Foveated Instant Radiosity

Lili Wang* [1;2] Runze Li [1] Xuehuai Shi [1] Ling-Qi Yan† [3] Zhichao Li [1]

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
[2] Peng Cheng Laboratory, Shengzhen, China
[3] University of California Santa Barbara, California, U.S.

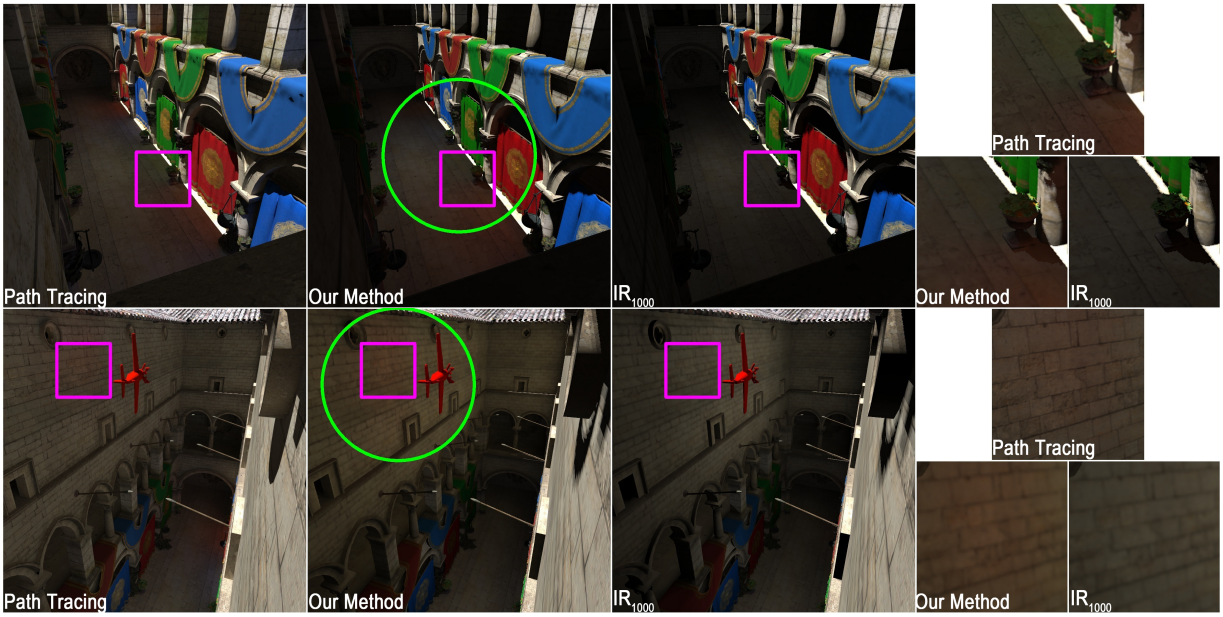

Figure 1: Illumination effect rendered with path tracing (ground truth, column 1), our method with 1000 virtual point lights (column 2) and instant radiosity method with 1000 virtual point lights IR_{1000} (column 3) in Sponza. Compared with IR_{1000}, our method achieves a $3\times$ speedup and smaller mean squared pixel error in the fovea marked with the green circle: our method vs IR_{1000}, 2.42×10^{-3} vs 4.03×10^{-3} for the top row, and 1.09×10^{-3} vs 3.16×10^{-3} for the bottom row. The details in the magenta square are magnified in column 4 (top: path tracing; bottom-left: ours; bottom-right: IR_{1000}). The color bleeding effect of our method is closer to the ground truth than that of IR_{1000}.

Abstract

Foveated rendering distributes computational resources based on visual acuity, more in the foveal regions of our eyes and less in the periphery. The traditional rasterization method can be adapted into the foveated rendering framework in a quite straightforward way, but it's difficult for estimating global illumination. Instant Radiosity is an efficient global illumination method. It generates Virtual Point Lights (VPLs) on the surface of the virtual scenes from light sources and uses these VPLs to simulate light bounces. However, instant radiosity can not be adapted into the foveated rendering pipeline directly, and is too slow for virtual reality experience. What's more, instant radiosity does not consider temporal coherence, therefore it lacks temporal stability for dynamic scenes. In this paper, we propose a foveated rendering method for instant radiosity with more accurate global illumination effects in the foveal region and less accurate global illumination in the peripheral region. We define a foveated importance for each VPL, and use it to smartly distribute the VPLs to guarantee the rendering precision of the foveal region. Meanwhile, we propose a novel VPL reuse scheme, which updates only a small fraction of VPLs over frames, which ensures temporal coherence and improves time efficiency. Our method supports dynamic scenes and achieves high quality in the foveal regions at interactive frame rates.

Index Terms: 3D scene rendering; Real-time rendering; Foveated rendering; Indirect illumination

1 Introduction

Immersion and interaction are the most important features of Virtual Reality (VR), therefore VR applications require high quality and performance. Nowadays, VR Head Mounted Display (HMD) acquires a refresh rate of 90Hz for each eye, which proposes a great challenge for traditional rendering algorithms. Foveated rendering is a rendering technique that provides users different rendering qualities in foveal regions and peripheral regions according to the model of human visual system [8, 33]. Oculus Go and Quest HMDs adopted the fixed foveated rendering technique to allow the periphery region of each eye to be rendered at a lower resolution than fixed foveal region at the center [34]. While other VR systems use eye trackers integrated with VR HMDs to reduce the rendering workload by reducing the image quality in the peripheral vision outside of the zone gazed by the fovea. In either case, the input of foveated rendering is the user's viewpoint, foveal region and geometry of

*e-mail: wanglily@buaa.edu.cn
†e-mail: lingqi@cs.ucsb.edu

virtual scenes, and the output is an image with different rendering qualities in different regions.

The traditional rasterization method can be adapted to foveated rendering directly because it can render scenes with multiple spatial resolutions easily. For example, Guenter et al. [8] used foveated graphics to generate a 3-layer image with different resolutions using a rasterization pipeline and composited them to generate the output images. But rasterization can not easily render indirect illumination. Radiosity method [39] is an important method to render high-quality global illumination effects for the virtual scenes with diffuse surfaces, such as color bleeding. But radiosity is only used to render static scenes offline due to the heavy computation of form factors. Instant radiosity [15] was proposed to accelerate illumination computation for the dynamic scenes with diffuse surfaces. instant radiosity generates VPLs on the surface of the virtual scenes from light sources and uses these VPLs' direct illumination to simulate indirect light bounces. The illumination quality of a region is determined by the number of VPLs that cast light on it. However, VPLs are independent of the user's viewpoint, so VPLs in the instant radiosity method can not be arranged according to the foveal region directly.

To adapt instant radiosity to foveated rendering, we use multi-resolution illumination as a high-level idea, which means that the numbers of VPLs contributing to the different regions of the output image are different, i.e. more VPLs cast light on the foveal region and less on the periphery region. To achieve this goal, two challenges need to be addressed: one is how to generate VPLs for each frame based on the fovea; the other is how to maintain the adjacent frames' stability when the foveal region, the viewpoint or the scene changes.

In this paper, we propose a foveated instant radiosity method to address these two problems. For the first problem, we generate VPL candidates on the surfaces that are directly visible from the foveal regions, and then compute the *foveated importance* of each VPL to measure its lighting contribution to the foveal region. After this, the VPLs with higher foveated importance values are selected to cast light on the virtual scenes. For the second problem, a VPL management method is introduced to evaluate the *essential VPLs* of the previous frame and select the reusable VPLs for the current frame, which helps to reduce the flickers of illumination when the viewpoint, the foveal region or the scene changes. Our method computes high-quality illumination in the foveal region and low-quality illumination in the peripheral region. Our method supports indirect illumination at interactive frame rates with the current hardware, and is capable to handle dynamic scenes with deformable objects.

Figure 1 shows a comparison global illumination rendered with path tracing, our method and instant radiosity. We take the results of path tracing as the ground truth. Our method shows better global illumination effect in the foveal region than instant radiosity with better time efficiency. Our method achieves 42 FPS for each eye with an HTC Vive.

