A Real–time Global Illumination Approach for High Resolution Reflective Shadow Maps in Open World Scenes

Daniel Bischoff, Tobias Schwandt and Wolfgang Broll
Ilmenau University of Technology, Virtual Worlds and Digital Games Group, Ehrenbergstraße 29, Ilmenau, Germany

Keywords: Global Illumination, Indirect Lighting, Real–time Rendering, Bidirectional Reflective Shadow Maps, Open World.

Abstract: The complexity of offline and real–time rendering of global illumination effects is a vast field of research. Covering the complexity of dynamic open–world environments with changing light conditions and moving objects is challenging in real–time rendering. Especially the quality of light properties needs to be physically correct and sufficiently fast even if the light and environment conditions are changing. In this paper, we present a fast global illumination approach suitable to achieve indirect lighting using high–resolution reflective shadow maps in real time. Based on LightSkin, we present an enhancement applying bidirectional shadow maps to improve the quality of shadows and global illumination effects in open–world environments. A novel combination of accumulating virtual area lights from the reflective shadow map with an indirect light representation using LightSkin's proxy light sources shows a significant improvement. In an urban scene typical for open–world environments, the proposed approach is up to approx. 12 times faster than the original LightSkin approach. This makes our novel approach suitable for high–quality indoor as well as outdoor global illumination.

1 INTRODUCTION

Indirect illumination contributes signficantly to the overall quality of rendered images - in particular as it increases their plausiblity. However, due to its rather high rending costs, it is generally difficult to provide sophisticated results in real-time. In particular the specific requirements for providing indirect illumination for open-world scenes in real-time were widely neglected by previous research.

Particularly light types like the sun usually have a shadow map size of $1024 \times 1024\,\text{px}$ or higher. This high resolution is difficult to render in case of Reflective Shadow Maps (RSMs) and Global Illumination (GI), since the naïve approach is to light each frame buffer pixel by each virtual light deducted from the RSM. Consequently this leads to a high computational complexity. An option is to pre–compute the lighting reducing the per frame complexity. However, dynamically changing objects or light conditions in the environment cannot be supported. Modern applications—like open–world computer games—consist of many integrated objects that are affected by environment properties like weather conditions, the position of the sun, or even clouds. Therefore, an on–the–fly rendering of dynamically changing open–world environments with GI needs to be addressed to enhance the visual output in terms of realism and physical correctness.

Similar to (Ritschel et al., 2011), we employ a reduced representation of the RSM for accumulating indirect light. This paper combines sampling the RSM using Bidirectional Reflective Shadow Maps (BRSMs) introduced in (Ritschel et al., 2011) with the coarse model representation introduced along with the LightSkin GI approach (Lensing, 2014; Lensing and Broll, 2013b). The indirect light is accumulated for a set of sparsely distributed light caches rather than all surface points and subsequently interpolated for the remaining surface points. A reduction of the amount of virtual lights is achieved by an importance sampling strategy. This allows us to use high–resolution RSMs in complex scenes while maintaining real–time frame rates. Apart from the contribution to open–world scenarios, our approach is applicable for GI in general.

This paper is structured as follows: in Section 2 we investigate related work and indicate where our novel contributions differ from previous approaches. Section 3 presents our approach for sampling large

Bischoff D., Schwandt T. and Broll W.
A RealâĂŞtime Global Illumination Approach for High Resolution Reflective Shadow Maps in Open World Scenes.
DOI: 10.5220/0006135601160126
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 116-126
ISBN: 978-989-758-224-0

RSMs more efficiently and dynamically. In Section 4, we provide details on our approach combining the principles of LightSkin and BRSMs while Section 5 & Section 6 present an evaluation of the results we achieved along with the limitations that are still in place. Finally, Section 7 concludes our findings and gives a prospect on future work.

2 RELATED WORK

Indirect Lighting

Global illumination is a very comprehensive field of research. As early as 1997, Alexander Keller (Keller, 1997) described the first rendering technique to compute global illumination in textured scenes. Based on quasi–random walk and a Monte Carlo integration he achieves physically correct results. This algorithm has a complexity of $O(NK)$ where N is the number of rays and K the number of elements in the scene. However, ray–based approaches or radiosity are not suitable for real–time graphic applications because of the requirements, especially with open–world scenarios in mind.

Dachsbacher and Stamminger presented an approach called Reflective Shadow Maps in 2005 (Dachsbacher and Stamminger, 2005). It extends shadow maps by additional buffers that hold world position, normal values, and the color of the surfaces being lit by the light source. The additional buffers are used in combination with a screen–space technique to achieve interactive frame rates even for complex scenes. This approach provides the basis of many of the following rendering techniques.

Ritschel et al. presented Imperfect Shadow Maps (ISMs) (Ritschel et al., 2008) which create low–resolution shadow maps rendered around the current camera based on Virtual Point Lights (VPLs). A finally generated shadow map atlas is used for indirect illumination in fully dynamic scenes. Using point sampling shows to be sufficient in most cases, but has some disadvantages in terms of accuracy.

A study by Kaplanyan & Dachsbacher demonstrated the possibilities of using lattices and Spherical Harmonics (SH) based on RSMs (Kaplanyan and Dachsbacher, 2010). Overall, a computation time of a few milliseconds can be reached without requiring pre–computation. Based on a grid–based lighting they achieve single–bounce indirect illumination with occlusion. The Light Propagation Volumes (LPV) lookup inside the cascaded grids is the most expensive part of the approach.

Crassin et al. have presented the Voxel Cone Tracing (VCT) method (Crassin et al., 2011). It is based on a pre–filtered voxel representation of the scene. They use a voxel–based cone that corresponds to the individual mipmap levels of the pre–integrated scene. Therefore, only a small number of sampling rays is required as they can be approximated via the cone.

Lensing & Broll addressed global illumination in terms of diffuse and glossy surfaces by a new approach called LightSkin (Lensing and Broll, 2012; Lensing and Broll, 2013b; Lensing and Broll, 2013a; Lensing, 2014). It is based on ideas from irradiance caching (Ward et al., 1988) and radiance caching (Křivánek et al., 2005) as it interpolates the indirect light from so called light caches. These caches are distributed in model space during a per–model pre–computation step which can be parametrized according to the desired density.

