

Photon Mapping with Visible Kernel Domains

Romuald Perrot ·
Lilian Aveneau ·
Frédéric Mora ·
Daniel Meneveau

Received: date / Accepted: date

Abstract Despite the strong efforts made in the last three decades, lighting simulation systems still remain prone to various types of imprecisions. This paper specifically tackles the problem of biases due to density estimation used in photon mapping approaches. We study the fundamental aspects of density estimation, and exhibit the need for handling visibility in the early stage of the kernel domain definition. We show that properly managing visibility in the density estimation process allows to reduce or to remove biases all at once. In practice, we have implemented a 3D product kernel based on a polyhedral domain, with both point-to-point and point-to-surface visibility computation. Our experimental results illustrate the enhancements produced at every stage of density estimation, for direct photon maps visualization and progressive photon mapping.

1 Introduction

The need for accurate, physically or photo realistic rendering systems has motivated many research efforts in the field of lighting simulation [22,4]. This paper specifically focuses on density estimation methods, widely used with photon mapping approaches [12,13,6,16] or extensions of bidirectional path tracing for instance [1,5,7,23].

Density estimation consists in approximating a value at a given location, based on several known samples in its neighborhood. Therefore, these latter are weighted

according to a *kernel* function. The *kernel estimation domain* corresponds to the region where samples are used by the kernel function. This method is known to be biased for several reasons, identified for a long time in the mathematical theory [25]. Several authors in the computer graphics community have studied the properties of kernel functions as well as normalization issues for bias reduction. The commonly admitted bias classification can be associated with various configurations in the case of photon-mapping approaches (illustrated in Figure 2). So far, they have been essentially managed independently [24,27,10]: (a) Proximity bias; (b) Boundary bias; (c) Topological bias and (d) Occlusion bias. Unfortunately, the corrections proposed in the literature partially deal with the problem since reducing only one bias may increase the others. As shown by these authors, biases are essentially due to an inaccurate definition of the kernel domain.

This paper shows why and how visibility should be employed for kernel normalization. Our global density estimation proves consistent and makes it possible to remove topological bias, to provide consistent density estimation, and to noticeably reduce the other biases altogether. More precisely, the contributions of this paper concern:

- An analysis of density estimation emphasizing the importance of visibility during the normalization process;
- A practical framework dedicated to consistent density estimation, which introduces visibility for determining the estimation domain;
- A practical implementation of density estimation methods for rendering purpose, with a 3-dimensional product kernel, avoiding topological bias, and reducing boundary and occlusion biases.

We illustrate the results obtained with our methods for direct density estimation [12,13] and progressive photon mapping [6,16]. They can be observed more precisely on our supplemental HTML page ([Link to HTML](#)). Typical difficult cases are experimented: Large kernel domains that increase bias; Small surfaces that tend to underestimate irradiance; Many occlusions that generally cause light leaking; Occluded caustics. Our results show that accounting for visibility during the density estimation process effectively reduces both biases and visual artifacts.

The remainder of this paper is organized as follows: Section 2 presents previous work concerning bias management in the context of photon mapping; Section 3 discusses kernel normalization, and introduces our formulation for density estimation based on the visible kernel domain; Section 4 describes the general methodology of our method; Section 5 presents the implementa-

R. Perrot, L. Aveneau, D. Meneveau
XLIM/ASALI – UMR CNRS 7252, University of Poitiers

F. Mora
XLIM/ASALI – UMR CNRS 7252, University of Limoges

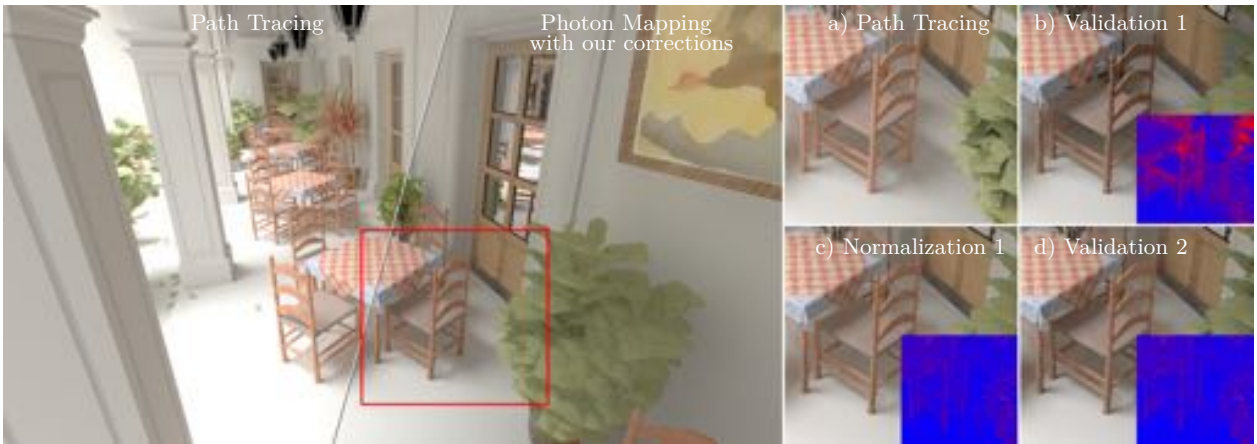


Fig. 1: San Miguel scene, comparisons between path tracing and density estimation with our photon mapping normalization method. a) path tracing reference image; b) photon validation from the observer side, and difference with path tracing; c) photon validation from the observer side plus normalization, and difference with path tracing; d) photon validation and normalization from observer side plus photon validation from photon ancestor side, and image difference with path tracing.

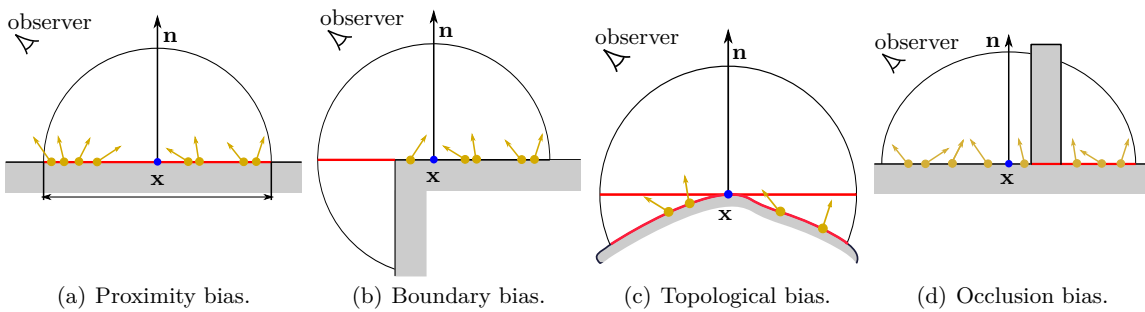


Fig. 2: Bias classification for photon mapping density estimation with limited tangent kernel support [24]. The stored photons are depicted in gold, the estimation point is \mathbf{x} . Red lines indicate the cause of bias: a) Proximity bias is due to the use of photons not precisely located at \mathbf{x} ; b) Boundary bias appears if the kernel domain contains parts where no photon can be found; c) Topological bias comes from the distortion due to photons projection onto the kernel domain; d) Occlusion bias corresponds to a combination of proximity and boundary biases, appearing when photons are used with an incorrect domain.

tion details and discusses the obtained results, before conclusion in Section 6.

2 Related Work

Global illumination methods based on photon mapping have been widely explored in the literature. Ray paths are traced from light sources, light interactions with the scene geometry are stored as *photons*, organized in a data structure called *photon map*. The rendering phase relies on density estimation, using photons belonging to the kernel estimation domain.

Density estimation is inherently biased, for a variety of reasons (see Figure 2) that have been classified by Schregle [24]:

- *Proximity bias*, due to the use of photons that are not located exactly on the estimation point (Figure 2.a);
- *Boundary bias*, induced by an overestimated kernel domain, and resulting in underestimate irradiance (Figure 2.b);
- *Topological bias*, corresponding to a wrong estimation domain for curved surfaces, resulting in overestimate irradiance (Figure 2.c).

Some authors [27,10] discriminate an additional category:

- *Occlusion bias*, due to the use of photons not visible from the observer or which ancestors are not visible from the estimation point. It leads for instance to light leaking under walls (Figure 2d).

