COMPLEX LUMINAIRES: ILLUMINATION AND APPEARANCE RENDERING

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Edgar Velázquez Armendáriz

August 2014

COMPLEX LUMINAIRES:

ILLUMINATION AND APPEARANCE RENDERING

Edgar Velázquez Armendáriz, Ph.D.

Cornell University 2014

Simulating a complex luminaire 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 dissertation, 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.

BIOGRAPHICAL SKETCH

Edgar Velázquez Armendáriz was born in Mexico City the year IBM released the PC XT. He completed his primary and secondary education at the Instituto Juventud del Estado de México. At the end of elementary school he was a national winner at the Infantile Knowledge Olympics, being recognized by the President of Mexico, Dr. Ernesto Zedillo Ponce de León.

During his secondary education studies Edgar obtained prizes at several regional mathematics competitions and started to cultivate his passion for computer programming and playing the piano. Upon graduating as the valedictorian of his high school class he received the ITESM-CEM Excellency Scholarship for undergraduate studies at the Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de México, majoring in Computer Systems Engineering. His first contact with research was via his participation as a member of the "TecRams" team of the Robocup four-legged league.

In his senior year of college Edgar was admitted to the Cornell's International Undergraduate Engineering Research Internship/Mexico, organized by Prof. Francisco Valero-Cuevas, where he worked at the Program of Computer Graphics under the tutelage of Prof. Kavita Bala. The results of this internship were published in a peer-reviewed conference [52]. He returned to Mexico to conclude his undergraduate studies, receiving in 2006 his B.S. degree with Highest Honors, and the Testimony of Outstanding Performance from CENEVAL (National Center of Evaluation for Higher Education) after obtaining the highest score at the national level in the standardized Computer Science examination.

Edgar started his graduate studies at Cornell University in 2006 as a Cornell-CONACyT (National Council of Science and Technology) fellow, collaborating

once again in the first instance with Prof. Kavita Bala. During the second part of his Ph.D. program we was advised by Prof. Donald P. Greenberg, founder and director of the Program of Computer Graphics at Cornell. As a graduate student he collaborated in several computer graphics research projects and co-authored papers in peer-reviewed journals [19, 53]. He received his M.S. degree from Cornell University in 2014. His research interests include predictive rendering, real-time computer graphics, high performance computing, applications of massively parallel processors, and topics at the intersection of science and art.

When not in front of a computer, Edgar may be found playing the piano, trying to improve his Italian to further enjoy opera, practicing salsa and ballroom dancing, or reading about art and universal history.

Para mi familia, gracias por su cariño y apoyo incondicionales.

ACKNOWLEDGEMENTS

I am greatly indebted to my advisor, Don Greenberg, who believed in me and offered me the opportunity to continue the journey through graduate school at a crossroads where I was certain my life would be taking a different path. A shared passion for Vermeer, Canaletto and opera led into a research topic at the intersection of science and art. Don has been a mentor, guide and counselor beyond the strictly academic and an invaluable support *in loco parentis*.

I thank the members of my committee, Charlie Van Loan and Hakim Weatherspoon, for their inspiring classes and continued guidance.

The tenacity of Prof. Francisco Valero-Cuevas in creating research opportunities for Mexican students gave me the opportunity of a lifetime to do research at Cornell during my senior year at college. I am grateful to Prof. Kavita Bala for welcoming me as part of her group and teaching me not only the fundamentals of computer graphics research but also valuable professional-demeanor lessons.

My colleagues and coauthors Zhao Dong, Miloš Hašan, and Bruce Walter always motivated and pushed me to go the extra mile. I have had the privilege to learn and work along such world-class leaders in the field. Along with fellow PCG alumni Adam Arbree, Ganesh Ramanarayanan, Jon Moon, Jonathan Kaldor, Changxi Zheng, Jaroslav Křivánek, Wenzel Jakob and Shuang Zhao they made working through the SIGGRAPH deadline marathons an always memorable experience. Todd Harvey, Lars Schumann, Matthew Low and Peter Kung brought a fresh perspective during those long nights at the office.

I am deeply thankful to Peggy Andersen, Linda Stephenson and Hurf Sheldon at the PCG, and Becky Stewart at CS, for all their help ever since I first arrived at Cornell as a visiting undergrad.

I thank my friends Anton Morozov, Bruno Abrahao, Mauricio Lozoya, Isauro Buendía, Tzácil Bravo, Lourdes del Carmen González Huesca, Lourdes Guzmán Castillo and Carolina Hernández for their encouragement; and Betty Spero for being like an aunt to me and making my time in Ithaca much more enjoyable. Very special thanks to Ellan Spero, for always emboldening me to do the right thing, and to Bistra Dilkina and Carlos Ortega Otero, with whom I have shared both great joys and sorrows; Carlos has always been side-by-side with me ever since elementary school. Their unconditional support meant a lot to me and helped me understand the true meaning of friendship.

I appreciate the backing from my family and family friends, particularly the cherishing from my godparents Juanita Jiménez and Jorge Hernández whose example I aspire to follow.

Last but not least I would like to thank my parents Rafaela and Edgar, and my sister Erika for their unconditional, unconstrained love and support. They were always with me especially in the moments of direst need, withstanding harsh winters and *ex abrupto* travels across the Atlantic when I needed them the most. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Why are depictions of luminaires mostly absent from computer graphics renderings? Luminaires are an important part of most environments, they provide light for extending working hours and shape the illumination to suit our needs; artificial lighting has become so essential to our standard of life that its presence may go unnoticed while at the same time mankind expends a considerable portion of its global energy output just for illumination [50]. Luminaires also often serve as decorative elements in their own right: renowned artists like Dale Chihuly and Antoni Gaudí have created luminaires as part of their opus. These aesthetic qualities encompass both the illumination patterns it projects onto the surrounding scene as well as the appearance of the luminaire itself.

A huge variety of luminaires are commercially available ranging from simple spot lights to elaborate chandeliers such as those depicted in Figure 1.1. However simulating real luminaires is frequently prohibitively expensive and in computer rendering practitioners often use highly simplified representations instead. In this dissertation 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., incandescent 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 makes them optically complex and expensive to simulate.

Metropolitan Opera,
New York City

State Academic Mariinsky Theatre,
Saint Petesburg, Russia

French Restaurant at the Municipal House
(*Obecní dům*), Prague, Czech Republic

Leonardo Room at the State Hermitage
Museum, Saint Petersburg, Russia

Figure 1.1: Examples of complex luminaires in the real world

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 in Chapter 3. For example, Larson and Shakespeare, in their book describing how to use the Radiance system [27], warn that it is necessary to use simplifications instead of the actual luminaire components for successful renderings (note that as of this writing the Radiance system is still the *de facto* reference tool used by lighting designers.)

> It is not always possible for a backwards ray-tracing algorithm [the particular global illumination method used by Radiance] to follow all of the bounces of light far enough and reliably enough to reach a light source that is hidden behind louvers, diffracting lenses, convex or concave mirror, and so on… A suspended direct/indirect luminaire is one type that requires impostor geometry for correct appearance. [27, Chapter 5.2 "Luminaire modeling", p. 317.]

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 which 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. However these only work well when the luminaire or its aperture is small. We present a new precomputation approach that is designed to work for more complex luminaires, ones whose aperture may be large or ill-defined (e.g., Figure 1.2.)

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 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 either 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 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.

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

(b) A canned lightsource 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.

Figure 1.2: 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 lightsource 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 using complex luminaires such as this one.

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 (Chapter 4.) 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 (Chapter 5.) Our methods provide higher accuracy at lower data sizes compared to previous precomputation approaches, allowing us to render with luminaires that were previously infeasible.

To summarize, the main contributions of this dissertation are as follows:

- Anisotropic point lights (APLs): a precomputed representation of the illumination from a luminaire accurate in both the near- and far-field, irrespective of the luminaire's geometric arrangement.
- An efficient technique for computing the appearance of a luminaire, based on depth-limited ray tracing and a precomputed radiance volume.
- These techniques for the illumination and the appearance of a luminaire are orthogonal to existing global illumination algorithms; we demonstrate that it is straightforward to integrate them with existing rendering systems.
- By decoupling the complexity of light transport within a luminaire from the rest of the scene our method empowers users to incorporate far greater lighting complexity in their designs than was previously possible.

The reminder of this dissertation is organized as follows. Chapter 2 surveys the existing literature related to luminaire rendering in the fields of computer graphics and illumination engineering. Chapter 3 uses a case study to emphasize the infeasibility of rendering scenes lit by complex luminaires using current state-of-the-art methods, and introduces our insights into a luminaire's illumination and appearance which are the basis of our method. Chapter 4 presents our proposed algorithm to render the radiometrically accurate illumination, and the perceptually plausible appearance of luminaires. The sections in this chapter describe how to generate our two precomputation structures, APLs for the illumination and the low-frequency radiance volume for the appearance, and then reuse them for rendering the same luminaire across different environments and configurations. Chapter 5 shows our results evaluating the accuracy and performance of our method using multiple luminaire models of varying degrees of complexity. Through qualitative and quantitative analysis we demonstrate that our method provides correct results while being two or three orders of magnitude faster than reference methods. Chapter 6 presents extensions to our method which further improve its performance and allow it to be used with existing, unmodified rendering systems. This allows to render complex scenes with full global illumination being lit by luminaires far more intricate than was possible before. We finalize with our conclusions in Chapter 7.

CHAPTER 2

RELATED WORK

Initial work on measuring and representing luminaires comes from the illumination engineering community. Their work permeated the computer graphics field at its beginnings. Researchers were aware of the limitations of these early luminaire representation models, which accounted only for the directional variation of the light (Section 2.1.) Unfortunately these representations were still the standard industry practice 30 years after their introduction in the 1980s. Early on in the computer graphics community there was some work on better representations of luminaires based on the plenoptic function [11] (Section 2.2) but most of the research was focused on solving the ever more complex global light transport using the existing luminaires models (sections 2.3 and 2.4.) While the latest rendering techniques aspire to have a unified approach to light transport, including the energy coming from luminaires, handling the pathologically difficult light paths emanating from complex light fixtures remains an unsolved problem.

2.1 Far-Field Approximation

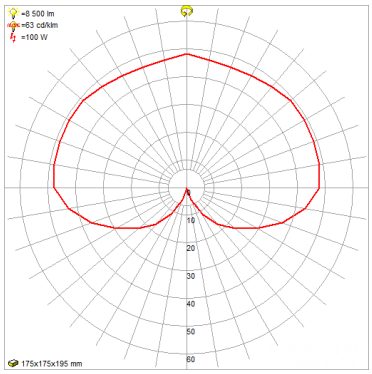
Initial models representing luminaires were based in lighting engineering and used a goniometer to capture the characteristics of the luminaires. These models, such as IES LM-63 [44] and EULUMDAT [2], provide the simplified far-field[1] luminance intensity distribution of the luminaire centered at a point in space.

[1]Far-field is often defined as the region farther than five times the length of the longest dimension of a luminaire [43].

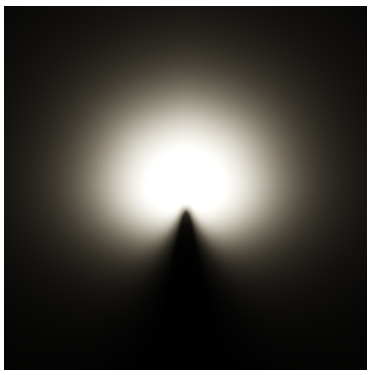
These formats consist of plain-text data which describes, among other features, the shape of the luminous aperture (e.g., rectangular, ellipsoidal, circular), the type of emitter used in the measurements (e.g., an incandescent light bulb or a fluorescent ballast) and the tabulated directional intensity. Figure 2.1 contains a basic example of such description as an IES LM-63 file. It is common to sample the polar angle more densely than the azimuthal angle; polar sampling rates of one degree are common place for high quality measurements. As of this writing, the far-field model is still the lighting industry standard, with data provided directly by the luminaires manufacturers and supported by the rendering and lighting simulation software packages such as Radiance [58] and Mental Ray [8]. The standard practice when rendering such photometric sources is to attach the emission distribution to a point or a surface in order to obtain soft shadows. Refinements on the point-centered discrete distribution model by Verbeck and Greenberg [54] have taken into account some of the luminaire geometry to generate more plausible soft shadows and better model its spectral distribution.

Although widely used, the far field models are not sufficiently accurate to reproduce realistic lighting simulations when the illuminated surfaces are noticeably close to the luminaires [2, 39]. For example, many if not most office light fixtures are closer to the work surfaces than its far-field region. Furthermore the discrepancies of using far-field models in such scenarios are relevant for energy simulations and light quality evaluations including luminance levels and glare.

Photograph



Goniogram
(photometric candela values, φ = 0°)



IES file rendering

Luminaire description

Measurement description

Candela values at each horizontal angle

```
IESNA:LM-63-1995
[TEST] BE4124
[DATE] 7-MAY-97
[MANUFAC] BEGA
[LUMCAT] 8522MH
[LUMINAIRE] BOLLARD
[LAMP] (1) 100W ED-17 MH
TILT=NONE
 1   8500   8.5   20   19   1   2   .175   .175   .195
 1   1   100
 0   5   10   20   30   40   50   60   70   80   90   100   110   120   130   140   150   160   170   180
 0   5   10   15   20   25   30   35   40   45   50    55    60    65    70    75    80    85    90
  .02735    .2115     .3002    4.674   11.07   17.09   22.78   29.02   36.61   43.29   47.79   φ = 0°
                      48.49    48.42    48.7    48.75   47.48   46.4    46.15   46.66   48.3
  .02735    .2186     .3006    4.647   11.02   17.05   22.73   29      36.7    43.47   48.05   φ = 5°
                      48.97    48.82    48.78   48.76   47.44   46.41   46.14   46.75   48.3
  .02735    .2182     .2999    4.505   10.82   16.8    22.72   29.24   37.12   44.04   48.88   φ = 10°
                      50.24    49.92    49.25   48.75   47.27   46.27   45.97   46.66   48.3
  .02735    .2181     .2999    4.308   10.49   16.8    23.11   29.95   38.17   45.33   50.79   φ = 15°
                      52.56    51.97    50.44   49.12   47.2    46.11   45.89   46.64   48.3
  .02735    .2216     .3103    4.015   10.24   17.06   24.04   31.17   39.58   47.03   53.27   φ = 20°
                      55.27    54.51    52.28   49.93   47.29   46.08   45.74   46.71   48.3
  .02735    .218      .3099    3.637   10.15   17.68   25.11   32.37   40.82   48.66   55.65   φ = 25°
                      57.73    56.87    54.21   50.91   47.55   46.01   45.53   46.6    48.3
  .02735    .2215     .3135    3.233   10.32   18.41   26.13   33.44   41.78   50.3    57.73   φ = 30°
                      59.9     58.95    56.06   51.94   48.1    46.25   45.56   46.69   48.3
  .02735    .2181     .3101    2.9     10.59   19.02   26.98   34.26   42.56   51.71   59.4    φ = 35°
                      61.58    60.63    57.51   52.94   48.48   46.5    45.45   46.64   48.3
  .02735    .2181     .3136    2.715   10.8    19.45   27.52   34.84   43.12   52.76   60.56   φ = 40°
                      62.75    61.77    58.55   53.73   48.89   46.73   45.43   46.68   48.3
  .02735    .2181     .3136    2.657   10.87   19.59   27.63   34.95   43.22   53.03   60.83   φ = 45°
                      63.04    62       58.82   53.95   48.86   46.83   45.43   46.75   48.3
  .02735    .2181     .3136    2.715   10.8    19.45   27.52   34.84   43.12   52.76   60.56   φ = 50°
                      62.75    61.77    58.55   53.73   48.89   46.73   45.43   46.68   48.3
  .02735    .2181     .3101    2.9     10.59   19.02   26.98   34.26   42.56   51.71   59.4    φ = 55°
                      61.58    60.63    57.51   52.94   48.48   46.5    45.45   46.64   48.3
  .02735    .2215     .3135    3.233   10.32   18.41   26.13   33.44   41.78   50.3    57.73   φ = 60°
                      59.9     58.95    56.06   51.94   48.1    46.25   45.56   46.69   48.3
  .02735    .218      .3099    3.637   10.15   17.68   25.11   32.37   40.82   48.66   55.65   φ = 65°
                      57.73    56.87    54.21   50.91   47.55   46.01   45.53   46.6    48.3
  .02735    .2216     .3103    4.015   10.24   17.06   24.04   31.17   39.58   47.03   53.27   φ = 70°
                      55.27    54.51    52.28   49.93   47.29   46.08   45.74   46.71   48.3
  .02735    .2181     .2999    4.308   10.49   16.8    23.11   29.95   38.17   45.33   50.79   φ = 75°
                      52.56    51.97    50.44   49.12   47.2    46.11   45.89   46.64   48.3
  .02735    .2182     .2999    4.505   10.82   16.8    22.72   29.24   37.12   44.04   48.88   φ = 80°
                      50.24    49.92    49.25   48.75   47.27   46.27   45.97   46.66   48.3
  .02735    .2186     .3006    4.647   11.02   17.05   22.73   29      36.7    43.47   48.05   φ = 85°
                      48.97    48.82    48.78   48.76   47.44   46.41   46.14   46.75   48.3
  .02735    .2115     .3002    4.674   11.07   17.09   22.78   29.02   36.61   43.29   47.79   φ = 90°
                      48.49    48.42    48.7    48.75   47.48   46.4    46.15   46.66   48.3
```

Figure 2.1: Example of an IES LM-63 file [44]. This file describes the far-field distribution of a Bega unshielded bollard luminaire using 20×19 discrete measurements; the distribution is symmetric in each quadrant of the horizontal plane. This measurement uses photometric candela values, losing the wavelength-dependent information.

2.2 Near-field Acquisition and Rendering

A more general way of describing the radiance emitted by a luminaire, instead of points and areas with homogeneous emission, is through its *light field* as proposed by Gershun [11]. The light field is a multidimensional function which describes the wavelength-dependent radiance at any point in space for a given direction. For a luminaire contained in a medium with negligible scattering, such as air over moderate distances, it is possible to accurately characterize its emission properties by describing its light field at a virtual surface enclosing the luminaire [39]. This approach decouples the emission representation from the actual luminaire geometry and provides a framework for more detailed near-field lighting capturing the full light field of a luminaire.

Ashdown proposed a method to measure this light field for existing luminaires using sensors close to the luminaire [1]. Initial approaches to capture the luminaire's light field used direct measurements obtained with a special goniophotometer with low resolution cameras to limit the amount of data. As with the current far-field data, the captured near-field information would be generated by the manufacturer and provided to the users for their use in simulation software. It is also possible to capture a coarser representation of the light field through an environment using standard cameras and image processing techniques [36].

To represent the light field Levoy and Hanrahan [29] concurrently with Gortler et al. [14] developed a data structure which they called *light slab* and *lumigraph* respectively. A light slab represents all straight-line light paths between a pair of points, each of them located in two different planes. To represent a larger space

with better accuracy it is possible to use multiple light slabs; the full light slab for an object is defined around the six faces of its bounding cube. The benefit of this representation is that a single perspective image (i.e. a rendered view or a photograph) constitutes a 2D slice of the light slab. Using this formulation an entire light slab can be stored as an image consisting of a set of tiles, each of which is a 2D slice of the full 4D dataset (Figure 2.2.) Fortunately the approach is amenable to compression techniques. Conceptually one of the planes in this structure, (u, v), represents the spatial resolution (sampling of the object) and other the directional resolution at each spatial sample, (s, t). The (s, t) plane may be placed at infinity to decouple the spatial and directional sampling rates. Because the target application of these papers is to synthesize new views from existing data, the authors note that directional resolution is more important for the perceived quality than the spatial resolution. Thus the directional plane (s, t) has approximately an order of magnitude more samples than the spatial plane (u, v) [29, 14, 20].

To overcome the aliasing introduced by sparse measurements, Goesele et al. [12] proposed projecting the light field into a finite basis before its capture using simple filters to improve reconstruction. This technique also permits a fast visualization of the luminaire's direct lighting using programmable graphics hardware [15], but is limited by the huge amounts of data that are needed for higher frequency details. For these reasons light field rendering methods typically use a combination of compression methods such as vector quantization, non-negative matrix factorization, PCA-analysis and standard image compression techniques [4, 5, 6, 29].

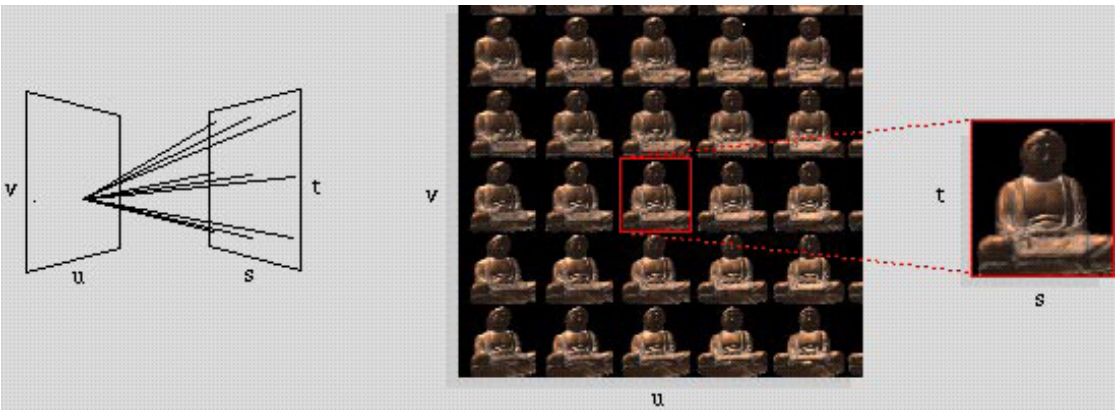


Figure 2.2: Light slab (lumigraph) representation of a light field from Levoy and Hanrahan [29]. Each image of Buddha in the array corresponds to the rays passing through all points in the directional plane (s, t) and arriving at a single location in the spatial plane (u, v). Thus each image is a 2D slice of the 4D light field.

Relying on global illumination (GI) techniques, the light fields for virtual luminaires can be simulated and stored. The canned lightsource method by Heidrich et al. [20] (Figure 2.3a) proposed computing a luminaire's 4D light field in a preprocess which can then be reused at render time to more efficiently compute the luminaire's illumination. The results can then be stored in a light slab structure [14, 29]. However discretizing and storing the 4D light field becomes very data-intensive and expensive if the luminaire covers a large area and contains high frequency details. As will be described later, other alternate representations can scale much better for more complex luminaires.

