



Photon Differential Splatting for Rendering Caustics

Frisvad, Jeppe Revall; Schjøth, Lars; Erleben, Kenny; Sporning, Jon

Published in:
Computer Graphics Forum

Link to article, DOI:
[10.1111/cgf.12347](https://doi.org/10.1111/cgf.12347)

Publication date:
2014

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Frisvad, J. R., Schjøth, L., Erleben, K., & Sporning, J. (2014). Photon Differential Splatting for Rendering Caustics. *Computer Graphics Forum*, 33(6), 252-263. <https://doi.org/10.1111/cgf.12347>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Photon Differential Splatting for Rendering Caustics

Jeppe Revall Frisvad¹, Lars Schjøth², Kenny Erleben³, and Jon Sporring³

¹Technical University of Denmark
jerf@dtu.dk

²Shape A/S, Denmark
ls@larum.dk

³University of Copenhagen, Denmark
{kenny,sporring}@diku.dk

Abstract

We present a photon splatting technique which reduces noise and blur in the rendering of caustics. Blurring of illumination edges is an inherent problem in photon splatting, as each photon is unaware of its neighbors when being splatted. This means that the splat size is usually based on heuristics rather than knowledge of the local flux density. We use photon differentials to determine the size and shape of the splats such that we achieve adaptive anisotropic flux density estimation in photon splatting. As compared to previous work that uses photon differentials, we present the first method where no photons or beams or differentials need to be stored in a map. We also present improvements in the theory of photon differentials, which give more accurate results and a faster implementation. Our technique has good potential for GPU acceleration, and we limit the number of parameters requiring user adjustment to an overall smoothing parameter and the number of photons to be traced.

Keywords: density estimation, ray differentials, particle tracing, photon mapping, photon splatting

ACM CCS: Computer Graphics [Computing Methodologies]: Rendering—

1. Introduction

Caustic illumination is common in both man-made and natural environments. It is light that goes from a light source through one or more specular reflections or refractions before reaching a diffuse surface which is observed by the eye (LS+DS+E, in light transport notation [Hec90]). Thus it is, for example, light coming through a window or sunlight focused by the water ripples at a shallow beach. A particular challenge in rendering caustic illumination is that it often has both very soft and very sharp (focused) features. If we shine light at a gold ring, the light reflected from the front of the ring is soft, whereas the light enveloped by reflection on the inside has a sharp cardioid border (see Figure 1).

In path tracing [Kaj86], caustics are prone to high-frequency noise as they often consist of high-intensity light taking a low-probability path. It is therefore common to use a biased technique such as photon mapping [Jen01] to render

caustics. Photon mapping relies on flux density estimation to reconstruct the illumination in a scene from a sparse sampling of light paths. The reconstruction introduces a trade-off between low-frequency noise and blurring effects; variance versus bias. With this trade-off, it is very difficult to get both sharp and soft illumination features at the same time, unless we trace a very large number of photons.

Photon differentials [SFES07] were introduced to improve the trade-off in the density estimation such that we diminish both noise and blur with the same number of photons. However, emitting, tracing, and storing differentials alongside the photons, as well as the density estimation that uses the photon differentials, all add computational costs to the standard photon mapping algorithm. To improve the rendering quality that we can obtain from using photon differentials, and to lower the additional costs, we present

- more accurate emission of photon differentials,
- a splatting method where elliptic splats adapt to the structure in the illumination without a need to store photons or beams or differentials in a map, and
- faster anisotropic flux density estimation.

The definitive version is available at wileyonlinelibrary.com

DOI: <http://dx.doi.org/10.1111/cgf.12347>

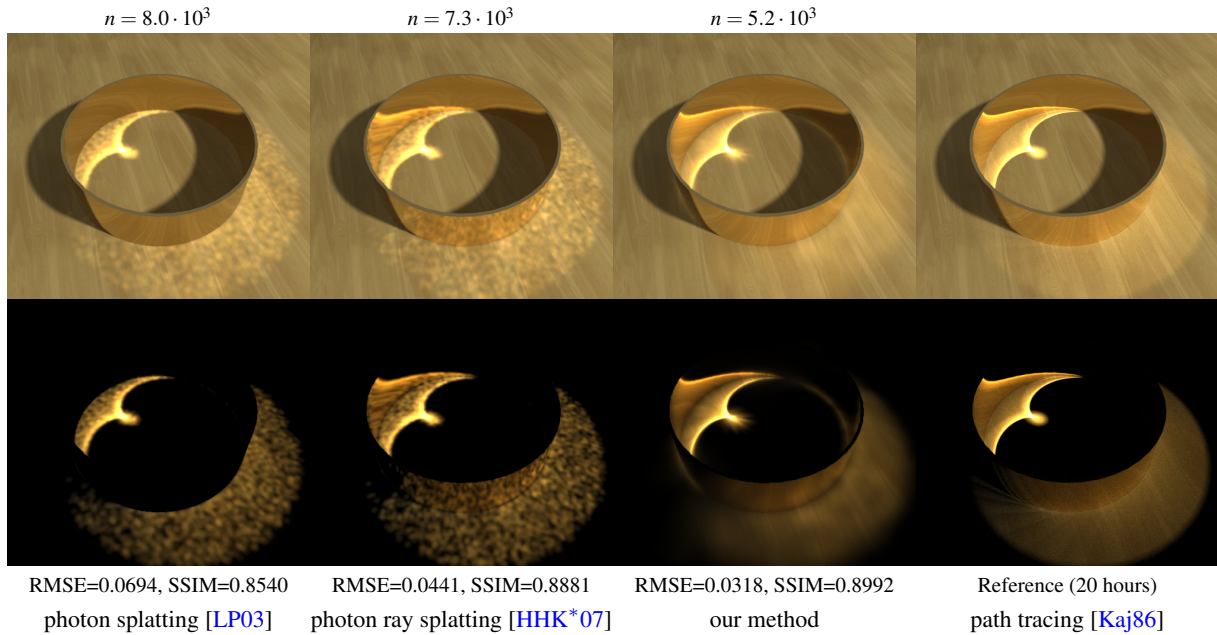


Figure 1: The classic gold ring that generates a cardioid caustic. Final renderings are in the top row. Renderings with caustic illumination only are in the bottom row. In the first three columns, the images were rendered in equal time with only 3.3 seconds for the caustics, and n is the number of caustic photons processed within this time budget. We report root-mean-squared error (RMSE) and structural similarity indices (SSIM) as compared to the path-traced reference (rightmost column). RMSE measurements refer to both rows, SSIM measurements refer to the bottom row only. The photon splatting implementation [LP03] uses fixed bandwidth and GPU rasterization for the splatting. The renderings illustrate that the density estimation in existing splatting techniques is more suitable for diffuse interreflections than for caustics.

2. Related Work

Methods for rendering caustics have mostly developed in two directions: (1) toward faster methods that come at the cost of excluding some light paths or not accounting for indirect shadows [RDGK12, Sec. 4.4]; (2) toward more time-consuming but consistent algorithms that progressively add in results from more photons [HOJ08, HJ09, KZ11, JNT*11, SJ13, KD13]. The method we present falls in-between these two categories. We include the same light paths as in the caustics part of photon mapping, while we seek to improve render quality using the same number of photons. There is a small family of existing methods with this profile [Mys97, Sch03, HHK*07, SFES07, SOS08, SJ09]. With the exception of photon ray splatting [HHK*07], all these methods require a data structure for storing photons. We trade the photon map for a map of eye path vertices. The map of eye path vertices was also used in photon ray splatting [HHK*07] and in progressive photon mapping [HOJ08], but both these methods retain a photon (ray) map for algorithmic purposes. We achieve some advantages from not using a photon map. As in the progressive techniques, there is no limit to the number of photons that we can trace. In addition, we have lower memory requirements and we save the time it takes to build the map.

Photon splatting [SB97, LP03] was introduced as a technique to speed up density estimation using rasterization. This technique is problematic if caustics are observed via specular surfaces (light paths LS+DS+E). The problem is that reflections and refractions see radiance from a different position in the scene than the position of the specular object itself. Thus, as the rasterized splats only contribute to the pixels they cover, reflected/refracted caustics remain absent (this problem appears in the leftmost column of Figure 1 and in similar renderings in references on photon splatting [LP03, ML09, YWC*10]). The splatting method of Herzog et al. [HHK*07] solves this problem. As they use a map of eye path vertices, they splat to both directly visible positions in a scene and positions seen via one or more interactions with specular surfaces. Since we also use this eye path map, we include all light paths in caustic illumination.

