Complex Luminaires: Illumination and Appearance Rendering

EDGAR VELÁZQUEZ-ARMENDÁRIZ, ZHAO DONG, BRUCE WALTER, and DONALD P. GREENBERG
Cornell University

Simulating a complex luminaire such as a chandelier is expensive and slow, even using state-of-the-art algorithms. A more practical alternative is to use precomputation to accelerate rendering. Prior approaches cached information on an aperture surface that separates the luminaire from the scene, but many luminaires have large or ill-defined apertures leading to excessive data storage and inaccurate results.

In this article, we separate luminaire rendering into illumination and appearance components. A precomputation stage simulates the complex light flow inside the luminaire to generate two data structures: a set of *anisotropic point lights* (APLs) and a radiance volume. The APLs are located near apparent sources and represent the light leaving the luminaire, allowing its near- and far-field illumination to be accurately and efficiently computed at render time. The luminaire's appearance consists of high- and low-frequency components, which are both visually important. High-frequency components are computed dynamically at render time, while the more computationally expensive low-frequency components are approximated using the precomputed radiance volume.

Results are shown for several complex luminaires, demonstrating orders of magnitude faster rendering compared to the best global illumination algorithms and higher fidelity with greatly reduced storage requirements compared to previous precomputed approaches.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Raytracing*

General Terms: Algorithms

Additional Key Words and Phrases: Luminaires, illumination, rendering, ray tracing, global illumination, preprocessing, virtual point lights

ACM Reference Format:

Edgar Velázquez-Armendáriz, Zhao Dong, Bruce Walter, and Donald P. Greenberg. 2015. Complex Luminaires: Illumination and appearance rendering. ACM Trans. Graph. 34, 3, Article 26 (April 2015), 15 pages.
DOI: http://dx.doi.org/10.1145/2714571

The authors gratefully acknowledge the support from Cornell University, Intel's Science and Technology Center for Visual Computing, and research funding for Global Illumination Rendering from Autodesk, Inc.

Authors' addresses: E. Velázquez-Armendáriz (corresponding author), Z. Dong, B. Walter, D. P. Greenberg, Program of Computer Graphics, Cornell University, Ithaca, NY; email: eva5@cornell.edu.

DOI: http://dx.doi.org/10.1145/2714571

1. INTRODUCTION

Why are depictions of luminaires mostly absent from computer graphics renderings? Luminaires are an important part of most environments: they provide artificial lighting, shape the illumination to suit our needs, and also often serve as decorative elements in their own right. A huge variety of luminaires are commercially available, ranging from simple spotlights to elaborate chandeliers. However, simulating real luminaires is frequently prohibitively expensive, and in computer rendering highly simplified representations are used instead. In this article we develop new efficient techniques to accurately simulate complex luminaires, enabling rendering with a much wider range of luminaires than was previously practical.

Real luminaires are assemblies that include both the sources of light emission (e.g., filaments or LEDs) and other associated geometry, such as reflectors, diffusers, brackets, and baffles, to form a complete lighting unit. They also contain a variety of materials such as lead glass, rough metals, plastics, and enamels that make them optically complex and expensive to simulate.

Existing simulation approaches can be divided into two categories: general global illumination algorithms and precomputation approaches. Global illumination algorithms are designed to simulate the total light flow within a scene. In theory, they can handle complex luminaires by simply including the luminaire geometry as part of the scene. In practice, however, the optics within a luminaire is often far more computationally challenging than in the rest of the scene, resulting in excessively long render times as we show later. Precomputation-based approaches simulate the luminaire a priori to create a simplified representation, or impostor, that is faster to evaluate. The impostor is typically a single point light or an area light over an aperture, which is a surface that separates the luminaire from the rest of the scene. In the lighting industry, a single point far-field representation has been widely adopted to represent luminaires due to its simplicity of acquisition and use (e.g., IES LM-63 [IESNA 2002]). However, these only work well when the luminaire or its aperture is small. We present a new precomputation approach that is designed to work, even for more complex luminaires whose aperture may be large or ill defined (e.g., Figure 1).

In our approach, we first divide the problem into two components: luminaire *illumination* (the luminaire's effect on the rest of the scene) and luminaire *appearance* (what we see when looking at the luminaire). Illumination and appearance have very different visual characteristics and requirements. For the illumination, we construct a set of *anisotropic point lights* (APLs) whose positions and directional distributions are optimized to reproduce the luminaire illumination in both the near- and the far-field. Our APLs are not constrained to lie on an aperture, thus enabling higher accuracy, and are true point sources, making them easy to support and evaluate.

The appearance of luminaires such as a chandelier can be quite detailed and computationally challenging. We have observed that the appearance often combines high- and low-frequency components. The high-frequency components are often highly view-dependent making them difficult to precompute or store. Instead we compute these dynamically at render time using depth-limited ray tracing to preserve visual detail while limiting cost. The low-frequency

(a) a single point light source with a directionally varying goniometric far-field emission profile

(b) a canned light-source method using six data-intensive light fields on the bounding box

(c) our method using precomputed APLs and radiance volume, along with the luminaire's geometry

Fig. 1. Comparison of three precomputation-based rendering methods for a room with a complex chandelier luminaire. (a) Single point far-field representations are fast and widely used, but produce visibly inaccurate illumination and shadows for this scene and do not provide a way to depict the luminaire's appearance; (b) the canned light-source method replaces the luminaire by a light-field proxy on its bounding surface. Light fields have difficulty reproducing high-frequency features and are susceptible to aliasing, causing objectionable artifacts in the illumination and appearance. The black region around the luminaire is due to using the bounding surface proxy to replace the luminaire geometry; (c) our method reproduces the correct illumination and shadows while also providing a visually good depiction of its appearance. Our method is the first practically feasible technique for rendering with luminaires such as this one.

components, in contrast, often involve light that has scattered many more times, becoming less view-dependent, but also more expensive to compute. We precompute and store a low-resolution radiance volume that records light inside the luminaire. We then query this radiance volume to quickly estimate lower frequency, but still visually important, aspects of the appearance.

For each luminaire model, we simulate its light flow in a preprocess by tracing light particles from the sources until they exit the luminaire. From this particle data we construct our two data structures: the APLs for illumination and radiance volume for appearance. This data can then be reused at render time for all instances of that luminaire, either within a single scene or across multiple scenes. Potentially, manufacturers could provide this data for their luminaires in the same way they supply geometry and far-field goniometric data for them now. We show results for several complex luminaires and demonstrate orders of magnitude speedup compared to general global illumination algorithms. Our methods provide higher accuracy at smaller data sizes compared to previous precomputation approaches, allowing to render with luminaires that were previously infeasible.

2. RELATED WORK

In this section, we review the prior work related to luminaire illumination and luminaire appearance, from both computer graphics and the lighting industry.

Far-Field Illumination. Illumination is often divided into near-field and far-field regimes based on distance. In the near field, the luminaire's spatial extent matters and the illumination is a 4D function of position and direction, while, in the far-field limit, the luminaire acts like a point source and the illumination simplifies to a 2D function of direction only. Because they are much easier to compute, measure, and store, far-field approximations are widely used. The far field is often defined as the region farther than $5\times$ the length of the longest dimension of a luminaire [Rea 2000], though the transition is actually gradual and luminaire-dependent.

Standard luminaire models, such as IES LM-63 [IESNA 2002] and EULUMDAT [Ashdown 2001], provide the far-field luminance intensity distribution, often measured from a real luminaire by a goniometer. Simple far-field point-source models are the lighting industry standard, with data provided directly by the luminaires manufacturers and supported by the rendering and lighting simulation software packages [Ward 1994; Driemeyer 2008]. Refinements on the far-field models include using spectral data and substituting a uniform area source to generate more plausible soft shadows [Verbeck and Greenberg 1984]. Although widely used, the far-field models are not accurate enough for lighting simulations of architectural interiors [Ashdown 2001].

Near-Field Methods. Near-field lighting can be represented by measuring the illumination at multiple distances or even by capturing its full 4D light field [Ashdown 1993; Levoy and Hanrahan 1996; Gortler et al. 1996]. Measuring this light field requires sensors close to the luminaire and is referred to as near-field photometry. Ashdown [1993] captures the luminaire's light field using a special goniophotometer with low-resolution cameras to limit the amount of data. The canned light-sources method [Heidrich et al. 1998] uses a computer simulation to generate the light field for a luminaire in a preprocess which can be reused at render time to accelerate the illumination computations. Goesele et al. [2003] project the illumination onto a finite basis using optical filters prior to measurement to avoid aliasing artifacts during acquisition. They also support the use of higher-order basis functions which can be advantageous when the light field is sufficiently smooth. Light-field reconstruction can be further accelerated using graphics hardware [Granier et al. 2003].

Ray sets [Ashdown and Rykowski 1998] are another way to represent illumination. A basic ray set is a set of exiting rays (origin and direction), with equal energy and without spectral distribution, which is stored on the luminaire's bounding surface or aperture, though many variants exist [Muschaweck 2011]. Ray sets can represent arbitrary illumination but are data intensive, and expensive to query. To accurately represent the illumination emanating from a complex luminaire, many billions of exiting rays have to be traced, which are far too many to be reasonably stored as a ray set. Mas et al. [2008] compress ray sets by clustering the rays to create hemispherical distributions stored at vertices on a bounding mesh. The resulting data structure is similar to an irregularly sampled 4D light field. Their decimation process is a form of agglomerative clustering but, unlike our method, they restrict their clusters to lie on a predetermined bounding surface and use a metric based on preserving the surface's radiant exitance under linear interpolation.

