

Reconstructing the Indirect Light Field for Global Illumination

Jaakko Lehtinen
NVIDIA Research

Timo Aila
NVIDIA Research

Samuli Laine
NVIDIA Research

Frédo Durand
MIT CSAIL

Abstract

Stochastic techniques for rendering indirect illumination suffer from noise due to the variance in the integrand. In this paper, we describe a general reconstruction technique that exploits anisotropy in the light field and permits efficient reuse of input samples between pixels or world-space locations, multiplying the effective sampling rate by a large factor. Our technique introduces visibility-aware anisotropic reconstruction to indirect illumination, ambient occlusion and glossy reflections. It operates on point samples without knowledge of the scene, and can thus be seen as an advanced image filter. Our results show dramatic improvement in image quality while using very sparse input samplings.

Keywords: light field, reconstruction, indirect illumination, ambient occlusion, defocus, motion blur

Links: [DL](#) [PDF](#) [WEB](#) [CODE](#)

1 Introduction

When generating digital images by simulating light transport, each pixel is computed as an integral of a multidimensional radiance function whose dimensions range over image xy , the lens (depth of field), time (motion blur), incident angle (indirect illumination and glossy reflection), and light source (area lighting). Such integrals are typically computed using Monte Carlo sampling. Evaluating the indirect illumination entails sampling the radiance over the hemisphere at each visible point, and is notoriously prone to noise due to the high variance of the integrand caused by visibility, texture, and illumination. We seek to generate high-quality images featuring glossy reflections and indirect illumination by intelligently reusing the samples drawn by a stochastic renderer at only a few samples per pixel.

We cast the problem as that of reconstructing an indirect light field at shading points, based on samples recorded during an initial coarse sampling step. The reconstructed light field is then used in a standard BRDF sampling step to generate the final image. We assume proper importance sampling is applied, but require no control over how the samples are generated. Throughout the paper, we assume direct illumination is computed using other techniques.

We build on recent work that has provided a solid understanding of the local anisotropies exhibited by the temporal light field [Durand et al. 2005; Ramamoorthi et al. 2007; Egan et al. 2009; Soler et al. 2009; Egan et al. 2011b]. But while previous sampling schemes and anisotropic reconstruction filters can capture the essence of the

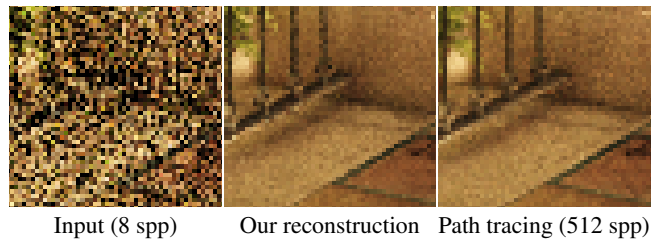


Figure 1: We demonstrate that even very noisy and sparse samplings can contain surprisingly much information about light transport and visibility in the scene. In this example, diffuse indirect illumination was reconstructed from 8 input samples per pixel.

integrand better than naive point sampling in situations such as motion blur, depth of field, and area lighting, the case for general indirect illumination remains unsolved due to complex visibility, highly irregular input sampling, and glossy materials.

We contribute a method for adapting to the sampling rate of non-uniform, sparse, uncontrolled input in both space and angle, a spatio-angular anisotropic reconstruction method for filtering radiance from the sparse samples, and a robust method for reasoning about occlusion based on only the input rays. In contrast to a large body of previous work aiming to reduce noise in global illumination, we do not utilize the scene geometry in the reconstruction pass. This gives us the crucial advantage that performance is only weakly dependent on the scene. In this sense, our algorithm can be seen as an advanced image filter.

Our algorithm produces megapixel images with glossy reflection, global illumination, and ambient occlusion in a few minutes. The results are of dramatically higher quality than was possible using earlier reconstruction methods, and comparable to using hundreds of paths per pixel (Figure 1) both visually and in terms of PSNR.

1.1 Related Work

Anisotropic reconstruction Image synthesis involves integrating a high-dimensional function across the image and auxiliary dimensions (time, lens, incident angle, etc.). A body of recent work concentrates on *anisotropy*, i.e., the fact that the signal often varies slowly along certain non-axis-aligned subspaces. Fourier analysis elegantly reveals that the signal is locally effectively of a lower dimension¹ in these cases [Durand et al. 2005; Egan et al. 2009]. This means that samples drawn at one pixel tell us something about the integrand at other pixels as well, which enables reconstruction of better images from the same samples.

The recent trend in anisotropic reconstruction can be traced back to the Multidimensional Adaptive Sampling and Reconstruction (MDAS) algorithm of Hachisuka et al. [2008]. While numerous earlier techniques had addressed the generation of samples according to the expected variance in the integrand (importance sampling and adaptive sampling methods), Hachisuka et al. concentrated on

This is an author-prepared preprint. The definitive version appears in the ACM Digital Library.

¹For example, the 4D integrand over the image and the lens that corresponds to a diffuse fronto-parallel planar object that is out of focus is actually only two-dimensional.

reconstruction. The authors measure the local anisotropy from the samples using a structure tensor, and use it for locally warping the distance metric used for reconstructing the integrand using a high-dimensional k -nearest-neighbor search (k -NN). This can yield significant benefits, but unfortunately, noise and texture may confuse the anisotropy estimator.

Egan et al. [2009] analyzed the anisotropy in the spacetime integrand of motion blur from first principles through Fourier analysis, and used predictions of the local spectra for driving adaptive sampling. A sheared filter that allows sharing of samples between pixels was then adapted to the predicted spectra. Similar analysis and algorithms were presented for soft shadows and directional occlusion by the same authors [Egan et al. 2011b; Egan et al. 2011a]. While the analysis is successful in driving adaptive sampling, the sheared reconstruction filter is less efficient when the direction of anisotropy changes, either continuously or discontinuously, across the image. This issue is evident, e.g., in occlusion discontinuities, and force a fallback to QMC sampling. These issues are addressed by Lehtinen et al. [2011], whose algorithm accounts for both the individual anisotropy information of each sample and discontinuities due to visibility, producing high-quality results for simultaneous motion blur, depth of field, and area lighting. The goal of the present work is to extend this approach to the more challenging and unstructured case of indirect illumination.

Sample-based algorithms Several techniques, e.g., anisotropic diffusion [McCool 1999] or a (cascade of) cross-bilateral filter(s) [Dammertz et al. 2010], post-process the samples or pixels generated in an initial rendering pass to reduce noise. Sen and Darabi [2012] improve the quality of cross-bilateral filters by computing the weights adaptively for each pixel based on apparent dependencies between domain variables, scene features, and sample colors. While their method can, for example, detect that samples in some pixels are heavily affected by time, it does not know the magnitude or direction of the motion. Overbeck et al. [2009] and Rousselle et al. [2011] describe methods that maintain a basis-representation of the image and adaptively request more samples from the renderer to the areas that exhibit high variance.