With a RSM from the light source, indirect lights are generated as Virtual Area Lights (VALs). The generated caches are used to compute diffuse and glossy incident lighting by accumulating the indirect light from the RSM into two proxy light sources for each cache, one representing diffuse light and the other representing glossy light. Moreover, the caches' proxy lights are attenuated to account for occlusion by a double–projection method that approximates occlusion computation efficiently. The final output is generated by interpolating the accumulated lighting from the proxy lights to the surface points in the vicinity of a cache.

However, for a single surface point, the proxy lights of several surrounding caches are used and weighted according to their contribution. The approach delivers plausible results with reasonable performance and is applicable to virtual and mixed reality scenarios when RSM sizes are small compared to commonly used shadow map resolutions, e.g. for 128^2 or 256^2 px.

Our analysis of the approach shows that high–resolution reflective shadow maps result in significantly lower frame rates, since all of the pixels in the RSM are used as virtual lights for cache lighting.

A new radiance caching approach from Vardis, Papaioannou & Gkaravelis combines a radiance field chrominance with an optimized cache population scheme (Vardis et al., 2014). The main idea is to generate caches only on surface that finally contribute to the result.

Laurent et al. use a stochastic decimation process chained with a partitioning strategy (Laurent et al., 2016). For every triangle inside the scene, a proper VPL is considered. The approach doesn't need any

pre–computation and can be used for dynamic scenes as well.

Importance Sampling

Of the many different approaches associated with the use of importance sampling, we would like to mention Monte-Carlo Simulation, pseudo-random, and Perlin noise textures (Anderson, 1999; Green, 2005). They allow an efficient sampling of virtual light sources.

Moreover, Dachsbacher & Stamminger developed an importance sampling strategy based on RSMs improving the performance significantly (Dachsbacher and Stamminger, 2006). Using a resolution of 512×512 px for the final image and indirect lights, 37 frames per seconds can be reached. This is not sufficient for our needs of a high–resolution output.

Ritschel et al. presented a view–adaptive approach for imperfect shadow maps by using BRSMs (Ritschel et al., 2009; Ritschel et al., 2011). Depending on the current camera view, a set of potentially influencing VPLs is estimated. Hereby, high frame rates can be achieved even in large scenes. However, for occlusion evaluation, ISMs are used which are based on a point–based representation of the scene and, thus, provide an approximated visibility evaluation.

Prutkin, Kaplanyan & Dachsbacher present a method called Reflective Shadow Map Clustering (RSMC) for clustering VPLs (Prutkin et al., 2012). In contrast to BRSMs, they use VALs instead of VPLs. Overall, a single frame can be provided in $140 - 250$ ms depending on the number of clusters and RSM size. The importance sampling is similar to BRSMs and provides a decent quality for complex scenes. Nevertheless, the performance of rendering indirect lights is not sufficient in case of high–resolution shadow maps.

Another filtering method was presented by Barák, Bittner & Havran (Barák et al., 2013). They developed an adaptive sampling of VPLs based on G–buffer information. It is based on Metropolis–Hastings sampling. In comparison to BRSMs, this approach exhibits better temporal coherence.

With an heuristic sampling Hedman, Karras & Lehtinen indroduced an efficient VPL distribution using multiple frames (Hedman et al., 2016). Basically, they achieve a significant performance boost by reusing VPLs from previous frames and creating new VPLs if necessary.

3 EFFICIENT LIGHTING WITH RSMs

Among the methods summarized in the previous section, one approach stands out due to its method of accumulating and applying indirect light to surfaces: LightSkin uses light caches for indirect light accumulation and subsequently interpolates the accumulated results for the remaining surface points. This is a quite efficient approach to avoid shading each pixel in the viewbuffer by each virtual light deducted from the RSM. However, LightSkin uses each pixel in the RSM as a VAL. Therefore, this work focuses on sampling the RSM more efficiently by employing Bidirectional Reflective Shadow Maps as presented in (Ritschel et al., 2011) and adapt the LightSkin rendering pipeline to use only a selected subset of VALs for computing the proxy light parameters. In order to make our work easier to understand, we provide a short summary of the LightSkin method presented in (Lensing, 2014; Lensing and Broll, 2013b) first, and subsequently, we discuss its drawbacks.

LightSkin Overview

Cache Distribution

As mentioned earlier, LightSkin requires caches to be distributed on the scene objects in model space. With our implementation, this is performed in a pre–computation step for each model. For this step, two control parameters may be used to affect the distribution results: the maximum distance between cache and surface point d_ε, and the maximum angular difference between cache and surface normal ϑ_ε. The main steps for cache generation are:

1. The model is voxelized into cubes with an edge length of $d_\varepsilon/2$.
2. Surface normals are clustered into bins in a normal container. These bins hold every normal in a voxel that matches the bin's direction.
3. For each voxel, the normal bin representing the largest surface in the voxel is selected. A cache's normal and position are averaged by taking into account the points with normals deviating less than an angle of ϑ_ε from the bin's center normal. The surfaces that have been used for averaging are removed from the normal container. This procedure is repeated until the normal containers are empty.
4. Optionally, the caches are snapped to existing triangles in case they are "floating".

5. In a final step, caches with similar properties (generated in separate voxels) are joined.

For each cache, the position in model space (3 components), normal (3 components), represented area (float), and reflection coefficient (float) are stored in two RGBA32F textures.

Cache Lighting

In order to generate the proxy lights that represent the incident indirect light to a cache, first, virtual light sources are generated by rendering a reflective shadow map for the light source. However, instead of virtual point lights, rather virtual area lights are used. The area of each VAL is estimated by the projective area A_{disk} of the RSM pixel according to

$$A_{\text{disk}} = z_{\text{xl}}^2 \frac{w_{\text{near}} h_{\text{near}}}{z_{\text{near}}^2 w_{\text{RSM}} h_{\text{RSM}}}, \tag{1}$$

where z_{xl} denotes the distance between the surface point and the camera and w_{near}, h_{near}, and z_{near} denote the width, height and depth of the near clipping plane while w_{RSM} and h_{RSM} hold the width and height of the RSM. Using a simple shape like a disk allows using form factors as analytical solutions for integration over the surface area in later computation steps.