In summary, the contributions of our method are as follows:

- A foveated instant radiosity method for global illumination of dynamic scenes, which computes high-quality illumination in the foveal region and low-quality illumination in the peripheral region;
- A foveated importance computation method to evaluate VPLs' contribution to the foveal region;
- A VPLs management method to optimize the VPLs' distribution and reuse VPLs of the previous frame for better temporal stability and better performance.

2 Related Work

In this section, we first introduce the prior work for 3D foveated rendering in recent years. After that, we briefly discuss the previous radiosity rendering work, upon which our foveated instant radiosity method is based on. Finally, we discuss some temporal coherence based rendering methods, which are related to the temporal stability of our method.

Foveated 3D Rendering was proposed for VR HMD applications [8], which required a high-quality stereoscopic display without latency. The basic idea of the foveated rendering is to render the high-quality image in the foveal region and the low-quality image in the peripheral region to save time. One challenge of 3D foveated rendering is to process the geometry based on the attributes of the human visual system. This level of detail technique is used to keep the high-quality geometry in the sensitive region of human visual perception and simplify the objects or part of objects in the insensitive region [21, 22, 27, 30].

The most important and difficult problem of foveated rendering is to generate images with different qualities in different regions. The idea of multiple spatial resolutions was used in foveated rendering. Patney et al. [31] proposed a contrast enhancement method to recover the image details and a saccade-aware temporal antialiasing algorithm to address aliasing in the peripheral region. M. Stengel, et al. [41] introduced adaptive image-space sampling to use more visual cues in the perception model to generate the sampling pattern for sparse shading. Okan et al. [43] proposed a luminance contrast aware foveated method, which analyzed displayed luminance contrast and combined it with eccentricity to achieve spatial resolution reduction. X. Meng et al. [24] designed a kernel log-polar mapping algorithm for 3D graphics, which provided a framework with a controlled trade-off between visual quality and time cost for foveated rendering. Y. He et al. [11] introduced a general rendering pipeline to shade the pixels with multiple rates in one output image adaptively. There were also some smart sampling methods to generate multiple spatial resolution images for ray tracing [19, 23, 25, 45]. Swafford et al. [42] designed a user study and used a visible difference metric for high dynamic range images to evaluate the foveated rendering methods implemented with different techniques. Recently, Koskela et al. [18] proposed a novel regression-based reconstruction pipeline to accelerate ray tracing. It traces a single ray for each pixel and uses feature buffers to eliminate rendering noise. This method can achieve high-quality rendering at an interactive frame rate (30fps), but it is still not fast enough to render images in real time for both eyes of the VR HMD applications.

Besides the multiple spatial resolutions idea, multi-resolution color and temporary ideas can also be used to accelerate the foveated rendering. Duchowski et al. [6] analyzed the characters of color vision in the fovea and periphery, and proposed a level of color details framework, in which color degradation mapping for gaze-contingent display was constructed. Yee et al. [46] introduced an aleph map, which represented spatio-temporal error tolerance for dynamic scenes. The same idea can be adapted to the foveated rendering easily.

Our method is a foveated rendering method based on the multi-resolution illumination idea. More VPLs are used to calculate high-quality indirect illumination in the foveal region, while fewer VPLs are used to illuminate the peripheral area. To our knowledge, this is the first time that multi-resolution illumination has been applied to the foveated rendering.

Radiosity method can achieve high-quality illumination effects for scenes with diffuse surfaces. But it is very slow for complex scenes due to heavy pair-wise computation of form factors between meshes. The instant radiosity method [15] is an interactive method for efficiently estimating global illumination for 3D scenes between diffuse surfaces. In the instant radiosity method, it's important to distribute VPLs into the visually important regions of the scene. Many researchers worked on the strategies of VPLs distribution [5, 7, 14, 35, 37, 38, 40]. On the other hand, there is some previous work focusing on the calculation of VPL's contribution to

the output images. Segovia et al. [37] proposed a simple way that emited a certain amount of rays from the surfaces directly visible from the viewpoint to the VPLs and uses the visibility of those rays as the contribution. Hedman et al. [12] proposed a mail-boxing scheme to determine indirect visibility for static scenes and keep the VPLs which were indirectly visible from the viewpoint. These methods transform the contribution computation into indirect visibility determination, but they are time-consuming due to the ray tracing based visibility test.

Unlike the rasterization methods and ray tracing methods, radiosity methods can not be integrated into the foveated rendering framework easily because the multiple spatial resolution idea can not be applied to VPLs directly. Our method makes the instant radiosity method available in the framework of foveated rendering to render high-quality illumination effects for dynamic scenes in the foveal region.

Temporal coherence based rendering can reduce flickers between frames and accelerate rendering. Wald et al. [44] enforced temporal coherence by fixing the random number sequence to generate VPLs for each frame. In the context of VPL-based rendering, Aine et al. [20] moved VPLs over frames to guarantee temporal stability. H. Ki et al. [16] used light clustering, which allowed a smart selection of VPLs for more temporally stable indirect illumination. Hăsan et al. [10] grouped point lights into clusters and reused the shaded results from the clusters over multiple frames in an animated video. G. Nichols et al. [29] used multi-resolution splatting to reduce overdraw by rendering illumination at varying frequencies. Knecht et al. [17] improved the stability of the indirect illumination with temporal reprojection filtering. Prutkin et al. [32] and Baŕak et al. [2] devised the methods to improve the temporal stability of VPL sampling. Our foveated instant radiosity method uses temporal coherence in VPL management, which reuses as many VPLs of the previous frame as possible to reduce temporal errors between the adjacent frames and the shadow maps needed to be updated.

3 Foveated Instant Radiosity

First, we give a high-level algorithm of our foveated instant radiosity method. Then we describe the generation of VPL candidates (Section 3.1), explain the computation of VPLs' foveated importance (Section 3.2), and introduce the management of VPLs (Section 3.3). Finally, we describe rendering virtual scenes with all essential VPLs to obtain the indirect illumination effect (Section 3.4).

The basic idea of foveated rendering is to generate high-quality images in the foveal regions and low-quality images in the periphery regions. Therefore, we design a foveated rendering framework for instant radiosity to do multi-resolution illumination for different regions by controlling the number of VPLs cast on them, i.e. the foveal regions are lit by more VPLs, while the periphery regions are lit by fewer VPLs. VPL candidates are generated and chosen to render the scene for each frame. In order to maintain the temporally coherent illumination effects of the adjacent frames, a part of the essential VPLs of the previous frame are selected and kept to render the scene. Given a 3D scene, a viewpoint and a foveal region, the output image for each frame is rendered using Algorithm 1.

There are five main steps in this algorithm:

1. **Scene voxelization**. To generate VPLs efficiently, we voxelize the static scene with the method in [36] (line 1). For each voxel, the average 3D position, normal and color of all triangles intersecting the voxel are stored in V_s. We set the object ID for each voxel to 0, and we initialize the direct illumination color as black. For each frame, we first voxelize the dynamic objects and merge them with V_s into V (line 4). The voxelization of dynamic objects is the same as that of the static scene, except with different object IDs for these voxels. Then the direct illumination color of each voxel in V is updated based on the voxelization (line 5).

Algorithm 1 Foveated Instant Radiosity Method

Input: 3D static scene S, dynamic objects D, output viewpoint O, foveal region f
Output: $Framebuf$
1: $V_s \leftarrow Voxelization(S)$
2: $L_0 \leftarrow \varnothing$
3: **for** i = 1 to T **do**
4: $\quad V \leftarrow V_s \cup Voxelization(D)$
5: $\quad V \leftarrow UpdateLighting(V)$
6: $\quad L_c \leftarrow GenerateCandidates(V, O, f)$
7: $\quad I_{c\&(i-1)} \leftarrow FoveaImport(L_c, L_{i-1}, V, f, O)$
8: $\quad L_i \leftarrow Management(L_c, L_{i-1}, I_{c\&(i-1)})$
9: $\quad Framebuf \leftarrow RenderScene(L_i, S, O)$
10: **end for**

2. **VPL candidates generation**. We generate VPL candidates L_c with path tracing. The essential VPLs set L_0 is initialized as empty in line 2. For each frame, rays are emitted from the viewpoint to the foveal region and bounced once in the voxelized scene, and we place a VPL candidate at the end of each path (line 6).