Shading reuse Several algorithms reuse shading results, radiance or irradiance, while determining visibility with the actual scene [Ward et al. 1988; Bala et al. 1999; Bekaert et al. 2002; Gassenbauer et al. 2009]. Sampling is often adaptive and driven by an error heuristic. In contrast, we do not adapt the sampling but instead post-process the samples given to us by a separate renderer, run no light- or scene-dependent preprocessing stages, and do not require the scene to be resident in the reconstruction phase. This makes our algorithm much less intrusive for the original renderer.

Geometric resampling Our method for reasoning about occlusion based on samples from the initial sampling has also connections to ray tracing point-based geometry [Schauffler and Wann Jensen 2000; Christensen 2008; Ritschel et al. 2009]. These algorithms convert the scene to a point-based representation as a pre-process, and then compute the solution using that representation. In contrast, we obtain a sparse set of path segments from a renderer (which uses whatever representation internally), treat the segments as samples of the indirect light field, and upsample the solution accounting for angular effects of gloss.

Photons and VPLs Some methods generate a cloud of point lights (called *photons* or *virtual point lights* or VPLs) that represents the entire indirect light field [Wann Jensen 1996; Keller 1997; Walter et al. 2005]. They can be seen as a resampling of the in-

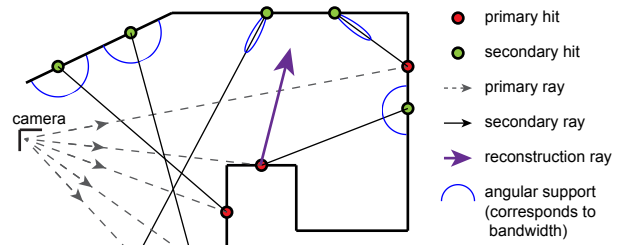


Figure 2: Input data consists of primary and secondary hit points with corresponding surface normals, the radiance along the secondary rays, and an angular bandwidth estimate of the reflectance function at the secondary hits.

tegrand to a form more amenable to final gathering. The original scene is still necessary for visibility determination.

2 Reconstruction of Indirect Light Field

We seek to render high-quality global illumination while only tracing a small number of rays across the hemisphere at each visible point. We reuse these rays across many pixels to reconstruct, at each visible point, an upsampled version of the incoming light field. The reuse is facilitated by the anisotropy exhibited by the incident light field [Durand et al. 2005]: provided we can quantify visibility and angular effects, radiance leaving a point along a ray provides information about radiance carried by other, nearby rays. For the entire paper, we disregard direct illumination, and assume it to be computed separately.

Algorithm overview In a first pass, we use a standard path tracer, but only draw few samples per pixel, typically 4–16. We store the rays cast from the visible points into a spatial hierarchy, indexed by their hit point (Section 2.1). Each ray is a sample of the indirect light field.

After this, we compute, for each input sample, a spatial radius of influence that determines how far it will contribute to light field reconstructions, adapting to the density of input samples in both space and angle (Section 2.2). We describe a modified k -NN technique that adapts to the density while accounting for tradeoffs between space and angle as determined by BRDF gloss. Furthermore, we need to treat visibility carefully to avoid the “fattening” of small features and silhouettes when performing reconstruction. We introduce an important step that shrinks the samples if they violate the visibility of input rays (Section 2.3).

In the second pass, we evaluate the shading equation at each visible point using a high-sampling-rate reconstruction of the incoming hemisphere. The key operation is a reconstruction of the 4D incoming light field along a given ray. For this, we find all input rays whose spatial radius overlaps the ray and group them into *apparent surfaces* using a visibility heuristic (Section 2.3.2). This handles two dimensions of the light field. We then reconstruct radiance from the remaining samples closest in angle in order to handle the remaining two dimensions (Section 2.4).

2.1 Input and Output

The radiance L from point \mathbf{p} towards the camera \mathbf{c} is computed by

$$L(\mathbf{p} \rightarrow \mathbf{c}) = \frac{1}{\pi} \int L_{\text{in}}(\mathbf{p} \leftarrow \omega) f_r(\mathbf{p}, \omega \rightarrow \omega_c) \cos \theta \, d\omega, \quad (1)$$

where $L_{in}(\mathbf{p} \leftarrow \omega)$ is the light field incident to \mathbf{p} from direction ω , $f_r(\mathbf{p}, \omega \rightarrow \omega_c)$ is the reflectance function (we disregard transmittance for simplicity), and the integral is taken over the positive hemisphere around the normal of \mathbf{p} .

The first pass computes and stores samples of $L_{in}(\mathbf{p} \leftarrow \omega)$ using a path tracer. Each input sample s_i corresponds to a secondary ray, and consists of the tuple

$$s : \{x, y, u, v, t, \omega\} \mapsto \{L, \mathbf{o}, \mathbf{h}, \mathbf{n}_o, \mathbf{n}_h, \mathbf{v}_o, \mathbf{v}_h, \kappa\},$$

where L is radiance, \mathbf{o} and \mathbf{h} are the origin and hit point of the secondary ray, $\mathbf{n}_{o,h}$ and $\mathbf{v}_{o,h}$ are their corresponding normals and motion vectors. The numbers $\{x, y, u, v, t, \omega\}$ are the image, lens, and time coordinates, and the incident direction in which $L(\mathbf{o} \leftarrow \omega)$ was sampled. To support angular variation of outgoing illumination on glossy surfaces, we also store a measure of the angular bandwidth determined from the outgoing slice of the BRDF at the hit point. We use an unnormalized spherical von Mises-Fisher distribution

$$A_i(\omega) = \exp(\kappa_i (\omega_i \cdot \omega) - \kappa_i) \quad (2)$$

and store κ , its concentration parameter. In particular, $\kappa = 0$ for a diffuse surface and increases with gloss (Section 2.5). Together, the input consists of an irregular set of ray segments that coarsely samples the indirect light field (Figure 2).

Given the samples recorded in the first pass, the second pass evaluates

$$L_{out}(\mathbf{p} \rightarrow \mathbf{c}) \approx \frac{1}{\pi} \frac{1}{N} \sum_{j=1}^N \text{RECONSTRUCT}(\mathbf{p}, \omega_j) f_r(\mathbf{p}, \omega_j \rightarrow \omega_c) \cos \theta \quad (3)$$

for each primary hit in the input samples, with $N \gg 1$. The total number of *reconstruction rays* per pixel is thus $N \times n$, where n is the number of primary hits per pixel. The function $\text{RECONSTRUCT}(\mathbf{p}, \omega)$ performs the upsampling of the incident light field based on the input samples.

Assumptions We assume that proper importance sampling is applied in the input stage. This guarantees that the distribution of rays generated during the final image reconstruction is similar to that in the input. Because we cannot hope to recover features not present in the input, we make the assumption that the input samples faithfully represent the true indirect light field, i.e., that they capture its frequency content. As the density of the input samples is a complex function of the scene, its materials, and the sampler used in the original renderer, we do not reason about it a priori, but instead discover the local sampling rate by a modified k -nearest-neighbors search (Section 2.2). We discuss possible noise in the radiance values L in Section 4.

