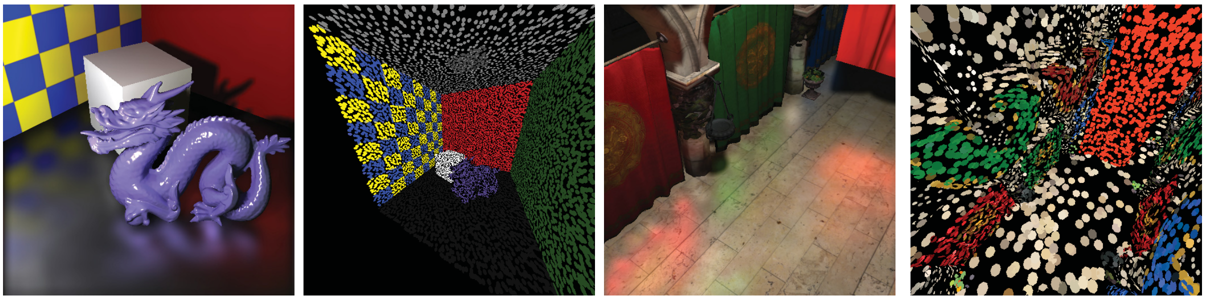
Pacific Graphics (2014)
J. Keyser, Y. J. Kim, and P. Wonka (Editors)

Visibility Filtering for Producing Indirect Illumination

Yu-Jung Chen[1] Chen-Yu Yen[1] Yen-Yu Chen[1] Wei-Chao Chen[2] Shao-Yi Chien[1]

[1]Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University
[2]Skywatch Inc.



(a) *Cornell Box* (b) 24k Point Proxies (c) *Crytek Sponza* (d) 170k Point Proxies

Figure 1: Fig. 1a is the *Cornell Box* and Fig. 1c is the *Crytek Sponza* scene. Both scenes are renderd with a glossy floor ($N_s = 50$) at resolution 512×512. Fig. 1b and Fig. 1d are the corresponding point proxies, rendered as discs to improve the quality of visualization.

Abstract

Indirect illumination is one of the most visually significant effects for a synthesized image. In order to render such effects at interactive rates, it is important to resolve the visibility between surfaces efficiently. In this work, we propose to approximate visibility by representing the scene through point proxies and hash them into voxels. The indirect illumination is then rendered in two stages. First, we cast sparse shadow rays to march and collect visible voxels for each low-resolution deferred pixel. The list of visible voxels are then processed using the proposed Visibility Filter (VF) to collect voxels missed from the first stage. Afterward, we properly weight the visible voxels as light sources to produce the full resolution image. Our method runs at interactive rates at 1.3 fps to 2.3 fps on GPUs with pleasing visual quality, in particular for glossy reflections.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

In the light transport process, the first indirect illumination bounce delivers the significant visual effects due to the amount of energy it contributes to the image. As such, there have been many interactive rendering algorithms that are designed to approximate this visual quality efficiently. Among these, the Reflectitve Shadow Map (RSM) [DS05] technique stands out as an important technique that has inspired many related research. Using the GPU rasterization pipeline, RSM directly distribute energy from the light source to the primary surfaces as texels. These texels can then be treated as small, quadrilateral Virtual Point Lights (VPLs) to illuminate the scene. Obviously, for each of the VPLs, we need to resolve its visibility against the surfaces it illuminates, and this can become very expensive due to the sheer number of VPLs in the scene. Of course, we may simply skip the visibility computation, but then otherwise invisible VPLs would

© The Eurographics Association 2014.

DOI: 10.2312/pgs.20141245

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

be incorrectly projected to surfaces, and both rendering performance and visual fidelity suffer as a result.

For scenes that are primarily diffuse or semi-glossy, however, it is possible to approximate visibility of indirect light sources with little loss in visual fidelity. This provides a good balance between performance and fidelity, as demonstrated in [RGK*08, REG*09, KD10, CNS*11]. We follow a similar route where the scene is represented using point proxies grouped as VPLs in a grid-like structure, and have the surface point query this structure to cull away parts of the VPLs that are not visible to it. This has several advantages. First, since the point proxies in the grid are relatively coarse, we may resolve the visibility at a lower resolution, and this leads to better performance. Also, instead of completely marching the cubes, we propose a Visibility Filter (VF) that improves the sparsely estimated visibility. Finally, when we render the full resolution image, the visibility computation is bilinearly interpolated with considering geometric constraints to give a smoother transition between neighboring pixels, thus producing pleasing visual results, with little rendering biases, at interactive rendering rates.

