Fast Indirect Illumination Using Two Virtual Spherical Gaussian Lights

Yusuke Tokuyoshi *
Sqaure Enix Co., Ltd.

Figure 1: *Rendering result (a) using two VSGLs, and its decomposed images (b)(c)(d)(e) for each light path (264 k triangles scene, minimum roughness: 0.04, resolution: 1920×1080, GPU: AMD Radeon™ R9 290X). Our method roughly approximates one-bounce glossy indirect illumination including caustics paths (d)(e) for scenes lit by a spot light. The computation time of indirect illumination is 0.7 ms (RSM: 0.224 ms, VSGL generation: 0.063 ms, shading: 0.407 ms).*

Introduction. A virtual spherical Gaussian light (VSGL) [Tokuyoshi 2015] is an approximation of a set of virtual point lights (VPLs) for real-time rendering. Thousands of VSGLs can be dynamically generated using mipmapped specialized reflective shadow maps (RSMs) to render glossy indirect illumination at 20-30 ms. Although this approach is efficient compared to VPLs, rendering at 20-30 ms is still expensive for some time-sensitive applications such as video games. This poster demonstrates glossy indirect illumination in 1 ms using only two VSGLs. In this poster, each VSGL has a single spherical Gaussian lobe to represent radiant intensity, and thus diffuse and specular reflections at the second bounce are represented with two VSGLs. This rough approximation is suitable for scenes which are locally lit by a spot light (e.g., flashlight in a cave). To generate these two VSGLs dynamically, this poster presents a specialized implementation using a parallel summation algorithm. Other than RSMs, our implementation uses only small temporary buffers with a resolution that is 1/64 of an RSM. Therefore, the proposed method is not only fast, but also memory saving.

Implementation. A VSGL is generated by summing parameters of directional/positional distributions of VPLs described in the supplemental document. Therefore, this poster employs a parallel summation algorithm using compute shaders instead of mipmapping. Our implementation uses two passes to generate each VSGL from a 128×128 of resolution RSM (i.e., 16384 VPLs) as shown in Fig. 2. In the first pass, a thread is dispatched for each RSM pixel, and calculates the above distribution parameters. Next, for each work group, these parameters are summed using shared memory in parallel. In our implementation, 8×8 threads are used for the work group size to reduce memory barriers. After that, a single thread stores the summed parameters in temporary buffers for each workgroup, and then the first pass terminates. Since 8×8 pixels are reduced, the resolution of these temporary buffers is 16×16. In our implementation, three *R32G32B32A32_FLOAT* textures are used for these temporary buffers. The second pass performs using a single work group whose size is 8×8 threads. First, each thread reads the temporary buffers using hardware-assisted bilinear texture filtering to simultaneously compute the sum of 2×2 pixels. Then, these parameters are summed in the same fashion as the first pass. Finally, a VSGL is generated using the total of the distribution parameters. In this poster, two VSGLs are generated using the above algorithm for diffuse and specular reflections. Unlike the original VSGLs, shadow maps are omitted for simplicity. Once VSGLs are generated, they are evaluated using an analytic formula for shading.

*e-mail:tokuyosh@square-enix.com

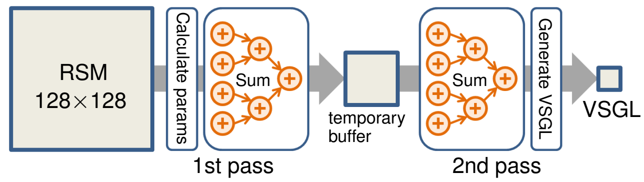

Figure 2: *VSGL generation using parallel summation.*

Table 1: *Comparison with LPVs for the LDDE light path (GPU: NVIDIA® GeForce® GTX™ 770).*

	VSGL	LPVs (32^3 voxels ×4 cascades)
Computation time	0.492 ms	2.232 ms
Memory usage	204 kB	1984 kB

Results. As shown in Fig. 1, our method approximates indirect illumination including caustics in 0.7 ms without any high-frequency artifacts (e.g., flickering). Table 1 shows comparison with cascaded light propagation volumes (LPVs) [Kaplanyan and Dachsbacher 2010] for a 699 k triangles scene. Since LPVs are inefficient for highly glossy materials, only the LDDE light path is evaluated using a single VSGL. For this experiment, our method is faster and more memory saving than LPVs. Although rendering using so few VSGLs can be a rough approximation, its performance and visual quality are a practical level for scenes lit by a spot light.

References

KAPLANYAN, A., AND DACHSBACHER, C. 2010. Cascaded light propagation volumes for real-time indirect illumination. In *I3D'10*, 99–107.

TOKUYOSHI, Y. 2015. Virtual spherical Gaussian lights for real-time glossy indirect illumination. *Comput. Graph. Forum 34*, 7.