While photon splatting is faster than standard photon mapping, the main problem is to find an appropriate splat size (bandwidth). Various heuristics have been employed in order to adapt the splat size to the illumination so that sharp features are not blurred out [LP03, HHK*07, WD08, Wym08]. These heuristics have been applied with some success, but they rarely achieve a better bias-variance trade-off than what we get with standard photon mapping. We use

photon differentials to adapt both the size and the shape of the splats to the structure of the light after it has interacted with specular surfaces. This means that we trace photon beams that change shape and size as they travel through a scene according to the concept of ray differentials [Ige99]. The result is a method for rendering caustics that offers improved density estimation while it also retains the speed of the splatting approach.

Figure 1 exemplifies the density estimation in two of the existing photon splatting methods [LP03, HHK*07]. In comparison to our anisotropic density estimation (Figure 1, third column), these splatting methods require a significantly larger number of caustic photons to render caustics of decent quality. The advantage of photon splatting [LP03] is rasterization based density estimation which enables real-time fly-through visualizations. However, when rendering a single image, the performance improvement is small. When the eye path map is introduced [HHK*07], we obtain reflected and refracted caustics at the cost of view-dependency. To estimate the single image performance differences, compare the reported number of caustic photons n processed in equal time with the different methods.

Photon differentials were combined with path probability density by Fabianowski and Dingliana [FD09]. This makes photon differentials useful for full global illumination instead of caustic illumination only. In comparison to our method, Fabianowski and Dingliana [FD09] only work with point lights and they do not take a splatting approach. Instead of splatting, they replace the traditional kd tree with a bounding volume hierarchy, and they achieve interactive frame rates for two light bounces by GPU acceleration. The idea of using path probability density to control the length of the differential vectors was introduced by Suykens and Willems [SW01]. Their technique is called path differentials, and it was the first decoupling of ray differentials from the image space uv -coordinates. This decoupling is necessary to emit and trace photon differentials accurately from arbitrary light sources instead of a point (see Section 3).

Photon differentials have also been used for volumetric photon mapping [Sch09, JNSJ11, JNT*11]. In this setup, Jarosz et al. [JNSJ11] describe emission of photon differentials from arbitrary light sources. However, they overlook that the initial photon position and direction are not sampled using the same local uv -coordinates. Decoupling is necessary in the same way as for path differentials. Jarosz et al. [JNT*11] describe how to employ photon differentials in progressive photon mapping. This means that they can progressively shrink the photon footprints. However, it also means that they must retain the photon (beam) map. They also describe a number of implementation speed-ups such as splatting of directly visible photon beams using rasterization. These speed-ups can also be used to accelerate our method.

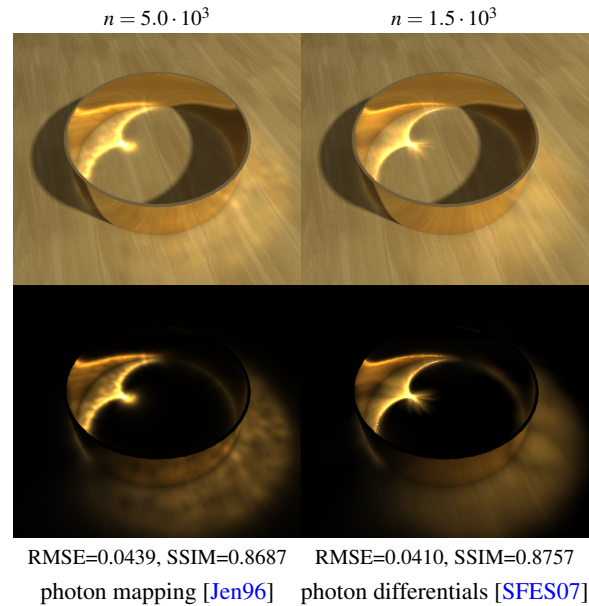


Figure 2: Gold rings rendered in equal time with 3.3 seconds for the caustics. We report RMSE and SSIM as in Figure 1 (same reference). As revealed by the n values in this figure and in Figure 1, the processing overhead of the original photon differentials technique [SFES07] is significantly reduced in the technique presented here.

2.1. Bandwidth Selection and Kernel Anisotropy

As mentioned above, it is a challenge to select bandwidth. In a photon splatting context, the splat size is the bandwidth. In standard photon mapping [Jen96], the distance to the k th nearest neighbor (k NN) in the photon map is the bandwidth unless we also range-restrict our nearest neighbor look-ups. This k NN adaptive bandwidth selection improves the bias-variance trade-off (compare the leftmost columns of Figures 1 and 2). When the photon map is available, there are many ways to further improve the bias-variance trade-off. This is usually done by locally investigating differences in estimated radiance based on the nearest neighbors in the photon map. Even the first presentation of photon mapping [JC95] includes a bias-reducing method called differential checking. In this method, k is adaptive so that a smaller number of neighbors is used if a large difference is detected in the radiance estimate for smaller k . Similar work exists [Mys97, Sch03] where the bandwidth selection based on the nearest neighbors is more advanced. Recently, it has been shown that an asymptotically optimal bandwidth can be computed in progressive photon mapping by estimating the Laplacian of the radiance in the photon map [KD13].

Another way to improve the bias-variance trade-off is to adaptively choose an anisotropic kernel shape (not bandwidth) using the gradient of the radiance in the photon

map. This approach is called diffusion-based photon mapping [SOS08]. The radiance gradient is estimated by a look-up into the photon map for every photon. This is quite expensive and it introduces two more parameters to tweak (maximum search radius and maximum number of photons in the gradient estimate) in addition to a diffusivity coefficient which is used to control the anisotropy in this method.

Since we choose to abandon the photon map, we cannot use an estimate of radiance or of the radiance gradient or Laplacian to choose splat size and shape. Our splatting method is thus incompatible with these techniques for improved density estimation. As we shall see in the following section, the key insight, which enables us to efficiently select bandwidth and kernel anisotropy without radiance estimation, is the relation between radiance and scene geometry. Light is emitted from an area in a solid angle and radiance is flux per projected area per solid angle. This means that changes in local radiance to some extent follow changes in first derivatives of light ray positions taken with respect to local geometric coordinates.

3. Theory

As in standard photon mapping [Jen96], we emit photons from the light sources and trace them through the scene using a path tracing approach. In photon mapping, photons that reach a non-specular surface are stored in a spatial data structure. Subsequently, the stored photons are used for illumination reconstruction by kernel density estimation. In our splatting approach, the photons need not be stored. Instead, they are splatted so that they contribute directly to all pixels that observe the surface area covered by the splat (via light paths DS*E).

To describe our contributions, we must reconsider the theory of photon differentials. In previous work, an emitted photon has been treated as if it were fully described by one set of local uv -coordinates. This is not true in general for arbitrary light sources. In the following, we show that it is possible to handle photon differentials from arbitrary light sources in the same way as ray differentials. However, for this approach to be accurate, the initial differential vectors must have specific directions. We also describe the splatting of photon differentials without mapping and provide an efficient method for splatting photons with elliptic footprints.