In contrast, our clustering tries to create roughly equal-power, spatially compact clusters within the volume of the luminaire. In our method, we generate many billions of exiting rays (far too many to reasonably store as a ray set) and, for our example luminaires, the APL point sources are more accurate, compact, and easier to use than 4D light-field approaches.

Luminaire Appearance. The accuracy requirements for luminaire appearance are frequently different than for illumination. Higher-frequency components such as sparkles and edges are more important and the surroundings can affect appearance through effects such as transparency and reflections. Light-field methods, such as canned light sources [Heidrich et al. 1998], can easily reproduce the low frequency components of appearance, but the storage requirements grow rapidly with luminaire size and appearance complexity. Following standard practice, we store the 4D light field on the faces of the luminaire's bounding box. This results in black borders whenever the luminaire's silhouette differs from its bounding box (e.g., Figure 1(b)). In theory, this could be mitigated by adding a 4D transparency channel to the light field, or by storing the light field on alternate surfaces that more closely match our complex silhouettes. However, to our knowledge these approaches have not yet been demonstrated and are left as future work. Kniep et al. [2009] proposed using directional photon mapping [Moon and Marschner 2006] to reproduce the appearance of complex car tail-light assemblies. They store photons on the luminaire bounding surface and then reconstruct its appearance using density estimation at render time. The visual appearance results are impressive, but require storing gigabytes of photon data for each luminaire. These methods are too data intensive to be practical for our larger, complex luminaires.

Global Illumination Methods. General-purpose global illumination methods, such as *bidirectional path tracing* (BDPT) [Lafortune and Willems 1993; Veach and Guibas 1995] and stochastic progressive photon mapping [Hachisuka and Jensen 2009], simulate the complete light flow in an environment, though they can be computationally expensive. Several recent methods have been proposed to improve the handling of difficult light paths. For example, manifold exploration [Jakob and Marschner 2012] greatly improves the exploration of connected caustic components, and two recent methods [Georgiev et al. 2012; Hachisuka et al. 2012] combine the strengths of photon mapping and BDPT to improve handling caustic components. While these methods are significant improvements to the state of global illumination rendering, examples such as our Statler luminaire (shown in Figures 3 and 4) still exhibit slow convergence and long render times.

Moon et al. [2007] presented a method to rapidly compute light scattering in randomly distributed refractive elements; however, this work is not applicable to our examples as our luminaire elements are not randomly distributed. Kaplanyan and Dachsbacher [2013] selectively modify the scene materials to simplify the handling of difficult paths, but our models already include surface roughness. It is hard to predict whether modifying the roughness can significantly reduce rendering cost without introducing too much error and unclear how to select such parameters in practice.

Irradiance Volumes. Irradiance volumes [Greger et al. 1998] store low-frequency lighting information at lattice points in a scene and are used in many *precomputation-based rendering* (PBR) methods [Ramamoorthi 2009], including hair rendering [Moon et al. 2008]. The lighting at each grid point is often stored in terms of spherical harmonics [Ramamoorthi and Hanrahan 2001; Sloan et al. 2002; Moon et al. 2008] for improved efficiency.

Our method can be combined with existing global illumination algorithms, using our precomputation for subpaths within a luminaire while using a global illumination algorithm to compute the indirect illumination in the rest of the scene. Because we use point lights, our method can be accelerated using scalable many-light algorithms such as Lightcuts [Walter et al. 2005; Dachsbacher et al. 2012] that automatically provide a light hierarchy and level-of-detail (or cut) selection. We demonstrate that even a simpler distance-based hierarchy strategy can result in significant cost reduction. In our appearance computations we use a variant of the radiance volume from Moon et al. [2008].

3. MOTIVATION

Our goal is to efficiently render environments with complex luminaires. As a motivating example, we were inspired by the chandeliers in the Statler Hotel (Figure 2(a), left) that contain 12 light sources within 117 glass shades. As a test, we built a computer model based on this chandelier (Figure 2(b), right) and room. While not an exact match for the real luminaire, our model is close enough to provide a similar look and computational challenge. When we tried rendering our model with existing methods, we found that none was a practically feasible solution. The global illumination algorithms had high noise and required hundreds of computation hours to get near a converged image, while light-field-based methods yielded poor visual quality even when set to use many gigabytes of precomputed data. The lack of a practical way to render this luminaire is what led us to develop our new luminaire rendering algorithm.

The model rendered with our method (Figure 2(b)) was also rendered with several different algorithms, and cropped images shown in Figures 3 and 4 showcase luminaire appearance and illumination, respectively. The figures use customized tone-mapping to highlight the different aspects, but all rendered images within a single figure use exactly the same tone mapping to allow fair comparisons. The global illumination algorithms, bidirectional path tracing [Veach and Guibas 1995], manifold exploration [Jakob and Marschner 2012], and stochastic progressive photon mapping [Hachisuka and Jensen 2009], were set to use equal rendering time as our method. The reference is rendered with bidirectional path tracing with a much longer render time. Render time excludes our precomputation because it can be reused and amortized across all uses of the same luminaire, but including it would not have changed the conclusions of this comparison. As a representative of light-field approaches, we also show canned light sources [Heidrich et al. 1998] set to use eight times more precomputed data than our method. These results demonstrate that our method produces much higher quality than prior methods given similar resources, and is the first practical rendering method demonstrated for such luminaires.

3.1 Luminaire Illumination Insights

In our method, we first split luminaire rendering into illumination and appearance components because we found these have significantly different accuracy and cost requirements. Illumination is generally the dominant cost, since luminaires usually occupy only a small fraction of the image pixels. Far-field point-source approximations are cheap to evaluate, but not accurate for near-field illumination. Because small sources tend to create less desirable effects such as harsher lighting and hard shadows, luminaires are often designed to be big enough such that much of the scene lies in the near field. Light fields can reproduce the near field of small sources well, but scale poorly with luminaire size and require a potentially expensive integration over slices of the light field. For our luminaires we need a method that reproduces both near- and far-field patterns, scales to large luminaires, and is efficient to evaluate. Inspired by

(a) actual scene photograph (b) our method

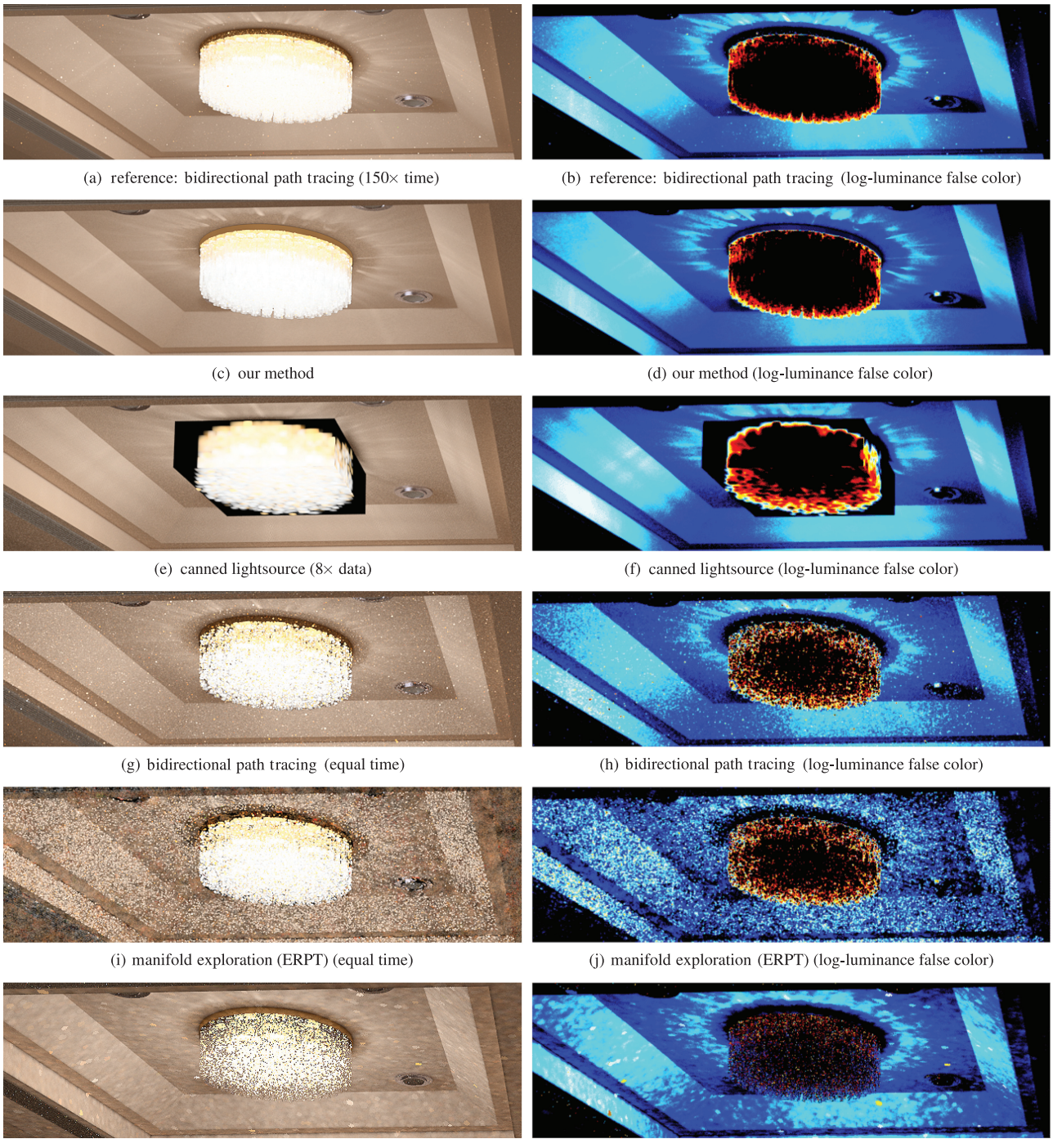
Fig. 2. Example scene lit by four chandeliers (Statler see Figure 13–based on a real-world luminaire). Such complex luminaires cast intricate patterns on their immediate surroundings and are aesthetically pleasing themselves, as shown in the photograph (a). Our method (b) efficiently and accurately renders both near- and far-field illumination from such complex luminaires at significantly faster speeds than previous methods. Our results achieve comparable effects to those present in the real scene.