2. Related Works

In terms of light source visibility, imperfect shadow maps (ISM) [RGK*08] improves RSM based indirect illumination by approximating visibility between the scene and Virtual Point Lights (VPLs) by using low-resolution shadow maps. Micro-rendering [REG*09] approximates visibility through rasterization of the micro framebuffers at points in the scene. For the low-resolution deferred pixels, their indirect illuminations are approximated using the micro framebuffers. Voxel cone tracing [CNS*11] approximates visiblility by using ray cones to intersect voxels to gather indirect energy.

It is also crucial to represent the light source energy efficiently for the sake of rendering performances. It is common to use cells or voxels [CNS*11], for example. In the light propagation volume (LPV) [KD10, BCK*11] technique, the energy is represented through low-order Spherical Harmonic (SH) decomposition. For many-lights problem [Kel97], their indirect contributions to the scene can be approximated by grouping these virtual light sources into clusters [WFA*05, OP11]. Besides, K-means has been used as an alternative solution as well [PKD12].

3. Proposed Techniques

Fig. 2 illustrates our proposed techniques. For the purpose of ray marching, common spatial data structures, such as Bounding Volume Hierachy (BVH) and KD-tree, are fairly demanding during the building of the structure. Since our main purpose is to approximate visibility for the purpose of evaluating corresponding indirect energy, we relax on the intersection accuracy, and sample the scene using point proxies. Each point proxy represents a disc whose size and the orientation is computed using the surface normal and sampling density on the original geometry. To efficiently query these point proxies, they are placed in a voxelized cells through hashing, and each cell contains associated point proxies that represents part of the geometry it covers.

After the point proxies are created, we treat them as VPLs to store energy from direct illumination, and they can be then used as high quality indirect illumination sources. When the VPLs are farther away from the illuminated surfaces, however, they may contribute very little to the surfaces, and it is often sufficient and more efficient to approximate these VPLs through low-order spherical harmonic decomposition [KD10, BCK*11].

At a lower resolution screen position, we find its location in the scene, and from there we cast sparse shadow rays into the cell structure. This produces a list of the intersected voxels, one for each shadow ray. This method can miss visible voxels due to insufficient sampling rate, and for this we propose a Visibility Filter (VF) to locally exchange and combine the voxel cell list while respecting geometric constraint. The indirect illumination results are then produced on a higher resolution through properly weighting of the visible voxel lists before being combined with direct illumination.

3.1. Point Proxies Voxelization and Sparse Ray Marching

To coarsely represent the scene with point proxies, the fundamental assumption is to uniformly place point samples. The number of point proxies is proportional to triangle area. Based on the sampling density, each point proxy has a corresponding disc radius to approximate the geometry. We adopt Halton sequence as the barycentric coefficients for interpolating point proxies within a selected triangle. While applying the uniform sampling strategy, care must be taken so as not to bias toward larger triangles, as a smaller triangle may individually generate less than than one point sample. We solve this by sampling in a Russian Roulette fashion. For each primitive, a position-based hash is used to generate a pseudo random number. Then, we compared this number against a threshold that is proportional to its area, and produce a sample when the number exceeds the threshold. After generating the point proxies, the scene depth is rasterized from the light perspective. With the rasterized scene depth, visible point proxies can be illuminated as VPLs as Fig. 2 (a) shows. After the energy distribution, we compute a voxel index for each point proxy based on its position, and linearly hash the point proxies according to these indices. We use the following hash function

$$H(v) = v.z \cdot D_x \cdot D_y + v.y \cdot D_x + v.x, \tag{1}$$

where v is a three dimensional voxel index, and (D_x, D_y) represent the sizes of the voxel grid. The point proxies are then sorted by the hash keys for indexing future queries.

