EUROGRAPHICS Workshop on ... (200x), pp. 1–9
N.N. and N.N. (Editors)

An Interactive Information Visualization Approach to Physically-Based Rendering

Gerard Simons[1], Marco Ament[2], Sebastian Herholz[3], Carsten Dachsbacher[2], Martin Eisemann[1,4], and Elmar Eisemann[1]

[1] TU Delft, Netherlands, [2] Karlsruhe Insitute of Technology, Germany, [3] Tübingen University, Germany, [4] TH Köln, Germany

Abstract

In this work, we present a novel information visualization tool to gain insight into the light transport in a physically-based rendering setting. The tool consists of a sampling-based data reduction technique, an extended interactive parallel coordinates plot providing an overview of the attributes linked to each light sample, 2D and 3D heat maps to represent different aspects of the rendering process, as well as a three-dimensional view to display and animate the light path transportation throughout the scene. We show several applications including differential light transport visualization for scene analysis, lighting and material optimization, reduction of rendering artifacts, and user-guided importance sampling.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Interactive feedback in a rendering system is important for lighting planners or rendering engineers to observe the effects of changes to the scene. Advertisement, and movie production focus on lighting design for mostly aesthetic purposes. Automotive engineering search advanced lighting systems to properly illuminate the environment, while not blinding opposing traffic. In architecture, lighting environments are designed for comfort or follow legal requirements of work environments [Rel]. Biologists optimize plant illumination to improve photosynthesis. All these applications require physically-based light transport (PBLT) [Kaj86], the industry standard for synthesizing realistic images, which can take hours of compute time per image.

Research on adapting general and established visualization tools to support users in light design is in its infancy and existing approaches are often limited to very specific scenarios. The intended users of our approach are rendering engineers who are facing challenges, such as finding out how light is distributed in a scene, and how it is influenced by certain objects, finding scene parts that cause significant noise in the final PBLT image [KKG*14], or they need to steer the computation to improve rendering convergence. These issues essentially stem from a lack of control and feedback in current renderers, which we aim to address by augmenting a state-of-the-art renderer with interactive visualization tools.

Other professionals, such as industrial designers could also benefit from the proposed solution, by being able to track how light is being transported. This could help in designing products such as car head lights, for example.

Our main contribution is a comparative light path visualization tool to improve the user's understanding of light transport within a single scene, including potential changes. In order to attain this, we save additional information for a carefully-chosen set of radiance samples, which we use for visualization and guidance of the rendering algorithm. For their visualization, we use parallel coordinate plots [ID91] with established brushing interaction metaphors and subset selection [Sii00]. We also offer heat-map information visualization within a 2D and 3D spatial context. Furthermore, light path trajectories can be animated to illustrate propagation and distribution. Special emphasis also lies on change visualization due to scene edits. Finally, we show how these techniques can be used to guide the rendering process to improve convergence. In our examples we use a unidirectional path tracer though the presented concept generalizes to any path space sampling technique.

2. Related Work

PBLT computes global light transport without participating media via the rendering equation [Kaj86]. As no analytical solution exists for non-trivial scenes, Monte Carlo (MC) methods [CPC84] approximate the required integrals. Here, samples are drawn from an appropriate probability distribution function to create paths simulating the light transport from source to camera throughout the scene [Kaj86, Jen01, VG97]. However, even for state-of-the-art MC methods [KKG*14] and efficient ray tracing engines [WFWB13], PBLT is far from real-time performance in complex scenes. Consequently, it is difficult for designers and rendering engineers to optimize their scene as any changes require a costly re-rendering.

Visualization of light transport is gaining popularity in computer graphics and visualization. Earlier approaches used a signal-processing framework to visualize light frequency content and its change upon interaction with materials [DHS*05]. In the spatial domain, light rays can be illustrated as geometric primitives (e.g., line segments) from a third-person's view [Rus99]. The ray tracing visualization toolkit [GFE*12] provides additional payloads for recording the ray state, which allows one to navigate in the ray tree or to filter rays according to attribute or type [LP14].

For a higher level of abstraction, spherical plots and particle flow tools for a selective inspection of light transport support digital artists in determining the origin of certain illumination features [RKRD12]. Edge bundling [HvW09] applied to light path visualization reduces visual clutter and supports artists in path retargeting [SNM*13]. However, none of these techniques provides a tool for comparing light transport in different scene setups.

High-dimensional data is often explored with information visualization techniques [TM04]. In light transport, dimensionality reduction can be used on quantities, such as the irradiance vector field (indicating the dominant light direction [CWW11]) or finite-time path deflection [ZAD15] (a scalar-valued quantity representing deflection characteristics of all light paths through a given point). Such an information compression may aggravate the analysis of complex scenes, especially for slight changes.

Parallel coordinates [ID91, HW13] are a useful technique to visualize high-dimensional data, as they scale linearly with the number of dimensions. We, therefore, make use of it together with interactive dimension reordering to visualize our attribute space, which is in the range of 20 dimensions. A parallel coordinates plot (PCP) depicts K-dimensional data by displaying K axes in a parallel arrangement rather than orthogonally. A K-dimensional data point corresponds to a poly-line connecting all axes, Fig. 1. To reduce visual clutter from larger data sets, density-based techniques [MW91, HW09] can be used. However, achieving interactivity is very challenging for large data sets, consisting of thousands or millions of data points, as in our case. Similarly, edge bundling [LWZK08, PBO*14], hierarchical clustering [War94] and other visual clutter reduction techniques [ED07] have been shown to be useful for up to a few thousand of points, but have not been tested for millions of points. It is unlikely that these techniques would be fast enough to still allow for real-time interactivity.

We, therefore, apply a sample-reduction technique beforehand. In light transport visualization, PCPs have been used to visualize photon distributions and explore their attribute space [SJL15], however, not for comparative visualization.

LiteVis [SOL*16] is a visual analytics approach to compare results of several simulation runs for interactive lighting design. For fast feedback, the approach builds upon virtual point lights [LTH*13], which is biased, while PBLT is unbiased. Also, their visual analysis is restricted to surface measurements, while our method supports general investigations of illumination, can be used to optimize spatial arrangements and scene appearance, and can be used to guide the rendering process for faster convergence. While inverse rendering [PP03] might be able to solve some of these problems, more fine-grained information visualisation and manual inspection is often required.

3. Overview

Our comparative approach provides insights into the light transport before and after a scene edit. It enables to guide the rendering process and improve its convergence in difficult scenes. To this extent, we collect, preprocess (Sec. 4), and visualize light transport data. The system provides according visual feedback, and respective tools for interaction (Sec. 5), Fig. 2(a).