3. **VPL foveated importance computation**. The foveated importance is determined by VPL's contribution to the foveal region. We compute the foveated importance $I_{c\&(i-1)}$ of all VPLs in the candidate set L_c and the essential VPL set L_{i-1} of the previous frame $frame_{i-1}$ (line 7). '&' in $I_{c\&(i-1)}$ means that we compute foveated importance for all VPLs in both L_c and L_{i-1}.

4. **VPL management**. VPL management is conducted to determine the essential VPLs L_i for the current frame $frame_i$ from L_c and L_{i-1}, according to their foveated importance (line 8).

5. **Rendering**. L_i is used to render the indirect illumination effect of the scene (line 9), which can be combined with the direct illumination to obtain the final global illumination result.

3.1 VPL Candidates Generation

VPL candidates are generated using the path tracing method, and the location of the foveal region is used to guide the generation of rays. Inspired by sequential Monte Carlo instant radiosity [12], we only use the rays emitted from the viewpoint and bounce them once to generate the VPL candidates. This is different from the original instant radiosity algorithm, which emits the rays from the light sources to generate VPLs where the rays intersect with the surface. The reason for this reverse tracing is to concentrate the VPLs that cast light on the foveal regions .

We consider that the areas in the foveal region are equally important for visual perception, and the periphery region is less important. Therefore, we render the scene with the conventional rendering pipeline and sample the foveal region uniformly to get the 2D sample points in the image plane. The rays are emitted from the viewpoint to these sample points and reflected in random directions to intersect with the voxelized scene. The intersections are used to place VPL candidates, which are not necessarily visible to the viewpoint.

The previous foveated rendering methods always generate a transition region between the foveal area and periphery area by decreasing the sampling rate gradually to smooth the rendering result and avoid noticeable sudden quality changes. However, since our method considers indirect lighting, the foveal region receives all VPLs' lighting, and the region outside the fovea receives less VPLs' lighting, so the rendering result will naturally be smoothed. The transition is therefore handled in our method implicitly.

3.2 VPL Foveated Importance Computation

We compute the foveated importance for each VPL in the candidate set L_c and the essential VPL set L_{i-1} of the previous frame to measure its lighting contribution to the foveal region of the output image. Inspired by previous work of calculating of VPL's contribution to the output images [12, 37], we propose a VPL foveated importance that indicates the VPL's lighting contribution to the foveal region of the output image.

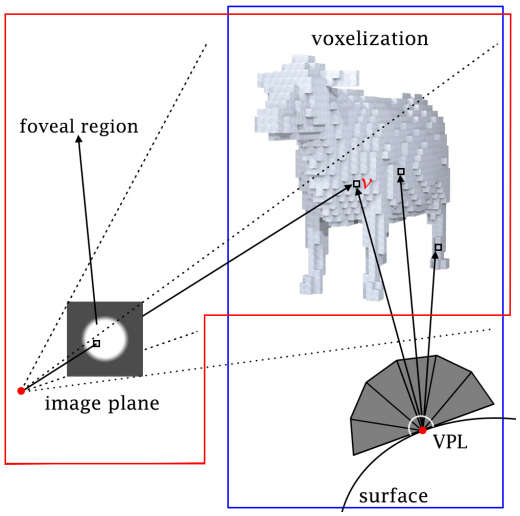

Figure 2: Computation of foveated weight per voxel and foveated importance per VPL. The part in the red polygon shows the process of voxel foveated weight estimation. Each visible voxel in the scene voxelization from the current viewpoint is projected on the image plane to obtain the foveated weight. The part in the blue box shows the process of VPL foveated importance calculation. The rays cast from the VPL intersect with the voxelization, and the foveated importance of the VPL is calculated with the voxel foveated weights of the intersected voxels.

The foveated importance of VPL can be computed in two steps:

1. Voxel foveated weight estimation.
2. VPL foveated importance calculation.

In Figure 2, the part in the red polygon shows the process of Step 1. We estimate a foveated weight for each visible voxel in the voxelized scene from the current viewpoint O. The foveated weight of the voxel is defined to be inversely proportional to the distance from its projection to the center of the foveal region on the image plane. In our implementation, a foveated weight map is generated on the image plane [41] and each visible voxel is projected onto this map from O to get the value of the foveated weight. The foveated weights of other voxels are set to 0.

The process of Step 2 is shown in the blue box of Figure 2. We calculate the foveated importance for each VPL based on the foveated weights of the voxels visible from the VPL. The foveated importance of a VPL indicates an approximate contribution of the VPL to output image, especially to the foveal region. VPLs with high foveated importance means they are important and they contribute more lighting to the visible surfaces of the scene projected inside the foveal region. To determine visible voxels from each VPL, we generate 6 rays to different directions uniformly. Then, we intersect these rays with the voxels, get the foveated weights of the first intersected voxels, and compute the foveated importance of the VPL by a weighted accumulation of the voxels' weights. The cosine of the angle between the ray and the VPL's normal is used as the weight according to the method in [4].

3.3 VPL Management

After all VPL candidates are generated for the current frame, we perform a VPL management to select the optimized set of VPLs with the largest foveated importance (defined in Section 3.2) to render high-quality indirect illumination in the foveal region, and we name them essential VPLs. We observe that the following rules are helpful to determine the essential VPLs:

Rule 1: The VPLs with large contributions should be added to the essential VPL list;

Rule 2: The essential VPLs of the previous frame should be reused to maintain temporal coherence;

Rule 3: The distribution of VPLs should be as uniform as possible for stable illumination effects in each frame.

Based on these rules, we introduce a foveated importance based VPL management method, which chooses the essential VPLs for the current frame from the essential VPLs L_{i-1} of the previous frame and the VPL candidates L_c of the current frame.

Before selecting the essential VPLs from L_{i-1}, the VPLs in L_{i-1} need to be updated as the scene geometry may change. For each VPL in L_{i-1}, we first determine whether it is on a dynamic object and update the position of the VPL based on the transformation matrix of the corresponding dynamic object. We update the VPL's normal based on the voxel's normal of the current frame and compare it with the VPL's normal of the previous frame. If the normal direction of the VPL changes significantly, the VPL is deleted from L_{i-1}. Then we recompute the foveated importance and direct illumination color of the VPL in L_{i-1}. After this, the importance change rate C of the VPL is obtained by comparing the current foveated importance I_i and the previous one I_{i-1}. The probability that the VPL can be selected p is calculated with Equation 1.

$$p = min\{C = \frac{I_i}{I_{i-1}}, 1\}. \quad (1)$$

If the foveated importance of a VPL increases, the VPL will always be accepted and added into the essential VPL set L_i of the current frame. Otherwise, it will be accepted and added into L_i with probability p.

We also select some new VPLs from the VPL candidates L_c generated in Section 3.1 since the viewpoint or the scene may change, and the essential VPLs in L_{i-1} may not illuminate the newly visible area from the current viewpoint. Before the selection, we compute the foveated importance for each VPL in L_c and probabilistically remove some VPLs with low importance using the classic cumulative distribution function based inverse transform method [1].

After this, we compare the VPLs in the candidates set L_c and the un-selected VPL set of the previous frame L'_{i-1}, and further select the essential VPLs from them for the current frame. The VPLs in L'_{i-1} were not added into L_i when we select the essential VPLs from L_{i-1}, but now still have a chance to be selected. We do this for two reasons: one is based on Rule 2 that more essential VPLs of the previous frame should be reused to maintain temporal coherence; the other is that in the previous steps, the VPLs of the previous frame have not been compared with the newly generated candidate VPLs, and we hope that the old VPLs with higher importance and lower density than the current candidate VPLs can be reused. The process of VPL selection from L_c and L'_{i-1} is in Algorithm 2.