Proximity bias management

The approach proposed by Schregle [24] relies on a series of density estimations according to k -nearest neighbors while varying the number k of photons. Conversely, Hachisuka et al. propose to progressively refine the kernel support area [6]. These methods propose a statistical approach to manage bias, without actually removing artifacts. Unfortunately, proximity bias inherently remains as long as density estimation is used.

Boundary bias management

Several authors propose to use an approximation of the surfaces that hold the photons in the kernel domain. The goal is to determine more precisely the corresponding normalization factor, so as to reduce boundary bias. For instance Lavignotte and Paulin [18] virtually extend scene surfaces; Jensen [13] uses convex hulls of photons; Tobler et al. employ *octoboxes* [27]. None of these approaches completely solves the boundary bias problem since the surface area estimation relies on an approximation leading to numerical instabilities in dark regions hit only by few photons. For a more precise domain estimation, Lavignotte et al. [19] propose to project photon splats in screen space and to estimate the intersection between the kernel domain disk and the underlying surface. Lastra et al. [17] and Havran et al. [9] propose to use photon paths instead of their impacts and determine the intersection between the photon paths and a virtual tangent disk (representing the density estimation domain) centered at the estimation point. These methods may also use photons that should not contribute to the estimation since they may come from surfaces that are not visible from the estimation point. Overall, the boundary bias is due to the lack of a precise estimation of the kernel domain, which has still not been completely solved so far to our knowledge.

Topological bias management

The underlying surface around the estimation point of often considered as locally flat for a small estimation domain. Unfortunately, this approximation changes the actual irradiance associated with photons because the normal does not correspond. Ray maps implicitly deal with topological bias since the underlying surface is ignored [9].

Occlusion bias management

Several authors have tackled the problem of occlusions

in the density estimation process. Tobler et al. [27] take visibility into account for reducing occlusion bias using a filter for validating photons. The surface area estimation only relies on the photons located in the initial domain. The approximation of the corresponding underlying surface leads to numerical instabilities when the region contains only few photons. Chen et al. [2] introduce *grouped photon mapping* for reducing boundary and occlusion biases. Although these papers propose a visible bias reduction, they do not completely solve the problem due to remaining approximations.

Discussion

To date, density estimation biases have mainly been processed independently, and each solution remains approximative, though producing better results. Some authors propose to avoid the density estimation process. For instance, Qin et al. [23] connect photon paths and eye paths, as an extension of both vertex connection and merging [5], and path space extension [7]. A russian roulette is employed to estimate the probability to link paths, with only few Bernoulli trials. Such a process actually removes biases, but variance increases. In this paper, we propose an analysis of density estimation that leads to a general framework used to reduce bias. Moreover our approach can be plugged into any photon mapping software with density estimation.

3 Visibility and 3D Kernel Normalization

Based on the formulation of density estimation, this section proposes a study on kernel domains, their 3D representations, and visibility issues. We introduce the *visible domain*, corresponding to the region of the kernel domain where photons remain actually valid. We also manage exact kernel normalization with 3D kernels, so as to: (i) Make the boundary bias consistent; (ii) Remove topological bias.

3.1 Photon Mapping and Visible Domains

This section presents the notations used in this paper, while introducing the visible domain. The rendering equation employed in this paper corresponds to the formulation proposed by Kajiya [15]. The radiance observed from a point \mathbf{y} coming from a point \mathbf{x} is given by:

$$L_i(\mathbf{y}, \mathbf{x} \rightarrow \mathbf{y}) = V(\mathbf{x} \leftrightarrow \mathbf{y}) \times L_o(\mathbf{x}, \mathbf{x} \rightarrow \mathbf{y}) \\ = V(\mathbf{x} \leftrightarrow \mathbf{y}) \times \left[L_e(\mathbf{x}, \mathbf{x} \rightarrow \mathbf{y}) + \int_{\Sigma} L_o(\mathbf{z}, \mathbf{z} \rightarrow \mathbf{x}) f_r(\mathbf{x}, \mathbf{z} \rightarrow \mathbf{x} \rightarrow \mathbf{y}) G(\mathbf{x}, \mathbf{z}) d\mathbf{z} \right], \quad (1)$$

where L_i (resp. L_o) is the incoming (resp. outgoing) radiance; L_e is the self-emitted radiance; f_r is the bidirectional reflectance distribution function (BRDF from now on); Σ is the set of surfaces in the scene, and G is the following geometrical term:

$$G(\mathbf{x}, \mathbf{z}) = V(\mathbf{x} \leftrightarrow \mathbf{z}) \frac{|\mathbf{n}_{\mathbf{x}} \cdot \omega_i| |\mathbf{n}_{\mathbf{z}} \cdot \omega_o|}{\|\mathbf{x} - \mathbf{z}\|^2},$$

where ω_i is the incoming light direction $\mathbf{x} \rightarrow \mathbf{z}$ and ω_o is the outgoing light direction $\mathbf{z} \rightarrow \mathbf{x}$; $V(\mathbf{a} \leftrightarrow \mathbf{b})$ is the visibility term between two points \mathbf{a} and \mathbf{b} , equal to 1 if they are visible, and 0 otherwise. With photon mapping, density estimation is used to estimate irradiance at \mathbf{x} , reflected toward a given point \mathbf{y} .

Generally speaking, photon-based approaches rely on two main steps, *photon tracing* and *irradiance estimation*. During photon tracing, random walk light paths produce photon positions \mathbf{x}_i . For each photon \mathbf{x}_i , the previous photon in the light path is called *ancestor*, and its position is denoted as \mathbf{x}_i^{-1} . The incident direction associated with a photon position \mathbf{x}_i is $\mathbf{x}_i^{-1} \rightarrow \mathbf{x}_i$, and the corresponding carried energy is α_i . According to the terminology employed by Veach [28] with the notations given by Pharr et al. [22], the energy α_i associated with a photon is estimated as follows:

$$\alpha_i = \alpha_{i-1}^* \frac{f_r(\mathbf{x}_i, \mathbf{x}_i^{-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{y}) |\mathbf{n}_{\mathbf{x}_i} \cdot \omega_o|}{q(\mathbf{x}_i) p(\mathbf{x}_i, \omega_o)},$$

with $p(\mathbf{x}_i, \omega_o)$ the probability to choose a random outgoing direction $\omega_o = \mathbf{x}_i \rightarrow \mathbf{y}$, and using a russian roulette; $q(\mathbf{x}_i)$ is the probability with which the random walk is continued at \mathbf{x}_i ; $\mathbf{n}_{\mathbf{x}_i}$ is the surface normal at point \mathbf{x}_i and α_{i-1}^* is the energy associated with the ancestor of \mathbf{x}_i , defined as:

$$\alpha_{i-1}^* = \begin{cases} \alpha_i^e & \text{if } \mathbf{x}_i^{-1} \text{ is located on a light source,} \\ \alpha_{i-1} & \text{otherwise,} \end{cases}$$

where α_i^e is the energy of a photon emitted by a light source:

$$\alpha_i^e = \frac{L_e(\mathbf{x}_i, \omega_o) |\mathbf{n}_{\mathbf{x}_i} \cdot \omega_o|}{p(\mathbf{x}_i, \omega_o)}.$$

Based on the photons spread in the scene, a density estimation process estimates incoming radiance and determines the reflected light flux. In the general case of a d -dimensional function $f(\mathbf{x})$, the density estimation of f at a given point \mathbf{x} from known samples in its neighborhood requires a d -dimensional kernel density estimator $\langle f_{\mathbf{H}}(\mathbf{x}) \rangle$:

$$\langle f_{\mathbf{H}}(\mathbf{x}) \rangle = \frac{1}{n |\mathbf{H}|} \sum_{i=1}^n K_d(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)), \quad \mathbf{x} \in \mathbb{R}^d, \quad (2)$$

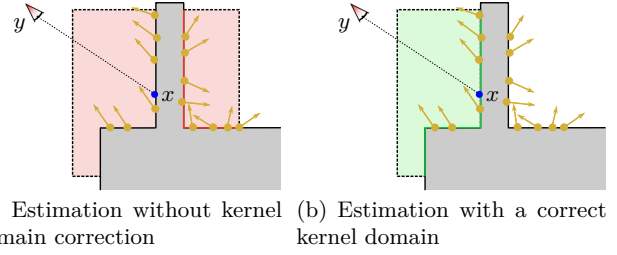


Fig. 3: Visibility from the estimation point allows to reduce the kernel support, correcting boundary and occlusion biases.

where K_d is a multivariate kernel function and \mathbf{H} is the $(d \times d)$ nonsingular matrix generalizing the bandwidth; $\{\mathbf{x}_i\}$ (with $|\{\mathbf{x}_i\}|=n$) corresponds to some estimation locations where f is supposed to be known.

