Transactions on Graphics, 30, 4, August 2011. <doi.acm.org/10.1145/1964921.1964924>

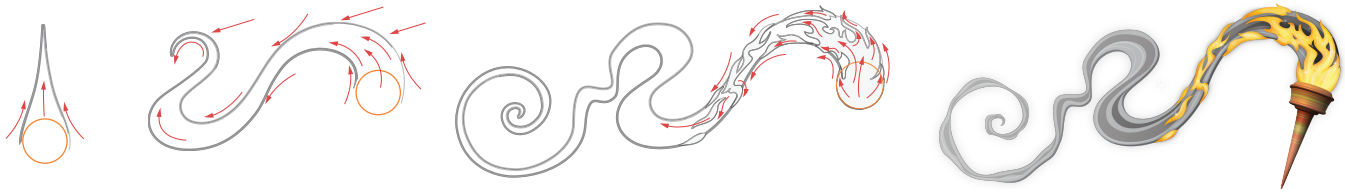

Figure 2: *Left to right: to hand draw a volumetric effect, artists first sketch a coarse high-level outline of the media, then progressively refine this sketch to include dynamics and increased levels of detail, before finally "rendering" the final image.*

2 Previous Work

Lighting and Material Control. Several works provide intuitive user controls for editing the lighting or materials in a scene to obtain a desired *surface shading* effect. Among these approaches, either the geometry and intensity of light sources [Pellacini et al. 2007], or the surface material properties [Obert et al. 2008; Song et al. 2009], are manipulated in order to control shadows and reflectance from local lights [Kerr et al. 2010] and environment lights [Obert et al. 2010; Pellacini 2010]. Kerr and Pellacini [2009; 2010] recently surveyed and compared several of these techniques. We generalize photon beams to both volumetric material and lighting control.

Song et al. [2009] consider editing sub-surface scattering on surfaces (using the dipole approximation). We target the control of more general *volumetric* effects, and thus use a different mathematical model (photon beams). Sadeghi et al. [2010] present an art-directable shading model for hair rendering in animated feature films. Kerr et al. [2010] present *BendyLights*, a non-linear spotlight for controlling lighting and shadows. We also incorporate non-linear lighting, and in fact each photon beam can be considered a volumetric generalization of a BendyLight. However, since we have thousands of these lighting primitives, we provide a more suitable control mechanism (see Sections 5 and 7.1).

Artistic, Non-Photorealistic Rendering. Schmid et al. [2010] expose a programmable model for controlling motion effects. Where Schmid et al. deal with imitating hand-animated motion effects, we instead focus on replicating hand-drawing workflows for volumetric effects using virtual tools. Like our approach, their model is based on traditional hand drawn effects (see Section 3), as well as supporting both physically-based and art-directable effects.

Non-photorealistic rendering (NPR) targets stylized rendering of images which we also address with our approach. Selle et al. [2004] generate cartoon renderings of smoke. Though their goals are similar to ours, we focus on a particular type of stylization: plausible but exaggerated realism, as opposed to cartoon-like results.

Steering Physical Simulation. There has been a push to use increasingly accurate rendering approaches in production; however, the challenge is providing procedural control to obtain a desired look. Manipulation of physically-based fluid animations has demonstrated the utility of this hybridization (see e.g. Angelidis et al. [2006]). At a high-level, these methods manipulate the simulation by injecting controlled source terms to the Navier-Stokes equation, instead of setting the initial fluid conditions. Analogously in light transport, light sources represent the initial conditions, but we add source terms locally to steer the transport of light. Using non-physical BRDFs can also be seen as an injection of source terms. Our method applies these concepts to participating media by using beams as an intuitive and controllable volumetric lighting primitive.

Programming Models. We build on top of existing content generation tools, which offer expressive customizability to author geometry and shading effects. Domain-specific programming languages (DSL) are an essential component for providing this flexibility. The classic example of this is the RenderMan™ Shading Language (RSL) [Cook et al. 1987]. Similarly, Tessendorf's *FELT* scripting language [2010] enables the controllable specification of fluid simulation behavior. Both of these systems have a long history of successful application in film production. Other tools such as Maya™ and Houdini™ also expose programmability through node-based visual DSLs, which provide artist-accessible models of programming. Our generalization of photons beams provides the flexibility of a DSL for the specific domain of volumetric lighting.

3 Volumetric Lighting Design Requirements

We first characterize the requirements of an intuitive and expressive volumetric lighting system. We gathered these requirements directly from feature-film lighting artists, and distilled them into the following core principles (in decreasing order of importance):

- the model should adhere as closely as possible to abstractions that artists normally use when thinking about volume effects,
- it should generate results spanning the entire gamut of believability, from physically accurate to completely art-directed,
- the system must integrate as seamlessly as possible into the existing production pipeline, and
- the system should expose flexibility through programmability.

