Imperfect Shadow Maps for Efficient Computation of Indirect Illumination

T. Ritschel* T. Grosch* M. H. Kim† H.-P. Seidel* C. Dachsbacher‡ J. Kautz†

MPI Informatik* University College London† Universität Stuttgart‡

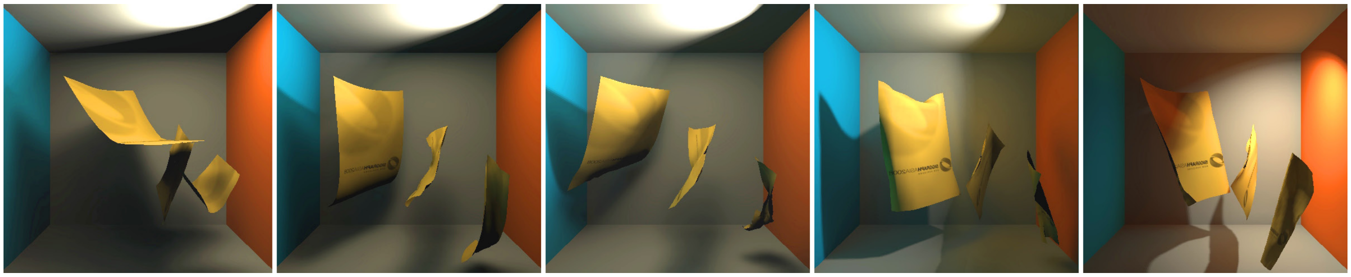

Figure 1: *Global illumination for a completely dynamic scene (light, view, geometry, material) rendered at 19 fps on an NVIDIA GeForce 8800 GTX. The scene is illuminated with a small spot light (upper right); all other illumination and shadowing is indirect (one bounce).*

Abstract

We present a method for interactive computation of indirect illumination in large and fully dynamic scenes based on approximate visibility queries. While the high-frequency nature of direct lighting requires accurate visibility, indirect illumination mostly consists of smooth gradations, which tend to mask errors due to incorrect visibility. We exploit this by approximating visibility for indirect illumination with *imperfect shadow maps*—low-resolution shadow maps rendered from a crude point-based representation of the scene. These are used in conjunction with a global illumination algorithm based on virtual point lights enabling indirect illumination of dynamic scenes at real-time frame rates. We demonstrate that imperfect shadow maps are a valid approximation to visibility, which makes the simulation of global illumination an order of magnitude faster than using accurate visibility.

Keywords: global illumination, visibility, real-time rendering

1 Introduction

Global illumination effects, such as indirect illumination, are known to be perceptually important [Stokes et al. 2004] but are often omitted or coarsely approximated due to their high rendering cost—especially in interactive applications. One of the most expensive components when computing indirect illumination is visibility determination, i.e., determining if two points are mutually visible or not. Commonly, this is done accurately using ray-casting or other similar visibility queries [Cohen-Or et al. 2003], despite anecdotal evidence that visibility does not need to be accurate when computing indirect illumination [Sillion 1995; Arikan et al. 2005].

We address the challenge of costly visibility with *imperfect shadow maps* (ISMs) in order to enable real-time global illumination of *large* and *fully dynamic* scenes. ISMs are low-resolution shadow maps that may contain inaccurate visibility information, i.e., some of the stored depths may be incorrect. Nonetheless, we demonstrate that ISMs can be combined with instant radiosity (IR) [Keller 1997] to compute convincing indirect illumination. IR represents indirect lighting with a set of virtual point lights (VPLs) for which visibility is usually determined accurately, either using shadow volumes [Keller 1997; Crow 1977] or high-resolution shadow maps [Williams 1978], constituting the main bottleneck. While shadow maps can be precomputed for static scenes and even partially recomputed on-the-fly to handle light changes [Laine et al. 2007], the recomputation of all shadow maps (or volumes) for a dynamic scene is no longer possible at interactive speed, even for modest-sized scenes. In contrast, we show that hundreds of ISMs can be computed very efficiently on current GPUs and that they can replace accurate visibility, therefore enabling IR-based real-time indirect illumination of fully dynamic scenes. Despite the inaccuracies contained in ISMs and the limitations inherited by the instant radiosity approach, the resulting renderings are very similar to reference renderings but rendered at ten times the original speed.

Our main contributions include:

- An approximate representation of visibility — imperfect shadow maps — to facilitate the computation of indirect illumination.
- An instant-radiosity based technique that uses ISMs to compute (multi-bounce) indirect illumination in large, dynamic scenes in real-time on modern GPUs.
- An analysis of the influence of ISMs on the resulting indirect illumination.

2 Related Work

Global Illumination Methods Many standard global illumination methods, such as path tracing, photon mapping, and ray-tracing [Dutré et al. 2006] use accurate visibility for direct as well as indirect illumination, usually by intersecting rays with the scene geometry. Even popular techniques that are geared towards efficient computation, such as instant radiosity [Keller 1997] and instant global illumination [Wald et al. 2002]—both based on virtual point lights (VPLs)—use accurate visibility either through shadow volumes/maps or ray-casting respectively.

Methods exist that make more explicit use of visibility approximations. Sillion and Drettakis [1995] proposed a radiosity method that approximates visibility such that a user-defined feature-size can still be resolved while blurring out smaller features. No distinction was made between direct and indirect illumination. The lightcuts method [Walter et al. 2005] groups a large number of light samples (VPLs representing the indirect illumination) into a hierarchy to speed up rendering. Visibility between a surface point and a node in the hierarchy (a collection of VPLs) is approximated with a single shadow ray. However, the algorithm bounds the resulting error, such that no perceptible artifacts appear. Arikan et al. [2005] propose an approximate global illumination method where irradiance arriving from nearby and distant geometry is decoupled: all incident lighting from nearby geometry is simply integrated without computing visibility at all. Methods have been proposed to reduce the geometric complexity when computing global illumination solutions [Rushmeier et al. 1993; Christensen et al. 2003; Tabellion and Lamorlette 2004]. Our method is based on similar insights; however, we use a simplified, point-based representation of the geometry only to determine visibility and not for shading calculations.