We propose to use a three-dimensional kernel, that avoids topological bias since all the surfaces located in the initial domain are directly handled. In addition, we show that occlusion and boundary biases are reduced and made consistent using visibility as described in this paper. Visibility between an observation point \mathbf{y} and photon locations $\{\mathbf{x}_i\}$ around \mathbf{x} should be handled in the process, as illustrated in Figure 3. Using density estimation with 3D kernels leads to the following approximation of Equation 1 for the reflected radiance:

$$L_i(\mathbf{y}, \mathbf{x} \rightarrow \mathbf{y}) \approx \frac{1}{n |\mathbf{H}|} \sum_{i=1}^n K_3(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)) L_o(\mathbf{x}_i, \mathbf{x}_i \rightarrow \mathbf{y}) V(\mathbf{x}_i \leftrightarrow \mathbf{y}),$$

where K_3 is a 3D kernel of bandwidth \mathbf{H} (See Appendix A), and L_o corresponds to the radiance reflected from position \mathbf{x}_i toward \mathbf{y} . The total estimated radiance reflected from \mathbf{x} toward \mathbf{y} depends on the incoming energy and BRDF associated with the photons $\{\mathbf{x}_i\}$ located in the neighborhood of \mathbf{x} :

$$L_i(\mathbf{y}, \mathbf{x} \rightarrow \mathbf{y}) \approx \frac{1}{n |\mathbf{H}|} \sum_{i=1}^n K_3(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)) f_r(\mathbf{x}_i, \mathbf{x}_i^{-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{y}) \times L_o(\mathbf{x}_i^{-1}, \mathbf{x}_i^{-1} \rightarrow \mathbf{x}_i) G(\mathbf{x}, \mathbf{x}_i^{-1}) V(\mathbf{x}_i \leftrightarrow \mathbf{y}). \quad (3)$$

In practice, this estimation should also account for visibility between \mathbf{x} and photon ancestors $\{\mathbf{x}_i^{-1}\}$, for all the considered photons $\{\mathbf{x}_i\}$ in the neighborhood of \mathbf{x} . Using the carried energy α_i of the i^{th} photon located

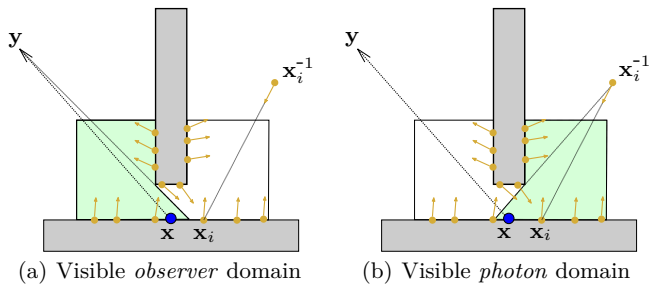


Fig. 4: The visible domain is defined as the intersection of: (a) The visible observer domain that corresponds to the kernel domain actually visible from the observer \mathbf{y} and (b) the visible photon domain, defined for each photon \mathbf{x}_i and which corresponds to the kernel domain actually visible from the ancestor \mathbf{x}_i^{-1} .

at \mathbf{x}_i leads to:

$$L_i(\mathbf{y}, \mathbf{x} \rightarrow \mathbf{y}) \approx \frac{1}{n |\mathbf{H}|} \sum_{i=1}^n K_3(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)) V(\mathbf{x}_i \leftrightarrow \mathbf{y}) \times \alpha_i f_r(\mathbf{x}_i, \mathbf{x}_i^{-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{y}) V(\mathbf{x} \leftrightarrow \mathbf{x}_i^{-1}). \quad (4)$$

Two conditions are thus mandatory for the kernel domain associated with K_3 : (i) Photons not visible from \mathbf{y} should not be used during the density estimation; (ii) Visibility between the density estimation location \mathbf{x} and the ancestor \mathbf{x}_i^{-1} should be ensured for all photons \mathbf{x}_i . These two conditions ensure the removal of light leaking under walls for instance. Unfortunately managing the visibility produces new domain boundaries, that should be handled in the estimation process (as described in Section 3.2).

Let us define $\mathcal{D}(\mathbf{y}, \mathbf{x}) \subset \mathbb{R}^3$, the *visible observer domain*, as the geometry located into the kernel domain centered on \mathbf{x} and visible from \mathbf{y} . This set contains the region of the density estimation domain visible from \mathbf{y} , where photons can be used (see Figure 4a).

Similarly, we define $\mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x}) \subset \mathbb{R}^3$, the *visible photon domain* (associated with each photon), as the geometry located into the kernel domain centered on \mathbf{x}_i and visible from its ancestor \mathbf{x}_i^{-1} (see Figure 4b). This domain should contain the estimation point \mathbf{x} , and corresponds in practice to the second visibility term $V(\mathbf{x} \leftrightarrow \mathbf{x}_i^{-1})$ in Equation 4.

Finally, the *visible kernel domain* associated with a given viewpoint \mathbf{y} , an estimation point \mathbf{x} , and a given photon \mathbf{x}_i is defined as:

$$\mathbb{V}(\mathbf{y}, \mathbf{x}, \mathbf{x}_i) = \mathcal{D}(\mathbf{y}, \mathbf{x}) \cap \mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x}). \quad (5)$$

3.2 Kernel Normalization

With *kernel normalization*, density estimation becomes consistent [14]. In practice, the normalization process solves boundary bias, caused by discontinuities in the estimation domain.

Boundary bias is due to the use of the same kernel domain at every estimation point, with the assumption that the underlying surface actually fits the kernel domain. However, in many cases, this assumption is wrong (see Figure 2b). With photon mapping, the associated kernel function should thus be normalized according to the estimation domain (it should integrate to 1 on the domain, as explained in Appendix B).

We propose to use the normalization method described by Jones [14] on the kernel domain (see Appendix B, Equations 20 and 21). Let us define the normalization factor $\mathbb{K}_3(\mathbf{x}, \mathcal{D})$ as:

$$\mathbb{K}_3(\mathbf{x}, \mathcal{D}) = \int_{\mathcal{D}} K_3(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{u})) d\mathbf{u}. \quad (6)$$

This factor is related to the kernel domain only; it is not used to link or validate photon paths with observer paths. Note that all the photons visible from the observer side are actually processed (there is no probabilistic selection). This normalization factor corresponds to the integration of the kernel over the bounded estimation domain \mathcal{D} . It should consider all the pieces of surface actually visible from the observer and from photons ancestors. With the visible observer domain, \mathcal{D} corresponds to $\mathcal{D}(\mathbf{y}, \mathbf{x})$, while with the visible photon domain, \mathcal{D} is $\mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x})$.

Considering the visible observer domain $\mathcal{D}(\mathbf{y}, \mathbf{x})$, the normalized kernel can be directly introduced into Equation 4:

$$L_i(\mathbf{y}, \mathbf{x} \rightarrow \mathbf{y}) \approx \frac{1}{n |\mathbf{H}| \mathbb{K}_3(\mathbf{x}, \mathcal{D}(\mathbf{y}, \mathbf{x}))} \sum_{i=1}^n K_3(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)) \times V(\mathbf{x}_i \leftrightarrow \mathbf{y}) \alpha_i f_r(\mathbf{x}_i, \mathbf{x}_i^{-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{y}) V(\mathbf{x} \leftrightarrow \mathbf{x}_i^{-1}).$$

The visible photon domain $\mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x})$ should also be accounted for. In this case, a normalization factor defined on the final domain $\mathbb{V}(\mathbf{y}, \mathbf{x}, \mathbf{x}_i)$ should be associated with each photon, leading to the following estimation:

$$L_i(\mathbf{y}, \mathbf{x} \rightarrow \mathbf{y}) \approx \frac{1}{n |\mathbf{H}|} \sum_{i=1}^n \frac{1}{\mathbb{K}_3(\mathbf{x}, \mathbb{V}(\mathbf{y}, \mathbf{x}, \mathbf{x}_i))} K_3(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)) V(\mathbf{x}_i \leftrightarrow \mathbf{y}) \times \alpha_i f_r(\mathbf{x}_i, \mathbf{x}_i^{-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{y}) V(\mathbf{x} \leftrightarrow \mathbf{x}_i^{-1}). \quad (7)$$

4 Methodology

This section provides an overview of the complete normalization process, including the visible domain computation.