(a) reference (150× time) (b) our method (c) canned light-source (8× data)

(d) actual scene photograph

(e) bidirectional path tracing (equal time) (f) manifold exploration (ERPT) (equal time) (g) stochastic progressive photon mapping (equal time)

Fig. 3. Comparison of the appearance of the Statler luminaire as rendered by different methods. The images correspond to cropped sections of Figure 2 using the histogram equalization tone mapping. Our method shows attributes for the illumination and the luminaire appearance qualitatively similar to those present in the real-world scene.

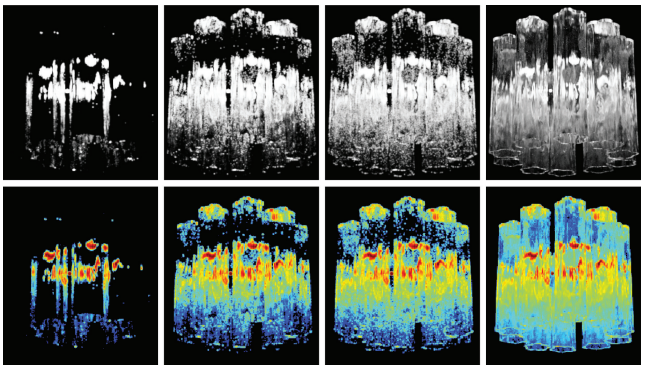


(a) reference: bidirectional path tracing (150× time)
(b) reference: bidirectional path tracing (log-luminance false color)
(c) our method
(d) our method (log-luminance false color)
(e) canned lightsource (8× data)
(f) canned lightsource (log-luminance false color)
(g) bidirectional path tracing (equal time)
(h) bidirectional path tracing (log-luminance false color)
(i) manifold exploration (ERPT) (equal time)
(j) manifold exploration (ERPT) (log-luminance false color)
(k) stochastic progressive photon mapping (equal time)
(l) stochastic progressive photon mapping (log-luminance false color)

Fig. 4. Comparison of the illumination from the Statler luminaire incident on the ceiling as rendered by different methods. The images shown in this figure correspond to a cropped section of the scene from Figure 2; images on the left are tonemapped using a global operator, and further enhanced by a sharpening filter to highlight the illumination patterns on the ceiling. The false-color images on the right illustrate the logarithmic luminance, making the variations in the illumination more apparent.

the success of many-light methods, we developed a new point-based method to meet these requirements.

For each luminaire we generate multiple anisotropic point lights (APLs), each with its own position and directional distribution. The summed effect of the APLs is optimized to closely match the luminaire's illumination in both the near and far field. Individually our APLs are very similar to traditional far-field single point sources, making them easy to support and evaluate. Multiple scattering and



(a) ≤ 5 bounces (b) ≤ 10 bounces (c) ≤ 30 bounces (d) reference

Fig. 5. This figure shows a simplified version of the Statler luminaire, rendered using path tracing with different maximum path lengths. The bottom row shows a false-color version of the top row to better illustrate the radiance variation. Images should be compared to a reference image with unbounded path lengths (d).

occlusion within the luminaire are already included in the APL properties, allowing the illumination to be computed solely from the APLs without testing against the luminaire geometry. We found that the choice of APL positions strongly affects the near-field quality, with the best locations often being in the interior rather than on an aperture or bounding surface. Thus we present an APL generation algorithm based on clustering to automatically select good locations. We also show that the number of APLs can be easily varied statically during precomputation, or selected dynamically during rendering (see Section 7) to optimize accuracy versus cost trade-offs.

3.2 Luminaire Appearance Insights

To investigate how to render the luminaire appearance, we first performed a series of experiments to understand the composition of such prohibitively complex effects. Figure 5 shows a series of appearance approximations for a simplified Statler luminaire (Figure 13(a)) which contains only four emitters surrounded by a single ring of glass shades. The experiments rendered this luminaire using the standard recursive path tracing while limiting the maximum number of ray bounces $\mathbf{n}_b$ to be 5, 10, and 30, respectively, along with a reference solution using unlimited bounces. As shown in Figure 5, there are some high-intensity and individually distinguishable light patterns in the luminaire's appearance which are already visible when $\mathbf{n}_b \leq 5$. These high-frequency light patterns are what we call *sparkles*, created by short paths connecting the emitter to the camera. When $\mathbf{n}_b \leq 10$, these high-frequency components of the luminaire appearance are already quite close to the reference while the major differences lie in the overall "glow" of the luminaire due to the multiple scattering within the complex luminaire. To show patterns more clearly, in the second row of Figure 5, each rendered result in the first row has been colorized using a false-color map covering the same data range (blue represents low, red represents high) to show the logarithmic intensity of each pixel. Clearly, most of the red sparkles have appeared when the number of ray bounces is less than or equal to 10 times, while the blue glow is still increasing even after 30 bounces.

Motivated by their very different characteristics, we choose to render sparkle and glow components using different strategies. High-frequency sparkles are well approximated by limiting the maximum number of bounces during ray tracing. Because the glow appears to be a very smooth signal, it may be approximated using a precomputed low-frequency representation.

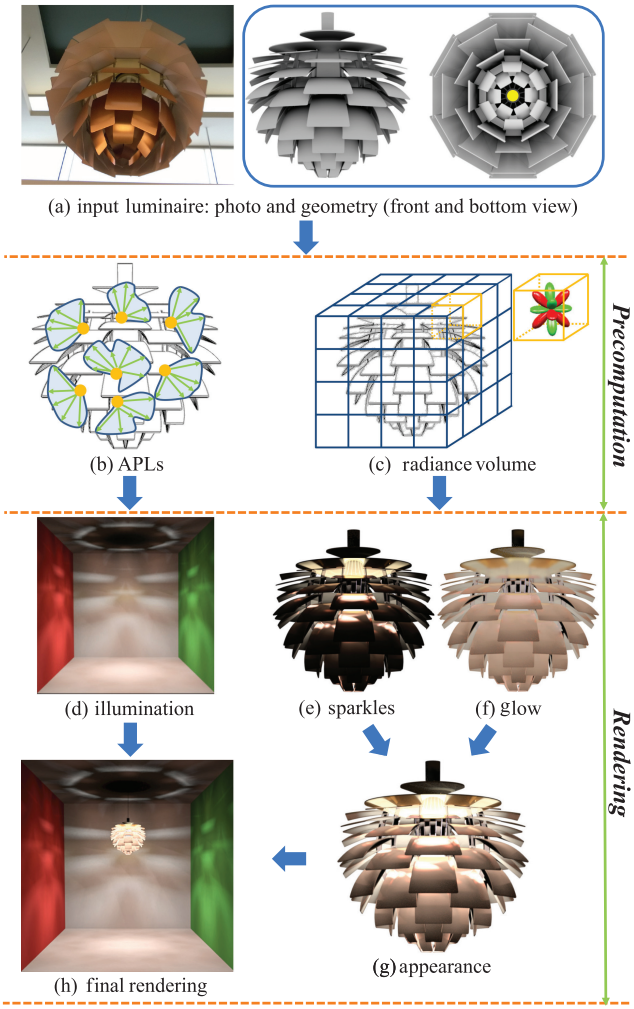


Fig. 6. Pipeline of our luminaire rendering method.

4. METHOD OVERVIEW

Figure 6 shows an overview of our luminaire rendering method using the P.H. Artichoke luminaire as an example. In a preprocess, we create two data structures for each luminaire model: a set of anisotropic point lights (APLs) and an internal radiance volume. The preprocess uses standard light-particle tracing with particles emitted from the luminaire's emitters and tracked until they exit the luminaire. The exiting rays are clustered to form the APLs which compactly represent all the light leaving the luminaire. Each APL consists of a position and a directional distribution (Figure 6(b)). The radiance volume is a low-frequency representation of the light flow inside the luminaire that is stored as a low-resolution 3D grid of spherical harmonic coefficients (Figure 6(c)). During the particle trace, each particle track is projected into the radiance volume such that each cell stores the average radiance within that cell. The APLs and radiance volume are constructed incrementally so that we do not store the full set of particles.

At render time, points are shaded using either the appearance procedure or illumination procedure, depending on whether they are located on the luminaire geometry or elsewhere in the scene. The illumination, at points not on the luminaire, is computed by evaluating the direct illumination from the precomputed APLs (Figure 6(d)). This is identical to evaluating standard directionally varying point sources, except that we need only check for shadow occlusion from non-luminaire geometry. The APL distributions already account for occlusion (and multiple scattering) due to the luminaire geometry.