Proxy Lights Generation

For each of the pre–computed caches, in each frame, two proxy light sources are generated. One representing the diffuse indirect light and the other representing the glossy indirect light. To this end, the reflected light at surface point $\mathbf{x}$ in the direction ω is split up into two separate components:

$$L_o(\mathbf{x},\omega) = L_{\text{o,d}}(\mathbf{x},\omega) + L_{\text{o,g}}(\mathbf{x},\omega), \tag{2}$$

where L_o is the overall outgoing luminance, $L_{\text{o,d}}$ and $L_{\text{o,g}}$ are the diffuse and glossy components respectively. Lensing computes these components according to

$$L_{\text{o,d}}(\mathbf{x}) \approx \sum_{j=1}^{N_{\text{VAL}}} \frac{\Phi_{\text{l},j}\,\rho_{\text{d}}}{\pi} \cdot \frac{\cos^+\vartheta_{\text{l}}\cos^+\vartheta_{\text{x}}}{A_{\text{l},j} + \pi\left\|\mathbf{x}_{\text{l},j} - \mathbf{x}\right\|^2}, \tag{3}$$

and

$$L_{\text{o,g}}(\mathbf{x},\omega) \approx \sum_{j=1}^{N_{\text{VAL}}} \frac{\Phi_{\text{l},j}\,\rho_{\text{g}}\,(k+2)\cos^k\vartheta_{\text{e}}}{2\pi} \cdot \frac{\cos^+\vartheta_{\text{l}}\cos^+\vartheta_{\text{x}}}{A_{\text{l},j} + \pi\left\|\mathbf{x}_{\text{l},j} - \mathbf{x}\right\|^2}, \tag{4}$$

where N_{VAL} is the number of RSM pixels, Φ_{l} is the light flux of the VAL, ρ_{d} and ρ_{g} are the diffuse and glossy color respectively, and $\mathbf{x}_{\text{l}}$ is a single VAL's position. ϑ_{l} is the angle between the light's normal and the vector from the light's center to the surface point $\mathbf{x}$, ϑ_{x} is the angle between the surface normal and the vector from the surface point to the light's center, ϑ_{e} is the angle between the view direction and the reflection vector while k is the reflection exponent and A_{l} is the area of a VAL. We let $\cos^+$ denote the cosine clamped to values above zero.

Subsequently, proxy lights are created that represent this luminance at the cache position. First, the position of the proxy lights is computed according to

$$\mathbf{x}_{\text{p}} = \frac{\sum_{j=1}^{m} w(\mathbf{c},\mathbf{x}_{\text{l}})\mathbf{x}_{\text{l},j}}{\sum_{j=1}^{m} w(\mathbf{c},\mathbf{x}_{\text{l}})}. \tag{5}$$

The weights w for the diffuse and glossy term are computed differently according to

$$\begin{aligned} w_{j,\text{d}}(\mathbf{c}) &= \varepsilon_p + F_{\vec{c}\leftarrow\text{l,disk}}, \\ w_{j,\text{g}}(\mathbf{c}) &= \varepsilon_p + \cos^k\vartheta_{\text{e}}\, F_{\vec{c}\leftarrow\text{l,disk}}, \end{aligned} \tag{6}$$

where ε_p is a very small, non–zero value avoiding division by zero in Equation 5 and $F_{\vec{c}\leftarrow\text{l}}$ is the form factor between the VAL's disk area and an infinitesimal surface at the cache's position $\mathbf{c}$. Once the position is known, the proxy light source's flux can be computed as

$$\Phi_p = \frac{4\pi L_{o,VAL}(\mathbf{x},\omega)\left\|\mathbf{x}_{\text{p}} - \mathbf{c}\right\|^2}{f_r(\mathbf{c},\omega,\omega_i)(\mathbf{n_c}\cdot\omega_i)^+}, \tag{7}$$

where $f_r(\mathbf{c},\omega,\omega_i)$ denotes the Bidirectional Reflectance Distribution Function (BRDF) at the cache's position, in viewing direction ω and incident direction ω_i, and $\mathbf{n_c}$ is the cache's normal. Here, we let $(\ldots)^+$ denote clamping the value in parentheses to positive values. This makes $(\mathbf{n_c}\cdot\omega_i)^+$ the clamped dot product. In order to create the diffuse and glossy proxy light, Equation 3 and Equation 4 are plugged in as $L_{o,VAL}$.

Interpolation using the Proxy Lights

The proxy lights of multiple caches are used to interpolate indirect light for the surface points in the viewbuffer. However, the proxy lights are not treated as stationary, but rather moved along with the surface point in focus. This means interpolating both, proxy light position and flux. To this end, the closest n caches need to be known. Lensing stores the references of 16 adjacent caches for each vertex during the per–model pre–computation step. The contributions of these caches' proxy lights need to be weighted before being applied to a surface point. These weights are defined based on (Ward et al., 1988) and adapted

in (Lensing and Broll, 2013b; Lensing, 2014) for the diffuse (w'_{d}) and glossy (w'_{g}) terms according to

$$w'_{\mathrm{d}}(\mathbf{x},\mathbf{c}) = \left(1 - \frac{\|\mathbf{x}-\mathbf{c}\|}{d_{\mathrm{c,max}}}\right)\sqrt{(\mathbf{n_x}\cdot\mathbf{n_c})^+},$$
$$w'_{\mathrm{g}}(\mathbf{x},\mathbf{c}) = \left(1 - \frac{\|\mathbf{x}-\mathbf{c}\|}{d_{\mathrm{c,max}}}\right)\sqrt{(\mathbf{r_x}\cdot\mathbf{r_c})^+}, \tag{8}$$

where $d_{\mathrm{c,max}}$ is the largest distance to one of the 16 caches, $\mathbf{n_x}$ is the normal at the surface point $\mathbf{x}$, and $\mathbf{r_x}$ and $\mathbf{r_c}$ are the reflection vectors (the incident light direction mirrored at the surface normal) at the surface point and cache positions.

