

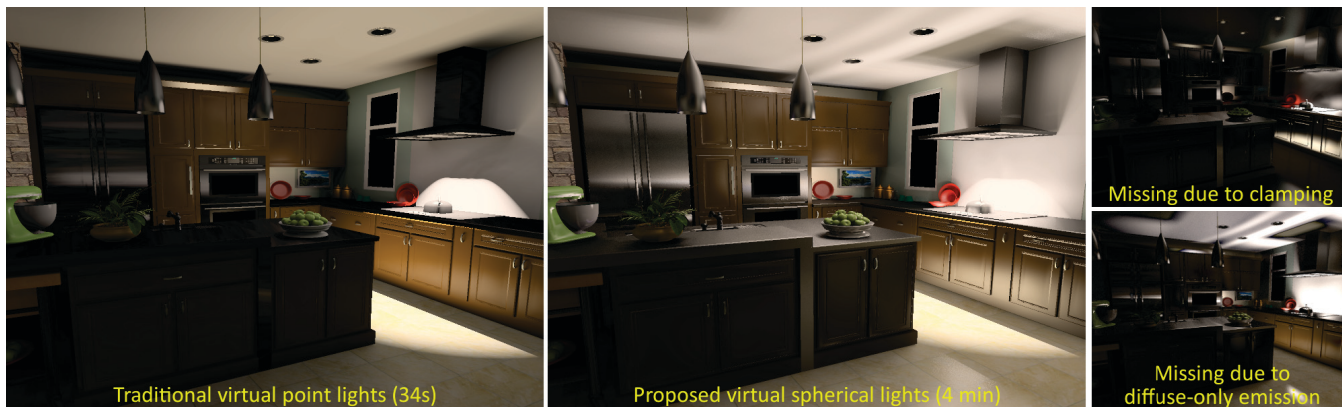
# Virtual Spherical Lights for Many-Light Rendering of Glossy Scenes

Miloš Hašan\*  
Cornell University

Jaroslav Křivánek  
Cornell University  
Charles University in Prague

Bruce Walter  
Cornell University

Kavita Bala  
Cornell University



**Figure 1:** Left: Many-light approaches to global illumination need to clamp the contributions of virtual point lights (VPLs). The resulting loss of glossy illumination may completely change the appearance of the scene. Right: We introduce virtual spherical lights (VSLs) that capture the illumination ignored by VPLs. Note the remarkably different appearance of the upper right part of the image, the steel counters and the floor. The images on the right illustrate the glossy effects that are lost due to the two approximations made in VPL rendering: clamping and diffuse-only emission. (Images are tone-mapped; original HDR images available for numerical comparisons.)

## Abstract

In this paper, we aim to lift the accuracy limitations of many-light algorithms by introducing a new light type, the *virtual spherical light* (VSL). The illumination contribution of a VSL is computed over a non-zero solid angle, thus eliminating the illumination spikes that virtual point lights used in traditional many-light methods are notorious for. The VSL enables application of many-light approaches in scenes with glossy materials and complex illumination that could previously be rendered only by much slower algorithms. By combining VSLs with the matrix row-column sampling algorithm, we achieve high-quality images in one to four minutes, even in scenes where path tracing or photon mapping take hours to converge.

**CR Categories:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture;

**Keywords:** many lights, glossy BRDF, global illumination

\*{mhasan,kb}@cs.cornell.edu, {jaroslav,bjw}@graphics.cornell.edu

## 1 Introduction

Fast computation of global illumination in complex scenes with many glossy surfaces is a long-standing unsolved problem of rendering research; indeed, no satisfactory solution exists. Pure Monte Carlo methods like path tracing, bidirectional path tracing or Metropolis [Veach 1997] take very long to converge. Photon mapping with final gathering [Jensen 2001] is generally much faster than pure Monte Carlo, but not so much in highly glossy scenes. Irradiance caching [Ward et al. 1988] or radiosity approaches [Cohen and Wallace 1993] are not applicable to non-diffuse materials, and their directional extensions [Křivánek et al. 2005; Christensen et al. 1997] can only handle moderate gloss.

Many-light methods [Keller 1997; Walter et al. 2005; Laine et al. 2007; Hašan et al. 2007; Ritschel et al. 2008] provide good performance, but are incorrect for two reasons. First, the generated virtual point lights (VPLs) are diffuse-only, so light bouncing off glossy surfaces is ignored. Second, the VPL contributions are clamped to prevent illumination spikes. These measures produce visually pleasing results in diffuse scenes, but they remove much of the interesting illumination in scenes with a substantial proportion of glossy materials; see Figure 1 for an example.

