

# Making Imperfect Shadow Maps View-Adaptive: High-Quality Global Illumination in Large Dynamic Scenes

Tobias Ritschel<sup>1,2</sup> Elmar Eisemann<sup>2</sup> Inwoo Ha<sup>3</sup> James D. K. Kim<sup>3</sup> Hans-Peter Seidel<sup>1</sup>

<sup>1</sup>MPI Informatik

<sup>2</sup>Télécom ParisTech / CNRS-LTCl

<sup>3</sup>Samsung Advanced Institute of Technology



**Figure 1:** Interactive global illumination rendered by our algorithm for a complex architectural scene (1 M polygons), containing several animated pieces of cloth, animated light and a dynamic camera (13 fps at 1600×800, Nvidia GTX 480).

## Abstract

We propose an algorithm to compute interactive indirect illumination in dynamic scenes containing millions of triangles. It makes use of virtual point lights (VPL) to compute bounced illumination and a point-based scene representation to query indirect visibility, similar to Imperfect Shadow Maps (ISM). To ensure a high fidelity of indirect light and shadows, our solution is made view-adaptive by means of two orthogonal improvements: First, the VPL distribution is chosen to provide more detail i. e. more dense VPL sampling, where these contribute most to the current view. Second, the scene representation for indirect visibility is adapted to ensure geometric detail where it affects indirect shadows in the current view.

## 1. Introduction

Indirect illumination is an important element of realistic image synthesis, but its computation is usually costly. Only recently, interactive global illumination techniques have emerged, but, unfortunately, most of these algorithms are only suited for scenes of small to moderate extent.

To address larger scenes, the present paper contributes two view-adaptive extensions to an Instant radiosity-type [Kel97] interactive global illumination technique based on Reflective [DS05] (RSMs) and Imperfect Shadow Maps [RGK\*08] (ISM). The first extension improves the indirect lighting quality by identifying sources of indirect light that strongly contribute to the final image and by culling unnecessary ones. The second extension, improves the accuracy of indirect shadows by adapting the occluder representation to the viewpoint. Via these ameliorations, the technique achieves high-quality

results, at interactive framerates, even for larger dynamic scenes (Fig. 1).

This paper is structured as follows: We describe previous work in Section 2 before reviewing the background of RSMs and ISMs in Section 3. Both extension are detailed in Section 4 and 5. Implementation details are exposed in Section 6. After presenting (Sec. 7) and discussing (Sec. 8) our results, we conclude in Section 9.

## 2. Previous work

State-of-the-art reference solutions usually rely on path tracing [Kaj86], photon mapping [Jen01], or ray tracing [WKB\*02], but these methods are costly because visibility is sampled accurately by testing rays against the entire scene geometry. Via precomputation [SKS02], rendering can be accelerated at run-time, but this implies several restrictions, such as static geometry and needs much storage space.

One way to improve performance is to simplify the global illumination computation [RPV93,CB04,TL04], or to perceptually approximate visibility queries [YCK\*09]. Hierarchical simplification [SD95] decrease computation time, but the required precomputation steps prevent a use in dynamic scenes.

Visibility is the key point in computing global illumination, which is particularly evident when looking at the high performance that is reached if visibility is simply ignored [DS05]. Evaluating visibility is difficult because the related rays are usually unorganized, whereas current graphics hardware is optimized for coherent rasterization. This observation was exploited by instant radiosity [Kel97]. Instead of assuming a continuous light repartition, virtual point lights (VPLs) are used to simulate indirect illumination.

Only recently, real-time global illumination became feasible. A reformulation of visibility during the light transfer leads to interactive framerates in scenes containing a few thousand triangles [DSDD07,DKTS07]. These solutions do not scale easily to more complex models. Recently introduced, light propagation volumes [KD10] compute approximate global illumination rapidly using volumetric light diffusion and volumetric visibility built from image-based blockers.

Laine et al. [LSK\*07] pointed out that, in static scenes, many VPLs can be reused and, consequently, their visibility computations. The result is of very high quality and the solution efficient, but the method does not extend to dynamic scenes. Segovia et al. [SIP06] use ray-tracing-based bidirectional importance sampling to discover important VPLs. Our goal is similar, but in contrast, we avoid ray-tracing and manage large and dynamic scenes (Fig. 1).

As pointed out by Arikan et al. [AFO05], nearby visibility events are of high importance for a better surface perception. Unfortunately, these are expensive to compute and Arikan et al. opted for a very coarse approximation in order to improve performance. ISM and also Microrendering [REG\*09] share this problem because a point-based approximation cannot be sufficiently precise for nearby geometry. In our approach, we adapt the blocker precision and significantly increase realism by capturing finer details.

A very coarse scene approximation is a purely image-based representation [RGS09,NSW09]. Such approaches achieve very high performance and capture some nearby visibility events, but they have a strong systematic bias. For high-quality rendering, the entire scene has to be involved, not just the information from a single image.

Walter et al. [WFA\*05] introduced lightcuts to cluster VPLs hierarchically for each pixel. The visibility for a group is then determined by simply assuming a point source per VPL group. The algorithm ensures accurate results by bounding the resulting error, but a per pixel grouping does not allow us to reduce the number of shadow maps that need to be created. Such an idea was pursued in [DGR\*09], but temporal coherence remained a major challenge. The same holds

for [HPB07] where light groups are established by sampling light-receiver relationships. Only if the entire sequence is known, temporally coherent results are possible [HVAPB08]. Further, both approaches exploit graphics hardware for the sampling process by rendering many (perfect) shadow maps. Unfortunately, rendering complex scenes is expensive and, hence, their computation times approach the order of minutes per image. Our strategy enables high-quality results in fully dynamic scenes in the order of milliseconds.

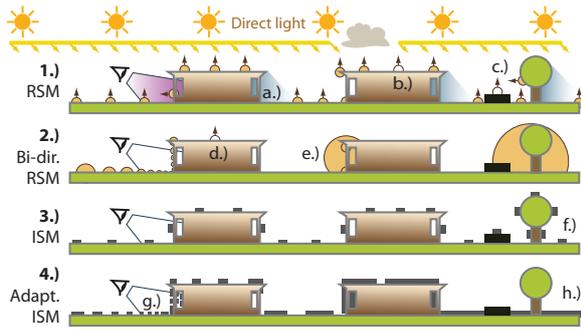
### 3. Background

The approach to interactive global illumination taken in this work is based on three key techniques: virtual point lights [Kel97], reflective shadow maps [DS05] and imperfect shadow maps [RGK\*08]. In this section, we briefly review those techniques (Fig. 2). The section ends with a summary of the contributions made in this paper.