With these weights, the interpolation of both, proxy light position and flux, can be carried out according to

$$\mathbf{x}_{\mathrm{v,d}} = \frac{\sum_{j=1}^{n} w'_{\mathrm{d},j}(\mathbf{x},\mathbf{c})\mathbf{x}_{\mathrm{p,d},j}}{\sum_{j=1}^{n} w'_{\mathrm{d},j}(\mathbf{x},\mathbf{c})},$$
$$\Phi_{\mathrm{v,d}} = \frac{\sum_{j=1}^{n} w'_{\mathrm{d},j}(\mathbf{x},\mathbf{c})\Phi_{p,\mathrm{d},j}}{\sum_{j=1}^{n} w'_{\mathrm{d},j}(\mathbf{x},\mathbf{c})}. \tag{9}$$

These parameters are used to light each surface point in the view buffer.

Occlusion

As pointed out in (Lensing and Broll, 2013b; Lensing, 2014), visibility is only evaluated for the proxy light sources rather than all VALs representing it. Instead of testing for a binary visibility result, the proxy lights are extended with an estimate for their geometrical extent: an axis–aligned bounding box with an extent of $(2\sigma_x, 2\sigma_y, 2\sigma_z)$, where σ represents the standard deviation of the averaged VAL positions for proxy light creation along each of the main axes. This bounding box is converted to a sphere with a radius matching the largest dimension of the bounding box. Inspired by Bunnel's ambient occlusion computation scheme (Bunnell, 2005), occlusion is approximated by creating a disk for each cache with an area that is computed after cache distribution during the pre–computation step based on a Voronoi diagram.

Both, the bounding sphere of the proxy light source as well as the disks of the other caches are first projected onto a unit sphere and, subsequently, to a unit disk centered around the target cache using the form factors for a sphere and a disk (cf. (Lensing and Broll, 2013b)). There, they are converted to quads with the same area in order to simplify computing the overlap. The amount of cache quads overlapping with the proxy light source's quad is then used as an estimate for the occlusion of that same proxy light.

Drawbacks

The performance figures of the original LightSkin approach in (Lensing and Broll, 2013b; Lensing and Broll, 2013a; Lensing, 2014) drop significantly if the RSM resolution exceeds 256×256 px while rendering scenes that contain more than $4,000$ caches. This effect can be attributed to using all virtual lights derived from the RSM for cache lighting. In open–world scenes, RSM resolutions of 1024×1024 px or higher are common, which illustrates why the original LightSkin approach was only suited for scenes with a rather small extent such as mixed–reality scenes containing a small number of virtual objects for which the algorithm was originally designed.

In order to improve on this and to make the Lig-

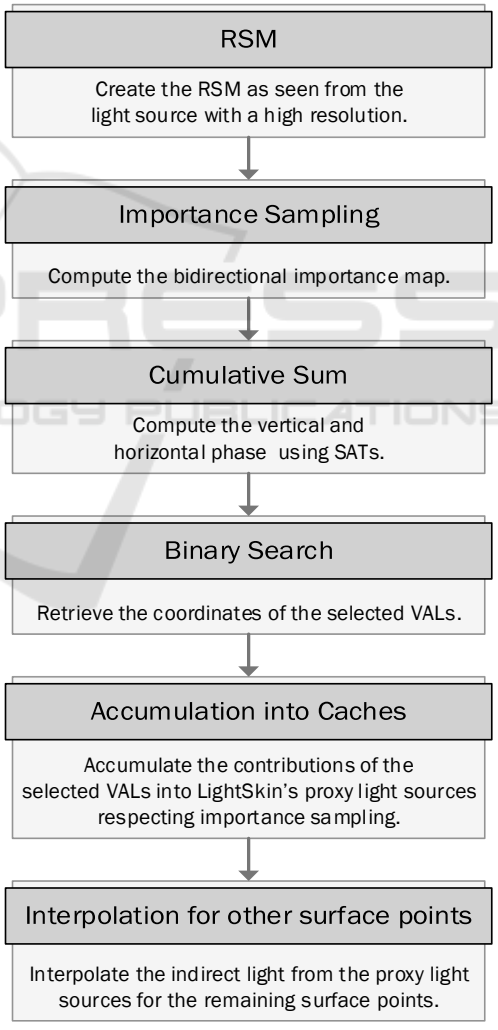

Figure 1: This figure shows the our provided pipeline for an efficient sampling of RSMs. After performing the highly efficient sampling using BRSMs, we combine this approach with LightSkin.

htSkin approach feasible for open–world scenes with larger extent, we examined several approaches that focus on finding virtual lights more efficiently, like BRSMs (Ritschel et al., 2011) and RSMC (Prutkin et al., 2012). Both methods rely on importance sampling in combination with an efficient approach to propagate the accumulated light into a reduced representation of the scene.

We decided to combine the importance sampling and virtual light selection of BRSMs with the LightSkin rendering pipeline since the additional feature of RSMC–the area representation of VPLs–is covered by the LightSkin approach already. In contrast to RSMC, where the geometric extent of the VPLs is transferred to polygons or discs, in (Lensing, 2014), the geometric extent is approximated using the standard deviation of the world position of the VALs.

With our contribution combining BRSMs with the LightSkin approach, we achieve plausible results at real–time frame rates for RSM resolutions of up to 2048×2048 px.

To this end, we suggest the pipeline illustrated in Figure 1 in order to first select a subset of VALs based on their estimated importance. Initially, we accumulate the importance map into cumulative sum textures vertically and horizontally and subsequently perform a binary search on these cumulative sums as described in (Ritschel et al., 2011). The selected VALs need to be plugged into LightSkin's indirect light accumulation which means adapting the generation of proxy lights. For the reduced subset, the contributions have to be weighted with respect to the importance sampling of the RSM. With the proxy light parameters computed, LightSkin's interpolation can be carried out as described in (Lensing and Broll, 2013a; Lensing, 2014). The details of this pipeline are described in the following section.