With photon mapping, kernels are usually defined radially, corresponding to disks in 2D and spheres in 3D. Both dimensions have been used in the literature, but managing boundary bias is difficult due to the intersection between the polygonal scene surfaces and the initial kernel domain. With radial domains, these intersections produce parts of conics cut by polygon edges.

Instead, we propose to use a polyhedral domain, with a product kernel function. In practice, this choice simplifies the intersection computations, while increasing robustness and numerical stability during visibility estimation and kernel normalization.

4.1 Overview

Our method is applied according to the following five main steps.

1. Given the estimation point \mathbf{x} , the polygons contained in the initial kernel domain are clipped so as to define the corresponding surfaces. We propose to employ a cubic kernel domain instead of a spherical one, for simplifying polygon cuts.
2. This set of surfaces is restricted to the visible observer domain $\mathcal{D}(\mathbf{y}, \mathbf{x})$, thanks to point-to-surface visibility requests. The photons lying on $\mathcal{D}(\mathbf{y}, \mathbf{x})$ are selected using point-to-point visibility requests.
3. The visible photon domain $\mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x})$ is a new restriction defined for each photon \mathbf{x}_i , only if \mathbf{x} is visible from \mathbf{x}_i^{-1} . $\mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x})$ corresponds to the region of $\mathcal{D}(\mathbf{y}, \mathbf{x})$ seen from each photon ancestor \mathbf{x}_i^{-1} . This process requires point-to-surface visibility requests.
4. Based on this new restriction, the normalization factor \mathbb{K}_3 (Equation 6) is calculated. Therefore, we propose both an exact solution and a Monte Carlo approximation (see Section 4.3).
5. The final density estimation process associated with a given viewpoint \mathbf{y} and an estimation point \mathbf{x} is performed thanks to Equation 7.

4.2 Visibility computation

Given an estimation point \mathbf{x} , let us consider the set of photons $\{\mathbf{x}_i\}$ located in the initial domain. Each photon \mathbf{x}_i should be visible from the observer \mathbf{y} , and the estimation point \mathbf{x} should be visible from ancestors \mathbf{x}_i^{-1} . This selection of photons (called *photon validation* from now on) corresponds to point-to-point visibility.

In addition, the estimation of $\mathbb{V}(\mathbf{y}, \mathbf{x}, \mathbf{x}_i) = \mathcal{D}(\mathbf{y}, \mathbf{x}) \cap \mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x})$ requires point-to-surface visibility, known as difficult to estimate. We have compared two methods for this computation: (a) Exact point-to-surface methods [26, 21, 8, 20], and (b) Monte Carlo ray-tracing and point-to-point visibility.

4.3 Kernel normalization

As explained above, we propose to use a polyhedral domain, with a product kernel function. Appendix A provides the mathematical expression for product kernels in any dimension and appendix B derives the expression for the normalization factor. Any product kernel function can be used, and we have chosen the Epanechnikov product kernel for its efficiency. It has proven to be accurate with proximity bias management [25]:

$$K_3^e(x, y, z) = \left(\frac{3}{4}\right)^3 (1-x^2)(1-y^2)(1-z^2).$$

Unfortunately, the integration required for $K_3^e(x, y, z)$ (Equation 22) is not straightforward since the kernel domain $\mathcal{D}(\mathbf{y}, \mathbf{x})$, is defined by surfaces. It can be performed using a change of variable, as an integral over triangles. For a given triangle defined by three vertices \mathbf{a} , \mathbf{b} , \mathbf{c} , and using barycentric coordinates (γ, α, β) , the normalization of the Epanechnikov product kernel corresponds to:

$$\mathbb{K}_3^e = J \int_0^1 \int_0^{1-\alpha} K_3^e(x, y, z) d\beta d\alpha, \quad (8)$$

where $(x, y, z) = \mathbf{a} + \alpha\mathbf{b} + \beta\mathbf{c}$, and J is the Jacobian corresponding to the change of variables $(x, y, z) \mapsto (\gamma, \alpha, \beta)$:

$$J = \det \begin{vmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \end{vmatrix} = \frac{1}{(\mathbf{a}\mathbf{b} \times \mathbf{a}\mathbf{c}) \cdot \mathbf{a}}.$$

An exact value of this integral can be obtained using a computer algebra system. It results in a polynomial expression (not included because of its length) that only depends on the triangle vertices \mathbf{a} , \mathbf{b} , \mathbf{c} .

From the observer viewpoint, each polygon located in the kernel domain is split depending on the visible surfaces. In practice, we have implemented two different methods: (a) Exact point-to-surface visibility algorithms which split the polygons that can be directly used in Equation 8, and (b) A Monte Carlo process, consisting in uniformly sampling points on each polygon located in the kernel domain. For this latter, the integration over $\mathcal{D}(\mathbf{y}, \mathbf{x})$, $\mathcal{D}(\mathbf{x}_i^{-1}, \mathbf{x})$ or $\mathbb{V}(\mathbf{y}, \mathbf{x}, \mathbf{x}_i)$ only relies on point-to-point visibility tests. Let us recall that the kernel domain associated with K_3 is a 3D

region; Its unbiased Monte Carlo normalization relies on point samples \mathbf{s}_i uniformly spread on the surfaces it contains (note that these points are completely independent from photons). Given n samples \mathbf{s}_i and an observer position \mathbf{y} , an estimator of the normalization factor \mathbb{K}_3 is:

$$\langle \mathbb{K}_3(\mathbf{y}, \mathbf{x}) \rangle \approx \frac{1}{n} \sum_{i=1}^n \frac{K_3(\mathbf{H}^{-1}(\mathbf{x}-\mathbf{s}_i))}{p(\mathbf{s}_i)} V(\mathbf{y} \leftrightarrow \mathbf{s}_i), \quad (9)$$

where $p(\mathbf{s}_i)$ is the probability density function used to draw the sample \mathbf{s}_i on the surfaces of the domain. In practice, $p(\mathbf{s}_i) = 1/A$, A being the surface area. The main advantage of such a solution is its constant cost: For a given scene, it only depends on the number of used samples.

5 Implementation and Results

The kernel normalization process handles the observer and photon domains. It has been implemented and applied to various scenes, with two photon mapping approaches:

1. Direct visualization of photon maps using density estimation, with fixed bandwidth h [12,13]; Irradiance estimate is applied to a 3D domain corresponding to a cube centered at the estimation point;
2. Progressive photon mapping (PPM) [6], with Knaus and Zwicker [16] implementation.

The experimental results presented in this paper have been produced with a bi-Xeon X5650 processor with 24 Gb RAM, using 24 threads. Our ray-tracing system is implemented in C, accelerated with Embree [29], that includes a QBVH data structure [3]; The method we employ for exact point-to-surface visibility determines the visible parts of a surface \mathcal{S} from a given point \mathbf{y} or \mathbf{x}_i^{-1} depending on the used domain. Any point-to-surface visibility algorithm can be used with our corrections, so as to compute visible parts of a set of surfaces from any given point of view. We have employed the hierarchical approach proposed in [20], which has the advantage of being efficient for coherent requests.

Figure 5 illustrates the scenes used for producing the results presented in this paper. Note that Half and Shutter scenes are defined so as to clearly illustrate the advantages of handling visibility during the density estimation process. The other scenes are more complex, so as to illustrate scalability. Difference images are provided based on HDR pixel values.