© The Eurographics Association 2014.

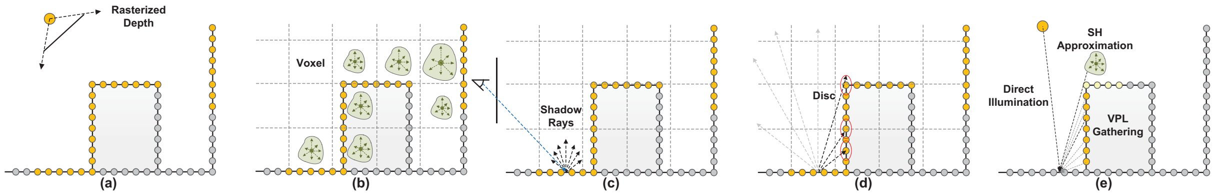


Figure 2: Illustration of the proposed techniques. (a). Determine visible point proxies from primary light source and distribute energy. (b). Hash point proxies into voxels and approximate voxel energy with SH. (c). Sparsely cast shadow rays for each deferred pixel. (d). Shadow rays march voxels and perform intersection test. (e). Evaluate indirect energy with VPLs and SH approximation.

Meanwhile, the voxel energy is approximated with low-order Spherical Harmonic (SH) expansion. For each VPL, we project the energy onto the SH basis of its located voxel in parallel with atomic operations as Fig. 2 (b) illustrates. The voxel energy I_v is then represented as follows

$$\sum_{p\in v} I_p = \sum_{p\in v}\sum_{l=0}^{\infty}\sum_{m=-l}^{l} c_{lm}y_{lm} \approx \sum_{p\in v}\sum_{l=0}^{1}\sum_{m=-l}^{l} c_{lm}y_{lm}, \qquad (2)$$

$$c_{lm} = \int_{4\pi} I_p y_{lm}(\omega)d\omega, \qquad (3)$$

where p is the set of point proxies within voxel v, I_p is the VPL energy, y_{lm} is the SH basis, and c_{lm} is the projected coefficients. Since the SH is adopted for approximating distant energy, we only use the order from l = 0 to 1.

Before rendering, we first examine the visibility with sparse ray marching for each pixel of a low-resolution G-buffer as Fig. 2 (c) illustrates. Different from using ray cones [CNS*11] for gathering indirect energy, we cast sparse shadow rays to approximately evaluate visible voxels. When a shadow ray marches to a voxel, the ray looks up the sorted point proxies, and the ray-disc intersection test is performed based on its spanned radius. If a shadow ray intersected with a point proxy, the marching process is terminated for that ray, and the voxel is claimed intersected as Fig. 2 (d) shows. Each deferred pixel keeps a record of the intersected voxel list for shadow rays, V_p, as its visibility information.

3.2. Visibility Filter

The sparse visibility test might cause discontinuity on the visibility records of the deferred pixels. Since a voxel stores many VPLs, even a missed visible voxel can cause discontinuous estimation on indirect energy. To produce pleasing visual quality, we propose a Visibility Filter to filter the visibility records in advance. Fig. 3 briefly illustrates an example with the filter size of 3 × 3. Each deferred pixel queries visibility lists from the neighbors as the following equation with considering geometric constraints including depth and normal,

$$V_p'(x,y) = \bigcup_{j=-m}^{m}\bigcup_{i=-n}^{n}\{V_p(x+i,y+j)|\delta_d\cdot\delta_n = 1\}, \qquad (4)$$

where the filter size is $(2m+1)\times(2n+1)$; V_p' is the filtered visibility list of the pixel; V_p is the visibility list of a pixel; δ_d and δ_n are binary decision functions which take depth and normal as geometric constraints. The binary decision function δ_d is 1 when the depth difference of the pair of deferred pixels $(x+i,y+j)$ and (x,y) is below a certain threshold. Similarly, the normal similarity δ_n is 1 when the dot product between a pair of deferred pixels is above a predefined threshold. The neighboring deferred pixels are considered to be on the same continuous surface when these two geometric decisions are satisfied. As Fig. 3 illustrates, the visibility lists of grey pixels are not used for union. The smooth approximation on visibility can provide better fidelity.