A typical use case is as follows: The user renders a scene and our application collects data in the background, which is visualized as a PCP. Using brushing, rays are selected based on certain light path attributes. These rays can be used to guide the sampling for faster convergence, without the need to select those regions in image space, which can be difficult, very tedious, or even impossible. It should be noted that known render-time statistics may also be used to discover convergence problems, but are insufficient in actually resolving them. This is useful, e.g., for removing firefly artifacts (high energy samples), which can be selected for examination and then their cause can be resolved. Another possibility is to edit the scene and visualize the resulting light transport differences. A great benefit of our approach is that the user does not need to wait for the renderer to finish, but the comparative visualization is updated iteratively on-the-fly. At any moment, the user is free to pause rendering and investigate the current situation. A typical example is the selection and visualization of the most different light paths after a scene edit, e.g., to optimize reflector placements in a greenhouse, which we will use as a running example, but we will present additional results in Sec. 6. These components provide convenient ways to guide modeling and rendering. It should be noted that the current application is a prototype using only a subset of the available data. The prototype is used merely to show the potential of visualisations in a physically-based rendering context.

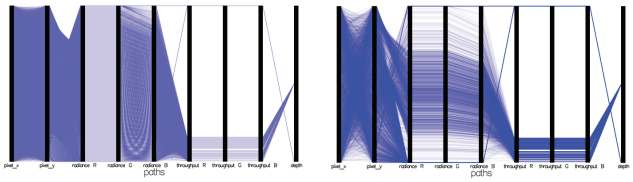
4. Data

We first discuss the data collected during rendering, which is the basis of our visualization and interaction techniques. For an in-depth explanation of the following attributes, we refer the reader to the textbook by Pharr and Humphreys [PH10].

First, for each **path** from the light to the camera, we collect the following properties: its *pixel position*, *exitant radiance*, *throughput* (computed from the BSDFs and pdfs along the path), and *depth* (number of intersections along the path). Second, for each **intersection** of a ray with the scene, we collect its *position*, *exitant radiance*, *object identifier*, *bounce number*, and *interaction type*, which can be either reflection or refraction. Finally, for each **light source**, we collect its emitted *radiance* along each sample and the *light identifier*. We display these three groups separately in our tool.

4.1. Data Reduction

MC rendering techniques require a large number of samples per pixel, up to thousands, for a high-fidelity image. Too many samples would clutter the visualizations and strongly limit interactivity. Consequently, we restrict the number N of visualized samples to a range of $N = 5,000$ to $N = 10,000$, which was found empirically



(a) (b)

Figure 1: Parallel coordinates are prone to suffer from over-plotting. (a) A PCP with 500,000 data points. (b) A PCP with 5,000 representative data points. In both figures, lines are drawn semi-tranparently, but the overall distribution of the data is much more apparent in (b) whereas this distribution is almost invisible in (a) due to over-plotting.

to be a good tradeoff between memory consumption and resolution. For a faithful representation, we need samples closely matching the true distribution of all $M >> N$ samples. Using standard clustering techniques, like Ward's method, is infeasible as saving all attributes for all samples is prohibitive in terms of memory requirements which is why we opted for a sampling approach.

During rendering, we compute and update a set of histograms from the true distribution (using the full set of M samples), but without storing them in memory. We compute one histogram per data attribute. We maintain a set of N samples, with $N << M$ in memory, whose normalized histograms H_N closely resembles the normalized histogram of the full sample set. To this extent, we combine the old set of samples with the set of new samples (computed during one rendering iteration) and find a new set of N samples via an iterative optimization process. The latter selects J random sets of N samples (in practice, $J = 5$), computes their respective histogram H_N^j and compares these to the true distribution H_M to determine the closest set according to the metric:

$$D(H_N, H_M) := \sum_{i=1}^{I} \left| \frac{m_i}{M} - \frac{n_i}{N} \right|, \tag{1}$$

where I is the number of discrete bins (in our case 10 per attribute), and m_i and n_i are the numbers of occurrences in bin i with respect to the true and reduced distribution, respectively. The computed distance is an estimate for the goodness of fit of each sample set, though we did not investigate the quality further as selecting the true best set is infeasible. A user-defined selection (Sec. 5) can be used for the sample-set construction to set regions of interest. Figure 1 shows an example of our data reduction with a PCP.

5. Visual Interface

Our visual interface consists of three components: the PCP for global data exploration (Sec. 5.1), the render view for image-space exploration (Sec. 5.2), and the scene view for object-space exploration (Sec. 5.3). All views are connected with the brushing-and-linking paradigm [Sii00]. In particular, our tool allows a user to compare two different scenes via color-coded or side-by-side views. Linking together the 2D and 3D views of the data is especially powerful as it brings together scientific and information visualisation in one application.

Fig. 2(a) shows an example of our tool. Here, two scenes with two plants (red and blue frame) are compared. They only differ by a highly reflective wall placed in the right scene (blue frame). The PCP (yellow frame) provides a quick overview of the data. Additionally, our interface allows the user to interact with the information. In Fig. 2(b), we brushed the paths that contain high radiance values in the green channel (orange rectangle). We see how both render-views now highlight the image parts corresponding to the selection. The bins in the PCP are also updated: the red view shows another plant and, hence, most bins appear as red.

The renderer itself is connected to the interface with an interactive feedback mechanism. It updates the relevant data more prominently and delivers precise insights more efficiently. This mechanism can also be used to steer the rendering process, as the chosen light paths contribute to the final solution (e.g., for noise reduction or firefly elimination), similar to importance sampling [PH10].

5.1. Parallel Coordinates Plot

For the global exploration of the collected light path data, we opt for parallel coordinates [ID91]. They are well suited for our roughly 20 data dimensions and allow us to easily select subsets using a brushing metaphor, but may be extended to any arbitrary number of dimensions. From Sec. 4, we recall that the data of our sampled **paths** consist of multiple **intersections**, which may be associated to multiple **light sources**. Because of these N-ary relationships, we use three separate PCPs which are then linked together: one for paths, one for its constituent intersections, and one for light sources. It should be noted that the PCP is not always the most effective method of visualising spatial data, e.g., the intersection points in a scene (for which we have developed a separate scene view linked to the PCP), but that it is nevertheless deemed useful in brushing these and other dimensions.

To facilitate the comparison of two different scenes, we render their data sets into a single PCP and use color mapping to visually distinguish them (yellow frame in Fig. 2). Red and blue-colored lines in the PCP indicate the two scenes.