Besides the method by Arikan et al. [2005], explicit use of approximate visibility to decrease the cost of computing indirect illumination has not been considered. We further explore the idea of approximate visibility and demonstrate that it enables real-time global illumination.

Real-Time Global Illumination Many techniques for real-time global illumination, such as precomputed radiance transfer (PRT) techniques, only support lighting changes and are restricted to static [Sloan et al. 2002] or semi-static scenes, which only allow movement of rigid objects [Iwasaki et al. 2007; Wang et al. 2007]. These restrictions enable the precomputation of light transport including all visibility information. Similarly, real-time rendering of static scenes with one-bounce indirect illumination is possible with a variant of instant radiosity that reuses VPLs and their respective shadow maps over multiple frames [Laine et al. 2007]. In contrast, our method enables fully dynamic scenes and multiple bounces of indirect illumination.

Real-time global illumination for dynamic models is possible if visibility for indirect illumination is completely neglected [Dachsbacher and Stamminger 2005; Dachsbacher and Stamminger 2006]. If visibility is taken into account, scene complexity is usually limited to a few thousand polygons in order to achieve interactivity [Bunnell 2005; Dong et al. 2007; Dachsbacher et al. 2007]. A first method to take visibility into account was presented by Bunnell [2005]. It uses a hierarchical link structure, where visibility between elements is approximated by ambient occlusion. Dong et al. [2007] proposed an interactive global illumination method for small scenes, where visibility is implicitly evaluated on the hierarchical finite element structure. A variant of hierarchical radiosity, which replaces explicit visibility queries with an iterative process using *antiradiance* (negative light), has been shown to enable interactive, dynamic global illumination [Dachsbacher et al. 2007]. However, the use of dynamic objects is restricted due to the need for highly tessellated surfaces near indirect shadows. Sloan et al. [2007] demonstrate real-time indirect illumination for larger scenes, but only low-frequency incident radiance as well as low-frequency visibility are supported as these are represented with a small number of spherical harmonics.

In contrast, ISMs enable real-time global illumination of fully dynamic scenes through the use of approximate visibility without being restricted to low-frequency illumination or small scenes. In fact, our technique scales to scenes with several hundred thousand polygons.

Visibility Approximations in Rendering Recently, it has been shown that accurate occlusion for glossy reflections is not always necessary [Kozlowski and Kautz 2007]—an observation that has already been exploited in rendering glossy reflections from environment maps [Green et al. 2007]. The motivation for our method is similar: we want to speed up the costly computation of global illumination by approximating visibility.

3 Imperfect Shadow Maps

Visibility tests between pairs of 3D points are often the most time-consuming part in global illumination algorithms. These queries can be made more coherent using algorithms that compute indirect illumination based on virtual point lights (VPLs) [Keller 1997; Wald et al. 2002], where visibility needs to be computed only between a small number of VPLs (several hundred) and all other scene points instead of many arbitrary pairs of scene points. A good choice for solving these coherent visibility queries is the use of shadow mapping [Williams 1978]. However, computing several hundred shadow maps per frame is too slow for even moderately-sized scenes and does not scale well with scene complexity [Laine et al. 2007].

Our goal is to achieve real-time frame rates for large and dynamic scenes, which necessitates a method that can quickly solve visibility queries for hundreds of dynamically generated VPLs. Realizing that fully accurate visibility queries are not necessary for computing indirect illumination, we propose the use of imperfect shadow maps (ISMs). ISMs are low-resolution shadow maps that are computed from a crudely simplified point-representation of the scene, followed by a pull-push pass to fill holes. While ISMs may contain incorrect depth values, the resulting errors in the indirect illumination are small but the computational gains are significant, which we will demonstrate in detail in Section 5.

3.1 Scene Preprocessing

In a preprocessing step, we approximate the 3D scene by a set of points with roughly uniform density. Each point is created by randomly selecting a triangle with probability proportional to the area of the triangle, and then picking a random location on the triangle. More elaborate schemes such as point repulsion can be used but did not prove to be necessary. For each point, we also store its barycentric coordinates relative to its triangle as well as the triangle index in order to support dynamic scenes without the need to recompute the point representation. The barycentric coordinates also enable us to retrieve the normal and reflectance for each point, which will be required for secondary indirect light bounces.

3.2 ISM Creation

A single ISM is created by splatting the point representation into the depth-buffer (with a box splatting kernel, i.e., GL points), where the size of a point splat is based on its squared distance to the corresponding VPL position (assuming each point sample represents the same average area). When an ISM is used for computing indirect illumination, it needs to cover a full hemisphere of depth information (VPLs emit light over a hemisphere) for which we use parabolic maps [Brabec et al. 2002]. In general, ISMs can be used with other common shadow map projections, such as orthographic or perspective projection. As indicated before, we can easily handle dynamic scenes by deforming the points according to the deformation of the corresponding triangles. While this leads to non-uniform point distributions, our results do not suffer as our technique does not depend on accurate uniform distributions.

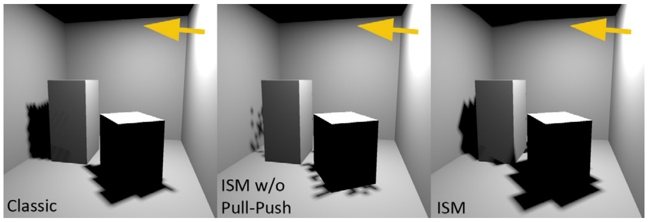

Figure 2: *Comparison of a classic shadow map and an imperfect shadow map with and without pull-push. The ISM contains holes and incorrect depth values. Most of these pixels are corrected after the pull-push operation. Note that a single ISM is not expected to produce an accurate shadow. However, shadows from many ISMs average out and artifacts disappear, which is ideal for rendering indirect illumination.*

Many low-resolution ISMs are created and rendered in one pass and stored in a single, large texture. To this end, a vertex shader splits up the incoming stream of points representing the scene and distributes an equal amount of points to each ISM within the large texture. Each ISM receives a fixed, random subset of the point set. We typically use parabolic ISMs with a resolution of 128×128 pixels, stored in a 4096×4096 texture.

