

Unbiased Caustics Rendering Guided by Representative Specular Paths

He Li
Shandong University
China
he.li@mail.sdu.edu.cn

Kun Xu
Tsinghua University
China
xukun@tsinghua.edu.cn

Beibei Wang*
Nankai University
China
beibei.wang@nankai.edu.cn

Nicolas Holzschuch
University Grenoble Alpes, Inria,
CNRS, Grenoble INP, LJK
France
nicolas.holzschuch@inria.fr

Changhe Tu*
Shandong University
China
chtu@sdu.edu.cn

Ling-Qi Yan
University of California, Santa
Barbara
USA
lingqi@cs.ucsb.edu

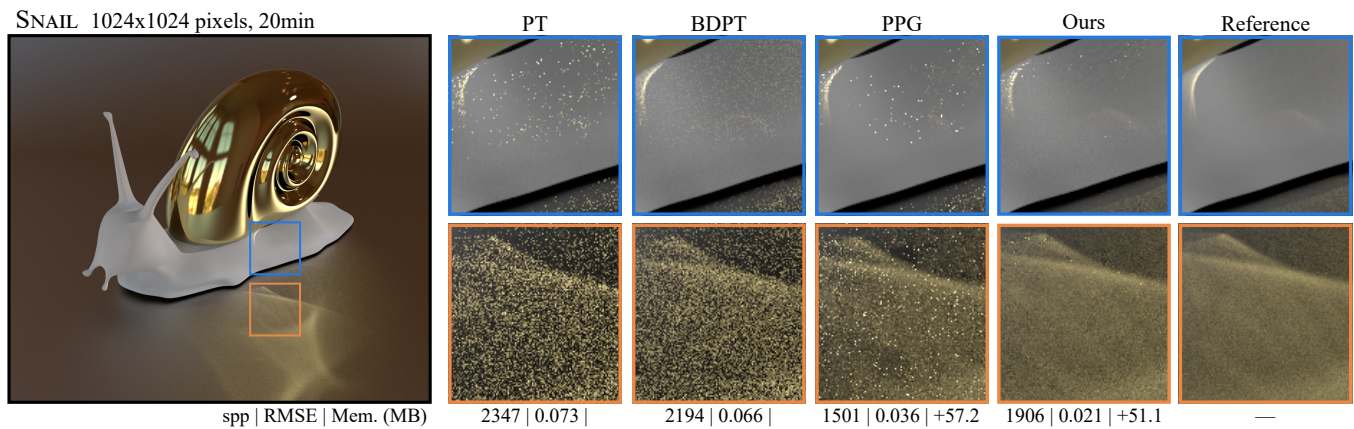


Figure 1: Equal-time (20 minutes) comparison between our method, path tracing (PT), bidirectional path tracing (BDPT), and practical path guiding (PPG) [Müller et al. 2017] on the Snail scene. This model is rendered using a dim environment light and a bright point light in front. Zoomed-in insets are rendered using only the point light. Our method has the lowest variance visually and quantitatively, and only requires a 6.5s precomputation time.

ABSTRACT

Caustics are interesting patterns caused by the light being focused when reflecting off glossy materials. Rendering them in computer graphics is still challenging: they correspond to high luminous intensity focused over a small area. Finding the paths that contribute to this small area is difficult, and even more difficult when using camera-based path tracing instead of bidirectional approaches. Recent improvements in path guiding are still unable to compute efficiently the light paths that contribute to a caustic. In this paper,

we present a novel path guiding approach to enable reliable rendering of caustics. Our approach relies on computing representative specular paths, then extending them using a chain of spherical Gaussians. We use these extended paths to estimate the incident radiance distribution and guide path tracing. We combine this approach with several practical strategies, such as spatial reusing and parallax-aware representation for arbitrarily curved reflectors. Our path-guided algorithm using extended specular paths outperforms current state-of-the-art methods and handles multiple bounces of light and a variety of scenes.

*Corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SA '22 Conference Papers, December 6–9, 2022, Daegu, Republic of Korea

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9470-3/22/12...\$15.00

<https://doi.org/10.1145/3550469.3555381>

CCS CONCEPTS

• Computing methodologies → Ray tracing.

KEYWORDS

path tracing, path cuts, path guiding

ACM Reference Format:

He Li, Beibei Wang, Changhe Tu, Kun Xu, Nicolas Holzschuch, and Ling-Qi Yan. 2022. Unbiased Caustics Rendering Guided by Representative Specular Paths. In *SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers)*,

December 6–9, 2022, Daegu, Republic of Korea. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3550469.3555381>

1 INTRODUCTION

Caustic effects are sharp, high-intensity illumination areas caused by curved specular or glossy reflectors that focus the incoming light on the receiver. Simulating them using light transport methods is challenging, because these caustics are caused by a tiny subset of the light paths. Finding all of these paths is difficult, resulting in large variance or temporal incoherence with most of the current light simulation methods.

Unidirectional Monte-Carlo approaches such as path-tracing usually exhibit large variance when rendering caustics, since they do not have information on their positions and perform sampling blindly. Path-guiding approaches [Herholz et al. 2016; Müller et al. 2017; Vorba et al. 2014] can reduce the variance using multiple iterations to learn the incoming radiance, but still cannot capture high-frequency caustics well since the online-learned radiance distribution is unable to capture these high frequencies, leading to inaccurate sampling.