**Virtual Point Lights (VPL)** [Kel97] compute global illumination by emitting secondary light sources from the initial light source into the scene. Illuminating the scene with these secondary sources simulates one bounce of indirect illumination. In interactive applications, the final rendering is computed by making use of deferred shading [ST90]. For each pixel, the illumination of the underlying scene point  $P$  is evaluated by combining direct and indirect illumination from all VPLs. If a VPL is not visible from  $P$ , an *indirect shadow* occurs and the VPL should not illuminate  $P$ . To resolve indirect shadows, the visibility relationships between all VPLs and all screen pixels have to be resolved, which is costly.

**Reflective Shadow Maps (RSMs)** [DS05] are a technique to efficiently generate VPLs by rendering the scene from the light's point of view. The texels of the resulting image can be used to define the position, normal and intensity of the VPLs. In practice, not all texels become VPLs, but only a subset [DS05, RGK\*08]. Picking the right VPLs for a complex scene is difficult. Some existing techniques [RGK\*08] use importance sampling to select the VPLs from the reflective shadow map that contribute much to the scene. To this end the VPLs are more dense, where outgoing irradiance (the product of direct light and reflectance) is high, which leads to more VPLs on strongly lit surfaces of high albedo. However, such distributions do not correspond to the actual impact of a VPL on the final image: If the scene size is increased, even strong VPLs might contribute only little to the final image.

**Imperfect Shadow Maps (ISM)** [RGK\*08] are one particular way to efficiently query secondary visibility. ISMs are low-resolution (e. g.  $32 \times 32$ ) shadow maps that are computed for each VPL to resolve visibility. Although it is possible to render the scene geometry several times i. e. once for every VPL, it proves very costly: for every VPL, the entire scene geometry needs to be transformed and every polygon needs to be rasterized. Instead, ISMs replace the scene geometry



**Figure 2:** Conceptual overview of classic Reflective Shadow Maps, our bi-directional extension, Imperfect Shadow Maps as well as our adaptive extension. 1: Direct light (e. g. a sun) illuminates a scene and places VPLs (small circles) on all surfaces not in direct shadow (a). Important sampling places less or no VPLs if no direct light is present e. g. a cloud shadow (b) or on surfaces with low albedo (c). Note, that this does not account for the final image content, resp. current view (eye, left). 2: Our extension to RSM places more VPLs where the direct light contributes more to the indirect light, effectively resulting in bigger VPLs. A VPL that cannot contribute is (d) because of its normal whereas larger VPLs are used for far away direct light such as (e). 3: Imperfect Shadow Maps sample the scene surfaces equally, even for far-away occluders like (f) that do not contribute much to the current view. 4: Our extension places many fine points closer to the current view (g), and fewer large points far away (h).

to be drawn into each VPL’s shadow map by a regular point-based representation computed in a preprocess. To produce the shadow maps, a vertex shader scatters all these points into their corresponding VPL shadow map. To make this process simple, all shadow maps are conveniently tiled in a large texture. Each scene point maps to a single VPL shadow map, hence, one draw call is sufficient to fill all VPL shadow maps at once. Hereby, for each VPL’s shadow map, only a few thousand points, instead of possibly millions of triangles needs to be drawn.

Nevertheless, the point-based representation can result in *imperfect* shadow maps that exhibit holes. These holes are filled in a push-pull postprocessing [MKC07] which diffuses surrounding depth values. Although this step is approximate, such imperfections, as well as the low resolution, are usually not noticeable when a large number of VPLs are evaluated per pixel [YCK\*09]. Nonetheless, this process only works for smaller scenes, where the regular point-based scene representation is still acceptable, for larger extents the number of points to capture the geometry quickly becomes prohibitive.

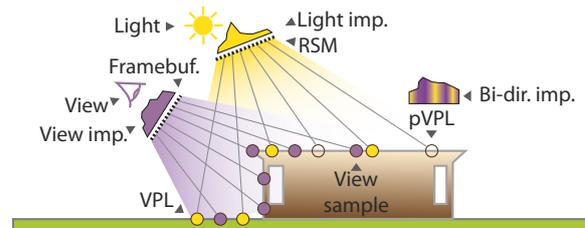
**Contributions** To summarize, two main shortcomings exist for ISMs: First, many VPLs are needed, but they are selected

without considering the contribution on the final image, leading to the evaluation of many unnecessary candidates. Second, a regular point-based geometry representation can be too coarse to avoid artifacts in large and complex scenes. Our work addresses both issues.

We propose to create VPLs according to a *bi-directional* estimate of their contribution to the final image (Sec. 4). Although presented in the context of ISMs, the idea can be used for other VPL-based approaches as well. To tackle the second problem, ISMs are created using an *adaptive* point-based geometry representation (Sec. 5). Because this representation is updated in each frame, it can efficiently adapt blocker precision to the current viewpoint and, hereby, better handle detailed indirect shadows in complex scenes than previous ISM approaches.

#### 4. Bidirectional Reflective Shadow Maps

To create VPLs, we propose an approach similar in simplicity and efficiency to [DS05] and with the bi-directional ability to focus on relevant VPLs [SIP06], but without the need to resort to costly ray tracing.



**Figure 3:** Bidirectional Reflective Shadow Maps combine importance of direct light and its contribution to the framebuffer.

First, the scene is rasterized from the current view into a framebuffer (“View” and “Framebuffer” in Fig. 3) as well as from the light’s point of view into a reflective cube shadow map (“Light” and “RSM” in Fig. 3). We store position, normal and reflectance in each pixel.

RSM texels represent *potential* VPLs (pVPL, empty circle in Fig. 3). Every pVPL could be transformed directly into an actual VPL, but for efficiency, it is a common to select only a subset [DS05] (typically we rely on 1 000) as actual VPLs (full yellow circles in Fig. 3). A uniform selection is possible, but then the resulting VPLs can be very poor making a large amount of samples necessary to achieve good quality. Consequently, performance slows down because each selected VPL also needs to be evaluated - including computation of a shadow map and visibility tests for all *view samples* (points in space corresponding to pixels in the current view). A better way to choose VPLs is importance sampling based on the outgoing irradiance [DS05, R GK\*08] (“Light imp.” in Fig. 3). In practice, this means storing the irradiance in each RSM texel and using it as a probability distribution to select VPLs

among the pVPLs. While this solution performs better than uniform sampling, it is far from optimal. VPLs might send out much light, but never actually illuminate the framebuffer’s view samples (violet circles in Fig. 3).