Shading points on the luminaire use an appearance procedure based on limited-depth recursive ray tracing. At each point, we compute the direct illumination from the luminaire's actual emitters and generate a scattered ray by BSDF sampling. The scattered ray is recursively traced to generate a new point until a ray leaves the luminaire or reaches the maximum recursion depth. If the ray leaves the luminaire, then it is shaded using the preceding illumination procedures. This is essential to compute scene-dependent aspects of the luminaire appearance from effects such as transparency or reflection. When the maximum recursion depth is reached, we query a value from the precomputed radiance volume and terminate the ray. As discussed in Section 3.2, luminaire appearance contains high-frequency sparkles (Figure 6(e)) and the low-frequency glow (Figure 6(f)). The sparkles are typically caused by relatively short, high-intensity, often specular, ray paths to the emitters. Since the sparkles are highly view-dependent, precomputation is not an effective approach and instead a limited-bounce ray tracing scheme is used to compute them. The glow, in contrast, is typically caused by highly scattered light from much longer paths inside the luminaire. The multiple scattering diffuses this light, making precomputation using our radiance volume an effective strategy. While the glow approximation is less accurate than the other parts of our method, we found it a good trade-off between quality and performance, and essential in achieving visually good appearance fidelity.

Together, our illumination (Figure 6(d)) and appearance (Figure 6(g)) procedures enable high-quality rendering of scenes with complex luminaires (Figure 6(h)).

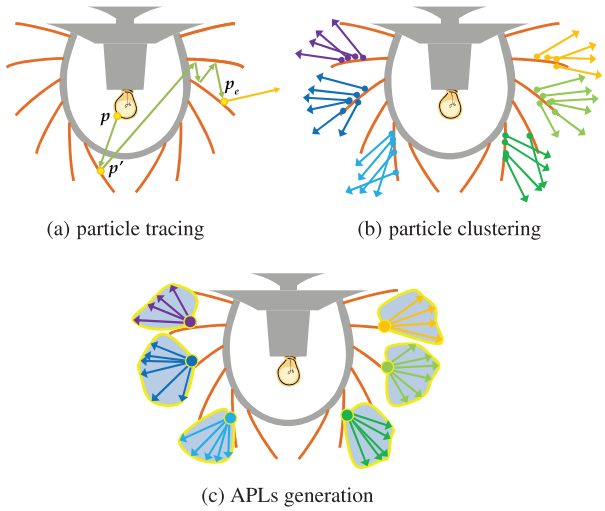
5. ALGORITHM

This section first presents the particle tracing step in the precomputation stage and how to use it to generate a set of APLs and a low-frequency radiance volume for the input luminaire. Afterward, we describe how to utilize these two precomputed data structures, by combining with a depth-limited ray tracing scheme for generating sparkles to fully render both accurate illumination and plausible appearance of a complex luminaire.

5.1 Particle Tracing

The precomputation stage starts by tracing random particles from each emitter through the luminaire geometry until they exit into the environment. A standard particle tracer, similar to those used for photon mapping and traditional VPL generation, generates the particles. This step only requires the luminaire with its geometry, materials, and emitters as input; hence it is scene independent.

As shown in Figure 7(a), for each particle $\mathbf{p}$ we record its starting position $\mathbf{x_p}$, direction $\mathbf{v_p}$, flux $\Phi_\mathbf{p}$, and path length $\mathbf{t_p}$; the latter is defined as the distance from $\mathbf{x_p}$ along $\mathbf{v_p}$ to the next intersection point within the luminaire geometry. When $\mathbf{p}$ intersects the luminaire geometry, a new particle $\mathbf{p}'$ is generated and the $\mathbf{t_p}$ of $\mathbf{p}$ is determined at the same time. Following standard practice, the direction and flux of $\mathbf{p}'$ are determined by importance sampling of the BSDF at the current intersection point. Such a particle tracing process continues until the particle $\mathbf{p_e}$ exits the luminaire geometry.



(a) particle tracing (b) particle clustering

(c) APLs generation

Fig. 7. Particle tracing and clustering in precomputation.

To faithfully represent both the illumination and appearance of a complex luminaire, we need to trace millions, even billions, of particles through the luminaire geometry. For the illumination we use only those exiting particles $\mathbf{p_e}$ which escape from within the luminaire to the surrounding environment (yellow vector in Figure 7(a).) Section 5.2 describes how to convert these particles into APLs. To generate the radiance volume for luminaire glow, we use all the particle segments $\mathbf{t_p}$ within the luminaire (green vectors in Figure 7(a)); Section 5.3 explains this process in more detail.

5.2 Generating APLs Using Clustering

Instead of storing all the exiting particles, we choose to group them into a set of clusters (Figure 7(b)), and treat each cluster as an APL (Figure 7(c)). We explored several different clustering methods, including k-means clustering, and achieved the best result with agglomerative clustering [Walter et al. 2008] using a metric that aims to generate spatially compact clusters with roughly equal power (i.e., flux). The clustering cost metric we used is

$$I = (\Phi_{C_1} + \Phi_{C_2}) \cdot \mathrm{Diag}(C_1 \cup C_2)^4, \tag{1}$$

where Φ_{C_1} and Φ_{C_2} are the powers of the clusters C_1 and C_2 which are potentially being merged and $\mathrm{Diag}(\cdot)$ is the diagonal length of the bounding box enclosing both C_1 and C_2. Unlike some ray-set methods, we do not require that all particles have equal power, as our metric and clustering method explicitly account for power.

Since agglomerative clustering builds a complete tree bottom-up from the leaves, the outcome of this clustering step is actually a binary tree; its leaf nodes correspond to individual particles. Each inner node of the binary tree represents a cluster containing all those child clusters below it in the tree. Clearly, the clustering cost of the tree nodes decreases from the tree, top to bottom. To generate a desired number of APLs, we use a *maximum priority queue* filled with tree nodes where the associated priority is computed using Eq. (1). The queue is initialized with the root node of the cluster tree. Then we iteratively remove from the queue the node with the highest clustering cost and add to the queue its two children from the binary tree. We repeat this process until the queue reaches the target

size (e.g., 512 nodes), and then each node in the queue becomes a cluster for generating the set of APLs.

Performing agglomerative clustering on a billion particles is unnecessarily expensive for generating high-quality clusters. Actually, clustering a subset with a much smaller number of particles (e.g., a million) is enough to generate good cluster centers, even though all the particles are required to get good directional resolution for each APL. Thus we first perform agglomerative clustering on an initial subset of particles to seed the cluster locations. Thereafter, every exiting particle can be simply assigned to the closest cluster to it, using the Euclidean distance between the particle's position $\mathbf{x_p}$ and the cluster's centroid.

Each cluster keeps track of its centroid and the directional radiance distribution of all particles assigned to it. At the end of the precomputation, these clusters turn into a set of APLs (Figure 7(c)), where the centroid of each cluster becomes the APL's position. To estimate the radiance distribution of an APL, we only use the power and direction of the particles within its corresponding cluster. Each particle gets assigned to a bin using a 2D projection of the sphere. After accumulating the contribution from all the relevant particles, we store the resulting high dynamic range bitmap to be queried during rendering.

5.3 Radiance Volume for Appearance

To generate the radiance volume for the luminaire low-frequency glow, we use all the particle segments within the luminaire (green vectors in Figure 7(a)). Keeping all these segments in memory would be impractical due to the excessive storage requirements. Instead, we project them into a low-resolution volume to create a more compact approximation of the spatially and directionally varying radiance field inside the luminaire.

During precomputation, the bounding box of the luminaire model is first discretized into a low-resolution uniform grid with cube-shaped voxels. To estimate the radiance within the grid from the particle data, we adopt the approach of Moon et al. [2008]. For each voxel that a particle segment intersects, it makes a contribution to the radiance, proportional to the path segment length within this voxel. The radiance estimate for each voxel is generated by gathering the contributions from all the intersecting light paths. The directional dependence is stored by projection onto spherical harmonics using a fixed number of coefficients per voxel. For the j-th voxel and k-th spherical harmonic, this coefficient is computed as

$$c_{j,k} = \sum_{\mathbf{p}} \frac{\ell_{\mathbf{p},j}\, Y_k(\mathbf{v_p})\, \Phi_{\mathbf{p}}}{V_j}, \tag{2}$$

where $Y_k(\cdot)$ is the k-th spherical harmonic basis function, V_j the voxel's volume, and $\ell_{\mathbf{p},j}$ the segment length within voxel j (thus $\ell_{\mathbf{p},j} \leq \mathbf{t_p}$). Recall that each particle segment $\mathbf{p}$ has a direction $\mathbf{v_p}$, length $\mathbf{t_p}$, and flux $\Phi_{\mathbf{p}}$. Equation (2) is an estimator of the radiance averaged over the volume of the voxel and projected onto a spherical harmonic basis. Please refer to Moon et al. [2008] for more details. At render time, this low-resolution volume can be quickly queried to approximate the low-frequency radiance for any position and direction inside the luminaire.

5.4 Rendering Process

At rendering time it is possible to use the APLs as the light emitting primitives. Since each individual APL is a true point source, any appropriate rendering routine can be used to gather their direct illumination and associated GI effects. One minor difference is that shadow rays to APLs are not blocked by luminaire geometry, since luminaire-internal occlusion is already baked into the directional distributions of each APL after precomputation.