We employ a standard light field parameterization, where rays are encoded by their intersections with two planes: an st plane, and an uw plane at unit distance from the former. We use the variant where uw coordinates are measured relative to the st coordinates [Chai et al. 2000; Durand et al. 2005] and essentially form a linearized angle. As we seek to reconstruct rays within arbitrary locations and directions in the scene, we internally store the ray segments using their 3D hit point, and dynamically reparameterize the light field around our reconstruction rays [Isaksen et al. 2000].

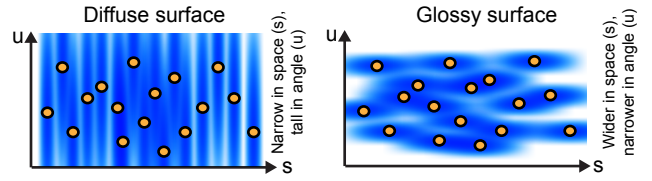


Figure 3: Shapes of reconstruction filters in a local parameterization around the samples (shown in blue). We use the known angular bandwidth to drive a modified k -nearest-neighbors search, where each sample finds the k th sample that falls within acceptable angular limits. This results in spatially narrow samples on diffuse surfaces (left), and wide samples on glossy surfaces (right).

2.2 Sampling Rate

The spatial and angular distribution of the input samples is determined by the scene geometry and the reflectance functions of the primary hits, as the input renderer performs importance sampling according to them. In order to perform meaningful reconstructions, we have to adapt to this sampling rate in both space and angle. As our input carries information about the angular variation in form of the bandwidth estimate, it remains to balance the known angular support with the spatial support. Specifically, for diffuse surfaces, where outgoing radiance does not vary over angle, the radiance of the input sample can be reused for any reconstruction ray that passes near the sample in space, no matter how far in angle, provided visibility is accounted for. For glossy surfaces, where radiance varies over angle, we might need to go further in space to find an input sample whose direction better matches that of the reconstruction ray.

Along the lines of previous point-based rendering algorithms, we treat the input samples as circular disks (splats) in 3D on their individual tangent planes. We determine their size by analyzing the local sample density. To set the spatial support w_i for sample s_i , we search for its k th nearest neighbor in space around its secondary hit point \mathbf{h}_i , but only considering samples s_j whose direction ω_j falls close enough to ω_i . The angle threshold is determined from κ_i by requiring $A_i(\omega_j) \geq 0.5$.

To illustrate the effect, first consider an almost diffuse surface. As the radiance varies slowly over outgoing direction, A_i is almost constant in angle (angular bandwidth is low and consequently κ is small), and the search simplifies to a regular k -NN in space. Consequently, the spatial support will be small (Figure 3, left). On a more glossy surface, the angular bandwidth is higher, causing the k -NN search to disregard spatially close-by samples in favor of those closer in angle. Consequently, the support will be wider in space (Figure 3, right).

We use a metric that penalizes movement in the direction of the normal to encourage the search for support to favor samples lying on locally flat regions. Specifically, we use a Mahalanobis distance that squashes the distance along the normal of each splat by a factor of α (we use $\alpha = 3$). This is implemented simply as computing the Euclidean length of $(\mathbf{p} - \mathbf{q}) + (\alpha - 1)((\mathbf{p} - \mathbf{q}) \cdot \mathbf{n})\mathbf{n}$, where \mathbf{n} is the normal.

2.3 Visibility

In addition to spatial filtering, the circular splats are used for resolving visibility. When performing reconstruction, we need to gather all input samples whose spatial support is intersected by the reconstruction ray, reminiscent of previous point-based rendering algo-

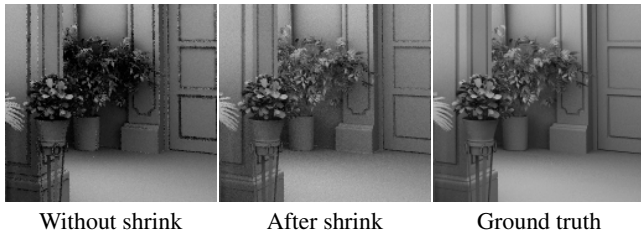


Figure 4: We need to shrink some samples so that the representation remains faithful to the original sampling. This is a closeup of ambient occlusion in SAN MIGUEL, shown fully in Figure 8.

rithms [Schaufler and Wann Jensen 2000]. To accelerate this, we organize the secondary hit points $\{\mathbf{h}_i\}$ into a motion bounding volume hierarchy [Glassner 1988] using the streaming algorithm of Kontkanen et al. [2011]. During reconstruction, the hierarchy is used in a standard fashion by walking the BVH and intersecting the reconstruction ray with the splats in the intersected leaf nodes.

2.3.1 Ensuring Consistent Visibility

In order for our reconstructions to be accurate, the visibility function induced by the splats should resemble true occlusion in the scene. However, the preceding k -NN procedure does not guarantee consistency with the original sampling. Small geometric features (e.g. leaves of a plant) might only be hit by few input samples, and when these get enlarged by the k -NN procedure, some rays present in the input might erroneously intersect them.

To force the representation to be consistent with the input, we observe that the input rays carry exact information about point-wise visibility in the scene: we know, with certainty, that a ray from \mathbf{o}_i terminated at \mathbf{h}_i without being blocked in between. Inspired by this, we retrace all the input rays using the BVH and shrink splats that intersect rays they should not. Finally, the bounding boxes of the BVH nodes are recomputed to accommodate the new sizes. This concludes the preprocessing of the input samples.

The resulting splats faithfully represent the visibility as determined by the original secondary rays, while adapting to their local density and causing no spurious occlusions, cf. Figure 4.

2.3.2 Resolving Occlusion

Treating the splats as simple opaque occluders leads to unacceptable results, and more sophisticated techniques are required for determining which splats are part of the same surface [Zwicker et al. 2001] to allow filtering between samples.

When reconstructing the radiance for a reconstruction ray, we first traverse the BVH and collect all input samples whose spatial support overlaps the reconstruction ray. Once these have been found, the samples are sorted according to distance from the ray origin, and then grouped into *apparent surfaces*. The grouping proceeds greedily by adding the next sample to the open surface as long as it does not conflict with any of the samples already in the surface in terms of visibility. When a visibility conflict is detected (see below), a new surface is started. Note that the operation is entirely local: it does not make use of samples whose supports are not intersected by the reconstruction ray.

To detect visibility conflicts, we generalize the heuristic of Lehtinen et al. [2011] that detects crossings of trajectories in the light field for defocus and motion blur. A geometric interpretation of their test is that two samples belong to different surfaces if an *event*

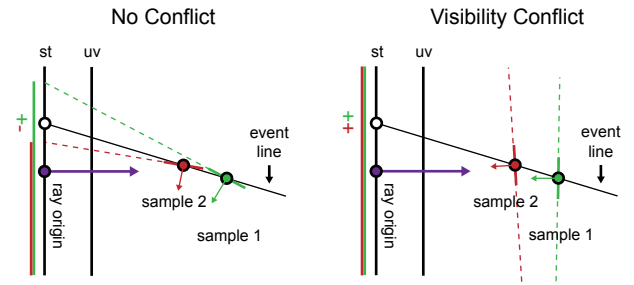


Figure 5: Determining visibility consistency for a reconstruction ray (purple). **Left:** The event line drawn through the hit points of samples 1 and 2 intersects the local st plane (white circle) outside the region of space where both samples face forward. Consequently their mutual visibility is determined to be consistent and the samples belong to the same apparent surface. **Right:** The event line intersects the lens in the positive half-space of both samples. The samples are determined to belong to different surfaces.

line, defined as the line that goes through the scene points corresponding to the samples, intersects the lens. However, when reconstructing the incident light field on the surfaces, the concept of lens is meaningless. In addition, our rays originate from and terminate at arbitrary 3D points in the scene.