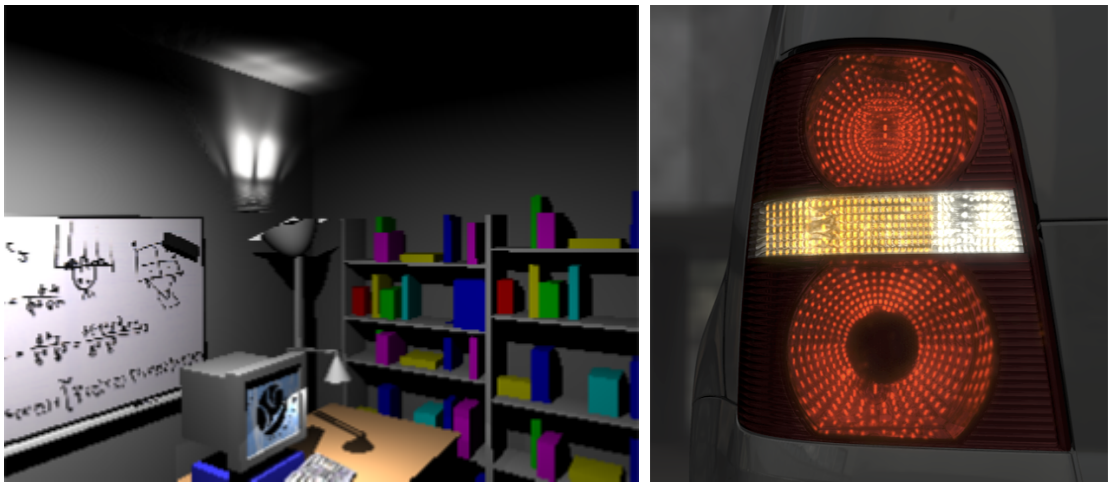
Another way to represent the results of the near-field acquisition is using a rayset [3, 37]. A rayset is a set of exiting rays (origin and direction) all with equal energy and without spectral distribution, leaving the luminaire and captured on a virtual bounding surface. To compress the rayset model of a luminaire, Mas et al. [31] proposed transforming the rayset into a set of clusters, each with a discrete hemispherical directional distribution. This data representation is suitable

for designing reflectors which will yield a determined far-field distribution [32]. This technique requires the rayset's origin points to lay on a convex hull which surrounds the luminaire tightly; such a surface is often ill-defined for irregular luminaires such as glass chandeliers.

To visualize the appearance of a complex luminaire, Kniep et al. [25] (Figure 2.3b) proposed an approach based on directional photon mapping [33]. They use photon mapping but only store photons when they intersect a bounding surface for the luminaire. At render time, density estimation was used to accurately reproduce the appearance of complex car tail-light assemblies. The approach is quite data intensive, requiring the storage of gigabytes worth of photon data for each luminaire, and thus does not scale well to large and complex luminaires. Furthermore this technique does not provide an explicit method to calculate the light source's illumination into the environment, requiring instead a very large number of expensive density estimation queries.

2.3 Global illumination methods

General global illumination methods, such as bidirectional path tracing [26, 51] and stochastic progressive photon mapping [17], can be used to simulate the luminaire without any precomputation. In this case, the luminaire geometry is simply treated as part of the scene geometry. However complex luminaires are often composed of challenging materials and elements such as glass, crystals, diffusers, reflectors and refractors. The light paths inside a complex luminaire are often the most difficult to handle in a scene, leading to slow convergence



(a) Heidrich et al. [20] (b) Kniep et al. [25]

Figure 2.3: Examples of previous work on luminaire rendering: (a) precomputed luminaire emission using the light slab data structure and (b) direct visualization through directional photon mapping.

for general global illumination methods. Several recent methods have been proposed to improve general handling of difficult caustic-like paths. Several recent contributions are described next.

The Manifold Exploration [22] algorithm, by Jakob and Marschner, greatly improves the exploration of connected caustic components (Figure 2.4). This technique introduces a strategy for modifying light paths by perturbing the end points (typically corresponding to the camera and a light emitter) while preserving a valid configuration. Two methods concurrently developed by Georgiev et al. [10] (Figure 2.5) and Hachisuka et al. [18] combine the strengths of photon mapping and bidirectional path tracing to improve handling caustic components. Another recent approach by Kaplanyan and Dachsbacher [23] selectively modifies materials with perfectly specular components to simplify the handling of difficult paths, but would not apply to scenes which already include glossy surfaces such as those present in light fixtures.

Figure 2.4: Example results from Jakob and Marschner [22] demonstrating how manifold exploration handles specular and near-specular transport on a scene with glossy, reflective and refractive materials. The chandelier on the right image is used to light the tableware on the left.

While these methods are significant improvements to the state of general global illumination, none of them can yet efficiently handle the complex combination of many disjoint, overlapping caustic paths such as those arising in light fixtures with several layers of intricate glass refractors and numerous emitters. These characteristics are commonly the case with ornamental chandeliers. Under these circumstances a slight perturbation of an existing light path results in a radically different path configuration; the manifold exploration method cannot switch between disjoint classes of paths. The vertex-merging techniques [10, 18] would degenerate to path tracing in the presence of paths on which most vertices correspond to highly glossy materials.

Another paradigm for solving the global light transport problem is through many-light methods [7], which discretize the light paths into a large collection of virtual point light sources (VPLs.) Henceforth it is possible to use scalable algorithms such as Lightcuts [55], in which the shading cost is strongly sublinear with respect to the number of light sources. An optimization which is standard practice with many-light methods is to generate separate sets of VPLs for direct and indirect illumination. In the case of the direct VPLs the renderers distribute

(a) Reference (b) Vertex connection merging

Figure 2.5: Example result from Georgiev et al. [10] showing the fast convergence of the vertex connection merging, achieved through efficiently combining the contributions of bidirectional path tracing and progressive photon mapping.

them over the surface of the emitters. However for complex luminaires the emitters are hidden from most the scene by the luminaire's multiple reflectors, refractors and diffusers; upon rendering such direct VPLs will not contribute energy to the vast majority of the scene.

2.4 Precomputation-based Rendering

Precomputation-based rendering (PBR) [41] methods have been extensively investigated in graphics. Since precomputation is involved, the PBR methods usually deal with the static scenes or scenes with rigid transformation only. In practice, complex luminaires are almost always static, hence the precomputation strategy fits the scene rendering very well. PBR methods often use functions defined over a spherical domain, such as incident radiance. To avoid the runtime and storage costs of detailed representations such as bitmaps, PBR methods

commonly use a simplified version of the original function by projecting it into a spherical basis function.

Spherical Harmonics (SH) is a representative basis function that has been widely adopted in PBR [42, 47] (Figure 2.6a.) The SH basis can smoothly reconstruct low-frequency signals very well and can be freely rotated; however it cannot efficiently represent high frequency signals. Other techniques better suitable for higher frequency signals but with higher complexity include wavelets [38] and spherical radial basis functions (SRBF) [49]. The 2D wavelet basis applied to a planar projection of the function can represent all-frequency effects with a small number of coefficients, but its rotation is non-trivial and can introduce temporal flickering due to its nonlinear reconstruction. Wavelet bases defined on the spherical domain [28] avoid projection artifacts but are prohibitively expensive to use. The SRBF can reconstruct all-frequency effects as well and is easy to rotate, but its precomputation cost is much higher than that of other functions [30].

PBR techniques have been successfully used to accelerate the calculation of challenging effects due to multiple scattering such as indirect lighting, subsurface scattering and volumetric rendering. The irradiance volume [16] method approximates the global light transport within an enclosed region by using a pre-computed, sparsely sampled representation of the irradiance (radiant power on a surface, per unit surface area) as a function of the position within the volume delimited by the space and a given direction. This representation provides an $O(1)$ approximation of the global illumination, reconstructed with trilinear interpolation, suitable for interactive applications.

(a) Sloan et al. [47] (b) Moon et al. [35] (c) Moon et al. [34]

Figure 2.6: Examples of previous work on precomputation-based rendering: (a) precomputed radiance transfer, (b) multiple scattering in hair and (c) discrete homogeneous isotropic random media.

The method places its irradiance samples at the vertices of a bi-level uniform grid, allowing for additional detail in the non-empty regions of the volume. At each sampling location the directional irradiance is stored as a piecewise function defined on a bitmap which uses a solid-angle preserving projection. While the irradiance at each orientation may be calculated using any global illumination method, the authors chose a piecewise approximation of the radiance function at the given location. This approach permits the reuse of the same set of radiance samples for all irradiance calculations at the same spatial location.

At runtime irradiance queries are computed by trilinear interpolation of the closest samples in the volume. While the method cannot handle specular surfaces *per se*, the authors note that it would be possible to support some moderate gloss by storing additional basis functions at each sample location, such as cosine lobes as used in the Phong shading model.

The method works well for volumes with smoothly changing irradiance features. Because of the sparse sampling it cannot be used efficiently to repre-

sent directly neither highly detailed radiance functions nor directly visible contours. Since most light fixtures, and certainly decorative luminaires, have intricate physical geometries which are easily perceived, it is not possible to depict them satisfactorily with an irradiance volume alone.

Moon et al. [35] (Figure 2.6b) have explored the radiance volume concept in the context of rendering multiple scattering in hair. Their method estimates the average radiance in each voxel of a coarse grid enclosing the hair fibers. Each voxel stores only a SH projection of the radiance estimate.

Under specific constraints it is possible to construct analytical models based on precomputed data. Moon et al. [34] (Figure 2.6c) present a method to efficiently render the multiple scattering in "discrete homogeneous isotropic random media", where the assumptions of traditional volumetric methods are no longer valid. This method introduces a statistical description of the scattering within a region of the discrete random media called the *shell transport function*. It describes the probability of a light path to leave at a certain direction and position on the bounding sphere, given its initial direction. Together with an appropriate reformulation of the rendering equation this approach allows paths to traverse the media in large steps while computing direct illumination at each interaction. The shell transport functions for different radii are generated during a preprocessing stage through path tracing and then compressed using non-negative matrix factorization which allows efficient importance sampling.

The authors note that since the low order scattering near the surface of the medium is not a smooth function, and the shell transport rendering equation is valid only at a certain depth from the media surface, a final image consists of two

passes. The first pass computes low order scattering using standard path tracing and a high sampling rate. The second pass generates the high order scattering using the shell function. This technique achieves converged images orders of magnitude faster than the brute force path tracing method. However this algorithm cannot be used for rendering the direct appearance of a luminaire, since its geometry and organization do not meet the definition of discrete random media.

In a luminaire the specific geometric arrangement of its parts is far more noticeable and cannot be represented by statistical methods. Its elements' spatial placement and orientation is almost always non-random and exhibit a high degree of anisotropy and directional dependence. Furthermore most decorative luminaires consist of multiple materials, each of which is far more complicated than either ideally diffuse reflection or perfectly specular reflection and transmission.

We have presented related work from the computer graphics and illumination engineering literature which describe a wide range of ways to synthesize, in principle, scenes lit by all types of luminaires. Nonetheless even the state-of-the-art techniques fail to simultaneously address the illumination and the appearance of complex luminaires. In the following chapters we will present our proposed simulation algorithm which addresses both the appearance of luminaires and both their near-field and far-field illumination. The luminaires can range from humble troffer fixtures to elaborate glass chandeliers. In all cases our methods vastly improve the computation times of existing approaches, and simultaneously provide more accurate results than previous precomputation-based solutions.

CHAPTER 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 3.1a, top), that contain twelve sources within 117 glass shades. As a test, we built a computer model based on this chandelier (Figure 3.1b, bottom) 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 were practically feasible solutions. 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 3.1b) was also rendered with several algorithms and cropped image results shown in Figures 3.2 and 3.3 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 [51], manifold exploration [22] and stochastic progressive photon mapping [17], were set to use equal rendering time as our method. The reference is rendered with bidirectional path tracing with a much longer render time. As a representative of light field approaches, we also show canned lightsources [20] set to use eight

(a) Real scene photograph: such complex luminaires cast intricate patterns on their immediate surroundings and are very aesthetically pleasing themselves.

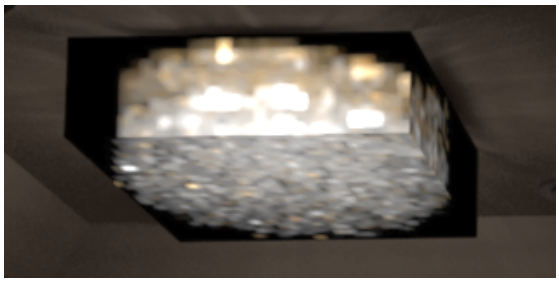
(b) Our method efficiently and accurately renders both near- and far-field illumination from such complex luminaires at significantly faster speeds than previous methods, achieving comparable effects to those present on the real scene.

Figure 3.1: Statler Hotel scene lit by four chandeliers: photograph and 3D model.

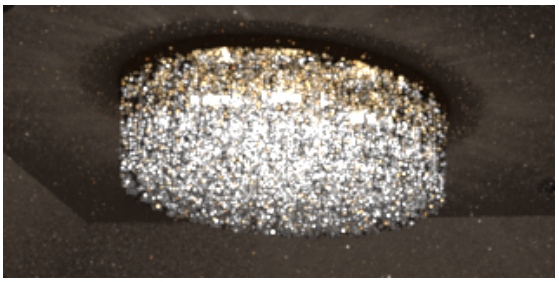
(a) Reference (150× time)

(b) Our method

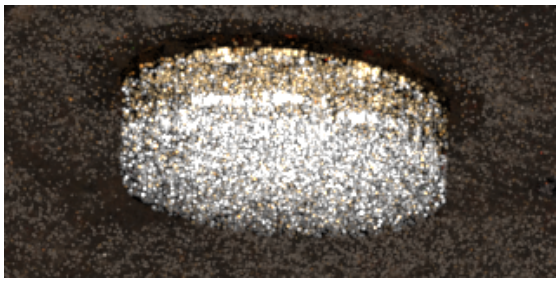
(c) Real scene photograph



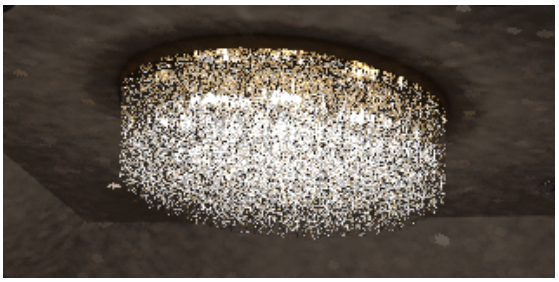
(d) Canned lightsource (8× data)



(e) Bidirectional Path Tracing (equal time)

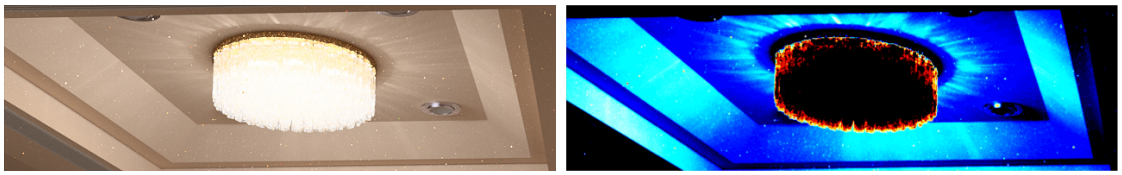


(f) Manifold Exploration (ERPT) (equal time)

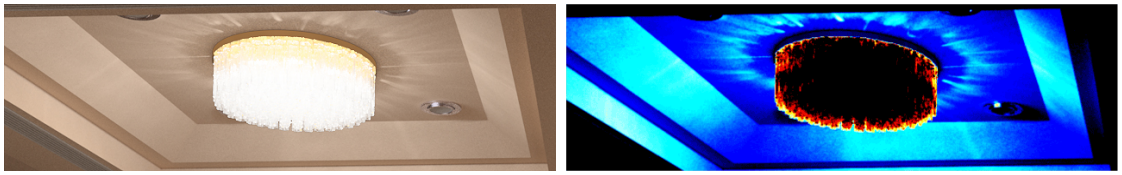


(g) Stochastic Progressive Photon Mapping (equal time)

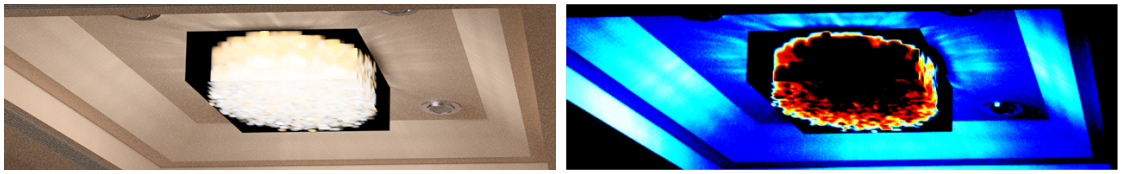
Figure 3.2: Comparison of the appearance of the Statler luminaire as rendered by different methods. The images correspond to cropped sections of Figure 3.1 using the histogram equalization tone mapping. Our method shows attributes for the illumination and the luminaire appearance qualitatively similar to those present on the real world scene.



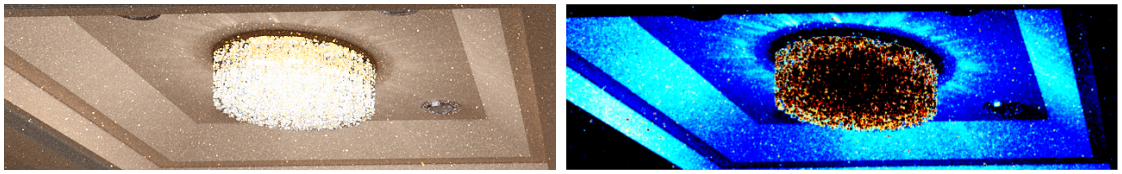
(a) Reference: Bidirectional Path Tracing (150× time)



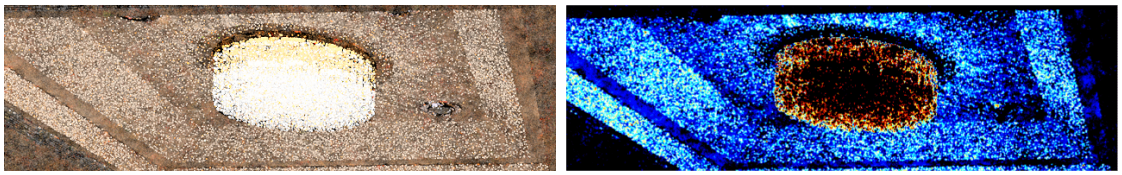
(b) Our method



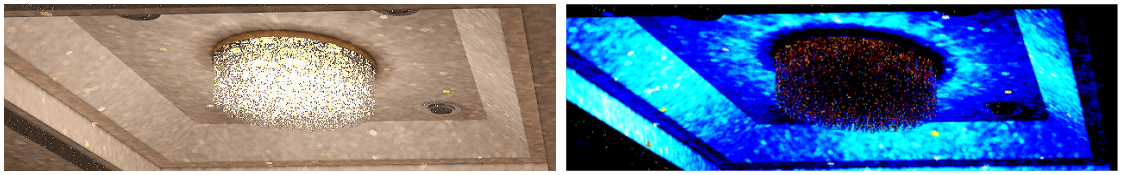
(c) Canned lightsource (8× data)



(d) Bidirectional Path Tracing (equal time)



(e) Manifold Exploration (ERPT) (equal time)



(f) Stochastic Progressive Photon Mapping (equal time)

Figure 3.3: Comparison of the illumination from the Statler luminaire incident on the ceiling as rendered by different methods. The images correspond to a cropped section of the scene from Figure 3.1. Images on the left column are tone mapped using a global operator, whereas false color images on the right illustrate the logarithmic luminance, making the variations on the illumination more apparent.

times more precomputed data than our method. These results demonstrate that our method produces much higher quality than prior approaches given similar resources and is the first practical rendering method demonstrated for such luminaires.

3.1 Luminaire Illumination Insights

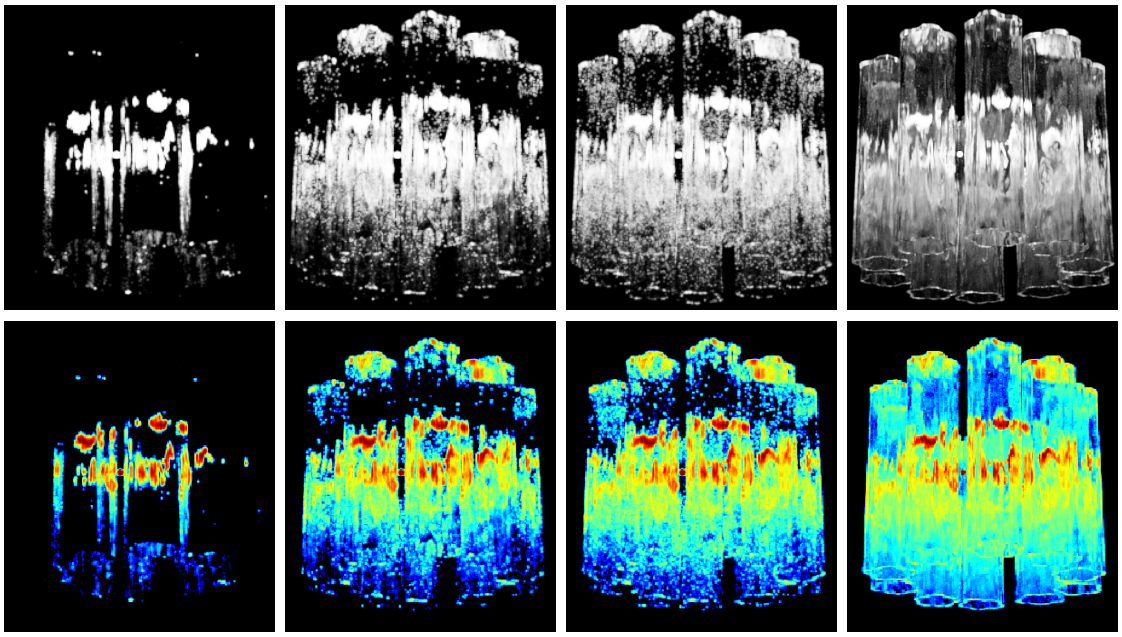
In our method, we first split luminaire rendering into illumination and appearance components because we found these have significantly different accuracy and cost requirements. Illumination is generally the dominant cost since luminaires usually occupy only a small fraction of the image pixels. Far-field point source approximations are cheap to evaluate, but are 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 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 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 or dynamically to optimize accuracy vs. 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 3.4 shows a series of appearance approximations for a simplified Statler luminaire (Figure 5.2a) 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 3.4, 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 over-



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

Figure 3.4: This figure shows a simplified version of the Statler luminaire, rendered using path tracing with different maximum bounces (path lengths.) The bottom row shows a false color version of the top row to better illustrate the radiance variation. All images should be compared to a reference image with unbounded path lengths (d).

all "glow" of the luminaire due to the multiple scattering within the complex luminaire. To show patterns more clearly, in the second row of Figure 3.4, 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 dissimilar characteristics, we choose different strategies to render "sparkle" and "glow" components. 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.

We used this same simplified luminaire model to test how the method proposed by Kniep et al. [25] would perform, since as of this writing it was the most recent related work dealing specifically with the appearance of luminaires. Figure 3.5 shows the results of our tests, in which we varied the number of photons (from ten thousand to one hundred million) and the angular bandwidth parameter (λ in the original paper.) The value of λ serves to weight the influence of the photons' direction during density estimation: a value of zero would behave as traditional photon mapping, where only the position matters, whereas larger values would lead to only include contributions from photons closely matching the query direction. Our tests indicate that even one hundred million particles are not enough to get a high quality rendering of the luminaire's appearance as they differ considerably from the reference, created with bidirectional path tracing. Nevertheless, we also found that with moderate values of λ (i.e., taking into account the photons' direction without making it more important than position) using even less than one million photons is enough to get a reasonable approximation of the "glow" component of the luminaire. Thus these experiments confirm that the "glow" of a luminaire is a low-frequency function which may be approximated by more adequate methods than using a 6D data-intensive photon map which requires expensive density estimation queries.