3.1. Emitting Photon Differentials

When photons are emitted from an arbitrary light source, the sampling of the photon origin $\mathbf{x}(u, v)$ and the sampling of the photon direction $\vec{\omega}(\theta, \phi)$ can be entirely unrelated. Let us model a photon ray by the parametrization of a straight line $\mathbf{r}(t) = \mathbf{x} + t\vec{\omega}$ with $t \in [0, \infty)$. If we let t' denote the distance to the first point along a ray where it intersects the scene geometry, we have that

$$\mathbf{r}(t') \mapsto \mathbf{r}(u, v; \theta, \phi) = \mathbf{x}(u, v) + t'(u, v; \theta, \phi) \vec{\omega}(\theta, \phi) , \quad (1)$$

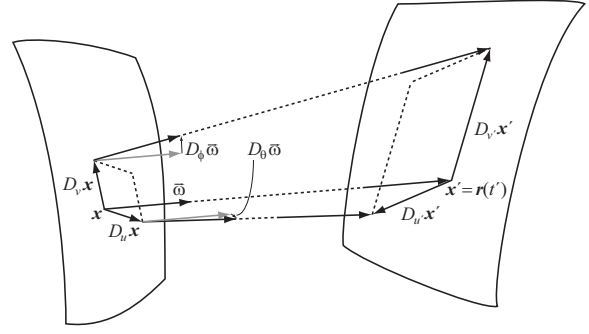


Figure 3: Illustration of photon differentials and the meaning of the positional and directional differential vectors. This illustration has appeared before [Fri12b], but the conditions necessary for $D_\theta\vec{\omega}$ to only influence $D_u\mathbf{x}$ and for $D_\phi\vec{\omega}$ to only influence $D_v\mathbf{x}$ have not previously been published.

where u and v parameterize the light source surface and θ and ϕ parameterize the emission solid angle (see Figure 3). Suykens and Willems [SW01] describe how to combine differentials taken with respect to different local coordinates. This is done by estimating the Minkowski sum (\oplus) of all the differential vectors. Thus, the emitted photon beam is in principle a zonohedron with an octagon footprint.

The positional differential vectors $D_u\mathbf{x}$ and $D_v\mathbf{x}$ start out as an orthogonal uv -basis of the surface tangent plane at \mathbf{x} ; the directional differential vectors $D_\theta\vec{\omega}$ and $D_\phi\vec{\omega}$ start out as an orthogonal $\theta\phi$ -basis of the plane perpendicular to $\vec{\omega}$. We find these directions using the surface normal \vec{n} at \mathbf{x} and the direction of emission $\vec{\omega}$. Since we can choose the uv - and the $\theta\phi$ -bases arbitrarily in their respective planes, we choose them such that the u - and θ -directions are identical and the v - and ϕ -directions become identical *after* transfer to the first intersection point. This is done using the intersection of the uv -plane with the $\theta\phi$ -plane:

$$\frac{D_u\mathbf{x}}{|D_u\mathbf{x}|} = \frac{D_\theta\vec{\omega}}{|D_\theta\vec{\omega}|} = \frac{\vec{n} \times \vec{\omega}}{|\vec{n} \times \vec{\omega}|} , \quad (2)$$

$$\frac{D_v\mathbf{x}}{|D_v\mathbf{x}|} = \frac{D_u\mathbf{x}}{|D_u\mathbf{x}|} \times \vec{n} , \quad \frac{D_\phi\vec{\omega}}{|D_\phi\vec{\omega}|} = \frac{D_\theta\vec{\omega}}{|D_\theta\vec{\omega}|} \times \vec{\omega} , \quad (3)$$

which works as long as the planes are not parallel. In the special case where the direction of emission is (almost) in the normal direction ($\vec{\omega} \approx \vec{n}$), or if the source has no normal, we use a method for building an orthonormal basis from a three-dimensional vector [Fri12a, for example]. In this way, the positional and directional vectors are always pairwise parallel after projection onto the tangent plane of the receiving surface at $\mathbf{x}' = \mathbf{r}(t')$, see Appendix A. The Minkowski sum now gives a parallelogram footprint.

As we would like to splat an elliptic kernel for every photon, we refer to the maximum-area ellipse inscribed in the parallelogram as the photon footprint, see Figure 4. We place

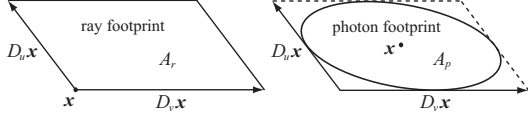


Figure 4: The difference between ray and photon footprints (appeared before [Fri12b] but included for completeness).

this ellipse such that its center is the photon position \mathbf{x} . The ellipse's semi-axes are then the column vectors in $\frac{1}{2}D\mathbf{x}$ and the area of the photon footprint is

$$A_p = \frac{\pi}{4}A_r = \frac{\pi}{4}|D_u\mathbf{x} \times D_v\mathbf{x}|, \quad (4)$$

where A_r is the area of the corresponding ray footprint. By analogy, the photon solid angle is

$$\omega_p = \frac{\pi}{4}|D_\theta\vec{\omega} \times D_\phi\vec{\omega}|.$$

For completeness, we describe how to set sensible initial lengths for the emitted differential vectors. A light source emits photons from points across an area A_e and in directions within a solid angle ω_e . Let us set the sum of the initial photon footprint areas as $s^2 A_e$, where s is a smoothing parameter discussed later. Since the positional differential vectors are initially orthogonal, their lengths would then be

$$|D_u\mathbf{x}| = |D_v\mathbf{x}| = 2s\sqrt{\frac{A_e}{\pi N_e}}, \quad (5)$$

where N_e is the number of photons emitted from the source. As a consequence, the initial positional differential vectors of a point light are zero vectors. The zero vectors are not really a basis, but the directional differential vectors will turn them into a basis at any distance from the point. This is similar to the fact that a point light cannot emit radiance, since it has no area. So it has intensity and we can measure the radiance due to the point light at any distance from the source.

The elliptic area spanned by the directional differential vectors corresponds to a solid angle. It is the photon footprint area that a photon would attain if emitted from a point source to the surrounding unit sphere, just as a solid angle is measured by the area on the unit sphere which the solid angle intercepts. Thus, we can set the sum of initial photon solid angles to $s^2\omega_e$. Since the directional differential vectors are initially orthogonal, their lengths would then be

$$|D_\theta\vec{\omega}| = |D_\phi\vec{\omega}| = 2s\sqrt{\frac{\omega_e}{\pi N_e}}. \quad (6)$$

In analogy with the point source, we here have the special case of collimated/directional light where the directional differential vectors are initially zero vectors.

3.2. Tracing Photon Differentials

Once a photon has been emitted with its associated differential, it is traced to the nearest surface intersection point

$\mathbf{x}' = \mathbf{r}(t')$, and the photon differential is transferred to this point by computing $D\mathbf{x}'$. Since we ensure that our differential vectors are pairwise parallel, we have

$$D\mathbf{x}' = [D_u'\mathbf{x}' \ D_v'\mathbf{x}'] = [(D_u + D_\theta)\mathbf{x}' \ (D_v + D_\phi)\mathbf{x}']. \quad (7)$$

To find the transferred positional differential vectors, we take the partial derivatives of the ray parametrization (1).

We let \vec{n}' denote the surface normal at \mathbf{x}' . In the first-order approximation, any offset of the intersection point must stay in the tangent plane. Thus, if (\vec{n}', d) are the coefficients that define the tangent plane, we have

$$\vec{n}' \cdot \mathbf{x}' + d = \vec{n}' \cdot (\mathbf{x} + t'\vec{\omega}) + d = 0 \Rightarrow t' = -\frac{\vec{n}' \cdot \mathbf{x} + d}{\vec{n}' \cdot \vec{\omega}}.$$

With this expression for t' in terms of the parameters $\mathbf{x}(u, v)$ and $\vec{\omega}(\theta, \phi)$, the transferred differential vectors become

$$D_u\mathbf{x}' = D_u\mathbf{x} + D_u t'\vec{\omega} = D_u\mathbf{x} - \frac{\vec{n}' \cdot D_u\mathbf{x}}{\vec{n}' \cdot \vec{\omega}}\vec{\omega} \quad (8)$$

$$D_\theta\mathbf{x}' = t'D_\theta\vec{\omega} + D_\theta t'\vec{\omega} = t' \left(D_\theta\vec{\omega} - \frac{\vec{n}' \cdot D_\theta\vec{\omega}}{\vec{n}' \cdot \vec{\omega}}\vec{\omega} \right), \quad (9)$$

where $D_v\mathbf{x}'$ is found by substituting the subscript u with v , and $D_\phi\mathbf{x}'$ is found by substituting θ with ϕ . The operator sums $D_u + D_\theta$ and $D_v + D_\phi$ result in precisely the same formula for transfer of photon differentials as the one presented by Igehy [Ige99] for transfer of ray differentials. We emphasize that this relation (7) is only true as long as we choose the directions of our initial differential vectors as in Equations 2–3. Otherwise, we would need the method of Suykens and Willems [SW01] to construct a pair of differential vectors that approximate the octagonal footprint.