Instead, we introduce a non-uniform VPL sampling that better estimates the impact on the view samples. To this end, we will not simply use the outgoing radiance, but define a different probability distribution that accounts for the current view (“Bi-dir. imp.” in Fig. 3).

Please note, that when VPLs are selected non-uniformly, careful normalization needs to be taken into account to avoid bias. In practice, this means that one needs to divide the VPL radiance by the probability with which it was selected. For the special case of importance sampling, where the sampling follows the outgoing irradiance, it implies that all VPLs are normalized and should be equally bright. A division by zero cannot occur because such VPLs are never selected by definition.

The question remains what probability distribution should be chosen to select the VPLs. An optimal choice would be to use the influence of each pVPL on all view samples, but this is exactly the result we seek to approximate because it is too costly to evaluate.

In order to make our approach practical, we introduce two simplifications. First, we rely on a stochastic solution: each pVPL only evaluates its impact on a few randomly-chosen view samples (in practice,  $\approx 0.1\%$ ). Second, we neglect visibility when evaluating the illumination contribution of each pVPL on the visible scene. This choice facilitates the computation substantially because the corresponding value can be derived directly from light and view sample position alone in closed form and in parallel without the need for ray-scene intersections. Each pVPL’s average contribution is then stored in a so-called *Bidirectional Reflective Shadow Map* (BRSM) and VPLs are selected according to this bidirectional importance.

#### 4.1. Technical Details

While theoretically simple, the two main technical hurdles are the construction of the BRSM and the selection of VPLs according to it on the GPU.

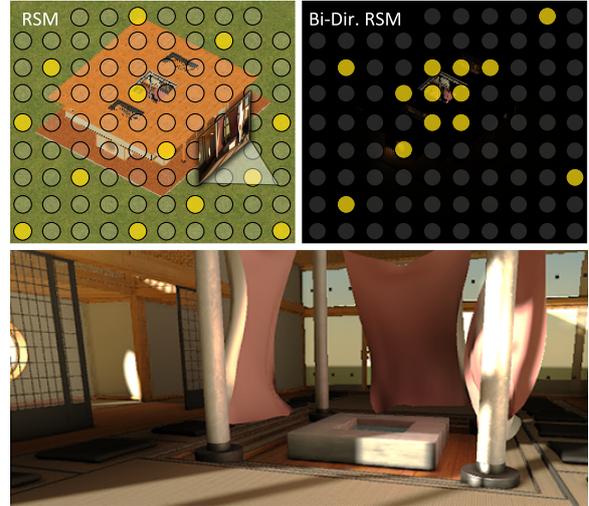
To construct the BRSM, we need a random view-sample selection. We rely on an approximate solution that starts with a regular grid, but jitters the lookup positions. Each pVPL uses different values from a random value texture to achieve a unique pattern, which gives good results in practice.

In order to choose VPLs according to the distribution defined by the BRSM, we will rely on several *cumulative density functions* (CDFs). To illustrate the process, let’s start with an example in 1D. Here, a CDF stores in texel  $i$ , the sum of all pixels  $j$  of the distribution  $D$  with  $j \leq i$ . Such a representation

can be computed efficiently in parallel, e. g. using parallel summed area tables [HSC\*05].

To sample according to the 1D distribution  $D$ , we make use of its CDF. We proceed as follows: Let’s assume we are given a budget of  $N$  samples, to find the position where we should place the  $i^{\text{th}}$  sample point, we perform a binary search in the CDF for the value  $i/N$ , resulting in a position  $k$  (in other words, we invert the CDF function  $C$  and find  $k := C^{-1}(i/N)$ ).

For the case of the BRSM, we deal with a 2D domain, but the process is almost similar. We first derive, in parallel, a CDF for each texture column  $C_y[i]$  of the BRSM. Second, we compute a single CDF  $C_x$  from the sums of all values in each column. Both computations can again be performed swiftly in a hierarchical manner. Based on these CDFs, we transform a uniform sampling into a sampling that respects the BRSM weights: For a uniform sample  $[x, y]^T \in [1, \dots, width] \times [1, \dots, height]$ , we first find a column position  $i := C_x^{-1}[x]$ , then a row position  $j := C_y[i]^{-1}[y]$ , to define the new sample location  $[i, j]^T$ . Figure 4 shows a comparison of our solution against competing methods.



**Figure 4:** *Bidirectional Reflective Shadow Maps* sample VPLs according to a distribution that favors VPLs contributing to the current view of the scene. E. g., RSM waste many VPLs on the roof, which have no effect on the final image.

In Monte Carol integration, samples have to be divided by the probability of selecting them. In our case, such as in all applications of importance sampling, the probability is not constant and therefore the contribution of each VPL is divided by the probability by which it has been chosen. In other words: VPLs that are dense in an area because it contributes much to the final image become weaker. Depending on the randomly selected view samples, this probability changes in each frame. Nevertheless, the probability to choose a VPL over several iterations converges towards its average impact on all view samples.

#### 4.2. Guided View-Sample Selection for BRSM

So far, each pVPL randomly chooses its view sample set. In this section, we will investigate how to improve the view-sample selection for a given pVPL in order to better estimate the impact of the pVPL on the final rendering. Our idea is to introduce a pVPL-dependent importance sampling to guide the view-sample selection to favor those view samples on which the pVPL has the strongest impact. For high efficiency, it is important that we estimate this impact without involving information about particular view samples. Unfortunately, the influence of a VPL on a view sample depends on the rendering equation and not a single term of this equation is independent of the actual view sample values, making this goal seemingly impossible.

Our idea to guide the sampling is to exploit the spatial arrangement of the framebuffer. Because view samples correspond to pixels, they are aligned on a grid in image space. Further, two view samples that are close in world space will also be close in the framebuffer. Although the opposite does not hold because even neighboring view samples can have very differing depth values, this observation can be used as an estimate. Basically, we project the VPL into the current view and then use the screen space distance as an approximation of the world space distance. In other words, a VPL will favor view samples in its vicinity in screen space. We chose a  $1/x^2$  distance falloff in accordance to the rendering equation. In practice, we clamp the  $1/x^2$  falloff to 1 in order to avoid the singularity at  $x = 0$ .