Our goal is to eliminate these limitations of many-light methods, thereby developing the first algorithm to render scenes with high amounts of glossy reflectance significantly faster than pure Monte Carlo techniques. Our key contribution is a new light type, the *virtual spherical light*, with the following desirable properties:

- The point-wise evaluations of traditional VPLs are replaced by integration over a non-zero solid angle, eliminating the spikes caused by narrow glossy BRDF lobes and the geometry term (see Figure 2(a)). Clamping is no longer needed, preserving illumination energy that would otherwise be lost.



(a) No clamping (34 sec) (b) Using glossy VPLs (34 sec) (c) Kollig & Keller (3.5 hours) (d) Photon mapping (1 hour)

**Figure 2:** *Alternative solutions to the problem are unsatisfactory. (a) Turning off clamping produces an unacceptable image. (b) Considering the full glossy BRDF at the VPL location does not help either, since even more clamping has to be applied to eliminate the spikes. (c) Compensating for the missing energy by the path tracing method of Kollig and Keller [2004] dominates the render time, and is still noisy after 3.5 hours on 8 CPU cores. (d) Photon mapping with final gathering converges slowly in glossy scenes with difficult illumination conditions; this result still has artifacts after 1 hour on 8 cores.*

- Since the VSL acts as a point light in visibility computations, fast GPU shadow mapping can be exploited. The VSL contribution depends only on values local to the surface point and light location, so its estimation becomes a purely numerical kernel ideally suited to evaluation in a GPU shader.

Our technique can prove useful in applications such as architectural lighting design by enabling the use of true material reflectance instead of diffuse approximations, and lead to new developments in the field of many-light rendering.

## 2 Problem Review and Related Work

**Many-Light methods.** Rendering algorithms based on the many-light formulation generate a number of virtual point lights (VPLs) to approximate indirect illumination, and shade visible surfaces from these lights. In the path space formulation of global illumination [Veach 1997], these methods create light subpaths of any length corresponding to the VPLs and camera subpaths of length one corresponding to the visible surface samples.

The computation of connections between the camera and light subpaths is where the many-light methods differ from each other. The original many-light technique, instant radiosity [Keller 1997], connects all lights to all surface samples, using shadow maps for visibility. Several improvements have been published recently. Laine et al. [2007] propose incremental instant radiosity for animated sequences, where only a subset of shadow maps is recomputed in each frame. Matrix row-column sampling [Hašan et al. 2007] samples a small number of rows and columns from the large matrix of light-surface connections. Lightcuts [Walter et al. 2005; Walter et al. 2006] use a hierarchy on the set of lights and/or surface samples to improve the scalability of connection computations. Ritschel et al. [2008] use approximate shadow maps to accelerate rendering.

**Problems with many-light methods.** If used naively, many-light formulations have a significant weakness – highly conspicuous artifacts in the form of illumination spikes will be produced in parts of the image that strongly depend on a single VPL; see Figure 2(a). Numerically, the problem is that when connecting the camera subpath to a VPL (i.e. a pre-generated light subpath), the BRDF and the geometry term can have a very large value compared to the probability density of generating the path. More specifically, connecting a surface sample at  $\mathbf{x}$  to a VPL at  $\mathbf{y}$  introduces the following terms:

$$f_r(\mathbf{x}_c \leftarrow \mathbf{x} \leftarrow \mathbf{y}) G(\mathbf{x} \leftrightarrow \mathbf{y}) f_r(\mathbf{x} \leftarrow \mathbf{y} \leftarrow \mathbf{y}') \dots \quad (1)$$

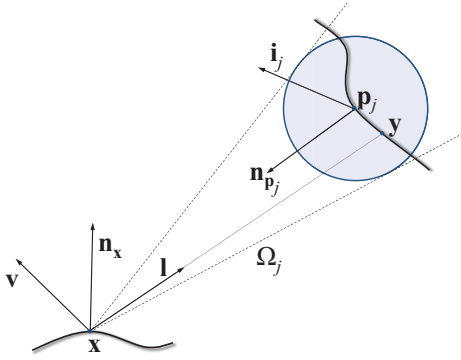
Here  $\mathbf{x}_c$  is the camera position. The geometry factor  $G(\mathbf{x} \leftrightarrow \mathbf{y})$  has the term  $\|\mathbf{x} - \mathbf{y}\|^2$  in the denominator, so when the VPL is very

close to the point being shaded, a spike will be produced. An even worse problem occurs when the scene contains glossy BRDFs, and the points  $\mathbf{x}$  and  $\mathbf{y}$  are aligned such that either of the two BRDF terms (or both) have a large value in that direction. While the former problem with the geometry term is localized (occurring mostly in corners), the latter can happen across any distance if the scene contains sufficiently glossy materials.