Approaches based on Metropolis light transport can provide accurate sampling for caustics, but suffer from temporal coherence issues. Several recent approaches, such as Zeltner et al. [2020] compute specular paths accurately with purely specular surfaces, but suffer from glossy reflectors. Our key idea is to use representative specular paths as a proxy to guide sampling for rendering caustics, so that the difficult paths can be found quickly and exhaustively. Our approach is based on specular path cuts [Wang et al. 2020]. We first propose a relaxed path cuts solver, encompassing the (potentially infinitely many) caustics-carrying paths with a finite number of representative specular path cuts. Then we accumulate the radiance along each representative specular path cut with Spherical Gaussians (SGs), which function as an incident radiance distribution to guide sampling. To further improve the performance, we propose a spatial reusing strategy, combined with a parallax-aware representation for arbitrarily curved reflectors. Our method can produce much less noise than prior works within equal time. In summary, our main contributions are the following:

- a path guiding approach via representative specular paths,
- a relaxed path cut approach to represent light transport among glossy surfaces,
- an SG-based representation to approximate the contribution from a representative specular path, and
- a spatial reuse strategy combined with a parallax-aware representation to improve the performance.

An open source implementation of our method is available at <https://github.com/Lihns/caustics-pathcut>.

2 RELATED WORK

We review three groups of approaches: bidirectional Monte Carlo sampling, path guiding, and specular manifold sampling.

Bidirectional Monte Carlo methods. Bidirectional path tracing (BDPT) [Veach and Guibas 1994] methods produce images with higher quality than path tracing, but still need a long time to converge for difficult paths, such as caustics and glossy inter-reflections.

Photon mapping [Jensen 2001] and its numerous variants [Hachisuka and Jensen 2009; Kaplanyan and Dachsbacher 2013; Lin et al. 2020] are efficient for caustics rendering, but these methods are usually biased and produce blurry caustics.

Path guiding methods. Path guiding methods learn the incoming radiance distribution from existing information about the scene to be rendered. The distribution is sampled, commonly combined with BRDF (bidirectional reflectance distribution function) sampling using Multiple Importance Sampling (MIS), to improve path tracing. Vorba et al. [2014] shot photons to learn the incoming radiance and represented it with a Gaussian Mixture Model (GMM), used for direction sampling in path tracing. Müller et al. [2017] used a spatial-directional tree instead to represent the incoming radiance distribution. Herholz et al. [2016] sample the product of the trained incoming radiance distribution and the BRDF instead of using MIS, resulting in higher sampling quality, at the cost of the product operator. Reibold et al. [2018] select paths which cause high variance from the sampled paths, build distribution with Gaussians around the selected paths, and progressively refine the guiding distribution by learning and sampling iteratively, similar to Müller et al. [2017].

Müller et al. [2018] proposed a deep neural network model to determine the accurate probability density function of samples, at the cost of expensive sampling. Bako et al. [2019] proposed an offline, scene-independent deep-learning approach that can importance sample first-bounce light paths for general scenes to avoid costly online training.

These path guiding approaches are consistent and temporally coherent, but they have difficulties discovering the difficult paths or identifying high-frequency effects since they rely on unidirectional path tracing or photon mapping to provide global information.

Specular manifold-based methods. Manifold exploration [Jakob and Marschner 2012] was originally proposed as an extension of Metropolis Light Transport [Veach and Guibas 1997]. It allows random walks on a specular manifold based on the Newton solver and is suitable for computing specular-diffuse-specular paths and caustics. However, it does not support glossy materials, as it requires a rigid separation between specular and diffuse materials. Half vector space [Kaplanyan et al. 2014] makes the separation between glossy and specular materials more natural and avoids the rigid separation. Hanika et al. [2015b] improved the approach to handle complex geometry and avoid the Jacobian computation issue. These approaches are based on Metropolis Light Transport (MLT) and inherit its temporal instability issues.

Hanika et al. [2015a] introduced manifold next event estimation (MNEE) in the Monte Carlo framework. It was extended to bidirectional path tracing by Speierer et al. [2018]. Recently, Zeltner et al. [Zeltner et al. 2020] proposed a specular manifold sampling for both caustics and glints. They change the specular manifold constraints of Jakob and Marschner [2012] to improve robustness and convergence. However, these methods have performance degradation when dealing with reflectors with large material roughness, as they handle roughness by integrating over many pure specular light paths with randomized offset normals. Since there are infinite points on a glossy surface contributing to the shading point, it becomes less efficient for large-roughness reflectors.

In addition to the methods above, Loubet et al. [2020] derived analytic expressions that predict the total radiance due to a single reflective or refractive triangle with a microfacet bidirectional scattering distribution function (BSDF), which is reduced to the Lambert boundary integral. The analytic expressions allow for efficient sampling. Their method is limited to one intermediate glossy interaction, and the extension to more bounces is not apparent.

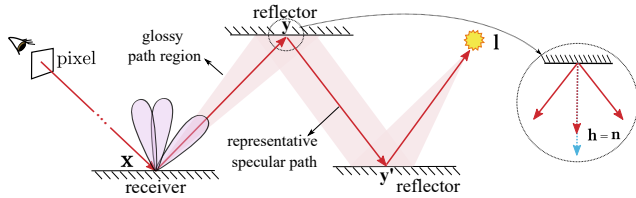


Figure 2: A shading point x on the receiver and a point on the light l can be connected by points on the reflectors, constructing an admissible path. If the reflector is pure specular, the number of paths is finite; if the reflector is glossy, the admissible paths are infinite. The neighboring admissible paths form a glossy path region. We use a specular path to represent this glossy path region. Note that there may be multiple glossy path regions that connect x and l .

3 OUR APPROACH

3.1 Problem analysis and motivation

We define the *reflectors* as the surfaces that are closer to the light source and are casting caustics, and the *receiver* as the surface that shows the caustics, as shown in Fig. 2.