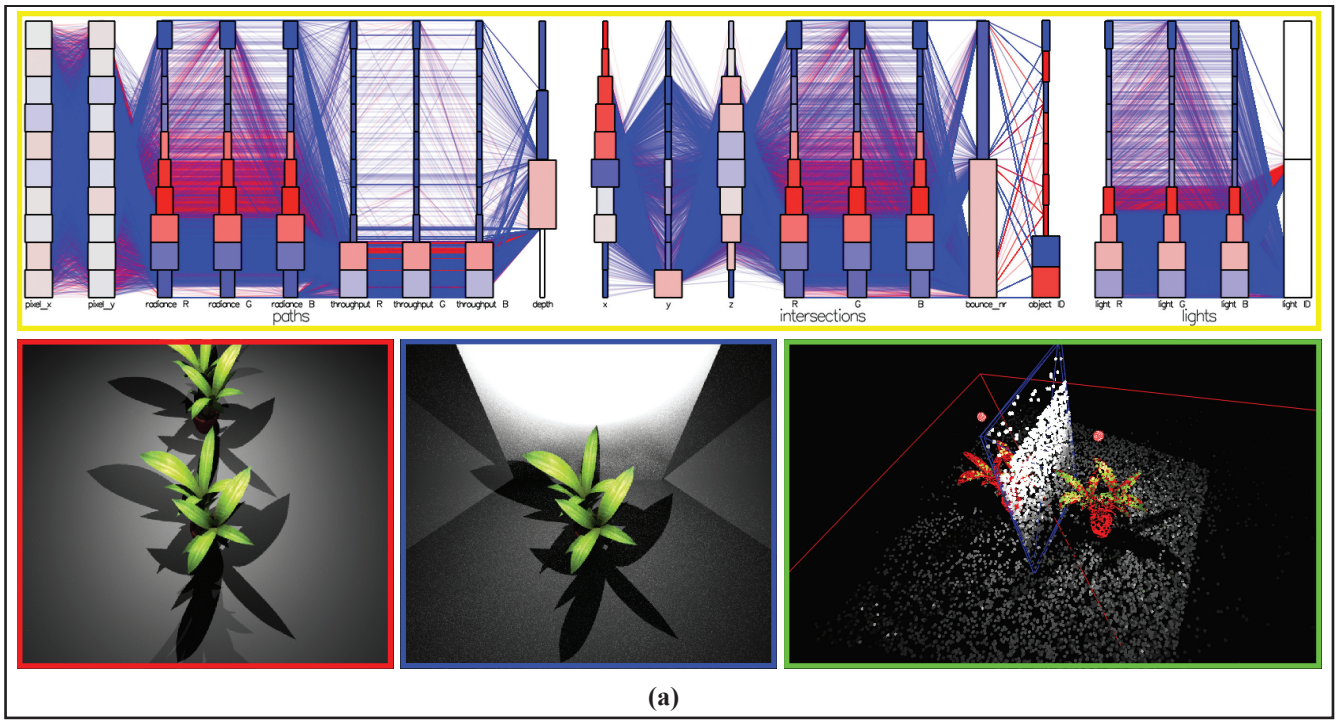
Adding binning [HLD02] to the PCP illustrates how many data points lie within discrete regions of each dimension. The width encodes the number of samples in each bin. Specifically, if we denote the value in bin b for data dimension x with $H_1(b,x)$ and $H_2(b,x)$ for the first and second data set, respectively, then the difference of these two values determines the color of the bin:

$$H_{\text{diff}}(b,x) = \frac{(H_1(b,x) - H_2(b,x))}{(H_1(b,x) + H_2(b,x))} \tag{2}$$

with $H_{\text{diff}}(b,x) \in [-1,1]$. The yellow frame in Fig. 2 is an example where the difference is mapped to a divergent color map of red-gray-blue. A saturated red bin indicates that all data points within this bin belong to the first scene, whereas blue stands for the second scene. A gray color is used to indicate a bin whose ratio is balanced.

5.2. Render View

The render views (red and blue frame in Fig. 2) display the current result of the rendering processes of two different scenes. As the rendering process progresses these views are updated. Besides