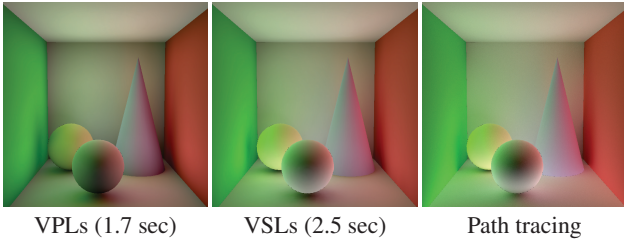
Bidirectional path tracing with multiple importance sampling does not exhibit this problem, because paths with a high BRDF or  $1/r^2$  value are usually generated with a correspondingly high probability density [Veach 1997]. This is not applicable in many-light rendering, so the problem is typically fixed by introducing two approximations: first, the contribution of any one VPL is clamped to a user-specified constant  $c$ , and second, the VPLs are assumed to be diffuse (i.e., only the diffuse part of the BRDF at the VPL location is considered). Clearly, these approximations will result in a loss of illumination. The effect of this energy loss is usually perceptually subtle in diffuse scenes, but can be very significant for glossy scenes; in fact, almost all indirect illumination in the scene can be missing. Glossy surfaces usually turn out nearly black in renderings due to clamping. Secondary highlights due to glossy reflectors are missing due to diffuse-only VPLs. Note that simply making the VPLs glossy does not help, since most of their contribution has to be clamped anyway to prevent the spikes; see Figure 2(b).

Kollig and Keller [2004] introduced an unbiased path tracing approach to compensate for the loss of energy in many-light rendering. Since the missing illumination in diffuse scenes is spatially localized, they gain performance by casting short rays. Unfortunately, the ray shortening idea does not work well in glossy scenes, because the missing illumination can be arbitrarily non-local. Furthermore, if the energy loss is significant, the compensation will essentially degenerate to standard path tracing and dominate running time. Suykens and Willems [Suykens and Willems 1999] augment radiosity solutions using bidirectional path tracing to compensate for the missing light paths, but their computation cost would also be dominant in highly glossy scenes.

Photon mapping (Figure 2(d)) is not an ideal solution for glossy scenes, since it requires either a large number of nearest neighbors for density estimation on glossy surfaces, or the continuation of paths that hit a glossy surface (but this happens all the time in a glossy environment, causing the algorithm to degenerate to recursive path tracing). Previous GPU-based photon mapping techniques [Purcell et al. 2003; Wang et al. 2009] either ignore final gathering or use irradiance caching, neither of which are appropriate for highly glossy scenes.



**Figure 3:** The virtual spherical light. A surface point  $\mathbf{x}$  with normal  $\mathbf{n}_x$  and viewing direction  $\mathbf{v}$  receives light from a VSL with location  $\mathbf{p}_j$ , normal  $\mathbf{n}_{\mathbf{p}_j}$ , and incoming vector  $\mathbf{i}_j$ . The contribution is computed as an integral of the product of BRDFs and cosine terms at the endpoints over the solid angle  $\Omega_j$  (see Equation (5)).



**Figure 4:** Our VSL approach (middle) is beneficial even in diffuse scenes, where the characteristic corner darkening from VPL methods (left) is greatly reduced. Only indirect illumination shown.

Finally, while VPLs are usually generated by particle tracing from the light, some techniques have been introduced to generate them using bidirectional or Metropolis sampling [Segovia et al. 2006; Fan et al. 2005]. These produce better VPL distributions but the fundamental limitations of many-light methods remain.

### 3 The Virtual Spherical Light

In a many-light method, we are faced with the key problem of computing the connection between a surface sample (an eye subpath) and a virtual light (a light subpath). Our insight is to replace the usual point sampling by integration over some non-zero domain. This domain might be a disk of a finite radius approximating a patch of surface at the virtual light’s location (a “point-to-disk form factor”); unfortunately, such a formulation has several problems. Estimating the integral by Monte Carlo sampling over the disk area leads again to the  $1/r^2$  singularity that we are trying to avoid. On the other hand, sampling the solid angle spanned by an oriented disk is non-trivial. Instead, we first assume that the light distributes its energy to all surfaces within radius  $r_j$  of its location, and this energy is then gathered at the surface point to be shaded. Second, we simplify this formulation for easier evaluation in a GPU shader, yielding the virtual spherical light (Figure 3).

#### 3.1 The Photon Light

Consider a virtual light that splats its energy to all surfaces within a sphere of radius  $r_j$  centered at  $\mathbf{p}_j$ , after which this energy is transported to surface sample  $\mathbf{x}$  via final gathering. This is similar to the contribution of a single photon to the image in reverse photon

mapping with final gathering [Havran et al. 2005], which is why we call it the *photon light*.