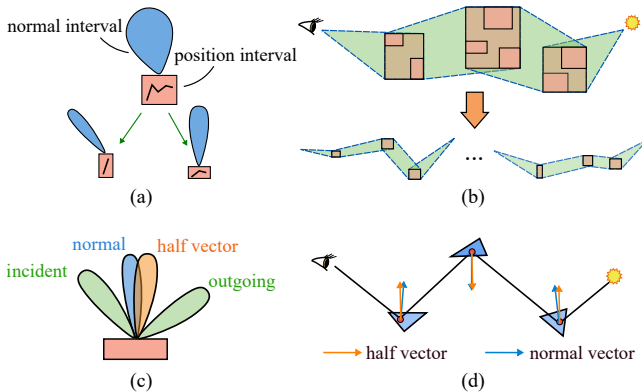


Figure 3: Illustration of the path cuts method [Wang et al. 2020]. (a) The scene is first organized into a spatial hierarchy, where each node has a position interval and a normal interval. (b) k nodes form a k -bounce path cut. Path cuts are subdivided and pruned to find valid leaf path cuts (i.e., all of the nodes are leaf nodes). (c) If for each node in a path cut, the normal interval and half vector interval have an intersection, then the path cut is valid. (d) For a leaf path cut, the Newton solver finds an admissible specular path by minimizing the differences between the half vectors and the normals.

Wang et al. [2020] proposed a path cuts approach for pure specular light transport, which finds pure specular paths that connect a pinhole camera and a point light source. As shown in Fig. 3, they organize the scene with a spatial hierarchy, where each node has a normal interval and a position interval. Then they construct k -bounce “thick paths” (called *path cuts*) with k -tuples of the tree nodes. By subdividing and validating the path cuts, they find valid leaf path cuts that might include a valid solution. Then they use a Newton-based solver to find an admissible specular path within a given path space region. In this way, the path cuts method can find most of the pure specular paths in the scene.

Inspired by their approach, we think that path cuts can also be used beyond specular light transport, e.g., for path guiding for caustics effects. With the specular paths as a guide, we can quickly discover the glossy path collections that contribute to the caustics, estimate the energy distribution along these paths, and then utilize it for sampling.

Given two endpoints (l from the light source and x on the receiver), if the reflector surface is purely specular, we can find points on the reflectors and construct the admissible paths or path cuts that satisfy the half vector h and the normal n aligned requirements. Then the caustics can be rendered immediately. For glossy reflectors, the set of admissible paths connecting the two endpoints forms a region in path space instead of a lower-dimensional manifold. We cannot compute all these caustics directly, but we can choose some paths that represent the region well, approximate the incident radiance distribution that provides a guide for sampling at the shading point, and compute the caustics using path guiding.

But how can we find a representative path for each glossy path region? And how to compute the radiance distribution, once we have a representative path? We answer these questions in the following subsections.

3.2 Solution Overview

Our basic idea is to find a representative specular path for each glossy path region, “blur” the representative paths with Spherical Gaussians to cover the regions, and estimate a radiance distribution at the shading point to guide path tracing (see Fig. 2). Our method consists of the following steps:

- In the precomputation step, we find representative paths for glossy path regions with a *relaxed* path cuts algorithm (Sec. 3.3).
- Then we approximate the incident radiance distribution at each vertex along with a representative path by approximating SG integrals along the path (Sec. 3.4) and store the approximated distributions at a set of cached points for spatial reusing.
- During rendering, we compensate the parallax of the cached approximated distributions at shading points to achieve more accurate guiding. (Sec. 3.5).

Our method is suitable for rendering caustics from point light sources and small area light sources. We only consider reflective surfaces and leave refractive surfaces (essentially similar) for future work. We assume constant roughness for each object and do not support roughness maps. Please see Sec. 5 for detailed discussions on the limitations.

3.3 Relaxed path cuts for glossy materials

Our first goal is to get a representative path for each glossy path region caused by a glossy reflector. The algorithm is similar to finding a specular path [Wang et al. 2020], with a few differences.

Compared to Wang et al. [2020], we relax the intersection conditions using the surface roughness: we compute the dot product of the normal interval and the half-vector interval and compare the resulting maximum value with the roughness-aware threshold, known as the Spherical Gaussian compact- ε support [Wang et al. 2009]:

$$\theta = \arccos \left(\frac{\ln(\varepsilon\pi\alpha^2)\alpha^2}{2} + 1 \right), \quad (1)$$

where ε is a threshold, set as 0.01 in most cases, and α is the surface roughness. If the maximum dot product is larger than $\cos \theta$, the node is valid.

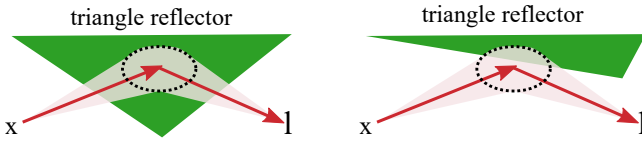


Figure 4: The point on the reflector that connects the representative specular path might be either inside the triangle (left) or outside the triangle (right). If the point falls outside the triangle, we use a point found inside the triangle with the smallest error.

For each leaf path cut found in the pruning step, we run a Newton solver to find the points on the triangles that connect the two endpoints. This gives us the representative specular path for the glossy reflection. The specular endpoint may fall outside the triangle while the glossy reflection still intersects with the triangle (see Fig. 4). To solve this issue, in the Newton solver, we find the point within the triangle with the smallest error. The details are shown in the supplementary material. We accept or reject this point by comparing the cosine of the angle between the half-vector and the normal at this point with the SG compact- ε support (see Eqn. 1).

Discussion. For each path cut, we find at most one solution as a representative path. As shown by Wang et al. [2020], multiple solutions might exist within a single path cut, when the triangles are exceptionally bumpy. Interval Newton could find all the solutions, but it is very costly. Here, we assume the normals of the triangle vertices are reasonably smooth, so that only one solution exists. When a reflector has large roughness, the path cut pruning becomes less efficient, resulting in many path cuts. This will lead to enormous memory costs and slow performance. Thus, we suggest using other approaches for this configuration.

3.4 Approximating incident illumination with SGs

For each representative specular path, we need to estimate the contribution of the glossy path region (see Fig. 2). Loubet et al. [2020] proposed to compute the contribution of a triangle in slope space, but their method is limited to one intermediate glossy interaction.