Kernel normalization requires to estimate the normalization factor corresponding to the integral given in

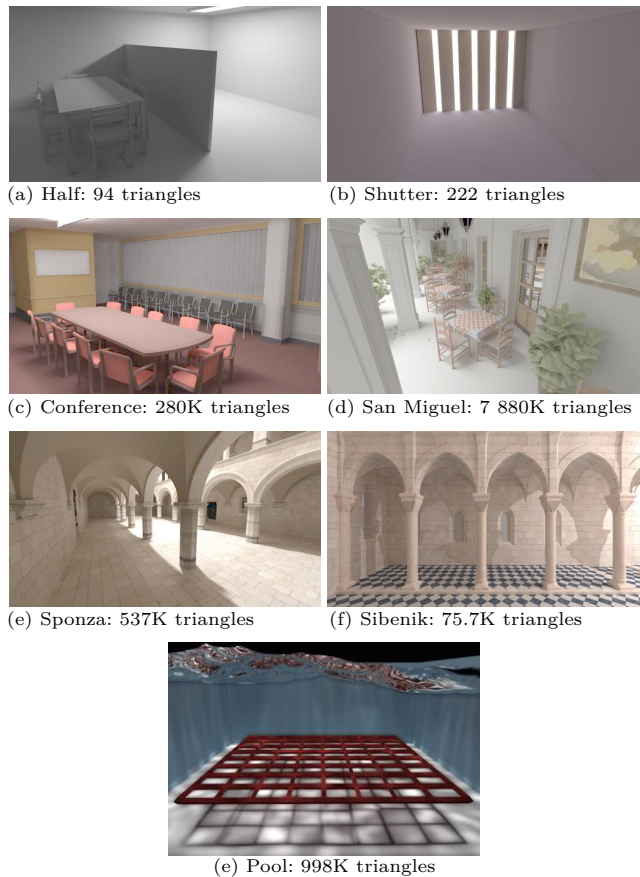


Fig. 5: Test scenes employed for illustrating our density estimation process.

Equation 6. We have implemented both Monte Carlo approximation (Equation 9, using $n=512$ samples in our implementation, corresponding to a good experimental trade-off between computation time and visual quality), and exact visibility requests with analytical piecewise integration.

Figures 6 and 7 illustrate our normalization process in the case of direct visualization of photon maps. Images concerning progressive photon mapping are shown in Figure 8. We also illustrate the case of occluded caustics, including a specular reflexion (mirror) in Figure 10. All our results can be interactively compared in the following HTML link [\(Additional Material\)](#).

5.1 General process

Density estimation is performed for a given estimation point \mathbf{x} , with an initial kernel domain that does not include visibility nor normalization. The four steps of our method is illustrated in the next paragraphs with direct visualization of photon maps. **In this section, comparisons are made using reference images correspond-**

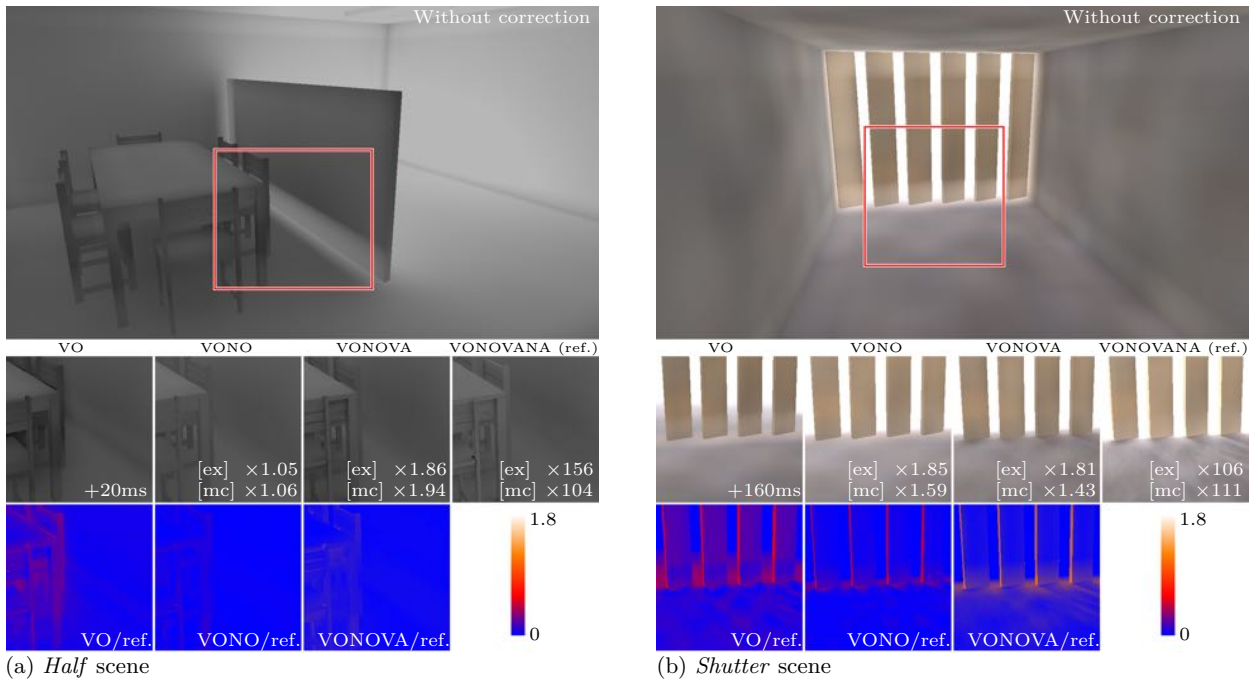


Fig. 6: Density estimation with direct visualization of photon maps, for *Half* and *Shutter* scenes. (VO) Photon validation from observer side, with additional computation time; (NO) Kernel normalization from observer side, including visibility, with increase time factor for exact visibility [ex] and Monte Carlo approximation [mc]; (VA) Photon validation from ancestors side, with increase time factor for exact visibility [ex] and Monte Carlo approximation [mc]; (NA) Kernel normalization including visibility from ancestors, with increase time factor for exact visibility [ex] and Monte Carlo approximation [mc]. The false color images correspond to the difference between the reference (VONOVANA) and the corresponding method, based on the HDR images.

ing to photon-mapping direct visualization, in order to illustrate the improvements associated with each correction. We do not provide comparisons with unbiased methods since the difference images would essentially exhibit proximity bias.

Observer domain - Photon validation

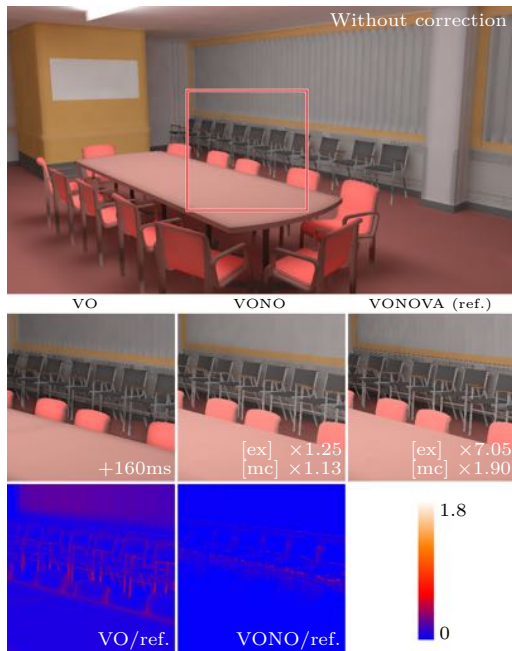
The observer domain is the region of the initial kernel domain (around the estimation point \mathbf{x}), actually visible from the observer. Firstly, photons located in the initial kernel domain should thus be handled only if they are visible from the observer (photon validation). This process aims at selecting photons in the kernel domain, only if they are visible from the observer. It allows to remove inconsistent light leaking, while reducing proximity bias. Since all the visibility requests are consistent (the ray origin is always the observer), they can be efficiently performed using either point-to-point or point-to-surface surface visibility requests. Furthermore, photon-observer visibility can be managed in a preprocessing step for all the image, so that for a given photon validation is performed only once. This process thus results in a very low additional computation cost.

In Figures 6 and 7, columns VO illustrate the effects of photon validation from the observer side. These images do not include direct lighting, so as to emphasize the differences. In these examples, light leaking is removed thanks to this validation, particularly under the wall of Half scene.

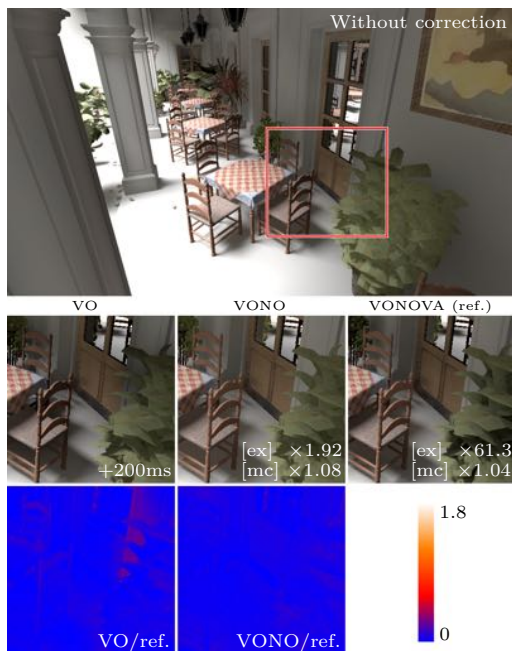
With photon validation, density estimation tends to produce darker surfaces because photons located behind observed surfaces in the estimation domain are not used. The resulting domain has to be normalized, as explained in the following paragraph.

