Foveated Photon Mapping

Xuehuai Shi, Lili Wang, Xiaoheng Wei, and Ling-Qi Yan

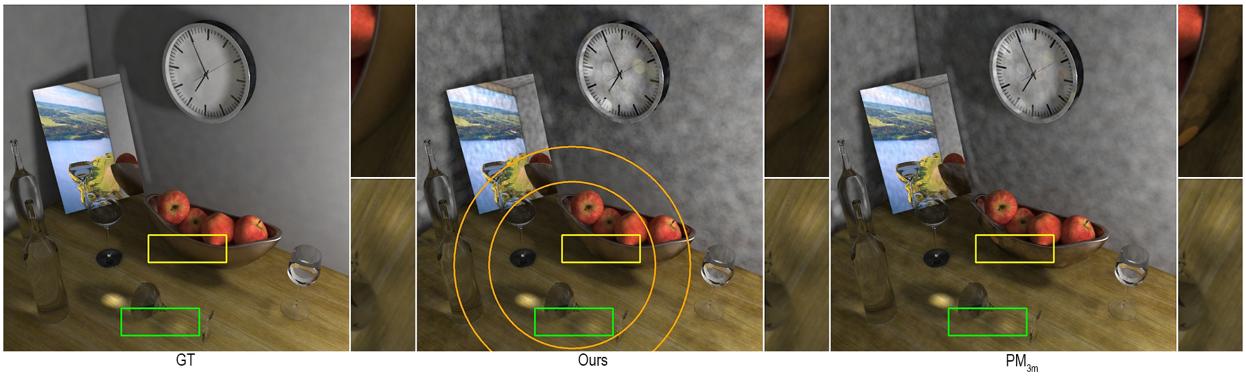

Fig. 1: Global illumination effects rendered by photon mapping method with 100 million photons (GT, column 1), our method with 3 million photons ($Ours$, column 2), and photon mapping with 3 million photons (PM_{3M}, column 3) in $Kitchen$. Compared with PM_{3M}, our method achieves 4× speedup, and smaller mean squared error in the foveal region: our method vs PM_{3M}, 0.62×10^{-2} vs 2.67×10^{-2}.

Abstract—Virtual reality (VR) applications require high-performance rendering algorithms to efficiently render 3D scenes on the VR head-mounted display, to provide users with an immersive and interactive virtual environment. Foveated rendering provides a solution to improve the performance of rendering algorithms by allocating computing resources to different regions based on the human visual acuity, and renders images of different qualities in different regions. Rasterization-based methods and ray tracing methods can be directly applied to foveated rendering, but rasterization-based methods are difficult to estimate global illumination (GI), and ray tracing methods are inefficient for rendering scenes that contain paths with low probability. Photon mapping is an efficient GI rendering method for scenes with different materials. However, since photon mapping cannot dynamically adjust the rendering quality of GI according to the human acuity, it cannot be directly applied to foveated rendering. In this paper, we propose a foveated photon mapping method to render realistic GI effects in the foveal region. We use the foveated photon tracing method to generate photons with high density in the foveal region, and these photons are used to render high-quality images in the foveal region. We further propose a temporal photon management to select and update the valid foveated photons of the previous frame for improving our method's performance. Our method can render diffuse, specular, glossy and transparent materials to achieve effects specifically related to GI, such as color bleeding, specular reflection, glossy reflection and caustics. Our method supports dynamic scenes and renders high-quality GI in the foveal region at interactive rates.

Index Terms—Virtual Reality, Foveated Rendering, Photon Mapping

1 Introduction

Nowadays, virtual reality (VR) enables us to enjoy new and unknown environments that cannot be explored in the real world. In order to provide users with an excellent experience, immersion and interaction are essential features of VR applications, which impose requirements on the performance of rendering algorithms. Moreover, virtual reality near-eye display devices, such as VR head-mounted displays, can have a single-eye refresh rate of 90hz. Therefore, further improvement of rendering algorithm efficiency becomes a key point for achieving low latency of display content.

Foveated rendering provides a solution to improve the performance of rendering algorithms. It allocates computing resources to different regions based on the human visual acuity, and renders images of different qualities in different regions. Rasterization-based methods and ray tracing methods can be integrated into the framework of foveated rendering in a quite straightforward way, due to the characteristics of multiple spatial resolution at the pixel level. However, rasterization-based methods are difficult to estimate global illumination (GI). Moreover, ray tracing methods are inefficient in rendering scenes that contain paths with low probability, such as scenes containing point light sources [45], caustic effects or specular-diffuse-specular (SDS) paths [56]. A massive number of traced paths are needed to achieve high-quality GI in these scenes.