4 COMBINING LIGHTSKIN GI WITH BRSMs

Initially, we create a traditional RSM with a high resolution from a light source inside the scene as described in (Dachsbacher and Stamminger, 2005). Based on the RSM and the current view information, we estimate the importance of each virtual light using 512 Halton–distributed view samples according to

$$I_{\mathrm{VAL}} = \frac{1}{N_{\mathrm{VS}}} \sum_{j}^{N_{\mathrm{VS}}} \Phi_{\mathrm{l}} \cdot \cdot \frac{(\mathbf{n}_{\mathrm{l}} \cdot \mathbf{d}_{\mathrm{VAL}\to\mathrm{VS}})^{+} \cdot (\mathbf{n}_{\mathrm{VS}} \cdot (-\mathbf{d}_{\mathrm{VAL}\to\mathrm{VS}}))^{+}}{D(\mathbf{l}_{\mathrm{VAL}\to\mathrm{VS}})}, \tag{10}$$

where I_{VAL} denotes the importance of a single VAL, N_{VS} is the number of view samples used for importance estimation, $\mathbf{n}_{\mathrm{l}}$ and $\mathbf{n}_{\mathrm{VS}}$ are the VAL's and the view sample's normals respectively, and $\mathbf{d}_{\mathrm{VAL}\to\mathrm{VS}}$ is the normalized version of the vector between the VAL and the view sample which is denoted as $\mathbf{l}_{\mathrm{VAL}\to\mathrm{VS}}$. In the denominator, we us an expression that penalizes importance depending on the distance between VAL and view sample by clamping the squared distance to values above 1, according to

$$D(\mathbf{l}_{\mathrm{VAL}\to\mathrm{VS}}) = = \begin{cases} 1 & \text{if} \quad ||\mathbf{l}_{\mathrm{VAL}\to\mathrm{VS}}||^2 < 1 \\ ||\mathbf{l}_{\mathrm{VAL}\to\mathrm{VS}}||^2 & \text{if} \quad ||\mathbf{l}_{\mathrm{VAL}\to\mathrm{VS}}||^2 > 1 \,. \end{cases} \tag{11}$$

The resulting importance map is stored in a two–dimensional R32F texture with the same size as the RSM. Every pixel represents the estimated importance of one VAL. Figure 2 illustrates the importance sampling with 512 Halton–distributed samples.

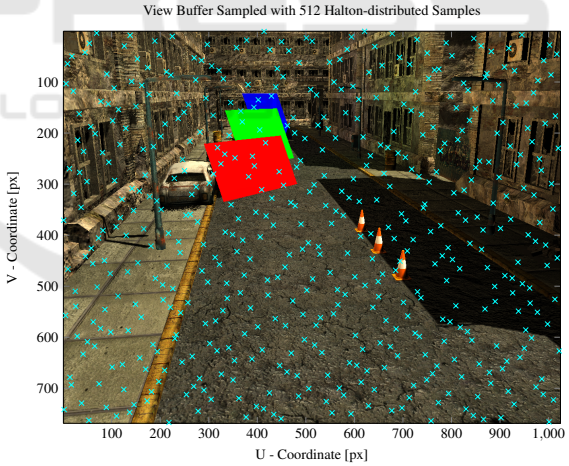

Figure 2: This figure shows the view buffer of size 1024×768 of a street scene in which the coordinates of 512 Halton–distributed samples are illustrated. The view buffer is sampled at these coordinates to estimate the bidirectional importance of each VAL in the RSM.

Subsequently, we generate a cumulative sum vertically and, based on that result, horizontally. We use Summed Area Tables (SATs) to build the cumulative sums (Crow, 1984; Hensley et al., 2005). This computation can be done in $O(\log(n))$ time with m draw calls per dimension according to

$$m = ceil(log_2(M)), \tag{12}$$

where M is the number of elements along the according dimension. One two–dimensional R32F texture is used to store the result of the cumulative sum in the vertical direction (same size as RSM) and another single–dimensional R32F texture is used for the horizontal result (same width as the RSM). To find the VAL selection, we do a binary search within an extra draw call and store the pixel coordinates of the selected VALs into a RG32F texture ($1 \times N_{sVAL}$) as described in (Ritschel et al., 2011).

In order to plug the VALs indexed by these coordinates into LightSkin's rendering pipeline, the selected subset of VALs needs to be integrated into LightSkin's computation of proxy light parameters for each cache. For each of these proxy lights, the light's position (Equation 5) is computed according to

$$\mathbf{x}_\mathrm{p} = \frac{\sum_{j'=1}^{N_\mathrm{sVAL}} w(\mathbf{c}, \mathbf{x}_\mathrm{l}) \cdot \mathbf{x}_{\mathrm{l},j'} \cdot 1/P_{\mathrm{VAL},j'}}{\sum_{j'=1}^{N_\mathrm{sVAL}} w(\mathbf{c}, \mathbf{x}_\mathrm{l}) \cdot 1/P_{\mathrm{VAL},j'}}. \tag{13}$$

where N_sVAL is the number of selected VALs, and P_VAL is the probability that the VAL is being selected by the importance sampling.

Using this position, LightSkin computes the proxy light's flux according to Equation 7 where the outgoing luminance is represented as a diffuse and glossy component (cf. Equation 2)

The components' terms in Equation 3 and Equation 4 have to be adapted accordingly to support using only the selected subset which, for the diffuse term, yields

$$L_{\mathrm{o,d}}(\mathbf{x}) \approx \sum_{j'=1}^{N_\mathrm{sVAL}} \frac{\Phi_{\mathrm{l},j'}\,\rho_\mathrm{d}}{\pi} \cdot \cdot \frac{\cos^+\vartheta_\mathrm{l}\cos^+\vartheta_\mathrm{x}}{A_{\mathrm{l},j'} + \pi\left\|\mathbf{x}_{\mathrm{l},j'} - \mathbf{x}\right\|^2} \cdot \frac{1}{P_{\mathrm{VAL,j'}}}, \tag{14}$$

and consequently for the glossy term

$$L_{\mathrm{o,g}}(\mathbf{x},\omega) \approx \sum_{j'=1}^{N_\mathrm{sVAL}} \frac{\Phi_{\mathrm{l},j'}\,\rho_\mathrm{g}\,(k+2)\cos^k\vartheta_\mathrm{e}}{2\pi} \cdot \cdot \frac{\cos^+\vartheta_\mathrm{l}\cos^+\vartheta_\mathrm{x}}{A_{\mathrm{l},j'} + \pi\left\|\mathbf{x}_{\mathrm{l},j'} - \mathbf{x}\right\|^2} \cdot \frac{1}{P_{\mathrm{VAL,j'}}}. \tag{15}$$