In order to reduce computation, we only use a sparse set of points, which may leave holes in the depth map. We therefore use a pull-push approach [Grossman and Dally 1998; Marroquim et al. 2007] to fill these holes and reconstruct a sensible depth map. The first step is the creation of an image pyramid in the pull phase where the image is downsampled by a factor of two in each stage. Only valid pixels in the finer level are used for averaging the pixels in the coarser level. The second step is the push phase where the holes are filled top-down by interpolating the pixels from the coarser level to approximate the undefined pixels in the finer level. Inspired by Marroquim et al. [2007], we use outlier rejection via thresholding in both the pull and push phase. In particular, we only combine depth values during the pull phase that are close to each other, and we only replace depth values during the push phase that are far from the coarse depth values that are pushed down. We use separate depth thresholds for the pull and the push phase, which are scaled by 2^l according to the mip-map level l (0 being the finest level); both threshold values are typically set to 5% of the scene extent. In all our results we only go up two levels in the pyramid, which virtually fills all holes.

3.3 Discussion

Given an effective resolution of $N_{\text{res}} = (\pi/4)\, n \times n$ for a single parabolic ISM and P points, an average of $\bar{P} = P/N_{\text{res}}$ points land on every ISM pixel. Assuming optimal point sampling, it is possible to find the correct depth if the scene has an average depth complexity of $\bar{P}$. In reality, the samples will be less optimal and only a much smaller depth complexity can be resolved [Wand et al. 2001]. Pull-push hole filling mitigates this; however, a fully correct depth map can never be expected due to the randomness of the point samples. Figure 2 demonstrates this: shadows from a single point light are rendered with traditional shadow maps as well as ISMs. In this case, artifacts are visible when using an ISM. The pull-push pass does improve the result considerably, yet differences to the reference shadows are still visible. However, these differences do not necessarily result in noticeable artifacts for indirect illumination, as the contributions of many ISMs are summed up, e.g., see Figure 1. Examples of ISMs with and without pull-push can be found in Figure 3 on the right.

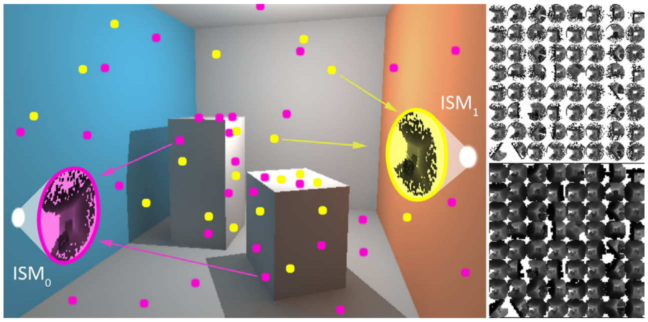

Figure 3: *Global Illumination with imperfect shadow maps: The indirect light in this didactic scene is represented by two VPLs. Each VPL generates a shadow map using a point-based representation of the scene. Since coarse visibility information is sufficient, each VPL uses only a sparse subset of the points (encoded as colors) to generate an* imperfect shadow map *at real-time rates. The contribution of all VPLs is summed up yielding the final indirect illumination. Examples of imperfect shadow maps (bottom with and top without pull-push) are shown on the right; dark value signify small depth values and light values represent large depths.*

We use a simple point-based geometry representation mainly because it can be easily used to create hundreds of ISMs in parallel and it offers straight forward support for dynamic scenes. We have considered alternative representations for rendering imperfect shadow maps, such as more advanced point representations [Wand et al. 2001; Dachsbacher et al. 2003] or polygonal level-of-detail [Hoppe 1996] but these either require location-dependent evaluation, which is costly for hundreds of ISMs, or cannot easily handle complex geometry, such as fences or trees.

4 Indirect Illumination with ISMs

As mentioned earlier, the indirect illumination in our work is based on the idea of instant radiosity [Keller 1997], where virtual point lights are distributed stochastically along light paths which then act as sender of indirect light. Summing up the contribution of all VPLs, while taking into account shadowing, yields the resulting indirect illumination.

We propose to use imperfect shadow maps to determine visibility for each VPL, as they are highly efficient—hundreds of ISMs can be rendered in a single pass—and their rendering cost is only loosely dependent on the overall polygon count. In order to maintain high frame rates, we create the virtual point lights from the light source completely on the GPU, similar to Dachsbacher and Stamminger [2006] (single bounce for now). The 3D position of each VPL is determined by rendering a cube map from the viewpoint of the (direct) point light source, on which we perform importance sampling by building the necessary cumulative and marginal distributions from which we select N_{vpl} VPLs. Of course, other methods that generate VPLs, such as photon shooting [Keller 1997], could also be used. Each VPL creates a parabolic ISM, oriented along the normal of the surface capturing the whole hemisphere in a single view. As described before, only a subset of all the points is used for each ISM and all ISMs are stored in one large texture (see Figure 3). To further reduce the rendering cost, a G-Buffer is used to efficiently gather from N_{vpl} VPLs [Segovia et al. 2006] during rendering of the final image. In other words, we only gather from a subset of VPLs at each pixel and perform a geometry-aware blur to remove noise.

Like most of the instant radiosity methods, this VPL-based illumination restricts us to diffuse and slightly glossy materials, unless a very large number of VPLs are used.