The algorithm has two parts. Part 1(lines 1-20) is to select VPLs from L_c and L'_{i-1} by comparing their foveated importance and density:

1: A uniform grid is generated in the 3D space to control the density of the essential VPLs for the current frame. Cell density is an attribute of the cell, which is determined by the number of the

Algorithm 2 VPL selection from L_c and $L_{i-1}^{'}$

Input: $L_i, L_c, L_{i-1}^{'}, I_{c\&(i-1)}$
Output: L_i
1: $D \leftarrow CellDensity(L_i)$
2: **for** each VPL l in $L_{i-1}^{'} \cup L_c$ **do**
3: $W[l] \leftarrow I_{c\&(i-1)}[l] + 1/(getDensity(D,l) + CONST)$
4: **end for**
5: $SortAscending\left(L_{i-1}^{'}, W\right)$
6: $SortDescending\left(L_c, W\right)$
7: $N \leftarrow min\left(|L_{i-1}^{'}|, |L_c|\right)$
8: **for** $j = 1$; $j < N$; $j++$ **do**
9: **if** $W[L_c[j]] > W[(L_{i-1}^{'}[j]]$ **then**
10: **if** $getDensity(D, L_c[j]) < D\_MAX$ **then**
11: $L_i \leftarrow L_i \cup \{L_c[j]\}$
12: $Update(D, L_c[j])$
13: **end if**
14: **else**
15: **if** $getDensity(D(L_{i-1}^{'}[j]) < D\_MAX$ **then**
16: $L_i \leftarrow L_i \cup \{(L_{i-1}^{'}[j]\}$
17: $Update(D, (L_{i-1}^{'}[j])$
18: **end if**
19: **end if**
20: **end for**
21: **for** $j = N$; $j < |L_c|$; $j++$ **do**
22: **if** $|L_i| < N\_MAX \& getDensity(D, L_c[j]) < D\_MAX$ **then**
23: $L_i \leftarrow L_i \cup L_c[j]$
24: $Update(D, L_c[j])$
25: **end if**
26: **end for**
27: **return** L_i

VPLs located in this cell. VPL density is an attribute of the VPL, which measures the number of other VPLs around this VPL. In our implementation, for a given VPL, we first determine which cell it locates, and then estimate the VPL density using the cell density (line 1);

2: For each VPL in L_c and L_{i-1}, we use its importance and VPL density to calculate its parameter W (lines 2-4). $getDensity(D,l)$ is to get the density of the cell where l locates in from D. CONST is a constant to avoid division by 0;

3: We sort the VPLs in $L_{i-1}^{'}$ in ascending order of W and sort the VPLs in L_c in descending order (lines 5-6). Then, we compare W of the VPLs with the same index in the sorted L_c and $L_{i-1}^{'}$, choose the ones with larger W values and add them to L_i if the density of the cell where the selected VPL locates in is smaller than a predefined threshold D_MAX (lines 7-20). At last, the corresponding cell density is updated with $Update(D,l)$ (lines 12, 17).

Part 2 (lines 21-26) of the algorithm is to add more VPLs from L_c to L_i if the VPL number of L_i does not meet the maximum VPL number N_MAX of the current frame. After the VPL management, we now have the essential VPLs for the current frame.

3.4 Rendering

We use the essential VPLs in L_i to simulate light transport in the scene. To compute the indirect illumination of the world position s of the shading point inside each pixel on the output image, instant radiosity approximates the reflected radiance $R(s,\omega_o)$ in direction ω_o with the essential VPLs:

$$R(s,\omega_o) = \sum_{l \in L_i} R_l(s,\omega_o) Vis(l,s), \tag{2}$$

where $R_l(s,\omega_o) = \rho \frac{\frac{\Phi_l}{\pi} cos(\theta_l) cos(\theta_s)}{d_l^2(s)}$. $d_l(s)$ is the distance between l and s. θ_l and θ_s are the angles between the normals of l and s and the transmission direction respectively. $Vis(l,s)$ is the binary visibility between l and s. ρ is the diffuse reflectivity. Φ_l is the radiant intensity of l.

A limitation of the VPL-based methods is that singularities occur when the VPLs and pixel samples are close to each other. The clamping method is always used to avoid these artifacts [12, 20]. In our implementation, we set a minimum clamp distance for each VPL, which means that if the distance from a VPL to a shading point is less than the minimum clamp distance, we use this pre-defined minimum clamp distance to compute the illumination of the VPL.

In order to compute the direct illumination at s with a VPL l, l's radiant intensity and visibility seen from the shading point need to be evaluated. To do this, a voxel mipmap is generated as a hierarchical structure of the scene voxelization, which records the average direct illumination and foveated importance of the voxels. Then, voxel cone tracing [4] is used to accelerate the intersection computation to obtain the radiance of l. For each l, six cones are emitted uniformly towards l's hemisphere and intersected with the voxel mipmap. The radiant intensity of l is determined with Equation 3.

$$\Phi_l = \sum_{i=1}^{6} ConeTracing\left(D_i\right).color \times D_i.weight, \tag{3}$$

where D_i is the i-th direction of 6 directions uniformly distributed on the hemisphere of the VPL. $ConeTracing\left(D_i\right).color$ is the color of the intersection between the cone in D_i and the mipmap of the voxelization. $D_i.weight$ is the weight of D_i, which is set as the cosine of the angle between D_i and the VPL's normal. Our method also can support multi-bounce indirect illumination by integrating the multi-bounce VPL radiance estimation.

The visibility of a VPL is estimated with the paraboloid shadow mapping method [3]. It is still time-consuming if we update all shadow maps of the reused VPLs in L_i. So, we only update the shadow maps of the reused VPLs with the top 5% foveated importance for each frame. Furthermore, we adopt the deferred shading and the interleaved sampling method [20] to reduce the computational cost and accelerate rendering.

In VR, the same essential VPLs are used to render a stereo pair of images, with almost all of the objects visible in the left eye also showing up in the right eye view. We shade the full left-eye view, and then render the right-eye view by sampling from the completed left-eye view using reprojection [28]. The right eye view only has to shade new texels in the case that no valid sample was found, rather than recalculating everything [26].

4 Results and Discussion

We used an HTC Vive (tracker, HMD, and wireless hand-held controller) with a Droolon aGlass to track the head motion and the foveated point of the user. The Vive was connected to a PC workstation with a 3.8 GHz Intel(R) Core(TM) i7-9800X CPU, 16 GB of memory, and an NVIDIA GeForce GTX 2080 Ti graphics card. We tested our method on five scenes: Sponza (279k tris., Figure 1), Cornellbox (5.8k tris., Figure 3, row 1), Room (476k tris., row 2), Yard (516k tris., row 3) and Balcony (192k tris., row 4). We use path tracing to generate ground truth images, specifically, we use the Path Tracer Integrator with 2000 spp, tracing depth with 4 in Mitsuba [13]. We compared our method with instant radiosity method in quality and performance. Instant radiosity method is implemented by the same framework as our method except the VPLs are generated from the light sources. The number of VPLs used in both our method