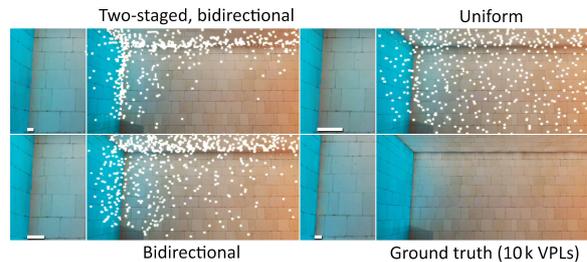
The projection of a VPL can result in a point at infinity, if it lies on the plane parallel to the view direction and passing through the center of projection. For such distant projections, the falloff function is mostly constant across the current view. Hence, the resulting sampling becomes uniform.

Please notice that the distance falloff is only used to guide the view-sample selection. It is not used to measure the impact of the VPL on the view samples. Dachsbacher and Stamminger [DS05] use a similar observation to gather light from a reflective shadow map, while we use it to reduce the variance of our estimate that controls which pVPLs in the BRSM are chosen to become VPLs.

Surprisingly, this coarse approximation is very successful in many challenging situations, e. g. corners. Here, VPL approaches are problematic because the VPLs can be very close to a receiving surface, giving rise to singularities. In such configurations, the discrete nature of VPL sampling can become visible and produce significant artifacts. Only through many VPLs can the illumination in corners be faithfully reproduced. Our heuristic to use the screen-space distance as an estimate will guide more VPLs exactly towards these locations. Usually, VPL contributions are simply clamped to an average VPL contribution when they are too close to a receiver [Kel97] to avoid *light blotches*. Our improved sampling strategy allows us to reduce this clamping cutoff significantly

because more VPLs with less intensity are used in areas of high importance.

The efficiency of our solution is illustrated in Figure 5 where the various approaches are compared to a reference rendering. Despite the scene’s simplicity, all competing solutions exhibit shortcomings. Without our guided view-sample selection, most VPLs would choose view samples from the main wall. Consequently, too many VPLs are placed on the side walls, hereby increasing the need to perform clamping. Using our guided search, the corner pVPLs on the main wall are likely to find view samples on the side walls. This improves the VPL creation process and our final solution closely resembles the reference rendering while being computationally much cheaper.



**Figure 5:** A comparison of various estimation strategies for the VPL sampling. A comparison with a ground truth shows how better sampling strategies deliver more faithful results. The white bar denotes the bias due to clamping in each solution: Smaller is better.

#### 5. Adaptive Imperfect Shadow Maps

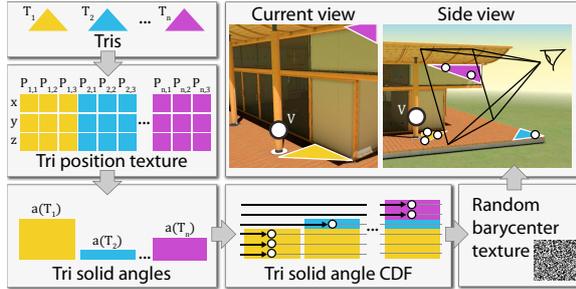
In this section, we show how to adaptively represent the blocker geometry using a point cloud of varying, i. e. adaptive density. This allows us to handle visibility in scenes larger than what was possible using ISMs (Fig. 9, c and d). The original ISM approach used a constant point density to represent all triangles in the scene. This led to increased holes, increased requirement for hole-filling and consequently decreased quality for larger scenes.

In adaptive ISMs, the blocker sampling is denser, i. e. is more detailed for geometry closer to and coarser, i. e. rough for geometry far away from the view samples (Fig. 2, 4). The reasoning is that occluders will tend to cast smoother indirect shadows on distant receivers which makes it unnecessary to maintain accuracy for distant blockers. On the other hand, small nearby occluders are likely to cast important shadows on nearby view samples and it is crucial to maintain their details.

An ideal and conservative blocker representation would consider the combination of all triangles and all view samples and then derive a suitable point-based blocker representation.

However, such an enumeration is not feasible in real time. In the following, we outline our efficient approach to perform an adaptive sampling.

**One-view Sample-based Adaptivity** For the moment, we will focus on a simple case and consider a single view sample  $V$ . Our goal is to adapt the point sampling of the scene geometry according to the distance to  $V$ . In other words, a triangle close to  $V$  should be sampled more aggressively than the same triangle farther away from  $V$ . To make this sampling efficient, we rely on a GPU implementation which is illustrated in Figure 6.



**Figure 6:** GPU implementation of Adaptive ISMs: For all tris, stored as vertex triples into a texture, their solid angle relative to a given view sample ( $V$ , blue circle) is computed. These solid angles are transformed into a cumulative density function (CDF) that is used to sample scene points. Here, the  $T_1$  receives more samples than  $T_2$  and  $T_3$  as it is closer to  $V$ .

We use a large three-channel *triangle texture* in which we store the  $(x,y,z)$ -coords of all vertices of all triangles. Based on this texture, we use a fragment shader to compute (in parallel over all triangles) a one-channel *triangle-importance texture*. This triangle importance texture stores the solid angle of the  $i^{\text{th}}$  triangle relative to  $V$  in texel  $i$ . Based on this triangle-importance texture, we produce, as previously for the BRSM, a cumulative density function (CDF) stored in a one-channel texture of the same size which will guide the point-sampling process of the scene geometry.

Let's assume we are given a budget of  $N$  points for the point-based geometry representation. The larger the importance of a triangle, the stronger its contribution to the CDF. Consequently, it is more likely that a sample point is placed on this particular triangle.

The CDF delivers for each sample point a suitable triangle  $k$ , but we still need to associate a point in  $k$ 's interior to derive the actual point-based scene representation. To this extent, we rely on a random value texture with  $N$  random variable pairs  $(r_1, r_2)$ , such that  $r_1 \in [0, 1]$  and  $r_1 + r_2 \leq 1$ . We interpret these random values as barycentric coordinates to define a position within triangle  $k$ , whose vertices are easily accessible via the triangle texture. The resulting point cloud is written into a VBO to be readily usable for the rendering step.