The splatted radiance leaving a point  $\mathbf{y}$  in direction  $\omega$  can be expressed as

$$L_j^{splat}(\mathbf{y}, \omega) = \frac{\Phi_j}{\pi r_j^2} f_r(\mathbf{y}, \mathbf{i}_j, \omega) (\|\mathbf{y} - \mathbf{p}_j\| < r_j) \quad (2)$$

where  $\Phi_j$  is the power of the light,  $\pi r_j^2$  is an estimate of the surface area within the sphere, and  $(\|\mathbf{y} - \mathbf{p}_j\| < r_j)$  is an indicator expression with the value of 1 if the condition is true and 0 otherwise.

To find the radiance leaving a surface point  $\mathbf{x}$  in viewing direction  $\mathbf{v}$  due to the illumination from the photon light, we apply an additional light bounce to the above radiance function:

$$L_j^p(\mathbf{x}, \mathbf{v}) = \int_{H^2} f_r(\mathbf{x}, \mathbf{l}, \mathbf{v}) \cos(\mathbf{n}_x, \mathbf{l}) L_j^{splat}(\mathbf{y}, -\mathbf{l}) d\mathbf{l}, \quad (3)$$

where  $H^2$  is the hemisphere above  $\mathbf{x}$ ,  $\mathbf{y}$  is the point visible from  $\mathbf{x}$  in direction  $\mathbf{l}$ ,  $\mathbf{n}_x$  is the normal at  $\mathbf{x}$ , and  $\cos(\mathbf{a}, \mathbf{b}) = \max\{\langle \mathbf{a}, \mathbf{b} \rangle, 0\}$ . By substituting Equation (2) into Equation (3), we obtain the full definition of the photon light:

$$L_j^p(\mathbf{x}, \mathbf{v}) = \frac{\Phi_j}{\pi r_j^2} \int_{\Omega_j} f_r(\mathbf{x}, \mathbf{l}, \mathbf{v}) \cos(\mathbf{n}_x, \mathbf{l}) f_r(\mathbf{y}, \mathbf{i}_j, -\mathbf{l}) (\|\mathbf{y} - \mathbf{p}_j\| < r_j) d\mathbf{l}, \quad (4)$$

We replaced the integration domain  $H^2$  by  $\Omega_j$ , the cone of directions from  $\mathbf{x}$  to the sphere of radius  $r_j$  centered at the photon location  $\mathbf{p}_j$ , since the photon’s contribution is zero outside this cone. (If  $\mathbf{x}$  is inside the sphere, we take  $\Omega_j = H^2$ .) The indicator function  $(\|\mathbf{y} - \mathbf{p}_j\| < r_j)$  has to remain in the integrand, because a ray sent into  $\Omega_j$  can still end up outside the sphere.

The contribution of the photon light to a given surface point could, in theory, be computed by sampling the cone of directions  $\Omega_j$  with rays. However, this is impractical – ray-tracing is unavoidable, and many rays intersect surfaces outside of the sphere and have zero contribution. Therefore, we introduce several approximations to the photon light, resulting in the virtual spherical light.

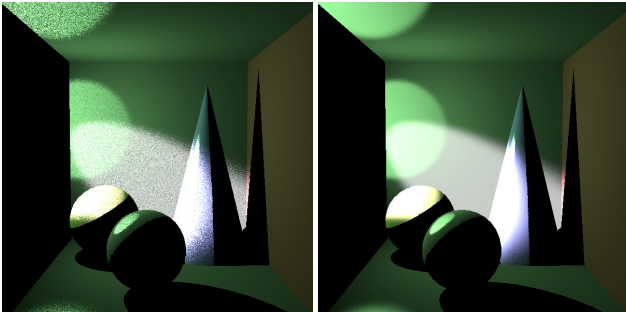
#### 3.2 The VSL approximation

We propose to compute the contribution of a VSL to a surface sample as

$$L_j^s(\mathbf{x}, \mathbf{v}) = \frac{\Phi_j}{\pi r_j^2} V(\mathbf{x}, \mathbf{p}_j) \int_{\Omega_j} f_r(\mathbf{x}, \mathbf{l}, \mathbf{v}) \cos(\mathbf{n}_x, \mathbf{l}) f_r(\mathbf{p}_j, \mathbf{i}_j, -\mathbf{l}) \cos(\mathbf{n}_{\mathbf{p}_j}, -\mathbf{l}) d\mathbf{l}. \quad (5)$$

This definition is obtained from the contribution of a photon light (Equation (4)) by introducing the following approximations:

- The visibility of the VSL is defined to be the visibility between the sample position  $\mathbf{x}$  and the photon location  $\mathbf{p}_j$ . This has the advantage that fast GPU shadow mapping can be used to compute the visibility.
- The normal vector and BRDF for all surface points inside the sphere are approximated by the normal and BRDF at the photon hit location  $\mathbf{p}_j$ . Therefore, all BRDF and cosine computations become local – they only depend on the properties of surface points  $\mathbf{x}$  and  $\mathbf{p}_j$ .
- The indicator function  $(\|\mathbf{y} - \mathbf{p}_j\| < r_j)$  is replaced by an additional cosine factor  $\cos(\mathbf{n}_{\mathbf{p}_j}, -\mathbf{l})$ . This factor corrects for rays that intersect the sphere but not the surface in it.



Uniform sampling of cone  $\Omega_j$  MIS (cone + BRDF sampling)

**Figure 5:** Uniform sampling of  $\Omega_j$  may produce a noisy estimate of the VSL integral in glossy scenes (left). Using Multiple Importance Sampling to combine uniform sampling of  $\Omega_j$  with BRDF importance sampling reduces the noise (right). In this example, a single VSL with relatively large radius is located on the left wall of the box. All surfaces have a glossy Ward BRDF component with  $\alpha = 0.03$ .

Our definition has several desirable properties. First, all computations depend only on information local to points  $\mathbf{x}$  and  $\mathbf{p}_j$ , and the visibility between them. Shadow mapping can be used to compute the visibility, and the integral in Equation (5) can be approximated in a GPU shader. Second, there is no  $1/r^2$  falloff term, and therefore no singularities when the light is close to the surface. No clamping is necessary, and corner darkening (so typical for many-light methods) is largely reduced in our results; see Figure 4. Spikes from glossy lobes similarly do not have to be clamped, because they are smoothed over a non-zero integration domain.

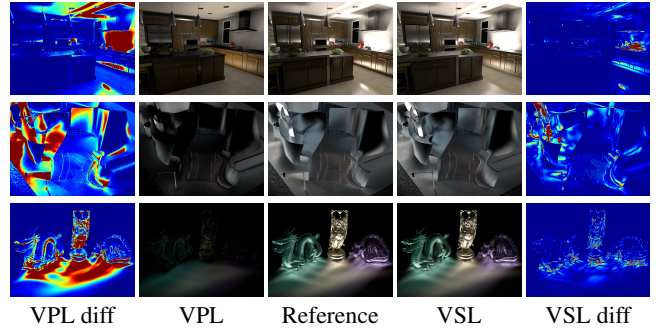
### 3.3 Implementation

To compute the contribution of a VSL to a surface point, we need to estimate the integral in Equation (5). A natural strategy would be to use uniform, stratified Monte Carlo integration of the cone of directions  $\Omega_j$ . This often works well, but if the surface point is close to the VSL and either one of the BRDFs is specular, then it might be beneficial to importance-sample the BRDFs in the integrand. We use *multiple importance sampling* [Veach 1997] to improve quality in this case (see Figure 5). We generate samples using three strategies: uniform sampling of the cone, importance sampling of the BRDF on the surface sample, and importance sampling of the BRDF on the VSL. These samples are combined using the balance heuristic.

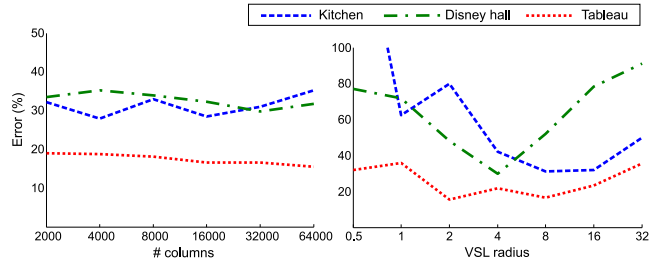
We use matrix row-column sampling [Hařan et al. 2007] to achieve scalability with large numbers of lights. The VSL is implemented simply as a new GPU shader. Including glossy transport in the matrix increases its numerical rank, so we used more rows (900–1,024) and columns (5,000–15,000) than when using VPLs.

We choose the VSL radii proportional to the local density of the lights. More specifically, for each VSL, we search for 10 nearest neighbors, and we set the VSL radius to the resulting search radius multiplied by a user-specified constant; we found values of 4–12 to work best in our scenes. However, it would be desirable to automate the radius selection completely, possibly based on the probability density of generating the light paths, or the screen space footprint of the VSLs.