Since we are only working with caustic illumination, every photon-surface interaction will be either reflection or refraction. The path is terminated once a non-specular surface is reached. Thus, photon differentials (for caustic illumination) can be traced in the same way as the ray differentials described by Igehy [Ige99]. This means that the positional differential vectors change after each transfer to a new surface and that the directional differential vectors change after each reflection/refraction.

3.3. Splatting Photon Differentials

A photon carries radiant flux Φ_p , but, since we also trace its differential, we can obtain the irradiance that it contributes

$$E_p = \Phi_p/A_p,$$

where A_p is the photon footprint area (4) after the positional differential vectors have been modified by transfers along the photon path.

Using a normalized kernel, we get the reflected radiance at a surface position \mathbf{x} in the direction $\vec{\omega}$ by [SFES07]

$$L_r(\mathbf{x}, \vec{\omega}) \approx \sum_{p=1}^k \pi K(|M_p(\mathbf{x} - \mathbf{x}_p)|) f_r(\mathbf{x}, -\vec{\omega}_p, \vec{\omega}) E_p, \quad (10)$$

where k is the number of photons in the estimate, f_r is the bidirectional reflectance distribution function (BRDF), and \mathbf{M}_p is a matrix that performs a change of basis to a filter space, where the photon footprint is a unit circle. In filter space, we can use any of the standard kernels that apply to a unit circle. Different options are available from Silverman [Sil86]. We prefer Silverman's second-order kernel

$$K(x) = \begin{cases} \frac{3}{\pi}(1-x^2)^2 & \text{for } x < 1 \\ 0 & \text{otherwise} \end{cases}, \quad (11)$$

as it has compact support and continuous derivative.

If we adapt the kernel as in standard photon mapping, \mathbf{M}_p is simply $\frac{1}{h}\mathbf{I}$ for all p . Here h is the distance to the k th nearest neighbor (the bandwidth) and \mathbf{I} is the 3×3 identity matrix. If we instead use the photon footprint for photon p , we get a 2×3 transformation matrix from the positional differential vectors as follows:

$$\mathbf{M}_p = \frac{2}{D_u \mathbf{x}_p \cdot (D_v \mathbf{x}_p \times \vec{n}_p)} \begin{bmatrix} D_v \mathbf{x}_p \times \vec{n}_p \\ \vec{n}_p \times D_u \mathbf{x}_p \end{bmatrix}. \quad (12)$$

This equation for \mathbf{M}_p is an efficient way of computing the top two rows of the inverse of a change-of-basis matrix with the footprint semi-axes and \vec{n}_p as columns. We have a singularity if the footprint has collapsed (zero area). This seems to be a rare event, so we discard the photon. It would be more accurate to store such photons and deal with them in a postprocess using a method such as bidirectional path tracing [Vea97].

In a photon mapping approach augmented by photon differentials, we would use Equations 10–12 directly. Splatting is a different way of evaluating the same equations. Instead of looping over all eye path vertices and gathering the contribution of the neighboring photons, we loop over all the photons and distribute their contributions to the neighboring eye path vertices. The end result is the same.

A photon is splatted if it reaches a non-specular surface after interacting with one or more specular surfaces. Splatting is done by a look-up into an eye path map (a kd tree of eye path vertices). The eye path map is constructed using a path tracing approach before photon emission starts. For each eye path vertex, we store hit position \mathbf{x} , ray direction $\vec{\omega}$, BRDF index, importance (color weight), and pixel index. With this information, we can progressively add the reflected radiance due to a single photon (a term in the sum in Equation 10) directly to the pixels that the photon footprint covers.

For each splat, we need to find all eye path vertices covered by the elliptic photon footprint. Using a range-restricted nearest-neighbor search, the maximum distance to look for eye path vertices is the major radius of the photon footprint:

$$r_{\max} = \frac{1}{2} \max(|D_u \mathbf{x}_p|, |D_v \mathbf{x}_p|).$$

Eye path vertices outside the ellipse are discarded by a simple check in filter space (discard if $|\mathbf{M}_p(\mathbf{x} - \mathbf{x}_p)| \geq 1$). This is much more efficient than the mapping approach [SFES07],

where the longest major radius of *all* the footprints in the entire photon map should be used as range restriction.

The idea to use only a 2×3 matrix for \mathbf{M}_p and the efficient formula (12) for getting this matrix is new compared to previous work. This approach gives a good speed-up (see Section 4.2) as we would otherwise need to take the inverse of a 3×3 matrix for every splatted photon. Using only the top two rows, we assume, as in standard photon mapping, that the surface is locally flat. In some cases, in the vicinity of sharp corners, for example, the assumption that the surface is locally flat is objectionable and results in topological bias. We can reduce this type of bias by checking the distance to the photon intersection point in the normal direction $|\vec{n}_p \cdot (\mathbf{x} - \mathbf{x}_p)|$. If this distance is above some threshold, we discard the contribution from the photon. Alternatively, we could insert $a\vec{n}_p$ as the third row in \mathbf{M}_p , where a is a threshold, and we would have an ellipsoidal anisotropic kernel that reduces topological bias of this kind.

The overall size of the photon footprints corresponds to the bandwidth in the radiance estimate (10). Larger footprints reduce noise but promote bias, whereas smaller footprints have the opposite effect. We control the overall footprint size, and thus the trade-off between variance and bias, using a smoothing parameter s (see Equations 5 and 6). Since the kernels are normalized, the energy in the scene does not change when the footprint size is changed (with the exception that larger footprints could lead to an increasing loss of energy due to boundary bias). Like the number of nearest neighbors k in standard photon mapping, the smoothing parameter s is determined empirically. Values in the range from $s = 5$ to $s = 40$ worked well in most of our test cases.

4. Results

In addition to our new technique for rendering caustics, we have implemented several existing techniques. Our purpose is to find the technique which achieves the best rendering quality in equal time. We provide comparisons to the extent that we find it necessary to reach conclusions toward this end. We can quickly establish that the other splatting techniques [LP03, HHK*07] cannot provide similar quality in equal time (Figure 1). With photon differential mapping [SFES07], we can obtain quality similar to what we get with the splatting technique presented here, but it is always slower (Figures 1, 2, and 10, consult the number of photons n processed in equal time).

Diffusion-based photon mapping [SOS08] is inferior to photon differential mapping [Sch09]. We validate in Section 4.1 that it is also inferior to our new splatting technique. This leaves standard photon mapping [Jen96] as the more serious competitor, so standard photon mapping is included in all comparisons. Finally, we provide a rendering which is similar to what you can get with progressive photon mapping (Figure 13). The image can be compared visually to a similar

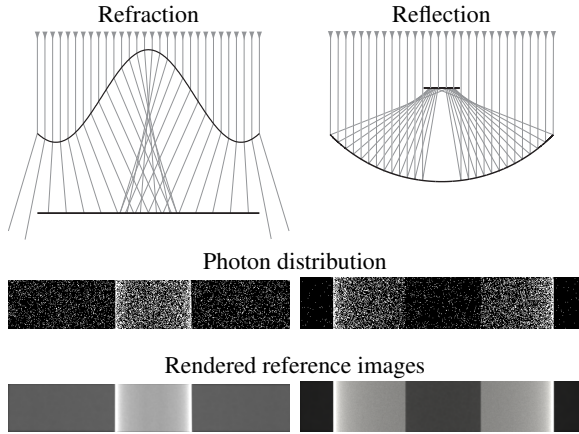


Figure 5: Two case studies: a sinusoidally shaped water wave illuminated from above by collimated light (left column) and a clipped metal ring illuminated by collimated light (right column).

rendering in the original paper on progressive photon mapping [HOJ08, Figure 7]. We get a similar result using two orders of magnitude fewer photons.

4.1. Simplistic Scenes

To validate our approach (photon differential splatting), we reproduce the simplistic case studies of Schjøth [Sch09] and compare them with standard photon mapping [Jen96] and diffusion-based photon mapping [SOS08]. The case studies are two simplistic scenes that produce caustics by reflection and refraction, respectively. In the case study scenes, which are illustrated in Figure 5, the camera has been placed so that it solely captures the caustic. The visualization of the photon distribution is point rendering of 10,000 caustic photons, whereas the reference images were rendered using standard photon mapping with 10 million photons in the map.