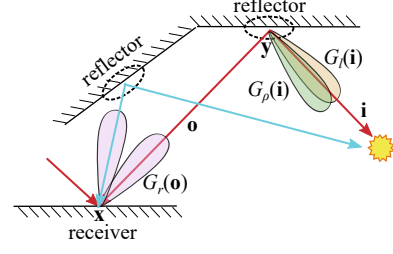


Figure 5: The radiance at the emitter is approximated with a SG $G_l(i)$ and a BRDF slice with outgoing direction \mathbf{o} is approximated with $G_r(i)$. With a representative specular path, the incoming radiance at \mathbf{x} from the current glossy path region will be $G_r(\mathbf{o})$. Two representative specular paths (blue and red) are shown.

We introduce an SG based-approach [Laurijssen et al. 2010; Xu et al. 2014], which can be extended to multiple bounces.

A Spherical Gaussian centered at direction \mathbf{p} with sharpness λ and amplitude A is defined by

$$G(\mathbf{v}; \mathbf{p}, \lambda, A) = Ae^{\lambda(\mathbf{p} \cdot \mathbf{v} - 1)}. \quad (2)$$

A BRDF slice for a view direction \mathbf{o} can be roughly approximated with a single SG [MJP 2016; Wang et al. 2009], by:

$$\rho(\mathbf{i}, \mathbf{o}) \approx G_\rho(\mathbf{i}; p_\rho, \lambda_\rho, C_\rho), \quad (3)$$

$$p_\rho = 2(\mathbf{n} \cdot \mathbf{o})\mathbf{n} - \mathbf{o}, \lambda_\rho = \frac{\lambda_{\text{NDF}}}{4\|\mathbf{n} \cdot \mathbf{o}\|}, C_\rho = C_{\text{NDF}},$$

where \mathbf{n} is the surface normal, $\lambda_{\text{NDF}} = \frac{2}{\alpha^2}$, $C_{\text{NDF}} = \frac{1}{\pi\alpha^2}$ is the sharpness and amplitude of the normal distribution function (NDF) approximation and α represents the surface roughness. Given an arbitrary reflector point \mathbf{y} , a point light or a small spherical light at position \mathbf{l} with radius r and intensity I can also be approximated with an SG:

$$L_e(\mathbf{i}) \approx G_l(\mathbf{i}; p_l, \lambda_l, C_l), \quad (4)$$

$$p_l = \frac{\mathbf{l} - \mathbf{y}}{\|\mathbf{l} - \mathbf{y}\|}, \lambda_l = \frac{4\|\mathbf{l} - \mathbf{y}\|^2}{r^2}, C_l = 2I.$$

The product integral of two SGs can be approximated as another SG [Xu et al. 2014], so that the incoming radiance at the shading point is approximated by

$$L(\mathbf{o}) = \int_{\Omega} G_l(\mathbf{i})G_\rho(\mathbf{i}) \cos \theta d\mathbf{i} \quad (5)$$

$$\approx G_r \left(\mathbf{o}; 2(\mathbf{n} \cdot \mathbf{p}_l)\mathbf{n} - \mathbf{p}_l, \frac{\lambda_\rho \lambda_l}{\lambda_\rho + \lambda_l}, \frac{(\mathbf{n} \cdot \mathbf{p}_l)2\pi C_\rho C_l}{\|\lambda_l \mathbf{p}_l + \lambda_\rho \mathbf{p}_\rho\|} \right),$$

where θ is the angle between \mathbf{i} and \mathbf{n} . Since the variation of $\cos \theta$ over varying \mathbf{i} is subtle, we use $\mathbf{n} \cdot \mathbf{p}_l$ to approximate it.

Eqn. 5 can be easily extended to more bounces by treating the computed SG as an emitter and continuing the accumulation.

At shading point \mathbf{x} , the incoming radiance distribution from a glossy path region is now represented with an SG, as shown in Fig. 5. Since there might be several representative specular paths for a shading point, we use a GMM to represent them. In practice, we notice that neighboring triangles might have the same reflector point. We compute the distance between the reflector points on the different representative paths. If the distance is smaller than a threshold, set as 0.1 of the triangle length, they are treated as the

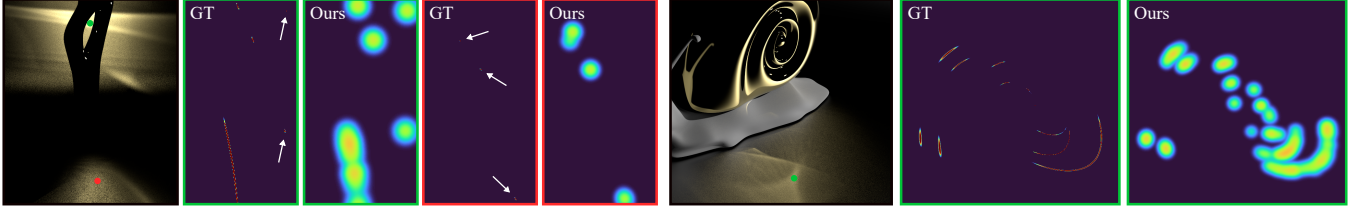


Figure 6: Comparison between our sampling distribution approximated by a SG and the target incident radiance distribution over the direction space at shading points. We mark out some unnoticeable details with white arrows. The target distribution shows that the shading points at the caustics have a high-frequency incoming radiance distribution. Our distribution shows the shape with some blurry of the target distribution. Please see the supplementary material for more discussions.

same reflector point. We only keep one representative path and remove the others.

In Fig. 6, we visualize the estimated incident radiance distribution at shading points. By comparing to the target distribution, we find that the distributions from representative specular paths act as a good approximation of the target distribution.