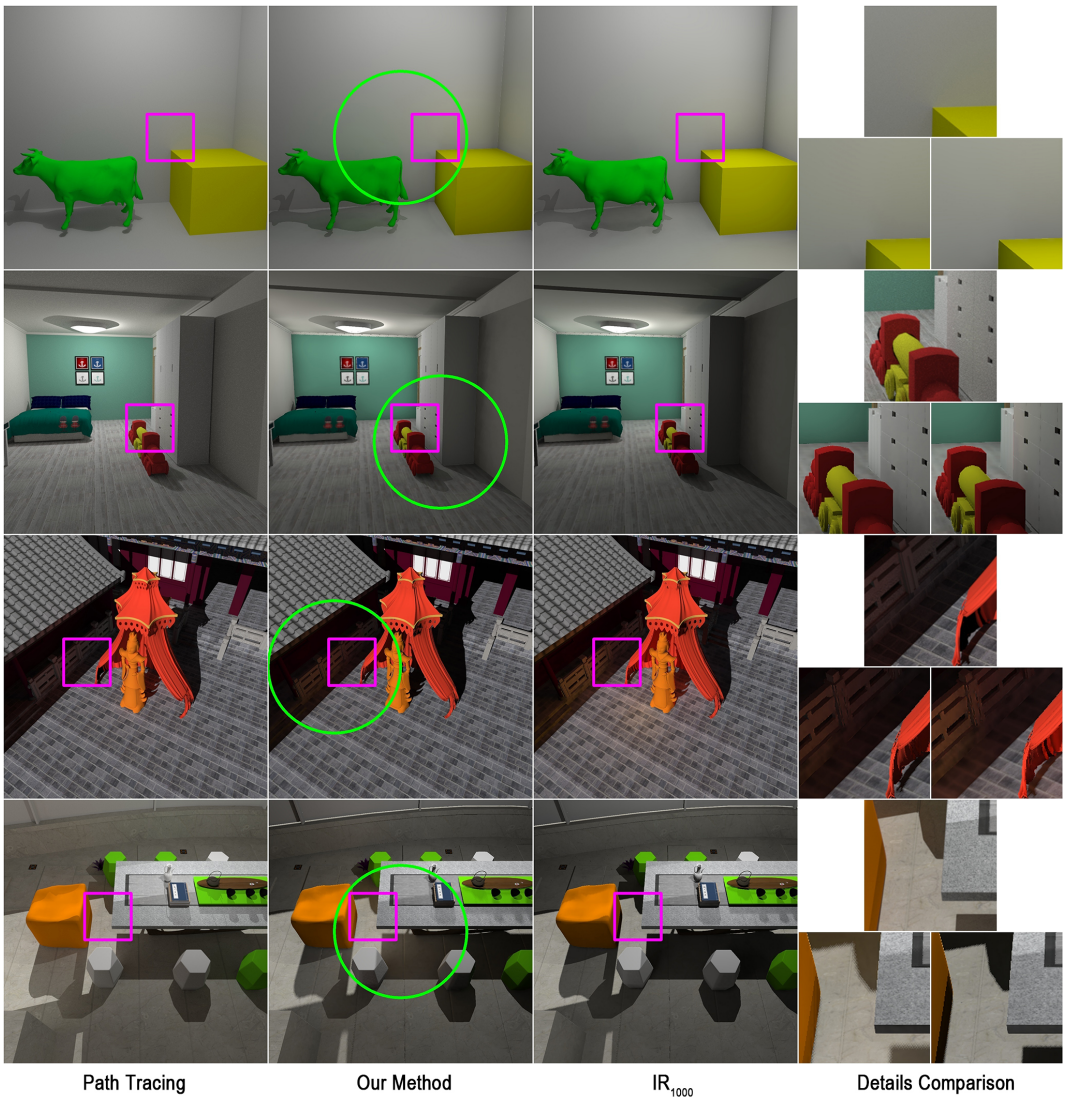

Figure 3: Comparison between path tracing (column 1), our method (column 2) and IR_{1000} (column 3). The details in the magenta square are magnified in column 4 (top: path tracing; bottom-left: ours; bottom-right: IR_{1000}).

and instant radiosity is 1000 (IR_{1000}). For direct illumination from actual light sources, the shadow map resolution is 2048×2048. For VPLs, the shadow map resolution is 128×128. The resolution of the output images is 1024×1024. According to the linear model of minimum angular resolution for human visual acuity in [8] and the parameters of HTC Vive, such as the focus distance and the dots per inch, the radius of the foveal region should be no less than 120 pixels. We used a more conservative, larger foveate radius to prevent users from noticing quality degradation in the periphery region. In our implementation, the foveal radius is set to be $1/4$ of the width of the output image. We also analyzed the effect of various shadow map resolutions, VPL numbers and foveal radii in this section.

4.1 Quality

The rendering results of path tracing (column 1), our method (column 2) and IR_{1000} (column 3) are shown in Figure 3. The green circles on the images rendered with our method indicate the foveal regions. We also crop and magnify the details inside the foveal region for comparison in column 4. Our results are closer to the ground truth image than instant radiosity results. This is because there are several problems clearly seen in instant radiosity: 1) there is no color bleeding from the yellow cube to the white wall (row 1); 2) the color of the train bleeds excessively on the closet (row 2); 3) the orange on the barrier is brighter than the reference (row 3); 4) there should be orange color bleeding on the ground (row 4).

We also quantified the quality with two metrics: the mean squared error MSE for all pixels of the results compared to the ground truth and the average temporal error MSE', which is defined as the average mean squared error of all pixels between the (n-1) consecutive frame pairs with Equation 4. We measured MSE and MSE' for the foveal region and periphery region respectively.

$$MSE' = \frac{\sum_{i=2}^{n} MSE_{i,i-1}}{n-1} \tag{4}$$

Table 1 shows the comparison of the MSE and MSE' of our method and the instant radiosity method for the images in Figure 1 bottom and Figure 3. IR_{1000} represents the conventional instant radiosity method, which generates VPLs from the light sources. IR_v represents the instant radiosity method whose VPLs are generated from the viewpoint. We do not manage VPLs for the instant radiosity

Table 1: MSE and MSE' ($\times 10^{-3}$) in the foveal and periphery regions with different methods.

Scene	MSE in Fovea				MSE in Periphery				MSE' in Fovea				MSE' in Periphery			
	IR		Our method		IR		Our method		IR		Our method		IR		Our method	
	IR_{1000}	IR_v	$Ours_r$	$Ours$	IR_{1000}	IR_v	$Ours_r$	$Ours$	IR_{1000}	IR_v	$Ours_r$	$Ours$	IR_{1000}	IR_v	$Ours_r$	$Ours$
Sponza	3.16	7.63	2.50	1.09	4.84	6.28	4.65	3.89	0.06	0.20	0.18	0.04	0.74	1.10	1.50	0.83
Cornellbox	1.61	2.47	0.92	0.74	1.59	3.36	2.07	1.12	0.14	0.08	0.14	0.04	0.24	0.24	0.21	0.14
Room	2.19	6.27	2.79	1.08	3.18	8.85	2.85	1.71	0.12	0.14	0.26	0.07	0.46	0.40	0.76	0.39
Yard	1.75	2.38	2.26	1.01	1.93	2.52	2.66	2.46	0.33	0.53	0.14	0.06	0.84	0.88	0.97	0.97
Balcony	8.61	6.13	4.65	4.30	7.43	6.20	6.46	5.90	1.37	0.87	0.34	0.31	0.44	0.86	0.60	0.61

method. We also tested our method with two conditions: $Ours_r$ selects the reusable VPLs from the essential VPLs of the previous frame randomly; $Ours$ selects the reusable VPLs from the essential VPLs of the previous frame with our method based on the foveated importance in Section 3.3.

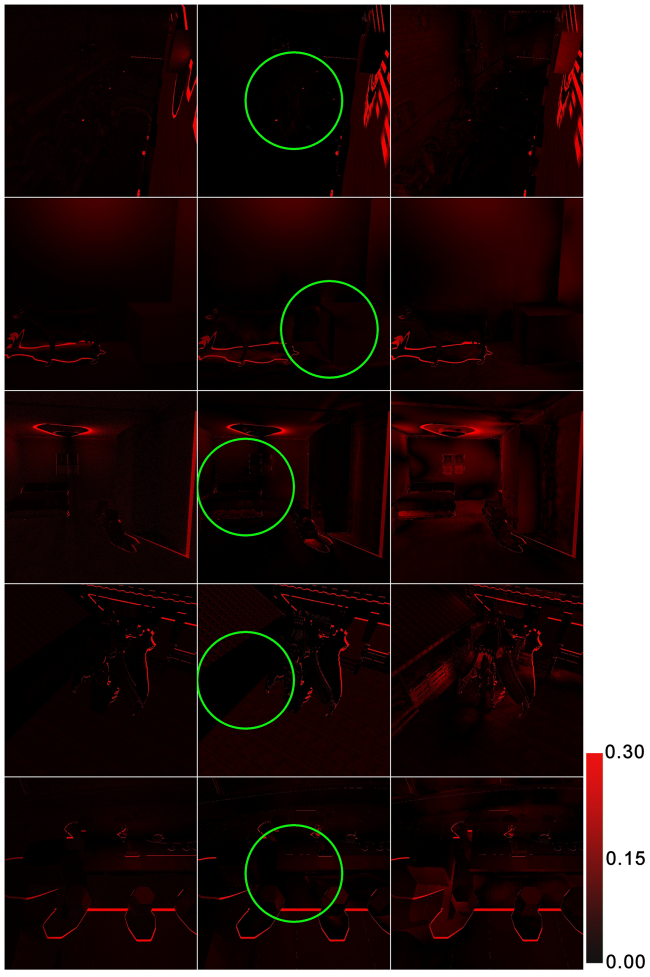

Figure 4: Visualization of temporal mean squared errors for the images rendered with path tracing (column 1), our method (column 2) and instant radiosity (column 3). The green circles in column 2 represent the foveal region of the user.