We estimate the quality of the renderings using two different objective image quality metrics, namely root-mean-squared error (RMSE) and the structural similarity index (SSIM, [WBSS04]). The former is a widely used, generic mathematical metric, the latter is based on a model of the human visual system. SSIM measures the similarity between two images with respect to contrast, luminance, and structure. An index of 1 means that the two images are identical, while an index of 0 means that the images have no similarity.

The best settings for a rendering algorithm are different for different image quality metrics. Using 20,000 caustic photons for the refraction case, and measuring quality as compared to the reference image, we systematically tune the rendering parameters and find different optimal bandwidths for each of the three rendering algorithms that we are comparing. This is illustrated in Figure 6. Renderings of this

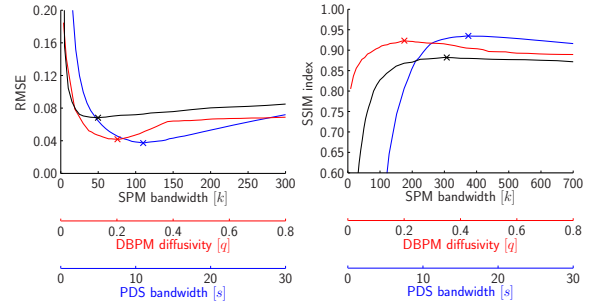


Figure 6: Curves plotting bandwidth against RMSE and SSIM. The measured images are renderings of the refraction case using 20,000 caustic photons. The black curves and black horizontal axes are for standard photon mapping (SPM), the red ones are for diffusion-based photon mapping (DBPM), and the blue ones are for photon differential splatting (PDS). Note that the relative placement of the curves along the horizontal axes is not important as each algorithm uses a different quantity on this axis.

Method	RMSE-optimal bandwidth	SSIM-optimal bandwidth
SPM	(a) RMSE = 0.0682	(b) SSIM = 0.8821
DBPM	(c) RMSE = 0.0418	(d) SSIM = 0.9234
PDS	(e) RMSE = 0.0370	(f) SSIM = 0.9348

Figure 7: The refraction case study using 20,000 caustic photons. Images were rendered at RMSE-optimal bandwidths (left column) and SSIM-optimal bandwidths (right column) using standard photon mapping (SPM, a–b), diffusion-based photon mapping (DBPM, c–d), and photon differential splatting (PDS, e–f).

case using the optimal bandwidths (the × marks in Figure 6) are in Figure 7. The RMSE-optimal images (a,c,e) contain clearly visible noise, indicating that RMSE favors noise over bias to a higher degree than SSIM.

According to the objective quality metrics, photon differential splatting clearly provides better render quality using the same number of photons. This is true in the refraction case study scene and in all other scenes we have tested. The next step is to find out how many caustic photons we need to get comparable rendering quality using the other methods. The images in Figure 8 were found by increasing the number of photons until the image quality measure for the optimal bandwidth was approximately the same as that of the photon differential splatting in Figure 7(e–f). From this comparison, we see that standard photon mapping requires one order of magnitude more caustic photons to obtain comparable RMSE and almost two orders of magnitude to get

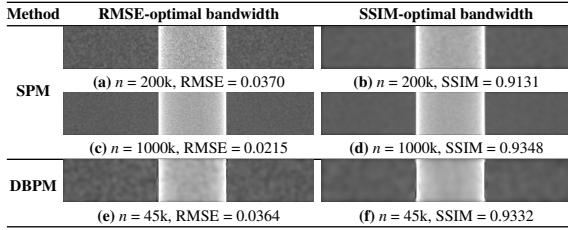


Figure 8: Tests to see how many photons we need to get quality similar to what we obtain with $n = 20k$ caustic photons using PDS (Figures 7e–f, k for kilo). Images were rendered at RMSE-optimal bandwidths (left column) and SSIM-optimal bandwidths (right column) using standard photon mapping (SPM, a–d) and diffusion-based photon mapping (DBPM, e–f).

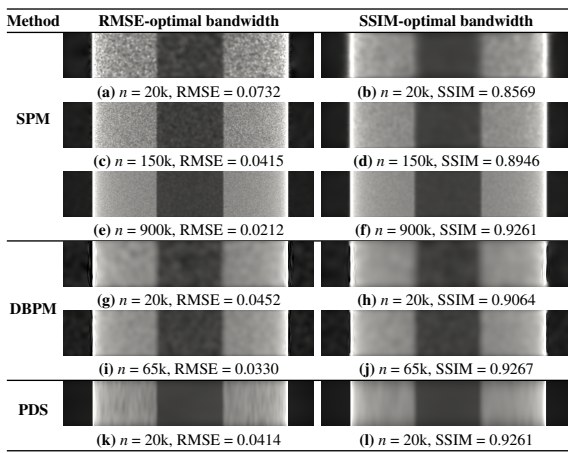


Figure 9: The reflection case study, where n is the number of caustic photons (k for kilo). Images were rendered at RMSE-optimal bandwidths (left column) and SSIM-optimal bandwidths (right column) using standard photon mapping (SPM), diffusion-based photon mapping (DBPM), and photon differential splatting (PDS).

SSIM comparable to that of photon differential splatting. Diffusion-based photon mapping requires only around twice as many caustic photons. The reflection case study is investigated in the same manner as the refraction case study. Here we see a similar trend (Figure 9).

The number of caustic photons that a method needs to reach a specific quality is one thing. For methods based on photon mapping, this is important with respect to memory requirements. The render efficiency of a method, however, does not necessarily go hand in hand with the number of caustic photons that it needs. Efficiency is rather a matter of quality obtainable in equal time. Table 1 is an overview of render times for the images in Figures 7–9. Red numbers indicate that the bandwidth is RMSE-optimal, blue numbers

Table 1: Render times in seconds for the renderings in Figures 7–9 using an Intel Core2 Duo 2.4 GHz laptop.

Figure	Method	Caustic photons	Bandwidth	RMSE	SSIM	Render time (s)
7a	SPM	20k	$k = 50$	0.0682	0.7184	1.16
7b			$k = 300$	0.0860	0.8821	3.24
8a		200k	$k = 100$	0.0370	0.6900	5.37
8b			$k = 800$	0.0624	0.9131	12.41
8c	DBPM	1000k	$k = 240$	0.0250	0.8007	23.28
8d			$k = 1800$	0.0479	0.9348	34.81
7c		20k	$h = 0.007$	0.0418	0.8976	3.67
7d			$h = 0.011$	0.0521	0.9234	6.50
8e	45k	$h = 0.005$	0.0364	0.8930	5.38	
8f		$h = 0.008$	0.0418	0.9332	10.34	
7e	PDS	20k	$s = 11.0$	0.0370	0.9121	1.91
7f			$s = 16.0$	0.0449	0.9348	3.07
9a	SPM	20k	$k = 30$	0.0732	0.6260	1.84
9b			$k = 260$	0.1057	0.8569	2.78
9c	150k		$k = 65$	0.0415	0.6692	8.96
9d			$k = 550$	0.0732	0.8946	11.03
9e	900k		$k = 240$	0.0212	0.8211	50.41
9f			$k = 1200$	0.0475	0.9261	56.31
9g	DBPM	20k	$h = 0.0014$	0.0452	0.8797	3.78
9h			$h = 0.0020$	0.0573	0.9064	5.44
9i		65k	$h = 0.0011$	0.0330	0.9134	13.08
9j			$h = 0.0014$	0.0400	0.9267	16.52
9k	PDS	20k	$s = 13.5$	0.0414	0.9154	4.95
9l			$s = 23.0$	0.0560	0.9261	10.08

indicate that it is SSIM-optimal. Within each case study, bold-font numbers of the same color indicate that the image quality is nearly the same. The colored bold-font render times reveal that we consistently get the same quality faster using photon differential splatting (except perhaps for the RMSE-optimal rendering in the reflection case, where diffusion-based photon mapping seems to be competitive).