To combine both the high- and low-frequency components of the luminaire appearance, we use a recursive ray tracing process. When an eye ray hits the luminaire geometry, by sampling the BSDF at the intersection point, we continue to track its path recursively to accumulate the sparkles up to a limited number of additional bounces. If the view ray hits an emitter inside the luminaire before reaching the maximum depth, we include this energy as a sparkle. The maximum ray depth $\mathbf{n}_b$ varies for the different geometry complexities of the luminaire models. If the view ray reaches the maximum depth while still remaining within the luminaire, we simply query the precomputed low-frequency radiance volume at the last ray intersection point and in its incident direction to compute the glow effect, terminating the path. When the eye ray exits the luminaire before $\mathbf{n}_b$ bounces, it simply gathers the incident radiance from all relevant sources.

ALGORITHM 1: Pseudocode of the rendering routine.

```
Function(LumRender(Intersection Record p, int iBounce))
    if p is on the luminaire geometry then
        Generate the new ray l by sampling the BSDF at p;
        if iBounce ≤ n_b then
            if l hits the emitter then
                Return the energy L_e from emitter;
            end
            Compute new intersection record p' with scenes;
            LumRender(p', iBounce + 1);
        else
            Compute the "glow" L_g using radiance volume;
            Return L_g;
        end
    else
        At p, the reflected radiance L_r = 0;
        for each generated APL do
            Query the incident radiance L_i from current APL;
            Modulate L_i by visibility and BSDF at p;
            L_r = L_r + L_i;
        end
        Return L_r;
    end
```

Algorithm 1 shows a simple variation on a traditional path tracer which illustrates the entire routine to render both the illumination and appearance of a luminaire. When an intersection record $\mathbf{p}$ between the eye ray and the scenes is computed, the routine first checks whether $\mathbf{p}$ is located on the luminaire. If $\mathbf{p}$ is on the luminaire, our appearance rendering method is applied to render the complex luminaire appearance. Otherwise, the reflected radiance L_r at $\mathbf{p}$ is computed by gathering all the incident radiance from the precomputed set of APLs. Note that Algorithm 1 only shows the simple illumination rendering routine that iterates over all the APLs without using an adaptive method such as a hierarchy.

5.5 Implementation Details

There are several parameters and design choices possible for implementing our method, all of which constitute different trade-offs between quality and both computational and memory costs.

Table I. Performance Data for Precomputation (in minutes)

	Luminaire				
Phase	Troffer	Artichoke	Statler	Sputnik	V&A
Particle trace	8.9	34.5	180.5	23.5	25.0
APLs gen.	2.5	2.7	2.6	2.6	2.4
Radiance vol.	4.2	4.7	1.3	3.5	2.8
Total	15.6	41.9	184.4	29.6	30.2
Rad. vol. size	$84\times10\times42$	$32\times33\times32$	$48\times15\times48$	$37\times26\times37$	$25\times56\times25$

The number of initial particles on which we run agglomerative clustering to seed the clusters is one million; overall, we use one billion particles to generate both the APLs and the radiance volume.

By default, we generate 512 APLs per luminaire. The directional radiance distribution for each APL is stored as a 2D texture map of 256×512 resolution using the concentric mapping projection [Shirley and Chiu 1997]. The pixels store RGB values in Float16 format.

The low-frequency radiance volume grid for each luminaire has 32^3 voxels on average, arranged across the luminaire's axis-aligned bounding box. Since the bounding box dimensions differ for different luminaires, the actual number of voxels along each axis varies depending on the shape of the corresponding luminaire model. Each voxel stores the first 16 SH coefficients (i.e., the first four bands) for each of the RGB components in Float32 format.

6. RESULTS AND EVALUATIONS

In this section, we present results to evaluate the accuracy and performance of our method for five complex luminaires.

—*Troffer.* This luminaire is an overhead office light, with three fluorescent ballasts and a grid refractor, exported from Autodesk Revit 2012.

—*P.H. Artichoke.* This is a single spherical light source surrounded by metallic leafs that is based on a design by Poul Henningsen.

—*Statler.* This luminaire consists of 12 light sources arranged in two concentric circles, surrounded by 117 glass shades and is based on the chandeliers at the Statler Hotel.

—*Met Sputnik.* This is made up of 22 light sources suspended between glass pendants, and is based on the design by Hans Harald Rath for J. & L. Lobmeyr at the New York Metropolitan Opera foyer.

—*V&A Chandelier.* This is a glass chandelier with 18 light sources that is based on a model at the Victoria and Albert Museum, London.

The actual emitters are modeled as small spherical area sources with diffuse emission. The glass components are modeled as 3D solids with a microfacet-based BSDF [Walter et al. 2007] to simulate slightly roughened glass, which we found essential for faithfully reproducing the characteristic glow seen in the real luminaires.

Our implementation is written in C++ and built on top of the Mitsuba rendering framework [Jakob 2010]. The timings were measured on a PC equipped with a 4-core, 8-thread Intel i7-4771 Haswell CPU, running Windows 8.1, 64 bit. All the reference images are generated using *bidirectional path tracing* (BDPT.)

6.1 Precomputation Performance

Table I shows the timing of the different steps during precomputation. The particle tracing step consumes most of the precomputation time. Its performance depends on the geometric complexity of the luminaire model and the number of particles. We traced one billion particles for each luminaire. Our most geometrically complex luminaire, Statler, with three million triangles, took just over three hours for particle tracing. When generating the radiance volume, the path formed by each particle is intersected with the volume grid and projected into the spherical harmonic basis. Fortunately, due to the low-frequency nature of the glow component, a low-resolution (e.g., 32^3) volume grid was sufficient even for complex luminaires such as Statler and the V&A Chandelier. The precomputed data depends only on the luminaire configuration and hence is scene independent. The data needs only to be computed once per luminaire model and can be further reused in different scenes or across multiple luminaire instances.

6.2 Quantitative Evaluation of Illumination

In order to quantify the illumination accuracy provided by the generated APLs, we conducted a series of experiments inspired by the integrating spheres used for analyzing actual light fixtures [Rea 2000]. We computed the irradiance on a series of virtual measurement spheres centered on the luminaire at various radii, using either all the particles emitted from the luminaire (as the *reference*) or our APLs. Distances are measured relative to the diameter of the luminaire's bounding sphere, so a distance of 0.5 corresponds to the surface of its bounding sphere, and a distance of five corresponds to the traditional start of the far-field region.

The error metric we use is the *root-mean-square error* (RMSE) of the irradiance $\mathbf{E}$ measured on equal-area patches of the virtual measurement sphere relative to the average irradiance across all patches. The formula to compute the error metric is

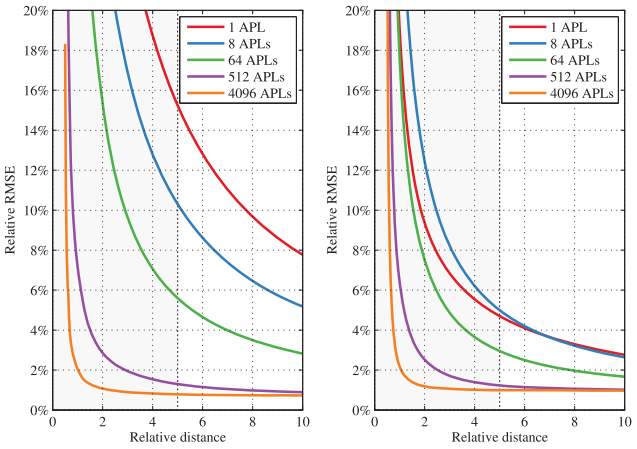
$$\text{Relative RMSE} = \frac{\sqrt{\sum_{i=1}^{N} (\mathbf{E}_{Ref}(i) - \mathbf{E}_{APL}(i))^2 / N}}{\sum_{i=1}^{N} \mathbf{E}_{Ref}(i)/N}, \qquad (3)$$

where $\mathbf{E}_{Ref}(i)$ and $\mathbf{E}_{APL}(i)$ represent the irradiance values for surface patch i, computed using all the particles or APLs, respectively, and N is the number of surface patches. Our experiments use $N = 128 \times 256$, thus each patch subtends $4\pi/(128 \times 256)$ steradians. Using a relative error metric makes it easier to compare accuracy across different distances and luminaire models.

Figure 8 shows the relative RMSE plots for the Artichoke and Statler luminaires. The horizontal axis represents distance from the luminaire relative to the diameter of its bounding sphere, and the vertical axis is the relative RMSE as a percentage. They also show how illumination accuracy varies with the number APLs generated by our clustering algorithm. The results show that the single point far-field representation (which is equivalent to using one APL) can be a poor approximation even in the traditional far-field region (i.e., relative distance ≥ 5), especially for fixtures with strong illumination patterns such as the Artichoke. Our method is able to achieve much higher accuracy in both the near- and far-field regions, and the error generally decreases with the number of APLs. The illumination using 512 APLs has low relative error, even at fairly close distances. Using even more APLs did not improve accuracy enough to justify the increased costs, and thus we use 512 APLs for all our results unless otherwise specified. The error plots do not fully converge to zero due to residual noise from the finite particle data.