3.5 Spatial reusing of representative specular path

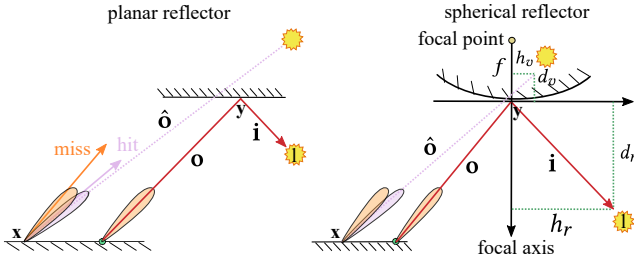


Figure 7: For both planar and spherical reflectors, the image of the emitter point is computed, and then connecting the image and the shading point x results in a corrected direction \hat{o} . Set the center direction of the SG from the cached point (orange) to \hat{o} , resulting in a rotated lobe (purple). The rotated lobe will correct the parallax between the shading and cached points.

Running a Newton solver for each shading point at run-time is too costly. We notice that neighboring shading points have similar incoming radiance distribution, and propose precomputing the GMMs representing the incoming radiance distribution at a set of cached points, and adapting these precomputed cached values for each new shading point.

At a shading point, we take the GMM from the nearest cached point, and adapt it to account for the parallax difference. Fig. 7 (left) explains the algorithm: the distribution at the nearest cached point (in orange) and the actual distribution at a shading point (in purple) are different, i.e., we rotate the cached SGs, so that the SGs are parallax compensated at the shading point. We describe how to correct the parallax in the following paragraphs.

Inspired by the parallax-aware representation for planar reflection [Ruppert et al. 2020], we propose to correct parallax for *arbitrarily curved* reflectors by treating the curved surface as two separate reflectors along with two principal directions, as shown in Fig. 8. We will first show the spherical reflector case and then present the arbitrary-curved reflector case.

Spherical reflector. As shown in Fig. 7 (right), for a spherical reflector, two axes are defined: the focal axis and the sphere’s tangent axis, which are the normal and tangent at the reflector point y , given a representative specular path. The emitter point’s image position is computed by

$$d_v = \frac{1}{\frac{1}{f} - \frac{1}{d_r}}, h_v = -\frac{d_v}{d_r} h_r, \quad (6)$$

where f is the focal length, h_r and d_r are distances from the emitter point to the focal axis and the tangent axis, respectively. h_v, d_v are distances from the image to the focal axis and the tangent axis, respectively. The image of the emitter point is determined by h_v, d_v . Notice that focal length f is positive for concave mirrors and negative for convex mirrors. The computed image position is connected to the shading point x , resulting in a direction \hat{o} . Then we rotate the SG from the current representative specular path to align its main axis with \hat{o} .

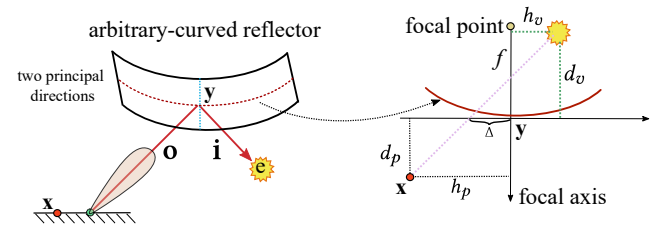


Figure 8: Left: an arbitrary-curved reflector can be treated as two separate spherical reflectors along with two principal directions (red and blue). Right: for each principal direction, an image is found, resulting in an offset Δ to the reflector point y along the principal direction. Combining the offsets along the two principal directions results in the final offset from y on the reflector surface.

Arbitrary-curved reflector. For an arbitrary-curved reflector, the view independent image position does not exist, so we treat the

curved surface as two separate spherical reflectors along the two principal curvature directions [Angelidis 2004], as shown in Fig. 8. For each principal direction, we compute the emitter point’s image and get an offset distance from the original reflector point y on the reflector by

$$\Delta = \frac{d_v h_p - d_p h_r}{d_r - d_p}, \quad (7)$$

where d_p and h_p are the distances from shading point x (projected to the plane of the principal curvature) to the tangent and focal axis, respectively.

For each principal direction, we compute the offset of intersection point along the two directions, and then sum up the two offset resulting in the final position:

$$\mathbf{y}' = \Delta_s \mathbf{s} + \Delta_t \mathbf{t} + \mathbf{y}, \quad (8)$$

where \mathbf{s} and \mathbf{t} are two principal directions. We use two perpendicular tangent axes of the tangent frame at reflector point y as the principal curvature directions. Then we get the SG at x by rotating the cached SG to the direction defined by \mathbf{xy}' .

The parallax compensation for arbitrary-curved reflector is only applicable for one-bounce reflection. For the multiple-bounce reflection case, we reduce it to the one-bounce case by regarding prior reflectors as spherical reflectors. Please see the supplementary material for details.

Discussion. We generate one cached point per triangle on the receiver and construct a point cloud. Note that our meshes consist of triangles of similar size. If the triangle size is arbitrary, we would suggest sampling points within each triangle as another option to generate the point cloud. At each cached point, we store a GMM generated with the method mentioned in previous subsections to represent the incoming radiance at this point. And also, we store the parallax-compensating information, such as d_v , h_v , reflector point y , and the tangent frame at y for each SG to do the parallax compensation during rendering.

3.6 Rendering

During rendering, a ray shot from the camera intersects with a triangle on the receiver at shading point x . We find the cached point associated with the intersected triangle and get the GMM, together with the parallax-compensating information for each SG at the cached point. Then we create a new GMM by warping the SGs as mentioned in Sec. 3.5, so that the GMM is parallax-compensated.