4.2. Common Scenes

We present equal-time renderings to illustrate that our method is more efficient and provides improved quality as compared to standard photon mapping [Jen96] and photon differential mapping [SFES07]. To get reasonable results with the latter method, we set a maximum number of photons to search for in the otherwise range-restricted kd tree look-up. If this is not done, rendering times become at least twice as long, and results for this method [SFES07] would then be inferior to standard photon mapping in most equal-time comparisons.

To have a fair comparison, all our tests use the CPU only (except the photon splatting [LP03] in Figure 1, where splatting is done using the GPU rasterization pipeline). However, the GPU speed-ups described by Jarosz et al. [JNT*11] can be applied to our method to make it even faster. This gives us an advantage compared with diffusion-based photon mapping [SOS08] and photon relaxation [SJ09], where speed-ups based on the rasterization pipeline do not apply.

Figures 1 and 2 contain renderings of the classic gold ring that generates a cardioid caustic. Reflection from the gold material is computed using the Fresnel equations with the complex refractive index of gold [Gla95]. In this scene, standard photon mapping tends to blur the sharp features,

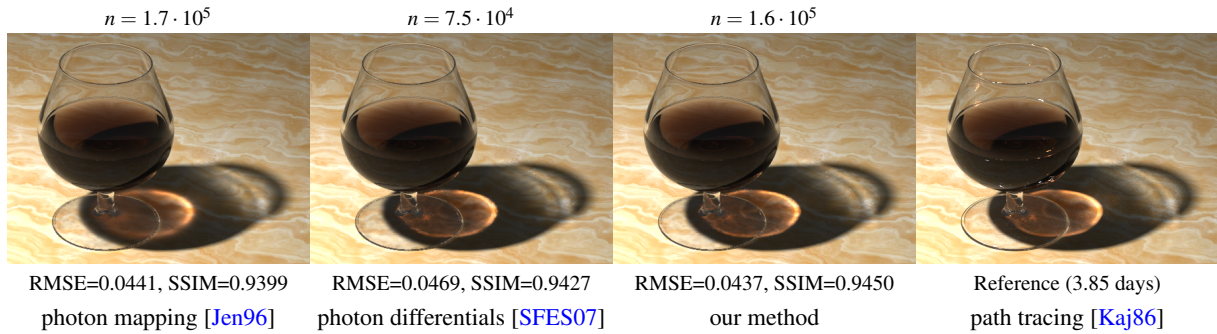


Figure 10: The cognac glass scene illuminated by a diffuse disk source. The first three images were rendered in equal time with 30 seconds for the rendering of the caustics. As opposed to these equal-time renderings, the path-traced reference image (rightmost) includes highlights (light paths $LS+E$).

whereas these are preserved by photon differentials. The equal-time renderings were allowed to spend only 3.3 seconds for the rendering of the caustics. With this budget, there was time to process 5.0 thousand caustic photons using standard photon mapping, 1.5 thousand using photon differential mapping, but 5.2 thousand using our method. This improvement in the number of elliptic caustic photons that we can process in equal time is quite significant. It is due to the splatting approach (the eye path map is faster to build and to search) and the faster density estimation (12). The latter contribution can also be used to improve photon differential mapping. This reduces the time required to render the caustics in the second column of Figure 2 to 3.0 seconds.

The alternative to Equation 12 is to invert a 3×3 matrix. One option is to use the method described by Doué [Dou94]. When using Equation 12, the improvement in total caustic rendering time varies a lot depending on the scene and the number of caustic photons n . In an isolated test, we are on average able to compute the matrix M_p 2.59 million times in one second using Doué’s method. Using Equation 12, we are able to compute this matrix 35.1 million times in one second. This means that we reduce the additional costs incurred by anisotropic density estimation by a factor 13.6.

Figure 10 is renderings of the cognac glass [Jen01] which is often used as a test scene for rendering caustics. The absorption of the cognac is that of a 40% Hennessy cognac [MS01]. The scene has soft caustics below the foot of the glass and sharp caustics in the shadow region. It is a more complex case as it involves multiple reflections and refractions from the glass and the cognac. Even so, photon differentials still have the ability to capture both the soft and the sharp illumination features. However, the quality metrics do not indicate a large improvement of the image as this scene has some very anisotropic photon footprints which proceed into parts of the caustic that should have remained dark.

The impact of our more accurate photon emission (2–3) is very small in terms of overall quality measurements

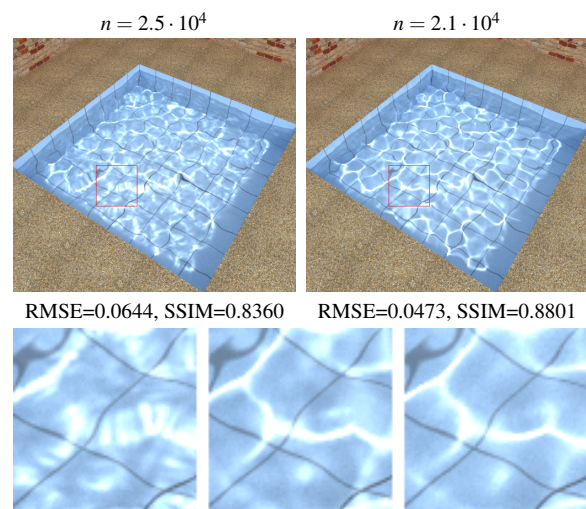


Figure 11: Swimming pool scene rendered in equal time using photon mapping [Jen96] (top left) and our method (top right). The bottom row is close-ups of the caustics in the red squares (left and middle), and the same part from the reference image is included to the right.

(RMSE and SSIM). The improvement applies to scenes with an area light source (Figures 1 and 10). Using a more arbitrary choice of directions for the initial orthogonal differential vectors when rendering the gold ring or the cognac glass, the quality measurements were degraded by only 0.5% or less. However, we also found that small noise-like photon differentials appear more often in inappropriate places if Equations 2 and 3 are not applied.

Figure 11 is renderings of a swimming pool. This type of scene is a typical test case for more advanced Monte Carlo methods like bidirectional path tracing and metropolis light transport [Vea97]. We use a directional light, so it is infeasible to render this image using standard path tracing. The

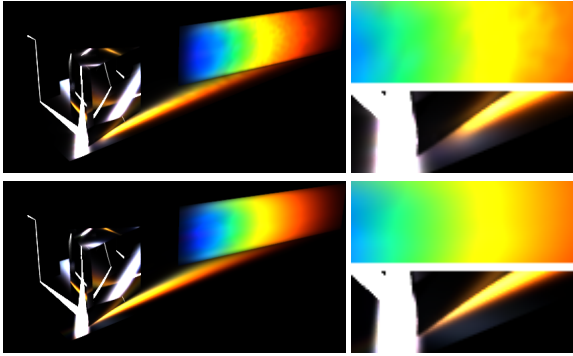


Figure 12: Dispersion prism experiment inspired by a photograph [WC83, Plate 1]. Rendered in equal time using photon mapping [Jen96] (top) and our method (bottom).

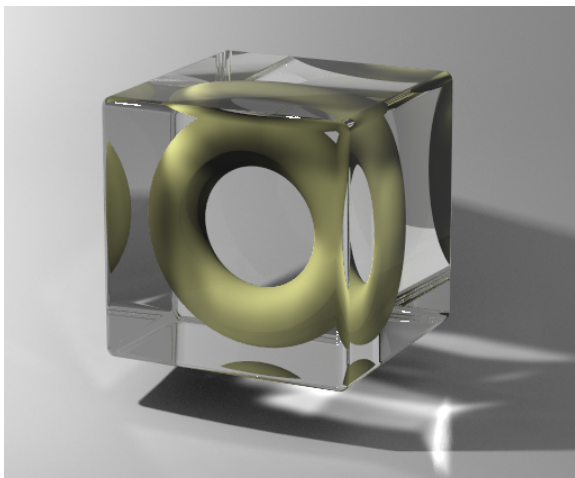


Figure 13: Embedded torus rendered using photon differential splatting with $s = 40$ and $n = 5.55 \cdot 10^5$. This rendering is included to illustrate that our method also works well for curved surfaces.

reference image was instead rendered using standard photon mapping with 10 million photons in the map. As indicated by the quality measurements, and as we can clearly observe in the close-ups, our method is particularly well-suited for this type of scene. The reason is that it neither requires long paths nor photons with large footprints.