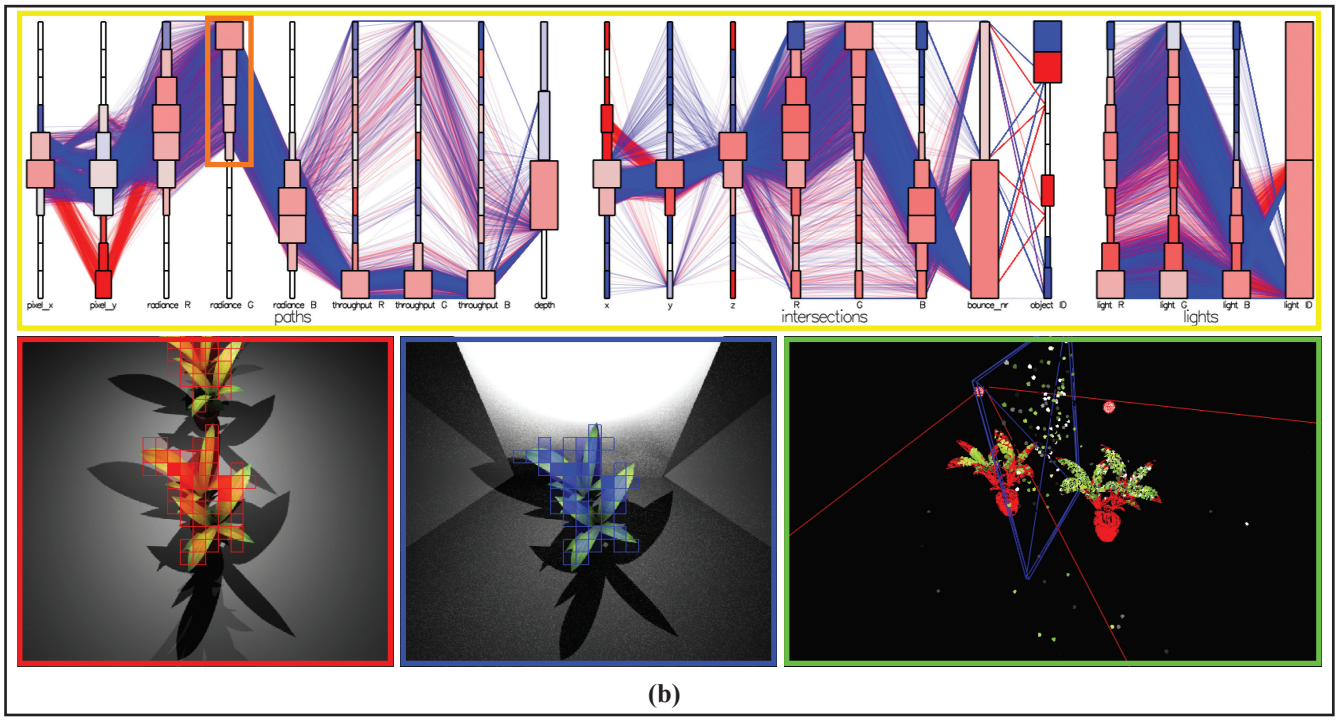


Figure 2: Overview of our visualization tool. The parallel coordinates plot (yellow), the render views (red and blue like the colors in the parallel coordinates plot), and the scene view (green). (a) Visualization of the full data set. (b) Brushing of light paths that have high radiance values in the green component (indicated by the orange rectangle on the axis).

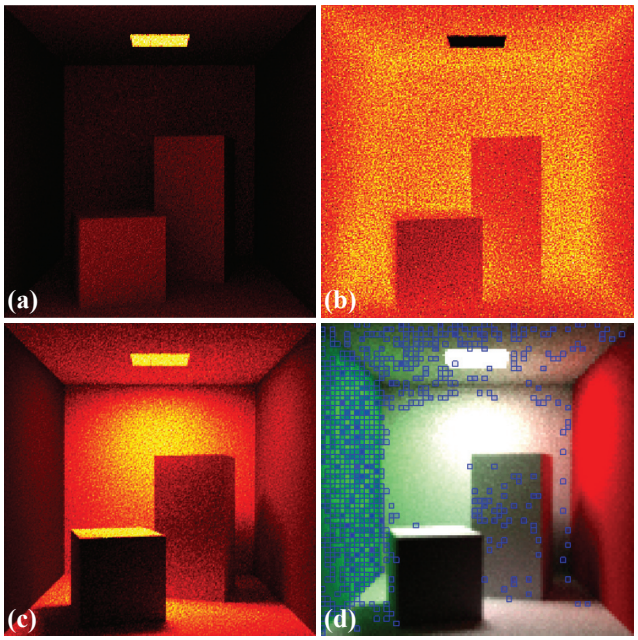


Figure 3: Heat map visualizations of the (a) throughput, (b) number of intersections, (c) radiance, and (d) brushed light paths.

showing the scene itself, the render view is also used to explore and visualize the collected data in 2D image space with heat maps.

Fig. 3 shows the Cornell box with four heat-map visualizations. In Fig. 3(a), the throughput distribution is displayed with the hot body color map, which shows high values around light source and boxes. In Fig. 3(b), the same color map is used to visualize the number of intersections (depth) and in Fig. 3(c) the radiance distribution. In addition, light paths that are brushed in the PCP can be visualized with a semi-transparent monochromatic heat map, as shown in Fig. 3(d) where only light paths with strong radiance contributions in the green channel are selected.

The render view also provides an interactive aspect. A user might be interested in a distinct region in the image, such as a specular highlight, and the render view allows a user to scribble in the image. The area enclosed by this scribble is then used to find paths that originate inside this region, which allows for non-rectangular selections in comparison to selecting pixel positions in the PCP.

5.3. Scene View

The scene view (green frame in Fig. 2) is used to explore and visualize the data in 3D object space. MC rendering consists of rays recursively hitting geometry in the scene, which makes displaying these paths along with their intersection points an intuitive way to convey light transport. Intersection points are colored according to the received energy, whereas paths are colored according to the gross accumulated energy. Alternatively when we are more interested in how light is transported at each bounce, we can color the points according to the number of bounces from the light source. We chose to render animated line segments to visualize the light paths, please see the supplemental video. Poly-lines connecting the intersection points would cause clutter and a loss of orientation.

3D Heat Map Visualizing energy on surfaces, instead of along paths, can be helpful for object placement, e.g., to expose it to a lot or little light. For this purpose, we create a 3D heat map of the energy distribution using a sparse voxel octree [LK11]. It should be noted that although the 3D heat maps themesleves are not a new contribution, their usage in this context is, to the best of our understanding, novel. We start with a single root node, which is a voxel containing all intersection points. We recursively split voxels into eight new voxels, append them as children and proceed with processing these, until a voxel no longer contains an intersection point, or a maximum tree depth is reached. The color of a voxel is computed according to the average of the intersection point energies over all (or selected) color channels. The average energy is then transformed into a voxel color by remapping the value with a linearly interpolated hot-body color map. Alternatively, if we are interested in the differences of two scenes, we subtract the average energy of the voxels of the two octrees and we use the red-blue divergent color map for visualization.

Fig. 4 depicts an example of a 3D heat map for the comparison the two scenes in Fig. 2. Fig. 4(a) shows the energy distribution in the scene, where a reflective wall is placed between the two plants, whereas Fig. 4(b) shows the result without the wall, and Fig. 4(c), the difference of both setups using the divergent color map. In the latter, blue/red color indicates that the scene with/without the wall receives more energy. Some of the leaves and the shadow receive more energy in the presence of the wall, as indicated by the blue speckles (highlighted with the yellow arrows).

Interaction The scene view also allows one to brush data in object space. Each object in the scene is associated with a unique identifier and by clicking on an object, the user restricts the light paths to those interacting with the object. We offer three different selection mechanisms. First, the *path-selection mode*, where the user chooses objects consecutively and all light paths interacting with these objects in order are chosen. For example, to choose the paths hitting an object A before hitting an object B. Second, the *shadow selection mode* is used to display rays that hit the shadow of an object. At each intersection, a shadow ray is cast to each light source to determine if the intersection point can be reached from the light source. When it cannot be reached the point is said to be occluded and the energy for this light point is not taken into account, whose application will be illustrated in the next section, among additional use cases. Third, a region of interest (ROI) selection, for which a gizmo is placed in the scene to collect localized information [RKRD12]. We extend the use of the gizmo for our comparative visualization tool. By placing two gizmos, one in each scene, we can restrict the samples to those intersecting with the gizmos and make a comparison between these two distinct ROIs. Fig. 5(a) shows an example where all paths are visualized, whereas in Fig. 5(b) only those paths that intersect the gizmo are displayed.

6. Results

We have combined the previously proposed visualization techniques into a custom-built C++ OpenGL application and adapted the unidirectional path-tracing EMBREE renderer [WFWB13] for our purposes. In this section, we will discuss various example applications and the obtained results.