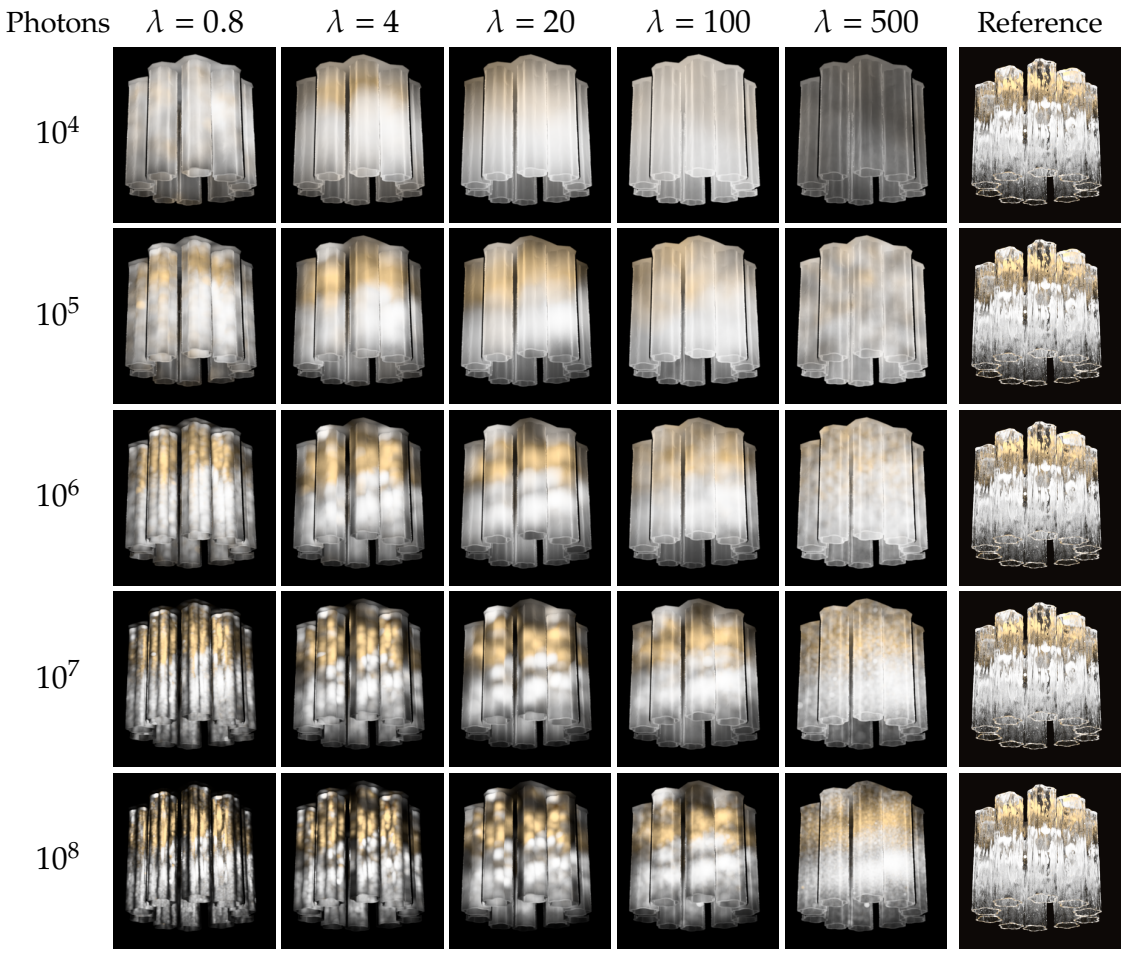


Figure 3.5: Appearance of a simplified version of the Statler luminaire, using the 6D photon mapping method by Kniep et al. [25]. Even when using huge amounts of photons and variations of the angular bandwidth parameter λ, the results using this technique are far from matching the reference (last column.)

CHAPTER 4

ALGORITHM

This chapter presents our algorithm to generate the data structures which enable rendering the illumination and appearance of complex luminaires, and describes how to use them effectively at rendering time (Section 4.1.) During the precomputation step we use a Monte Carlo particle-tracing process (Section 4.2) to create the APLs, which store the exiting illumination from the luminaire (Section 4.3), and the low-frequency radiance volume, which approximates the costly multiple scattering during the luminaire appearance rendering (Section 4.4.) When rendering an environment using a luminaire processed with our method, we gather the illumination form the APLs and use a limited-depth ray tracing strategy along with the radiance volume for the luminaire appearance (Section 4.5.) We conclude this chapter by explaining the design and parameter choices we made in our implementation of this algorithm (Section 4.6.)

4.1 Method Overview

Figure 4.1 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 4.1b.). The radiance volume is a low fre-

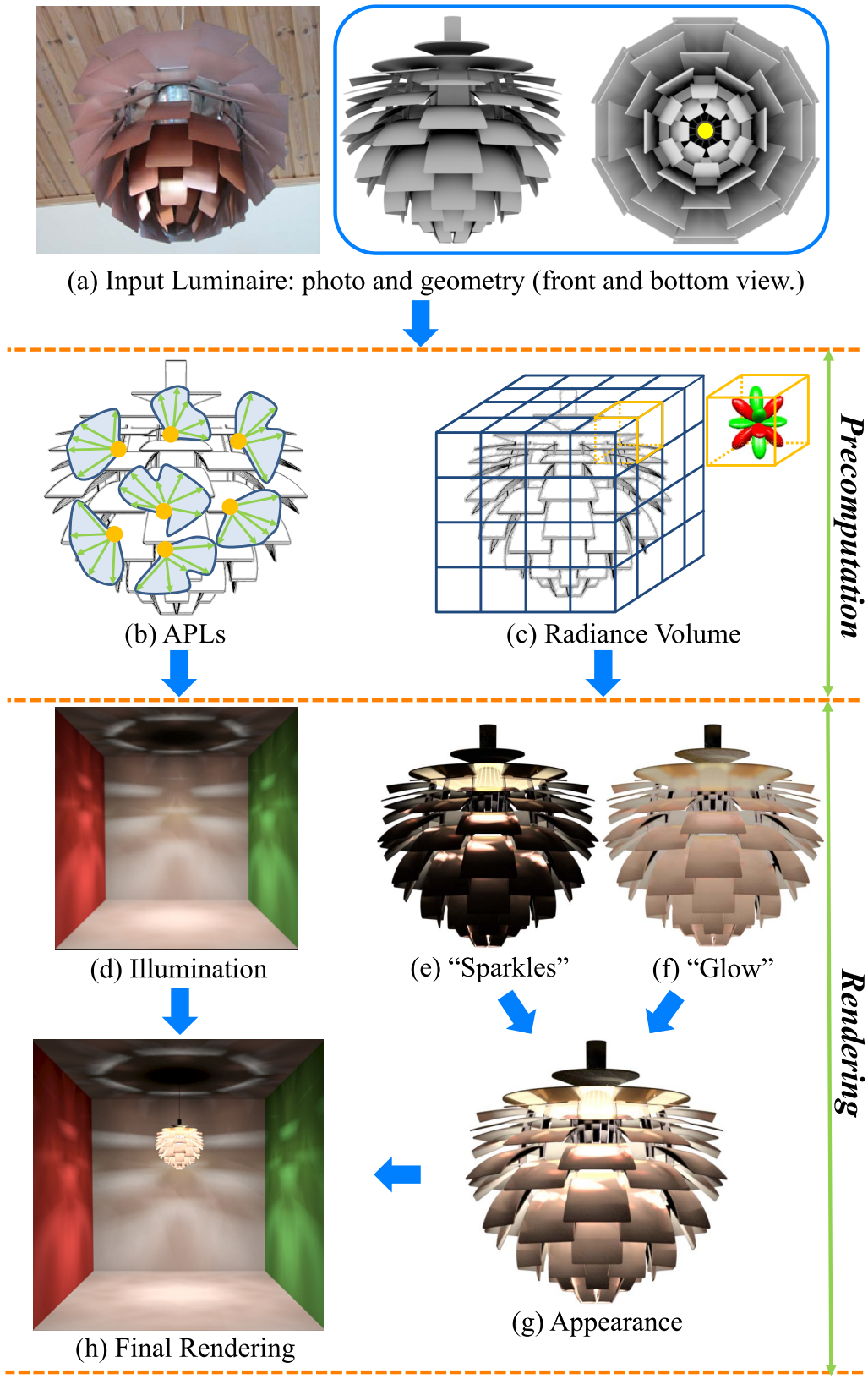


Figure 4.1: Pipeline of our luminaire rendering method.

quency representation of the light flow inside the luminaire that is stored as a low-resolution 3D grid of spherical harmonic coefficients (Figure 4.1c.) 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 the radiance volume are constructed incrementally so that we do not store the full set of particles, i.e., we continuously integrate additional groups of particles into the current estimates of both structures until we achieve the desired quality.

At render time, points are shaded using either the appearance or illumination procedures depending on whether they are located on the luminaire geometry or elsewhere in the scene. The illumination, at points not hitting the luminaire's geometry, is computed by evaluating the direct illumination from the precomputed APLs (Figure 4.1d.) 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 illumination procedures above. This is essential to compute scene-dependent aspects of the luminaire appearance from effects such as transparency or reflection. If the maximum recursion depth is reached, then we look up a value from the precomputed radiance

volume and terminate the ray. As discussed in Section 3.2, luminaire appearance contains high-frequency "sparkles" (Figure 4.1e) and the low-frequency "glow" (Figure 4.1f.) 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 ray 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 to be a good trade-off between quality and performance, and essential in achieving visually good appearance fidelity.

Together our illumination (Figure 4.1d) and appearance (Figure 4.1g) procedures enable the high-quality rendering of scenes with complex luminaires (Figure 4.1h.)

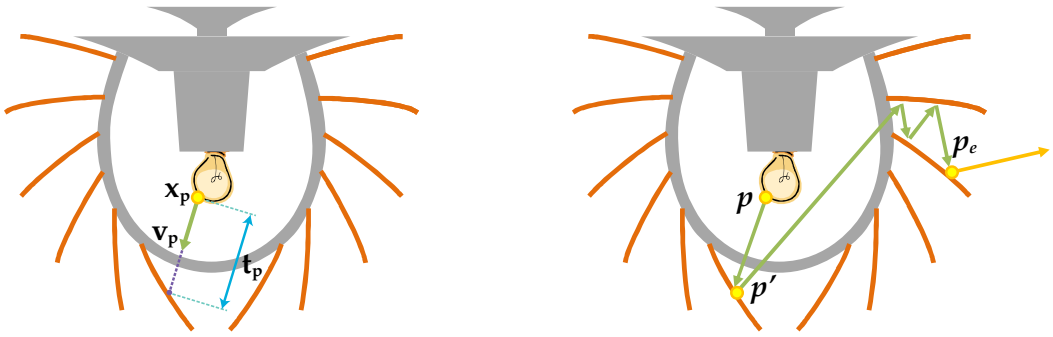
The following sections first present the particle tracing step in the precomputation stage and how to use it to generate a set of APLs, and a low-frequency radiance volume for the input luminaire, respectively. Afterward, we describe how to utilize these two precomputed data structures, combining with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

4.2 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 virtual point light (VPL) generation, generates the particles [9]. The source for the particles may be any adequate representation of the light emitters: phenomenological such as area light sources and point lights, or experimental such as goniometric data and ray sets [37]. This stage requires only the luminaire with its geometry, materials, and emitters as input; hence it is scene independent.

As shown in Figure 4.2a, 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 $\mathbf{t_p}$ value of $\mathbf{p}$ is determined at the same time. Following the standard practice the direction and flux of $\mathbf{p'}$ are determined by importance sampling of the BSDF at the current intersection point. Such particle tracing process continues until the particle $\mathbf{p_e}$ exits the luminaire geometry (Figure 4.2b.)

As standard in Monte Carlo particle tracing, at each interaction with an element of the luminaire some of the energy from $\Phi_\mathbf{p}$ is absorbed (unless its material is perfectly reflective or transmissive, ideal conditions mostly absent in the real world.) Therefore the original particle flux is greater than or equal to that of the newly generated particle, i.e., $\Phi_\mathbf{p} \geq \Phi_{\mathbf{p'}}$. Since luminaires generally use materials on which only little absorption occurs in order to maximize their energy



(a) Initial generation of particle $\mathbf{p}$, originating on the surface of an actual emitter.

(b) Propagation of $\mathbf{p}$ within the luminaire, culminating in exiting particle $\mathbf{p_e}$.

Figure 4.2: Particle tracing process during precomputation.

efficiency, a particle originating at an emitter may follow very long paths before its flux diminishes substantially. For example, a particle being scattered through a clear glass chandelier with 90 % reflection and transmission will bounce more than 42 times before it loses 99 % of its original flux. The actual path-lengths depend on the luminaire complexity and while a typical particle goes through less than a dozen bounces, we found that some high-energy particles may go through hundreds of bounces before exiting. To avoid infinite path lengths we use Russian roulette [9, Section 5.3.2] as an additional stopping criteria.

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 4.2b.) Section 4.3 describes how to convert those 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 4.2b); Section 4.4 explains this process in more detail.

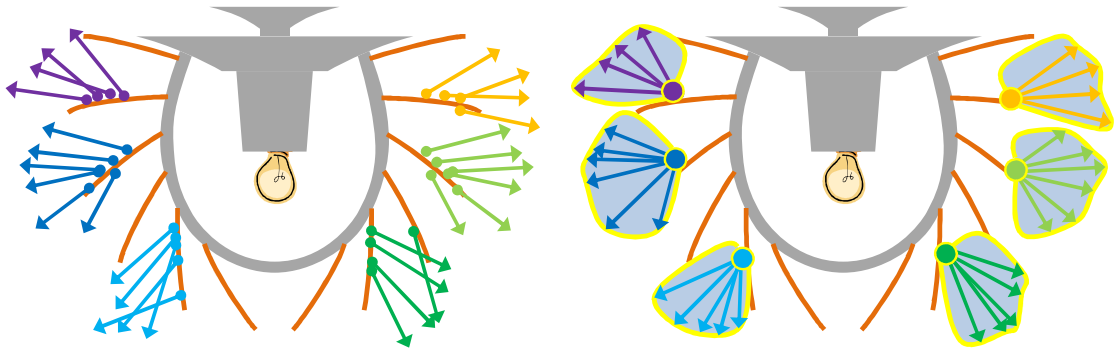
4.3 Generating APLs Using Clustering

Instead of storing all the exiting particles, we choose to group them into a set of clusters (Figure 4.3a), and treat each cluster as an APL (Figure 4.3b.) It is not evident what constitutes a good cluster center. Both previous work [20] and our tests corroborate that choosing cluster centers *a priori* does not lead to good near-field illumination results without requiring prohibitively large amounts of precomputed data, and instead tend to introduce aliasing artifacts. Hence we looked into using unsupervised machine learning though clustering. We explored several different clustering methods, including k-means clustering, and achieved the best result with agglomerative clustering [57] using a metric that aims to generate spatially compact clusters with roughly equal power (i.e., flux.) We rejected other clustering methods because in our experiments the APLs created by them failed to represent the near-field illumination of the luminaire, even when using a large number of clusters. The clustering cost metric we used is:

$$I = \left(\Phi_{C_1} + \Phi_{C_2}\right) \cdot \mathrm{Diag}\left(C_1 \cup C_2\right)^4 \tag{4.1}$$

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. At each step of the clustering process it merges the two clusters which, once combined, will yield the minimum value of the clustering cost metric among all possible combinations. The process finishes when all clusters have been merged into a single root node.

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



(a) Particle clustering (b) APLs generation

Figure 4.3: Particle clustering for APLs generation in precomputation.

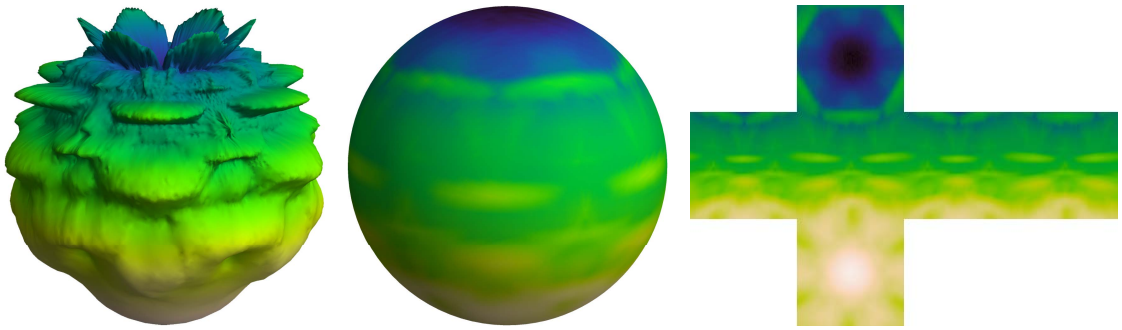
corresponding to each individual particle. Each inner node of the binary tree represents a cluster containing all the child clusters below it in the tree. Clearly, the clustering cost of the tree nodes decreases from the tree top to the bottom. To generate a certain number of APLs, we adopt a *maximum priority queue* filled with tree nodes; the associated priority is the clustering cost computed using Equation 4.1 for each node. Starting by pushing the root node into the queue, we will check whether the current number of nodes in the queue has already reached the target number (e.g., 512.) If the current queue size is less than the target, the node with the largest clustering cost will be *popped* out from the queue while its child nodes will be *pushed* in. We repeat this process until the number of tree nodes equals the target amount, and then each node in the queue become a cluster for generating the set of APLs.

Finally, the critical piece of information we need is the centroid of each cluster. Its value is simply the center of the bounding box containing all the seed particles assigned to the cluster. We tried more sophisticated approaches, e.g., calculating the center of mass of the cluster weighted by each of its particle's power, but the resulting illumination did not show any improvements with re-

spect to the simpler approach. Having obtained the centroid of each cluster, their positions remain unchanged through the rest of the particle process, i.e., assigning additional particles to a cluster during the APL generation process does not change the cluster's position.

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 is enough to generate cluster centers which lead to APLs providing accurate near-field illumination, 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 (see Section 4.6.) Thereafter, every exiting particle can be simply assigned to the closest cluster, using the Euclidean distance between the exiting particle's position and the cluster's centroid as the evaluation metric.

Each cluster keeps track of its centroid and the directional radiance distribution of all the particles assigned to it. At the end of the precomputation, these clusters turn into a set of APLs (Figure 4.3b), where the centroid of each cluster becomes the APL's position. Effectively, what this process does is "move" each of the particles assigned to the cluster from their original exiting position to the APL's position, while maintaining their original outgoing direction and flux. To estimate the radiance distribution of an APL we only use the flux and direction of the particles within its corresponding cluster, This way the radiance distribution of an APL becomes a function defined on the sphere of directions (see Figure 4.4.) Each particle gets assigned to a bin using a 2D projection of the sphere to which we add the current particle's flux; this is analogous to gener-



(a) Function as vectors on the sphere.

(b) Function mapped on sphere surface.

(c) Function unfolded into a surface as faces of a cube.

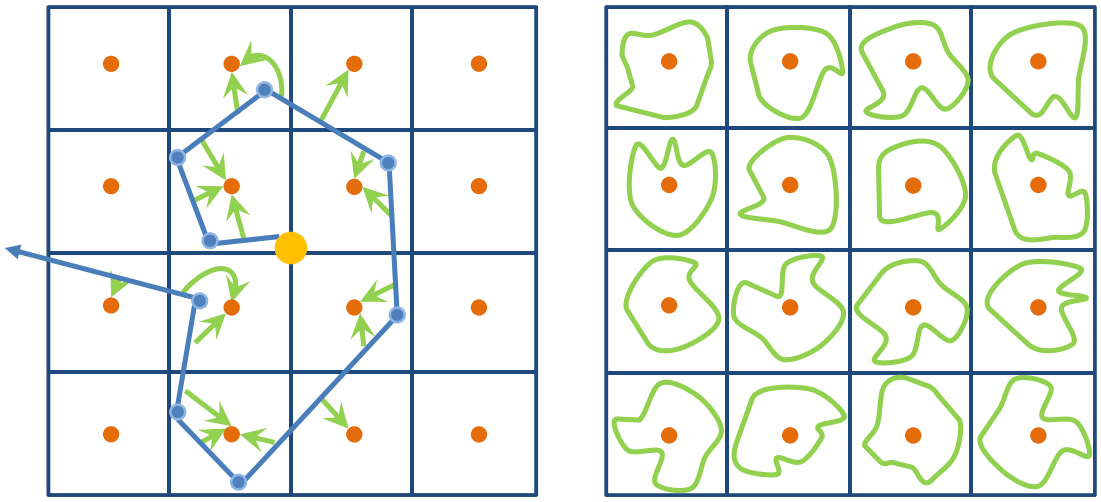
Figure 4.4: Representation of functions defined on the sphere of directions. Color represents the magnitude of the radiance vectors from the actual APL (a). Values are mapped to geometric abstractions such as a sphere (b) or an unfolded cube (c) for computational efficiency.

ating a two-dimensional histogram for the directional radiance distribution of the APL. After accumulating the contribution from all the relevant particles we store the resulting high dynamic range bitmap to be queried during rendering.

The clustering part of our method is related to that proposed by Mas et al. [31] but differs in several important respects. Our clusters are not restricted to lie on a bounding surface and are converted to true point sources rather than vertices on an interpolation mesh. This allows us to represent the luminaire illumination far more accurately using fewer clusters and greatly simplifies the importance sampling of the clusters at render time.

4.4 Radiance Volume for Appearance

To generate the radiance volume for the luminaire's low-frequency "glow", we use all the particle segments within the luminaire (green vectors in Figure 4.2b.)



(a) Particle path showing each segment's contribution to all traversed voxels.

(b) Spherical harmonics coefficients after precomputation in each voxel.

Figure 4.5: Computation of the radiance volume. During the precomputation, each light particle adds energy to every voxel cell it intersects proportional to the path segment length within the voxel (a). Each cell stores spherical harmonic coefficients to represent its low-frequency directional distribution (b). Figure adapted from Moon et al. [35].

It is impractical to directly save all of these elements and instead we project them into a low resolution volume which stores the spatial and directional variations of the internal radiance field. Figure 4.5 illustrates this process.

During precomputation, the bounding box of the luminaire model is firstly discretized into a low-resolution uniform grid with cube-shaped voxels. We adopt the approach from Moon et al. [35] in order to record the contribution of a light path to the grid voxels intersecting with its trajectory. For each voxel that a light path segment $\mathbf{t_p}$ intersects, it makes a contribution to the local spherical radiance function of the current voxel, proportional to the path segment length within the voxel. The spherical radiance function of each voxel is generated by gathering the contributions from all the intersecting light paths.

Due to its low-frequency nature, to efficiently represent each voxel within the radiance volume, we project the spherical radiance function into the spherical harmonics (SH) basis and hence each voxels needs to store only a few SH coefficients. At render time, this volume can be quickly queried to approximate the low-frequency radiance for any position and direction inside the luminaire.

4.5 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 global illumination 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, 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 complexity 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's intersection point and incident direction to compute the "glow" effect, terminating the path. When

```
Function LuminaireRendering(Intersection Record p)
    if p is on the luminaire geometry then
        Start to trace ray within the luminaire geometry;
        while The number of bounces ≤ n_b do
            if l hits the emitter then
                Return the energy L_e from emitter;
            end
            if l exits the luminiaire then
                Compute new intersection record p' with scenes;
                LuminaireRendering(p');
            end
        end
        Compute the "glow" L_g using radiance volume;
        Return L_g;
    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 BRDF at p;
            L_r = L_r + L_i;
        end
        Return L_r;
    end
```

Figure 4.6: Pseudocode of the rendering routine.

the eye ray exits the luminaire before $\mathbf{n}_b$ bounces it simply gathers the incident radiance from all relevant sources.