The MSE values in the foveal regions of our method are consistently smaller than those in the periphery regions. In contrast, with instant radiosity methods (IR_{1000} and IR_v), the results are not consistent. This is because our method selects the VPLs based on their foveated importance and gives more benefits to the foveal region. Instant radiosity methods consider all regions in the output image as the same, so the errors of different regions do not exhibit a certain pattern. Although both IR_v and our method generate VPLs from the viewpoint, there is a difference: IR_v emits rays to the whole output image and bounces them into the scene to generate VPLs; our method only emits rays to the foveal region of the output image and bounces, and the VPLs are guaranteed to contribute to the foveal region. Therefore the MSE values in the foveal regions of $Ours$ are smaller than the results of IR_v. Also, in our method, the MSE values in the foveal region of $Ours$ are smaller than those of $Ours_r$, which indicates that the reusable VPLs selection strategy based on the foveated importance is more effective than random selection.

Next, we compared the average temporal errors MSE' of our method and the instant radiosity method in the foveal region and the periphery region. As expected, in the foveal region, the MSE' values of IR_{1000}, IR_v and $Ours_r$ do not have much difference over all the 5 different scenes. While the MSE' values of $Ours$ are always less than those of IR_{1000}, IR_v and $Ours_r$ for all scenes. This is because IR_{1000} and IR_v do not consider the temporal coherence. They regenerate VPLs from scratch for each frame, and can not maintain good temporal stability for the adjacent frames. In our method, both $Ours_r$ and $Ours$ reuse the essential VPLs of the previous frame. The only difference is that $Ours_r$ selects the reusable VPLs randomly, while $Ours$ selects the reusable VPLs based on the foveated importance computation in Section 3.3. Thus, the comparison indicates that our foveated importance based reusable VPLs selection can keep better temporal stability for the adjacent frames. Also, for our method, the temporal errors of the foveal regions are smaller than the errors of the periphery regions due to the foveated importance computation. However, with instant radiosity methods, larger temporal errors may appear in any different regions for different scenes.

The images in Figure 4 visualize the temporal mean squared errors for the images in Figure 1 bottom and Figure 3 when moving the light source slightly. The darker pixels represent the smaller error. Compare to instant radiosity, the temporal error visualization of our method is more similar to the result of path tracing. The temporal error visualization shows that our method has less error than instant radiosity, especially in the foveal regions, which shows that our method reduces the temporal error and maintains high time stability between adjacent frames.

Several parameters of our method affect the image quality in the foveal and periphery regions. Figure 5 shows MSE of our method as functions of the resolution of shadow maps, the VPL number, and the foveal radius. (a) shows the errors as a function of the shadow map resolution. For both foveal and periphery regions, the errors decrease slightly when the resolution of the shadow map becomes large. Although high-resolution shadow maps provide more accurate visibility of the VPLs, pixel values blended from lower quality illumination of 1000 VPLs are sufficiently good. (b) shows the errors as a function of the number of VPLs. In both regions, errors decrease when the number of VPLs increases. That is because more VPLs bring better indirect illumination. It also can be seen from (b) that the errors of the foveal region decrease faster than those of the periphery region. This is because our method optimizes the VPL distribution according to the location of the foveal region, so the increased VPLs bring more benefits to the foveal region than the periphery region. (c) shows the errors as a function of the

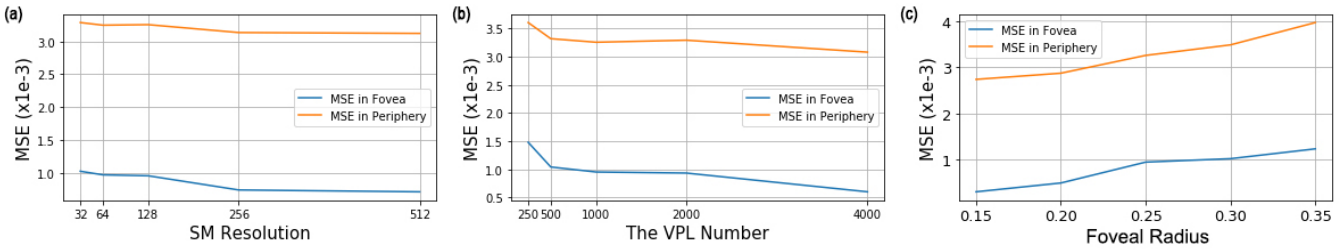

Figure 5: MSE of Sponza as a function of (a) the resolution of shadow maps, (b) the VPL number and (c) the foveal radius.

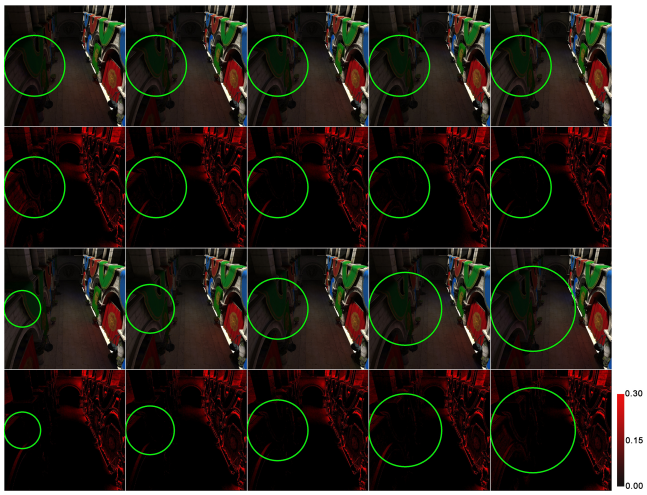

Figure 6: Rendered effects and pixel error visualization of our method with the different VPL numbers: 250, 500, 1000, 2000, 4000 and with the different foveal radii: 0.15, 0.20, 0.25, 0.30, 0.35. The green circles in the image represent the foveal region of the current scene.

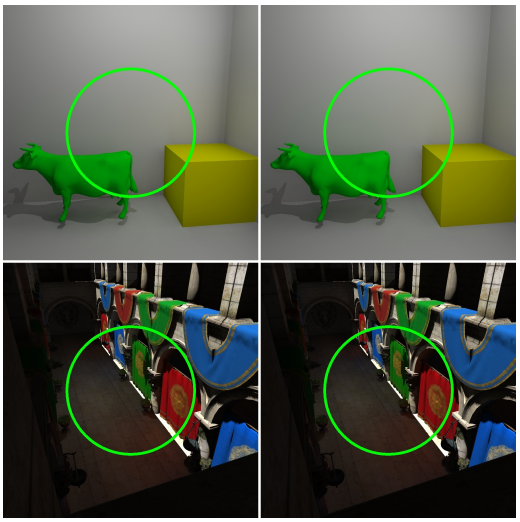

Figure 7: Comparison of the images of our method with cosine (left) and linear (right) term in Equation 1. No obvious visual difference between the results of using cosine and linear terms. The MSE values of the foveal and peripheral regions are also similar with these two different terms. MSE (in fovea, in periphery, $\times 10^{-3}$): Cornellbox(left): (0.74, 1.12), Cornellbox (right): (0.74, 0.95); Sponza (left): (2.42, 4.80), Sponza (right): (2.49, 4.99).

foveal radius. The errors increase when the foveal radius becomes large. This is because the quality of an image is proportional to the density of the VPLs, and a smaller foveal region has a denser VPL distribution.

The images rendered with different numbers of VPLs (row 1), with different fovea sizes (row 3) are shown in Figure 6. The corresponding mean squared errors are visualized in row 2 and 4, where darker colors indicate smaller errors. As can be seen from the images in row 2, as the number of VPLs increases, there are more black pixels in the foveal region indicated by the green circle, which means that the error is decreasing. The images in row 4 show that as the radius of the foveal region increases, the black pixels in the circle become less, that is, the error in the fovea increases.