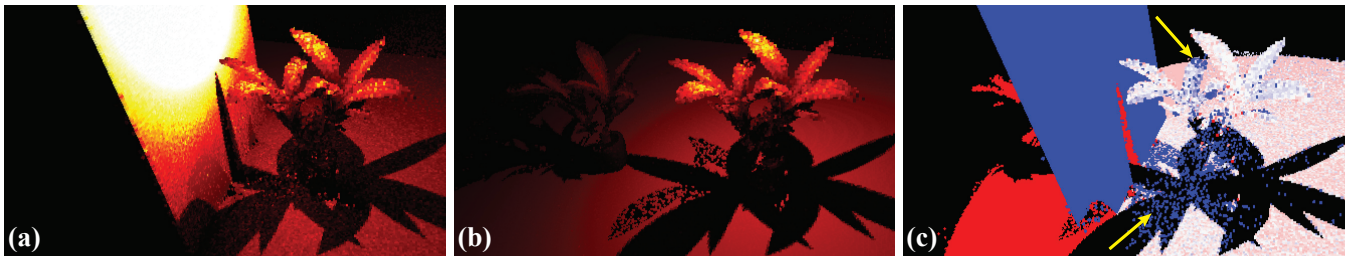


Figure 4: 3D heat map visualization of the energy distribution in a scene (a) with and (b) without a reflective wall between two plants. (c) Visualization of the difference of the energy distributions of both scenes with a divergent red-blue color map.

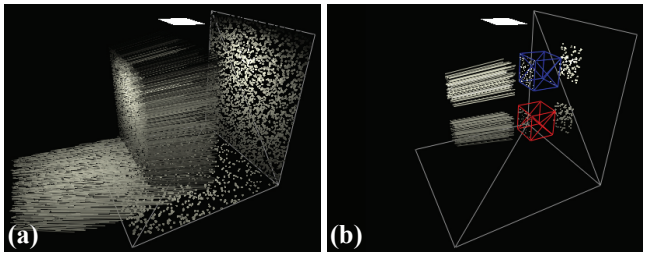


Figure 5: ROI-based path selection with gizmos. (a) Visualization of all light paths. (b) Visualization of the light paths that intersect the gizmos (red and blue cubes).

6.1. Scene Optimization

To demonstrate how our tool can aid in understanding the light distribution within a scene, we choose different prototypical use-cases from a greenhouse planning scenario. Here, the use of artificial light plays a significant role in the efficiency of the system and care has to be taken when setting up the environment.

Emitter Placement A constant uniform illumination of all plants is crucial to guarantee equal growth rates and to optimize the overall energy consumption. In our test, the radiance distribution of two light settings is to be evaluated (Fig. 6); one with two intense light sources (left column) and another with five smaller light sources (right column). The total sum of the energy of the light sources is the same. The first row shows the rendered results, while the second and third visualize the distribution of energy emitted by the two different light setups. While Fig. 6(c) and (d) show the distribution over the whole scene, Fig. 6(e) and (f) show a close up of the plants. The scene and leaves of the plants on the left in Fig. 6(d) and (f) are more uniformly illuminated as on the right in Fig. 6(c) and (e), which is considered beneficial for plant growth.

Reflector Placement In another scenario (Fig. 7(a)), the emitter (light source) positions and receiver (plant) are fixed. We want to add an additional reflector object to the scene to increase the light transport from the light source to the plant (Fig. 7(b)). To guide the placement of the reflector, we use our tool to visualize *how much* and *how* light reaches certain regions in the scene. The PCPs (Fig. 7(c)) of the initial scene in Fig. 7(a) are used to select data on secondary bounces that have a lot of energy (see the brushed markings on the "R" and "bounce_nr" axes). Furthermore, to exclude paths that go through the glass window at the front, only bounces

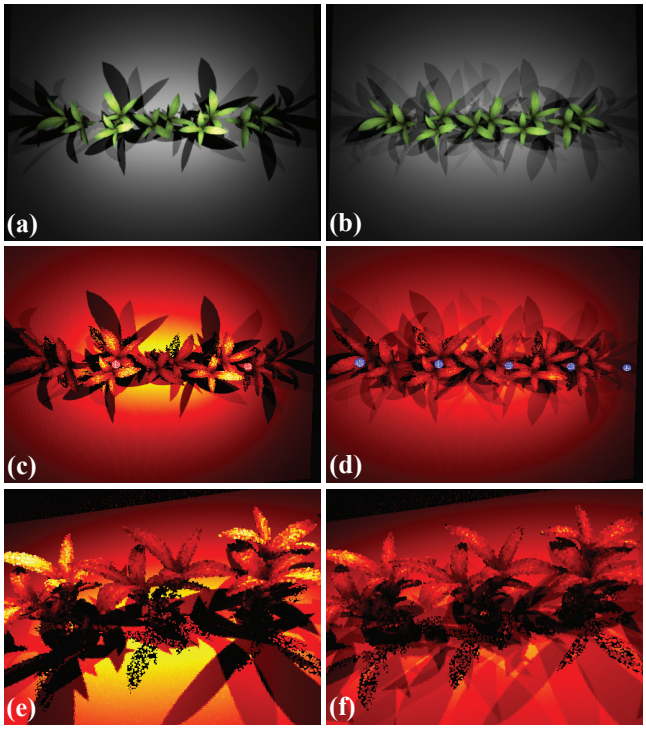


Figure 6: A 3D heat map visualization of two different lighting settings. The left column shows a scene with two large light sources, whereas the right column shows the result for five smaller light sources. (a) and (b) show the rendered result, and (c)–(f) the visualization of the energy distribution.

that are reflective rather than transmissive are brushed (see the axis "interaction_type"). The animation of the light paths (Fig. 7(d)–(g)) reveals how the light is diffusely reflected at the back wall without the reflector (red paths), while it is more focused on the plant with a reflector at the back wall (blue paths).

Receiver Placement In our third test scenario, Fig. 8, we compare two potential positions of the plant by placing two gizmos (red and blue) in the scene to record the light samples passing through them. We can observe in the respective PCP (Fig. 8(a)) that the overall radiance for the red gizmo (closer to the reflector) is higher, denoted by the reddish bins in the PCP, and the position therefore preferable, see also the animation in Fig. 8(b)–(e).