The challenge is to define the reflected radiance, L_r due to the luminaire itself while not including environmental reflectance. The algorithm in Figure 4.6 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 or not. If $\mathbf{p}$ is on the luminaire, we apply our previously explained method for combining both the high and low frequency components of the luminaire appearance (limited-depth

ray tracing followed by a radiance volume query upon exceeding the maximum number of bounces $\mathbf{n}_b$.) 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 this algorithm only shows the simple illumination rendering routine that iterates over all the APLs without using an adaptive method such as a hierarchy.

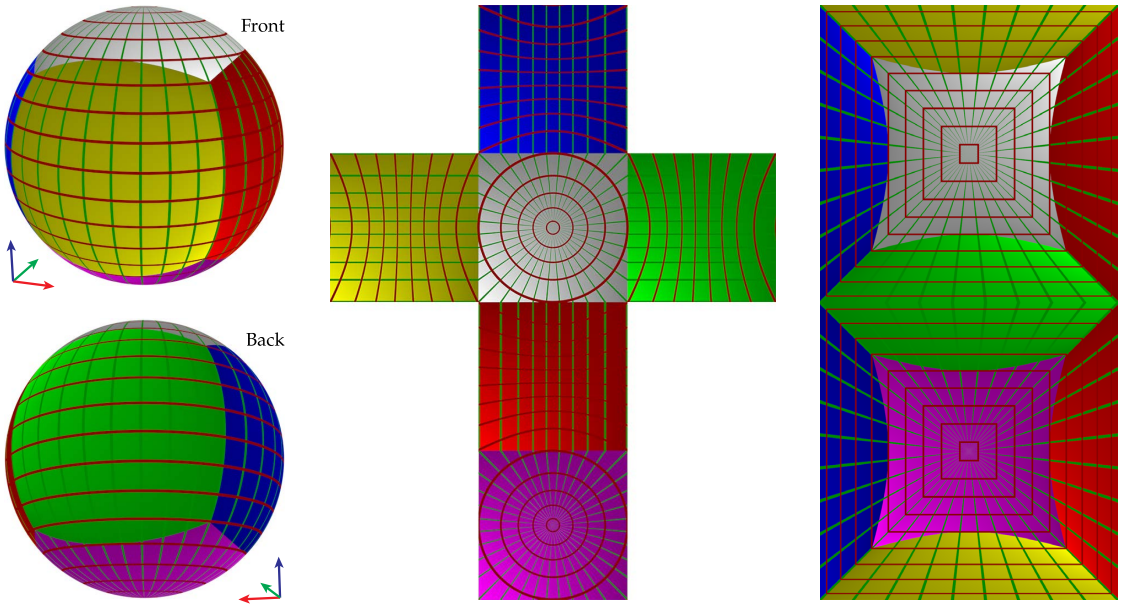
4.6 Implementation Details

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. The main ones are the number of particles to use, the format of the APLs, and the configuration of the radiance volume grid.

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. We chose this approach due to performance considerations: even though the version of agglomerative clustering we use performs better than $O(n^2)$ with respect to the number of seed particles [57], clustering costs do have a considerable increase from the few seconds it takes to cluster up to one million particles. Actually, our tests showed no difference in quality when using more than 250 000 seed particles to determine the cluster's centroids. Furthermore, we found than when using more than one million particles numerical precision becomes an issue. Since agglomerative clustering starts by placing each seed particle into its own cluster, they can be so close that their initial clusters already intersect before the merging process has even started.

While it is possible to use more sophisticated density estimation techniques for deriving the directional radiance distribution of an APL from the particles associated with its cluster, we use the simple bin-accumulation approach because it proved to be very efficient. It is easy to parallelize using a mostly lock-free implementation and allows iterative refinement of the APLs. This simply requires projecting and accumulating new particles into the existing bitmaps of the APLs, improving their quality with each additional pass.

By default, we generate 512 APLs per luminaire. As we will demonstrate later in Sections 5.2 and 5.3, we found that 512 APLs provide accurate illumination for both the near- and far-field. The directional radiance distribution for each APL is stored as a 2D texture map of 256×512 resolution using the concentric mapping projection [45]. The pixels store RGB values in Float16 format (half-precision, IEEE 754-2008 binary16, see [61, Table 3.5].) We use the concentric mapping projection because, unlike more commonly used options such as latitude/longitude or cube maps, it is an area-preserving projection; each pixel covers the same solid angle, thereby conversions between the projected-image and spherical measures only need a single scaling constant. The 1:2 aspect ratio reduces the maximum anisotropy of the projection when covering the whole sphere [48]. Figure 4.7 illustrates the different distortions incurred when mapping a spherical function to a 2D domain, using both the cube map and concentric mapping projections. Even though the cube map distorts the original domain less that the concentric mapping projection, its uneven solid angle coverage and its inability to map the entire sphere on a single image made it less convenient for our software-base implementation.



(a) Original function (b) OpenGL cube map (c) Concentric mapping

Figure 4.7: Mapping of a function defined on the sphere of directions using the OpenGL cube map [13] and the area-preserving concentric mapping [45] projections. Both the parallels (red) and the meridians (green) are uniformly distributed every 10°.

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 disparate 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 (single-precision, IEEE 754-2008 binary32.) The SH projection is the dominant cost of the radiance volume generation process since a particle often contributes to multiple voxels. We use an evaluation routine for SH coefficients based on pre-factored recurrence relations [46] which proved to be vastly more efficient than the independently evaluated polynomials commonly employed for low-order SH.

CHAPTER 5

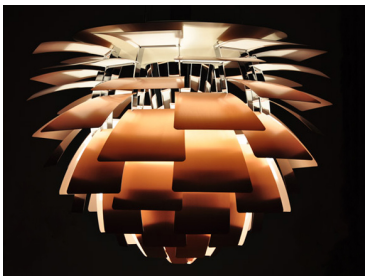
RESULTS AND EVALUATIONS

In this chapter, we present results to evaluate the accuracy and performance of our method for the following seven complex luminaires, based the real-world models shown in Figure 5.1:

- **Troffer:** An overhead office light, with three fluorescent ballasts and a grid refractor; exported from Autodesk Revit 2012.
- **P.H. Artichoke:** A single spherical light source surrounded by metallic leafs. Based on a design by Poul Henningsen.
- **Vita Silvia:** A single spherical light source surrounded by 60 translucent leafs, in an arrangement reminiscent of the P.H. Artichoke.
- **Statler:** Twelve light sources arranged in two concentric circles, embedded within 117 glass shades. Based on the chandeliers at the Statler Hotel.
- **Met Sputnik:** 22 light sources suspended between glass pendants. Based on the design by Hans Harald Rath for J. & L. Lobmeyr at the New York Metropolitan Opera foyer.
- **V&A Chandelier:** Glass chandelier with 18 light sources, based on a model at the Victoria and Albert Museum, London.
- **Chihuly:** Blown glass chandelier with three spherical and two cylindrical light sources, based on the 2001 chandelier by Seattle artist Dale Chihuly at the Victoria and Albert Museum rotunda.

Even though these seven luminaires are not representative of general light sources, we chose them for several reasons. All but the Troffer office light source

Troffer[1]



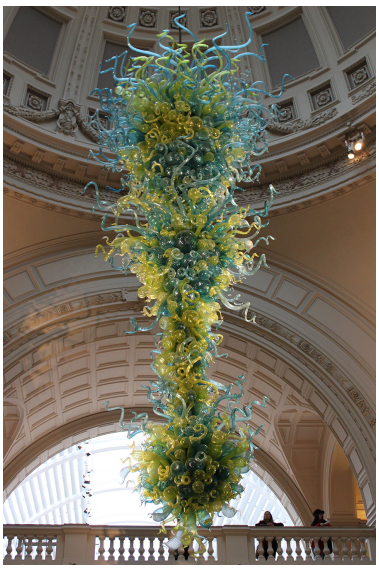
P.H. Artichoke[2]

Vita Silvia[3]

Statler[4]

Met Sputnik[5]

V&A Chandelier[6]



Chihuly[7]

Figure 5.1: Real-world luminaires' photographs.

[1] Ameleco Electric Inc. "Troffer fixture 2x4 432K." *Ameleco Web Store*. 11 December 2010.
[2] Stardust. "PH Artichoke Lamp." *Modern Design*. 23 February 2010.
[3] Vita Living ApS. "Silvia." *Vita Media Kit*. May 2014.
[4] E. Velázquez-Armendáriz. "Chandelier by ballroom foyer, Statler Hotel." 20 December 2010.
[5] J. & L. Lobmeyr. "Met Foyer Chandelier." *Lobmeyr: The Magic of Sensuality*. 13 May 2013.
[6] Victoria and Albert Museum, London. "Chandelier C.1-1936". *V&A Collections*. March 2003.
[7] E. Velázquez-Armendáriz. "Dale Chihuly chandelier at the V&A rotunda." 8 May 2010.

have significant geometric and material complexity. The convex hulls of the light sources represent a wide variety of shapes and sizes thus proving the generality of our algorithms. Only the Troffer, P.H. Artichoke and Vita Silvia luminaires may be closely covered by a recognizable shape (i.e., a bounding box or sphere.) Most importantly, many have been designed to generate attractive illumination patterns, and for a simulation to be valid, it should be able to duplicate this near-field illumination phenomena in addition to the luminaire's appearance. As will be shown in this chapter, in addition to reducing the computational time, our algorithms generate results which meet these criteria.

The actual emitters of these luminaires are modeled as small spherical area light sources with diffuse emission. The glass components are modeled as 3D solids with with a microfacet-based BSDF [56] to simulate slightly roughened glass, which we found to be 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 [21]. 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.)

5.1 Qualitative Evaluation of Appearance

To evaluate the quality of our luminaire appearance rendering, Figure 5.2 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 luminaire images shows significant noise even after nearly 60 hours of computation. As shown in Figure 3.2 other potential reference 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. More details on the acceleration speed-ups and memory requirements are presented in Section 5.4.

There are small differences between our results and the reference images, mostly due to the low resolution nature of our radiance volume, but the overall appearance is perceptually very similar. A canned lightsource result is shown in Figure 5.2e 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 black region in Figure 1.2c.)

As discussed in Section 4.5, 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 is usually less than 10. The exception is the Statler luminaire with its dense arrangement of glass shades where we found setting $\mathbf{n}_b = 16$ was necessary to generate high quality "sparkles". We tune the

(a) Reference (59.86 h)

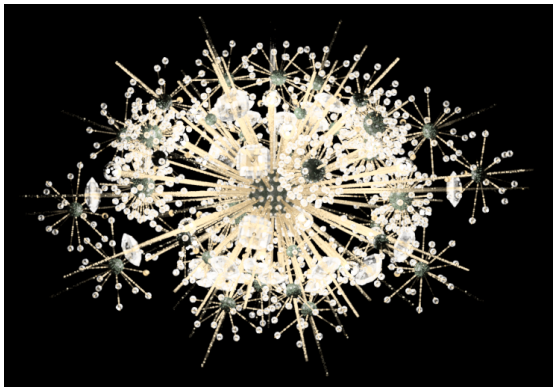
(b) Ours with 16 bounces (5.94 min, 605×)

(c) Reference (10 h)

(d) Ours with 8 bounces (11.86 min, 51×)

(e) Canned lightsource (1 min)

(f) Reference (10.35 h)



(g) Ours with 6 bounces (8.11 min, 77×)

Figure 5.2: Luminaire appearance comparison between simulated reference images and our results for Statler (top), V&A Chandelier (middle), and Met Sputnik (bottom) luminaires. Timings indicate the computation time required after completing the precomputation step.

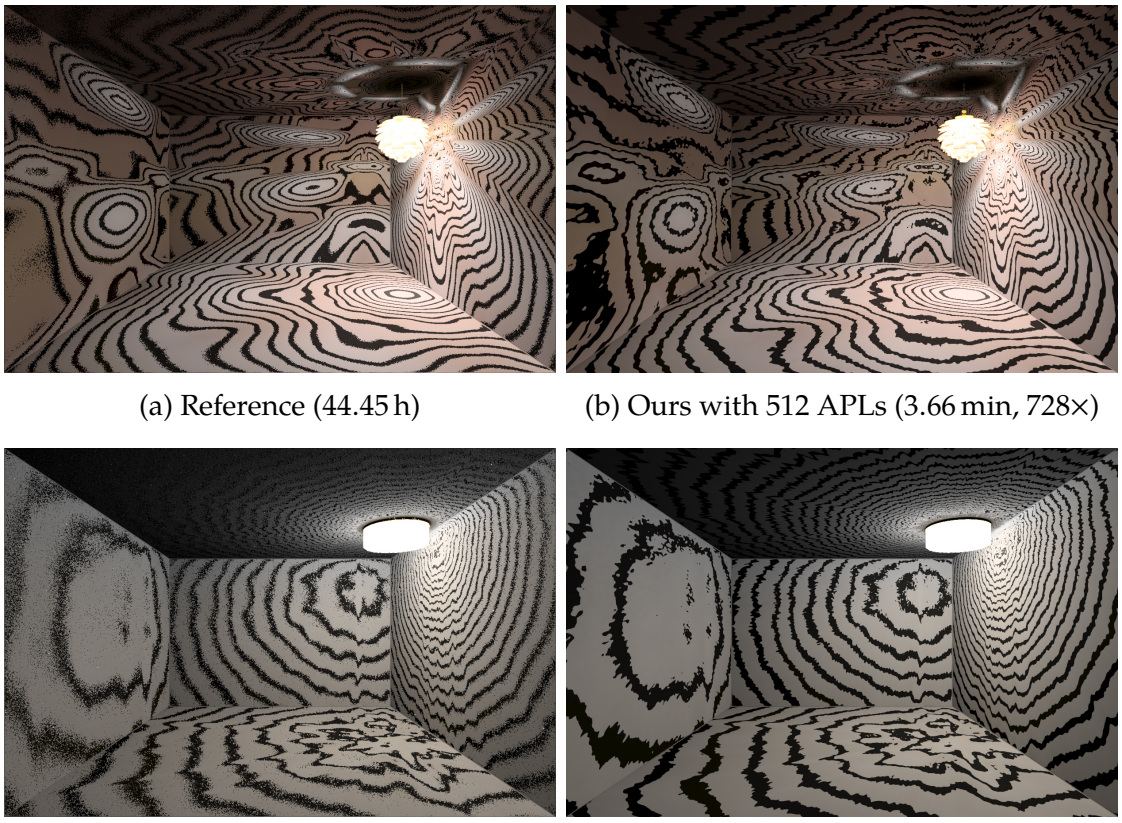
depth limit manually for each luminaire, but we expect this would be done as part of the preprocess and included with the luminaire data.

5.2 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. For this purpose we rendered each of our luminaires in an empty $3\,\mathrm{m} \times 3\,\mathrm{m} \times 2\,\mathrm{m}$ room. The luminaires are placed near one wall so that we can observe both near- and far-field effects. Motivated by Verbeck and Greenberg [54], we enhance these images with *iso-contours* 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 iso-contours help to visualize the details of the illumination and judge its accuracy.

Figures 5.3 and 5.4 show the iso-contours for the reference BDPT renderings and for our results. The iso-contours 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 iso-contour images are 1248×842 and include illumination from the luminaire but do not include indirect illumination from the rest of the scene to better isolate illumination quality. To achieve this, our results use the direct illumination from 512 APLs; the reference renderer was constrained to only allow paths with at most one non-luminaire vertex.

Figure 5.5 compares illumination using our method and two prior methods to a reference solution for the V&A Chandelier (Figures 5.5a and 5.5b.) Single point

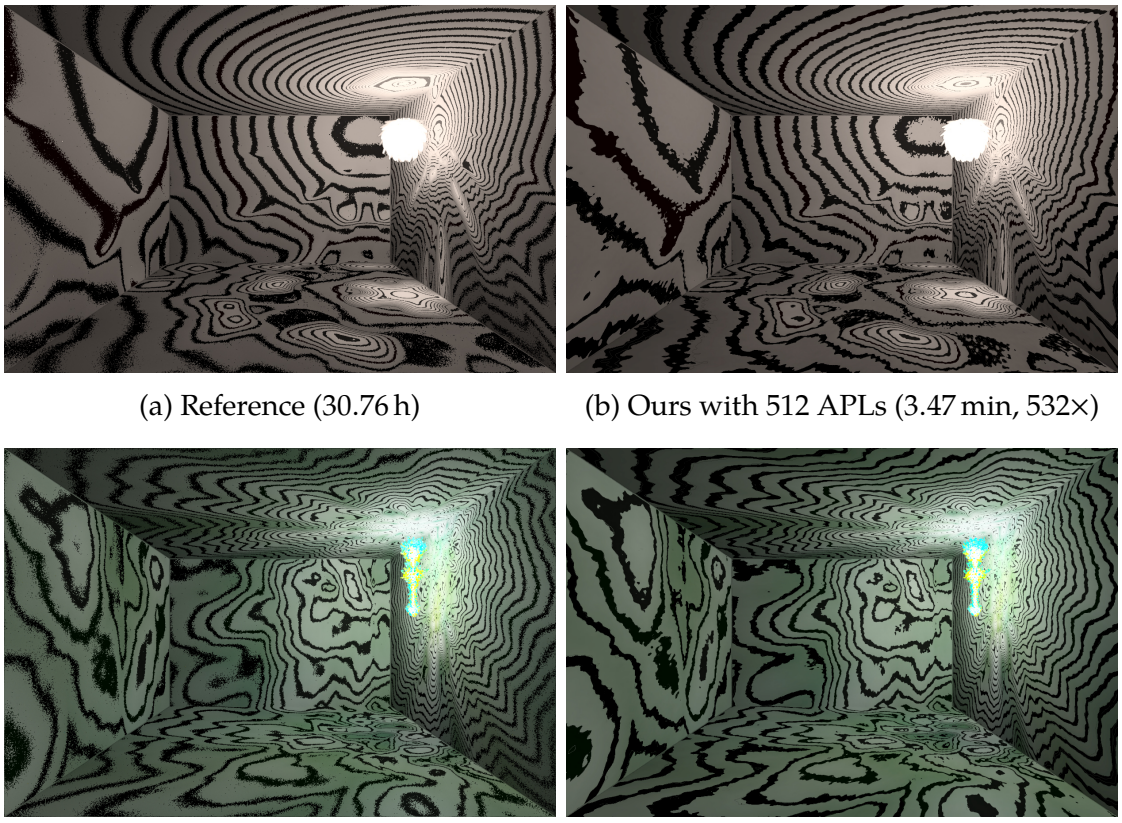


(a) Reference (44.45 h) (b) Ours with 512 APLs (3.66 min, 728×)

(c) Reference (190.96 h) (d) Ours with 512 APLs (3.99 min, 2874×)

Figure 5.3: Comparison of the illumination iso-contours for the Artichoke (top) and Statler (bottom) luminaires. Note that both the near-field and far-field distributions are close to the reference solution.

far-field methods (Figure 5.5d) 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 lightsource method (Figure 5.5c), although more visually reasonable, still shows many illumination differences. It also shows visual artifacts caused by the fact that it stores the illumination on the luminaire bounding box, and reveals the aliasing which occurs from undersampling despite using eight times more data than our method. Our method using 512 APLs (Figure 5.5b) is



(a) Reference (30.76 h) (b) Ours with 512 APLs (3.47 min, 532×)

(c) Reference (55.37 h) (d) Ours with 512 APLs (4.06 min, 818×)

Figure 5.4: Comparison of the illumination iso-contours for the Vita Silvia (top) and Chihuly (bottom) luminaires. Note that both the near-field and far-field distributions are close to the reference solution.

the most accurate and closely matches the reference.

In Figure 5.6, we compare our method with the canned lightsource for a very 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 and identifiable 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. Once again the canned lightsource result (Figure 5.6c) shows

(a) Reference (52.03 h)

(b) Ours with 512 APLs (3.86 min, 810× speedup)

(c) Canned lightsource with 4278 spatial samples (77.73 min)

(d) Single Far-Field (1.99 min)

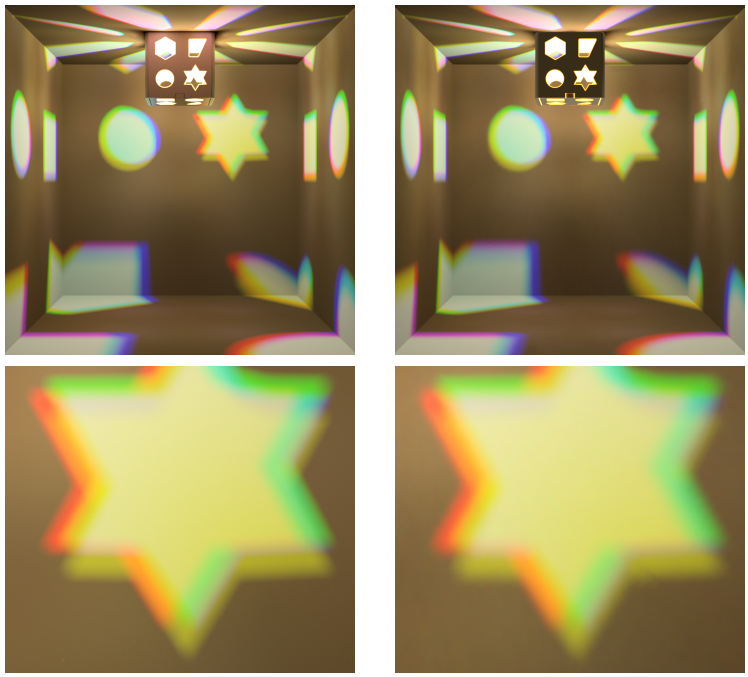
Figure 5.5: Illumination rendering comparison for the V&A Chandelier luminaire. Our method (b) produces illumination and contours that closely match the reference image (a) while the canned lightsource (c) and the single point far-field (d) methods are much less accurate with visually obvious illumination errors.

strong aliasing artifacts even when using eight times more data than our method. This is because the illumination is stored as a light field located on the luminaire's bounding box which is far from the actual emitters. Using more samples in the light field (Figure 5.6d) 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 5.6e. In general, we find that our method produces higher quality illumination results than light field based approaches while also requiring much less data and being easier to evaluate.