The first requirement is by far the most important: the manner in which artists conceptualize volumetric effects strongly influences the way they reason about digitally replicating these effects. Since no existing tool directly addresses the problem of designing volumetric effects, artists have had to warp their intuitive models to fit the capabilities of existing tools. We found that the biggest problem with volumes for artists is that, unlike surfaces, they are not intuitive. In particular, the lack of a geometric model makes them more difficult to grasp. We provide this missing link for volumetric lighting, using photon beams. This hybrid light/geometry representation allows artists to think about light in a volume as its own geometric entity, to be manipulated and sculpted. Lighting volumetric effects then reduces to familiar modeling and shading problems.

The beam representation also relates to the way artists draw volumetric effects. Artists often employ a two-stage procedure (see Figure 2 for a sketch inspired by Gilland [2009]): first, shapes which coarsely define the media's volume are sketched and refined; then, given these shapes, shading is applied to obtain the final result. We similarly separate the representation of the media from its shading: beam primitives define the "shape" and material properties of the media, to which a shading model is applied. Our solution generalizes the photon beams algorithm, where both the process of generat-

ing (Section 7) and shading (Section 6) beams can be performed using physically-accurate or non-physical, art-directable procedures.

4 Theoretical Foundation

We define light transport in participating media, and briefly describe the photon beams algorithm. In the subsequent sections we generalize this approach to facilitate artistic control.

4.1 General Light Transport

We express light at a point x (e.g. the camera) from direction ω as:

$$L(x,\omega) = L_s(x,\omega) + L_m(x,\omega). \quad (1)$$

Surface radiance (potentially attenuated by the media), L_s, is governed by the rendering equation [Kajiya 1986]. The second term is the radiance due to participating media [Chandrasekar 1960],

$$L_m(x,\omega) = \sigma_s \int_0^d e^{-\sigma_t z} \int_{\Omega_{4\pi}} \rho(\theta_z) L(x_z,\omega_z)\, d\omega_z\, dz, \quad (2)$$

which accumulates light at points x_z along the eye ray (until the ray hits a surface d units away). This light recursively depends on radiance arriving at x_z from directions ω_z over the sphere $\Omega_{4\pi}$. The phase function is ρ, where $\cos\theta_z = \omega \cdot \omega_z$. For simplicity, we describe the case for homogeneous media with absorption, scattering and extinction coefficients, σ_s, σ_a and $\sigma_t = \sigma_s + \sigma_a$. For brevity, we refer to the set of (potentially spatially-varying) scattering coefficients as $\sigma = \{\sigma_s, \sigma_a, \sigma_t\}$.

Previous work has focussed primarily on controlling or editing the L_s term in Equation 1. We focus on controlling Equation 2.

4.2 Photon Beams for Physically Accurate Rendering

Photon mapping methods [Jensen and Christensen 1998] compute Equation 1 in two-steps. During precomputation, a collection of photons, each with power Φ_p and direction ω_p, are traced through the scene and stored at points x_p corresponding to intersections with surfaces and within volumetric media. These photons can be interpreted as a point-sampled representation of the light distribution, and locally approximate the radiance $L(x,\omega)$. Second, during rendering, a *shading pass* queries the data and applies a physically accurate shading model to compute final radiance towards the eye.

In the photon beams method the first pass remains, however each photon is treated as a beam of light starting at x_p and going in direction ω_p, and density estimation uses these beams, instead of photon points. Jarosz et al. derived a "Beam × Beam 1D" estimate for computing L_m along camera rays given a collection of beams:

$$L_m(x,\omega) = \sigma_s \sum_p k_r(u)\, e^{-\sigma_t z} \rho(\theta_p)\, e^{-\sigma_t v} \frac{\Phi_p}{\sin\theta_p}. \quad (3)$$

The summation loops over all beams and evaluates the terms at the intersection of the camera ray with each photon beam (see Figure 3). The $e^{-\sigma_t z}$ term computes transmittance towards the camera, where z is the distance between the intersection and the camera position x, and $e^{-\sigma_t v}$ computes transmittance along the beam where v is the distance to x_p. The $\sin\theta_p$ term takes foreshortening into account for a flat beam as it rotates relative to ω. Each beam has a finite width determined by a kernel k_r, which weights photons according to the 1D distance u between the camera ray and the beam. Surface shading is naturally handled using photon mapping: the endpoints of the photon beams are surface photons and are used for density estimation of L_s. Figure 3 illustrates the geometric setup for the physically accurate shading model in Equation 3.

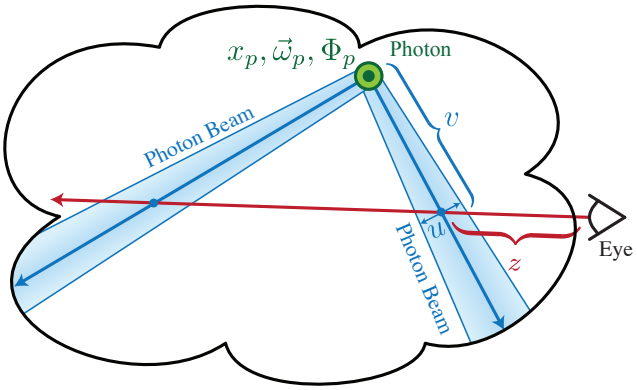

Figure 3: *Geometry for the Beam × Beam 1D of Equation 3.*

Discussion. Photon beams have an intuitive physical interpretation: a beam can be thought of as a small spotlight, attenuated volumetrically along its central axis. This geometric interpretation is tangible and easy to grasp. Furthermore, the approach provides a natural interpretation of geometry and shading in volumes. Together, these properties satisfy the first requirement presented in Section 3, making photon beams an ideal approach for us to extend. Unfortunately, photon beams do not satisfy all requirements set forth in Section 3: unmodified, the approach is limited to a very indirect form of editing not suitable for art-directability.