Figure 12 is spectral renderings of Newton’s classical dispersion prism experiment. As the prism is placed on a table, this case requires a method that handles both soft and sharp caustic illumination. Compared with a photo of the experiment [WC83], the caustics on the table should be very sharp while the caustic on the screen should be very soft. In the sharp caustic on the table, which runs from the prism to the screen, the different colors in the dispersion pattern should be clearly distinguishable. It is nearly infeasible to

render this accurately using standard photon mapping. Even with millions of caustic photons, we still either get blurred edges on the table or noise on the screen. Photon differentials sharpen the caustics on the table, but the extensive smoothing necessary for the soft caustic on the screen still blurs the dispersion pattern in the sharp caustic on the table.

Figure 13 is a rendering of the embedded torus which appeared as a test scene in the original paper on progressive photon mapping [HOJ08]. It is included to support the claim that we can deal with topological bias as in standard photon mapping (see Section 3.3). For the embedded torus (and the dispersion prism experiment in Figure 12), we needed anti-aliasing in our caustics. Supersampling of pixels added some extra costs as we then needed a denser eye path map. We used a rather large smoothing parameter $s = 40$ to render the embedded torus. Nevertheless, we still capture the sharp caustics well using only $n = 5.55 \cdot 10^5$ photons, and our result compares well to the result obtained with progressive photon mapping [HOJ08].

5. Discussion

Density estimation entails bias [Sil86]. This bias is often divided into three categories [Sch03]: proximity bias, boundary bias, and topological bias. Proximity bias is the more fundamental, as it refers to the effect of using neighboring path vertices instead of the vertex that the path is currently at. Our results indicate that use of photon differentials reduces this kind of bias, especially around sharp illumination features. Boundary bias is when the filter kernel proceeds beyond the boundaries of the geometry. The result is a darkening toward object edges, as we are dividing by too large an area compared to what the geometry can support. To deal with boundary bias in a splatting context, we must either consider the geometry where the photon is splatted [LP03] or the eye path vertices in the vicinity of each photon path [HHK*07]. This is currently not a part of our implementation. Topological bias is overestimation of the illumination when a photon is incident on a surface which is not locally planar. In Section 3.3 we suggested ways of dealing with topological bias which are similar to what is possible in standard photon mapping.

The use of photon differentials has two basic issues. Photon tracing is more expensive, since there is an overhead in computing the differentials, and photon footprints may become highly anisotropic such that we get line-like illumination artifacts. The cognac glass renderings (Figure 10) illustrate both these issues. Looking at a particular region of the cognac glass caustic, see Figure 14, we can illustrate that highly anisotropic footprints are an important source of error when using photon differentials. A simple solution is to trace more photons. Other than that, we believe that an adaptive quadrature approach, where photons with too large and/or too anisotropic footprints are split and retraced, would be a good candidate to resolve this issue of extreme anisotropy.

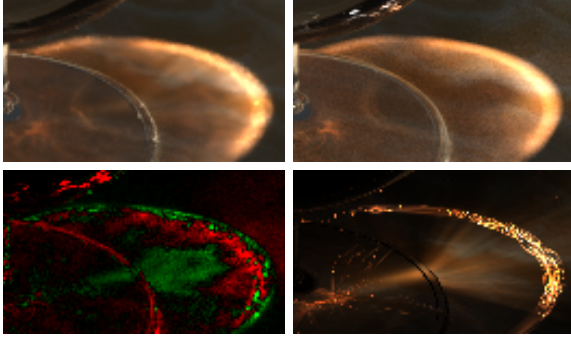


Figure 14: Part of the cognac glass caustic. From top left to bottom right: our method, path traced reference, difference image (red is negative error, green is positive error), and splatting of photons with highly anisotropic footprints only. The last two images have been scaled by 5. The metric used to identify highly anisotropic footprints is in Appendix B.

Finally, our splatting approach can be extended to include motion blur and other temporal aspects by using full spatio-temporal photon differentials [SFES11]. This means that we need third positional and directional differential vectors, where the partial derivatives are taken with respect to time. The formulae behind transfer, reflection, and refraction of spatio-temporal photon differentials are available in a technical report [SSE09]. Another extension would be to include glossy and diffuse reflections by splatting the footprints of path differentials [SW01, FD09]. The footprint size of a path differential after non-specular reflection is, however, largely based on heuristics.

6. Conclusion

We have presented a faster and more accurate way to render caustics using photon differentials. Better accuracy is obtained by more accurate emission of photon differentials from arbitrary light sources. A more efficient method is obtained by a faster transformation to filter space and by taking a splatting approach. Since our method is based on splatting, it can easily be accelerated further using rasterization. In comparison to standard photon mapping, the tracing of photon differentials carries some overhead, and highly anisotropic footprints sometimes cause rendering artifacts. In an equal-time comparison, these drawbacks mean that our method does not greatly improve the caustic illumination in scenes that require long paths and have highly anisotropic photon footprints. On the other hand, in scenes that mostly require short paths, the improvement is significant.

Acknowledgement. Thanks to Anders Wang Kristensen for the swimming pool scene.

Appendix A: Differential Vectors After First Transfer

In this appendix, we check that Equations 2 and 3 result in pairwise parallel vectors *after* the first transfer. Considering Equations 2, 8, and 9, we have $D_\theta \mathbf{x}' = t' \frac{|D_\theta \vec{\omega}|}{|D_u \mathbf{x}|} D_u \mathbf{x}'$. Thus, the vectors $D_u \mathbf{x}$ and $D_\theta \vec{\omega}$ are parallel after transfer to the first surface. To check the other pair of vectors, $D_v \mathbf{x}$ and $D_\phi \vec{\omega}$, we investigate whether it holds true that $D_v \mathbf{x}' \times D_\phi \mathbf{x}' = \mathbf{0}$. Inserting Equation 2 in Equation 3 (left and right), we get two triple vector products which we insert in Equations 8 and 9 (using v and ϕ subscripts) to get expressions for $D_v \mathbf{x}'$ and $D_\phi \mathbf{x}'$. Using that \vec{n} and $\vec{\omega}$ are unit vectors and that the cross product of a vector with itself is $\mathbf{0}$, we arrive at the desired result after application of some vector algebra.

Appendix B: Photon Footprint Anisotropy Metric

To measure the anisotropy of a photon footprint, we un-skew the footprint ellipse and take the ratio of the minor radius to the major radius. The anisotropy metric is then

$$m_a = \left| D_{\min} \mathbf{x}_p - \frac{D_{\max} \mathbf{x}_p \cdot D_{\min} \mathbf{x}_p}{|D_{\max} \mathbf{x}_p|^2} D_{\max} \mathbf{x}_p \right| |D_{\max} \mathbf{x}_p|^{-1},$$

where $D_{\max} \mathbf{x}_p$ and $D_{\min} \mathbf{x}_p$ refer to the positional differential vectors of longest and shortest length, respectively. By design, we have $m_a \in [0, 1]$, and $m_a = 1$ means that the footprint is isotropic. We use $m_a < 0.1$ to identify the highly anisotropic photon footprints rendered in the bottom right image of Figure 14.