Fig. 7 shows an example of our approach. A light is pointed



**Figure 7:** Comparison between uniform and adaptive scene sampling. Only a random view sample was chosen from the framebuffer to guide the sampling.

at the wall towards the end of the corridor. The observer watches a statue at the other end. The statue is solely illuminated by indirect light. The insufficient occlusion precision due to a uniform geometry sampling makes indirect shadows completely vanish. Our adaptive sampling enhances the blocker resolution where it was needed. Even when involving just a single view sample (here, at the center of the image), the algorithm usually performs better than uniform sampling. Only in scenes with a very large extent, a single view sample can be too restrictive.

#### Stochastic Solution for Multi-View Sample Adaptivity

To better estimate good blockers for the current view, we could test the triangles' blocking contribution, i. e., its average projected area with respect to *all* view samples in the framebuffer. Unfortunately, computing this value for each triangle is prohibitively slow and it would not be possible to create the corresponding CDF in real time.

Our solution is to evaluate a different randomly-chosen set of view samples for each triangle. In this way, we only rely on a few view samples per triangle, but exploit the entire content of the framebuffer. This stochastic sampling remains fast and the final result approaches the wanted solution.

In practice, eight view samples per triangle works well and although it might seem appropriate to increase the number of samples with respect to the area of the triangle, we found this unnecessary. In particular, it is always possible to ensure a roughly constant triangle size in the scene, especially as we are interested in scenes that exhibit a high level of detail everywhere.

**Dynamic Scenes** Interestingly, there is a tradeoff between higher accuracy and temporal stability. If all blocker samples are changed from one frame to the next, flickering may occur. A simple solution to combat this problem is to rely on a lazy update scheme. Only a subset of all scene points are updated in each frame and we maintain the rest. In practice, this means that our solution converges to a perfectly adapted state

only when scene and viewpoint are static. During motion, the representation might not be optimal, but the sampling is usually more precise than for a uniform sampling. Further, a human observer perceives fewer details during motion, which makes this choice an excellent tradeoff.

Another advantageous side effect is that less sample points need to be recreated, reducing the cost of the sampling step. It is important to realize that, even when updating only a subset of the points, all points undergo animation transformations because they are expressed in barycentric coordinates. Consequently, points may lag in adaptivity, but participate correctly in the visibility sampling. In practice, we recompute  $1/8^{\text{th}}$  of the points in each frame.

Our scene sampling algorithm is very fast. It is compatible with dynamic scenes, as well as LOD mechanisms for which we generate the triangle texture in each frame (we describe an efficient implementation for this conversion step in Section 6). In contrast, previous approaches that make use of precomputed patterns have difficulties to address such scenarios.

## 6. Implementation Details

Our approach has been implemented using OpenGL 3.0, but can also be implemented on DX9 hardware. In this section, we will discuss some additional details, as well as improvements enhance quality and performance of our approach.

**Improved Shadow Map Projections** The original ISM used a paraboloid projection. We found that better results are achieved if the projection adapts to the radiance leaving the VPL. While such projections are challenging for triangular meshes [GHFP08], they are trivial for points. For Lambertian surfaces, it is best to use a parametrization that has a constant solid angle times the cosine with respect to the surface normal, sometimes called the *cos-sphere* parametrization [GHFP08]. For specular materials, including caustics, we can rely on a projection based on BRDF importance sampling. Here, the projection is warped to reflect the directional component of the outgoing radiance. Solutions for on-the-fly warping of arbitrary BRDFs exist [REG\*09].

**Geometry-Aware Filtering and Antialiasing** Similar to ISM, we can accelerate computations, by evaluating a differing VPL subset for neighboring pixels [KH01]. We then use a geometry-aware blur to combine the illumination of neighboring pixels. Such an operation is reasonable because indirect illumination is often of low frequency.

**Triangle VBOs and triangle textures** When deriving the point-based scene representation, we made use of a triangle texture that stores the world coordinates of all scene triangles. However, meshes usually come as an indexed face set in form of a VBO. To efficiently convert between the two, we use the following approach. We draw the entire triangle VBO once,

using a geometry shader (here, DX10-extensions are needed or one can use a workaround via texture coordinates) that turns a triangle into three points. Usually, a geometry shader amplifies data, but, here, we use it to perform a conversion. Receiving the  $i^{\text{th}}$  triangle, the geometry shader sends its three vertices to three consecutive pixels  $3i$ ,  $3i+1$ ,  $3i+2$  in the triangle texture. Alternatively, we could have used three textures (one for each vertex) and rely on multiple render targets to fill them with a single point, but, in practice, it proved more cache efficient to localize the points in the same texture.

**Transparency** Dithering is used to render transparent surfaces [ESSL10], as well as direct and indirect shadows from transparent surfaces (Fig. 11). To this end, before writing a fragment from a surface with transparency  $t$  to the framebuffer,  $t$  is compared to  $x$ , a value read from a repeating texture between 0 and 1 applied in screen-space. If  $t < x$ , the fragment is skipped, otherwise the rasterization proceeds as usual. The same is done when drawing surfaces into the shadow map or points into the ISM. In all results a  $2 \times 2$  dithering pattern is used, allows to produce four different opacity levels and becomes invisible when combined with a  $2 \times 2$  downsampling.

**Inter-texel VPL placement** The VPL creation follows an importance sampling according to a density function that estimates the influence of a pVPL on the final rendering. To improve the stability of this estimate, we propose two ameliorations. First, we apply a geometry-aware blur kernel (of approximate 1% of the BRSM). Second, when sampling VPLs, we will also allow inter-texel positions. Usually, one would snap the final VPL to the closest texel location, but if the BRSM has a low resolution, the VPLs tend to jump from one texel to the next leading to popping artifacts. In particular, in large scenes, as the ones we address, it is common that the resolution of the BRSM can be low when compared to the present geometrical details.

Inter-texel positions are produced when VPLs are determined from the CDF. The applied binary search (Sec. 4) can easily retrieve fractional positions in the BRSM. The final inter-texel VPL is defined by a linear weighting of the four pVPLs captured in the surrounding texels  $T_0, \dots, T_3$ . Let  $w_0, \dots, w_3$  be the bilinear weights of these texels with respect to the VPL position. If one would simply define the final VPL via a bilinear interpolation ( $\sum w_i T_i$ ), severe problems arise: the VPLs slide along depth discontinuities in world space. Hereby, surfaces can be wrongly lit and VPLs might even start to float in space. To avoid this issue, we use a geometry-aware bilinear weighting.