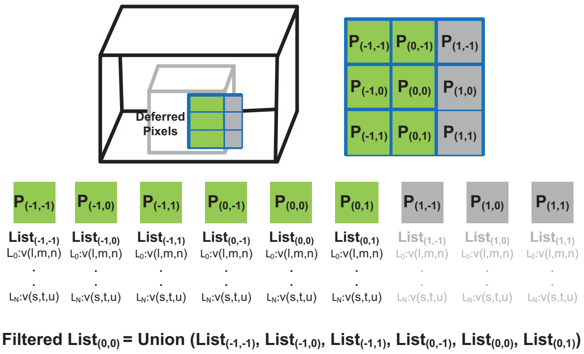

Deferred Pixels
P(-1,-1) P(0,-1) P(1,-1)
P(-1,0) P(0,0) P(1,0)
P(-1,1) P(0,1) P(1,1)
P(-1,-1) P(-1,0) P(-1,1) P(0,-1) P(0,0) P(0,1) P(1,-1) P(1,0) P(1,1)
List(-1,-1) List(-1,0) List(-1,1) List(0,-1) List(0,0) List(0,1) List(1,-1) List(1,0) List(1,1)
L0:v(l,m,n)
LN:v(s,t,u)
Filtered List(0,0) = Union (List(-1,-1), List(-1,0), List(-1,1), List(0,-1), List(0,0), List(0,1))

Figure 3: The proposed Visibility Filter.

3.3. Indirect Energy Gathering

With the filtered low-resolution visibility lists, indirect illumination is produced in higher resolution, where the high-resolution pixels utilize the neighboring visibility lists to look-up voxels as effective indirect contribution. Since the energy contribution from a voxel is discretized, we properly weight the contribution from the lists as shown in

© The Eurographics Association 2014.

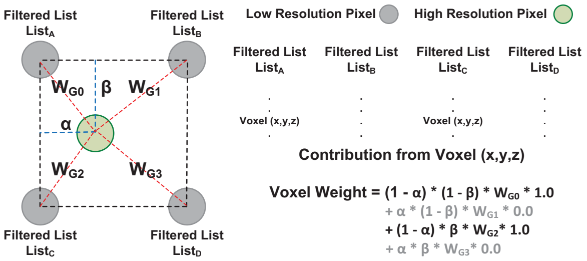


Figure 4: Indirect energy gathering in high resolution.

Fig. 4, where the upsampled high-resolution pixel looks-up the visibility lists from the four neighbors. As illustrated, $Voxel(x,y,z)$ is visible by pixel A and C. The voxel contribution is linearly weighted, the geometric constraint is further taken into consideration:

$$I_{indir} = \frac{1}{N_c} \sum_{p' \in Neighbor} w_G(V'_{p'}) \cdot w_l(V'_{p'}) \cdot L(V'_{p'}), \qquad (5)$$

where I_{indir} is the indirect energy contribution, w_G and w_l are the geometric and linear weights respectively. N_c is the normalization constant which represents the effective number of VPLs. The geometric weight w_G is a joint bilateral weight using the depth difference and normal similarity under the Gaussian distribution. L is the illumination function, which accumulates the energy of visible voxels v_l within a list, $L(V_p) = \sum_{v_l \in V_p} f(v_l)$, where f is the measurement function. To efficiently approximate the indirect energy, the energy of distant voxels are gathered through SH approximation; otherwise, point proxies are utilized as VPLs for detailed illumination as shown in the following equation as well as Fig. 2(e):

$$f(v_l) = \begin{cases} \sum_{x_i \in v_l} f_{vpl}(x_i, p) & dist(v_l, p) < R \\ f_{SH}(I_{v_l}, p) & otherwise \end{cases}, \qquad (6)$$

where x_i is the VPL, p is the current pixel, and R is the effective range; f_{vpl} and f_{SH} are the measurement functions based on VPL and SH approximation respectively. We adopt the local Phong reflection model as the measurement function. The position of voxel v_l is approximated with the centroid of the virtual light sources within it. The effective range R is a parameter depending on the attenuated energy of the light sources.