We combine the GMM and BRDF sampling via MIS to get the outgoing direction ω_x at location x . We use a constant probability (set as 0.5) to choose one of the sampling strategies. If the ray starting from x with direction ω_x intersects with the scene, we recursively perform the path guiding as the first bounce until no intersections are found. Note that we use the next event estimation at each bounce. We find that our method could work up to two intermediate glossy reflections. Since our method benefits a particular type of light transport, we leave the other light transport to path tracing.

Note that we add a protective smooth lobe that covers the whole hemisphere into each of the GMMs to ensure all possible directions can be sampled. This is an unbiased sampling strategy for path tracing, and we leave the proof of unbiasedness for future work.

4 RESULTS

We have implemented our method inside the Mitsuba renderer [Jakob 2010]. Our implementation only supports the “rough conductor” plugin as the reflector material for now. We compare against path tracing (PT), bidirectional path tracing (BDPT), practical path guiding (PPG) [Müller et al. 2017], specular manifold sampling (SMS) [Zeltner et al. 2020], and specular next event estimation (SNEE) [Loubet et al. 2020] with equal time, and a converged BDPT (rendered with 65536 samples per pixel (spp)) as reference. The implementations of the previous work are from the authors’ websites. All timings in this section are measured on a 4.20GHz Intel i7 (8 cores) with 16 GB of main memory. We use Root Mean Square Error (RMSE) to measure the difference between each method and the ground truth.

In our results, we use our method to handle point light sources and small sphere lights with less than 0.01 radian (radius/distance) and use the original rendering methods (PT) to handle the environment lighting. Scenes are designed to highlight one-bounce or multi-bounce reflective caustics. Some of the results include environment lighting for better appearance. All the timings for our method include the precomputation and the rendering.

Snail and Musa scenes. Fig. 1 shows a low-roughness snail ($\alpha = 0.01$) on a high-roughness floor ($\alpha = 0.3$) under a point light source. Fig. 9 shows a low-roughness Musa ($\alpha = 0.01$) casting caustics on a high-roughness background ($\alpha = 0.2$) under a point light source. In both figures, we compare our method, PT, BDPT and PPG [Müller et al. 2017] with equal time (20 minutes for the Snail scene and 12 minutes for the Musa scene). By comparison, our results have the lowest amount of noise. PPG produces better results than PT and BDPT on the Snail scene, but suffers from fireflies. In the Musa scene, the result of PPG is noisier than BDPT. The memory cost of PPG is higher than our method.

Reflective Surface. Fig. 10 shows a low-roughness (left: $\alpha = 0.005$, right: $\alpha = 0.05$) bumpy surface on top of a high-roughness floor ($\alpha = 1.0$). In this figure, we compare our method with SMS [Zeltner et al. 2020] with equal time. Our method produces a result similar to SMS when the floor has a very low roughness (0.005). Increasing the floor roughness has a significant impact on SMS, but not on our method, resulting in much better quality with equal time.

Three Bumpy Cylinder scene. In Fig. 11, we compare our result with SNEE [Loubet et al. 2020] with equal time (12 minutes). In this scene, we consider both one-bounce (“DS” and “SDS”) caustics and two-bounce (“DSS” and “SDSS”) caustics. Since SNEE can not handle multiple intermediate reflections, we notice apparent energy missing in their result. Thanks to the flexibility of the Spherical Gaussian approximation, our method supports arbitrary bounces in theory, although path cuts computation becomes slower when increasing the number of bounces.

Ring Scene. In Fig. 12, we validate the impact of our parallax compensation by comparing our results rendered with and without parallax compensation with equal sample count (spp = 16). The noise is significantly reduced with the parallax compensation, as parallax compensation improves the path guiding accuracy.

Sphere Scene. To show the impact of our representative specular paths (found with the Newton solver), we compare the results

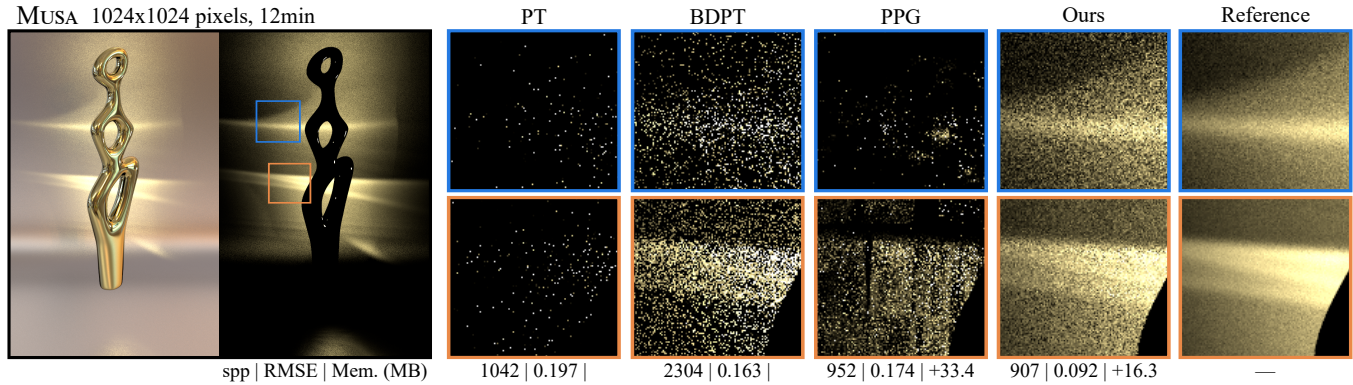


Figure 9: Equal-time comparison (12 minutes) among our method, PT, BDPT, PPG [Müller et al. 2017] on the Musa scene. We show the fully rendered result combining a point light and environment lighting (left). Zoomed-in insets are rendered with the point light only. Our precomputation takes 1.3s.