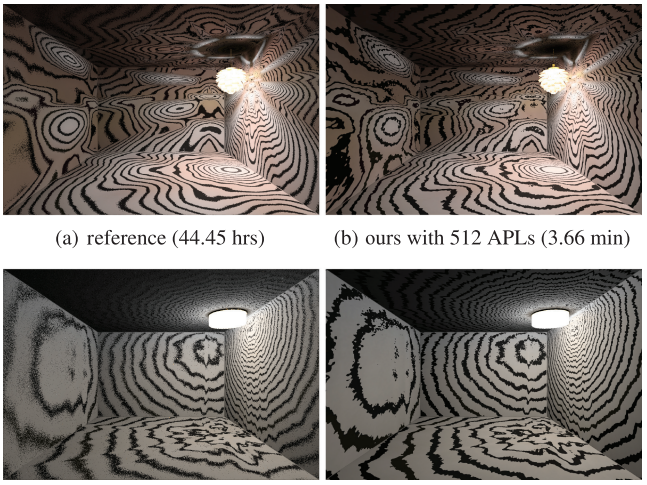
6.3 Qualitative Evaluation of Illumination

Ultimately, what we want is to create visually faithful images of scenes lit by interesting luminaires, thus we need a way to visually evaluate the quality of the illumination provided by our APLs



(a) Artichoke (b) Statler

Fig. 8. Relative RMSE at different distances using incremental numbers of APLs. As shown in the diagrams for two of the luminaires, we found that 512 APLs provide sufficiently accurate near-field illumination (shaded region on the left side of each plot). Note that using one APL is equivalent to the single point far-field representation.



(a) reference (44.45 hrs) (b) ours with 512 APLs (3.66 min)

(c) reference (190.96 hrs) (d) ours with 512 APLs (3.99 min)

Fig. 9. Comparison of the illumination isocontours for the Artichoke (top) and Statler (bottom) luminaires. Note that both the near and far-field distributions are close to the reference solution.

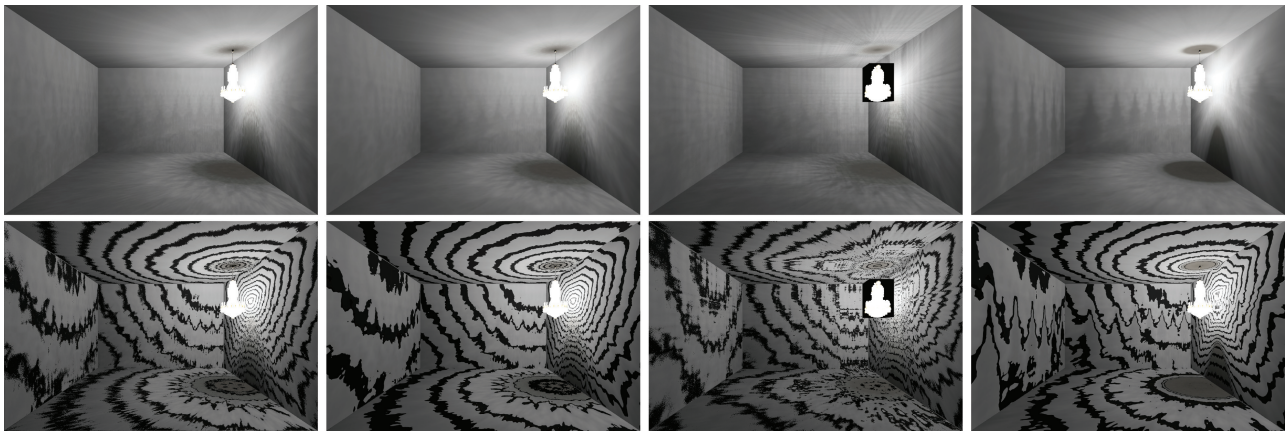
in comparison with the results by reference or previous methods. For this purpose we rendered each of our luminaires in an empty $3 \times 3 \times 2$m room. The luminaires are placed near one wall so that we can observe both near- and far-field effects. The excessively long time required to obtain fully converged reference images through unbiased rendering methods precludes from employing traditional per-pixel metrics such as the RMS error. Instead, motivated by Verbeck and Greenberg [1984], we enhance the result images with *isocontours* by modulating the luminance with a pulse-train function. This generates a series of black bands that are equally spaced in the logarithm of the luminance. These isocontours help to visualize the details of the illumination and judge its accuracy.

Figure 9 shows the isocontours for the reference BDPT renderings and for our results. The isocontours in our results are closely matched to the reference for both the near-field and far-field regions, which demonstrates that our method is radiometrically accurate. All isocontour images are 1248×842 and include illumination from the luminaire, but not indirect illumination from the rest of the scene, to better isolate illumination quality. To achieve this, our results are computed using the direct illumination from 512 APLs and the reference renderer was restricted to only allow paths with at most one nonluminaire vertex. Figures 9, 10, and 13 show the rendering time only. The precomputation times are presented separately in Table I because, typically, the precomputation can be done once per luminaire and then reused for all images and scenes containing that luminaire. However, our method would still be much faster than the reference even without such reuse.

Figure 10 compares illumination using our method and two prior methods to a reference solution for the V&A Chandelier. Single point far-field methods (Figure 10(d)) are widely used in industry, but produce the least accurate result here with illumination patterns that are clearly different from the reference in both near- and far-field regions. In this case, the single point far-field result was computed using our method but restricted to use only a single APL. The canned light-source method (Figure 10(c)) is more accurate, but still shows many illumination differences. It also shows visual artifacts caused by the fact that it stores the illumination on the luminaire bounding box as well as aliasing from undersampling, despite using eight times more data than our method. Our method using 512 APLs is the most accurate and closely matches the reference.

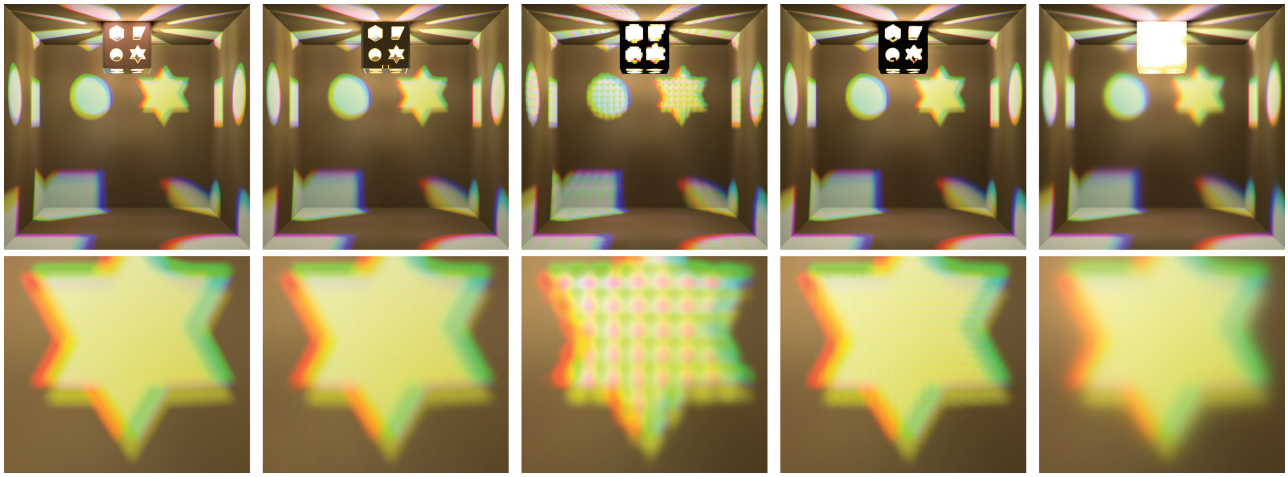
In Figure 11, we compare our method with the canned light-source method for a different kind of luminaire. The shape-sorter luminaire consists of a hollow box with various shapes cut out of it and containing four small spherical emitters, each with a different color. It is not based on a real luminaire, but intended to test the handling of luminaires that project strong directional patterns. Our result closely matches the reference rendering, with only a small amount of blurring due to the limited resolution of the radiance maps stored with our APLs. The canned light-source result (Figure 11(c)) shows strong aliasing artifacts even when using $8\times$ more data than our method. This is because the illumination is stored on a light field on the luminaire's bounding box which is far from the actual emitters. Using more samples in the light field (Figure 11(d)) reduces the aliasing issues, but greatly increases the storage requirements. Aliasing can be reduced by introducing a blur kernel in the light-field reconstruction, but this also blurs the features as illustrated in Figure 11(e). In general, we find our method produces higher-quality illumination results than light-field-based approaches while also requiring much less data and being easier to evaluate. For these comparisons, we tried to emphasize the luminaire illumination and exclude indirect illumination, however, the reference image still includes some indirect illumination from the scene onto the luminaire. This causes the luminaire box to appear brighter in the reference than in the other images and is an artifact of the method we used to generate the reference image.

Illumination accuracy is also important for correct material rendition. In Figure 12, we compare the effects of different luminaire representations when lighting an irregular metallic object. In this scene setting, the object is placed 30cm below our 48×96 inches Troffer luminaire. Our method (Figure 12(b)) closely matches the reference rendering (Figure 12(a)) even at such close range. Note the distinctive bright patterns cast by the Troffer's three ballasts. In comparison, the single point far-field method produces highlights in the wrong places (Figure 12(c)), whereas the commonly used uniform area light-source approximation yields a flat appearance which conveys a different material impression.