Our visibility heuristic works by considering the positive half-spaces, i.e., points \mathbf{x} where $(\mathbf{x} - \mathbf{h}) \cdot \mathbf{n}_h > 0$, defined by the hit point and normal of each input sample. Visibility conflicts are detected by computing the intersection of the st plane and the event line formed by connecting the two endpoints \mathbf{h}_i and \mathbf{h}_j . If the intersection point lies within the positive halfspaces of both samples, a conflict is declared, cf. Figure 5.

This approach detects a conflict when two surfaces that are on top of each other, but allows blending of samples that come from the same locally flat, convex, or concave surface.

2.4 4D Reconstruction

Once the visibility heuristic has been evaluated and the samples grouped into apparent surfaces, the samples in the nearest apparent surface are input rays whose hit points are determined to be visible from the origin of the reconstruction ray. It now remains to reconstruct the radiance based on their locations and directions. This requires adapting to the anisotropy and bandwidth of the light field around the reconstruction ray.

Consider a situation where a reconstruction ray intersects the spatial supports of three samples (Figure 6, left). In a local two-plane parameterization $stuv$ constructed around the reconstruction ray, the reconstruction ray corresponds to a point at the origin (Figure 6, middle), and each input ray corresponds to a single point. For each input sample, the set of *all* rays that intersect the spatial support forms a set with planar boundaries (Figure 6, middle). Clearly, the reconstruction ray belongs to this set.

For a diffuse surface, where angular bandwidth is zero, reconstruction is simple: we evaluate a spatial filter (tent in our implementation) at the intersection of the reconstruction ray and the splat, and take a weighted average among all samples. In a local coordinate system centered at each sample (Figure 3, left), this corresponds to a separable 4D filter that is the product of a tent in space and a constant in angle (corresponding to the zero angular bandwidth), and reparameterization to the $stuv$ coordinates around the reconstruction ray shears this filter into the anisotropic shape shown (Figure 6, middle).

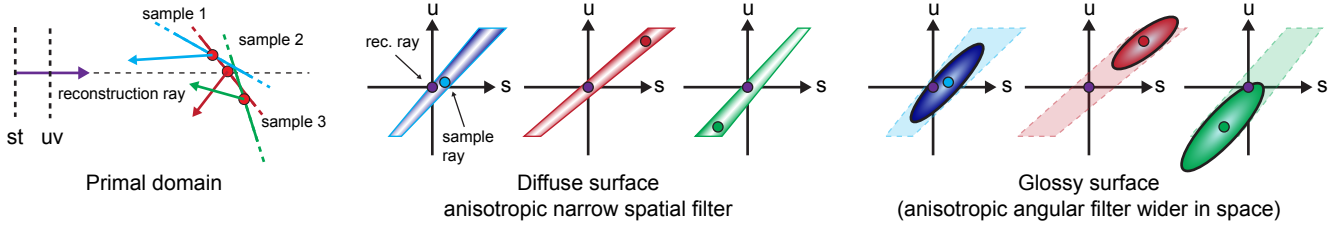


Figure 6: 4D reconstruction from visible samples. **Left:** The reconstruction ray (purple) intersects three samples. **Middle:** If the samples lie on a diffuse surface, their angular bandwidth is zero. The samples are filtered by evaluating a spatial tent filter at the intersection of the ray with the splat. **Right:** Glossy surface (see text).

Radiance leaving glossy surfaces varies over outgoing angle, and we need to account for it by reusing samples whose direction is similar to that of our reconstruction ray. The non-zero angular bandwidth of glossy samples leads to a wider spatial support as a result of the bandwidth-aware k -NN search. In a local coordinate system oriented with each sample, this is visible as a narrower angular filter that is wider in space (Figure 3, right). When this filter is transformed into the $stuv$ coordinate system, we obtain anisotropic shapes oriented similarly to the diffuse filters (Figure 6, right). We perform reconstruction by evaluating the transformed filters at the $stuv$ origin, and picking the sample whose filter has the largest magnitude. This can be seen as an anisotropic nearest neighbor filter.

We use nearest neighbor for glossy surfaces to mitigate the effects of the highly non-uniform angular distribution of our input samples. Using a linear angular filter noticeably shifts the reconstruction towards the denser sampling, which is visible in the final image as shifting highlights: angular shifting results in spatial shifts after propagation. This is not necessary for diffuse reflectance where outgoing angle plays no role.

Discussion Our filtering scheme accounts for both spatial and angular bandwidth of the outgoing light field at the input samples. The spatial and angular dimensions are balanced by the k -NN algorithm used for determining the spatial supports for the samples. Effects such as curvature and the anisotropy of the incident light field (for instance, near-field illumination) at the sample points play a role in how space and angle interact in the outgoing light field, potentially causing additional shears [Durand et al. 2005]. As our samples lie on the actual curved surfaces, our input as a whole contains these effects. But as our input does not carry information about the curvature and the properties of the incident light field, the reconstruction filters of the individual samples cannot account for the additional shears. Regardless, our treatment allows faithful reconstruction of glossy surfaces reflecting off other glossy surfaces. This is demonstrated in the results.

When the reconstruction ray points towards a fronto-parallel surface with uniform angular sampling, our reconstruction procedure corresponds to performing nearest neighbor in space that has been warped using a Mahalanobis distance aligned with the anisotropy of the light field — this renders the isocontours of the filters locally isotropic.

2.5 Implementation Details

In our prototype implementation, we support the Torrance-Sparrow BRDF model with the Blinn microfacet distribution (the “Metal” material in PBRT), and take $\kappa = 4\sqrt{1/r}$ where r is the roughness parameter of the Blinn distribution [Pharr and Humphreys 2010]. The formula is chosen empirically.

Since PBRT treats geometric primitives as two-sided, we need to do the same. When a reconstruction ray hits a back-facing splat, it contributes to visibility similarly to front-facing splats. However, back-facing splats contribute to radiance only if all splats of the first surface are back-facing, otherwise their radiance contribution is ignored.

In rare occasions, a reconstruction ray may hit a surface that consists of only one or two samples. Although the k -NN search tries to guarantee that this never happens, the subsequent shrinking may affect the result. This happens, for instance, inside dense foliage where individual leaves may be hit by only one input sample. To avoid over-estimating the occlusion, we empirically reduce the splat size by $\frac{1}{2}$ for single-sample surfaces and by $\frac{1}{3}$ for two-sample surfaces when such undersampling is detected. This treatment affects less than 1% of the reconstruction rays in our scenes.

The reconstruction results are not particularly sensitive to the choice of k , and we use $k = 12$ in all examples of this paper.