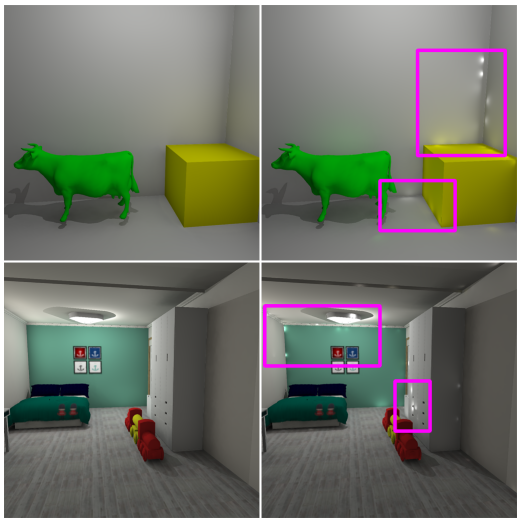

Figure 8: Comparison of the images of our method with clamping (left) and without clamping (right). The clamped images are smooth, while the un-clamped images have some artifacts in the magenta squares. Some spiky artifacts appear at the corner of the walls and on the surface of the yellow cube in Cornellbox, at the corner of the roof and on the closet door in Room.

In VPL management (Section 3.3), we claimed that if the cosine curve was used in filtering the reusable VPLs, VPLs with small importance changes were easier to retain, and VPLs with larger changes were more likely to be deleted. Here we briefly verify our choice against using a cosine falloff of linear term, i.e. $1 - \cos\left(\frac{\pi}{2} \times C\right)$ was used to replace C in Equation 1. However, the comparison in Figure 7 shows that there is no significant difference between the results of using cosine and linear terms in the calculation, and the MSE values of the foveal and peripheral regions are also similar. Therefore, we choose linear term in the implementation to simplify the calculation.

Figure 8 shows images of our method with and without clamping

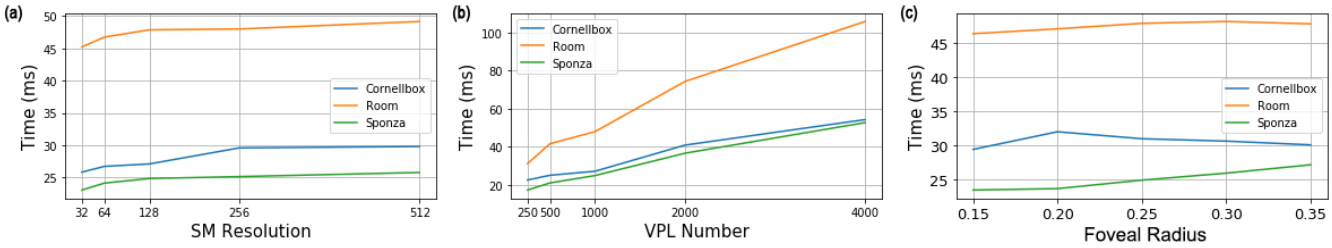

Figure 9: Time cost for Sponza as a function of (a) the resolution of shadow maps, (b) the VPL number and (c) the foveal radius.

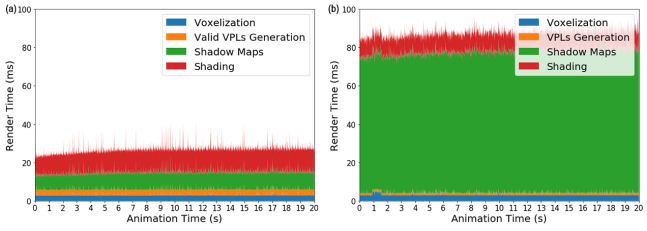

Figure 10: Rendering time of our method (left), instant radiosity (IR_{1000}, right) for Sponza.

Table 2: Performance of our method compared to instant radiositywith the different VPL numbers (ms)

Scene	Cornellbox	Room	Sponza	Yard	Balcony
Ours	13	26	24	28	27
vs. IR_{1000}	3×	3×	3×	3×	4×
vs. IR_{eq}	11×	13×	12×	13×	10×
Num. of VPL	2700	3000	2700	2400	2400

(Section 3.4). The un-clamped image (right) has some spiky artifacts in the corner of the walls and the surface of the yellow cube in Cornellbox, the corner of the roof and the closet door in Room. This is because if the VPLs and pixel samples are close to each other, the VPL-based method will have singularities. We set a minimum clamping distance to avoid this singularity, for example, 1.5m in our implementation. If the distance between the pixel and the VPL is less than 1.5m, we set the distance to 1.5m. Therefore, the clamped images (left) do not have any unreasonable highlights.

4.2 Performance

Table 2 shows the frame rendering time of our method and the speedup versus the instant radiosity method. Compared with the instant radiosity method with 1000 VPLs IR_{1000} (still with much worse quality than ours), our method already achieves an acceleration of 3-4 ×. We also performed an equal quality comparison of instant radiosity IR_{eq}. Compared to IR_{eq}, our method achieves a speedup of 10-13×. The VPL numbers are given in the last row of Table 2, which achieve similar MSE values in the foveal regions to our method.

For the stereo rendering using VR HMDs, the time cost for rendering the second view is 12ms for Sponza, 8ms for Cornellbox, 12ms for Room, 11ms for Yard, 12ms for Balcony. The corresponding average numbers of new pixels to be shaded in the second view are only 8%, 9%, 8%, 6% and 7%, thanks to our reprojection scheme.

Figure 9 (a), (b) and (c) show the frame rendering time for our method as a function of the resolution of shadow maps, the VPL number and the foveal radius. As expected, the frame time increases when rendering large shadow maps of essential VPLs and when using more VPLs. The performance shown in (c) indicates the time cost varies little with the foveal radius since we keep approximately the same number of VPLs and shadow maps for different foveal radii.

Figure 10 visualizes the time spent on each step separately for the Sponza using our method (left) and IR_{1000} (right). The voxelization and shading time for the two methods are almost the same. Due to VPL's importance computation and management, our essential VPLs generation is 2-3× slower than IR_{1000}. However, since our VPL management scheme, and we only update the shadow maps of the reused VPLs with the top 5% foveated importance for each frame, our method can save 80% of the time cost.

5 Conclusion, Limitations and Future Work

We have presented a foveated instant radiosity method for rendering high-quality illumination effects in the foveal region based on optimizing the distribution of VPLs over frames. Our method is based on the idea of multi-resolution illumination and has both quality and performance advantages over the instant radiosity method. Our method produces smooth illumination effects, suppresses the temporal artifacts caused by sudden changes of VPLs, and achieves better temporal stability than the instant radiosity method. Our method also supports a variety of dynamic scenes, thanks to our VPL reusing scheme.

One limitation of our method is that our method doesn't work well for scenes with rapidly moving objects. Our method reuses VPLs temporally. Like any other temporal methods [12], when temporal information is hard to exploit, for example, the dynamic objects move too fast, flickers may appear and the pixel errors may increase. We have tested the plane with various moving speeds (range from 100 to 150 pixels/s in the screen space) in Sponza. The MSE' in fovea is less than 0.06×10^{-3} when the moving speed of the plane is about 130 pixels/s, and the visual effect is smooth. It turns out that our method can faithfully achieve a relatively smooth color bleeding effect for common VR applications with moving objects at medium speeds at least. One possible future solution is to select VPLs from both the light sources and the viewpoint, which may generate more stable overall illumination. Another limitation is that strong visual acuity in the periphery region may be ignored by foveated rendering in some cases. So another possible future direction is to extend the foveated importance to account for a broader range of visual acuity. Also, since our method is based on instant radiosity, it shares the same limitations with the original instant radiosity, which can only render the objects with the diffuse surfaces. Our method makes no assumption of specific types of energy-carrying particles in instant radiosity. We believe that it is also possible to combine our method with orthogonal methods [9] that extends instant radiosity to handle glossy reflections, e.g. replacing VPLs with virtual spherical lights (VSLs) through a different way of calculating the foveated importance of VSLs according to their "glossy lobes".

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China through Projects 61932003 and 61772051, by National Key RD plan 2019YFC1521102, by the Beijing Natural

Science Foundation L182016, by the Beijing Program for International ST Cooperation Project Z191100001619003, by the funding of Shenzhen Research Institute of Big Data(Shenzhen 518000).

REFERENCES

[1] D. Babahenini, A. Gruson, M. C. Babahenini, and K. Bouatouch. Efficient inverse transform methods for vpl selection in global illumination. *Multimedia Tools and Applications*, 77(11):13571–13595, 2018.

[2] T. Barák, J. Bittner, and V. Havran. Temporally coherent adaptive sampling for imperfect shadow maps. In *Proceedings of the Eurographics Symposium on Rendering*, EGSR '13, p. 87–96. Eurographics Association, Goslar, DEU, 2013. doi: 10.1111/cgf.12154

[3] S. Brabec, T. Annen, and H.-P. Seidel. Shadow mapping for hemispherical and omnidirectional light sources. In *Advances in Modelling, Animation and Rendering*, pp. 397–407. Springer, 2002.