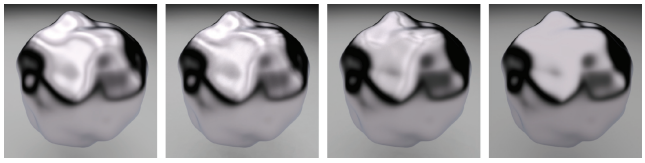
(a) reference (52.03 hrs) (b) ours with 512 APLs (3.86 min) (c) canned light-source with 4278 spatial samples (77.73 min) (d) Single Far-Field (1.99 min)

Fig. 10. Illumination rendering comparison for the V&A Chandelier luminaire. Our method (b) produces illumination and contours that closely match the reference image (a) while the canned light-source (c) single point far-field (d) methods are much less accurate with visually obvious illumination errors.



(a) reference (b) ours with 512 APLs (c) canned light-source with 4158 spatial samples (d) canned light-source with 32540 spatial samples (e) canned light-source with 32540 samples, low-pass filter

Fig. 11. Trade-offs of using precomputed data for representing illumination: four colored, small spherical sources inside the shape sorter cast well-defined patterns and exhibit color fringing around the edges. While this configuration is particularly unsuitable for our method, it can nonetheless represent such patterns with only a slight loss in detail due to the limited directional resolution on each APL. The canned light-source representation requires either an unpractical spatial sampling rate to minimize aliasing artifacts or strong low-pass filters which reduce the detail.



(a) reference (b) 512 APLs (c) single far field (d) area light

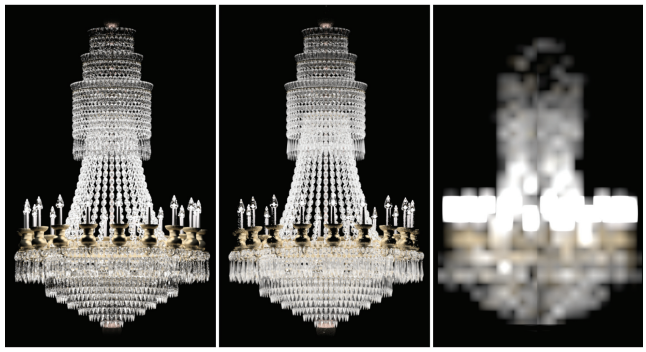
Fig. 12. Effects of illumination accuracy on material rendition. Note that the highlights on the reference image (a) are closely matched by our method (b). The single point far-field (c) and uniform area light-source (d) approximations convey a different material impression.

6.4 Qualitative Evaluation of Appearance

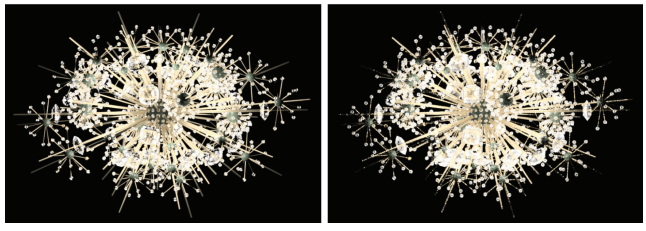
To evaluate the quality of our luminaire appearance rendering, Figure 13 shows renderings of three challenging luminaires comparing our method to reference images generated using BDPT. Due to the luminaires' computationally difficult optical properties, the reference images require a long time to converge and the Statler images show significant noise even after nearly 60 hours of computation. As shown in Figure 3, other potential reference global illumination algorithms exhibit similar noise problems. In contrast, our method produces visually good results with orders of magnitude faster render times, specifically 6 to 12 minutes for these images. There are small differences between our results and the reference

(a) reference (59.86 hrs)

(b) ours with 16 bounces (5.94 min)



(c) reference (10 hrs) (d) ours with 8 bounces (11.86 min) (e) canned light-source (1 min)



(f) reference (10.35 hrs) (g) ours with 6 bounces (8.11 min)

Fig. 13. Luminaire appearance comparison between simulated reference images and our results for Statler (top), V&A Chandelier (middle), and Met Sputnik (bottom) luminaires.

images, mostly due to the low-resolution nature of our radiance volume, but the overall appearance is perceptually very similar. Figure 13(e) shows a canned light-source result for comparison. While the render time is faster, the appearance quality is poor. Light-field approaches have trouble representing high-frequency features, and reconstructing an acceptable appearance would require using a much higher-resolution light field with impractically larger storage requirements. Also, because light fields use bounding surface proxies instead of the actual geometry, they do not provide any natural way to integrate scene-dependent aspects of appearance such as those due to transparency or reflections (e.g., causing the black region in Figure 1(b)).

As discussed in Section 5.4, our method combines depth-limited ray tracing to handle sparkles combined with a low-resolution radiance volume to fill in the perceptually important glow aspects of appearance. The ideal depth limit $\mathbf{n}_b$ is luminaire dependent, but usually less than 10. The exception is the Statler luminaire with its dense arrangement of glass shades, where we found $\mathbf{n}_b = 16$ is sufficient to generate high-quality sparkles. We tune the depth limit manually for each luminaire, but we expect this could be done as part of the preprocess and included with the luminaire data.

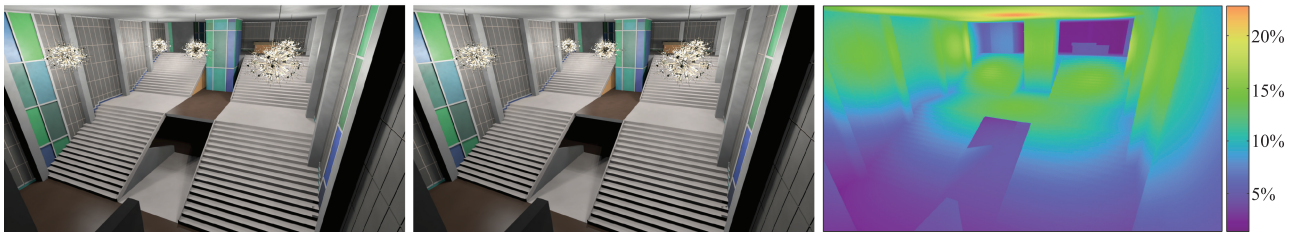
6.5 Discussions

Memory Consumption. The memory consumption of our method for different luminaires is roughly constant. For each APL, our method stores its directional radiance distribution map as a 256×512 Float16 RGB texture map. Thus loading all the 512 APLs into memory for rendering takes 384MB; standard block texture compression formats could further reduce the memory footprint to 64 MB [Werness and Daniell 2011]. For a radiance volume with 32^3 voxels, using 16 spherical harmonic coefficients (stored in Float32 format), the total memory cost is just 6 MB. As discussed in Section 5.5, the actual number of voxels we used depends on the aspect ratio of the luminaire's bounding box (Table I). The storage requirements are low enough to allow multiple luminaires to be easily loaded into memory simultaneously, allowing practitioners to simulate scenes with many complex luminaires not only in final renderings but also during the design phase.

Limitations and Future Work. Most of our results use 512 APLs for each luminaire for the illumination, regardless of the viewing distance or the distance between luminaire and scene points. While this accurately reproduces the illumination across all our tests, as discussed in Section 7, in many cases an APL-based light hierarchy with much a smaller number of APLs would have been sufficient.

When generating the luminaire appearance, our method currently manually chooses the maximum number of bounces $\mathbf{n}_b$ to separate sparkles and glow for each luminaire. An automated way to determine this parameter and a more adaptive way to separate sparkle and glow ray paths are left as future work. We assume the luminaire is much brighter than its surroundings. Our luminaire appearance includes short path reflection and transparency effects, but neglects longer path environmental effects, which are typically negligible compared to the luminaire's own illumination.

We rely on the distinction between appearance paths (i.e., "looking at the luminaire") and illumination paths (i.e., "looking elsewhere"), which can sometimes be ambiguous. The reflection of a luminaire in a mirror is clearly an appearance path, while a blurry reflection (or glossy highlight) is currently treated as an illumination path, which is not always appropriate. Better path classification heuristics could help. Adding a radius to the APLs or treating them more like *virtual spherical lights* (VSLs) [Hašan et al. 2009] could also improve the handling of glossy materials.



(a) baseline: all 512 APLs per luminaire (b) hierarchical APL selection, 9× speedup (c) percentage of baseline APLs used per pixel

Fig. 14. Dynamically choosing the APL set to query during rendering allows considerably better performance while preserving the image quality. On this atrium lit by five instances of the Sputnik luminaire, each pixel chooses among sets of 1, 8, 64, 256, or 512 APLs based on a simple distance-based heuristic.

(a) our method - V&A Chandelier (b) detail

Fig. 15. Rendering of the presbytery at the Basilica of San Vitale (Ravenna, Italy) solely lit by the V&A Chandelier luminaire using our method, including global illumination. The detail image (b) shows a cropped section of the full view using different tone mapping parameters to better appreciate the appearance of the luminaire.

Our method only accelerates multiple scattering within the luminaire and does not attempt to account for indirect illumination involving other parts of the scene. Including other types of indirect illumination requires combining our method with a more general global illumination algorithm. In Figure 11(b), the luminaire appears darker because we did not use such an algorithm in that example. Nevertheless, for our luminaires, this combination is still much faster than the alternative of applying the global illumination algorithm without our method. We assume a spatial decomposition in luminaire and the rest of the scene such that the scene geometry does not protrude into the interior of the luminaire. Our renderer does not currently simulate dispersion which can cause refractive rainbow effects, but our method could be extended to support this.

We use a simulation of the luminaire to compute its illumination that assumes the computer model of the luminaire is accurate. Measured data of luminaire components, such as a filament, can be used to improve the model, however, we currently do not have a way to incorporate measured data of a complete luminaire such as far-field data from a goniophotometer.