3 Test Results

We have implemented our algorithm as a standalone library that takes a buffer of samples as input, and modified PBRT [Pharr and Humphreys 2010] to generate the input. We have produced both a multicore CPU and a GPU implementation of the reconstruction algorithm, and our test platform is a 3.2GHz quad core Intel Core i7 and NVIDIA GTX480. All result images were rendered at 1280×720 (720p).

In this section, we first test reconstruction of indirect illumination in diffuse scenes, followed by glossy reflections. We then perform more targeted tests and demonstrate that our occlusion heuristic allows faithful reproduction of visibility. We use ambient occlusion as a test case because its quality is directly determined by the accuracy of visibility. Finally, we show results for defocus and motion blur. We focus on the quality of the reconstructed images, and postpone discussion about scalability, memory consumption, and extensions to Section 4. We compare our results to methods that, similarly to us, reconstruct the image based on a small number of “fat” samples [Lehtinen et al. 2011; Dammertz et al. 2010; Sen and Darabi 2012].

The auxiliary material contains all of the uncompressed images that are shown or referred to in this section.

3.1 Indirect Illumination in Diffuse Scenes

We reconstruct multi-bounce diffuse global illumination in SAN MIGUEL lit with an area light source (Figure 7). The input contains 8 samples per pixel (leftmost inset). We reconstruct the incident light field over the hemisphere at visible points, and integrate to produce the result (third inset from the left), as per Equation 3. We

pay no special attention to the fact that the radiances L of the input samples contain noise due to the multiple bounces rendered by the path tracer. Our reconstruction is more true to ground truth (second left) than the comparison methods, A-Trous wavelets [Dammert et al. 2010] (fourth inset from left) and random parameter filtering (RPF, last one on right) [Sen and Darabi 2012]. We use both comparison methods for reconstructing incident irradiance, i.e., before multiplication by texture.

Rendering the 8 spp input with PBRT takes 36.6s. Our reconstruction with 256 reconstruction rays per pixel on the GPU takes 189.5s. Of this, pre-processing, including building the BVH, sizing the samples using k -NN, and shrinking the samples using the input rays takes 9.5s. The rest is spent on reconstruction rays. This is considerably more efficient than rendering the similar-quality 512 spp image with PBRT, which takes 3910.5s, or 18x longer than combined input+reconstruction. Our 256 spp result exhibits significantly less noise than PBRT’s 256 spp rendering, thanks to our spatial filtering of the samples. Notice that our ability to perform image antialiasing is limited by the number of primary hits in the input samples.

The supplemental video contains a short animation rendered from 8 input samples per pixel using 512 reconstruction rays per pixel. Each frame is processed in isolation; no temporal filtering of any kind is employed. The result demonstrates the temporal stability of our reconstruction even from severely noisy input.

Comparison methods A-Trous is an iterative approximation to a cross-bilateral filter, with the twist that when spatial support widens between iterations, the range filters tighten. This allows the application of a very large spatial filter without disproportionate blurring. The original implementation uses three range filters: color, normal, and world-space position. Since our input is very noisy, and we do not wish to clamp the input samples’ energies, we chose not to use color as a range filter (using it gave worse results). While the quality leaves some room for improvement, the A-Trous filter is efficient: our CPU implementation takes 58s in this scene.

Since no reference implementation of random parameter filtering (RPF) is available, we reimplemented it by following the authors’ detailed guidelines [Sen and Darabi 2011]. RPF is a sophisticated cross-bilateral filter that automatically tunes the weights for each feature, at each pixel, based on mutual information between domain variables, scene features (position, normal, texture, etc.), and radiance. Their input samples are almost identical to ours. Unfortunately, the original formulation of RPF makes heavy use of the color channel as a range filter, requiring special treatment for bright samples (“HDR clamp”), which essentially discards high-energy spikes. This loses a significant amount of energy, up to 50% in our tests. We modified their algorithms to reinsert the lost energy evenly to all samples within the pixel. Our reasonably tuned CPU implementation filters the image in 2895s, roughly in line with their results. We estimate a 10x possible speedup from a GPU port. We use the parameters recommended in the authors’ guidelines.

3.2 Glossy Reflection

Figure 9 shows MONKEY HEADS, a scene with three statuettes of decreasing gloss² on a glossy ground plane. The scene is lit by single-bounce indirect illumination from a distant point light source (direct illumination not shown), and consequently the input samples are noise-free. The input contains 8 samples per pixel. Our result is computed using 512 reconstruction rays per pixel. A-Trous and RPF results are shown below. We noticed the results for A-Trous

²The roughnesses are 0.01, 0.05 and 0.25 for the statuettes and 0.01 for the ground plane.

could be significantly improved by introducing an additional range filter over the normal of the secondary hit, which is conveniently present in our data. We present this extension in our comparison results in addition to the original algorithm.

Our results are significantly closer to ground truth than either comparison method both visually and in terms of PSNR (38.6dB vs. 31–33dB). The full-size images are provided in the auxiliary material for closer inspection. In particular, we reconstruct the position and sharpness of the reflections faithfully, thanks to the 4D treatment of the input samples that accounts for both space and angle. Figure 10 demonstrates the importance of accounting for angular effects. While the comparison methods produce smooth results, the reflections have shifted noticeably and are not as sharp as they should. Note that unlike the comparison methods, we use the rendering system (PBRT) to evaluate the BRDFs of the final bounce towards the eye, as per Equation 3.

While closely matching ground truth, our algorithm is currently 2.6x slower in this scene than the reference method. This is mainly due to the extreme geometric simplicity of the scene, which plays to our disadvantage because our algorithm scales weakly with scene complexity (Section 4). Our spatial hierarchy does not currently include angular subdivisions [Arvo and Kirk 1987], and we may therefore process redundant splats on glossy surfaces, where the spatial supports are larger (Section 2.2). Another possible improvement is discussed in Section 4. Regardless, the result demonstrates that the information required for reconstructing high-quality reflections is present in the sparse input, and that our algorithm correctly extracts it to produce a high-quality result.

3.3 Ambient Occlusion

Our algorithm can be easily used for rendering ambient occlusion by bypassing the radiance filter and thresholding the distance to the closest apparent surface found. Figure 8 shows our ambient occlusion reconstruction in SAN MIGUEL. Although the scene contains fine geometric features such as grooves and vegetation, our reconstruction from only 4 input samples per pixel is in very close agreement with the ground truth computed using 2048 samples per pixel. In this case, image antialiasing is particularly limited since the input contains only 4 primary hits per pixel. The GPU execution time was 145 seconds for 256 reconstruction rays per pixel at 720p.

In contrast to the ambient occlusion technique of Egan et al. [2011a], we do not resort to brute force Monte Carlo sampling of the original scene at all.

3.4 Motion and Defocus

Finally, we demonstrate the generality of our reconstruction by applying it to motion blur and defocus (Figure 11). Specifically, instead of secondary rays, we now store the primary rays cast from a lens, and reconstruct the incident temporal light field at different points on the lens instead of indirect lighting at scene points. We use the BUTTERFLIES dataset from Lehtinen et al. [2011] and compare against their publicly available implementation.