When physically accurate results are desired, it is difficult to construct a targeted shading result by directly manipulating σ in Equation 3. For example, a common volumetric design goal is to induce a desired color gradient across the media's volume; to do so with physically-based shading, the artist must manipulate the σ parameters. Unfortunately, these parameters only have an indirect, and often unintuitive, influence on the observed color of the medium. Instead of modifying the σ parameters, we provide a system that allows an artist to directly specify a desired color gradient. In Section 5 we show how to automatically deduce the physically-based scattering properties of the medium from this input. This provides a more intuitive parameter space for controlling the appearance of participating media, while maintaining physical correctness.

A more pressing limitation is that, often, a desired shading result *requires* non-physical shading. Equation 3 cannot satisfy this requirement, by definition. In Section 6, we abstract Equation 3 into a few high-level effects, resulting in a generalized shading model suitable for both physical and non-physical shading.

In addition to shading, the generation and resulting distribution of beams influences the media lighting. While the physically-based approach is suitable in some scenarios, it is often difficult to create a desired effect by directly manipulating the physically-accurate light transport process. For this, we draw upon ideas from steerable fluid simulation. In Section 7, we similarly introduce steerable source terms by allowing artists to procedurally sculpt and generate (potentially time-dependent) distribution of photon beams.

5 Deducing Scattering Parameters

For surface reflectance, specifying material properties is fairly intuitive and predictable. The user can, for instance, graphically choose the surface color or specular sharpness. Such surface parameters have fairly predictable behavior and artists can therefore rapidly produce a desired appearance. Unfortunately, specifying physically-based media parameters is a more difficult problem. The σ parameters in Equation 3 do not influence the final shading in

such a direct, intuitive manner. For example, when directly manipulating σ_s and σ_t, the media close to a white light source will have a color equal to σ_s, however, this color will diffuse to the complement of σ_t with distance. Another complication is that the media parameters combined with the light source intensity are an overcomplete parameter set. This means that an infinite number of parameter choices can lead to identical results. To overcome this, we present a tool that allows an artist to specify the observed color at two points along a beam in the medium, and returns the physical parameters for Equation 3 which induce this desired color gradient.

5.1 Determining σ Given Beam Power

We allow the user to specify two target colors, C_1 and C_2, within the medium (one at the start of a beam and one at a canonical distance (assumed to be 1)[1] along the beam). Considering a beam perpendicular to the view at some canonical distance z, Equation 3 dictates the following behavior along the length of the beam:

$$L_m(v) = \sigma_s \, \Phi_p \, e^{-\sigma_t \, z} \, e^{-\sigma_t \, v}. \tag{4}$$

We omit the phase function ρ, since it is constant in this configuration. If the color and power Φ_p of the beam is fixed (corresponding to a fixed light source color in single-scattering), we wish to obtain the parameters σ_s, σ_t under the constraints imposed by C_1 and C_2:

$$C_1 = \sigma_s \, \Phi_p \, e^{-\sigma_t \, z}, \qquad C_2 = \sigma_s \, \Phi_p \, e^{-\sigma_t \, z} \, e^{-\sigma_t}. \tag{5}$$

We have two equations and two unknowns and we solve for the parameters directly. Dividing the equations provides an estimate for σ_t and then plugging back into either equation yields:

$$\sigma_t = -\log\left(C_2/C_1\right), \qquad \sigma_s = \left(C_1/\Phi_p\right)\left(C_1/C_2\right)^z. \tag{6}$$

Note that this solution is physically meaningful only if $\sigma_s \leq \sigma_t$ which can be satisfied if $C_2 < C_1$ in all color channels.

5.2 Solving for All Photon Beam Parameters

We also provide the ability to specify all the parameters for photon beams using this gradient specification. In this case, we deduce not only σ_s and σ_a, but also the photon powers, using one interface. Unfortunately, the space is now overcomplete, so to solve for σ_t as before we need a way to disambiguate between the infinitely many solutions for Φ_p and σ_s. Given the albedo, $\alpha = \sigma_s/\sigma_t$ (which the user typically does not modify from its default value), we can solve $\sigma_s = \alpha \, \sigma_t$ and plug back into Equation 6 to obtain:

$$\sigma_s = -\alpha \log\left(C_2/C_1\right), \qquad \Phi_p = \left(C_1/\sigma_s\right)\left(C_1/C_2\right)^z. \tag{7}$$

Our video shows a user setting the color gradient of a beam while our system automatically deduces the physical media parameters.