Regarding to the glossy effect, the reflecting direction is crucial to the rendering quality. For the indirect contribution, we evaluate the diffuse and specular term respectively. Similar to the specular cone in Voxel Cone Tracing [CNS*11], the importance on the specular contribution of the local illumination model is assigned while gathering VPLs and SH approximation,

$$w_{spec} = \frac{1}{\sigma_\theta \sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\| 1 - \cos^n \theta_i \|}{\sigma_\theta})^2}, \qquad (7)$$

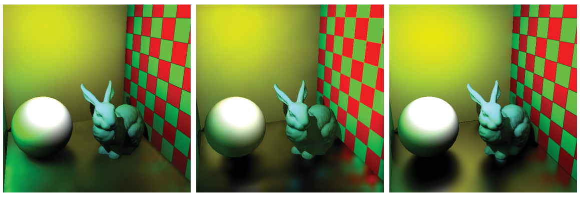
where the θ_i is the spanned angle difference between the light source and the reflection vector; σ_θ determines the shape of Gaussian weighted function, which in turn is dependent on surface glossiness. The effective number of the light sources in the specular term is similarly weighted to improve visual quality. Finally, after accumulating the indirect contribution, the final rendered result is generated by combining the direct illumination and the upsampled indirect illumination with a geometry-aware bilateral filter.

4. Evaluation

We implement our proposed techniques using OpenGL 3.3 and CUDA 5.5 and collected the results on a PC with NVIDIA GeForce GTX 670 graphics card. Our test scenes include the *Cornell Box* with 24k point proxies and $6 \times 6 \times 6$ voxels, and *Crytek Sponza* scene with 170k point proxies and $20 \times 20 \times 20$ voxels. The *Marching Size* represents the lower screen resolution where we conduct the visibility computation at. We use a 7×7 kernel size for the visibility filter, and compute the indirect illumination at a resolution of 256×256. The final image is further upsampled to the resolution of 512×512 through bilateral filtering and combined with direct illumination. See Table 1 for the detailed performance figures, and Fig. 1 for actual rendered results.

	Cornell Box		*Crytek Sponza*
Marching Size	64×64	128×128	128×128
Voxelization	0.26 ms		1.56 ms
Ray Marching	89.6 ms	272.7 ms	182.3 ms
Visibility Filter	18.9 ms	77.2 ms	110.6 ms
Illumination	326 ms	396.6 ms	323.4 ms
Upsampling	7.9 ms	7.9 ms	8.1 ms
Avg. fps	2.3 fps	1.4 fps	1.3 fps

Table 1: Performance Summary



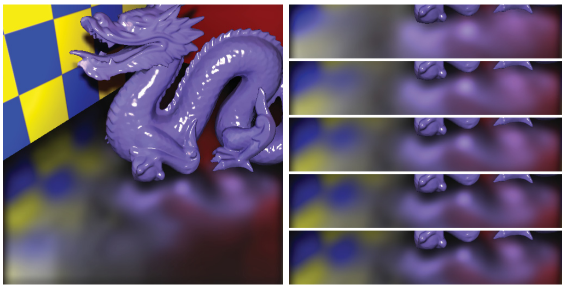
(a) ISM [RGK*08] (b) This work (c) Reference

Figure 5: Quality comparison (glossiness $N_s = 50$).

We compare the visual qualities between our work, a ray traced reference image, and the ISM [RGK*08] method in Figure 5. For the ISM, we adopt 4096 VPLs, 4096×4096 imperfect shadow maps, and 8K points to render the result. ISM can quickly approximate one bounce indirect illumination while considering visibility among the indirect bounces. Although this approach works well for a purely diffuse scene, it is difficult to produce detailed glossy illumination without using much more point light sources and larger

ISMs, which can degrade the rendering performance considerably. For this test scene, the rendering speed of our ISM implementation runs at about 1 fps. The reference image, which takes several minutes to generate, is rendered through tracing 1024 gathering rays for each deferred pixel to collect the indirect illumination energy. Our proposed technique can reasonably approximate the indirect glossy illumination at about 2 fps.