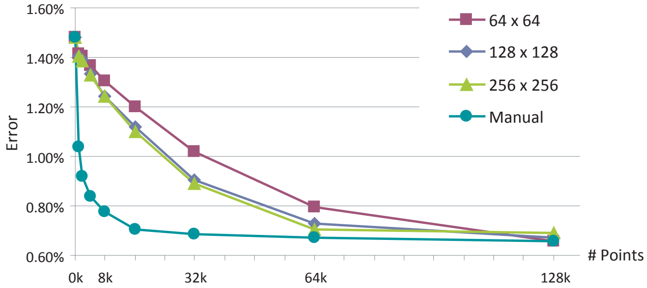

Figure 4: *Image error (normalized RMS) of the Sponza scene w.r.t. the reference GPU solution. The number of point samples as well as ISM resolution is varied. The error goes down by increasing either of them. Manual adjustment of all parameters (ISM resolution, point samples, splat size, etc.) yields the lowest overall error.*

4.1 Multiple Bounces

For multiple bounces, we generalize ISMs to *imperfect reflective shadow maps* (IRSM) in the same way classic shadow maps generalize to reflective shadow maps [Dachsbacher and Stamminger 2005]. This allows us to generate VPLs for each additional bounce directly on the GPU, which we then render with ISMs. Given the VPLs for the first bounce, we generate VPLs for the second bounce as follows: the point-based representation of the scene is used to create imperfect reflective shadow maps for each of the initial VPLs; i.e., instead of rendering the shaded geometry we render *shaded points* into the IRSMs. This results in a large texture which contains all paraboloid maps of the first bounce VPLs, but each paraboloid now stores its own indirect illumination (instead of depth values). Importance sampling is then applied to this texture in order to generate the second bounce VPLs (similar to the first bounce VPL generation from the direct light cube map). The position and normal information extracted from the points is used to position and orient the second bounce VPLs onto the geometry. The illumination of these second-bounce VPLs is then rendered with imperfect shadow maps much like the illumination from the first-bounce VPLs. The same principle can be applied to render further bounces.

5 Results

In the following, we present results rendered at real-time rates with our technique on an 1.8 GHz CPU with an NVIDIA GeForce 8800 GTX. As mentioned before, all scene components can be fully dynamic (geometry, materials, and lights). Unless mentioned otherwise, all results are rendered with a single-bounce of indirect illumination at 640 × 480 pixels and a G-Buffer block size of 8 × 8 pixels.

Figure 12 (last page) shows four scenes rendered with our technique (one- or two-bounce, diffuse indirect illumination). We vary both the ISM resolution as well as the number of points used for each ISM to demonstrate their influence on the result. Furthermore, we compare our results to a reference GPU solution, which uses accurate (256 × 256) shadow maps, as well as a reference solution computed using path tracing (using the same number of bounces). Generally speaking, our method produces results that are very similar to the reference GPU solution but is an order of magnitude faster. However, there are some differences. Our method produces indirect shadows that are bit softer than with classic shadow maps (e.g., see first scene). This is to be expected and is not a visually distracting difference. Indirect shadows of small or thin objects require a sufficient number of points to be resolved (see column of second scene with 4000 vs. 16000 points). Furthermore, if the ISMs' resolution is insufficient, indirect shadow areas can become overly dark or bright, e.g., see the close-ups of the last scene.

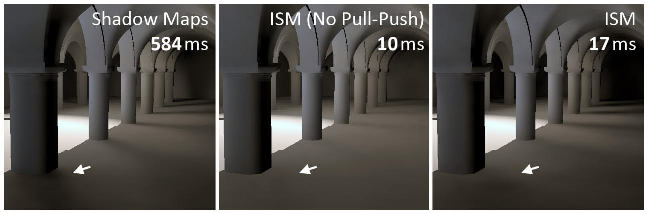

Figure 5: *Using ISMs without pull-push often creates light leaks. Here, the columns do not cast a dark enough (indirect) shadow. Using pull-push largely removes these artifacts.*

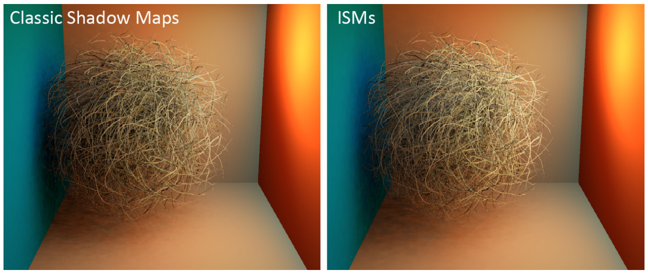

Figure 6: *ISMs can handle geometrically complex scenes. Left: Classic Shadow Maps. Right: ISMs (1024 VPLs, 256 × 256, and 4K points).*

Figure 4 shows the normalized root-mean-square error w.r.t. the GPU reference for the Sponza scene for three different ISM resolutions as well as a varying number of points, where all other parameters are fixed (512 VPLs). As expected, based on the observations from Figure 12, more point samples reduce the error. Increasing ISM resolution decreases the error as well; however, the difference in error between 128 × 128 and 256 × 256 is small. The manual curve demonstrates that adjusting parameters by hand (ISM resolution, manually modified splat size, G-Buffer setting) can achieve high fidelity with a small number of point samples[1]. Note that the error does not fully vanish due to the G-Buffer.

Figure 5 shows how the pull-push pass effectively removes small light leaks. The additional overhead is small and worth the cost. In comparison to classic shadow maps, ISMs can be created more than 30 times faster for this particular scene (using 256 VPLs and an ISM resolution of 256 × 256), even with pull-push. Furthermore, even complex scenes with fine geometric detail can be handled with ISMs and pull-push, see Figure 6.

Our generalization to multiple bounces using imperfect reflective shadow maps is illustrated in Figure 7. The Cornell box is shown with one, two and three indirect bounces. Again, the scene is fully dynamic, geometry, view, lighting, and materials can be changed on-the-fly. Nonetheless, it is possible to render multiple indirect bounces at interactive rates.

Our method can handle diffuse as well as glossy indirect illumination. Figure 8 demonstrates rings of varying glossiness with a diffuse cloth occluding some of the light that is reflected from the ring. While there are faster dedicated methods for rendering caustics,

[1] Our strategy here was to first choose parameters such that the number of point samples times minimum splat size divided by N_{res} remained roughly constant, followed by a manual adjustment.