In addition to allowing more intuitive manipulation of scattering parameters, the derivations above also provide some interesting insights about the behavior of light in participating media. Considering the color gradient only along a single beam is similar to restricting our derivations to single-scattering (single-scattering is a beam emanating directly from a light with the beam's power set to that of the light). This implies that we could apply a similar procedure to obtain media parameters for an arbitrary single-scattering technique. Furthermore, we see from our derivations that either albedo or the light's power are irrelevant to the final image color: by changing one, we can modify the other to obtain identical results.

[1] Arbitrary distances for C_2 simply result in uniform scaling of σ.

6 Procedural Shading of Photon Beams

While the above mapping provides an intuitive parameter space for specifying physically-based participating media, our system must also support non-physical shading, as discussed in Section 3. We will distill Equation 3 into a few simple concepts that we then use to devise a non-physical generalization.

Volumetric Radiance Estimation in a Nutshell. Given a collection of photon beams, we notice that the radiance on the image is influenced by only a few high-level parameters (see Figure 3): the angle between the eye ray and the beam (θ_p), the distances along and across the beam (v and u), and the distance to the camera (z).

These parameters influence the observed radiance through four physical processes, each associated with a function below:

- color change due to attenuation along the beam, $f_b(v)$,
- color change due to attenuation towards the eye, $f_e(z)$,
- shading depends on the viewing angle, $f_f(\theta_p)$, and
- shading is influenced by the photon beam's thickness, $f_t(u)$.

These high-level parameters, four physical processes, and media parameters σ, fully describe physically accurate shading as well as arbitrary non-physical shading behavior. For non-physical controllability, we allow artists to specify their own instantiations of the four functions with a programmable shading language.

Our art-directable radiance estimate replaces Equation 3 with:

$$L_m(x, u, v, z, \theta_p) = \sum_p f_t(u) \, f_b(v) \, f_e(z) \, f_f(\theta_p) \, . \tag{8}$$

While this generalized model can be used for art-directable shading, it can also replicate physically accurate shading with:

$$f_t = \Phi_p k_r(u), \; f_f = \sigma_s \frac{\rho(\theta_p)}{\sin\theta_p}, \; f_e = e^{-\sigma_t z}, \text{ and } f_b = e^{-\sigma_t v}.$$

This reproducibility satisfies the second requirement in Section 3: generating physically-accurate to completely art-directable results[2].

6.1 Implementation and Flexibility

We implemented our system using shade trees and a template Renderman™ shader, allowing artists to leverage familiar techniques, such as texture mapping and color spline manipulation. These can be used to, for example, specify the beam intensity falloff rate (f_t) or the view-dependent scattering profile (f_f). Replacing the physically-accurate processes with procedural expressions can yield interesting shading effects. Figure 4 combines physically-based parameter setting and non-physical shading in a scene with physically-accurate beam data from Walter et al. [2009].

Entries in Figure 4 marked with — denote functions which match the physically accurate model. We note a purposeful abuse of mathematical and programmatic notation: `noise` and `tex` are noise and texture look-up functions, and `beamID` is a unique integer assigned by our system to each beam. When `tex` is used, we include the associated texture map (all 1D textures in these examples).

Note that with simple modifications, a large breadth of artistic manipulations can be explored, even with photon beams generated using physically-based emission.

[2] Note that this requirement is only partially satisfied here, for shading. In Section 7, we satisfy the remainder of this requirement using physically accurate beam generation and propagation.