Photon mapping method [18] is an efficient method to render GI for the scenes with different materials. Photon mapping is also a consistent method, and the accuracy of the renderer can be increased with the number of photons [52]. To further improve the rendering efficiency, we adapt photon mapping to the framework of foveated rendering, which brings about two problems. The first one is how to generate high-density photons in the foveal region to render high-quality images. The second one is how to improve the performance of the time-consuming photon tracing process with high-density photons.

In this paper, we propose the foveated photon mapping method to address the above two problems. To solve the first problem, we introduce the foveated photon tracing method to generate high-density photons in the foveal region based on the photons traced from the light sources

- *Lili Wang is with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China; Peng Cheng Laboratory, Shengzhen, China; and Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing, China. Lili Wang is the corresponding author. E-mail: wanglily@buaa.edu.cn.*
- *Xuehuai Shi and Xiaoheng Wei are with State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. E-mail: shixuehuaireal,2006318@buaa.edu.cn. Ling-Qi Yan is with University of California Santa Barbara, California, U.S. E-mail: lingqi@cs.ucsb.edu.*

Manuscript received 15 Mar. 2021; revised 11 June 2021; accepted 2 July 2021. Date of publication 27 Aug. 2021; date of current version 1 Oct. 2021. Digital Object Identifier no. 10.1109/TVCG.2021.3106488

with the traditional photon tracing method. We estimate the radiance of the 3D scene with a novel radiance estimation function using photons generated by the traditional photon tracing and the foveated photon tracing in rendering. To solve the second problem, the temporal photon management method is proposed to select and update the valid foveated photons of the previous frame. Furthermore, our method supports dynamic scenes with multiple materials such as diffuse, specular, glossy, and transparent ones. Our method renders high-quality images in the foveal region at interactive rates for GI effects, such as color bleeding, reflection, and caustics.

We compare the monocular images rendered by photon mapping with 100 million photons (Ground Truth, *GT*), our method with 3 million photons (*Ours*), and photon mapping with 3 million photons (PM_{3M}) shown in Figure 1. The binocular images rendered for head-mounted displays are shown in our supplementary video. Our method shows better GI effects in the foveal region than PM_{3M}. The region in the smaller orange circle is the foveal region, and the region between the smaller and larger orange circles is the transition region. We smoothly blend the peripheral radiance estimation results and foveal radiance estimation results in the transition region. The details in the rectangular region are magnified in the right of each rendering image. In the yellow rectangular region, the glossy effects on the fruit tray are too coarse in the result of PM_{3M}. In the green rectangular region, there are some noised points around the caustics in the result of PM_{3M}. Our method is efficient and achieves 42 FPS for each eye with an HTC Vive.

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

- The foveated photon mapping method, which can render high-quality global illumination in real-time on dynamic scenes with multi-material surfaces in the foveal region;
- The foveated photon tracing method to distribute the high-density photons in the foveal region for rendering high-quality global illumination;
- The temporal photon management pipeline, this is the first time that temporal reprojection is applied to photon reuse, which can effectively delete invalid photons and trace valid photons for the current frame.

2 Related Work

In this section, we first introduce the prior work of 3D foveated rendering in recent years, then discuss the existing visual importance based photon tracing, and discuss the strategies for maintaining temporal coherence in photon tracing methods, which are related to the temporal photon management of our method.

2.1 Foveated 3D Rendering

Guenter et al. [11] proposed the foveated 3D rendering to accelerate general, interactive 3D graphics rendering. Foveated rendering is based on the characteristic that the acuity of the human visual system (HVS) falls off with eccentricity. In other words, high-quality images need to be provided for the foveal region. Even if a low-quality image is provided for the peripheral region, it will not be perceived by HVS. The foveated 3D rendering can be grouped into several categories according to the acceleration strategies, such as level of details (LOD), multiple spatial resolution, multiple color resolution, and multiple illumination resolution.

LOD-based foveated 3D rendering generates different levels of geometric details of the scene in the preprocessing or online process. The foveal region is rendered with high-quality geometry, and the peripheral region is rendered with simplified geometry [24, 25, 32, 35].