Our result is slightly more accurate, approx. 1 dB in terms of PSNR. The images are available in the supplemental material. Our better result is explained by our better adaptation to the input sampling rate. They use a worst-case radius derived from the dispersion of the sampling pattern, which tends to blur in-focus geometry slightly, but may still result in lack of support in regions of the temporal light field that contain features that are visible from only a small fraction of the lens and shutter interval. In contrast, our k -NN approach adapts to the local sampling density, leading to sharp results in well-



Figure 7: Diffuse global illumination in SAN MIGUEL. Only indirect illumination is shown. Our result upsamples the 8 spp incident light field using 256 reconstruction rays per pixel. The supplemental material contains the full images.

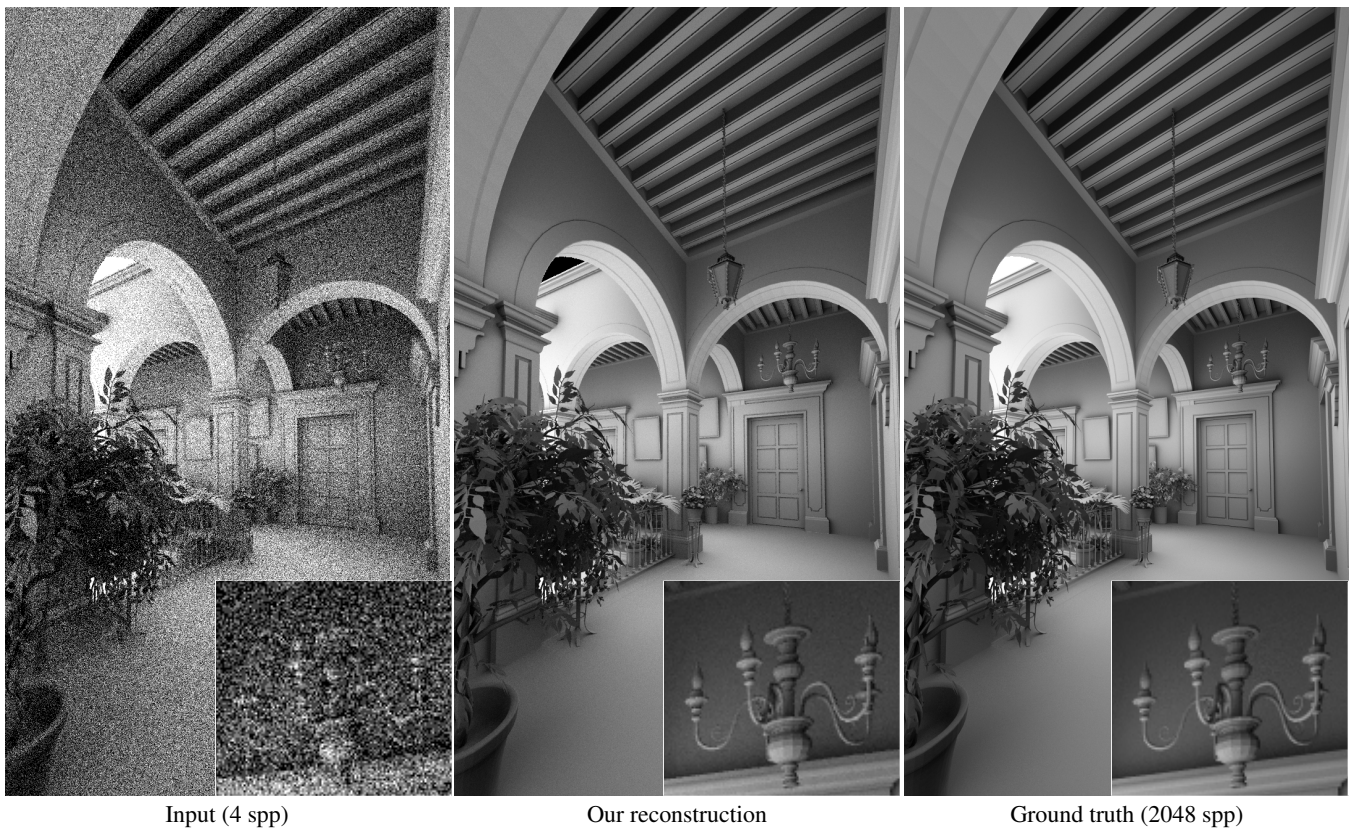


Figure 8: Our reconstruction of ambient occlusion from 4 input samples per pixel, compared to 2048 spp ground truth from PBRT.

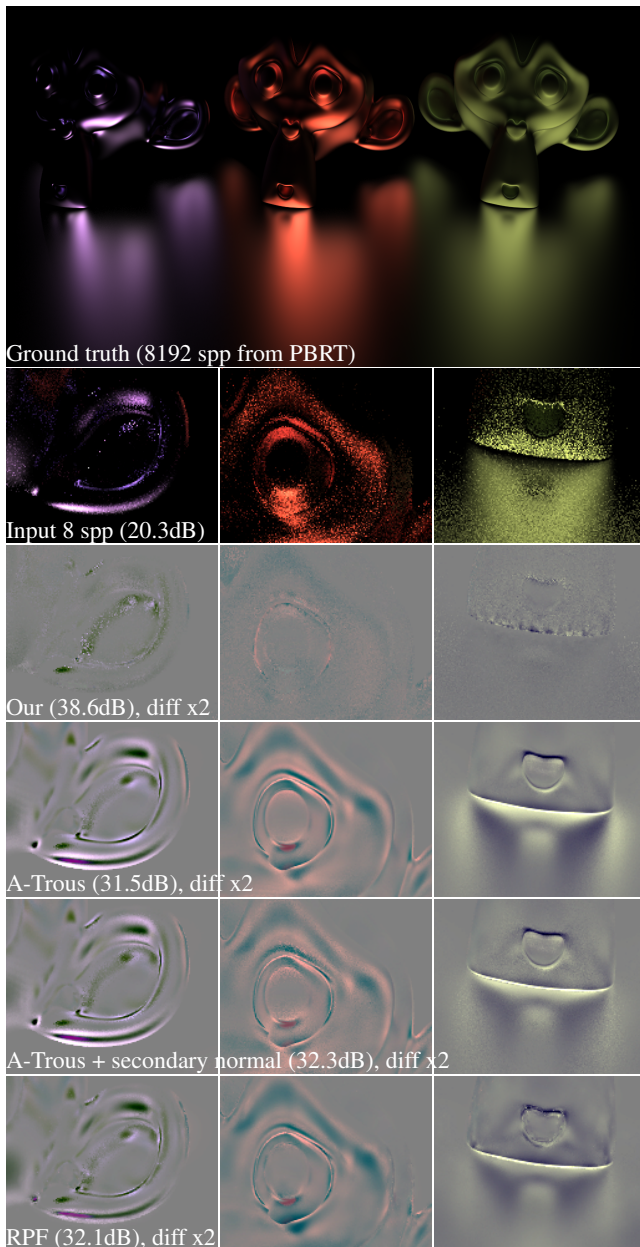
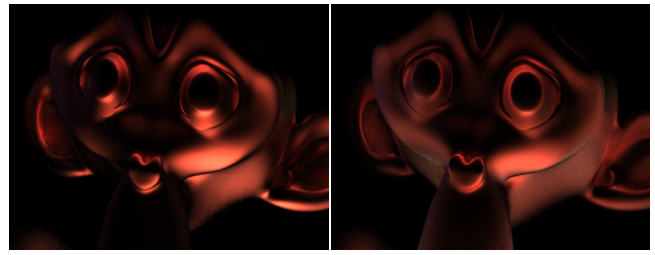


Figure 9: Glossy reflectance. The closeups show differences from the ground truth (multiplied by 2). Our result upsamples the incident light field from 8 to 512 samples per pixel. The full uncompressed images are provided as auxiliary material.

sampled areas, and better support for the lower sampling density areas, as demonstrated in Figure 12.