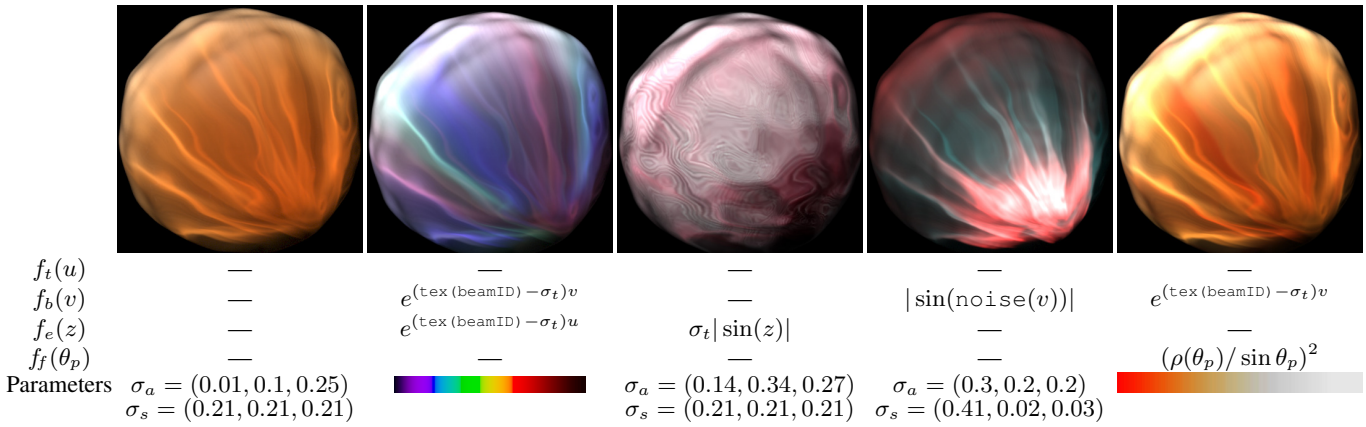

Figure 4: *We replicate ground truth results of the Bumpy Sphere scene from Walter et al. and explore non-physical shading of the same scene using simple procedural modifications of the four physical processes. Entries marked with — use the physically-based definition of the associated physical process, and example shader code for reproducing the left most image is included in supplemental material.*

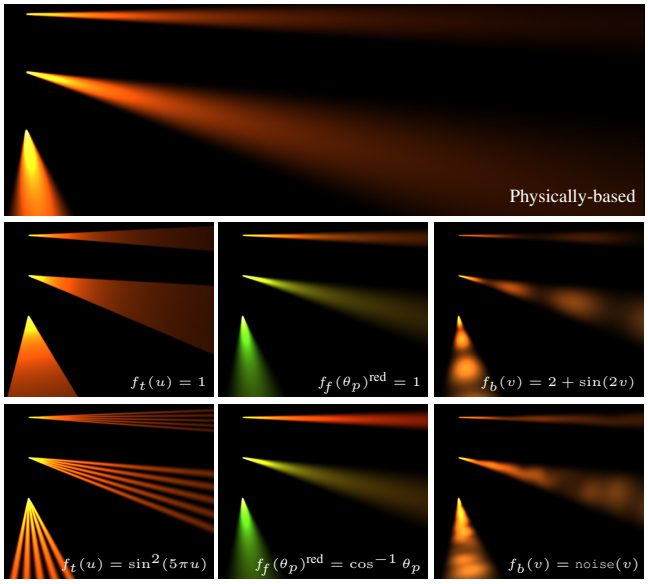

Figure 5: *Beams lit with the physically-based definitions of the four physical processes (top), and the effect on the final shade due to simple, isolated procedural modifications to three of the four physical processes (middle and bottom rows).*

Figure 5 isolates simple changes to the individual physical processes, in a simple scene, to illustrate the impact that each process can have on the final shade.

7 Procedural Photon Beam Generation

So far, we discussed how artists can shade beams in both physical and non-physical ways, given beams generated using physical simulation [Jarosz et al. 2011]. Lighting artists are familiar with tools used for generating such physically-based solutions and we have implemented a shadow mapping [Williams 1978] based approach to quickly generate single-scattered beams in a scene. While these methods of beam generation are useful for physically-accurate results, in some scenes a stylized volumetric effect is necessary. In these cases art-directable beam generation becomes important.

As discussed in Section 4, the beams in a scene are a geometric lighting primitive that explicitly define the distribution of scattered light in a volume, or more simply, the volume's *lit shape*. This intuitive interpretation provides a key insight when designing a tool for art-directable beam generation. Traditional volume rendering includes the problem of defining a field of volumetric shading parameters, but now an artist can instead think of sculpting a volume as if it were a geometric entity; this is a natural 3D extension of the process of hand sketching 2D volumetric effect (see Figure 2).

These observations justify and motivate our combination of traditional CG geometric design tools with art-directable volumetric beams. Procedural geometric modeling is used to deposit and evolve beams over time. One advantage of this approach is that existing tools handle explicit geometry much better than volume modeling. Another advantage is that it allows artists to focus on directly creating volumetric appearance instead of the potentially expensive process of indirectly inducing a desired appearance through physical simulation and volumetric shaders.

We illustrate this approach with light "pouring out of" a door crack and keyhole in Figure 8. This is a common lighting scenario in movie production, but unfortunately a notoriously difficult sampling problem for physical simulation. Instead of manual placing lights (in the occluded room) and relying on expensive physical simulation to generate beams, we allow the artist to sculpt this distribution directly. The artist models the beams of light coming through the door with a procedural particle system, distributing points along the crack with the necessary beam attributes (direction, length, power and width). Jittering the beam start points (procedurally) adds to the dramatic effect and the artist can easily add procedural, time-dependent perturbations to animate the beams.

7.1 Curved Beams

Procedurally generated beams can form curved, warped frustra, as opposed to the perfectly conical frustra generated with physically accurate light transport. This is especially common when generating beams from procedural or fluid-based simulation. These curved beams can be related to the recent work on *BendyLights* [Kerr et al. 2010]. As discussed in Section 4.2, a single photon beam can be interpreted as a spatially-varying volume of light. Thus, each curved photon beam induces *volumetric* a bendy light, albeit with a more flexible, art-directable spatial and angular radiance distribution.