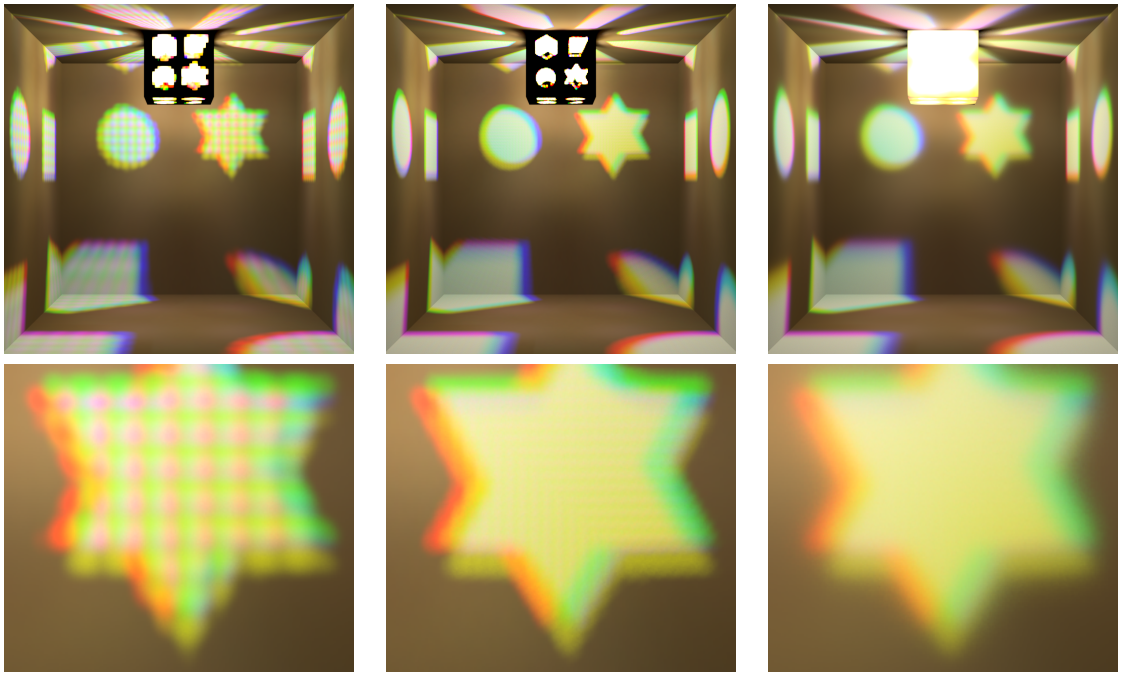
Illumination accuracy is also important for correct material rendition. In Figure 5.7, we compare the effects of different luminaire representations when lighting an irregular metallic object. In this scene setting, the object is placed 30 cm bellow our 1.219 m $\times$ 2.438 m troffer luminaire. Our method (Figure 5.7b) closely matches the reference rendering (Figure 5.7a) even at such close range. Note the distinctive bright patterns cast by the troffer's three ballasts. In comparison, the single point far-field produces highlights in the wrong places (Figure 5.7c), whereas the commonly-used uniform area light source approximation (Figure 5.7d) yields a flat appearance, which conveys a different material impression due to the missing highlights.

5.3 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

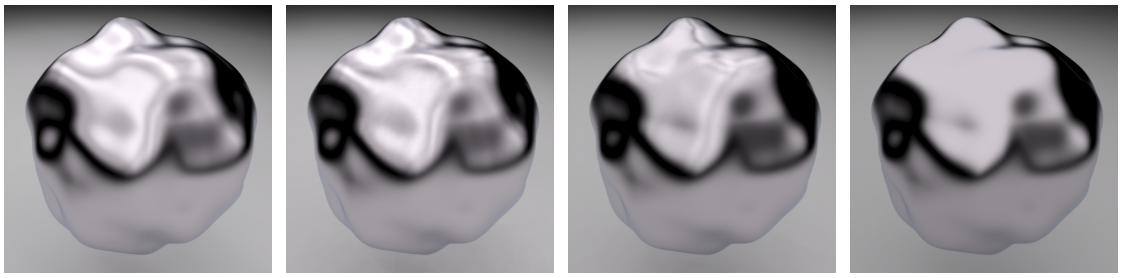


(a) Reference (b) Ours with 512 APLs



(c) Canned lightsource, 4158 samples (d) Canned lightsource, 32 540 samples (e) Canned lightsource, filtered 32 540 samples

Figure 5.6: Small lights inside the shape sorter cast sharp patterns which our method can represent with only a slight loss in detail. The canned lightsource requires an impractical 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 source

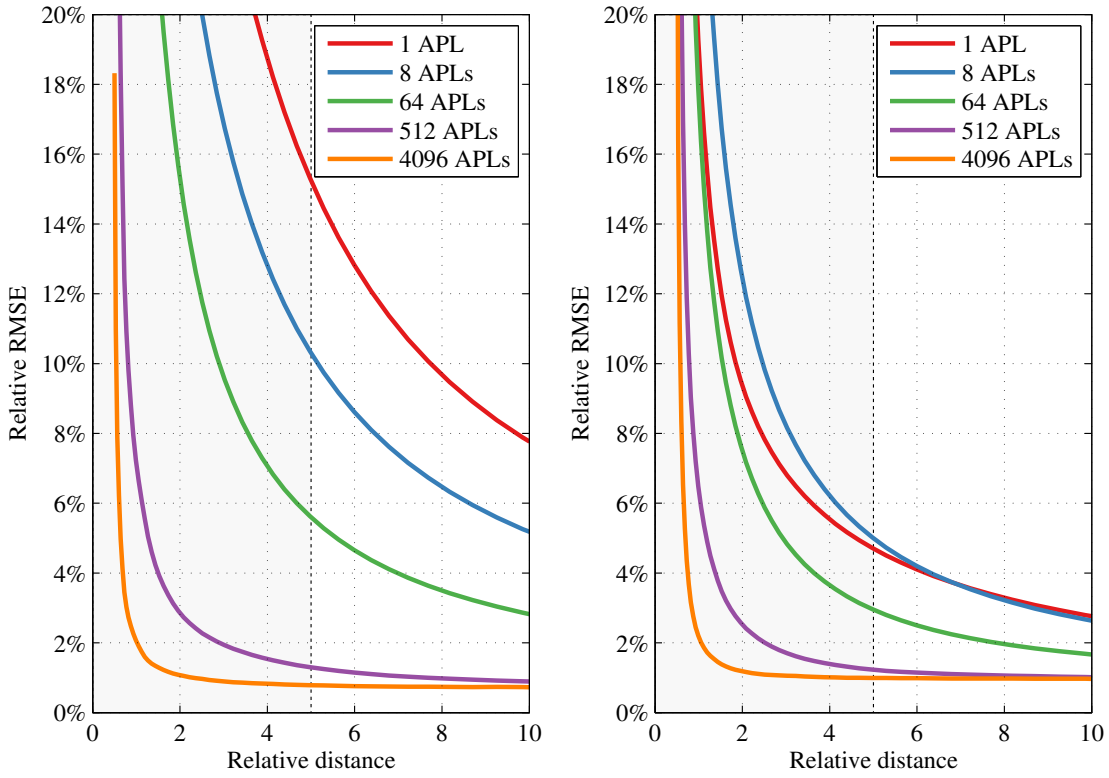
Figure 5.7: Effects of illumination accuracy on material rendition. Note that the highlights on the reference image (a) are closely matched by our method (b). The single point far-field (c) and uniform area light source (d) approximations convey a different material impression.

for analyzing actual light fixtures [43]. 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 as follows:

$$\text{Relative RMSE} = \frac{\sqrt{\sum_{i=1}^{N} (\mathbf{E}_{Ref}(i) - \mathbf{E}_{APL}(i))^2 / N}}{\sum_{i=1}^{N} \mathbf{E}_{Ref}(i) / N} \tag{5.1}$$

where $\mathbf{E}_{Ref}(i)$ and $\mathbf{E}_{APL}(i)$ represent the irradiance values for surface patch i, computed using all the particles or the APLs respectively, and N is the number of surface patches. Using a relative error metric makes it easier to compare accuracy across different distances and luminaire models.



(a) Artichoke (b) Statler

Figure 5.8: 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.

Figure 5.8 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. Using a drastically reduced number of APLs in the presence of detailed spatial variations introduces aliasing artifacts, causing the quality to be worse than that of the single point far-field representation (Figure 5.8b.) 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.

5.4 Precomputation Performance

As illustrated in the previous three sections of this chapter, we have been able to prove that our precomputation methods can generate computer images showing the plausible appearance of very complex luminaires. More importantly, the results indicate that the near-field illumination is qualitatively and quantitatively very similar to the best simulation results possible using the reference, unbiased bidirectional path tracing method for the seven examples we chose. We also illustrated an error metric for the irradiance provided by the luminaire. The approaches and algorithms which we demonstrated provide roughly between two and three orders of magnitude faster computation times when using any of the currently existing rendering algorithms. The question arises, "How can these techniques then be used by manufacturers and illumination engineers and thus allow real-time experimentation in the selection of lighting fixtures for unbuilt or re-designed environments?" Although this will be described in the next chapter,

Table 5.1: Performance data for precomputation (in minutes.)

	Luminaire						
Phase	Troffer	Artichoke	Silvia	Statler	Sputnik	V&A	Chihuly
Particle trace	8.9	34.5	30.2	180.5	23.5	25.0	107.0
APLs gen.	2.5	2.7	2.5	2.6	2.6	2.4	2.9
Radiance vol.	4.2	4.7	4.5	1.3	3.5	2.8	1.7
Total time	15.6	41.9	37.2	184.4	29.6	30.2	111.6

it is worthwhile to discuss the limitations of the precomputation methods as well as the requirements of their compute environments to justify their usage.

Table 5.1 and Figure 5.9 show 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. We found through our experiments that this amount of particles was sufficient to achieve high quality illumination; using more particles did now show an increase in quality commensurable to the increased computation time. Our most geometrically complex luminaire, Statler, with three million triangles, took just over three hours for particle tracing (geometry intersections dominate the cost.)

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 (i.e., 32^3) volume grid was sufficient even for complex luminaires such as Statler and the V&A Chandelier. Table 5.2 contains the actual size of the radiance volume for each of our luminaires, and the average number of voxels to which a single particle contributes. The latter value is correlated with the com-

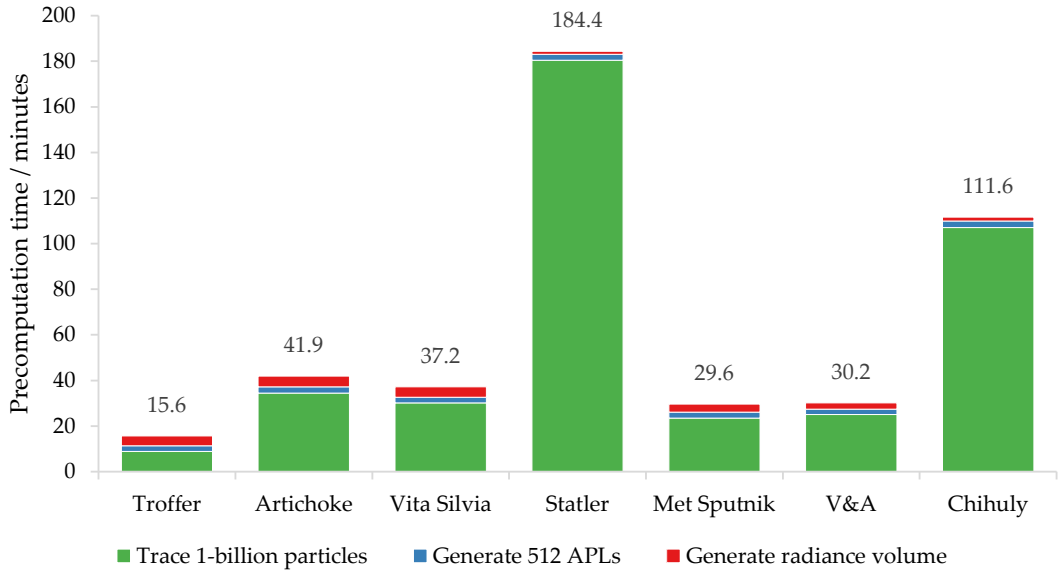


Figure 5.9: Performance data for precomputation.

putation time for the radiance volume from Table 5.1. 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.

Although there are many options which should be available for the users' discretion, we have chosen the use of 512 APLs for illustration through our results. As seen in Figure 5.8, it is obvious that using 512 APLs or more, as indicated by the "hockey stick" error graphs for the P.H. Artichoke and Statler examples, reveal that we can provide accurate, near-field illumination with this amount. At this level of simulation, the memory requirements are such that they usually require less than 400 MiB for each luminaire model used at rendering time (see Table 5.3.) Since most workstations at engineering and design offices now contain memory far in excess of this, it seems quite reasonable that manufacturers could take the responsibility of providing the precomputation results us-

Table 5.2: Radiance volume properties per luminaire.

Luminaire	Volume grid resolution	Ratio of target 32^2 voxels	Avg. voxels per particle
Troffer	$84 \times 10 \times 42$	1.077	7.32
P.H. Artichoke	$32 \times 33 \times 32$	1.031	10.33
Vita Silvia	$33 \times 30 \times 35$	1.057	10.99
Statler	$48 \times 15 \times 48$	1.055	2.97
Met Sputnik	$37 \times 26 \times 37$	1.086	6.48
V&A Chandelier	$25 \times 56 \times 25$	1.068	5.58
Chihuly	$23 \times 67 \times 23$	1.082	4.14

ing our methods, just as they do today with the far-field goniometric measured data. This way when practitioners are evaluating the suitability of a luminaire for their projects they would be able to download the precomputed data and use it in their renderings, showing high-quality illumination and appearance of the luminaires, a process which is unfeasible using current rendering techniques.

The memory consumption of our method for different luminaires is roughly constant, as shown in Table 5.3. 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 384 MiB; standard block texture compression formats could further reduce the memory footprint to 64 MiB [59]. For a radiance volume with 32^3 voxels, using 16 spherical harmonic coefficients (stored in Float32 format), the total memory cost is just 6 MiB. As discussed in Section 4.6 the actual number of voxels we used depends on the aspect ratio of the luminaire's bounding box (see Tables 5.2 and 5.3.) The storage requirements are low enough to allow multiple luminaire models to be easily loaded into local memory simultaneously, allowing practitioners the capability

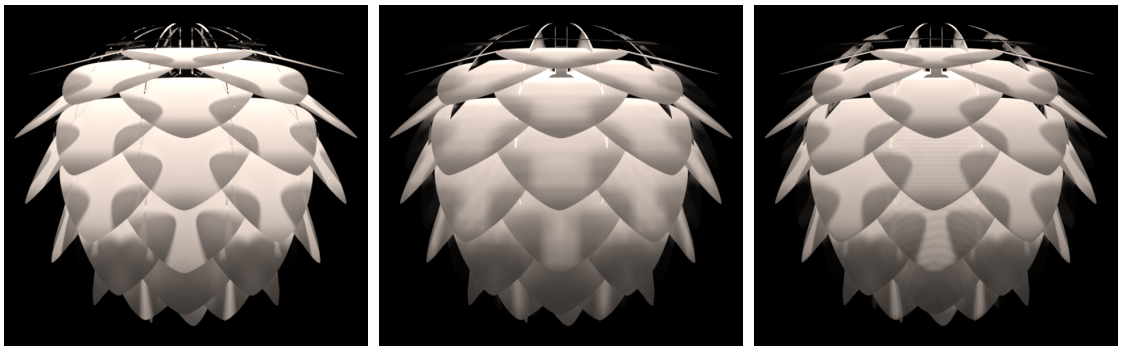
Table 5.3: Memory requirements for the precomputed data (in MiB.)

	Luminaire						
	Troffer	Artichoke	Silvia	Statler	Sputnik	V&A	Chihuly
512 APLs				384.000			
Radiance vol.	6.460	6.188	6.345	6.328	6.517	6.409	6.490

to simulate scenes with many complex luminaires not only in final renderings but also during the design phase.

When generating the luminaire appearance, our method currently requires one to manually choose the maximum number of bounces $\mathbf{n}_b$ in order to separate "sparkles" and "glow" for each luminaire. An automated way to determine this parameter or a more adaptive way to separate "sparkle" and "glow" ray paths is left as future work. Figure 5.10 further illustrates limitations of our method for rendering the luminaire appearance, using the Vita Silvia fixture as an example. Using the baseline radiance volume, containing $33 \times 30 \times 35$ voxels (Figure 5.10b), our method cannot faithfully resolve high-frequency features such as the edges of each shade. Increasing the volume resolution to $131 \times 118 \times 137$ (i.e., slightly over 61 times more data) generates a substantially improved result (Figure 5.10c) while maintaining the same rendering time. However, due to the extremely small number of bounces used in this luminaire ($\mathbf{n}_b = 2$), the structure of the volume grid is still apparent in spite of its higher resolution and is the cause of the banding artifacts seen of the front shades. This issue does not affect other luminaires which query the radiance volume after larger numbers of bounces.

As we have mentioned before, most of our results use 512 APLs for each lu-



(a) Reference (6.7 h)

(b) Ours with 2 bounces 1× appearance data (2.4 min, 169× speedup)

(c) Ours with 2 bounces 61× appearance data (2.4 min, 169× speedup)

Figure 5.10: Limitations of the appearance rendering method. Using the Vita Silvia luminaire as an example (a), the low-resolution radiance volume cannot resolve high-frequency details such as the contours of occluded shades (b). Using a higher-resolution, more data-intensive radiance volume generates a better result, however due to the small number of bounces the underlying grid structure is still apparent (c).

minaire 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 we will discuss in Chapter 6, in many cases an APL-based light hierarchy with much a smaller number of APLs would have been sufficient.

CHAPTER 6

EXTENSIONS AND APPLICATIONS

After having established the accuracy and visual quality of our method for representing the illumination and appearance of luminaires, in this chapter we show how to build upon those foundations and improve the efficiency and applicability of our method. The straightforward strategy we present in Section 6.1 provides a way to reduce the computational cost associated with using our APLs to simulate the luminaire's illumination while preserving its quality. Afterward we demonstrate how to incorporate our techniques into unmodified existing rendering systems, thereby decoupling the complexity of light transport within a luminaire from the rest of the scene. This enables users to render complex scenes with full global illumination being lit by luminaires far more complicated than was previously feasible.

6.1 Hierarchical APL Selection

The error measurements in Figure 5.8 demonstrate it is not always necessary to use numerous APLs to achieve good illumination accuracy. The "point light" nature of the APLs makes them ideally suited for scalable many lights algorithms [7]. 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 (see Figure 6.1.)

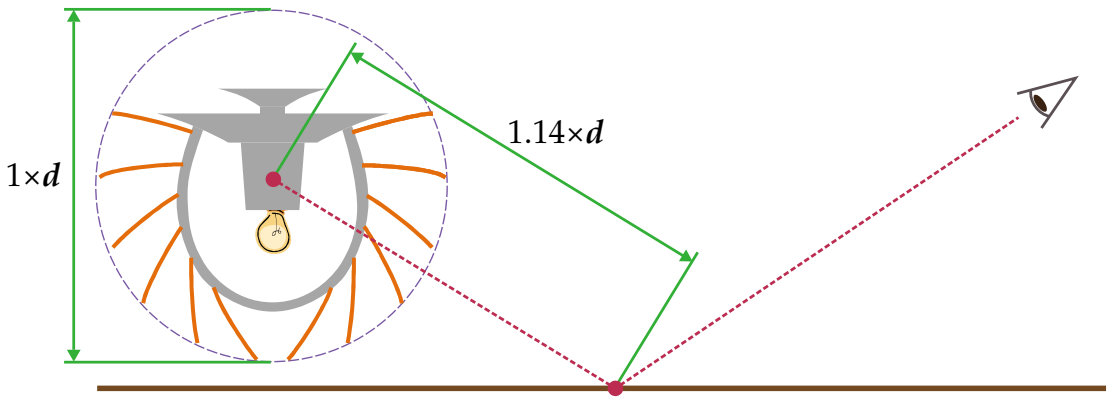


Figure 6.1: Relative distance d used in Equation 6.1. The value of d is the distance from the gathering point to the center of the luminaire's bounding sphere expressed as a ratio of said sphere's diameter.

The heuristic is based on linear regression for achieving a 4 % relative error for the illumination from the Artichoke luminaire[1] (Figure 5.8a) and returns the desired number of APLs, $f(d)$, for a given relative distance d:

$$f(d) = 630.31 \exp(-0.333\ d) \tag{6.1}$$

For better computational efficiency we use a $(2,3)$ rational polynomial approximation to Equation 6.1, calculated via the Remez method as implemented by the Mathematica software package in the `MiniMaxApproximation` function [60]:

$$f'(d) = \begin{cases} \dfrac{104741.36 + d\,(-8309.86309 + 172.5216419\ d)}{155.5587609 + d\,(55.55720 + d\,(0.20698246 + d))} & \text{if } d \leq 20 \\ 1 & \text{if } d > 20 \end{cases} \tag{6.2}$$

Note that during rendering each gathering point randomly chooses between the two APL levels closest to one requested by the heuristic and then evaluates the contribution of all its APLs. Such a process is analogous to separating the

[1]We used the error plot for the P.H. Artichoke luminaire as the basis for deriving the heuristic because it is the exemplar with the largest error ratios among our examples. We found through testing that targeting a 4 % error for the illumination due to each level of the hierarchy did not produce visible error artifacts while yielding considerable performance improvements.

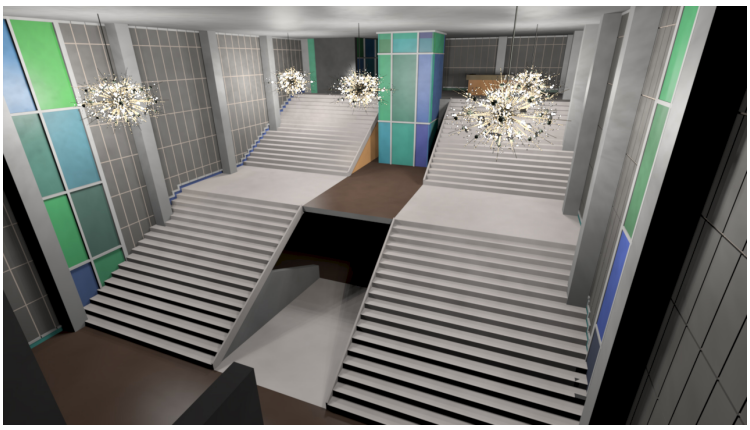
evaluation of a trilinear texture query by evaluating only a single MIPMAP level at each pixel sample. This randomized approach is simply a naïve version of the sophisticated Lightcuts method [55], where the expected value of the number of APLs to evaluate (i.e., the "cut size" metric from Lightcuts) matches the recommendation from the heuristic. Results in Figure 6.2 show that the illumination using this simple hierarchy and the heuristic function provide considerably better performance while preserving the image quality.

One limitation of this strategy is the quality of the shadows. The metric assumes that all the light from the APLs in the hierarchy reaches the gathering point. When the luminaire is partially blocked by occluders, the chosen number of APLs might not be enough to adequately approximate the penumbra region.

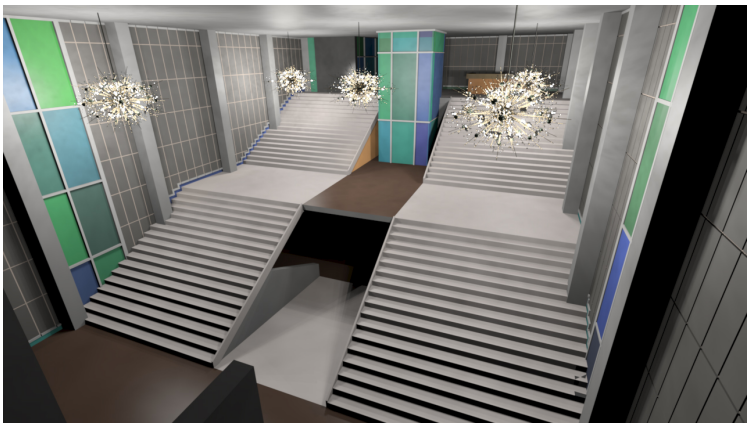
6.2 Integration with Global Illumination Methods

Predictive rendering software is a particularly complex class of software. By providing implementations of the formal abstractions used by a rendering system it is possible to enhance it with new capabilities *a posteriori* without requiring any changes to its core components. As a proof of concept we extended the Mitsuba rendering system [21] by incorporating our APLs as new class of light emitters.