The number of samples for the VSL integral estimation is proportional to the size of the solid angle  $|\Omega_j|$ , with a maximum of up to 100 samples for  $|\Omega_j| = 2\pi$ . In practice, this strategy means that many VSL contributions are estimated with only one sample.



**Figure 7:** Illustration of the error of the traditional VPLs and our proposed VSLs as compared to the path-traced reference. All of the images show only indirect illumination. The difference is computed on the original HDR images (i.e. in radiance space).



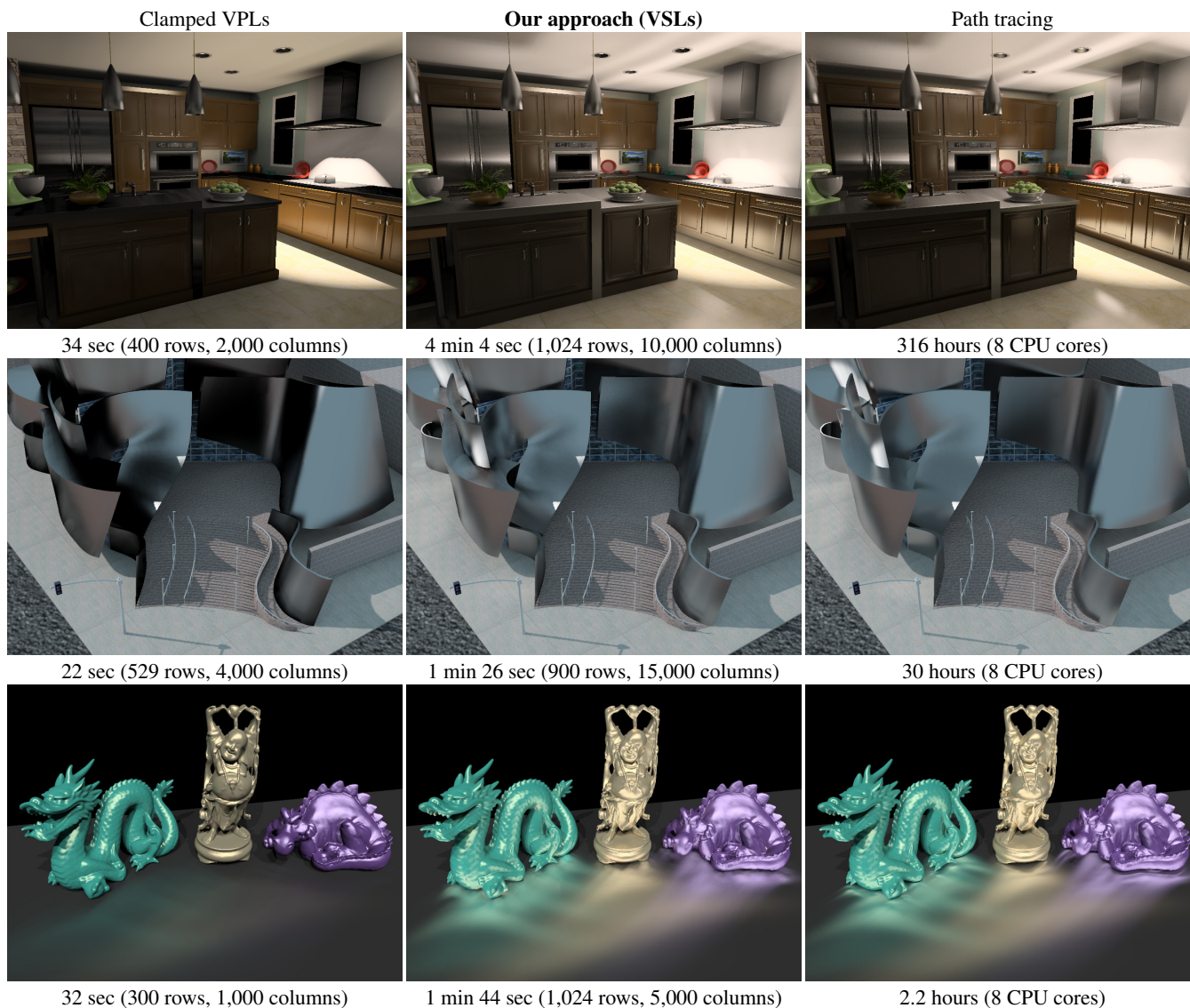
**Figure 8:** These plots show how the L2 error (measured on indirect illumination only) depends on the number of columns used in matrix row-column sampling (left) and the radius multiplier (right, see Section 3.3). On the left we set the radius multiplier to 8 (4 for Disney Hall), while on the right we fixed the number of columns to 32,000.