By these modifications, the existent LightSkin rendering pipeline is extended to enable compatibility with BRSM–based VAL selection. With the computed parameters for each of the two (diffuse and glossy) proxy lights representing the incident light at each cache, the interpolation can be carried out as described in Section 3. Subsequently, the incident light at each cache is occluded with the double–projection method summarized earlier. The resulting scene is lit with indirect diffuse and glossy light with indirect occlusion, the quality of which depends on cache density.

5 RESULTS

To assess the results and as a reference for the performance, we use the original LightSkin approach (accumulating all VALs) as a comparison. LightSkin has been compared to other state–of–the–art methods and ground truth data extensively in (Lensing, 2014; Lensing and Broll, 2013b). Our method is an extension to this approach to enhance the performance of indirect lighting without losing visual quality. Therefore, we evaluate our approach by comparing our results to the standard LightSkin approach. For our evaluation, we examined several levels of RSM resolutions: square buffers with sizes of 128^2, 256^2, 512^2, 1024^2, and 2048^2 pixels. These were each reduced to subsets of different sizes: 2048, 4096, and 8192 VALs were selected by importance sampling the RSM. For all these computations, we used 512 view samples for importance estimation. However, we also examined the results for reducing that number to 256 in order to gain performance with larger selection sets in high–resolution RSMs as described later in this section.

Visual Quality

The visual quality of the results rendered using only the selected subset of 2048 VALs is almost indistinguishable from the original results when comparing any RSM resolution. This is illustrated by Figure 6 which compares the indirect light buffers computed using all VALs (left column) and only the selected VALs (center column). To illustrate the differences spatially, we computed the absolute difference between the two buffers (right column). The top row shows the scene for which the buffers were computed. We used a commonly employed urban street scene and planted some colored planes to illustrate color bleeding (note that the buildings on the right are lit by the light reflected off the planes) with approx. 5500 caches. Despite these images using only 2048 VALs, we also examined the errors for subsets of 4096 and 8192 VALs. The average over the entire absolute difference image is plotted in Figure 3. We observe that the average error decreases when more VALs are selected but it also increases with the RSM resolution. This is due to a constant number of VALs being selected while an increasing number is rejected. However, the difference images demonstrate that this increasing error is barely noticeable.

Even in other scenes the results of the indirect lighting are comparable to other approaches. In Figure 4, we demonstrate the output of the indirect lighting inside the Sponza scene. It is clearly visible that the indirect reflection of the green curtain is visible on

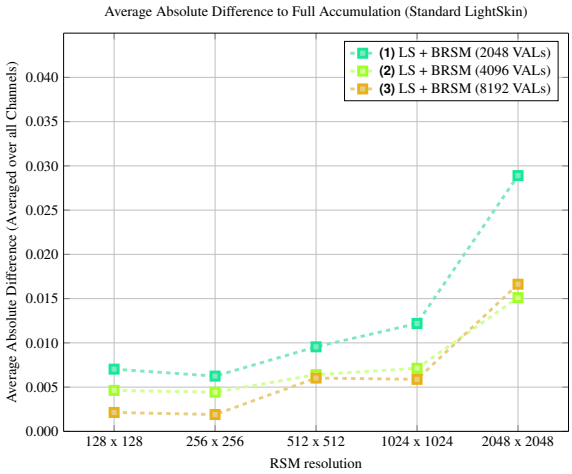

Figure 3: Average absolute differences between the results computed with the standard LightSkin algorithm and the results computed with our method. The evaluation is performed for different numbers of selected VALs and the results are plotted over the increasing RSM resolution.

the stone wall. We have used LightSkin with BRSMs and a sun light with a RSM resolution of 2048^2 pixels. 8192 VALs are selected to compute the final indirect lighting.

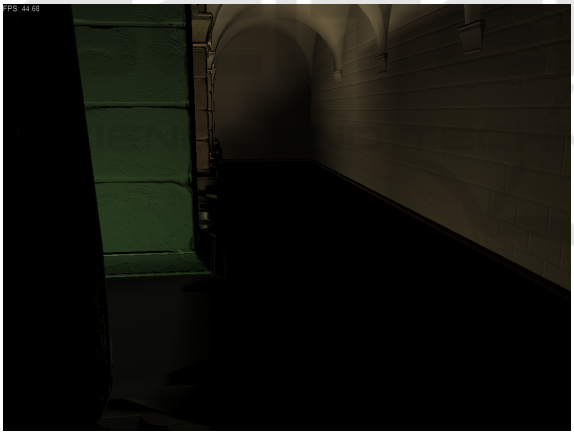

Figure 4: This figure shows the indirect lighting output using LS + BRSM with a RSM resolution of 2048^2 pixels and a subset of 8192 VALs. Using an NVIDIA GeForce GTX 1080 we achieve more than 40 FPS.

Temporal Coherence

An important aspect to consider with importance sampling is temporal coherence. Small changes inside the scene change only a few pixels of the RSM and therefore the importance sampling output. But with significant changes in scene geometry, light sources, or camera, we observe some flickering. Especially for the smaller VAL selection sets and higher resolution RSMs, the flickering becomes clearly visible. However, increasing the subset to 8192 VALs made the flickering imperceptible, while still performing at real–time frame rates—even with only 256 view samples for importance estimation. This way, we achieve a stable output despite camera, object, or light source movements.