Figure 6 was generated using the 2D illustrations in Figure 2 as motivation. In this case, an artist created two turbulent particle simulations and associated a time-dependent path to each particle in the simulation. Particles were chained together to form the beam data used to render the smoke and fire. This entire process was completed by a single artist working with our system.

Figure 6: *A torch inspired by the illustration in Figure 2. Two particle systems are used to model beams for the smoke and fire.*

8 Results and Discussion

Our system was used in the production of two scenes (Figures 1 and 8) in the feature film *Tangled*. In Figure 1, curvy beams from a character's chest slowly fill a room, creating intricate, non-physical lighting both volumetrically and indirectly on surfaces. The artist for this scene designed spiral and flower shapes on paper, implemented them as curves in Houdini™, and used them as both paths and target shapes for beams.

Although we focus on volumetric effects, we note that beams can naturally be sampled as light sources for surface shading. In Figure 1, beams cast light on the characters' faces, as well as being reflected in the environment. Reflections were computed by ray-tracing beams as geometry. Figure 7 shows a simpler scene with procedurally generated beams which induce reflections and indirect diffuse lighting on other surfaces (see corresponding video clip).

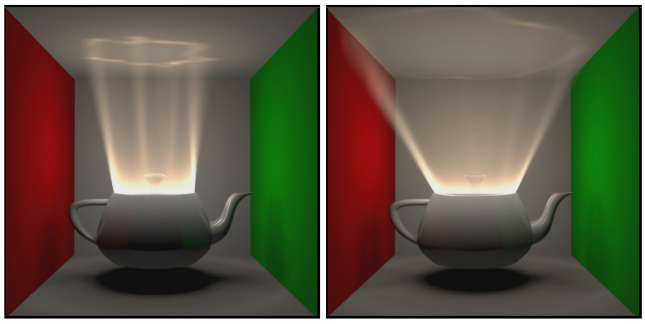

Figure 7: *Procedural beams can also be used to light surfaces. Note the reflections of the beams on the teapot and the indirect surface illumination induced by the beams.*

8.1 Point-based Global-illumination from Beams

To compute indirect diffuse transfer from beams onto surfaces, we use point-based global-illumination (PBGI) [Christensen 2008] and extend it to handle photon beams. First, we point-sample beam geometry and store the results in a regular grid. At shading time, the grid is down-sampled and the reduced set of samples are treated as point lights (with inverse squared fall-off) and used to shade all surface points. We optionally trace a shadow ray to each virtual source. Alternatively, shadows could be computed separately (with rasterization or ray tracing) at a lower spatial resolution.

8.2 Artist Feedback

Although we have not conducted a formal user study, we solicited feedback and gathered important usability observations during production. We categorize and summarize this information below.

Keyhole Sequence. Prior to the availability of our system, an initial keyhole scene had already been finalized for the teaser (Figure 8, top). However, after releasing a preliminary version of our system to artists, this shot was recreated from scratch using beams.

When designing the original keyhole sequence, our artist commented on the labor intensive process of modeling light streaks using tapered polygons which were carefully placed around the door cracks, as well as manually animated/deformed to match camera movement in a manner that achieved the desired *fanned-out* look. Several frustrations with this workflow were explicitly brought to our attention: difficulties designing the *flaring* effects of camera-facing lighting, the incorrect accumulation of these lighting geometries, and the limited patch-level control of noise which restricted the artist to a limited type of shader-level appearance manipulation.

In contrast, the same artist commented on several improvements our system provided for this scene. Generating light primitives was much simpler and faster, as beam starting locations were simply painted along the door cracks and any other emissive regions. Lighting was accumulated in a much more natural and predictable manner; the density of painted beams directly mapped to expected intensity variation. More intuitive and fine-grained control over beams was exposed via the beam attributes (e.g., `beamID`) used to steer fall-off, color, intensity, spread and directionality behavior. Beams behave correctly when intersecting the camera plane, eliminating the need for manual flaring composition.

The success of the keyhole shot prompted a lead artist to use our approach for the pivotal revival scene (Figure 1), and he explained that modeling and shading beams using our system afforded much more creative freedom than previous ad-hoc techniques. Work on this shot also prompted the development of the curvy beam construct, which was easily integrated into our modeling framework.

Target Matching. We asked an artist unfamiliar with our system to replicate an animation of a rotating torus in volumetric fog lit by a single area source (see supplemental document). The artist was first instructed to use any techniques and tools available. Afterwards, the artist was given a short introduction to our system and asked to once again replicate the result, this time with beams.

Without beams, the artist used a *ray surface operation* in Houdini™ to compute a deep shadow map (DSM) of the torus used to extrude a volume that approximated the lit media in the scene. Some additional processing of this explicit volume geometry was also necessary. With beams, the artist commented that the workflow was both more intuitive and more efficient, stating that beams were a "closer analogy" to the volumetric light. The artist elaborated on how the initial attempt (without beams) did not result in a very satisfying animation, required several "cheats", and was tedious to finalize. On the other hand, beams were simple to setup and manipulate, and resulted in a more satisfying design session.