## 4 Results and Discussion

We ran our results on a system with a quad-core Intel i7 processor, 6GB memory, and an NVIDIA 280 GTX GPU. The ray-traced results were run on 16 cluster nodes, each with 8 cores, and we report the cumulative times for one such node. We show three scenes, which are illustrated in Figure 6. For all of these scenes, we compute our result (i.e., matrix row-column sampling with VSLs), a clamped VPL solution (using standard row-column sampling) and a pure Monte Carlo path tracing reference. All of the techniques were only applied to computation of indirect illumination. Direct illumination was computed separately by shadow mapping to prevent subtle differences that are irrelevant to this paper. The timings do not include direct illumination, KD-tree construction and deep frame-buffer creation, all of which are shared by all methods. All images are  $800 \times 600$  with  $2 \times 2$  super-sampling.

Our final results are tone-mapped using the global operator of Reinhard et al. [2002], where the same curve has been applied to VSL, VPL and reference images. The indirect-only images in Figures 4, 5 and 7 and the missing components in Figure 1 use the sRGB model, which gives near-linear mapping to screen radiance. Original HDR images are available as supplemental material for numerical comparisons.

**Kitchen.** This scene (515,375 polygons) is lit mostly by indirect illumination, with the only direct lights above the stove. Most materials have a glossy component, and many are anisotropic. The metallic surfaces of the counters and fridge look significantly darker



**Figure 6:** Left: Results generated with standard diffuse-only VPLs and clamping look clean, but are missing significant amounts of illumination. Middle: Our method preserves most of the glossy illumination effects. Right: Reference images, with indirect illumination computed by pure Monte Carlo path tracing.

in the VPL solution due to clamping. The metallic stove right under the direct lights reflects most of the light energy back into the scene through its glossy BRDF, and creates a distinct stripe effect on the ceiling, which cannot be captured using diffuse VPLs. Furthermore, algorithms like path tracing and photon mapping take a long time to converge, because they keep missing paths with strong contribution. In fact, our reference solution took 316 hours to render on 8 cores (cumulative from 16 cluster machines). Metropolis or bidirectional tracing would probably also take many hours. We are not aware of an algorithm other than ours that could produce a comparable result in a few minutes.

**Disney Hall.** This outdoor scene (38,035 polygons) is lit by a sun-sky model converted to directional lights. The main material of the building has no diffuse component, so the VPL solution cannot capture any light reflection from the building whatsoever; furthermore, it strongly clamps illumination reflected from the ground and received by the building. Our method preserves the appearance of the

glossy surfaces much better, although it blurs some lighting features compared to the path-traced reference.

**Tableau.** The tableau scene (797,641 polygons) is composed of glossy objects on an anisotropic Ward BRDF plane. This is a rather difficult stress test for our technique. The scene is lit by three distant lights, the direct illumination of which is barely visible due to the very small diffuse component on the plane. While the traditional VPL approach cannot capture almost any inter-reflection between the objects and the floor, our VSL does quite well, only slightly blurring compared to the path-traced compensation.

**Error and discussion.** A fundamental limitation is that our method is biased. It is consistent, in the sense that it converges to the correct solution as the number of VSLs increases and their radius goes to zero. Unfortunately, in our current implementation the number of VSLs cannot be increased much beyond 250,000 due to memory consumption of the reduced matrix used in row-column sampling.

Finding a more scalable algorithm that could handle many millions of VSLs is an important direction for future work.

Figure 7 shows indirect-only images produced with VPLs and VSLs, and color-coded difference images illustrating the deviation from the path-traced reference. The difference is computed on the original HDR images. The normalized  $L_2$  error in our results is 27.7%, 34.3% and 17.5% for the Kitchen, Disney Hall and Tableau scene, respectively. This is much reduced compared to VPLs, which incur an  $L_2$  error of 94.6%, 96.7% and 97.4% respectively. Figure 8 shows the dependence of the error on the radius of the VSLs and the number of columns used in row-column sampling. The graphs suggest that tweaking these parameters can only provide limited improvement, and further decrease in error can be achieved only by significantly increasing the total number of VSLs.

Two other limitations of our formulation require future work. First, choosing the number of samples based on the solid angle  $\Omega_j$  is good for uniform cone sampling, but not necessarily for sampling the two BRDFs (Section 3.3); here one would prefer to base the number of samples on the intersection between the BRDF lobe and the cone. Second, the cosine term in the VSL definition that approximates the indicator expression of the photon light is incorrect for points close to the light, and is responsible for some remaining corner darkening (Figure 4(b)).

**Cost of VSLs.** Computing the contribution of a VSL is more computationally expensive than with VPLs. However, in complex scenes the computation is dominated by shadow map rendering, which is common to VPLs and VSLs; for example, the slowdown due to computing the VSL integral in the Kitchen scene is only about 16%. The rest of the slowdown is because of the higher number of rows and columns required for VSL rendering.