[4] C. Crassin, F. Neyret, M. Sainz, S. Green, and E. Eisemann. Interactive indirect illumination using voxel cone tracing. In *Computer Graphics Forum*, vol. 30, pp. 1921–1930. Wiley Online Library, 2011.

[5] T. Davidovič, J. Křivánek, M. Hašan, P. Slusallek, and K. Bala. Combining global and local virtual lights for detailed glossy illumination. In *ACM Transactions on Graphics (TOG)*, vol. 29, p. 143. ACM, 2010.

[6] A. T. Duchowski, D. Bate, P. Stringfellow, K. Thakur, B. J. Melloy, and A. K. Gramopadhye. On spatiochromatic visual sensitivity and peripheral color lod management. *ACM Transactions on Applied Perception (TAP)*, 6(2):9, 2009.

[7] I. Georgiev and P. Slusallek. Simple and robust iterative importance sampling of virtual point lights. In *Eurographics (Short Papers)*, pp. 57–60, 2010.

[8] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3d graphics. *ACM Transactions on Graphics (TOG)*, 31(6):164, 2012.

[9] M. Hašan, J. Křivánek, B. Walter, and K. Bala. Virtual spherical lights for many-light rendering of glossy scenes. In *ACM Transactions on graphics (TOG)*, vol. 28, p. 143. ACM, 2009.

[10] M. Hašan, E. Velázquez-Armendáriz, F. Pellacini, and K. Bala. Tensor clustering for rendering many-light animations. In *Computer Graphics Forum*, vol. 27, pp. 1105–1114. Wiley Online Library, 2008.

[11] Y. He, Y. Gu, and K. Fatahalian. Extending the graphics pipeline with adaptive, multi-rate shading. *ACM Transactions on Graphics (TOG)*, 33(4):142, 2014.

[12] P. Hedman, T. Karras, and J. Lehtinen. Sequential monte carlo instant radiosity. *IEEE transactions on visualization and computer graphics*, 23(5):1442–1453, 2017.

[13] W. Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[14] H. W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., USA, 2001.

[15] A. Keller. Instant radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, p. 49–56. ACM Press/Addison-Wesley Publishing Co., USA, 1997. doi: 10.1145/258734.258769

[16] H. Ki and K. Oh. A gpu-based light hierarchy for real-time approximate illumination. *The Visual Computer*, 24(7-9):649–658, 2008.

[17] M. Knecht, C. Traxler, O. Mattausch, W. Purgathofer, and M. Wimmer. Differential instant radiosity for mixed reality. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pp. 99–107. IEEE, 2010.

[18] M. Koskela, K. Immonen, M. Mäkitalo, A. Foi, T. Viitanen, P. Jääskeläinen, H. Kultala, and J. Takala. Blockwise multi-order feature regression for real-time path-tracing reconstruction. *ACM Transactions on Graphics (TOG)*, 38(5):1–14, 2019.

[19] M. Koskela, A. Lotvonen, M. Mäkitalo, P. Kivi, T. Viitanen, and P. Jääskeläinen. Foveated real-time path tracing in visual-polar space. *Eurographics Symposium on Rendering - DL-only and Industry Track*, 2019. https://diglib.eg.org/handle/10.2312/sr20191219. doi: 10.2312/sr.20191219

[20] S. Laine, H. Saransaari, J. Kontkanen, J. Lehtinen, and T. Aila. Incremental instant radiosity for real-time indirect illumination. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp. 277–286. Eurographics Association, 2007.

[21] T. Lindeberg. Concealing rendering simplifications using gaze contingent depth of field. Master's thesis, KTH, School of Computer Science and Communication (CSC), 2016. http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A947325dswid=452.

[22] D. Luebke and B. Hallen. Perceptually driven simplification for interactive rendering. In *Rendering Techniques 2001*, pp. 223–234. Springer, 2001.

[23] R. Mantiuk, B. Bazyluk, and A. Tomaszewska. Gaze-dependent depth-of-field effect rendering in virtual environments. In *International Conference on Serious Games Development and Applications*, pp. 1–12. Springer, 2011.

[24] X. Meng, R. Du, M. Zwicker, and A. Varshney. Kernel foveated rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):5, 2018.

[25] E. N. Molenaar. Towards real-time ray tracing through foveated rendering. Master's thesis, 2018.

[26] H. Moreton and N. Stam. Turing texture space shading, 2018. http://https://devblogs.nvidia.com/texture-space-shading.

[27] H. Murphy and A. T. Duchowski. Gaze-contingent level of detail rendering. *EuroGraphics 2001*, 2001.

[28] D. Nehab, P. V. Sander, J. Lawrence, N. Tatarchuk, and J. R. Isidoro. Accelerating real-time shading with reverse reprojection caching. In *Graphics hardware*, vol. 41, pp. 61–62, 2007.

[29] G. Nichols and C. Wyman. Multiresolution splatting for indirect illumination. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pp. 83–90. ACM, 2009.

[30] D. Parkhurst, I. Law, and E. Niebur. Evaluating gaze-contingent level of detail rendering of virtual environments using visual search. In *Symposium on Eye Tracking Research and Applications (ETRA)*, pp. 105–109, 2000.

[31] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):179, 2016.

[32] R. Prutkin, A. Kaplanyan, and C. Dachsbacher. Reflective shadow map clustering for real-time global illumination. In *Eurographics (Short Papers)*, pp. 9–12, 2012.

[33] E. M. Reingold, L. C. Loschky, G. W. McConkie, and D. M. Stampe. Gaze-contingent multiresolutional displays: An integrative review. *Human factors*, 45(2):307–328, 2003.

[34] C. P. Remi Palandri, Samuel Gosselin. Optimizing oculus go for performance, 2018. https://developer.oculus.com/blog/optimizing-oculus-go-for-performance/.

[35] T. Ritschel, E. Eisemann, I. Ha, J. D. Kim, and H.-P. Seidel. Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. In *Computer Graphics Forum*, vol. 30, pp. 2258–2269. Wiley Online Library, 2011.

[36] M. Schwarz and H.-P. Seidel. Fast parallel surface and solid voxelization on gpus. *ACM transactions on graphics (TOG)*, 29(6):179, 2010.

[37] B. Segovia, J. C. Iehl, R. Mitanchey, and B. Péroche. Bidirectional instant radiosity. In *Rendering Techniques*, pp. 389–397, 2006.

[38] B. Segovia, J. C. Iehl, and B. Péroche. Metropolis instant radiosity. In *Computer Graphics Forum*, vol. 26, pp. 425–434. Wiley Online Library, 2007.

[39] F. X. Sillion, C. Puech, et al. *Radiosity and global illumination*, vol. 1. Springer, 1994.

[40] F. Simon, J. Hanika, and C. Dachsbacher. Rich-vpls for improving the versatility of many-light methods. In *Computer Graphics Forum*, vol. 34, pp. 575–584. Wiley Online Library, 2015.

[41] M. Stengel, S. Grogorick, M. Eisemann, and M. Magnor. Adaptive image-space sampling for gaze-contingent real-time rendering. In *Computer Graphics Forum*, vol. 35, pp. 129–139. Wiley Online Library, 2016.

[42] N. T. Swafford, J. A. Iglesias-Guitian, C. Koniaris, B. Moon, D. Cosker, and K. Mitchell. User, metric, and computational evaluation of foveated rendering methods. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 7–14. ACM, 2016.

[43] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics (TOG)*, 38(4):98, 2019.

[44] I. Wald, C. Benthin, and P. Slusallek. Interactive global illumination in complex and highly occluded environments. In *Rendering Techniques*, pp. 74–81, 2003.

[45] M. Weier, T. Roth, E. Kruijff, A. Hinkenjann, A. Pérard-Gayot, P. Slusallek, and Y. Li. Foveated Real-Time Ray Tracing for Head-Mounted Displays. *Computer Graphics Forum*, 2016. doi: 10.1111/cgf.13026

[46] H. Yee, S. Pattanaik, and D. P. Greenberg. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Transactions on Graphics (TOG)*, 20(1):39–65, 2001.