Observer domain - Normalization

The kernel domain should be clipped according to the corresponding observer visibility, in order to determine the normalization factor $\mathbb{K}_3(\mathbf{x}, \mathcal{D})$, defined in Equation 6. It relies on the surfaces of the domain $\mathcal{D}(\mathbf{y}, \mathbf{x})$, corresponding to all the pieces of surface visible from the observer in the initial kernel domain. Each of them is used as a subdomain of the global integration domain for the normalization factor $\mathbb{K}_3(\mathbf{x}, \mathcal{D})$. For both exact calculation and Monte Carlo estimation of the normalization factor, computation time only depends on the



(a) Conference scene



(b) San Miguel scene

Fig. 7: Density estimation with direct visualization of photon maps, for *Conference* and *San Miguel* scenes. The false color images correspond to the difference between the reference (VONOVA) and the corresponding method, based on the HDR images.

domain geometry. It is thus independent of the number of photons.

In Figures 6 and 7, columns NO illustrate the correction with photon validation (observer side) with and without normalized kernel domain. Computation time increase is provided on each image for both exact visibility and Monte Carlo integration. The normalization process based on the observer domain improves notably the density estimation quality. Note that the normalization process associated with a 3D kernel on an unbounded domain (such as the center of a flat surface without occultation) slightly changes the image since the surface does not cover the whole kernel domain. In this very specific case, 2D kernels would not introduce any difference.

Photon domain - Photon validation

Photons should be employed in the kernel domain only if their ancestors are visible from the estimation point. Also, the kernel domain should contain only surfaces visible from all the validated photon ancestors.

Point-to-point or point-to-surface visibility requests are more costly in this case due to the spreading of photon ancestors in the scene: Visibility computations do not benefit from memory coherent queries. The computation time corresponding to this process also depends on the number of emitted photons.

In Figures 6 and 7, columns VA illustrate the correction improvement and additional computation time for photon validation from their ancestor side. Photons are used for density estimation only when their ancestors are visible from the estimation point. This process has a visual impact essentially for light leaking, and in dark regions where occlusions prevent from indirect lighting.

Photon domain - Normalization

From each photon ancestor, the visible pieces of surface in the kernel domain are identified, each of them corresponding to one normalization factor. Table 1 provides the number of computed normalization factors for each scene. This process is time consuming due to the calculation of a high number of normalization factors, and visibility requests are not spatially coherent.

Figure 6 illustrates the resulting difference on *Shutter* and *Half* scenes only, showing that: (i) This process dramatically increases computation time, and (ii) the corresponding improvements brought by this process remain low. Even though the theoretical aspects remain interesting from our point of view, we believe that for this part the gain is not worth the cost.

Discussion

Some important observations can be made from these

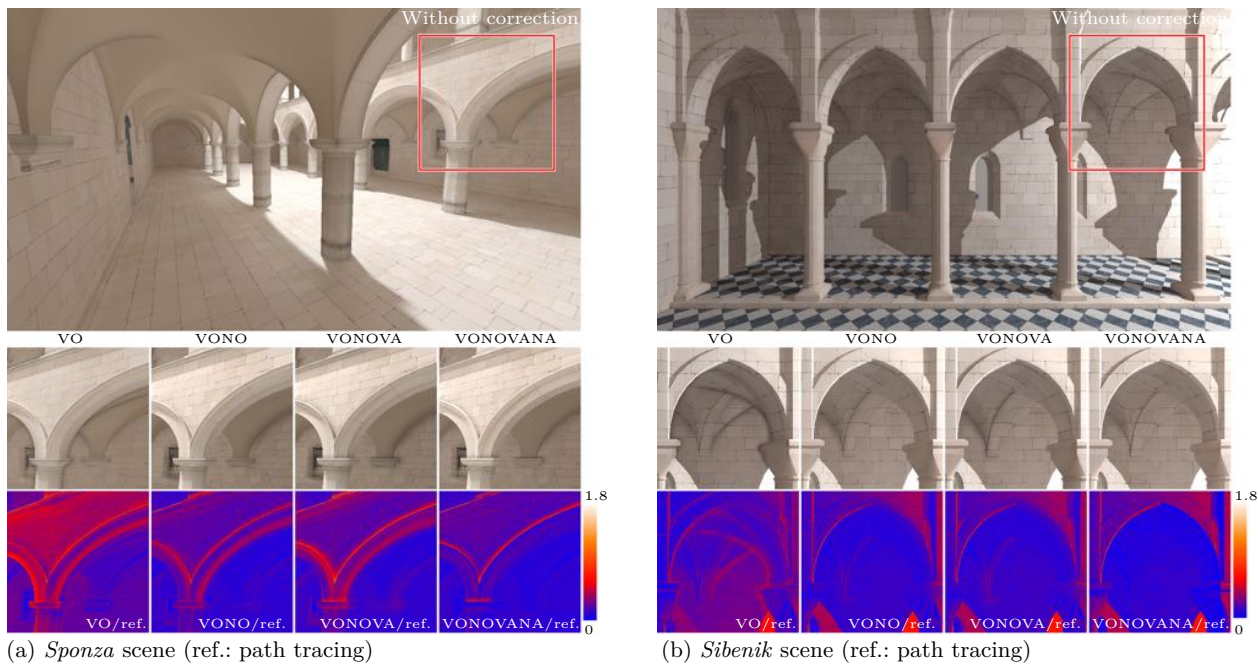


Fig. 8: Progressive photon mapping, with *Sponza* and *Sibenik* scenes. Validation and normalization is illustrated, and compared with the reference image. The improvements are visible even with progressive photon mapping, which proves that visibility actually plays an important role in the whole normalization process. [The false color images correspond to image difference with path tracing.](#)

Scene	#Factors
Half	2 880 410 255
Shutter	223 038 090
San Miguel	1 489 318 778
Conference	647 165 668
Sponza	349 747 806
Sibenik	308 785 572
Pool	276 086 077

Table 1: Number of factors corresponding to the ancestor domain normalization, used for each scene and its associated image.

experiments. Firstly, normalization factor computations corresponding to photon ancestors do not benefit from memory coherent queries, drastically increasing computing time. However, the visual impact corresponding to this improvement remains very limited. Conversely, the observer normalization part is faster because it relies on the same viewpoint, and thus benefits from memory coherent queries. Despite our expectations, even in this latter case, exact point-to-surface visibility estimation does not actually reduce computation time. Generally speaking, the best trade-off can be found with Monte Carlo integration and point-to-point visibility.

5.2 Progressive Photon Mapping

Our kernel normalization process has also been applied to the progressive photon mapping (PPM) method proposed by Knaus and Zwicker [6,16]. The integration of our method is straightforward in any existing PPM rendering framework. In the following figures, 100K photons are shot per iteration, and 8192 iterations are performed for the following images.

As illustrated in Figure 8, bias reduction is visible for each part of the global kernel normalization process, even after 8192 iterations. For these images, photon-mapping is employed for estimating the indirect lighting only. The false color images clearly show where the differences are higher. [The reference image used for comparisons with PPM corresponds to path tracing.](#)

Progressive photon mapping approaches progressively shrink the kernel domain, statistically reducing the number of photons used for the density estimation. Consequently, the number of visibility tests accordingly decreases for each iteration, and our method deserves to be employed.

Photon validation can also be applied with direct lighting (Figure 9). This process clearly shows that sharp shadows are produced, contrary to usual density estimation, they are clearly visible for instance near the pillars. One advantage of this approach is that direct

and indirect lighting can be performed homogeneously with photons only.

5.3 Influence on occluded caustics

The impact of visibility on occluded caustics is clearly visible in the case of validation from the ancestor side, producing sharper shadows. Figure 10 illustrates a pool with a grid that occludes caustics on the ground. The caustics are handled properly due to the density estimation process, with the removal of only some photons that should not contribute. This process also handles natively specular-diffuse-specular paths (the caustics are noticeable in the mirror). Compared to Qin et al. [23] (which rather relies on vertex merging), progressive photon mapping density estimation process ensures that all the contributing photons are considered.