## 5 Conclusion

We have introduced the *virtual spherical light*, which addresses fundamental limitations in many-light rendering of glossy scenes: the loss of energy due to clamping and the use of diffuse VPLs. We have shown that rendering with VSLs produces high-quality images in minutes, even in relatively difficult lighting scenarios where current approaches are either incorrect, or converge very slowly (taking hours). We believe our technique takes an important step towards solving the difficult problem of glossy inter-reflections, and will stimulate new developments in the field of many-light rendering.

## Acknowledgments

We thank Edgar Velázquez-Armendáriz and Fabio Pellacini for implementing important parts of our code base. This work was supported by NSF CAREER 0644175, NSF CPA 0811680, NSF CNS 0403340, and grants from Intel Corporation and Microsoft Corporation. Miloš Hašan was partly funded by the NVIDIA Fellowship. Jaroslav Krivánek was supported by the European Commission Marie Curie Fellowship PEOF-GA-2008-221716. The kitchen model was originally developed by BHI Media for Autodesk, Inc. as part of Autodesk Labs Project Showroom technology preview, and is used with permission from Autodesk, Inc.

## References

- CHRISTENSEN, P. H., LISCHINSKI, D., STOLLNITZ, E. J., AND SALESIN, D. H. 1997. Clustering for glossy global illumination. *ACM Trans. Graph.* 16, 1, 2–33.
- COHEN, M. F., AND WALLACE, J. R. 1993. *Radiosity and Realistic Image Synthesis*. Morgan Kaufmann, San Francisco, CA.
- FAN, S., CHENNEY, S., AND CHI LAI, Y. 2005. Metropolis photon sampling with optional user guidance. In *Rendering Techniques '05 (Proceedings of the 16th Eurographics Symposium on Rendering)*, 127–138.
- HAVRAN, V., HERZOG, R., AND SEIDEL, H.-P. 2005. Fast final gathering via reverse photon mapping. *Computer Graphics Forum* 24, 3, 323–333.
- HAŠAN, M., PELLACINI, F., AND BALA, K. 2007. Matrix row-column sampling for the many-light problem. *ACM Trans. Graph.* 26, 3, 26.
- JENSEN, H. W. 2001. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA.
- KELLER, A. 1997. Instant radiosity. In *Proceedings of SIGGRAPH 97*, 49–56.
- KOLLIG, T., AND KELLER, A. 2004. Illumination in the presence of weak singularities. In *Monte Carlo And Quasi-monte Carlo Methods*, 245–257.
- KŘIVÁNEK, J., GAUTRON, P., PATTANAİK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination. In *IEEE Transactions on Visualization and Computer Graphics*, 550–561.
- LAINÉ, S., SARANSAARI, H., KONTKANEN, J., LEHTINEN, J., AND AILA, T. 2007. Incremental instant radiosity for real-time indirect illumination. In *Proceedings of Eurographics Symposium on Rendering*, 277–286.
- PURCELL, T. J., DONNER, C., CAMMARANO, M., JENSEN, H. W., AND HANRAHAN, P. 2003. Photon mapping on programmable graphics hardware. In *Graphics Hardware*, 41–50.
- REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Trans. Graph.* 21, 3, 267–276.
- RITSCHEL, T., GROSCH, T., KIM, M. H., SEIDEL, H.-P., DACHSBACHER, C., AND KAUTZ, J. 2008. Imperfect Shadow Maps for Efficient Computation of Indirect Illumination. *ACM Trans. Graph.* 27, 5, 129.
- SEGOVIA, B., IEHL, J.-C., AND PÉROCHE, B. 2006. Bidirectional instant radiosity. In *Proceedings of Eurographics Symposium on Rendering*, 389–398.
- SUYKENS, F., AND WILLEMS, Y. D. 1999. Weighted multipass methods for global illumination. In *Computer Graphics Forum*, 209–220.
- VEACH, E. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University.
- WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K., DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: a scalable approach to illumination. *ACM Trans. Graph.* 24, 3, 1098–1107.
- WALTER, B., ARBREE, A., BALA, K., AND GREENBERG, D. P. 2006. Multidimensional lightcuts. *ACM Trans. Graph.* 25, 3, 1081–1088.
- WANG, R., WANG, R., ZHOU, K., PAN, M., AND BAO, H. 2009. An efficient gpu-based approach for interactive global illumination. *ACM Trans. Graph.* 28, 3, 91.
- WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In *Proceedings of ACM SIGGRAPH 88*, 85–92.