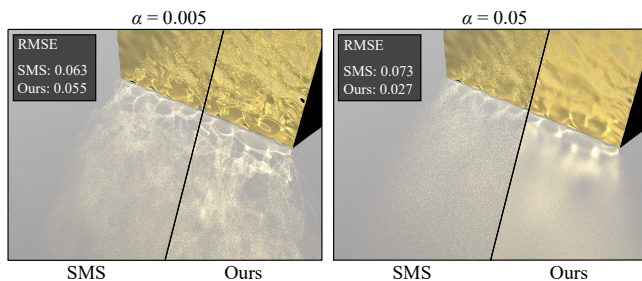


Figure 10: Comparison between our method and SMS [Zeltner et al. 2020] with equal time (7 minutes). This scene shows a bumpy plane (left: $\alpha = 0.005$, right: $\alpha = 0.05$) on a floor ($\alpha = 1.0$) under a point light. Our method produces a similar result as SMS at low roughness ($\alpha = 0.005$), and has much less noise than SMS for a slightly higher roughness ($\alpha = 0.05$). Our precomputation takes 18s for $\alpha = 0.005$ and 52s for $\alpha = 0.05$.

with and without the Newton solver in Fig. 13. The result with representative specular paths found by the Newton solver has much less variance and costs less time. And the memory cost is lower than without the Newton solver, since the path cuts have a more efficient pruning with the Newton solver.

5 DISCUSSION AND LIMITATIONS

We discuss the main limitations of our method as follows. For more detailed discussions, please refer to the supplementary material.

Performance. The number of path cuts significantly increases with the roughness of the reflector and increases exponentially with the number of bounces. This is a typical issue that also exists in other methods [Zeltner et al. 2020]. We recommend to use our method for reflectors with small roughness (from 0.005 to 0.05). Our method can handle up to two intermediate reflections in Fig. 11.

Extremely low-roughness materials. For a near-specular reflector with very small roughness, our SG approximation and parallax

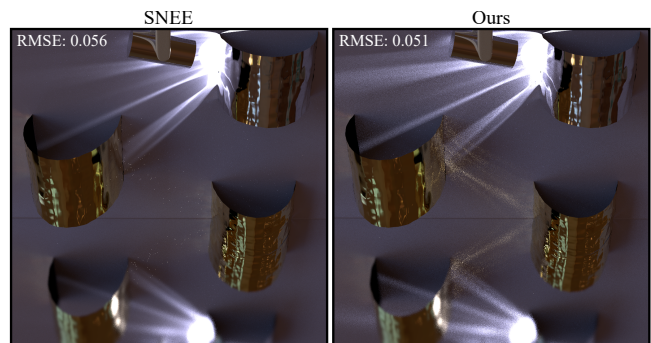


Figure 11: Equal-time (12 minutes) comparison between our method and SNEE [Loubet et al. 2020]. SNEE produces less noise on diffuse-specular (DS) caustics and specular-diffuse-specular (SDS) caustics, while our method can render diffuse-specular-specular (DSS) caustics and specular-diffuse-specular-specular (SDSS) caustics. Our precomputation takes about 10 seconds. This scene shows three bumpy cylindrical reflectors ($\alpha = 0.01$) above a glossy floor ($\alpha = 0.01$) and a diffuse background, under a tiny spherical emitter that can be approximated as a point light.

compensation is not accurate enough, and the results would show artifacts when the geometry is complex, as shown in Fig. 14.

Subdivision of the meshes. Our method inherits the drawback of the original path cuts method [Wang et al. 2020]. The meshes have to be densely tessellated for efficient path cuts finding, which usually involves subdivision of the meshes.

6 CONCLUSION

We have presented a practical path guiding approach for caustic rendering using representative specular paths. We extend the path cuts idea from pure specular light transport to glossy light transport and accumulate the contribution along the attained representative specular paths with Spherical Gaussian integral approximation, leading to an approximate incident radiance distribution with a Gaussian

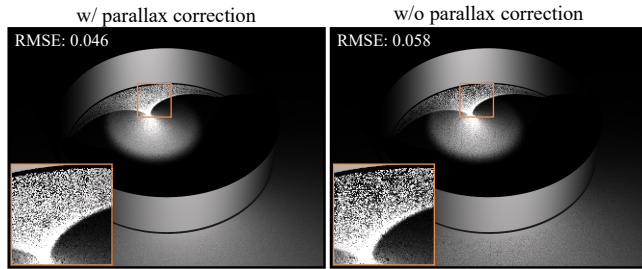


Figure 12: Equal-sample (spp = 16) comparison of our methods with and without parallax compensation. This scene shows a ring ($\alpha = 0.005$) on top of a floor ($\alpha = 0.5$) under a point light.

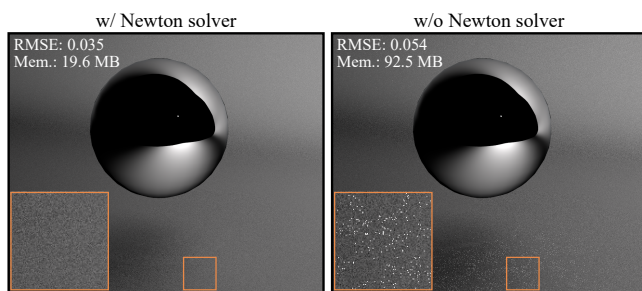


Figure 13: Equal-time (1 minute) comparison of our methods with and without the Newton solver (i.e., using triangle centers as the vertices of representative path). This scene shows a sphere ($\alpha = 0.001$) on a floor ($\alpha = 0.5$) under a point light.

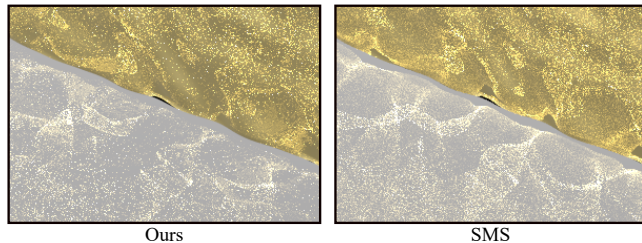


Figure 14: Failure case of our method with equal time (7 minutes). When the reflector has complex normal variation and very small roughness ($\alpha = 0.001$), our method does not show much benefit.