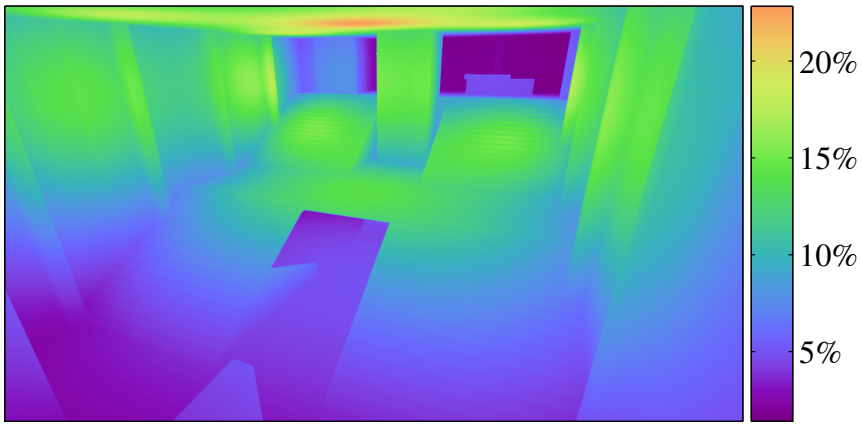
In addition to implementing the direct light query (i.e., the radiance emitted by an APL at a given direction, as used by the results in Section 5.2) we also implemented importance sampling (i.e., selection of an outgoing direction with a probability density which approximates the APL's directional power distribution) and it dual, evaluation of the directional sampling density (i.e., given an



(a) Baseline: all 512 APLs per luminaire



(b) Hierarchical APL selection, 9× speedup



(c) Percentage of baseline APLs used per pixel

Figure 6.2: 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 and 512 APLs based on a simple distance-based heuristic.

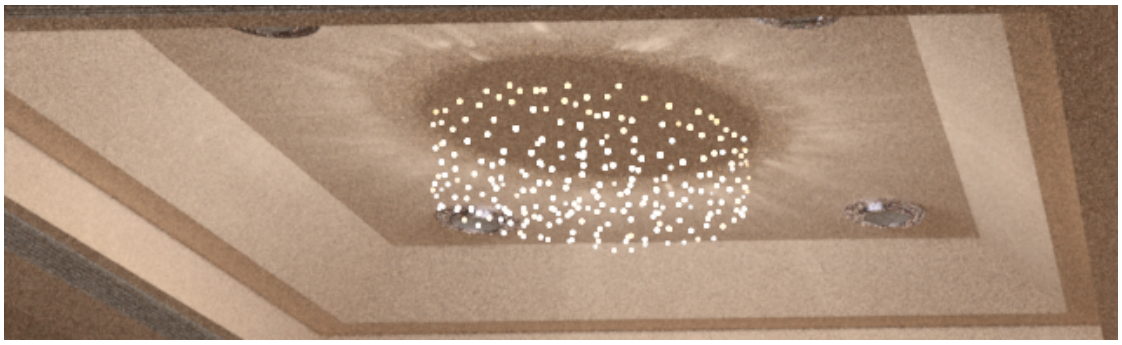
outgoing direction, finding the probability which with the importance sampling routine would have chosen said direction.) We used the simple and efficient method by Pharr and Humphreys [40] for implementing these two functions.

Importance sampling is a crucial component of all Monte Carlo-based algorithms and enables faster convergence rates. It is used extensively by path tracing and particle shooting-based methods such as photon mapping and many-lights techniques. Evaluation of the directional sampling density enables a wider variety of methods such as bidirectional path tracing and Metropolis. Figure 6.3 shows the quality improvements due to importance sampling with respect to uniform sampling (i.e., choosing any emission direction on the sphere with the same probability) using equal computation time.

Moreover, since our APLs are true point sources, they can be used for real-time pre-visualization. In our proof of concept implementation, an OpenGL fragment shader enables to visualize the illumination from our APLs in Mitsuba's unmodified interactive live preview (see Figure 6.4.)

Mitsuba's interactive live preview uses a straightforward implementation of instant radiosity [24]. It converts direct lights and indirect illumination into a set of Virtual Point Lights (VPLs) and then it leverages graphics hardware to accumulate the contribution of each VPL into the framebuffer. Since our APLs are already point sources they do not need further processing by instant radiosity. By converting the illumination due to complex luminaires into direct point lights our method enables much higher quality renderings using many-light methods at the same cost as the low-fidelity traditional impostors, all while featuring effects such as caustics which are unsupported by standard VPL approaches.

(a) Uniform sampling



(b) Importance sampling

Figure 6.3: Section of the Statler Foyer scene (Figure 3.1) illuminated by our APLs and excluding the luminaire appearance, rendered with using 64 samples per pixel. Uniform sampling (a) produces a result with very high variance; importance sampling (b) is a substantial improvement.

Combined with our method for rendering the luminaire appearance, complex scenes with full global illumination can be faithfully rendered as demonstrated in Figures 6.5, 6.6, and 6.7. In these scenes, rendered with bidirectional path tracing, most of the computation time is devoted to solving the light transport within the scene. By decoupling the illumination complexity from the rest of the environment, rendering this class of scenes with full global illumination when using a given number of our APLs has roughly the same computational cost as using the same number of omnidirectional point light sources.

(a) Troffer — direct only (4 s)

(b) Troffer — direct and indirect (20 s)

(c) Artichoke — direct only (4 s)

(d) Artichoke — direct and indirect (20 s)

(e) V&A — direct only (4 s)

(f) V&A — direct and indirect (20 s)

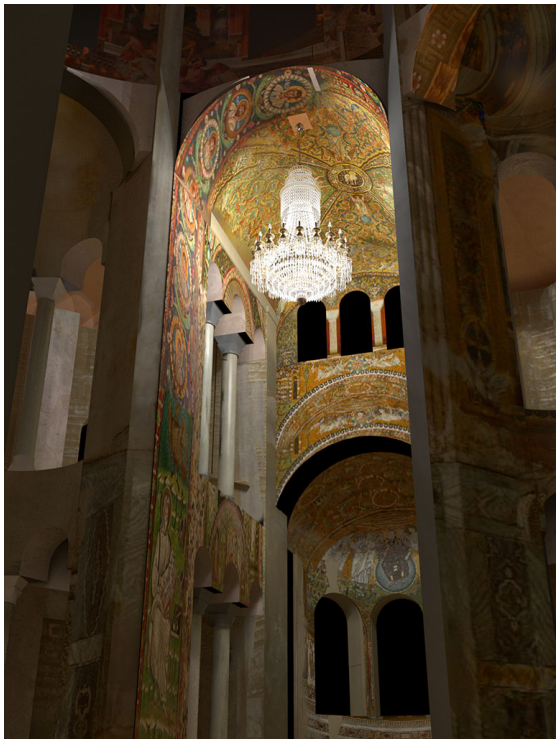
Figure 6.4: Kitchen scene rendered with the instant radiosity method from Mitsuba's interactive live preview. The sole source of illumination are the 512 APLs corresponding to each luminaire. The luminaire's appearance is excluded since these images come from the real-time preview.

(a) Our method — Troffer

(b) Our method — P.H. Artichoke

Figure 6.5: A luminaire can have a substantial effect on the appearance of an interior. Each image shows the same kitchen 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.

(a) Our method — V&A Chandelier, wide angle view



(b) Our method — V&A Chandelier

(c) Detail

Figure 6.6: Rendering of the chancel at the Basilica of San Vitale (Ravenna, Italy) solely illuminated by the V&A Chandelier luminaire using our method, including global illumination. The detail image (c) shows a cropped section of the full view using different tone mapping parameters to better appreciate the appearance of the luminaire.

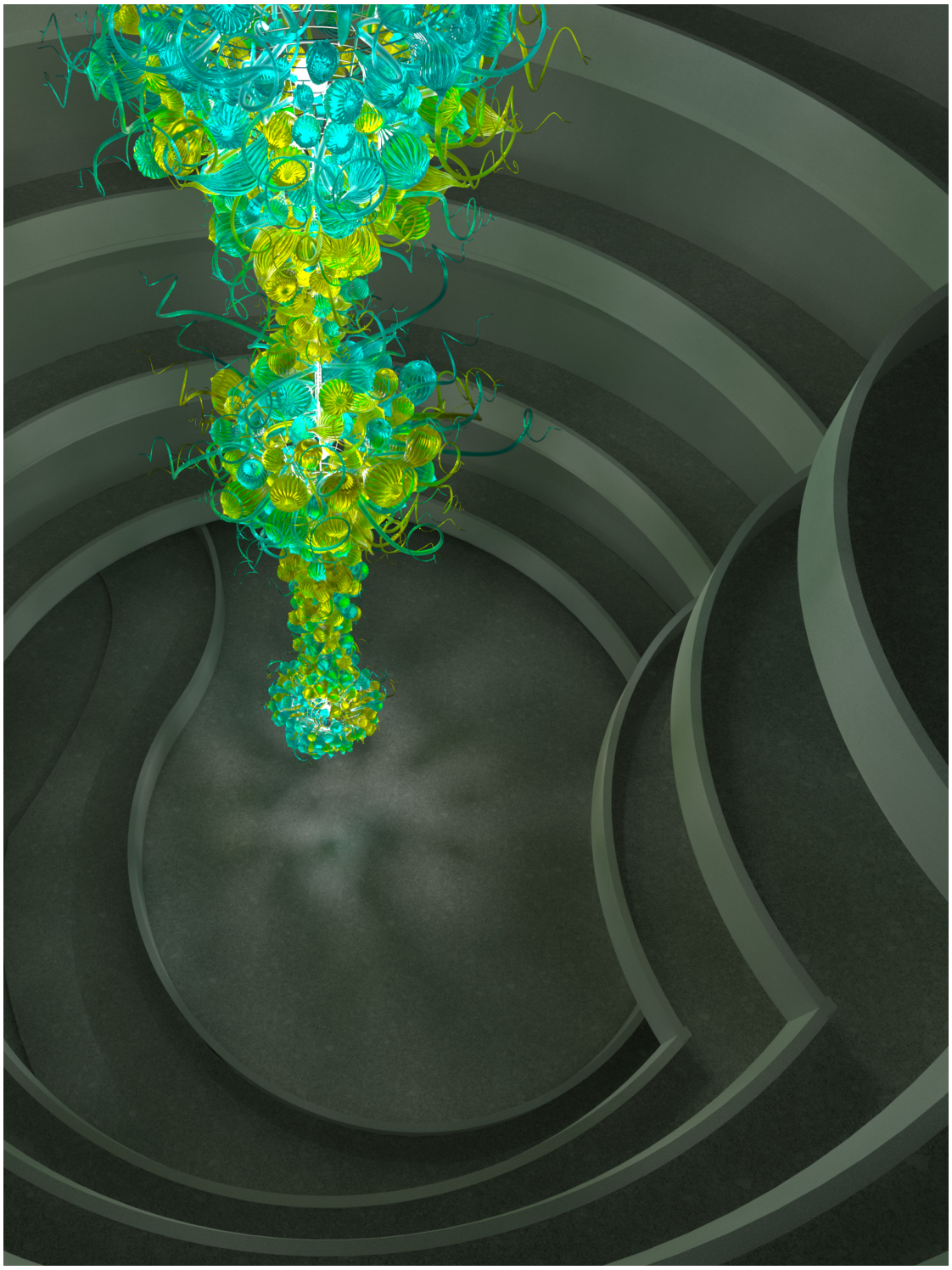


Figure 6.7: Rendering of the Solomon R. Guggenheim Museum rotunda in New York City, solely illuminated by the Chihuly luminaire using our method, including global illumination.

6.3 Applicability to Lighting Design Workflows

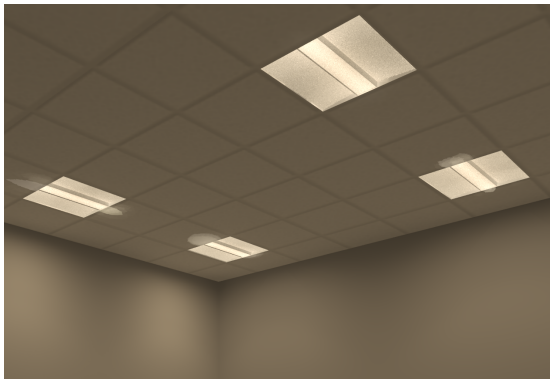
Furthermore, our luminaire rendering method can be applied for interior lighting design. As shown in Figure 6.5, with our method, high-quality interior lighting can be quickly simulated and changed so as to identify the best lighting conditions to achieve a seamless combination of functionality and style. Since the data structures used by our method (APLs and radiance volume) are generated only once for each luminaire model, they could be provided directly by the manufacturers. This way practitioners could download these structures from the manufacture's website as they do with the single point far-field data today.

Modern Building Information Modeling (BIM) systems, such as Autodesk Revit, feature a deep integration with the product catalogs of luminaire manufacturers. They provide not only single point far-field data, but also the geometry of the luminaire. They allow users to simply drag-and-drop a luminaire selected from a catalog into an adequate location for placement such as walls or ceilings. However during rendering these representations can neither generate a faithful representation of the luminaire nor accurate illumination in both the near- and far-fields. Figure 6.8 shows an example of the suboptimal results obtained by directly using data provided by manufacturer to BIM systems.

Since simulating the full light transport within a luminaire using existing techniques is prohibitively expensive, the renderers used by BIM packages ignore the geometry of the luminaire (i.e., shadow rays from other elements of the scene do not intersect the luminaire's geometry.) Thus the illumination from a luminaire comes exclusively from limited sources such as measured far-field

(a) Photograph of the luminaires from the marketing brochure.[2]

(b) Rendering using data as provided by the manufacturer.[3]



(c) Rendering after manual modifications to the luminaires.[4]

Figure 6.8: State-of-the-art of luminaire in rendering within BIM systems using the typical data provided by manufacturers: single point far-field data and luminaire geometry using simplistic material models.

[2]Dave Baldacchino. "Manufacturer Content — Lighting". *Do U Revit?*. March 23 2014.
[3]Ibid.
[4]Ibid.

data, possibly distributed over a limited aperture to get soft-shadows, or impostors such as a few omnidirectional point lights. By ignoring the interaction of light with the luminaire itself, these representations fail to represent the luminaire's appearance (Figure 6.8b.) Users have then to manually modify the manufacturer's data to improve the quality of the images as to make them suitable for presentations with clients (Figure 6.8c.) Such modifications require expert knowledge, and not only yield compromised representations of the luminaire's appearance, but also by altering the luminaire's definition users inadvertently alter their illumination characteristics. The resulting illumination is less accurate than that provided originally by the single point far-field data.

By incorporating our data structures into the data accessible to BIM software, users would be able to get the accurate illumination and plausible appearance provided by our method without any modifications to their existing design work-flow. Using our precomputed data would allow them to get presentation-quality results without required cumbersome manual modifications. The memory requirements for each luminaire using our method (less than 400 MiB in uncompressed form as we discussed in Section 5.4) are not burdensome for today's workstations: 16 GiB of main system memory are handily available even on portable computers. Manufactures would run the precomputation (which in our examples took a median of 37.2 min) in tandem with the luminaire design process and integrate the data into the digital catalogs available to the end users, enabling more accurate, higher quality results than those possible nowadays.

CHAPTER 7

CONCLUSIONS

In this dissertation we present 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 in their earlier design phase, a prospect which becomes feasible because of the way our method decouples lighting intricacies from geometric and materials complexity.

APPENDIX A

SUPPLEMENTAL ILLUMINATION RESULTS

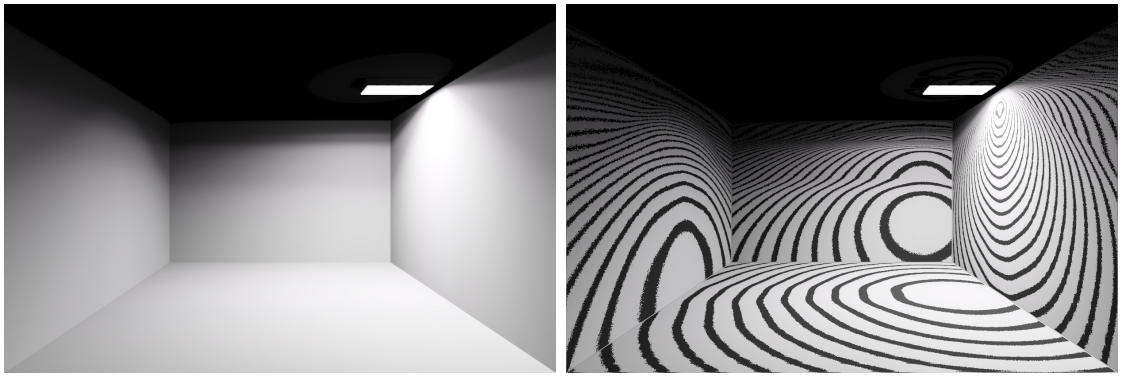
In this appendix we present additional results for the qualitative evaluation of illumination to those presented in Section 5.2. We rendered each of our luminaires inside an empty 3 m × 3 m × 2 m room, placing them 10 cm away from the right wall to observe both near- and far-field effects. The luminaires are scaled so that, in average, their longest dimension is 50 cm long.

Each figure shows three renderings of the direct illumination incident on the scene: a reference using bidirectional path tracing (restricted to only allow paths with at most one non-luminaire vertex), illumination from 512 APLs and the illumination from only one APL (which is equivalent to the single point far-field representation.) The latter two also show the luminaire's appearance as rendered by our method (see Section 4.5.) For each configuration we show the true-color rendering, tone mapped using a global operator, and an enhanced version with overlaying iso-contours equally spaced in the logarithm of the luminance. These iso-contours help to visualize the details and accuracy of the illumination by directly reflecting the luminance distribution on the receiving surfaces.

Note that the iso-contours in our results using 512 APLs closely match those on the reference solution for both the near- and far-field regions, indicating that our method produces a radiometrically accurate solution. The single point far-field representation yields visually obvious illumination errors. The timings shown indicate the computation time required after the precomputation step has been completed.

Table A.1: Rendering time, direct illumination on 3 m × 3 m × 2 m room

Luminaire	Reference (h)	512 APLs Time (min)	512 APLs Speedup	1 APL Time (min)	1 APL Speedup
Troffer	36.41	3.51	622.9×	1.88	1162.7×
P.H. Artichoke	44.45	3.66	727.7×	1.93	1379.5×
Vita Silvia	30.76	3.47	531.6×	1.97	937.0×
Statler	190.96	3.99	2873.5×	2.32	4934.2×
Met Sputnik	47.97	3.83	751.3×	1.93	1491.6×
V&A Chandelier	52.03	3.86	809.6×	1.99	1565.1×
Chihuly	55.37	4.06	817.7×	2.07	1606.1×



(a) Reference (36.41 h)

(b) Ours with 512 APLs (3.51 min, 622.9× speedup)

(c) Ours with one APL (1.88 min, 1162.7× speedup)

Figure A.1: Illumination patterns for the Troffer luminaire.

(a) Reference (44.45 h)

(b) Ours with 512 APLs (3.66 min, 727.7× speedup)

(c) Ours with one APL (1.93 min, 1379.5× speedup)

Figure A.2: Illumination patterns for the P.H. Artichoke luminaire.

(a) Reference (30.76 h)

(b) Ours with 512 APLs (3.47 min, 531.6× speedup)

(c) Ours with one APL (1.97 min, 937.0× speedup)

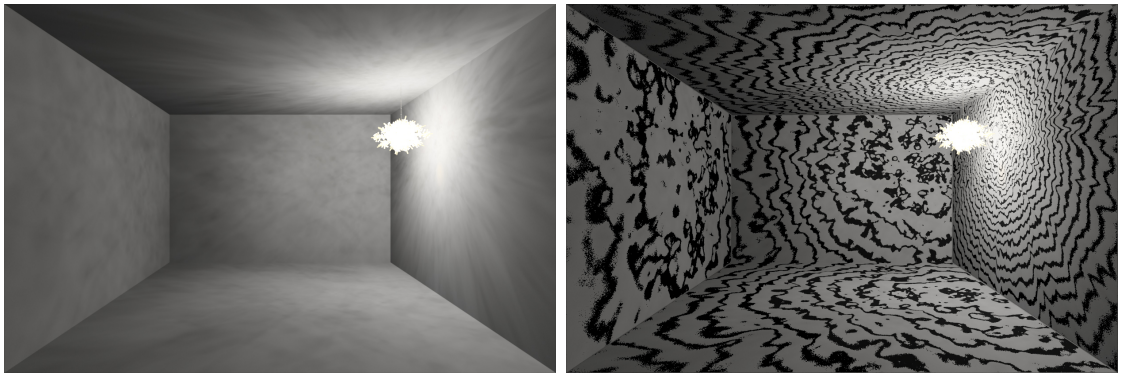
Figure A.3: Illumination patterns for the Vita Silvia luminaire.

(a) Reference (190.96 h)

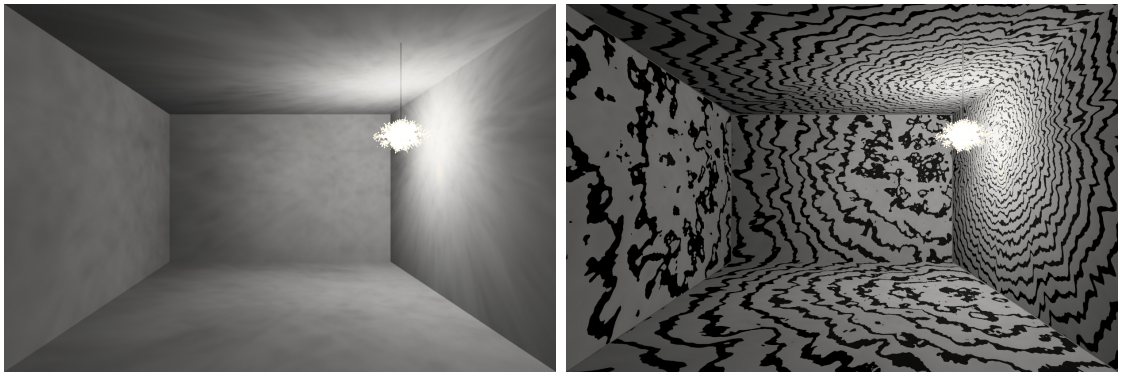
(b) Ours with 512 APLs (3.99 min, 2873.5× speedup)

(c) Ours with one APL (2.32 min, 4934.2× speedup)

Figure A.4: Illumination patterns for the Statler luminaire.



(a) Reference (47.97 h)



(b) Ours with 512 APLs (3.83 min, 751.3× speedup)

(c) Ours with one APL (1.93 min, 1491.6× speedup)

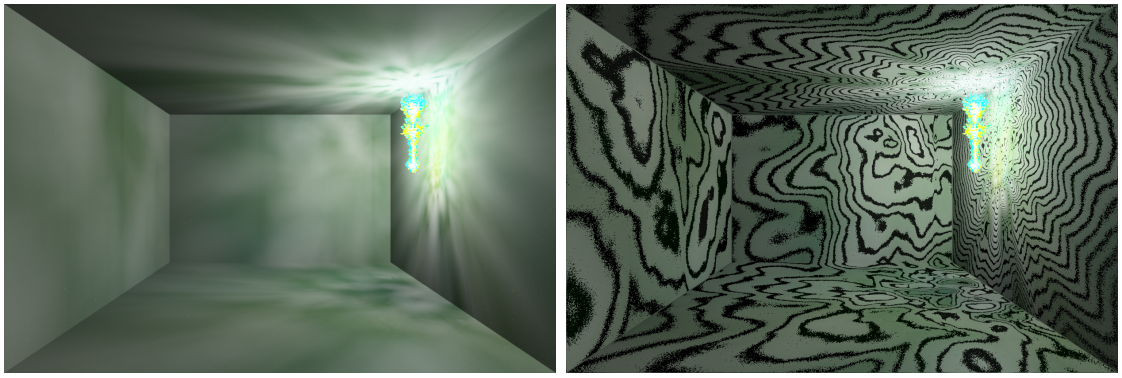
Figure A.5: Illumination patterns for the Met Sputnik luminaire.