The foveated rendering methods with multiple spatial resolution strategy shade the pixel with multiple rates in one output image adaptively. The rasterization and ray tracing based foveated rendering methods fall into this category. Guenter et al. [11] used foveated graphics to generate a 3-layer image with different resolutions using a rasterization pipeline and composited them to generate rendering results. He et al. [15] introduced a general rendering pipeline to shade the pixels with multiple rates in one output image adaptively. Patney et al. [36] presented a contrast enhancement method to recover the image details and a saccade-aware temporal antialiasing algorithm to address aliasing in the peripheral region. 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. Meng et al. [29] 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. Franke et al. [10] comprised recycling pixels in the periphery by spatiotemporal reprojecting them from previous frames and introduced a formalized perception-based metric to evaluate the artifacts and disocclusions. Meng et al. [28] presented the eye-dominance-guided foveated rendering method to render the scene at a lower foveation level (with greater details) for the dominant eye than the non-dominant eye for better performance.

Koskela et al. [22] proposed a theoretical estimation of performance gains available for the techniques related to optimizing ray tracing with foveated rendering, which showed that 94% of the paths could be omitted. They extended their work on 360-degree videos [23] and proposed a method to speed up the preview of progressively rendered images by applying foveated rendering to ray tracing. After this, Koskela et al. [21] demonstrated the foveated real-time path tracing system and proposed a novel Visual-Polar space in which both real-time path tracing and denoising are done before mapping to screen space. Weier et al. [50] adopted foveated rendering with reprojection rendering using previous frames that could drastically reduce the number of new samples that need to be traced with ray tracing. Molenaar et al. [30] proposed the non-uniformly sampling method for ray tracing based on the properties of HVS and reconstructing the image using these samples through an interpolation algorithm. Okan et al. [46] proposed a luminance contrast aware foveated method, which analyzed displayed luminance contrast and combined it with eccentricity to achieve spatial resolution reduction.

Multiple color resolution, and multiple illumination resolution were also used in foveated rendering. Duchowski et al. [6] proposed a level of color details framework, in which color degradation mapping for gaze-contingent display was constructed and used for evaluating the spatiochromatic peripheral sensitivity. Wang et al. [47] proposed a foveated instant radiosity for foveated rendering, which used more virtual point light sources (VPLs) to illuminate the foveal region, and fewer VPLs to illuminate the peripheral region. Yee et al. [53] introduced an aleph map representing the spatio-temporal error tolerance for dynamic scenes. The idea can be adapted to the foveated 3D rendering easily. 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 with different techniques.

Our method is a foveated rendering method based on the idea of multiple illumination resolution. We trace light photons from the light sources and foveated photons from the foveal region, gather light photons and foveated photons, and estimate the radiance through a novel radiance estimation function to render higher-quality GI effects in the foveal region.

2.2 Visual Importance based Photon Tracing

Jensen [17] presented the photon mapping method (PM) that can robustly render GI effects. PM is widely used for handling scenes with specular-diffuse-specular (SDS) paths. To improve PM performance, some researchers worked on using visual importance to measure the contribution of photons to the final image and tracing the photons according to the visual importance.

Peter et al. [38] introduced a global data structure named the visual importance map to control the emission and scattering in photon tracing. Keller et al. [20] extended this method and introduced a new importance sampling scheme for controlling the deposition of photons to render GI more accurately and faster. However, these methods are not suitable for interactive or real-time rendering because large scenes require more than 20,000 importons. Due to the need to construct a probability distribution function, choosing a new shooting direction is very time-consuming [40].

Metropolis-Hastings algorithm [8] was applied to light path mutation to take advantage of the coherence between important light paths connecting the light sources to the viewpoint. Chen et al. [2] integrated Metropolis-Hastings algorithm to mutate the current visible photon path for exploring more visible photon paths, this method reduced the errors in the image region with large artifacts considering the local coherence among light paths. Hachisuka et al. [13] used the adaptive Markov chain sampling to mutate the current visible light path and replica-exchange to reduce the local peak sampling stuck of the visual importance map. Zheng et al. [54] proposed a novel importance function that combined visual importance with Markov chain Monte Carlo methods to achieve the target distribution of photon paths sampling, then introduced a hybrid mutation strategy for generating photon paths. Collin et al. [3] applied a visual importance map to progressive photon volume rendering, the mutation strategy is the same to that proposed by Hachisuka [13]. The disadvantage of these methods is that light path mutation has a certain probability of generating photons that do not contribute to the visual important region. Compared with these methods, our method neither traces the tentative paths that are not used in subsequent calculations nor generates additional photons that do not contribute to the visual important region. All the foveated photons generated by our method contribute to the GI of the foveal region, so it is more efficient.

Besides photon tracing, there were some other methods that focused on photon density estimation to improve rendering quality. In recent research, Zhu et al. [56] predicted the kernel function of photon density estimation through a deep neural network, which can achieve higher quality GI with fewer photons compared with previous photon mapping methods. For simplicity, we gather photons in the radius disk near the hit point of each pixel for density estimation.