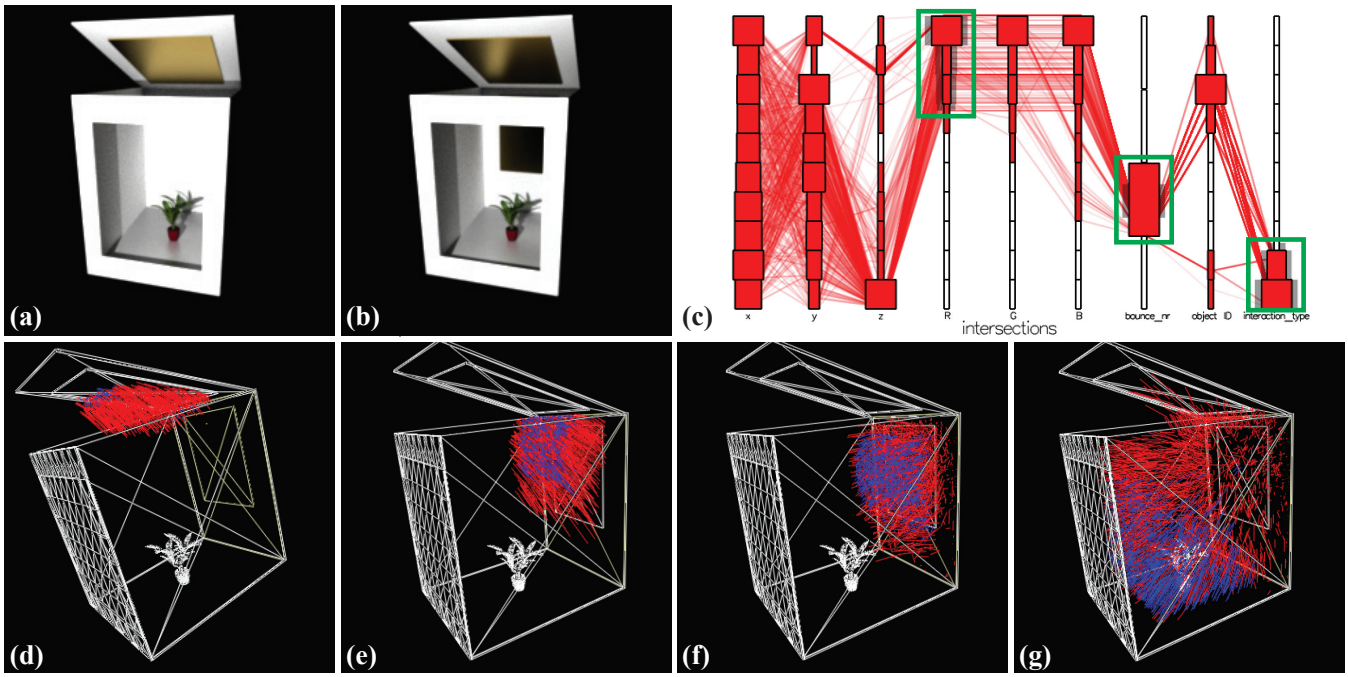


Figure 7: The greenhouse scenario for optimal reflector placement. Rendering of the scene (a) without and (b) with the reflector. (c) The PCP with data brushing (green markings). The selected light paths are visualized and animated in the scene view in (d)–(g), once with (blue paths) and without a reflector (red paths).

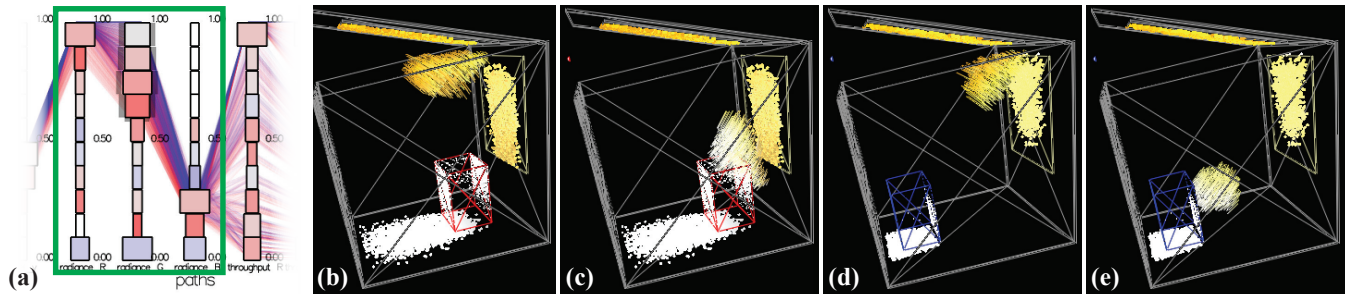


Figure 8: Gizmos can be used to compare two possible candidate positions for a plant. (a) The PCP displays the data associated with each gizmo. The selected paths are visualized and animated in the scene view for the first position in (b) and (c) as well as for the second position in (d) and (e).

6.2. Rendering Optimization

In the following, we show how our tool can be used to directly influence the rendering process to increase convergence rate and, therefore, speed up rendering times.

User-guided Rendering Convergence of an image is highly affected by the distribution of samples. We use user-guided rendering to assign more samples to areas containing complex path interactions, to improve convergence. Fig. 9 shows a result where this technique was used to concentrate sampling on the areas containing caustics on the bunny and wall which often require more samples. By gathering initial light path samples (e.g. 1,024 spp) and selecting those that lie in the shadow of the glass sphere (using the shadow selection), we can guide the computation towards this area. Note that our shadow definition ensures that caustics are always in the shadow of the casting object. Selecting paths interacting with the sphere would be another option here, though with a slower conversion as this affects more samples.

We can use the distribution of these paths (Fig. 9(a)) to guide the sampling to choose more paths interacting with the shadow of the glass sphere. This guided sampling leads to a significantly better result (Fig. 9(c)) compared to uniform sampling (Fig. 9(b)). The user-guided rendering outperforms uniform sampling in terms of mean square error (MSE) and structural similartiy (SSIM) [WBSS04] when compared to the reference (Fig. 9(d)).