Mixture Model. Furthermore, we propose a spatial caching strategy and a parallax compensating strategy to improve performance and quality. Our method outperforms the previous works and allows for multiple intermediate reflections.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable suggestions. This work has been partially supported by the National Natural Science Foundation of China under grant No. 62172220, 62072284. Ling-Qi Yan is supported by gift funds from Adobe, Dimension 5 and XVerse.

REFERENCES

- Alexis Angelidis. 2004. Principal curvatures and directions. <http://www.cs.otago.ac.nz/postgrads/alexis/DiffGeom/node24.html>
- Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline Deep Importance Sampling for Monte Carlo Path Tracing. *Computer Graphics Forum* 38, 7 (2019), 527–542.
- Toshiya Hachisuka and Henrik Wann Jensen. 2009. Stochastic Progressive Photon Mapping. *ACM Transactions on Graphics* 28, 5 (2009).
- Johannes Hanika, Marc Droske, and Luca Fascione. 2015a. Manifold Next Event Estimation. *Computer Graphics Forum* (2015).
- Johannes Hanika, Anton Kaplanyan, and Carsten Dachsbacher. 2015b. Improved Half Vector Space Light Transport. *Computer Graphics Forum* 34, 2 (2015), 65–74.
- Sebastian Herholz, Oskar Elek, Jiri Vorba, Hendrik Lensch, and Jaroslav Krivanek. 2016. Product Importance Sampling for Light Transport Path Guiding. In *Eurographics Symposium on Rendering*.
- Wenzel Jakob. 2010. Mitsuba renderer. <http://www.mitsuba-renderer.org>.
- Wenzel Jakob and Steve Marschner. 2012. Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport. *ACM Trans. Graph.* 31, 4 (July 2012), 58:1–58:13.
- Henrik Wann Jensen. 2001. A Practical Guide to Global Illumination Using Photon Mapping. In *SIGGRAPH 2001 Course Notes CD-ROM*. Association for Computing Machinery, ACM SIGGRAPH. Course 38.
- Anton S. Kaplanyan and Carsten Dachsbacher. 2013. Adaptive Progressive Photon Mapping. *ACM Transactions on Graphics* 32, 2 (April 2013), 16:1–16:13.
- Anton S. Kaplanyan, Johannes Hanika, and Carsten Dachsbacher. 2014. The Natural-Constraint Representation of the Path Space for Efficient Light Transport Simulation. *ACM Transactions on Graphics* 33, 4 (July 2014).
- J. Laurijsen, R. Wang, Ph. Dutré, and B.J. Brown. 2010. Fast Estimation and Rendering of Indirect Highlights. *Computer Graphics Forum* 29, 4 (2010), 1305–1313.
- Zehui Lin, Sheng Li, Xinlu Zeng, Congyi Zhang, Jinzhu Jia, Guoping Wang, and Dinesh Manocha. 2020. CPPM: Chi-Squared Progressive Photon Mapping. *ACM Trans. Graph.* 39, 6, Article 240 (Nov. 2020), 12 pages.
- Guillaume Loubet, Tizian Zeltner, Nicolas Holzschuch, and Wenzel Jakob. 2020. Slope-Space Integrals for Specular Next Event Estimation. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 39, 6 (dec 2020).
- MJP. 2016. SG Series Part 4: Specular Lighting from an SG Light Source. <https://therealmjp.github.io/posts/sg-series-part-4-specular-lighting-from-an-sg-light-source/>
- Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum (Proceedings of EGSR)* 36, 4 (June 2017), 91–100.
- Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2018. Neural Importance Sampling. *arXiv preprint arXiv:1808.03856* (2018).
- Florian Reibold, Johannes Hanika, Alisa Jung, and Carsten Dachsbacher. 2018. Selective Guided Sampling with Complete Light Transport Paths. *ACM Trans. Graph.* 37, 6, Article 223 (dec 2018), 14 pages. <https://doi.org/10.1145/3272127.3275030>
- Lukas Ruppert, Sebastian Herholz, and Hendrik PA Lensch. 2020. Robust fitting of parallax-aware mixtures for path guiding. *ACM Trans. Graph.* 39, 4 (2020), 147–1.
- Ryusuke Villemin Sebastien Speierer, Christophe Hery and Wenzel Jakob. 2018. Caustic Connection Strategies for Bidirectional Path Tracing. Pixar Technical Memo 18-01.
- Eric Veach and Leonidas Guibas. 1994. Bidirectional Estimators for Light Transport. In *Fifth Eurographics Workshop on Rendering*. Darmstadt, Germany, 147–162.
- Eric Veach and Leonidas J. Guibas. 1997. Metropolis Light Transport. In *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, Vol. 31. 65–76.
- Jiří Vorba, Ondřej Karlik, Martin Šik, Tobias Ritschel, and Jaroslav Krivanek. 2014. On-line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)* 33, 4 (aug 2014).
- Beibei Wang, Miloš Hašan, and Ling-Qi Yan. 2020. Path Cuts: Efficient Rendering of Pure Specular Light Transport. *ACM Trans. Graph.* 39, 6, Article 238 (Nov. 2020), 12 pages.
- Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. 2009. All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance. *ACM Trans. Graph.* 28, 5 (Dec. 2009), 1–10.
- Kun Xu, Yan-Pei Cao, Li-Qian Ma, Zhao Dong, Rui Wang, and Shi-Min Hu. 2014. A practical algorithm for rendering interreflections with all-frequency brdfs. *ACM Transactions on Graphics (TOG)* 33, 1 (2014), 1–16.
- Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. 2020. Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints. *ACM Trans. Graph.* 39, 4, Article 149 (jul 2020), 15 pages.