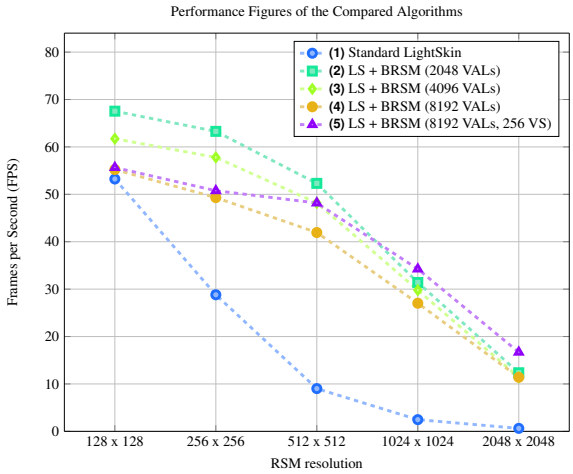

Figure 5: Performance figures comparing the standard LightSkin algorithm to our new approach which uses a reduced subset of VALs. The performance is measured in terms of frames per second for several parameterizations of our method.

Performance

The main advantage of our more sophisticated approach is that it maintains quality as shown earlier while decreasing computational complexity drastically. Figure 5 compares performance in terms of frames per second of LightSkin with our enhanced algorithm. Even though LightSkin (1) delivers a competitive number of frames per second for smaller, impractical RSM resolutions, our method excels for higher resolution shadow maps. For evaluating our method, we used 512 view samples for importance estimation and 2048 (2), 4096 (3), and 8192 (4) selected VALs. The number of VALs directly affects performance, however, the more limiting factor is the number of view samples: when using only 256 view samples (5), we can use 8196 VALs and achieve higher frame rates for the largest resolution RSMs.

6 LIMITATIONS

The most noticeable limitation compared to the original LightSkin approach is that we do not support

materials with Sub–surface Scattering (SSS). Sub–surface Scattering does not affect importance estimation in Equation 10. Therefore, VALs that illuminate models on their non–camera–facing side are not considered important. If SSS is to be supported, importance estimation needs to be adapted and indirect light accumulation for SSS materials needs to be adapted as demonstrated and implemented in (Lensing, 2014).

Furthermore, indirect shadows do become less accurate due to using the standard deviation of the geometric distribution of the VALs as an estimate for the indirect light's area. Since the selected subset contains less VALs than the original RSM, this standard deviation becomes less representative. However, the occlusion itself is based on the caches distributed on the models. Therefore, the approximation is still accurate for the selected subset of VALs.

Finally, we still consider temporal coherence an issue even though we could make it unnoticeable by increasing the number of selected VALs. In the future we plan to explore further methods to achieve inter–frame stability, for example by employing lazy–update schemes for the VAL selection, using pseudo-random seeds, reuse information of the previous frame, or more sophisticated importance estimation. This could improve performance further and allow us to employ this approach for more than just the primary RSM.

7 CONCLUSION

We presented an approach extending LightSkin. While preserving its outstanding visual quality it has proven to be much more efficient. As for the performance gain, we achieve approximately 30 fps for a RSM resolution of 1024×1024px where the original approach delivered only 2.5 fps which results in our method being roughly 12 times faster than the original approach. Furthermore, our method is highly customizable to a desired trade–off between quality and performance by adapting parameters like the number of view samples, number of selected VALs, the RSM resolution, and finally, cache distribution density. Although some limitations exist, our method delivers promising results considering real–time global illumination in open–world scenes with high resolution RSMs.

In order to further improve this combination, several paths can be undertaken: For better temporal coherence and to reduce the susceptibility to flickering, methods for improving the selected subset could be explored. For example, updating only part of the subset could be possible, but this lazy–updating would introduce some lag to the computation. One possibility would be to use the visible caches' view buffer position as view samples for importance estimation. In the light of the findings of this work, RSMC can be considered an additional possibility for improving temporal stability as the more accurate area approximation might decrease flickering artifacts. Future work should explore the effect on temporal stability and mitigations to flickering while observing the additional complexity introduced to the rendering pipeline. Moreover, other approaches for rendering the indirect lighting should be taken into account.

REFERENCES

Anderson, E. C. (1999). Monte Carlo Methods and Importance Sampling. *Lecture Notes*, 1999(October):1–8.

Barák, T., Bittner, J., and Havran, V. (2013). Temporally coherent adaptive sampling for imperfect shadow maps. *Computer Graphics Forum*, 32(4):87–96.

Bunnell, M. (2005). Dynamic Ambient Occlusion and Indirect Lighting. *GPU Gems*, 2(2):223–233.

Crassin, C., Neyret, F., Sainz, M., Green, S., and Eisemann, E. (2011). Interactive Indirect Illumination Using Voxel Cone Tracing. *Computer Graphics Forum*, 30(7):1921–1930.

Crow, F. C. (1984). Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques - SIGGRAPH'84*. Association for Computing Machinery (ACM).

Dachsbacher, C. and Stamminger, M. (2005). Reflective Shadow Maps. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games - SI3D '05*, pages 203–231. Association for Computing Machinery (ACM).

Dachsbacher, C. and Stamminger, M. (2006). Splatting indirect illumination. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games - SI3D '06*. Association for Computing Machinery (ACM).

Green, S. (2005). Implementing improved perlin noise. *GPU Gems*, 2:409–416.

Hedman, P., Karras, T., and Lehtinen, J. (2016). Sequential Monte Carlo instant radiosity. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '16*. Association for Computing Machinery (ACM).

Hensley, J., Scheuermann, T., Coombe, G., Singh, M., and Lastra, A. (2005). Fast summed-area table generation and its applications. *Computer Graphics Forum*, 24(3):547–555.

Kaplanyan, A. and Dachsbacher, C. (2010). Cascaded light propagation volumes for real-time indirect illumination. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D 10*. Association for Computing Machinery (ACM).

Keller, A. (1997). Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*. Association for Computing Machinery (ACM).

Křivánek, J., Gautron, P., Pattanaik, S., and Bouatouch, K. (2005). Radiance caching for efficient global illumination computation. *Visualization and Computer Graphics, IEEE Transactions on*, 11(5):550–561.

Laurent, G., Delalandre, C., Riviere, G. D. L., and Boubekeur, T. (2016). Forward Light Cuts: A Scalable Approach to Real-Time Global Illumination. *Computer Graphics Forum*, 35(4):79–88.