Firefly Detection Fireflies emerge when a high intensity path is sampled with a small probability resulting in very bright pixels which are difficult to compensate for with additional samples. The fireflies cause is not always obvious and might even lie outside the

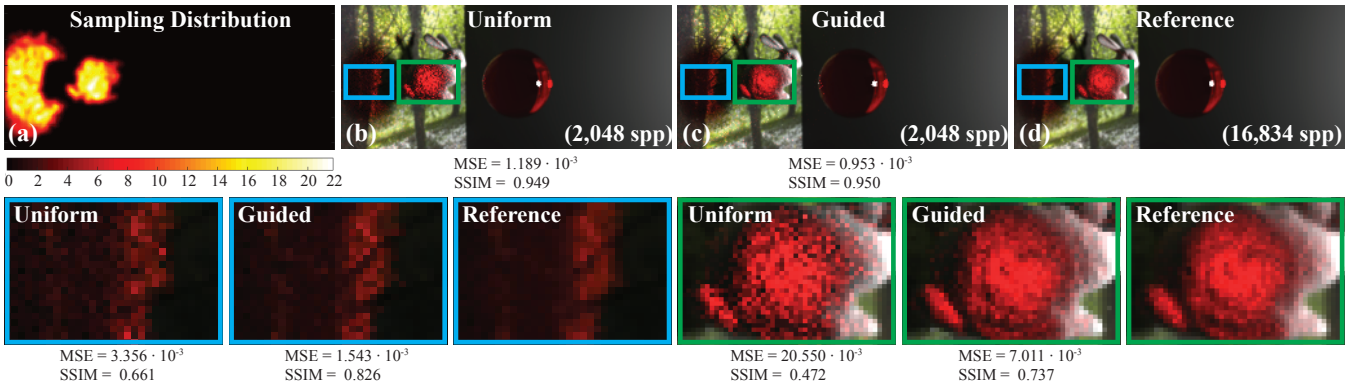


Figure 9: An example application of our user-guided rendering technique. (a) Visualization of the sampling distribution used to guide the rendering based on light paths whose shadow rays intersect the glass sphere. (b) Rendering with uniform sampling. (c) Our user-guided rendering technique. (d) Reference image.

rendered image. Standard techniques like clamping, or path skipping result in physically inaccurate renderings which may or may not be acceptable. A user informed of the causes of fireflies otherwise may be willing to make some changes in materials, geometry or camera parameters to reduce these effects.

Fig. 10 shows the results of such a scenario. We have created a gallery containing various objects on pedestals. We brush high energy paths, possible candidates for fireflies, to influence the sample selection. It should be noted that the according samples may not be visible in the PCP due to the sample reduction but the probability can be increased by selecting samples according to intensity. After some time, depending on scene complexity and firefly prevalence, enough such samples have been gathered and we can visualize how these paths interact with the objects in the scene, which are likely to be responsible for the fireflies. Fig. 10(a) shows the according selection in image space. Note how most firefly pixels are selected. In Fig. 10(b) and 10(c) snapshots from the animation of the high energy paths are shown and it becomes evident that many paths hit the sphere and its pedestal on the left (marked in green), as well as the statue in the back (marked in yellow) (please also refer to the accompanying video). By changing the material properties of the sphere, the pedestal and the statue to a more diffuse material, we can greatly reduce the number of fireflies (Fig. 10(d)). The improvement becomes especially obvious in the close-ups.

7. Conclusion and Future Work

Information visualization techniques can be of added value to users involved with light transport tasks. We covered engineering, botanic, and artistic examples that benefit from our analysis tools and user-guided rendering technique. Many other potential applications and also interesting future challenges remain, e.g., at the moment our technique is limited to static scenes without participating media. Investigating how to extend the technique, especially the rendering optimizations, for animations remains future work.

Our approach was kept as general as possible, with only a few specific optimizations, in order to retain a wide applicability, but customization options could be interesting in certain scenarios. As could other visualization techniques, which illustrate information at a more abstract level. Especially, to visualize the interaction with multiple objects (e.g., their total energy exchange or their impact on each other upon material changes). Although these changes may become apparent in our visualization tool, a more direct visualization might prove useful.

References

[CPC84] COOK R. L., PORTER T., CARPENTER L.: Distributed ray tracing. *ACM SIGGRAPH Computer Graphics 18*, 3 (1984), 137–145. 1

[CWW11] CHAJDAS M. G., WEIS A., WESTERMANN R.: Assisted environment map probe placement. In *Proceedings of SIGGRAD* (2011), pp. 17–25. 2

[DHS*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F. X.: A frequency analysis of light transport. *ACM Transactions on Graphics 24*, 3 (2005), 1115–1126. 2

[ED07] ELLIS G., DIX A.: A taxonomy of clutter reduction for information visualisation. *IEEE transactions on visualization and computer graphics 13*, 6 (2007), 1216–1223. 2

[GFE*12] GRIBBLE C., FISHER J., EBY D., QUIGLEY E., LUDWIG G.: Ray tracing visualization toolkit. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2012), pp. 71–78. 2

[HLD02] HAUSER H., LEDERMANN F., DOLEISCH H.: Angular brushing of extended parallel coordinates. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.* (2002), IEEE, pp. 127–130. 3

[HvW09] HOLTEN D., VAN WIJK J. J.: Force-directed edge bundling for graph visualization. *Computer Graphics Forum 28*, 3 (2009), 983–990. 2

[HW09] HEINRICH J., WEISKOPF D.: Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1531–1538. 2

[HW13] HEINRICH J., WEISKOPF D.: State of the art of parallel coordinates. In *STAR Proceedings of Eurographics* (2013), pp. 95–116. 2

[ID91] INSELBERG A., DIMSDALE B.: Parallel coordinates. In *Human-Machine Interactive Systems*. Springer, 1991, pp. 199–233. 1, 2, 3

[Jen01] JENSEN H. W.: *Realistic image synthesis using photon mapping*. AK Peters, Ltd., 2001. 1

[Kaj86] KAJIYA J. T.: The rendering equation. *ACM SIGGRAPH Computer Graphics 20*, 4 (1986), 143–150. 1

[KKG*14] KŘIVÁNEK J., KELLER A., GEORGIEV I., KAPLANYAN A. S., FAJARDO M., MEYER M., NAHMIAS J.-D., KARLÍK O., CAÑADA J.: Recent advances in light transport simulation: Some theory and a lot of practice. In *ACM SIGGRAPH 2014 Courses* (2014), pp. 17:1–17:6. 1