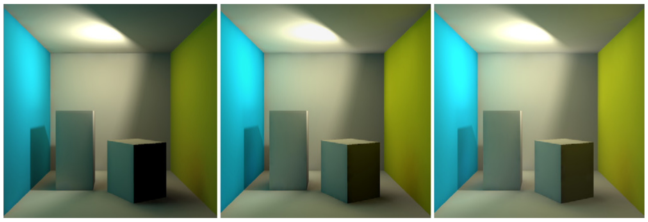

Figure 7: *From left to right: One, two and three bounces of indirect illumination rendered at* 1024 × 1024 *with 14.0, 7.5, and 5.5 fps respectively (using 1024 VPLs for the first bounce and 256 VPLs each for the second and third).*

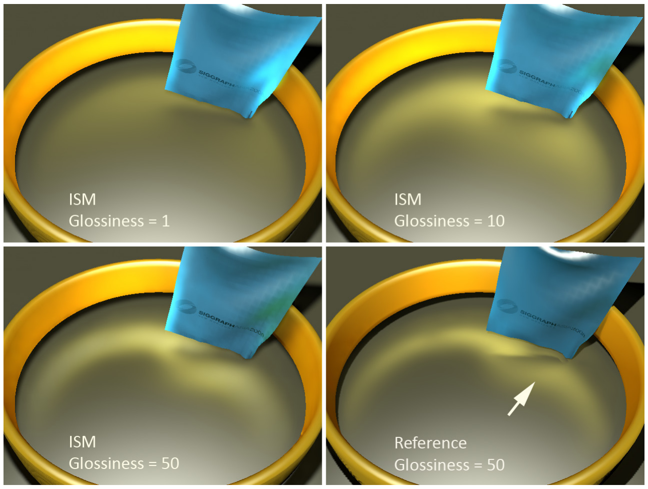

Figure 8: *A deforming cloth inside the classic metal ring with one bounce at 13 fps. The ring's BRDF varies from slightly glossy (Phong with exponent of* $N = 1$*) to more specular (exponent of* $N = 50$*). Note how the glossy bounce casts shadows that significantly contribute to the appearance. We used 4096 VPLs,* 64 × 64 *ISMs, and 2K points each. The reference rendering (converged IR solution) shows that some of the high-frequency effects, such as crisp indirect shadows, cannot be reproduced with low-resolution ISMs.*

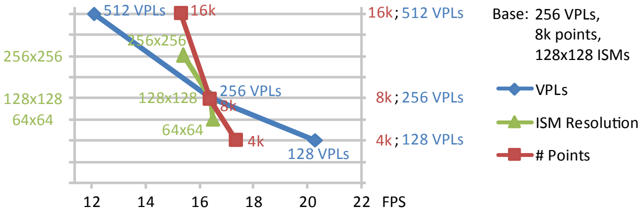

Figure 9: *Performance scalability of the Sponza scene with respect to number of VPLs, number of point samples and shadow map resolution. One parameter at a time is varied and the achieved frame rate is graphed.*

Resolution	Same G-Buffer		Same Quality	
	FPS	G-Buffer	FPS	G-Buffer
640 × 480	15.1 fps	4 × 4	15.1 fps	4 × 4
800 × 600	13.2 fps	4 × 4	17.0 fps	8 × 8
1024 × 768	10.7 fps	4 × 4	17.3 fps	8 × 8
1280 × 1024	8.6 fps	4 × 4	15.2 fps	16 × 16
1600 × 1280	6.3 fps	4 × 4	14.7 fps	16 × 16
1920 × 1600	4.6 fps	4 × 4	12.1 fps	16 × 16

Table 1: *Varying screen size for Sponza scene (*128 × 128 *and 4K points), while either keeping G-Buffer settings or quality fixed.*

such as Dachsbacher and Stamminger [2006] or Wyman [2008], they do not allow for occluded glossy indirect illumination of dynamic objects.

Temporal coherence is generally very high. However, temporal flickering can occur if an insufficient number of VPLs is used, much like the original instant radiosity method [Keller 1997]. This is further exacerbated when too few point samples are used. However, for all our test scenes we were able to achieve temporally coherent results at interactive rates (10 fps or more).

For a typical scene (Sponza with 256^2 ISM resolution and 8K points in this case), the rendering times can be broken down as follows: 7 ms for the reflective shadow map including importance sampling (1024 VPLs), 44 ms for creating the ISMs, 8 ms for the pull-push pass, 15 ms to render the scene using ISMs (16 VPLs per pixel), 4 ms for the G-Buffer blur, and 11 ms for the direct light. Less complex scenes require fewer VPLs, and the computation time goes down accordingly; see the blue curve in Figure 9, where the number of VPLs is varied for the Sponza scene. The number of points used has a direct influence on the amount of time it takes to create the ISMs, see the red curve in Figure 9, which is to be expected. However, note that our method scales well with polygon count: an increase in scene tessellation does not increase the cost to compute indirect illumination, which is decoupled through the use of a point representation; only the cost of direct illumination increases. Increasing the resolution of the shadow maps has little impact on rendering performance (green curve in Figure 9). Of course, larger ISMs require more point samples in order to actually improve rendering quality. The rendering performance can be kept almost constant for increasing screen resolutions by increasing the G-Buffer block size accordingly (see Table 1). This also keeps the quality approximately equivalent, whereas fixing the G-Buffer setting effectively increases the rendering quality for larger screen resolutions. As these numbers illustrate, there is no single performance bottleneck in our algorithm. In fact, our method is specifically designed to eliminate the geometry and render setup bottleneck that was common to previous VPL-based methods.