(a) our method - Troffer (b) our method - P.H. Artichoke

Fig. 16. A luminaire can have a substantial effect on the appearance of the interior design. Each image shows the same kitchen room rendered with a single luminaire as the only light source using our method, including global illumination. Notice how the shadows cast by the stools and the highlights on the left cabinet mirror the structure of each luminaire.

7. EXTENSIONS AND APPLICATIONS

The error measurements in Figure 8 demonstrate that it is not always necessary to use a large number of APLs to achieve good illumination accuracy. The "point light" nature of the APLs makes them ideally suited for scalable many-light algorithms [Dachsbacher et al. 2012]. It is also straightforward to generate an APL hierarchy to reduce rendering costs. As a proof of concept, we created a simple five-level hierarchy, each containing 1, 8, 64, 256, and 512 APLs, which corresponds to a specific level of the binary tree generated by the clustering step. To select a level during rendering we use a simple metric based on the relative distance from the gathering point to the luminaire's bounding sphere. Results in Figure 14 show that the illumination using this simple hierarchy and the heuristic function provide considerably better performance while preserving the image quality.

To improve the applicability of our method, we implemented the interfaces required for particle shooting and BDPT for our APLs (importance sampling and evaluation of the directional sampling density), allowing them to be used along the unmodified integrators from Mitsuba. Combined with our method for rendering the luminaire appearance, complex scenes with full global illumination can be faithfully rendered as demonstrated in Figure 15. Furthermore, our luminaire rendering method can be applied for interior lighting design. As shown in Figure 16, with our method, high-quality interior lighting can be quickly simulated and changed so as to identify the best lighting conditions to achieve a seamless combination of functionality and style.

8. CONCLUSIONS

In this article we presented a precomputation-based method for efficiently rendering both the illumination and appearance of complex luminaires. Such luminaires had previously been absent from renderings because they are impractical to simulate, even when using state-of-the-art algorithms. To deal with the complex illumination, in the precomputation stage we store the illumination leaving the luminaire by simulating the light transport within, and then apply a novel clustering strategy to transform the illumination into a set of APLs that is fast to evaluate. More importantly, these precomputed APLs accurately reproduce both near- and far-field illumination of the luminaire, which previous approaches failed to address. To handle the appearance of luminaires, we also construct a low-resolution radiance volume during the precomputation stage to record the low-frequency glow. At render time this radiance volume (together with a limited-bounce ray tracing strategy which fills in high-frequency, view-dependent sparkles), plausibly reproduces the characteristic appearance of complex luminaires such as chandeliers. Our technique renders accurate illumination and plausible appearance, even with extremely complicated luminaires at orders of magnitude faster speed than state-of-the-art global illumination algorithms. Our new method vastly reduces storage costs compared with existing precomputation-based approaches.

Since our precomputation data only needs to be generated once per luminaire model and may be reused afterward, generating the APLs set and the corresponding radiance volume could become part of the luminaire design pipeline such that manufacturers can provide these structures just like they do for single point far-field data today. This way, practitioners would be encouraged to incorporate more realistic, intricate light sources into their earlier design phase, a prospect which becomes feasible because of the way our method decouples lighting intricacies from geometric and material complexity.

ACKNOWLEDGMENTS

We thank Wenzel Jakob for providing early access to his Manifold Exploration implementation and his valuable suggestions and comments. Adrianne Ngam, Gretchen Craig, and Sonny Xu created the Met Sputnik, V&A Chandelier, and the San Vitale models. Andrés Lara provided the texturing for the San Vitale scene and Jorge Escamilla helped with additional modeling, both *pro bono*. The "Verano" kitchen model is used under permission from Autodesk.

REFERENCES

I. Ashdown. 1993. Near-field photometry: A new approach. *J. Illumin. Engin. Soc.* 22, 1, 163–180.

I. Ashdown. 2001. Thinking photometrically part ii. In *Proceedings of the LIGHTFAIR Pre-Conference Workshop.* 1–46.

I. Ashdown and R. Rykowski. 1998. Making near-field photometry practical. *J. Illumin. Engin. Soc. North Amer.* 27, 1, 67–79.

C. Dachsbacher, J. Krivanek, M. Hasan, A. Arbree, B. Walter, and J. Novak. 2012. Scalable realistic rendering with many-light methods. *Comput. Graph. Forum* 33, 1, 88–104.

T. Driemeyer. 2008. *Rendering with Mental Ray*, 3rd Ed. Springer.

I. Georgiev, J. Krivanek, T. Davidovic, and P. Slusallek. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.* 31, 6, 192:1–192:10.

M. Goesele, X. Granier, W. Heidrich, and H.-P. Seidel. 2003. Accurate light source acquisition and rendering. *ACM Trans. Graph.* 22, 3, 621–630.

S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. 1996. The lumigraph. In *Proceedings of the Annual ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96).* 43–54.

X. Granier, M. Goesele, W. Heidrich, and H.-P. Seidel. 2003. Interactive visualization of complex real-world light sources. In *Proceedings of the Pacific Graphics Conference (PG'03).* 59–66.

G. Greger, P. Shirley, P. M. Hubbard, and D. P. Greenberg. 1998. The irradiance volume. *IEEE Comput. Graph. Appl.* 18, 2, 32–43.

T. Hachisuka and H. W. Jensen. 2009. Stochastic progressive photon mapping. *ACM Trans. Graph.* 28, 5, 141:1–141:8.

T. Hachisuka, J. Pantaleoni, and H. W. Jensen. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph.* 31, 6, 191:1–191:10.

M. Hasan, J. Krivanek, B. Walter, and K. Bala. 2009. Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. Graph.* 28, 5, 143:1–143:6.

W. Heidrich, J. Kautz, P. Slusallek, and H.-P. Seidel. 1998. Canned lightsources. In *Proceedings of the EUROGRAPHICS Rendering Techniques Workshop (EGRW'98).* 293–300.

IESNA. 2002. *IES LM-63-2002, IESNA Standard File Format for the Electronic Transfer of Photometric Data and Related Information*. Illuminating Engineering Society of North America.

W. Jakob. 2010. Mitsuba renderer. www.mitsuba-renderer.org.

W. Jakob and S. Marschner. 2012. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.* 31, 4, 58:1–58:13.

A. S. Kaplanyan and C. Dachsbacher. 2013. Path space regularization for holistic and robust light transport. *Comput. Graph. Forum* 32, 2, 63–72.

S. Kniep, S. Haring, and M. Magnor. 2009. Efficient and accurate rendering of complex light sources. *Comput. Graph. Forum* 28, 4, 1073–1081.

E. P. Lafortune and Y. D. Willems. 1993. Bi-directional path tracing. In *Proceedings of the Computer Graphics Conference.* Vol. 93. 145–153.

M. Levoy and P. Hanrahan. 1996. Light field rendering. In *Proceedings of the Annual ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96).* 31–42.

A. Mas, I. Martin, and G. Patow. 2008. Compression and importance sampling of near-field light sources. *Comput. Graph. Forum* 27, 8, 2013–2027.

J. T. Moon and S. R. Marschner. 2006. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph.* 25, 3, 1067–1074.

J. T. Moon, B. Walter, and S. Marschner. 2008. Efficient multiple scattering in hair using spherical harmonics. *ACM Trans. Graph.* 27, 3, 31:1–31:7.

J. T. Moon, B. Walter, and S. R. Marschner. 2007. Rendering discrete random media using precomputed scattering solutions. In *Proceedings of the EUROGRAPHICS Symposium on Rendering Techniques (EGSR'07).* J. Kautz and S. Pattanaik, Eds., Eurographics Association, 231–242.

J. Muschaweck. 2011. What's in a ray set: Moving towards a unified ray set format. In *Illumination Optics II*, SPIE.

R. Ramamoorthi. 2009. Precomputation-based rendering. *Found. Trends Comput. Graph. Vis.* 3, 4, 281–369.

R. Ramamoorthi and P. Hanrahan. 2001. An efficient representation for irradiance environment maps. In *Proceedings of the Annual ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01).* ACM Press, New York, 497–500.

M. Rea, Ed. 2000. *The IESNA Lighting Handbook: Reference and Application*, 9th Ed. Illuminating Engineering Society of North America.

P. Shirley and K. Chiu. 1997. A low distortion map between disk and square. *J. Graph. Tools* 2, 3, 45–52.

P.-P. Sloan, J. Kautz, and J. Snyder. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.* 21, 527–536.

E. Veach and L. Guibas. 1995. Bidirectional estimators for light transport. In *Proceedings of the Annual ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'95).* 145–167.

C. Verbeck and D. Greenberg. 1984. A comprehensive light-source description for computer graphics. *IEEE Comput. Graph. Appl.* 4, 7, 66–75.

B. Walter, K. Bala, M. Kulkarni, and K. Pingali. 2008. Fast agglomerative clustering for rendering. In *Proceedings of the IEEE Symposium on Interactive Ray Tracing (IRT'08).* 81–86.

B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, and D. P. Greenberg. 2005. Lightcuts: A scalable approach to illumination. *ACM Trans. Graph.* 24, 3, 1098–1107.

B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the Eurographics Symposium on Rendering (EGSR'07).* 195–206.

G. J. Ward. 1994. The RADIANCE lighting simulation and rendering system. In *Proceedings of the Annual ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'94).* 459–472.

E. Werness and P. Daniell. 2011. ARB texture compression BPTC. http://www.opengl.org/registry/specs/ARB/texture_compression_bptc.txt.

Received June 2014; revised November 2014; accepted December 2014