References

- [Dou94] DOUÉ J.-F.: C++ vector and matrix algebra routines. In *Graphics Gems IV*, Heckbert P. S., (Ed.). Academic Press, San Diego, California, USA, 1994, pp. 534–557. 260
- [FD09] FABIANOWSKI B., DINGLIANA J.: Interactive global photon mapping. *Computer Graphics Forum (Proceedings of EGSR 2009)* 28, 4 (June–July 2009), 1151–1159. 254, 262
- [Fri12a] FRISVAD J. R.: Building an orthonormal basis from a 3d unit vector without normalization. *Journal of Graphics Tools* 16, 3 (August 2012), 151–159. 255
- [Fri12b] FRISVAD J. R.: Photon differentials: Adaptive anisotropic density estimation in photon mapping. In *State of the Art in Photon Density Estimation*, Hachisuka T., Jarosz W., (Eds.), ACM SIGGRAPH Course Notes, August 2012. Article 6. 255, 256
- [Gla95] GLASSNER A. S.: *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, California, 1995. 259
- [Hec90] HECKBERT P. S.: Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics (Proceedings of ACM SIGGRAPH 90)* 24, 4 (August 1990), 145–154. 252
- [HHK*07] HERZOG R., HAVRAN V., KINUWAKI S., MYSZKOWSKI K., SEIDEL H.-P.: Global illumination using photon ray splatting. *Computer Graphics Forum (Proceedings of Eurographics 2007)* 26, 3 (September 2007), 503–513. 253, 254, 257, 261
- [HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2009)* 28, 5 (December 2009), Article 141. 253

- [HOJ08] HACHISUKA T., OGAKI S., JENSEN H. W.: Progressive photon mapping. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2008)* 27, 5 (December 2008), Article 130. [253](#), [258](#), [261](#)
- [Ige99] IGEHY H.: Tracing ray differentials. In *Proceedings of ACM SIGGRAPH 1999* (Los Angeles, California, USA, August 1999), ACM Press/Addison-Wesley, New York, USA, pp. 179–186. [254](#), [256](#)
- [JC95] JENSEN H. W., CHRISTENSEN N. J.: Photon maps in bidirectional Monte Carlo ray tracing of complex objects. *Computers & Graphics* 19, 2 (March 1995), 215–224. [254](#)
- [Jen96] JENSEN H. W.: Global illumination using photon maps. In *Rendering Techniques '96* (June 1996), Pueyo X., Schröder P., (Eds.), Springer, Vienna, Austria, pp. 21–30. Proceedings of the Seventh Eurographics Workshop on Rendering, Porto, Portugal. [254](#), [255](#), [257](#), [258](#), [259](#), [260](#), [261](#)
- [Jen01] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. A K Peters, Natick, Massachusetts, 2001. [252](#), [260](#)
- [JNSJ11] JAROSZ W., NOWROUZEZAHRAI D., SADEGHI I., JENSEN H. W.: A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics* 30, 1 (January 2011), Article 5. [254](#)
- [JNT*11] JAROSZ W., NOWROUZEZAHRAI D., THOMAS R., SLOAN P.-P., ZWICKER M.: Progressive photon beams. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2011)* 30, 6 (December 2011), Article 181. [253](#), [254](#), [259](#)
- [Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proceedings of ACM SIGGRAPH 86)* 20, 4 (August 1986), 143–150. [252](#), [253](#), [260](#)
- [KD13] KAPLAYAN A. S., DACHSBACHER C.: Adaptive progressive photon mapping. *ACM Transactions on Graphics* 32, 2 (April 2013), Article 16. [253](#), [254](#)
- [KZ11] KNAUS C., ZWICKER M.: Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics* 30, 3 (May 2011), Article 25. [253](#)
- [LP03] LAVIGNOTTE F., PAULIN M.: Scalable photon splatting for global illumination. In *Proceedings of GRAPHITE 2003* (Melbourne, Australia, 2003), ACM, pp. 203–210. [253](#), [254](#), [257](#), [259](#), [261](#)
- [ML09] MCGUIRE M., LUEBKE D.: Hardware-accelerated global illumination by image space photon mapping. In *Proceedings of the Conference on High Performance Graphics 2009 (HPG '09)* (New Orleans, Louisiana, USA, 2009), ACM, pp. 77–89. [253](#)
- [MS01] MCCLURE W. F., STANFIELD D. L.: Near-infrared spectroscopy of biomaterials. In *Handbook of Vibrational Spectroscopy*, Chalmers J., Griffiths P., (Eds.), vol. 1: Theory and Instrumentation. John Wiley & Sons, Hoboken, New Jersey, USA, 2001. [260](#)
- [Mys97] MYSZKOWSKI K.: Lighting reconstruction using fast and adaptive density estimation techniques. In *Rendering Techniques '97* (June 1997), Dorsey J., Slusallek P., (Eds.), Springer, Vienna, Austria, pp. 251–262. Proceedings of the 8th Eurographics Workshop on Rendering, Saint-Etienne, France. [253](#), [254](#)
- [RDGK12] RITSCHEL T., DACHSBACHER C., GROSCH T., KAUTZ J.: The state of the art in interactive global illumination. *Computer Graphics Forum* 31, 1 (February 2012), 160–188. [253](#)
- [SB97] STÜRZLINGER W., BASTOS R.: Interactive rendering of globally illuminated glossy scenes. In *Rendering Techniques '97* (June 1997), Dorsey J., Slusallek P., (Eds.), Springer, Vienna, Austria, pp. 93–102. Proceedings of the 8th Eurographics Workshop on Rendering, Saint-Etienne, France. [253](#)
- [Sch03] SCHREGLE R.: Bias compensation for photon maps. *Computer Graphics Forum* 22, 4 (December 2003), 729–742. [253](#), [254](#), [261](#)
- [Sch09] SCHJØTH L.: *Anisotropic Density Estimation in Global Illumination*. PhD thesis, University of Copenhagen, Denmark, 2009. [254](#), [257](#), [258](#)
- [SFES07] SCHJØTH L., FRISVAD J. R., ERLEBEN K., SPORRING J.: Photon differentials. In *Proceedings of GRAPHITE 2007* (Perth, Australia, December 2007), ACM, pp. 179–186. [252](#), [253](#), [254](#), [256](#), [257](#), [259](#), [260](#)
- [SFES11] SCHJØTH L., FRISVAD J. R., ERLEBEN K., SPORRING J.: Photon differentials in space and time. In *Computer Vision, Imaging and Computer Graphics: Theory and Applications*, Richard P., Braz J., (Eds.), vol. 229 of *Communications in Computer and Information Science*. Springer, Berlin Heidelberg, Germany, December 2011, pp. 274–286. [262](#)
- [Sil86] SILVERMAN B. W.: *Density Estimation for Statistics and Data Analysis*, vol. 26 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, UK, 1986. [257](#), [261](#)
- [SJ09] SPENCER B., JONES M. W.: Into the blue: Better caustics through photon relaxation. *Computer Graphics Forum (Proceedings of Eurographics 2009)* 28, 2 (April 2009), 319–328. [253](#), [259](#)
- [SJ13] SPENCER B., JONES M. W.: Progressive photon relaxation. *ACM Transactions on Graphics* 32, 1 (January 2013), Article 7. [253](#)
- [SOS08] SCHJØTH L., OLSEN O. F., SPORRING J.: Diffusion based photon mapping. *Computer Graphics Forum* 27, 8 (December 2008), 2114–2127. [253](#), [255](#), [257](#), [258](#), [259](#)
- [SSE09] SPORRING J., SCHJØTH L., ERLEBEN K.: *Spatial and Temporal Ray Differentials*. Tech. Rep. 2009/04, Department of Computer Science, University of Copenhagen, Denmark, 2009. [262](#)
- [SW01] SUYKENS F., WILLEMS Y. D.: Path differentials and applications. In *Rendering Techniques 2001* (June 2001), Gortler S. J., Myszkowski K., (Eds.), Springer, Vienna, Austria, pp. 257–268. Proceedings of the 12th Eurographics Workshop on Rendering, London, UK. [254](#), [255](#), [256](#), [262](#)
- [Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, California, USA, December 1997. [257](#), [260](#)
- [WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (April 2004), 600–612. [258](#)
- [WC83] WILLIAMSON S. J., CUMMINS H. Z.: *Light and Color in Nature and Art*. John Wiley & Sons, Hoboken, New Jersey, USA, 1983. [261](#)
- [WD08] WYMAN C., DACHSBACHER C.: Reducing noise in image-space caustics with variable-sized splatting. *Journal of Graphics, GPU, and Game Tools* 13, 1 (January 2008), 1–17. [253](#)
- [Wym08] WYMAN C.: Hierarchical caustic maps. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games (I3D 2008)* (Redwood City, California, USA, February 2008), ACM, pp. 163–171. [253](#)
- [YWC*10] YAO C., WANG B., CHAN B., YONG J., PAUL J.-C.: Multi-image based photon tracing for interactive global illumination of dynamic scenes. *Computer Graphics Forum (Proceedings of EGSR 2010)* 29, 4 (June 2010), 1315–1324. [253](#)