We further evaluate our method for glossy reflections under different glossiness parameters in Fig. 6, It can be observed that our method is fairly suitable for such purposes, as the dragon and textured wall are both faithfully conveyed for different types of surfaces.



(a) $N_s = 50$ (b) Different glossies

Figure 6: Rendered results under different glossiness settings. In Fig. 6b from top to bottom, $N_s = 10$ to 50.

Discussion In this work, we approximate the light source visibility by marching on the shadow rays through the voxel cells with point proxies. This process may undersample the point proxies and makes it difficult to reproduce a highly glossy surface reflections with complex geometric variations. On the other hand, by sharing occluded voxel cells, we reduce the number of required shadow rays, making it acceptable both quality-wise and performance-wise for reproducing semi-glossy reflections at interactive frame rates. The size of the deferred pixels directly affects the performance, and can be used as a parameter to trade-off between the performance and the fidelity of the approximation.

Although the distant indirect contribution is efficiently approximated with SH, the surface is still illuminated with many point light sources, which degrades performance. Therefore, this switch between SH and light sources can also be a trade-off parameter between performance and quality.

Limitations Shadow rays are casted using the low-resolution G-Buffers, and as a result, their origins and directions may not be consistent across different viewpoints. Because the indirect illumination is sensitive to the amount of visible voxel cells, this in turn causes some flickering artifacts during viewpoint animation. Our Visibility Filter smooth out most of these artifacts, but it can not guarantee to remove temporal artifacts due to the approximation inconsistencies across frames.

5. Conclusions and Future Works

In this paper, we described an interactive, indirect illumination technique by indexing the point proxies on a voxel cell structure, and marching through this structure using sparsely-casted shadow rays. The proposed Visibility Filter further reduces artifacts caused by the sparsity of shadow rays. We have demonstrated our method at interactive frame rates, and believe that this class of methods hold great potential for further research opportunities.

As the addressed issues and limitation, there are certain future research directions for this technique. First, we will focus on improving the efficiency on the sparse ray marching part, since the Visibility Filter already associates the visibility among neighboring surfaces. Moreover, comprehensively investigating the consistent visibility across frames is necessary to solve the flickering artifact. Finally, the sparse point proxies (e.g., the *Crytek Sponza* scene) occasionally result in missed shadow rays and thus lower visibility estimation quality. The use point proxies begets further study and insight on its sampling and distribution.

References

[BCK*11] BØRLUM J., CHRISTENSEN B. B., KJELDSEN T. K., MIKKELSEN P. T., NOE K. O., RIMESTAD J., MOSEGAARD J.: Sslpv: Subsurface light propagation volumes. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics* (New York, NY, USA, 2011), HPG '11, ACM, pp. 7–14. 2

[CNS*11] CRASSIN C., NEYRET F., SAINZ M., GREEN S., EISEMANN E.: Interactive indirect illumination using voxel cone tracing. *Computer Graphics Forum (Proceedings of Pacific Graphics 2011) 30*, 7 (sep 2011). 2, 3, 4

[DS05] DACHSBACHER C., STAMMINGER M.: Reflective shadow maps. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2005), I3D '05, ACM, pp. 203–231. 1

[KD10] KAPLANYAN A., DACHSBACHER C.: Cascaded light propagation volumes for real-time indirect illumination. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, ACM, pp. 99–107. 2

[Kel97] KELLER A.: Instant radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 49–56. 2

[OP11] OU J., PELLACINI F.: Lightslice: Matrix slice sampling for the many-lights problem. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 179:1–179:8. 2

[PKD12] PRUTKIN R., KAPLANYAN A., DACHSBACHER C.: Reflective Shadow Map Clustering for Real-Time Global Illumination. In *Eurographics* (2012), pp. 9–12. 2

[REG*09] RITSCHEL T., ENGELHARDT T., GROSCH T., SEIDEL H.-P., KAUTZ J., DACHSBACHER C.: Micro-rendering for

© The Eurographics Association 2014.

scalable, parallel final gathering. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 132:1–132:8. 2

[RGK*08] RITSCHEL 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. 27*, 5 (Dec. 2008), 129:1–129:8. 2, 4

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

© The Eurographics Association 2014.