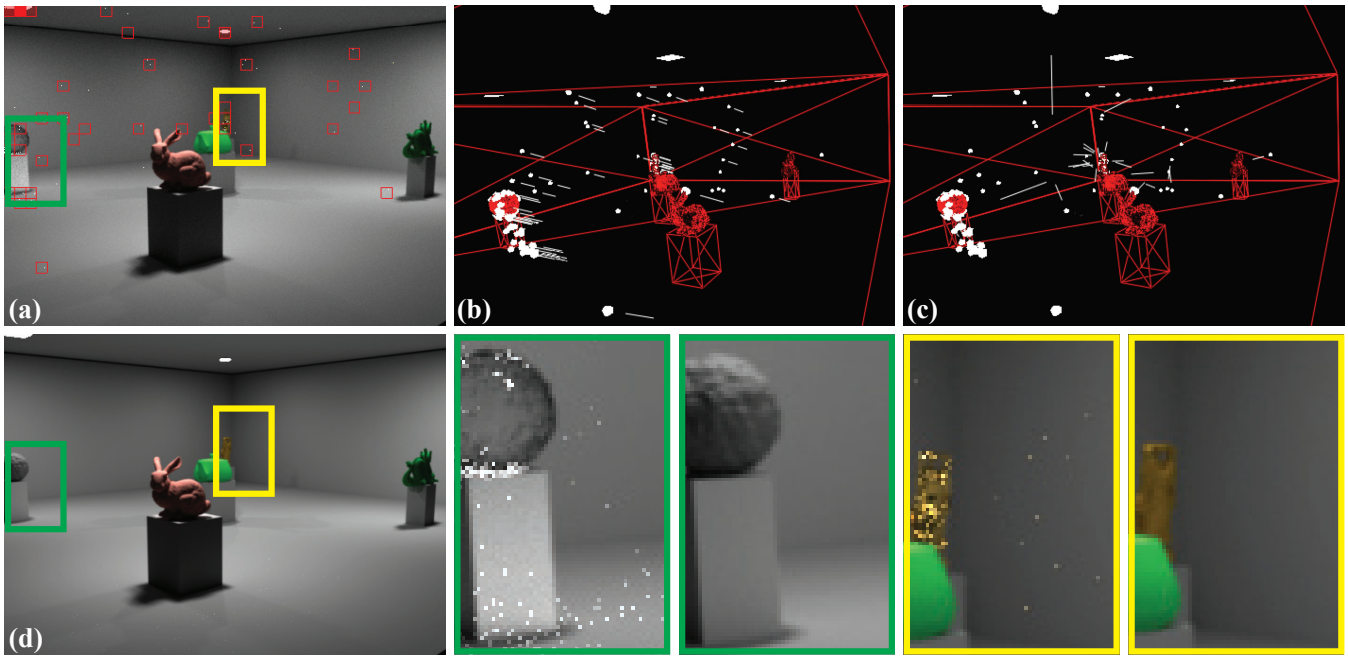


Figure 10: Collecting and animating high-energy paths to determine possible causes of fireflies. (a) The original scene and the selected light paths in the render view. (b) and (c) show the animation of these paths in the scene view. (d) The scene with the changed material in the render view. Close-ups are shown in the bottom right.

[LK11] LAINE S., KARRAS T.: Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics 17*, 8 (2011), 1048–1059. 5

[LP14] LESEV H., PENEV A.: A framework for visual dynamic analysis of ray tracing algorithms. *Cybernetics and Information Technologies 14*, 2 (2014), 38–49. 2

[LTH*13] LUKSCH C., TOBLER R. F., HABEL R., SCHW"ARZLER M., WIMMER M.: Fast light-map computation with virtual polygon lights. In *Proceedings of ACM Symposium on Interactive 3D Graphics and Games 2013* (2013), pp. 87–94. 2

[LWZK08] LUO Y., WEISKOPF D., ZHANG H., KIRKPATRICK A. E.: Cluster visualization in parallel coordinates using curve bundles. *IEEE Transaction on Visualization and Computer Graphics 14* (2008). 2

[MW91] MILLER J. J., WEGMAN E. J.: Construction of line densities for parallel coordinate plots. In *Computing and Graphics in Statistics*, Buja A., Tukey P. A., (Eds.). Springer-Verlag New York, Inc., 1991, pp. 107–123. 2

[PBO*14] PALMAS G., BACHYNSKYI M., OULASVIRTA A., SEIDEL H. P., WEINKAUF T.: An edge-bundling layout for interactive parallel coordinates. In *2014 IEEE Pacific Visualization Symposium* (2014), IEEE, pp. 57–64. 2

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., 2010. 2, 3

[PP03] PATOW G., PUEYO X.: A survey of inverse rendering problems. In *Computer graphics forum* (2003), vol. 22, Wiley Online Library, pp. 663–687. 2

[Rel] RELUX INFORMATIK AG: Reluxsuite. http://www.relux.info/. [accessed 27-Nov-2015]. 1

[RKRD12] REINER T., KAPLANYAN A., REINHARD M., DACHSBACHER C.: Selective inspection and interactive visualization of light transport in virtual scenes. *Computer Graphics Forum 31*, 2 (2012), 711–718. 2, 5

[Rus99] RUSSEL J. A.: *An interactive web-based ray tracing visualization tool*. Master's thesis, University of Washington, 1999. 2

[Sii00] SIIRTOLA H.: Direct manipulation of parallel coordinates. In *Proceedings of the International Conference on Information Visualization* (2000), pp. 373–378. 1, 3

[SJL15] SPENCER B., JONES M. W., LIM I. S.: A visualization tool used to develop new photon mapping techniques. *Computer Graphics Forum 34*, 1 (2015), 127–140. 2

[SNM*13] SCHMIDT T.-W., NOVAK J., MENG J., KAPLANYAN A. S., REINER T., NOWROUZEZAHRAI D., DACHSBACHER C.: Path-space manipulation of physically-based light transport. *ACM Transactions On Graphics 32*, 4 (2013), 129. 2

[SOL*16] SORGER J., ORTNER T., LUKSCH C., SCHWÄRZLER M., GRÖLLER E., PIRINGER H.: Litevis: Integrated visualization for simulation-based decision support in lighting design. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 290–299. 2

[TM04] TORY M., MÖLLER T.: Rethinking visualization: A high-level taxonomy. In *Proceedings of the IEEE Symposium on Information Visualization* (2004), pp. 151–158. 2

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), pp. 65–76. 1

[War94] WARD M. O.: Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization'94* (1994), IEEE Computer Society Press, pp. 326–333. 2

[WBSS04] WANG Z., BOVIK A., SHEIKH H., SIMONCELLI E.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4 (2004), 600–612. 7

[WFWB13] WOOP S., FENG L., WALD I., BENTHIN C.: Embree ray tracing kernels for CPUs and the Xeon Phi architecture. In *ACM SIGGRAPH 2013 Talks* (2013), pp. 44:1–44:1. 1, 5

[ZAD15] ZIRR T., AMENT M., DACHSBACHER C.: Visualization of coherent structures of light transport. *Computer Graphics Forum 34*, 3 (2015), 491–500. 2