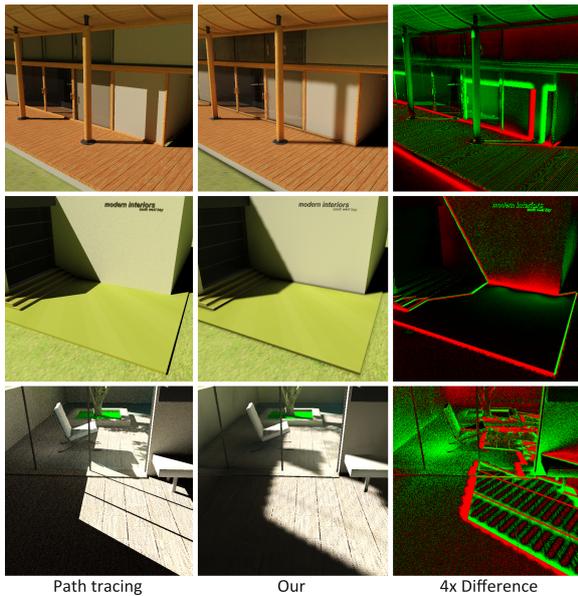
The nearest pVPL, say  $T_0$ , with respect to the selected VPL is used to define a reference. We then compute supplementary weights  $g_0 := 1, g_1, g_2, g_3$  according to the similarity of the surrounding pVPLs with respect to  $T_0$ . We define the final VPL as  $(\sum w_i g_i T_i) / (\sum w_i g_i)$ , where  $g_i$  are weights defined via a gaussian weighting function with a variance of 0.01 that is

applied to the VPL distance in the post-projective space of the current viewpoint. This choice performs well in practice and we keep more involved kernels as future work.

The inter-textel VPLs lead to a smooth movement with respect to the current view and avoid incoherencies: VPLs slide over surfaces, but jump across depth discontinuities.

## 7. Results

We tested our approach on an Nvidia GTX 480 and various scenes. Fig. 11 shows three architectural scenes, including framerates and triangle counts. All scenes are of significant complexity, have many fine details and hundreds of thousands of triangles. The viewpoint is chosen to show how light and shadow details are preserved. In all scenes, our solution obtains interactive to real-time performance at a resolution of  $1600 \times 800$ . Our approach depends only marginally on the actual viewpoint and the framerate is almost constant when navigating in the scene. A typical timing breakdown for a scene such as the one in Figure 11 leads to: 37% indirect lighting, 15% direct lighting, 12% ISM creation including blocker adaptation, also 12% reflective shadow maps including bidirectional sampling and 9% geometry-aware blur.

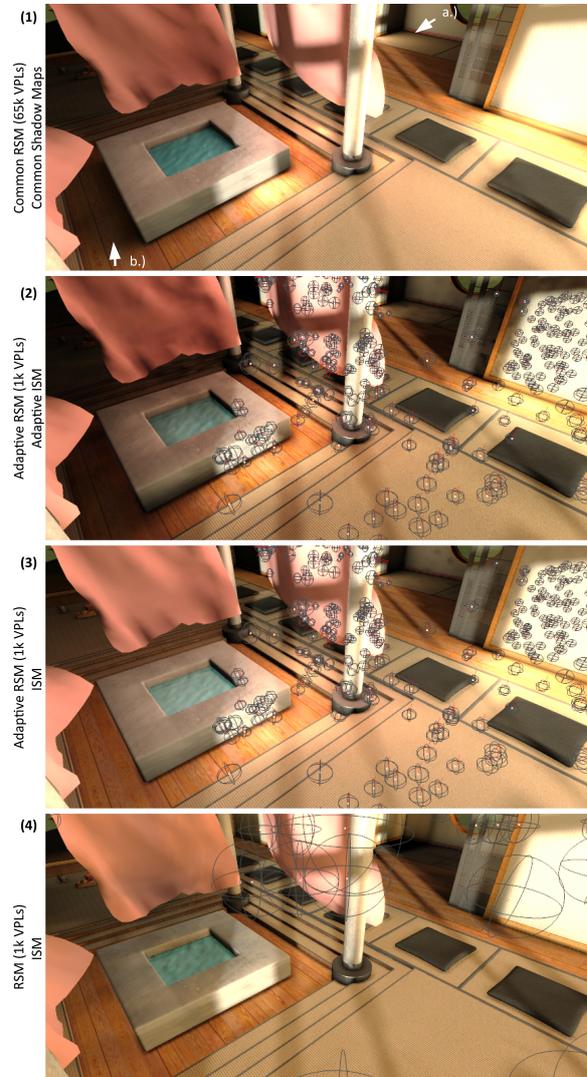


**Figure 8:** Although many orders of magnitude faster (20 Hz vs. 20 min), our solution is close to the reference image. Some color deviations are due to the usage of 8 bit textures for our interactive result and 32 bit textures for the path tracing.

Fig. 8 shows a comparison between our results and a path-tracing solution. Shadows due to indirect illumination are well captured, despite the enormous difference in performance. It is important to point out that the scene is large and the model is not only an exterior, but contains many objects in its interior.

## 8. Discussion

This section, compares the proposed approach to classic imperfect shadow maps [RGK\*08], bidirectional instant radiosity [SIP06], matrix-row-column sampling [HPB07] and Microrendering [REG\*09], before concluding with a discussion of existing limitations.



**Figure 9:** A comparison of common Instant Radiosity with shadow maps, our approach, our approach without Adaptive ISMs and our approach without Bidirectional RSMs. (1): A reference solution taking minutes to compute. (2): Our solution appears similar to the reference and reproduces its subtle lighting (a) and shadow (b) details. (3): Without ISM adaptivity the indirect shadow is lost. (4): Without bidirectional RSMs the indirect light as well as indirect shadow details are lost. Note, how the VPLs are placed where they contribute to the view in (3) and placed equally in (4).

**Imperfect shadow maps** [RGK\*08], our approach adds only 14 ms GPU overhead to the algorithm, but produces results of significantly higher quality. Especially in large scenes, the common ISM approach is not able to reproduce illumination details (Fig. 9).