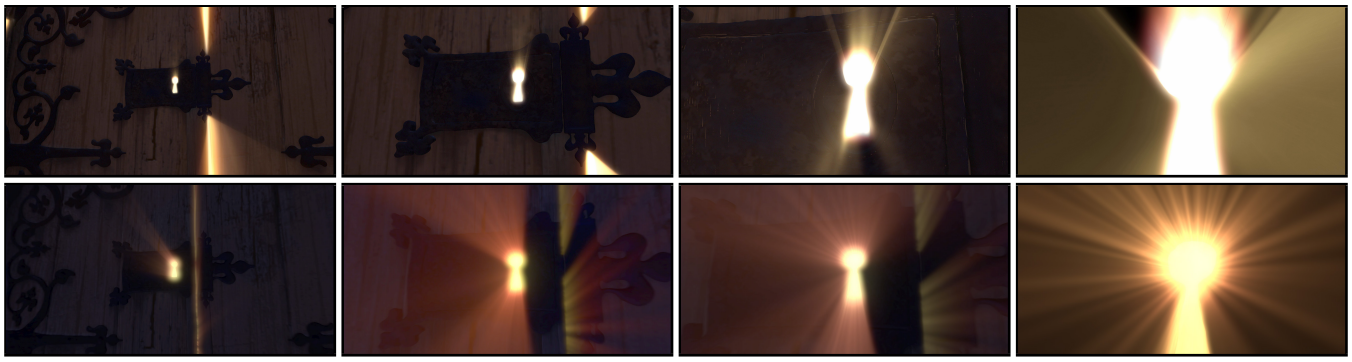

Figure 8: *A door jam and a keyhole emit volumetric light in these production scenes.* ***Top:*** *Sequence created using manually placed, static emissive curve geometry.* ***Bottom:*** *beams created by distributing particles at the door jam and keyhole, animating these beam origins, and shading with our model. Artists' comment on both the improved workflow and results generated using our system (Section 8.2).*

Common Shading Approaches. In *Tangled*, the most common deviations from physically-based shading involved f_b and f_e. For example, in Figure 1, beam attenuation was set according to custom splines and fall-off towards the eye was disabled, whereas the keyhole scene used physical fall-off along beams but non-physical fall-off towards the eye. Figure 9 illustrates the influence of similar changes to f_b and f_e. However, our artists made it clear that the settings of the shading model are driven by an artistic process that is very sensitive to the requirements of a particular sequence.

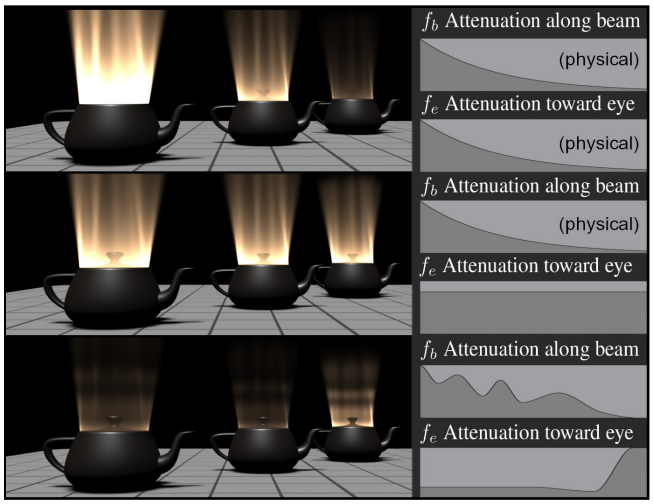

Figure 9: *Editing attenuation along beams and towards the eye.*

8.3 Rendering Backend and the Beams Metaphor

In general, our system can be implemented on top of almost any existing rendering backend. Any system capable of programmable shading of beam geometry is suitable, whether it be using RiCurves in PRMan, ray-tracing of beam geometry, or rasterization with programmable pixel shaders. We have investigated the use of our system in all three of these rendering backends. Our final results are generated using PRMan's REYES-style renderer, with the exception of a customized ray-tracer used for shadow-ray computation in our experimental PBGI approach. Furthermore, we developed an expression-editing front-end to our shader (see supplemental material for source code) which allows programmable control.

8.4 Limitations

One limitation we hope to address in future work stems from the fact that beams are not well suited to all types of volumetric phenomena in practice. However, wispy effects, which were traditionally difficult to model with either point primitives or density grids, can be easily authored with beams.

9 Conclusion

We have presented a system for generating art-directable volumetric effects. We observe that aspects of the physically-accurate photon beams approach relate closely to the methods artists normally use to reason about, and hand draw volumetric effects. By generalizing this approach and tying it to a geometric interpretation for volumes, and the lighting within them, we are able to expose an intuitive programmable model for designing volumetric effects.

Acknowledgements

We thank Peter-Pike Sloan for fruitful discussions on our system, Alessia Marra for sketching Figure 2, Bruce Wright for conducting the target matching test, Bruce Walter for the bumpy sphere geometry and lighting data, and the reviewers for their recommendations and corrections. Figures 1 and 8 are © Disney Enterprises, Inc.