5.1 Other Applications of ISMs

Area Lights ISMs can also be used as a method to render soft shadows from (textured) area lights of complex shape. Figure 10 shows an animated scene with a glossy BRDF illuminated by a multicolored, curved area light, which we approximated with 512 point lights. The shadows from each of these point lights are computed with ISMs. The resulting rendering does not exhibit noticeable artifacts, despite each individual ISM being inaccurate. While there are techniques solely targeted at rendering rectangular area lights which provide higher performance, e.g. Guennebaud et al. [2006], ISMs enable textured light sources of complex shape as well as the integration of the BRDF with the incident lighting yielding glossy reflections.

Environment Maps The same idea can be used to render direct illumination from environment maps. We approximate the environment map with 1024 VPLs (again using importance sampling) and use ISMs to compute the shadows. Figure 11 shows an example where a glossy object is illuminated by an environment map, rendered both with classic shadow maps and ISMs. There are only minor differences between the two, demonstrating that ISMs can be a useful technique for environment map lighting of fully dynamic scenes including glossy reflections. There are other techniques that enable the efficient evaluation of direct illumination from environ-

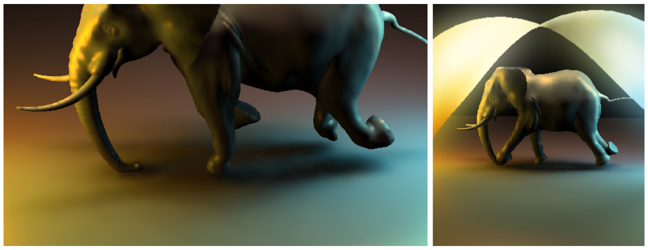

Figure 10: *A curved area light (right) illuminating an elephant (85k faces) is approximated with 512 point lights and then rendered with ISMs at 17 fps (256 × 256 and 8K points each).*

ment maps for dynamic scenes [Annen et al. 2008]; however, our method allows for the correct integration of the BRDF with the environment map lighting.

5.2 Discussion

While ISMs are a powerful method to speed up the computation of indirect illumination, there are some limitations. In our current implementation, we use reflective shadow maps (RSMs) to generate VPLs [Dachsbacher and Stamminger 2005], restricting us to point and spotlight illumination. However, other techniques to create VPLs can be used to lift this restriction.

Our technique is not limited by the total polygon count of the scene; however, good results require a sufficient number of point samples, which we currently choose by hand. Scenes with a high depth complexity require more samples, otherwise the ISMs cannot contain meaningful information. For very large scenes with many different rooms, portal-based techniques could be used to break up the point representation into smaller chunks but this remains future work.

In order to maintain high frame rates, we only use low-resolution ISMs. As a consequence, indirect shadows from small geometry cannot be resolved (e.g., see thin columns in Figure 12). This is usually not crucial if the direct illumination covers large parts of a scene yielding only very low-frequency indirect shadows in the first place. As our technique is based on instant radiosity, we inherit some of the limitations. A sufficient number of VPLs need to be created, or the indirect illumination is prone to artifacts such as temporal flickering. Light leaking due to shadow biasing may appear, as can be seen in the horse example in Figure 12. In general though, failure cases can be prevented by increasing the ISM resolution and the number of VPLs and point samples as well as adjusting shadow biasing parameters.

Our method is not fully automatic—parameters need to be chosen by the user, such as the number of point samples or depth bias. It is possible to provide some rough guidelines on how to set these parameters. In all our (diffuse) test scenes, 1024 VPLs have been used and provide good quality and very little flickering. Fewer VPLs would even suffice for some of our less complex scenes. Empirically, we have found that it is sufficient to use $P \approx {}^{1}\!/_{4} N_{\text{res}} \times N_{\text{vpl}}$ point samples for scenes with N_{vpl} VPLs (and an equivalent number of ISMs) and a modest depth complexity. Further, an ISM resolution of 128 × 128 or 256 × 256 is generally sufficient, see Figure 4.

6 Conclusion and Future Work

We have presented a method to compute real-time global illumination in large and fully dynamic scenes allowing for light, geometry, and material changes. The key ingredient to our instant radiosity simulation are imperfect shadow maps which can be used for fast visibility queries based on a coarse point-based representation of the scene. We have demonstrated that inaccurate visibility only has a minor impact on the result but leads to significant performance gains enabling real-time global illumination. In addition, we have shown that ISMs can also be used for direct illumination in certain cases, such as textured area lights or environment map illumination.

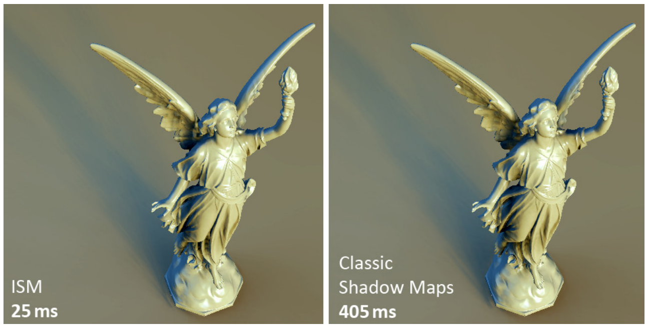

Figure 11: *Depth map computation time for direct environment map illumination from 1024 VPLs using a model of 525 k faces with classic and imperfect shadow maps (256 × 256 and 8K points each).*

In the future, we intend to reduce the number of required point samples and increase scalability, possibly by distributing point samples only to nearby ISMs, which is too costly on current GPUs. Furthermore, we want to explore clustering of similar VPLs [Walter et al. 2005] to further reduce computation time.

Acknowledgements

We would like to thank Marko Dabrovic for the Sponza model, MIT CSAIL for the animal animation sequences, Stanford University for the Lucy model, and the anonymous reviewers for their valuable feedback. Part of the research was funded by EPSRC grant EP/E047343/1.

References

ANNEN, T., DONG, Z., MERTENS, T., BEKAERT, P., SEIDEL, H.-P., AND KAUTZ, J. 2008. Real-Time, All-Frequency Shadows in Dynamic Scenes. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 27*, 3, 34:1–34:8.

ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2005. Fast and Detailed Approximate Global Illumination by Irradiance Decomposition. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 24*, 3, 1108–1114.

BRABEC, S., ANNEN, T., AND SEIDEL, H.-P. 2002. Shadow Mapping for Hemispherical and Omnidirectional Light Sources. In *Proc. of Computer Graphics International*, 397–408.

BUNNELL, M. 2005. Dynamic Ambient Occlusion and Indirect Lighting. In *GPU Gems 2*, M. Pharr, Ed. Addison Wesley, 223–233.

CHRISTENSEN, P., LAUR, D., FONG, J., WOOTEN, W., AND BATALI, D. 2003. Ray Differentials and Multiresolution Geometry Caching for Distribution Ray Tracing in Complex Scenes. In *Proc. of the Eurographics Symposium on Rendering*, 543–552.

COHEN-OR, D., CHRYSANTHOU, Y., SILVA, C., AND DURAND, F. 2003. A Survey of Visibility for Walkthrough Applications. *IEEE Transactions on Visualization and Computer Graphics 9*, 3, 412–431.

CROW, F. 1977. Shadow Algorithms for Computer Graphics. In *Proc. of ACM SIGGRAPH*, 242–248.

DACHSBACHER, C., AND STAMMINGER, M. 2005. Reflective Shadow Maps. In *Proc. of the Symposium on Interactive 3D Graphics and Games*, 203–213.

DACHSBACHER, C., AND STAMMINGER, M. 2006. Splatting Indirect Illumination. In *Proc. of the Symposium on Interactive 3D Graphics and Games*, 93–100.

DACHSBACHER, C., VOGELGSANG, C., AND STAMMINGER, M. 2003. Sequential Point Trees. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 22*, 3, 657–662.

DACHSBACHER, C., STAMMINGER, M., DRETTAKIS, G., AND DURAND, F. 2007. Implicit Visibility and Antiradiance for Interactive Global Illumination. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 26*, 3.

DONG, Z., KAUTZ, J., THEOBALT, C., AND SEIDEL, H.-P. 2007. Interactive Global Illumination Using Implicit Visibility. In *Pacific Graphics*, 77–86.

DUTRÉ, P., BALA, K., AND BEKAERT, P. 2006. *Advanced Global Illumination*. AK Peters.

GREEN, P., KAUTZ, J., AND DURAND, F. 2007. Efficient Reflectance and Visibility Approximations for Environment Map Rendering. *Computer Graphics Forum (Proc. of Eurographics) 26*, 3, 495–502.

GROSSMAN, J., AND DALLY, W. 1998. Point Sample Rendering. In *Proc. of the Eurographics Workshop on Rendering*, 181–192.

GUENNEBAUD, G., BARTHE, L., AND PAULIN, M. 2006. Real-time Soft Shadow Mapping by Backprojection. In *Eurographics Symposium on Rendering*, 227–234.

HOPPE, H. 1996. Progressive Meshes. In *SIGGRAPH '96*, 99–108.

IWASAKI, K., DOBASHI, Y., YOSHIMOTO, F., AND NISHITA, T. 2007. Precomputed Radiance Transfer for Dynamic Scenes Taking into Account Light Interreflection. In *Proc. of the Eurographics Symposium on Rendering*, 35–44.

KELLER, A. 1997. Instant Radiosity. In *SIGGRAPH '97*, 49–56.

KOZLOWSKI, O., AND KAUTZ, J. 2007. Is Accurate Occlusion of Glossy Reflections Necessary? In *Symposium on Applied Perception in Graphics and Visualization*, 91–98.

LAINE, S., SARANSAARI, H., KONTKANEN, J., LEHTINEN, J., AND AILA, T. 2007. Incremental Instant Radiosity for Real-Time Indirect Illumination. In *Proc. of the Eurographics Symposium on Rendering*, 277–286.

MARROQUIM, R., KRAUS, M., AND CAVALCANTI, P. R. 2007. Efficient Point-Based Rendering Using Image Reconstruction. In *Symposium on Point-Based Graphics 2007*, 189–203.

RUSHMEIER, H., PATTERSON, C., AND VEERASAMY, A. 1993. Geometric Simplification for Indirect Illumination Calculations. In *Graphics Interface*, 227–236.

SEGOVIA, B., IEHL, J.-C., MITANCHEY, R., AND PÉROCHE, B. 2006. Non-interleaved Deferred Shading of Interleaved Sample Patterns. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Graphics Hardware*, 53–60.

SILLION, F., AND DRETTAKIS, G. 1995. Feature-based Control of Visibility Error: A Multi-Resolution Clustering Algorithm for Global Illumination. In *SIGGRAPH '95*, 145–152.

SILLION, F. 1995. A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters. *IEEE Transactions on Visualization and Computer Graphics 1*, 3, 240–254.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 21*, 3, 527–536.

SLOAN, P.-P., GOVINDARAJU, N., NOWROUZEZAHRAI, D., AND SNYDER, J. 2007. Image-Based Proxy Accumulation for Real-Time Soft Global Illumination. In *Pacific Graphics*, 97–105.

STOKES, W. A., FERWERDA, J. A., WALTER, B., AND GREENBERG, D. P. 2004. Perceptual Illumination Components: A New Approach to Efficient, High Quality Global Illumination Rendering. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 23*, 3, 742–749.

TABELLION, E., AND LAMORLETTE, A. 2004. An Approximate Global Illumination System for Computer Generated Films. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 23*, 3, 469–476.

WALD, I., KOLLIG, T., BENTHIN, C., KELLER, A., AND SLUSALLEK, P. 2002. Interactive global illumination. In *Proc. of Eurographics Symposium on Rendering*, 9–20.

WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K., DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: A Scalable Approach to Illumination. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 24*, 3, 1098–1107.

WAND, M., FISCHER, M., PETER, I., AUF DER HEIDE, F. M., AND STRASSER, W. 2001. The Randomized z-Buffer Algorithm: Interactive Rendering of Highly Complex Scenes. In *SIGGRAPH '01*, 361–370.