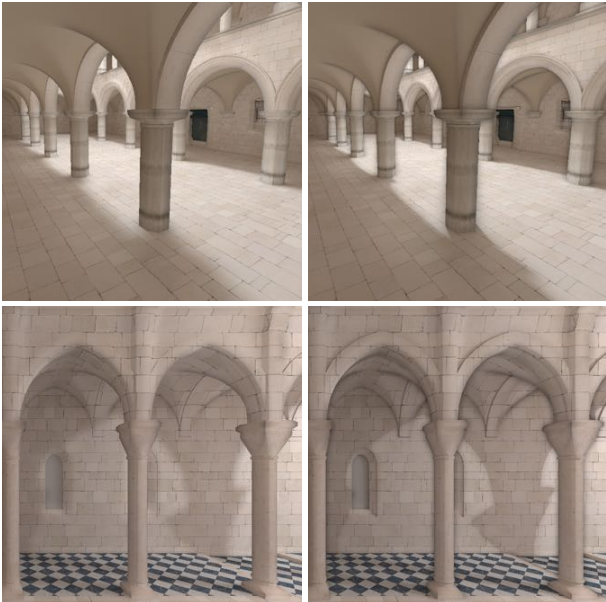


Fig. 9: Illustration of the photon validation process, using density estimation with direct plus indirect illumination (top) Sponza, (bottom) Sibenik, (left) no correction (right) with photon validation from observer and ancestor sides. Shadows are much sharper with our method.

5.4 Discussion and Limitations

The results obtained with our approach (boundary bias and occlusion bias reduction) show that visibility is a very important factor for density estimation with photon mapping. Unfortunately, including visibility in den-

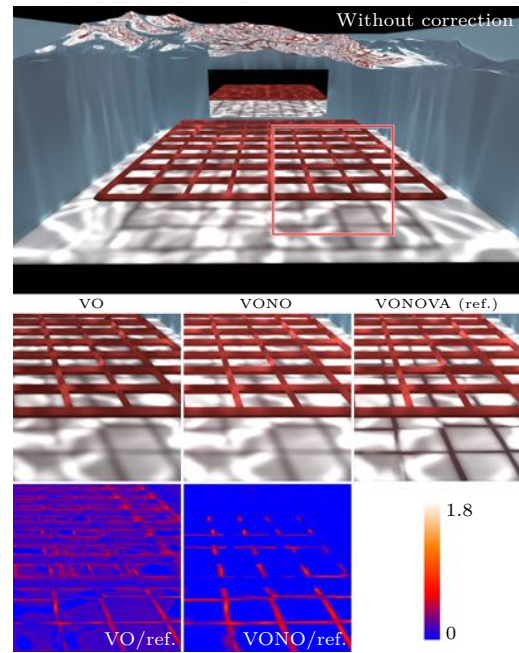


Fig. 10: Pool scene, illustrating the benefit of our density estimation process in the case of occluded caustics. In this case, the validation process from the ancestor side has a very important impact on the visual aspect of the rendered shadows.

sity estimation also requires additional computations, which notably influence performances.

Validation from observer and ancestor sides may have an important influence on the resulting images, depending on the geometric configuration. As shown in the above results, both of them are mandatory for removing light leaking. Normalization is required for a consistent density estimation process, but the visual impact from the observer side is much more visible than from the ancestor side. The latter is very time-consuming, and we think it could be disabled in practice for most scenes. The most interesting impact on quality is provided by the two validations and by the normalization from the observer side.

The most important limitation of this work concerns the time required for visibility computations. Photon validation from the observer side can be precomputed based on primary rays (even for anti-aliasing and depth of field). However, normalization always relies on an estimation of the domain, that varies for each estimated point and for each PPM iteration.

6 Conclusion and Future Work

This paper presents a complete study of density estimation for photon mapping approaches. We show that

visibility has a significant impact on photon mapping methods using density estimation. Bias can be significantly reduced, provided that kernel and normalization methods are carefully defined. Our implementation choices make boundary and occlusion biases consistent, while removing topological bias.

Our results exhibit the benefit of taking visibility into account, not only with direct photon-maps visualization with density estimation, but also with progressive photon mapping.

References

1. Bekaert, P., Slusallek, P., Cools, R., Havran, V., Seidel, H.P.: A custom designed density estimation method for light transport. Tech. Rep. MPI-I-2003-4-004, Max-Planck-Institut für Informatik, Saarbrücken. (2003)
2. Chen, L.H., Tsai, T.C., Chen, Y.S.: Grouped photon mapping. *The Visual Computer* **26**, 217–226 (2010)
3. Dammertz, H., Hanika, J., Keller, A.: Shallow Bounding Volume Hierarchies for Fast SIMD Ray Tracing of Incoherent Rays. *Computer Graphics Forum* **27**(4), 1225–1233 (2008)
4. Dutre, P., Bala, K., Bekaert, P., Shirley, P.: *Advanced Global Illumination*. AK Peters Ltd (2006)
5. Georgiev, I., Krivánek, J., Davidovič, T., Slusallek, P.: Light transport simulation with vertex connection and merging. *ACM TOG, Proceedings of SIGGRAPH Asia* **31**(6) (2012)
6. Hachisuka, T., Ogaki, S., Jensen, H.W.: Progressive photon mapping. *ACM TOG, Proceedings of ACM SIGGRAPH Asia* **27**(5) (2008)
7. Hachisuka, T., Pantaleoni, J., Jensen, H.W.: A path space extension for robust light transport simulation. *ACM TOG, Proceedings of SIGGRAPH Asia* **31**(6) (2012)
8. Haumont, D., Mäkinen, O., Nirenstein, S.: A Low Dimensional Framework for Exact Polygon-to-Polygon Occlusion Queries. In: *Eurographics Symposium on Rendering Techniques*, pp. 211–222 (2005)
9. Havran, V., Bittner, J., Herzog, R., Seidel, H.P.: Ray Maps for Global Illumination. In: *Eurographics Symposium on Rendering Techniques*, pp. 43–54 (2005)
10. Herzog, R.: Advanced density estimation techniques for global illumination. Master’s thesis, Universität des Saarlandes (MPI Informatik) (2005)
11. Izenman, A.: *Modern Multivariate Statistical Techniques, Regression, Classification, and Manifold Learning*. Springer New-York (2008)
12. Jensen, H.W.: Global Illumination using Photon Maps. In: *Eurographics Workshop on Rendering Techniques*, pp. 21–30 (1996)
13. Jensen, H.W.: Realistic image synthesis using photon mapping. A. K. Peters, Ltd. (2001)
14. Jones, M.C.: Simple boundary correction for kernel density estimation. *Statistics and Computing* **3**(3), 135–146 (1993)
15. Kajiya, J.T.: The rendering equation. In: *ACM SIGGRAPH* (1986)
16. Knaus, C., Zwicker, M.: Progressive photon mapping: A probabilistic approach. *ACM TOG* **30**(3) (2011)
17. Lastra, M., Ureña, C., Revelles, J., Montes, R.: A Particle-Path based Method for Monte-Carlo Density Estimation. In: *Eurographics Workshop on Rendering Techniques* (2002)
18. Lavignotte, F., Paulin, M.: A New Approach of Density Estimation for Global Illumination. In: *WSCG, Plzen, Czech Republic*, pp. 263–273 (2002)
19. Lavignotte, F., Paulin, M.: Scalable photon splatting for global illumination. In: *GRAPHITE ’03*. ACM, New York, NY, USA (2003)
20. Mora, F., Aveneau, L., Apostu, O.L., Ghazanfarpour, D.: Lazy visibility evaluation for exact soft shadows. *Computer Graphics Forum* **31**(1), 132–145 (2012)
21. Nirenstein, S., Blake, E., Gain, J.: Exact from-region visibility culling. In: *Eurographics Workshop on Rendering Techniques* (2002)
22. Pharr, M., Humphreys, G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2010)
23. Qin, H., Sun, X., Hou, Q., Guo, B., Zhou, K.: Unbiased photon gathering for light transport simulation. *ACM TOG, Proceedings of SIGGRAPH* **34**(6) (2015)
24. Schregle, R.: Bias Compensation for Photon Maps. *Computer Graphics Forum* **22**, 729–742 (2003)
25. Silverman, B.W.: *Density estimation for statistics and data analysis*. Chapman and Hall (1998)
26. Teller, S., Hanrahan, P.: Global visibility algorithms for illumination computations. In: *ACM SIGGRAPH* (1993)
27. Tobler, R.F., Maierhofer, S.: Improved Illumination Estimation for Photon Maps in Architectural Scenes. *WSCG* pp. 257–261 (2006)
28. Veach, E.: Robust Monte-Carlo methods for light transport simulation. Ph.D. thesis, Stanford University (1997)
29. Wald, I., Woop, S., Benthin, C., Johnson, G.S., Ernst, M.: Embree: a kernel framework for efficient cpu ray tracing. *ACM TOG, Proceedings of ACM SIGGRAPH* **33**(4) (2014)
30. Wand, M.P., Jones, M.C.: *Kernel Smoothing, Monographs on Statistics and Applied Probability*, vol. 60. Chapman and Hall, London (1995)