References

ANGELIDIS, A., NEYRET, F., SINGH, K., AND NOWROUZEZAHRAI, D. 2006. A controllable, fast and stable basis for vortex based smoke simulation. In *SCA*, Eurographics Association, 25–32.

CHANDRASEKAR, S. 1960. *Radiative Transfer*. Dover Publications, New York, New York.

CHRISTENSEN, P. H., 2008. Point-based approximate color bleeding. Pixar Technical Memo 08-01, July.

COOK, R. L., CARPENTER, L., AND CATMULL, E. 1987. The reyes image rendering architecture. In *Computer Graphics (Proceedings of SIGGRAPH 87)*, 95–102.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 15–22.

GILLAND, J. 2009. *Elemental Magic: The Art of Special Effects Animation*. Focal Press.

HONG, J.-M., SHINAR, T., AND FEDKIW, R. 2007. Wrinkled flames and cellular patterns. *ACM Transactions on Graphics 26*, 3 (July), 47:1–47:6.

JAROSZ, W., NOWROUZEZAHRAI, D., SADEGHI, I., AND JENSEN, H. W. 2011. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics 30*, 1 (Jan.), 5:1–5:19.

JENSEN, H. W., AND CHRISTENSEN, P. H. 1998. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, 311–320.

KAJIYA, J. T. 1986. The rendering equation. In *Computer Graphics (Proceedings of SIGGRAPH 86)*, 143–150.

KERR, W. B., AND PELLACINI, F. 2009. Toward evaluating lighting design interface paradigms for novice users. *ACM Transactions on Graphics 28*, 3 (July), 26:1–26:9.

KERR, W. B., AND PELLACINI, F. 2010. Toward evaluating material design interface paradigms for novice users. *ACM Transactions on Graphics 29*, 4 (July), 35:1–35:10.

KERR, W. B., PELLACINI, F., AND DENNING, J. D. 2010. Bendylights: Artistic control of direct illumination by curving light rays. *Computer Graphics Forum 29*, 4, 1451–1459.

KŘIVÁNEK, J., FAJARDO, M., CHRISTENSEN, P. H., TABELLION, E., BUNNELL, M., LARSSON, D., AND KAPLANYAN, A. 2010. Global illumination across industries. In *SIGGRAPH Courses*, ACM.

MCNAMARA, A., TREUILLE, A., POPOVIĆ, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *ACM Transactions on Graphics 23*, 3 (Aug.), 449–456.

OBERT, J., KŘIVÁNEK, J., PELLACINI, F., SÝKORA, D., AND PATTANAIK, S. N. 2008. iCheat: A representation for artistic control of indirect cinematic lighting. *Computer Graphics Forum 27*, 4, 1217–1223.

OBERT, J., PELLACINI, F., AND PATTANAIK, S. N. 2010. Visibility editing for all-frequency shadow design. *Comput. Graph. Forum 29*, 4, 1441–1449.

PELLACINI, F., BATTAGLIA, F., MORLEY, R. K., AND FINKELSTEIN, A. 2007. Lighting with paint. *ACM Transactions on Graphics 26*, 2 (June), 9:1–9:14.

PELLACINI, F. 2010. envylight: An interface for editing natural illumination. *ACM Transactions on Graphics 29*, 4 (July), 34:1–34:8.

SADEGHI, I., PRITCHETT, H., JENSEN, H. W., AND TAMSTORF, R. 2010. An artist friendly hair shading system. *ACM Transactions on Graphics 29*, 4 (July), 56:1–56:10.

SCHMID, J., SUMNER, R. W., BOWLES, H., AND GROSS, M. 2010. Programmable motion effects. *ACM Transactions on Graphics 29*, 4 (July), 57:1–57:9.

SELLE, A., MOHR, A., AND CHENNEY, S. 2004. Cartoon rendering of smoke animations. In *Non-photorealistic Animation and Rendering*, ACM, 57–60.

SONG, Y., TONG, X., PELLACINI, F., AND PEERS, P. 2009. Subedit: A representation for editing measured heterogeneous subsurface scattering. *ACM Transactions on Graphics 28*, 3 (July), 31:1–31:10.

TABELLION, E., AND LAMORLETTE, A. 2004. An approximate global illumination system for computer generated films. *ACM Transactions on Graphics 23*, 3 (Aug.), 469–476.

TESSENDORF, J., AND KOWALSKI, M. 2010. Resolution independent volumes. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH, ACM.

TREUILLE, A., MCNAMARA, A., POPOVIĆ, Z., AND STAM, J. 2003. Keyframe control of smoke simulations. *ACM Transactions on Graphics 22*, 3 (July), 716–723.

WALTER, B., ZHAO, S., HOLZSCHUCH, N., AND BALA, K. 2009. Single scattering in refractive media with triangle mesh boundaries. *ACM Transactions on Graphics 28*, 3 (July), 92:1–92:8.

WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In *Computer Graphics (Proceedings of SIGGRAPH 78)*, 270–274.