Lensing, P. (2014). *LightSkin: Echtzeitbeleuchtung für Virtual und Augmented Reality*. PhD thesis, Ilmenau University of Technology.

Lensing, P. and Broll, W. (2012). Instant indirect illumination for dynamic mixed reality scenes. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Institute of Electrical & Electronics Engineers (IEEE).

Lensing, P. and Broll, W. (2013a). Efficient shading of indirect illumination applying reflective shadow maps. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '13*. Association for Computing Machinery (ACM).

Lensing, P. and Broll, W. (2013b). LightSkin: Real-time Global Illumination for Virtual and Mixed Reality. In *Proceedings of the 5th Joint Virtual Reality Conference*, JVRC '13, pages 17–24, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Prutkin, R., Kaplanyan, A., and Dachsbacher, C. (2012). Reflective Shadow Map Clustering for Real-Time Global Illumination. In Andujar, C. and Puppo, E., editors, *Eurographics 2012 - Short Papers*. The Eurographics Association.

Ritschel, T., Eisemann, E., Ha, I., Kim, J. D. K., and Seidel, H.-P. (2011). Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. *Computer Graphics Forum*, 30(8):2258–2269.

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 Transactions on Graphics*, 27(5):1.

Ritschel, T., Grosch, T., and Seidel, H.-P. (2009). Approximating dynamic global illumination in image space. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games - I3D '09*. Association for Computing Machinery (ACM).

Vardis, K., Papaioannou, G., and Gkaravelis, A. (2014). Real-time radiance caching using chrominance compression. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):111–131.

Ward, G. J., Rubinstein, F. M., and Clear, R. D. (1988). A Ray Tracing Solution for Diffuse Interreflection. *ACM SIGGRAPH Computer Graphics*, 22(4):85–92.

APPENDIX

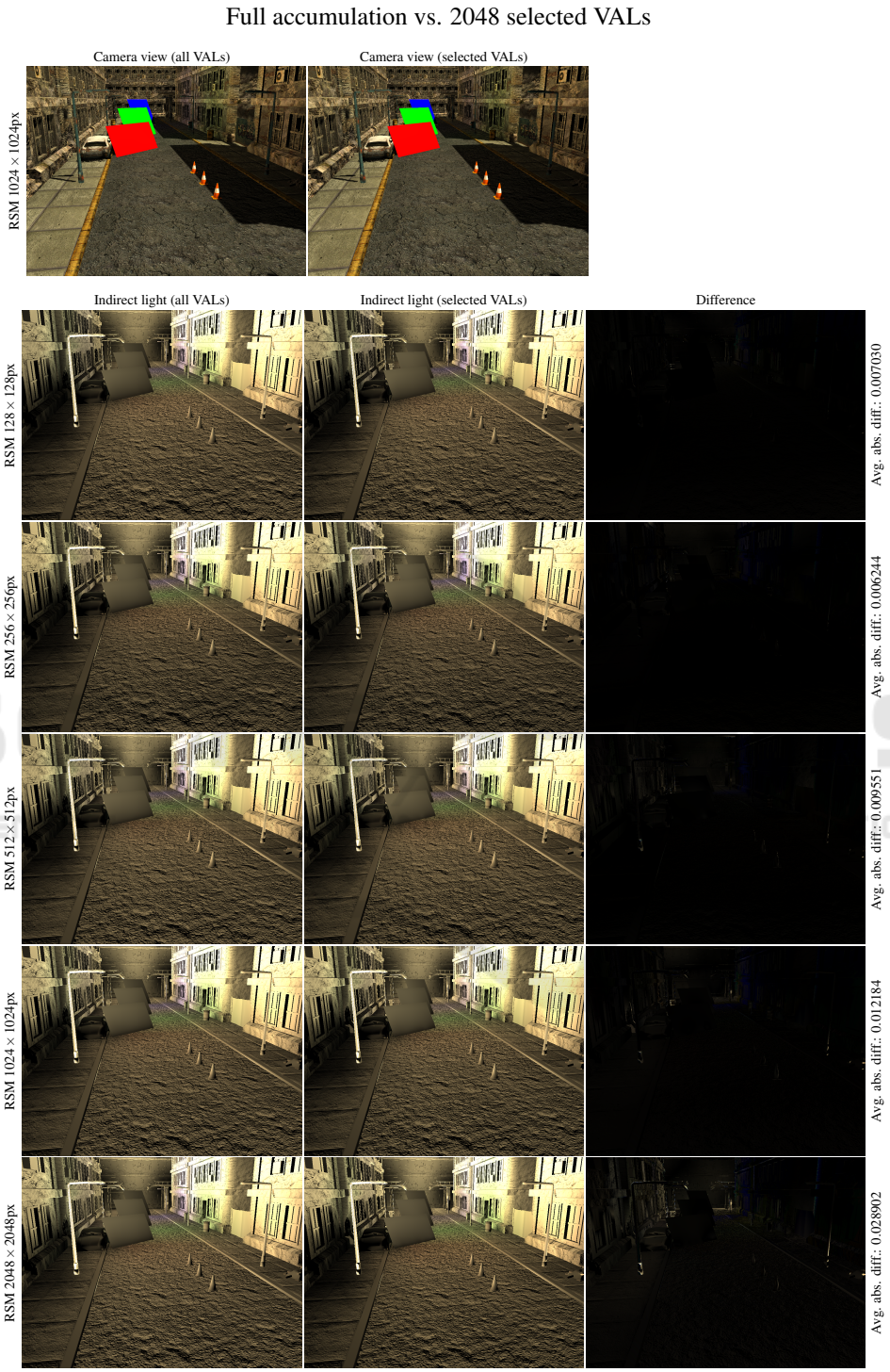

Figure 6: This figure illustrates the indirect light quality comparing the accumulation of all VALs (left column) to using only a selected subset of 2048 VALs (center column). The right column illustrates the difference between the two indirect light buffers. Adjacent to the difference images the average absolute error is listed. This comparison is presented for several resolutions of the RSM (rows) while he first row of images provides the camera view of the full accumulation vs. using only 2048 VALs. The scene shown in these images holds approx. 5500 caches.