Our reconstruction takes 3–4x longer than the algorithm of Lehtinen et al., primarily because they use a specialized 2D hierarchy in post-perspective space, essentially tailoring the data structure to the known distribution of reconstruction rays (they all originate from the lens). As our focus is reconstructing the incident light field at arbitrary points in the scene, we use a true 3D hierarchy.



Our result

No angular bandwidth

Figure 10: Disregarding angular effects, effectively treating all surfaces hit by secondary rays as diffuse, has a large negative impact on the result.



Figure 11: Motion blur and depth of field rendered using our algorithm using 1 sample per pixel as input. Please refer to the supplemental material for images showing input data and ground truth.

4 Discussion

Scalability and memory consumption The execution time of our algorithm scales linearly with the number of reconstruction rays, but is only very weakly affected by the structure of the scene and the number of input samples. For example, increasing the number of input samples from 1 to 8 in SAN MIGUEL increases the execution time by only about 40%. Also, SAN MIGUEL takes only 80% longer to reconstruct than CORNELL with the same parameters, even though the original model contains about 5–6 orders of magnitude more geometry, and has much more challenging structure (foliage vs. straight walls).

Memory consumption is almost completely determined by the number of input samples. Each sample consists of 30 floating-point numbers, totaling up to ~1GB at 720p with 8 input samples per pixel, independent of scene structure. In addition, the BVH consumes about 80MB with these parameters, again almost independently of the scene structure. All other memory usage is negligible.

Recursive filtering With very low input sampling rates, artifacts can appear near corners: when we integrate over the hemisphere, a large fraction of the reconstruction rays may hit the same few samples. If the samples are noisy, the reconstructed image can suffer from structured artifacts, as demonstrated in Figure 13. In this example, CORNELL was rendered with up to eight-bounce indirect illumination from a point light at 1024×1024 with only one sample per pixel. The multiple bounces lead to noise in the samples' radiances. We implemented an experimental step that reduces variance by performing a quick reconstruction of the *incident lighting at the secondary hits* and replacing the secondary hits' radiances

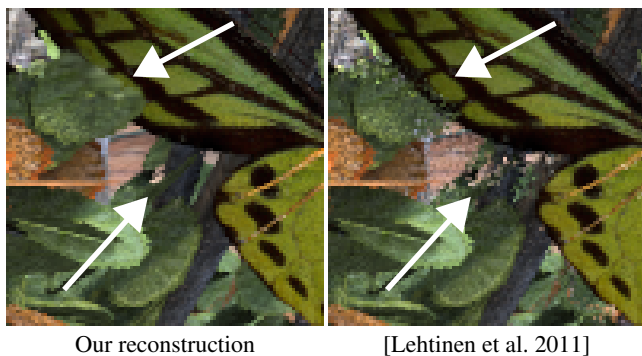


Figure 12: Close-up of a pinhole reconstruction from BUTTERFLIES. Lehtinen et al. [2011] do not adapt to the local density of the input samples and fail to reconstruct the marked regions. Our novel visibility heuristic reproduces the correct visibility.

with the reconstructed values before reconstructing incident lighting at the primary hits. In this scene, performing this step with 16 reconstruction rays adds less than 10% to execution time, and almost completely removes the artifacts. As a result, our reconstruction matches the ground truth closely, accurately preserving the boundaries of indirect shadows. Image antialiasing is still missing, because we have only one primary ray per pixel. Notice that the black borders near the corners are present in the ground truth PBRT rendering as well.

Glossy GI with noisy samples It should be possible to generalize the recursive filtering to more reliable handling of highly specular surfaces. In the extreme, it is clearly not very useful to reconstruct incident radiance onto a mirror from sparse samples. Instead it would make more sense to reconstruct at the next path segment to remove the noise on the surface seen in the mirror. Since our reconstruction works in 3D, it should be possible to extend it to walk forward in the path segments adaptively based on the angular bandwidth of the reflectance function.

Information content We find it interesting that even very sparse input samplings can contain nearly all of the relevant information about visibility and light transport, particularly in diffuse scenes. For example, in diffuse CORNELL one input path per pixel allows an essentially perfect reconstruction of indirect illumination, and even in the much more complex SAN MIGUEL four samples per pixel yields very good reconstructions.

Multi-layer materials Most realistic material models are built from multiple layers. It would be interesting to extend our work to reconstruct the layers independently. For example, a varnished wood consists of an almost-diffuse base layer, a specular top layer, and the two are mixed according to a Fresnel term. Treating the two layers separately would allow very broad reuse of the samples from the base layer, while the glossy top layer could only be used from a narrow set of directions. Overall this would allow broader reuse than collapsing the layers beforehand and selecting the maximum bandwidth for the sample.

Sheared filters In contrast to sheared reconstruction of (temporal) light fields [Egan et al. 2009; Egan et al. 2011b] where reconstruction and integration are combined to a single, large filter, we perform integration over incident angle numerically. This allows us to reason about the anisotropy of each input sample individually,

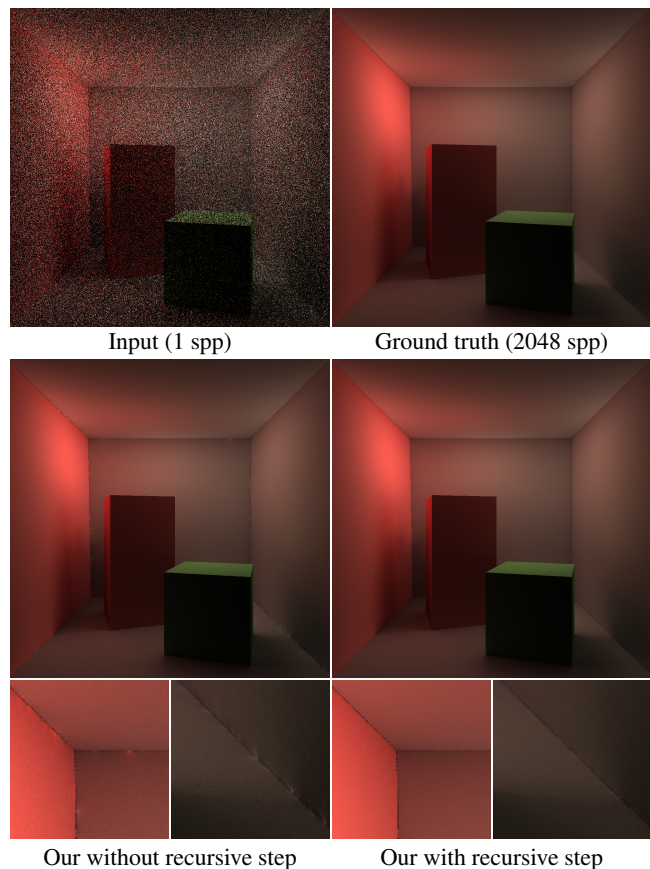


Figure 13: Multi-bounce indirect illumination in Cornell box reconstructed from 1 spp input. Recursive filtering removes the artifacts in corners.

but prevents a clean frequency analysis due to the non-translation-invariant nature of our reconstruction filters. However, in simple uniformly sampled planar scenarios, our filter reduces to earlier linear sheared formulations.