WANG, R., ZHU, J.-J., AND HUMPHREYS, G. 2007. Precomputed Radiance Transfer for Real-time Indirect Lighting using A Spectral Mesh Basis. In *Eurographics Symposium on Rendering*, 13–21.

WILLIAMS, L. 1978. Casting Curved Shadows on Curved Surfaces. In *Computer Graphics (Proc. of SIGGRAPH '78)*, 270–274.

WYMAN, C. 2008. Hierarchical Caustic Maps. In *Proc. of the Symposium on Interactive 3D Graphics and Games*, 163–171.

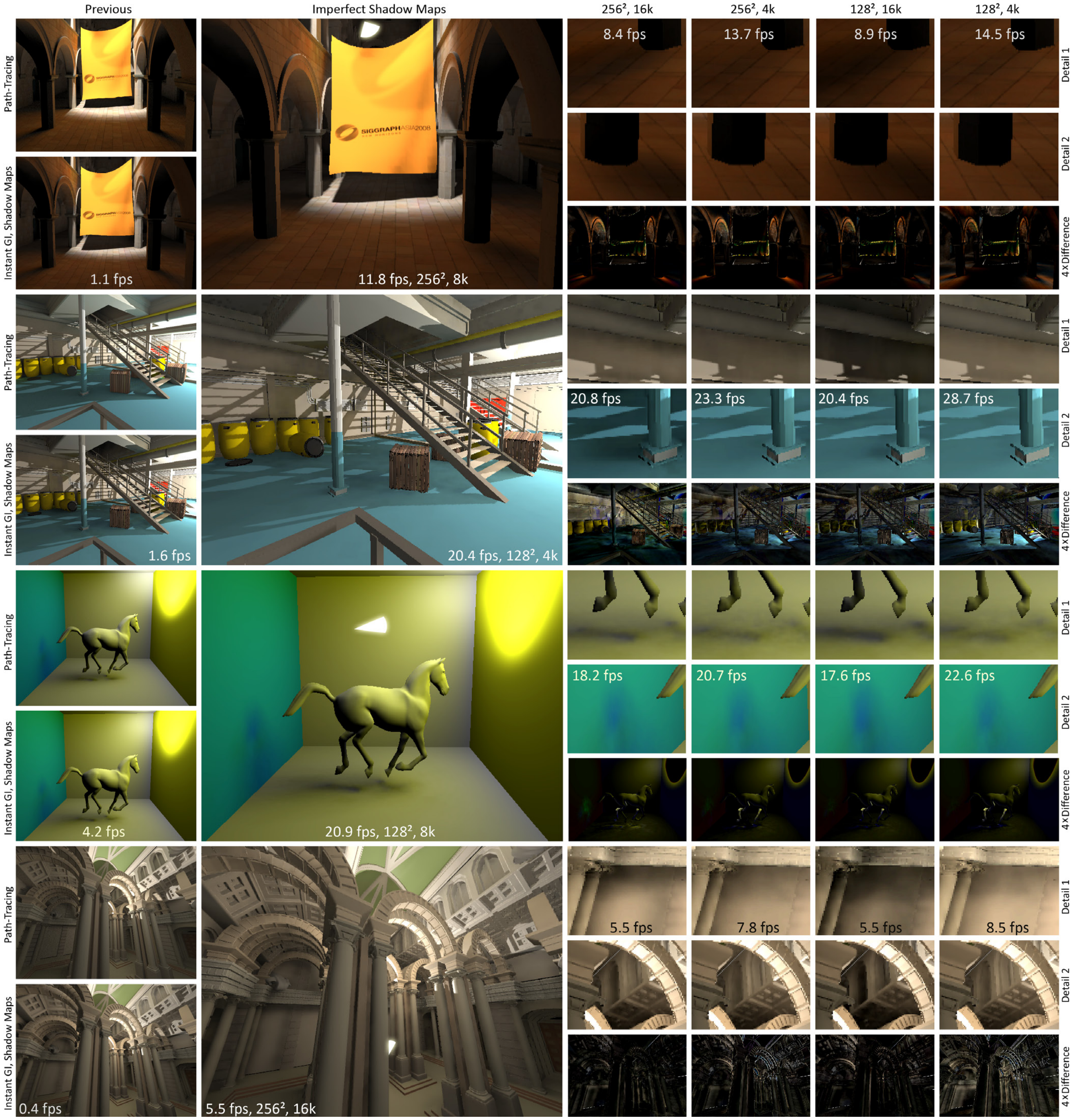

Figure 12: *Four different scenes rendered with our method. The first three scenes are rendered with single-bounce indirect illumination, whereas the last scene with two bounces. ISM resolution as well as the number of point samples are varied (resolution of* 256×256 *and* 128×128*; 16000 and 4000 point samples per VPL); all scenes use 1024 VPLs. We compare to a reference GPU solution (using classic* 256×256 *shadow maps) as well as a path-traced reference image. Cutouts of each scene and difference images (4 times magnified) w.r.t. the GPU reference are depicted for each parameter set. The first scene (Sponza, 70k faces) achieves good results for all settings, only the indirect shadows are a bit smoother overall. The industry scene (76k faces) requires larger ISMs and more points to faithfully reproduce shadows from thin objects, such as the columns. The horse scene (17k faces) is close to the reference for all settings. A small part of the leg is consistently too bright, which is due to shadow biasing. In the last scene (Palazzo, 380k faces), very faithful renderings can be achieved with the highest settings. Frame rates are generally lower due to the additional second indirect bounce (another 256 VPLs) and higher image resolution (*1280×960 *to prevent aliasing). Generally speaking, the resolution of the ISMs influences the penumbra size that can be resolved (e.g., see column in second scene). Not surprisingly, using more point samples increases the quality of the resulting indirect illumination but also decreases the overall frame rates. Overall, our results are very similar to the GPU reference renderings, but are rendered at about 10 times the frame rate.*