**Bidirectional instant radiosity** [SIP06] also addresses the selection of VPLs that contribute significantly to the framebuffer, similar to the bidirectional RSMs proposed in this work (Fig. 10). The downside is that this approach needs to trace many test rays, which induces a high performance hit and makes real-time performance infeasible. Instead, our BRSMs also significantly improve the VPL distribution as well without the need to trace any rays, even for the u-shaped scene which is a perfect match for bidirectional IR (Fig. 4 in [SIP06]).

**Matrix row-column sampling** [HPB07] is another approach to reduce computation time for global illumination by rendering the scene only from a few optimized points of view. Every rendering corresponds to one row or column for one light resp. view sample. However, the computation required for each view is not reduced as done in our work. Therefore, the time to render a single high-resolution shadow map of the scene for one row resp. column is close to the amount of time we spend on our entire refinement process. A test showed, that using  $32 \times 32$  buffers, we could only render the entire scene around five times before exceeding the cost of our entire pipeline. Five renderings correspond to a solution with five rows resp. columns, which is much less than the several hundred used in row-column sampling (cf. Fig. 1 in [HPB07]). An ISM-like evaluation is inappropriate for row-column sampling, as the analysis step requires a high-quality sampling.

**Microrendering** [REG\*09], in theory, works for larger scenes because the point cloud is hierarchically traversed. Nevertheless, the fact that micro-buffers gather the illumination makes it more difficult to scale the quality of the final rendering. Further, for complex materials, it is very costly to evaluate the shading for all points in the scene and an evaluation during gathering might not always be practical.

**Limitations** Our lazy adaption scheme actually improves temporal coherence, but might introduce some lag in the computation. In such a case, the details in indirect shadows continuously appear with the adaptive refinement of the blockers. The speed of this process is given by the ratio of updated points in each frame. We kept the same update frequency (1/8) in all videos.

The discretization of rasterized images could become a problem if important parts of the reflective shadow map project to less than a pixel, as spatial details would be lost.

Our approach is not independent of the total scene size, as we still rely on a pass over all triangles, nonetheless, this is

common and almost inevitable for dynamic scenes. Furthermore, our approach is compatible with LOD schemes.

We don't introduce any new restrictions to ISMs. Instead, while area-preserving deformations were required for ISM to ensure high quality, these restrictions are removed as a byproduct of our adaptive extensions. However, our solution shares some of the limitations of ISMs (see [RGK\*08]): The limited directional resolution requires indirect light to be smooth, although our approach already performs better due to the use of warped VPL frusta. For specular effects we can even take the viewer into account. The second limitation is that scene geometry needs to be well representable by a reasonable amount of points – our approach significantly improves this issue, but might still lead to a few limitations. We could rely on a variable number of samples per VPL ensuring a fixed quality (less GPU suitable), instead of our current solution which is to use a fixed number of samples per VPL which can lead to varying quality, but is more GPU-friendly.

## 9. Conclusion and Future Work

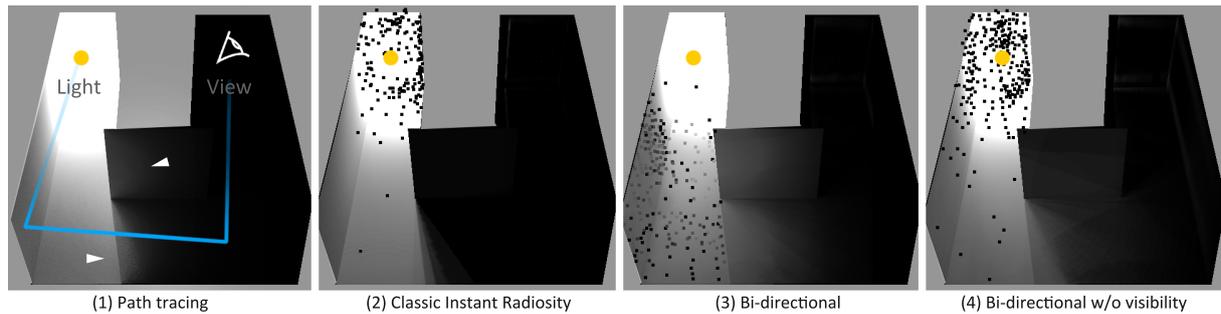
We presented a new algorithm that extends ISM and achieves convincing indirect global illumination in dynamic scenes of millions of polygons with interactive to real-time performance. While previous work was mostly limited to scenes of a small extent, our solution scales well with size due to its GPU-adapted scene resampling. We presented a new method to select VPLs according to their contribution to the actual output. It leads to a distribution that is often beneficial and is applicable beyond the algorithm of this paper. We adapt VPL projections to increase the quality of the rendering and showed an efficient way of defining inter-textel VPLs to increase coherence and avoid rendering artifacts.

BRSMs can also be used to importance-sample environment maps or area lights as well. Adaptive imperfect shadow maps further extend to multiple bounces similar to ISMs Reflective Imperfect Shadow Maps [RGK\*08].

In future work, we would like to extend the two main ideas of this work – efficiently accounting for the output framebuffer combined with a simple GPU-based adaptive geometry representation – to other rendering problems, such as (glossy) reflections. Beyond rendering, more general simulation problems such as crowds, granulars or fluids that are re-sampled on the GPU to account for an output framebuffer could benefit from the techniques suggested in this work.

## References

- [AFO05] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Fast and detailed approximate global illumination by irradiance decomposition. In *ACM Trans. Graph. (Proc. SIGGRAPH)* (2005), pp. 1108–1114. 2
- [CB04] CHRISTENSEN P. H., BATALI D.: An irradiance atlas for global illumination in complex production scenes. In *Proc. EGSR* (June 2004), pp. 133–141. 2



**Figure 10:** A comparison between ground truth path tracing, classic instant radiosity, bi-directional instant radiosity and our approach. In this scene, the camera (white triangle, right) faces the back of one side of the u-shaped scene and the light is placed on the other side (yellow, left), resulting in a difficult light configuration where only very few view samples are connected with light samples (blue line). Importance of pVPLs is dominated by visibility in such a scene. 2: Classic IR, does not reproduce much of the light and shadows on the front of the U (arrows) because most VPLs (black dots) clump around the light source. 3: Bidirectional IR places more VPLs where they contribute more to the current view using costly ray tracing to determine visibility. 4: Our VPL distribution is less optimal than Bidirectional IR, but much better than classic IR and computable in an efficient manner.