(a) Reference (52.03 h)

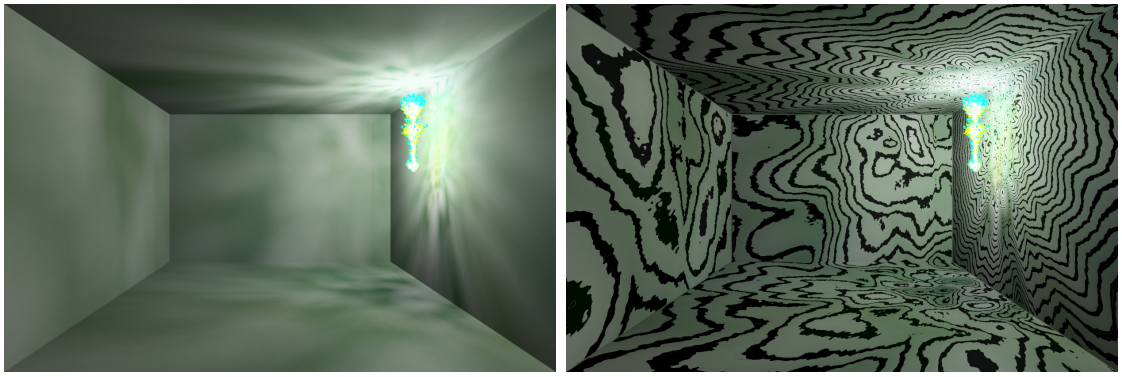
(b) Ours with 512 APLs (3.86 min, 809.6× speedup)

(c) Ours with one APL (1.99 min, 1565.1× speedup)

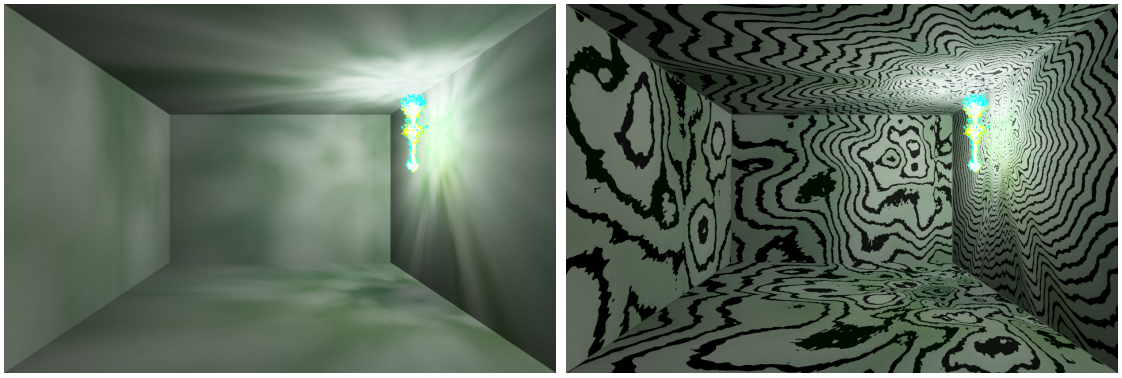
Figure A.6: Illumination patterns for the V&A Chandelier luminaire.



(a) Reference (55.37 h)



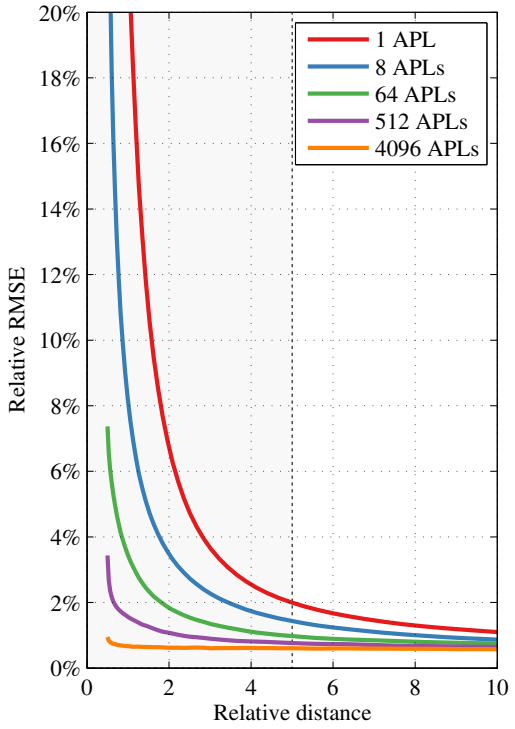
(b) Ours with 512 APLs (4.06 min, 817.7× speedup)



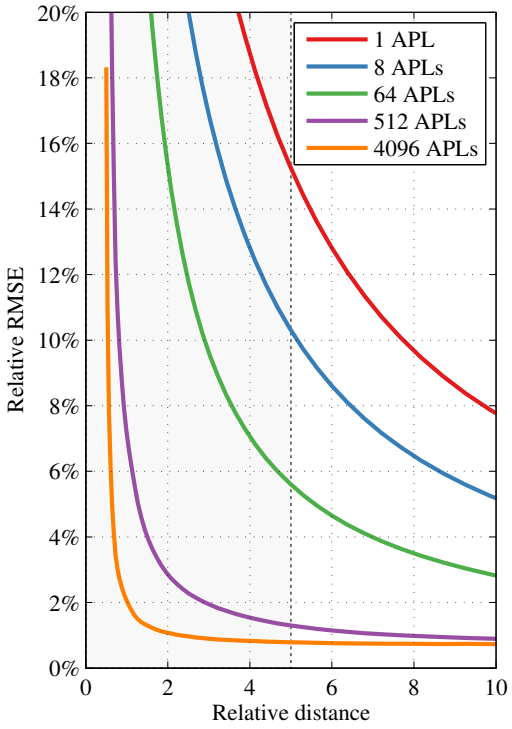
(c) Ours with one APL (2.07 min, 1606.1× speedup)

Figure A.7: Illumination patterns for the Chihuly luminaire.

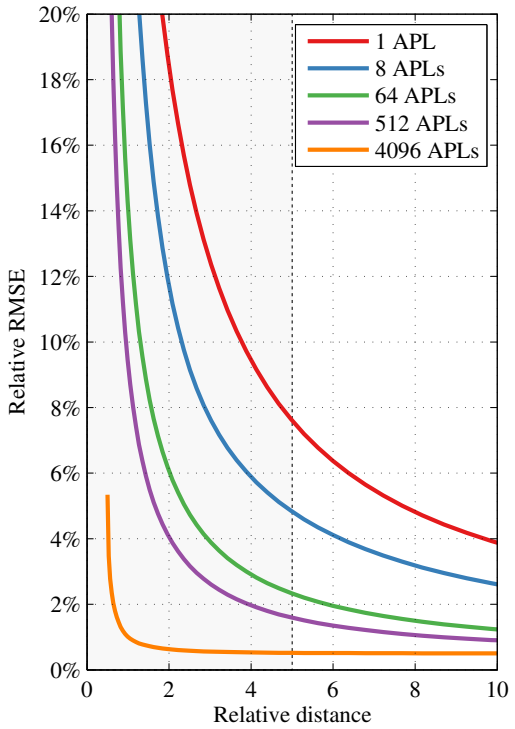
Figure A.8: Relative RMSE at different distances using incremental numbers of APLs. As shown in the diagrams for all of our luminaires, we found that 512 APLs provide sufficiently accurate near-field illumination (shaded region on the left side of each plot.)



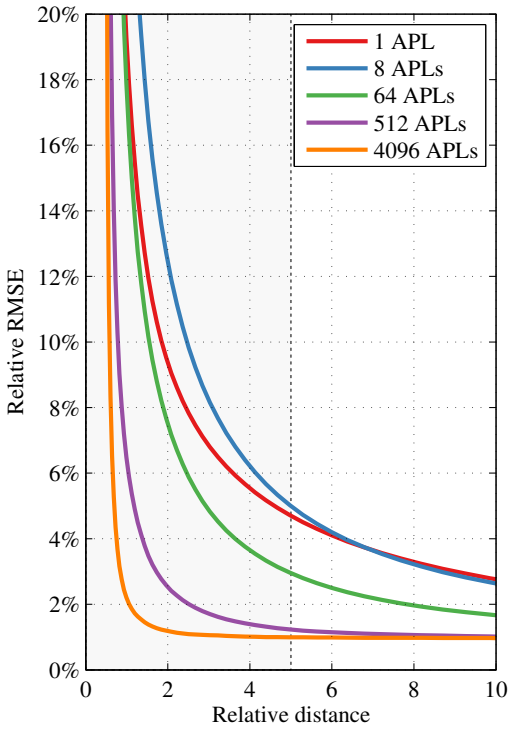
(a) Troffer



(b) Artichoke

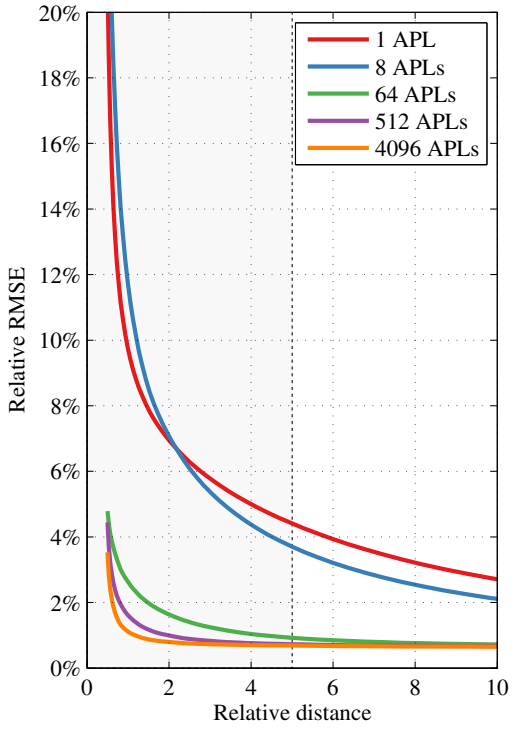


(c) Vita Silvia

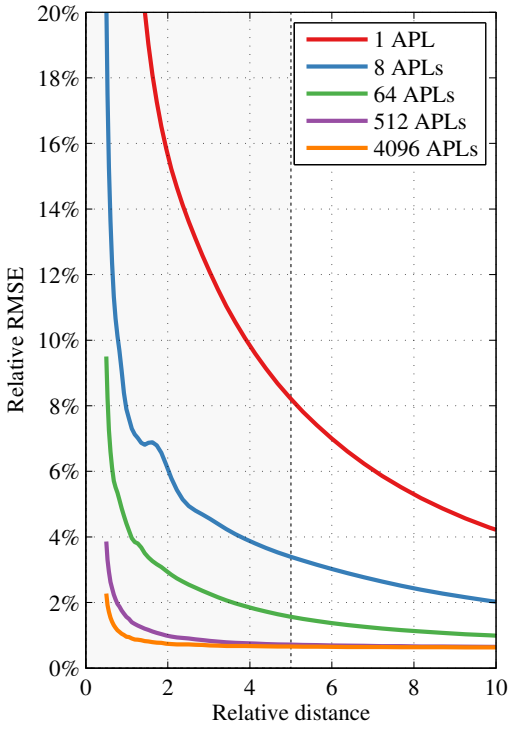


(d) Statler

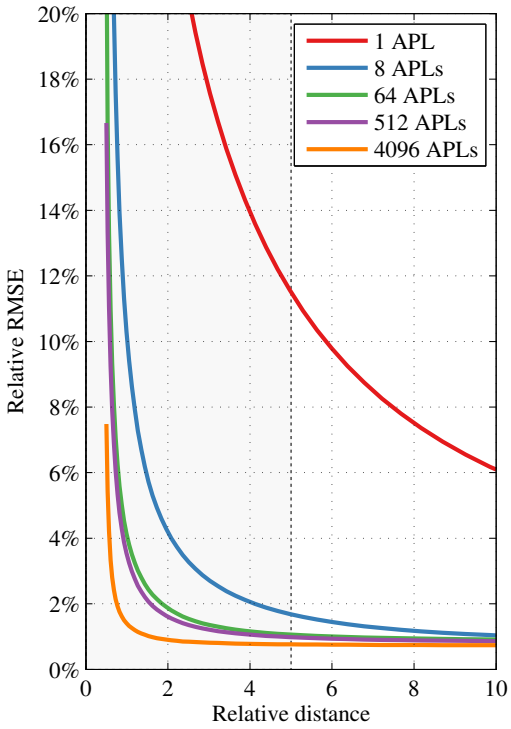
Figure A.8 (Continued)



(e) Met Sputnik



(f) V&A Chandelier



(g) Chihuly

APPENDIX B

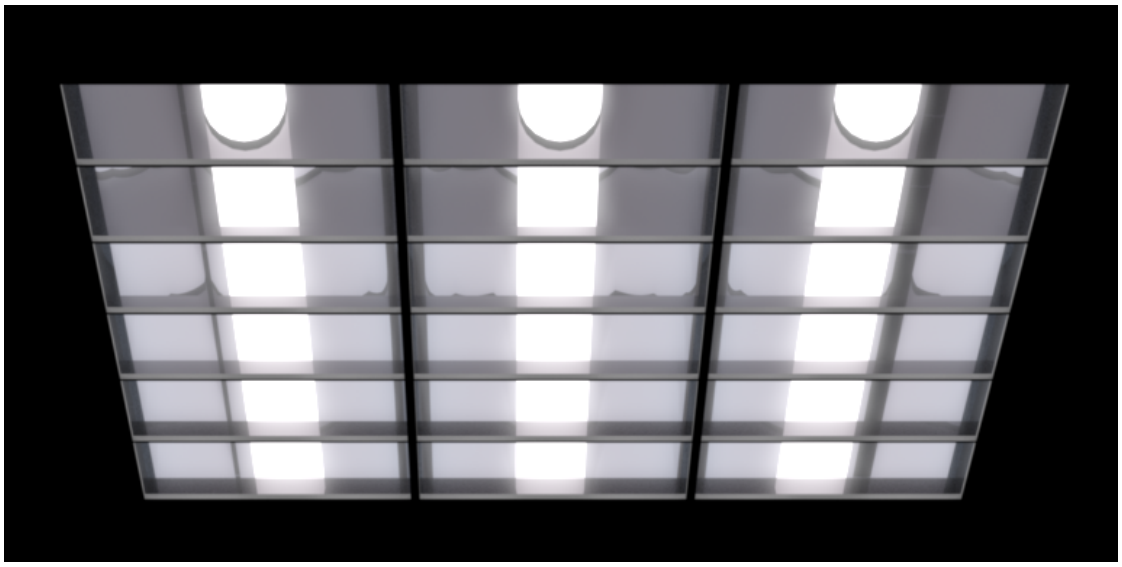
SUPPLEMENTAL APPEARANCE RESULTS

This appendix contains larger versions of the images found in Figure 5.2, which compare our method for rendering visually plausible versions of the luminaires appearance, based on limited-depth path tracing and our precomputed radiance volume structure (see Section 4.5), to reference renderings using bidirectional path tracing.

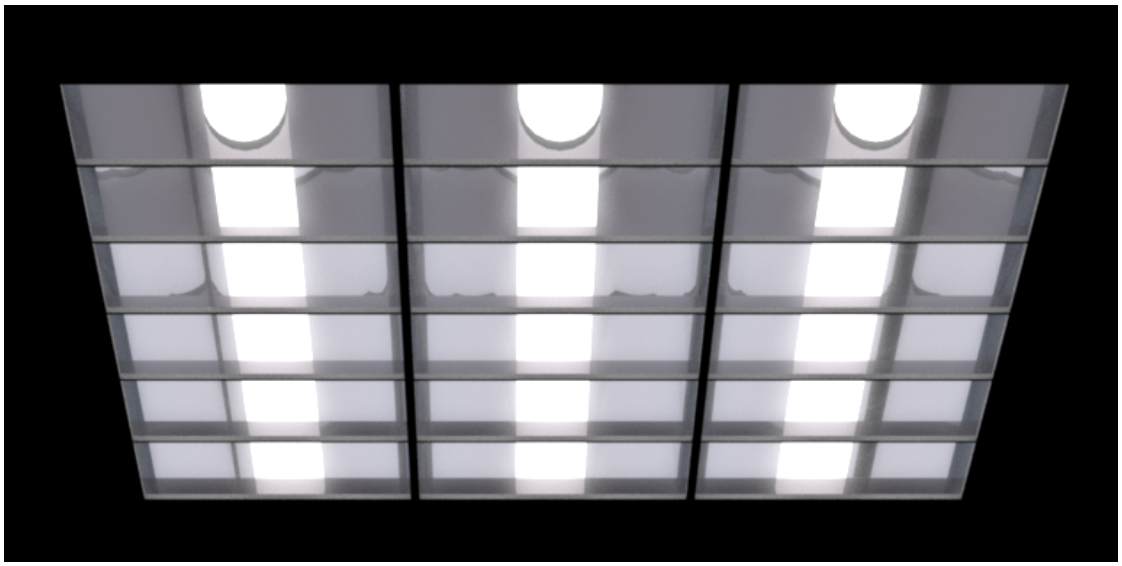
In these enlarged images it is easier to appreciate that, in spite of taking almost two orders magnitude more time to render, the reference images are not fully converged yet, exhibiting visible noise artifacts. The timings shown indicate the computation time required after the precomputation step has been completed.

Table B.1: Rendering time, luminaire appearance

Luminaire	Reference/h	Our method		
		Time/min	Speedup	Bounces
Troffer	1.13	0.48	141.25×	5
P.H. Artichoke	5.65	13.48	25.15×	5
Vita Silvia	6.69	2.38	168.66×	2
Statler	59.86	5.94	604.64×	16
Met Sputnik	10.35	8.11	76.57×	6
V&A Chandelier	10.00	11.86	50.59×	8
Chihuly	42.83	5.93	433.34×	8

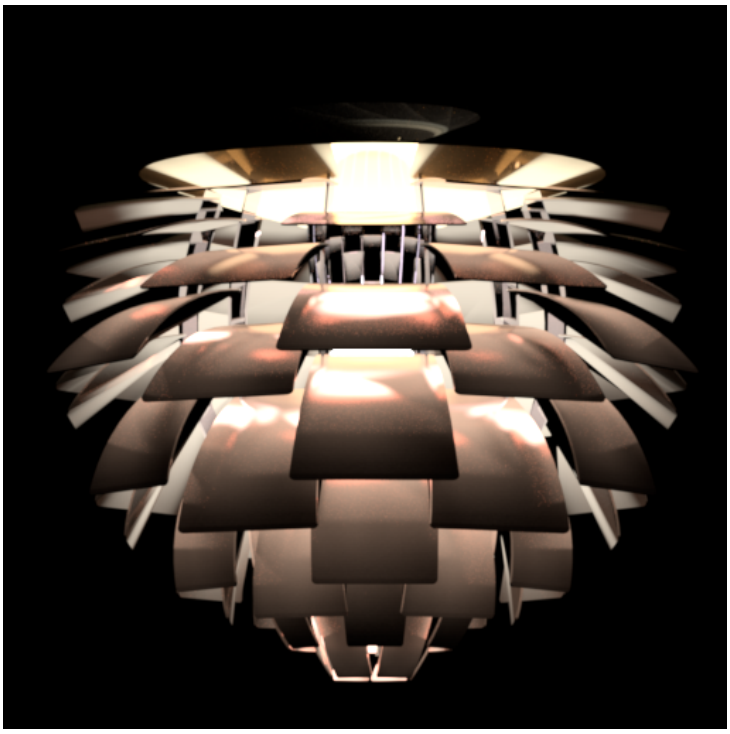


(a) Reference (1.13 h)

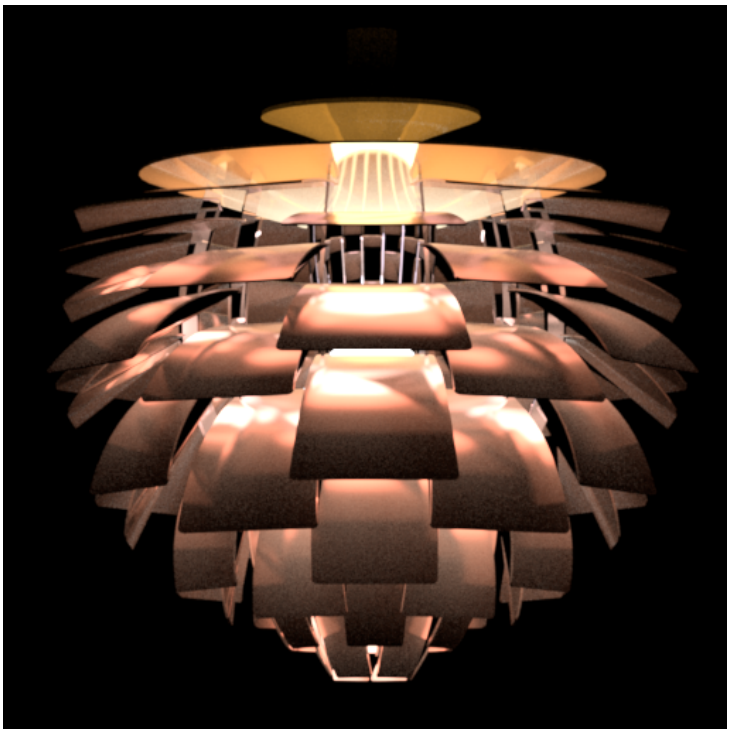


(b) Ours with 5 bounces (0.48 min, 141.25× speedup)

Figure B.1: Appearance comparison for the Troffer luminaire

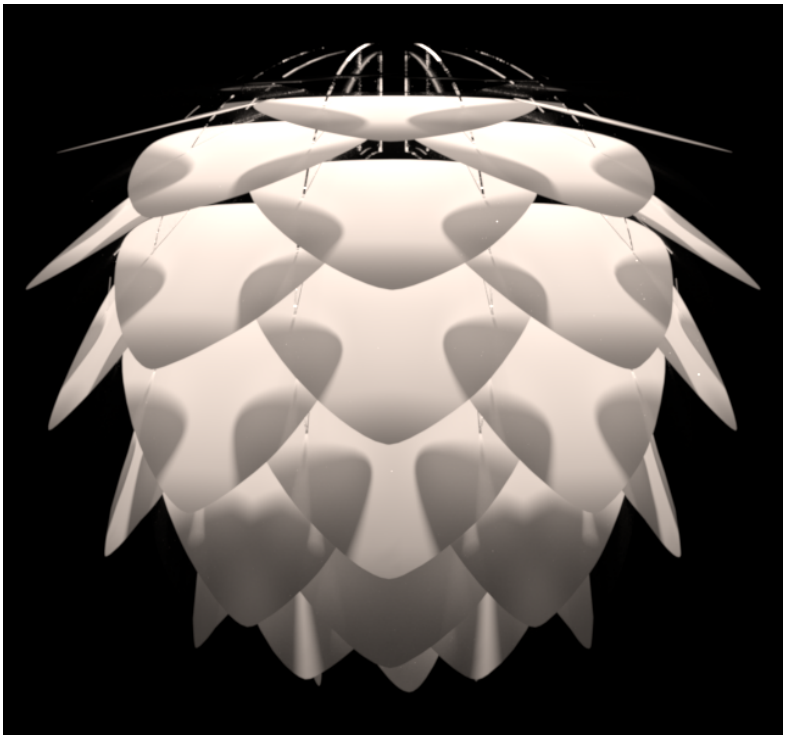


(a) Reference (5.65 h)