Failure cases The proposed algorithm can fail if the scene contains higher-frequency shading and visibility than what the input sampling rate can capture. For example, a sparse sampling of hair will probably not suffice to accurately describe its visibility and shading. In these cases the resulting reconstruction is consistent with the input sampling, but may not be faithful to the original scene. The foliage in our diffuse results is an example of difficult visibility that is handled gracefully at low sampling rates. Also, the distribution of input rays, which is driven by reflectance at the primary hits, may be insufficient to capture very high-frequency angular effects (e.g., a mirror) at the secondary hits. This would lead to slight blurring in the reconstructed light field.

Specialized hierarchies Our bandwidth-aware k -NN produces a hierarchy that serves two purposes: it is used to resolve visibility and also to guarantee that we find the nearby splats with similar angular support. An alternative design would be to have separate hierarchies for the two tasks. In that design, the visibility hierarchy would ignore angular bandwidth and implement the shrinking step. The second hierarchy would take bandwidth into account and omit shrinking. This slightly more complicated approach could improve the execution speed in case of very glossy materials.

Acknowledgements

We thank Guillermo Lleal Laguna for the San Miguel scene. Frédo Durand is supported by the National Science Foundation.

References

- ARVO, J., AND KIRK, D. 1987. Fast ray tracing by ray classification. *Computer Graphics (Proc. ACM SIGGRAPH '87)* 21, 55–64.
- BALA, K., DORSEY, J., AND TELLER, S. 1999. Radiance interpolants for accelerated bounded-error ray tracing. *ACM Trans. Graph.* 18, 213–256.
- BEKAERT, P., SBERT, M., AND HALTON, J. 2002. Accelerating path tracing by re-using paths. In *Proc. Eurographics Workshop on Rendering 2002*, 125–134.
- CHAI, J.-X., TONG, X., CHAN, S.-C., AND SHUM, H.-Y. 2000. Plenoptic sampling. In *Proc. ACM SIGGRAPH 2000*, 307–318.
- CHRISTENSEN, P., 2008. Point-based approximate color bleeding. Pixar Technical Memo #08–01.
- DAMMERTZ, H., SEWITZ, D., HANIKA, J., AND LENSCH, H. P. A. 2010. Edge-avoiding \hat{A} -trous wavelet transform for fast global illumination filtering. In *Proc. High Performance Graphics 2010*, 67–75.
- DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. X. 2005. A frequency analysis of light transport. *ACM Trans. Graph.* 24, 3, 1115–1126.
- EGAN, K., TSENG, Y., HOLZSCHUCH, N., DURAND, F., AND RAMAMOORTHY, R. 2009. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph.* 28, 3, 93:1–93:13.
- EGAN, K., DURAND, F., AND RAMAMOORTHY, R. 2011. Practical filtering for efficient ray-traced directional occlusion. *ACM Trans. Graph.* 30, 6, 180:1–180:10.
- EGAN, K., HECHT, F., DURAND, F., AND RAMAMOORTHY, R. 2011. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Trans. Graph.* 30, 2, 9:1–9:13.
- GASSENBAUER, V., KŘIVÁNEK, J., AND BOUATOUCH, K. 2009. Spatial directional radiance caching. *Computer Graphics Forum* 28, 4, 1189–1198.
- GLASSNER, A. 1988. Spacetime ray tracing for animation. *IEEE Computer Graphics and Applications* 8, 2, 60–70.
- HACHISUKA, T., JAROSZ, W., WEISTROFFER, R. P., DALE, K., HUMPHREYS, G., ZWICKER, M., AND JENSEN, H. W. 2008. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph.* 27, 3, 33:1–33:10.
- ISAKSEN, A., MCMILLAN, L., AND GORTLER, S. J. 2000. Dynamically reparameterized light fields. In *Proc. ACM SIGGRAPH 2000*, 297–306.
- KELLER, A. 1997. Instant radiosity. In *Proc. ACM SIGGRAPH '97*, 49–56.
- KONTKANEN, J., TABELLION, E., AND OVERBECK, R. 2011. Coherent out-of-core point-based global illumination. *Computer Graphics Forum* 30, 4, 1353–1360.
- LEHTINEN, J., AILA, T., CHEN, J., LAINE, S., AND DURAND, F. 2011. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph.* 30, 4, 55:1–55:12.
- MCCOOL, M. D. 1999. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Trans. Graph.* 18, 2, 171–194.
- OVERBECK, R., DONNER, C., AND RAMAMOORTHY, R. 2009. Adaptive wavelet rendering. *ACM Trans. Graph.* 28, 5, 140:1–140:12.
- PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering, 2nd ed.* Morgan Kaufmann.
- RAMAMOORTHY, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Trans. Graph.* 26, 1, 2:1–2:21.
- RITSCHEL, T., ENGELHARDT, T., GROSCH, T., SEIDEL, H.-P., KAUTZ, J., AND DACHSBACHER, C. 2009. Micro-rendering for scalable, parallel final gathering. *ACM Trans. Graph.* 28, 5, 132:1–132:8.
- ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2011. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.* 30, 6, 159:1–159:12.
- SCHAUFLEER, G., AND WANN JENSEN, H. 2000. Ray tracing point sampled geometry. In *Proc. Eurographics Workshop on Rendering 2000*, 319–328.
- SEN, P., AND DARABI, S. 2011. Implementation of random parameter filtering. Tech. Rep. EECE-TR-11-0004, University of New Mexico.
- SEN, P., AND DARABI, S. 2012. On filtering the noise from the random parameters in monte carlo rendering. *ACM Trans. Graph.*, To Appear.
- SOLER, C., SUBR, K., DURAND, F., HOLZSCHUCH, N., AND SILLION, F. 2009. Fourier depth of field. *ACM Trans. Graph.* 28, 2, 18:1–18:12.
- WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K., DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: a scalable approach to illumination. *ACM Trans. Graph.* 24, 3, 1098–1107.
- WANN JENSEN, H. 1996. Global illumination using photon maps. In *Proc. Eurographics Workshop on Rendering '96*, 21–30.
- WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In *Computer Graphics (Proc. ACM SIGGRAPH '88)*, 85–92.
- ZWICKER, M., PFISTER, H., VAN BAAR, J., AND GROSS, M. 2001. Surface splatting. In *Proc. ACM SIGGRAPH 2001*, 371–378.