- [DGR\*09] DONG Z., GROSCH T., RITSCHER T., KAUTZ J., SEIDEL H.-P.: Real-time indirect illumination with clustered visibility. In *Proc. VMV* (2009). 2
- [DKTS07] DONG Z., KAUTZ J., THEOBALT C., SEIDEL H.-P.: Interactive global illumination using implicit visibility. In *Proc. Pacific Graphics* (2007). 2
- [DS05] DACHSBACHER C., STAMMINGER M.: Reflective shadow maps. In *Proc. I3D* (2005), pp. 203–213. 1, 2, 3, 5
- [DSDD07] DACHSBACHER C., STAMMINGER M., DRETTAKIS G., DURAND F.: Implicit visibility and antiradiance for interactive global illumination. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26, 3 (2007). 2
- [ESSL10] ENDERTON E., SINTORN E., SHIRLEY P., LUEBKE D.: Stochastic transparency. In *Proc. I3D* (2010), pp. 157–164. 7
- [GHFP08] GASCUEL J.-D., HOLZSCHUCH N., FOURNIER G., PEROCHE B.: Fast non-linear projections using graphics hardware. In *Proc. I3D* (Feb. 2008). 7
- [HPB07] HAŠAN M., PELLACINI F., BALA K.: Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26, 3 (2007). 2, 8, 9
- [HSC\*05] HENSLEY J., SCHEUERMANN T., COOMBE G., SINGH M., LASTRA A.: Fast summed-area table generation and its applications. *Comp. Graph. Forum (Proc. Eurographics)* 24, 3 (2005), 547–555. 4
- [HVAPB08] HAŠAN M., VELÁZQUEZ-ARMENDÁRIZ E., PELLACINI F., BALA K.: Tensor clustering for rendering many-light animations. *Comput. Graph. Forum (Proc. of EGSR)* 27, 4 (2008), 1105–1114. 2
- [Jen01] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, 2001. 1
- [Kaj86] KAJIYA J. T.: The rendering equation. *Comput. Graph. (Proc. SIGGRAPH)* 20, 4 (1986), 143–150. 1
- [KD10] KAPLAYAN A., DACHSBACHER C.: Cascaded light propagation volumes for real-time indirect illumination. In *Proc. I3D* (2010). 2
- [Kel97] KELLER A.: Instant radiosity. In *Proc. SIGGRAPH* (1997), pp. 49–56. 1, 2, 5
- [KH01] KELLER A., HEIDRICH W.: Interleaved sampling. In *Proc. EGWR* (2001), pp. 269–276. 7
- [LSK\*07] LAINE S., SARANSAARI H., KONTKANEN J., LEHTINEN J., AILA T.: Incremental instant radiosity for real-time indirect illumination. In *Proc. EGSR* (2007), pp. 277–286. 2
- [MKC07] MARROQUIM R., KRAUS M., CAVALCANTI P. R.: Efficient point-based rendering using image reconstruction. In *Point-Based Graphics, Proc. Symp. on* (2007), pp. 101–108. 3
- [NSW09] NICHOLS G., SHOPF J., WYMAN C.: Hierarchical image-space radiosity for interactive global illumination. *Comp. Graph. Forum (Proc. EGSR)* 28, 4 (2009). 2
- [REG\*09] RITSCHER T., ENGELHARDT T., GROSCH T., SEIDEL H.-P., KAUTZ J., DACHSBACHER C.: Micro-rendering for scalable, parallel final gathering. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 28, 5 (2009). 2, 7, 8, 9
- [RGK\*08] RITSCHER T., GROSCH T., KIM M. H., SEIDEL H.-P., DACHSBACHER C., KAUTZ J.: Imperfect shadow maps for efficient computation of indirect illumination. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 27, 5 (2008). 1, 2, 3, 8, 9
- [RGS09] RITSCHER T., GROSCH T., SEIDEL H.-P.: Approximating dynamic global illumination in image space. In *Proc. I3D* (Feb. 2009), pp. 75–82. 2
- [RPV93] RUSHMEIER H., PATTERSON C., VEERASAMY A.: Geometric simplification for indirect illumination calculations. In *Proc. Graphics Interface* (1993), pp. 227–236. 2
- [SD95] SILLION F., DRETTAKIS G.: Feature-based control of visibility error: A multi-resolution clustering algorithm for global illumination. In *Proc. SIGGRAPH* (Aug. 1995), vol. 29, pp. 145–152. 2
- [SIP06] SEGOVIA B., IEHL J.-C., PEROCHE B.: Bidirectional instant radiosity. In *Proc. EGSR* (June 2006). 2, 3, 8, 9
- [SKS02] SLOAN P.-P. J., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. (Proc. SIGGRAPH)* 21, 3 (2002), 527–536. 1
- [ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-D shapes. *Comp. Graph. (Proc. SIGGRAPH)* 24, 4 (1990), 197–206. 2



**Figure 11:** Results obtained with our method at different levels of zoom resp. lighting detail. Note the indirect shadows, i.e., in the top-right image. All images rendered at  $1600 \times 800$  with  $2 \times 2$  super-sampling from a  $8 \times 8$  G-buffer with 1024 VPLS,  $2048 \times 2048$  ISM with 8k points,  $4 \times 4$  samples bidirectional importance. All scenes are out-of-the box commercial architectural models without any manual data-preparation.

[TL04] TABELLION E., LAMORLETTE A.: An approximate global illumination system for computer generated films. In *Proc. SIGGRAPH* (2004), pp. 469–476. 2

[WFA\*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: A scalable approach to illumination. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (2005), 1098–1107. 2

[WKB\*02] WALD I., KOLLIG T., BENTHIN C., KELLER A., SLUSALLEK P.: Interactive global illumination using fast ray-

tracing. In *Proc. EGRW* (2002), pp. 15–24. 1

[YCK\*09] YU I., COX A., KIM M. H., RITSCHER T., GROSCH T., DACHSBACHER C., KAUTZ J.: Perceptual influence of approximate visibility in indirect illumination. In *ACM Trans. Appl. Percept. (Proc. APGV)* (2009), pp. 24:1–24:14. 2, 3