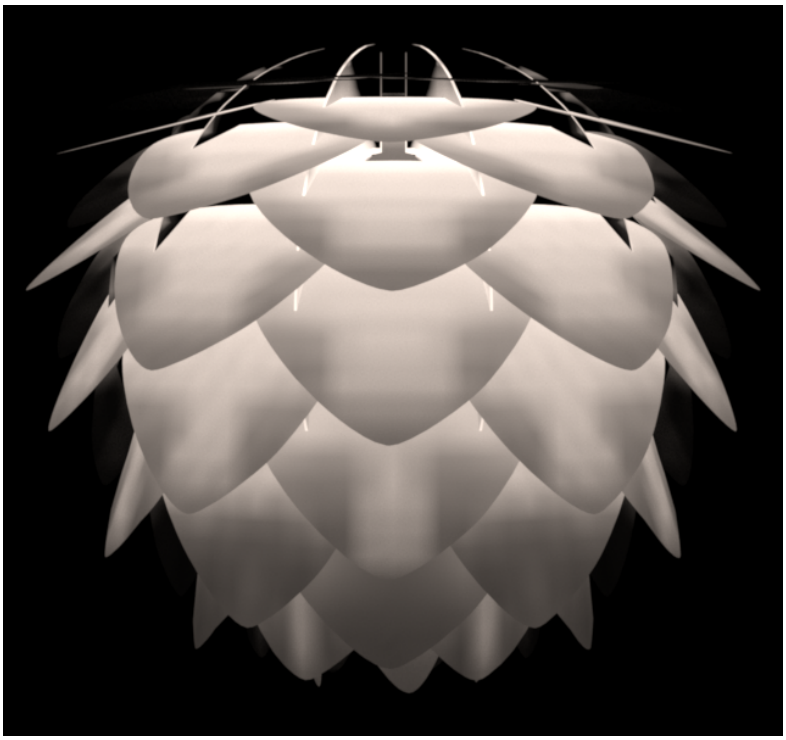


(b) Ours with 2 bounces (13.48 min, 25.15× speedup)

Figure B.2: Appearance comparison for the Artichoke luminaire

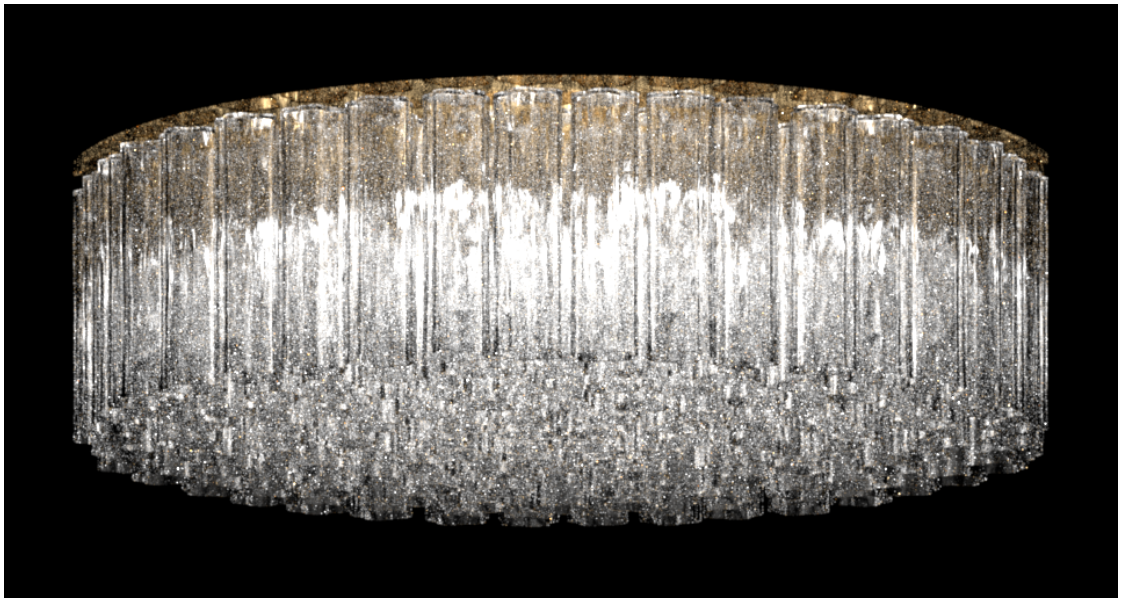


(a) Reference (6.69 h)



(b) Ours with 2 bounces (2.38 min, 168.66× speedup)

Figure B.3: Appearance comparison for the Vita Silvia luminaire

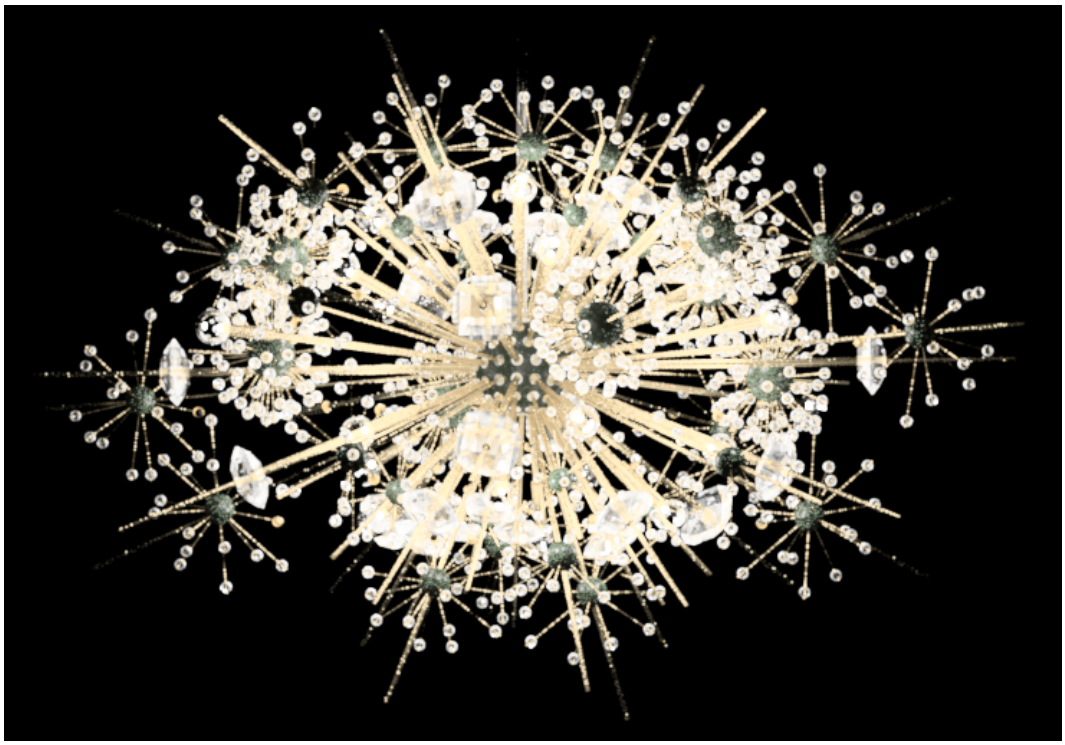


(a) Reference (59.86 h)

(b) Ours with 16 bounces (5.94 min, 604.64× speedup)

Figure B.4: Appearance comparison for the Statler luminaire

(a) Reference (10.35 h)



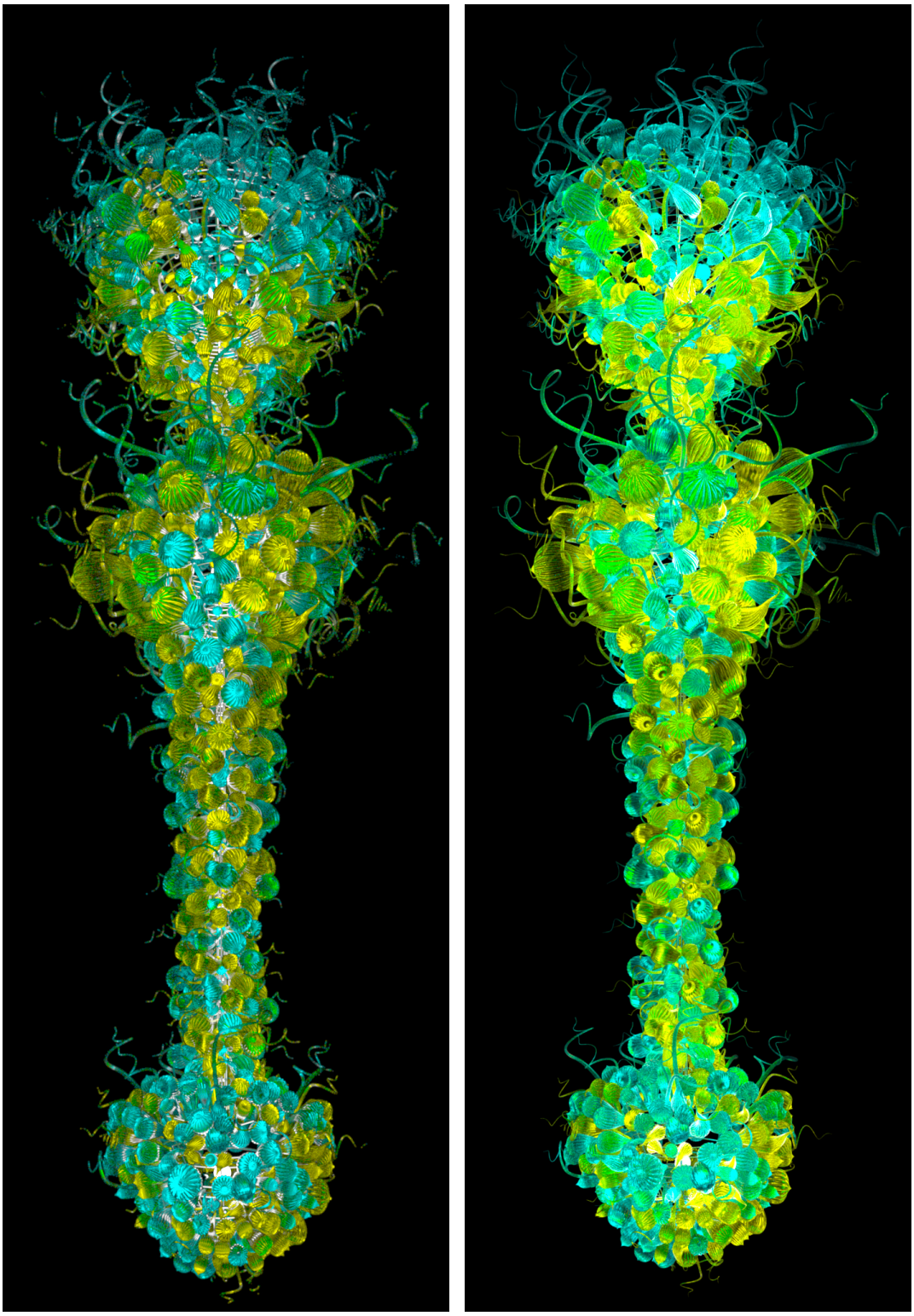
(b) Ours with 6 bounces (8.11 min, 76.57× speedup)

Figure B.5: Appearance comparison for the Met Sputnik luminaire

(a) Reference (10 h)

(b) Ours with 8 bounces
(11.86 min, 50.59× speedup)

Figure B.6: Appearance comparison for the V&A Chandelier luminaire



(a) Reference (42.83 h)

(b) Ours with 8 bounces
(5.93 min, 433.34× speedup)

Figure B.7: Appearance comparison for the Chihuly luminaire

REFERENCES

[1] Ian Ashdown. Near-field photometry: Measuring and modeling complex 3-D light sources. In *ACM SIGGRAPH Course Notes*, pages 1–15, 1995.

[2] Ian Ashdown. Thinking photometrically part II. In *LIGHTFAIR 2001 Pre-Conference Workshop*, pages 1–46, 2001.

[3] Ian Ashdown and Ron Rykowski. Making near-field photometry practical. *Journal of the Illuminating Engineering Society of North America*, 27(1):67–79, 1998. doi: 10.1080/00994480.1998.10748212.

[4] Chuo-Ling Chang, Xiaoging Zhu, Prashant Ramanathan, and Bernd Girod. Inter-view wavelet compression of light fields with disparity-compensated lifting. In *Proc. of VCIP*, pages 694–706. SPIE, 2003. doi: 10.1117/12.509883.

[5] Chuo-Ling Chang, Xiaoqing Zhu, Prashant Ramanathan, and Bernd Girod. Light field compression using disparity-compensated lifting and shape adaptation. *IEEE Transactions on Image Processing*, 15(4):793–806, 2006. doi: 10.1109/TIP.2005.863954.

[6] Wei-Chao Chen, Jean-Yves Bouguet, Michael H. Chu, and Radek Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Trans. Graph.*, 21(3):447–456, 2002. doi: 10.1145/566654.566601.

[7] Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. Scalable realistic rendering with many-light methods. In *EG 2013 — State of the Art Reports*, pages 23–38, Girona, Spain, 2012. Eurographics Association. doi: 10.2312/conf/EG2013/stars/023-038.

[8] Thomas Driemeyer. *Rendering with mental ray®*, volume 1 of *mentay ray® Handbooks*. Springer, third edition, November 2008. ISBN 978-3211228753.

[9] Philip Dutré, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination*. A K Peters, Ltd., Wellesley, Massachusetts, second edition, 2006. ISBN 978-1-56881-307-3.

[10] Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.*, 31(6):192:1–192:10, 2012. doi: 10.1145/2366145.2366211.

[11] Alexander Gershun. The light field. *Journal of Mathematics and Physics*, 18: 51–151, 1939.

[12] Michael Goesele, Xavier Granier, Wolfgang Heidrich, and Hans-Peter Seidel. Accurate light source acquisition and rendering. *ACM Trans. Graph.*, 22 (3):621–630, 2003. doi: 10.1145/882262.882316.

[13] Michael Gold. ARB texture cube map, December 1999. URL `http://www.opengl.org/registry/specs/ARB/texture_cube_map.txt`.

[14] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proc. of SIGGRAPH*, pages 43–54, 1996. doi: 10.1145/237170.237200.

[15] Xavier Granier, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Interactive visualization of complex real-world light sources. In *Proc. of Pacific Graphics*, pages 59–66, 2003. doi: 10.1109/pccga.2003.1238247.

[16] Gene Greger, Peter Shirley, Philip M. Hubbard, and Donald P. Greenberg. The Irradiance Volume. *IEEE CG&A*, 18(2):32–43, 1998. doi: 10.1109/38.656788.

[17] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Trans. Graph.*, 28(5):141:1–141:8, December 2009. doi: 10.1145/1618452.1618487.

[18] Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A path space extension for robust light transport simulation. *ACM Trans. Graph.*, 31(6):191:1–191:10, 2012. doi: 10.1145/2366145.2366210.

[19] Miloš Hašan, Edgar Velázquez-Armendáriz, Fabio Pellacini, and Kavita Bala. Tensor clustering for rendering many-light animations. *Computer Graphics Forum (CGF)*, 27(4):1105–1114, June 2008. doi: 10.1111/j.1467-8659.2008.01248.x.

[20] Wolfgang Heidrich, Jan Kautz, Philipp Slusallek, and Hans-Peter Seidel. Canned lightsources. In *Rendering techniques' 98*, pages 293–300, 1998. doi: 10.1007/978-3-7091-6453-2_27.

[21] Wenzel Jakob. Mitsuba physically based renderer, 2010. URL `http://www.mitsuba-renderer.org`.

[22] Wenzel Jakob and Steve Marschner. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics*, 31(4):58:1–58:13, July 2012. doi: 10.1145/2185520.2185554.

[23] Anton S. Kaplanyan and Carsten Dachsbacher. Path space regularization for holistic and robust light transport. *Computer Graphics Forum*, 32(2):63–72, May 2013. doi: 10.1111/cgf.12026.

[24] Alexander Keller. Instant radiosity. In *Proc. of SIGGRAPH '97*, pages 49–56, 1997. doi: 10.1145/258734.258769.

[25] S. Kniep, S. Häring, and M. Magnor. Efficient and accurate rendering of complex light sources. *Computer Graphics Forum*, 28(4):1073–1081, June 2009. doi: 10.1111/j.1467-8659.2009.01484.x.

[26] Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In *Proc. of Computer Graphics*, volume 93, pages 145–153, 1993.

[27] Greg Ward Larson and Rob Shakespeare. *Rendering With Radiance: The Art And Science Of Lighting Visualization*. Booksurge Llc, April 2004.

[28] Christian Lessig and Eugene Fiume. SOHO: Orthogonal and symmetric Haar wavelets on the sphere. *ACM Trans. Graph.*, 27(1):4:1–4:11, March 2008. doi: 10.1145/1330511.1330515.

[29] Marc Levoy and Pat Hanrahan. Light field rendering. *Proc. of SIGGRAPH*, pages 31–42, 1996. doi: 10.1145/237170.237199.

[30] Thomas Murray MacRobert. *Spherical harmonics; an elementary treatise on harmonic functions, with applications*. Dover Publications, 1948.

[31] Albert Mas, Ignacio Martín, and Gustavo Patow. Compression and importance sampling of near-field light sources. *Computer Graphics Forum*, 27(8): 2013–2027, December 2008. doi: 10.1111/j.1467-8659.2008.01180.x.

[32] Albert Mas, Ignacio Martín, and Gustavo Patow. Fast inverse reflector design (FIRD). *Computer Graphics Forum*, 28(8):2046–2056, December 2009. doi: 10.1111/j.1467-8659.2009.01430.x.

[33] Jonathan T. Moon and Stephen R. Marschner. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph.*, 25(3): 1067–1074, July 2006. doi: 10.1145/1141911.1141995.

[34] Jonathan T. Moon, Bruce Walter, and Stephen R. Marschner. Rendering discrete random media using precomputed scattering solutions. In Jan Kautz and Sumanta Pattanaik, editors, *SR '07 Rendering Techniques*, pages 231–242, Grenoble, France, 2007. Eurographics Association. doi: 10.2312/EGWR/EGSR07/231-242.

[35] Jonathan T. Moon, Bruce Walter, and Steve Marschner. Efficient multiple scattering in hair using spherical harmonics. *ACM Trans. Graph.*, 27(3):31:1–31:7, August 2008. doi: 10.1145/1360612.1360630.

[36] Alexander A. Mury, Sylvia C. Pont, and Jan J. Koenderink. Representing the light field in finite three-dimensional spaces from sparse discrete samples. *Applied optics*, 48(3):450–457, January 2009. doi: 10.1364/AO.48.000450.

[37] Julius Muschaweck. What's in a ray set: moving towards a unified ray set format. In *Illumination Optics II*, volume 8170, 2011. doi: 10.1117/12.896757.

[38] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.*, 22: 376–381, July 2003. doi: 10.1145/882262.882280.

[39] P.Y. Ngai. On near-field photometry. *Journal of the Illuminating Engineering Society*, 16(2):129–136, 1987. doi: 10.1080/00994480.1987.10748693.

[40] Matt Pharr and Greg Humphreys. Infinite area light source with importance sampling. Technical report, Physically Based Rendering, 2004. URL `http://web.archive.org/web/20130413084759/http://www.pbrt.org/plugins/infinitesample.pdf`.

[41] Ravi Ramamoorthi. Precomputation-based rendering. *Foundations and Trends in Computer Graphics and Vision*, 3(4):281–369, April 2009. doi: 10.1561/0600000021.

[42] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proc. of SIGGRAPH*, pages 497–500. ACM, 2001. doi: 10.1145/383259.383317.

[43] Mark. S. Rea, editor. *The IESNA lighting handbook: reference & application*. Illuminating Engineering Society of North America, 9th edition, 2000.

[44] Todd Saemish, P. Ericson, G. Hauser, E. Gibson, R. Heinisch, C. Loch, and IESNA. ANSI/IESNA standard file format for the electronic transfer of photometric data and related information, 2002.

[45] Peter Shirley and Kenneth Chiu. A low distortion map between disk and square. *Journal of Graphics Tools*, 2(3):45–52, January 1997. ISSN 1086-7651. doi: 10.1080/10867651.1997.10487479.

[46] Peter-Pike Sloan. Efficient spherical harmonic evaluation. *Journal of Computer Graphics Techniques (JCGT)*, 2(2):84–83, September 2013. ISSN 2331-7418. URL `http://jcgt.org/published/0002/02/06/`.

[47] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.*, 21:527–536, 2002. doi: 10.1145/566654.566612.

[48] John Snyder and Don Mitchell. Sampling-efficient mapping of spherical images. Technical report, Microsoft Corp., 2001.

[49] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on Graphics*, 25:967–976, July 2006. doi: 10.1145/1141911.1141981.

[50] J. Y. Tsao, H. D. Saunders, J. R. Creighton, M. E. Coltrin, and J. A. Simmons. Solid-state lighting: an energy-economics perspective. *Journal of Physics D: Applied Physics*, 43(35):354001, September 2010. doi: 10.1088/0022-3727/43/35/354001.

[51] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*, Focus on Computer Graphics, pages 145–167. Springer, 1995. doi: 10.1007/978-3-642-87825-1_11.

[52] Edgar Velázquez-Armendáriz, Eugene Lee, Kavita Bala, and Bruce Walter. Implementing the render cache and the edge-and-point image on graphics hardware. In *Proceedings of Graphics Interface 2006*, GI '06, pages 211–217, 2006. ISBN 1-56881-308-2.

[53] Edgar Velázquez-Armendáriz, Shuang Zhao, Miloš Hašan, Bruce Walter, and Kavita Bala. Automatic bounding of programmable shaders for efficient global illumination. *ACM Transactions on Graphics*, 28(5):142:1–142:9, December 2009. doi: 10.1145/1618452.1618488.

[54] Channing Verbeck and Donald P. Greenberg. A comprehensive light-source description for computer graphics. *IEEE CG&A*, 4(7):66–75, July 1984. doi: 10.1109/MCG.1984.275906.

[55] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. Lightcuts: a scalable approach to illumination. *ACM Trans. Graph.*, 24(3):1098–1107, July 2005. doi: 10.1145/1073204.1073318.

[56] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proc. of EGSR*, pages 195–206, 2007. doi: 10.2312/EGWR/EGSR07/195-206.

[57] Bruce Walter, Kavita Bala, Milind Kulkarni, and Keshav Pingali. Fast agglomerative clustering for rendering. In *IEEE Symposium on Interactive Ray Tracing, 2008. RT 2008*, pages 81–86, 2008. doi: 10.1109/RT.2008.4634626.

[58] Gregory J. Ward. The RADIANCE lighting simulation and rendering system. *Proc. of SIGGRAPH*, pages 459–472, 1994. doi: 10.1145/192161.192286.

[59] Eric Werness and Piers Daniell. ARB texture compression BPTC (block partitioned texture compression), January 2011. URL `http://www.opengl.org/registry/specs/ARB/texture_compression_bptc.txt`.

[60] *Mathematica Documentation: MiniMaxApproximation*. Wolfram Research, November 2012. URL `http://reference.wolfram.com/mathematica/FunctionApproximations/ref/MiniMaxApproximation.html`.

[61] Dan Zuras and Mike Cowlishaw. IEEE standard for floating-point arithmetic. IEEE Std 754-2008, IEEE, New York, NY, August 2008.