Appendices

A Insight into density estimation

This section discusses several aspects of density estimation, and the importance of consistency with kernel density estimators [11]. In the univariate case, given n independent identically distributed observations x_1, x_2, \dots, x_n of a given function f , the kernel density estimator $\langle f_h(x) \rangle$ of $f(x)$ is:

$$\langle f_h(x) \rangle = \frac{1}{nh} \sum_{i=1}^n K_1 \left(\frac{x - x_i}{h} \right), \quad x \in \mathbb{R}, h > 0 \quad (10)$$

where K_1 is a 1D kernel function (with mean 0 and integral equal to 1), and h is the kernel bandwidth. In the following, we are mainly interested by kernels with a compact domain, i.e., kernels K_1 such that

$$K_1(x) = 0, \quad \forall x \notin [-1, +1].$$

The multivariate kernel density estimator of f is:

$$\langle f_{\mathbf{H}}(\mathbf{x}) \rangle = \frac{1}{n |\mathbf{H}|} \sum_{i=1}^n K_d(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)), \quad \mathbf{x} \in \mathbb{R}^d$$

(11)

where \mathbf{H} is a $(d \times d)$ nonsingular matrix generalizing the bandwidth, and K_d is a multivariate kernel function.

A popular representation for multivariate kernels consists in using product kernels, corresponding to product of the same univariate kernel function:

$$K_d(\mathbf{y}) = \prod_{j=1}^d K_1(y_j), \quad \forall \mathbf{y} = [y_1, \dots, y_d], \quad (12)$$

defined on a polyhedral domain.

Kernels are generally chosen with the following properties:

$$\forall \mathbf{y} \in \mathbb{R}^d, K_d(\mathbf{y}) \geq 0 \quad (\text{positive}), \quad (13)$$

$$\int_{\mathbb{R}^d} K_d(\mathbf{y}) \, d\mathbf{y} = 1 \quad (\text{normalized}), \quad (14)$$

$$\int_{\mathbb{R}^d} \mathbf{y} K_d(\mathbf{y}) \, d\mathbf{y} = 0 \quad (\text{symmetric}), \quad (15)$$

$$\mu_2(K_d) \mathbf{I}_d = \int_{\mathbb{R}^d} \mathbf{y} \mathbf{y}^T K_d(\mathbf{y}) \, d\mathbf{y} \quad (\text{bounded}), \quad (16)$$

where $\mu_2(K_d) = \int y_i^2 K_d(\mathbf{y}) \, d\mathbf{y}$ is independent of $i \in [1 \dots d]$, $\mu_2(K_d) < \infty$, and \mathbf{I}_d is the $d \times d$ identity matrix.

Equations [13](#) and [14](#) ensure the kernel can also be seen as a normalized probability density function; Equation [15](#) states that K_d is a symmetric function, which is generally a good property for a convolution, except near support boundaries; Equation [16](#) is only useful to express the bias as a function over a finite value $\mu_2(K_d)$.

Density estimation is intrinsically biased. The bias is defined as:

$$b(\langle f_{\mathbf{H}}(\mathbf{x}) \rangle) = E\{\langle f_{\mathbf{H}}(\mathbf{x}) \rangle\} - f(\mathbf{x}),$$

where $E\{\langle f_{\mathbf{H}}(\mathbf{x}) \rangle\}$ is the expected value of function $\langle f_{\mathbf{H}}(\mathbf{x}) \rangle$. When the estimator is biased, the expected value does not tend to the actual function:

$$b(\langle f_{\mathbf{H}}(\mathbf{x}) \rangle) \neq 0 \quad \Rightarrow \quad E\{\langle f_{\mathbf{H}}(\mathbf{x}) \rangle\} \neq f(\mathbf{x}).$$

In practice, users are mostly interested in density estimators that are consistent: $\forall \mathbf{x} \in \mathbb{R}^d, b(\langle f_{\mathbf{H}}(\mathbf{x}) \rangle) \rightarrow 0$, when all values of \mathbf{H} tend to 0.

A good trade-off between bias and variance is difficult to find since increasing the bandwidth h allows to reduce variance but increases bias; conversely decreasing h reduces bias but variance increases.

B Boundary bias in statistics

Boundary bias is well-known in statistics. When the estimation domain is bounded, bias increases at the boundary, leading to inconsistent density estimation. This can be explained using a Taylor expansion (see [30](#) for the multivariate case, exhibiting the same behavior). In the univariate case, a kernel defined on a compact domain $[-1, +1]$ leads to the following bias expression:

$$b(\langle f_h(x) \rangle) = f(x) \int_{-h}^{+h} K_1\left(\frac{z}{h}\right) \, dz - f(x) \quad (17)$$

$$- h f'(x) \int_{-h}^{+h} z K_1\left(\frac{z}{h}\right) \, dz \quad (18)$$

$$+ \frac{h^2}{2} f''(x) \int_{-h}^{+h} z^2 K_1\left(\frac{z}{h}\right) \, dz + o(h^2). \quad (19)$$

Using Equation [14](#) cancels the first two terms, Equation [15](#) removes the third one, and the condition corresponding to Equation [16](#) allows to bound the last term and to conclude that bias is proportional to h^2 . Finally, when h decreases down to 0, the bias also tends to 0: The estimator is consistent.

Let us now consider a function f defined over $[0, +\infty]$. Density estimation is performed using Equation [10](#) for $0 < x < h$, and the previous bias approximation leads to:

$$\begin{aligned} b(\langle f_h(x) \rangle) &= f(x) \int_0^{+h} K_1\left(\frac{z}{h}\right) \, dz - f(x) \\ &\quad - h f'(x) \int_0^{+h} z K_1\left(\frac{z}{h}\right) \, dz \\ &\quad + \frac{h^2}{2} f''(x) \int_0^{+h} z^2 K_1\left(\frac{z}{h}\right) \, dz + o(h^2). \end{aligned}$$

In this case, Equations [13](#), [14](#), [15](#), [16](#) do not ensure a consistent estimator anymore. Hence, the bias becomes a function of $f(x)$, which cannot be decreased with more observations.

In order to ensure a consistent estimation, Jones proposes to normalize the kernel at each estimation point [14](#); in other words, a *normalization factor* is calculated for each estimated point x :

$$\mathbb{K}_1(x) = \int_{-h}^{+x} K_1\left(\frac{x-z}{h}\right) \, dz, \quad (20)$$

and density estimation becomes:

$$\langle f_h(x) \rangle = \frac{1}{nh \mathbb{K}_1(x)} \sum_{i=1}^n K_1\left(\frac{x-x_i}{h}\right). \quad (21)$$

Consequently, density bias is in $\mathcal{O}(h)$, and thus consistent.

The generalization in any dimension d of this normalization process requires to determine the normalization factor $\mathbb{K}_d(\mathbf{x})$ resulting from the kernel integration over a domain $\mathcal{D} = \{\mathcal{D}_1 \times \mathcal{D}_2 \times \dots \times \mathcal{D}_d\}$:

$$\begin{aligned}\mathbb{K}_d(\mathbf{x}) &= \int_{\mathcal{D}} K_d(u_1, \dots, u_d) du_1 \dots du_d \\ &= \int_{\mathcal{D}_d} \dots \int_{\mathcal{D}_1} K_1(u_1) du_1 \dots du_d,\end{aligned}\tag{22}$$

where each sub-domain \mathcal{D}_i is orthogonal to the others, which makes it separable and thus easier to estimate.