2.3 Temporal Coherence Strategies for PM

In order to improve the quality and accelerate rendering, some temporal coherence strategies were exploited for photon mapping methods. Myszkowski et al. [33] proposed the energy and perception based error metrics to guide lighting computation that can keep noise inherent below the sensitivity level of the human observer with the known dynamic frames. Dmitriev et al. [5] identified the scene regions that require illumination update by a small number of the pilot photons and exploited temporal coherence of illumination by tracing photons selectively to these scene regions. Cammarano et al. [1] derived a time-related radiance estimate, which could distribute photons in space and time, so that the photon mapping method could be used to render the effect of motion blur. Tawara et al. [44] stored the previously calculated incoming radiance samples for reducing the number of tracing rays, and refreshed incoming radiance samples evenly in space and time using aging criteria. We also adopt the idea of reducing the number of rays being traced for speeding up the rendering process. However, Tawara et al. [44] needed to perform a nearest neighbor search for removing redundant caches and may introduce bias into the rendering results since the copied samples may not be valid or reweighted properly across frames. While our method effectively deleting invalid photons through the set of foveated photon lists in each pixel and retraces all valid photons according to changes in the viewpoint and fovea, and the movement of dynamic objects. Weber et al. [49] proposed bilateral filtering to reconstruct lighting based on the local density of the hit points, and reduced flickering between subsequent animation frames. Weiss et al. [51] extended the stochastic progressive photon mapping method to render animated sequences with dynamic scenes. It identified photons that can be reused in the current frame from multiple frames in an animation sequence. Reusing these photons can reduce the number of rays traced in the photon tracing step. The hit points of all frames are grouped by their 3D positions to speed up the time-consuming kd-Tree traversal in the photon density estimation step. But this method is an off-line method which requires pre-defined animations of objects and materials. Elek et al. [7] used photon mapping for interactive cloud rendering. For each frame, a low-resolution grid was up-sampled to the cloud density field resolution to maintain temporal coherence.

We propose the temporal photon management (TPM) with the consideration of temporal coherence. The standard temporal reprojection [34] reuses the pixels' color of the previous frame based on visibility. Compared with this temporal coherence strategy, TPM reuses the foveated photon list and the first bounce surface information for each pixel based on visibility and material, and efficiently deletes invalid photons and traces the photons that are valid for the current frame. TPM is a new pipeline for photon reuse based on the temporal reprojection.

3 Foveated Photon Mapping

Foveated photon mapping is based on the idea of multiple illumination resolution in foveated rendering, and it computes high-quality GI in the foveal region by achieving higher photon density in the foveal region than in the peripheral region. For tracing more photons in the foveal region, after the step of the regular photon tracing that traces photons from the light source (light photons), we introduce a foveated photon tracing method to trace photons from the foveal region (foveated photons). For handling dynamic scenes, we use Tawara's idea [43] to divide photons into two categories: photons with positive flux for static scenes and photons with positive or negative flux for dynamic objects. Therefore, we store these photons in 4 photon maps. M_{ls} stores the light photons of the static scene, M_{ld} stores the light photons of the dynamic objects, M_{fs} stores the foveated photons of the static scene, and M_{fd} stores the foveated photons of the dynamic objects. Each photon has four attributes: dir represents which direction the photon is traced from; $flux$ is the energy of the photon; pos represents the location of the photon; $isAcc$ is a two-valued flag, and will be used in the energy normalization in Section 3.1. Foveated photon has two extra attributes: $nextf$ is the ID of the next foveated photon along the tracing path; $nearl$ is the ID of the nearest light photon if the flux of this light photon is positive, otherwise, the value is set as '-1'. For better performance and maintaining the temporally coherent GI effects of adjacent frames in the foveal region, the previous frame's foveated photons are updated and reused for the current frame. For rendering the current frame, the radiance in the foveal region is estimated using both the light photons and the foveated photons with a novel radiance estimation function, and the radiance in the peripheral region is estimated utilizing only with the light photons.

Given a 3D scene with static parts S and dynamic objects D, the light source L, a viewpoint of current frame V, a viewpoint of last frame V', the foveal region r_f, the focus region r_t (the foveal and the transition region), the output resolution (w,h), the maximum foveated photon path per pixel N, the framebuffer is rendered by using Algorithm 1.

Algorithm 1: Foveated Photon Mapping

input : static scene S, dynamic objects D, light source L, viewpoint of current frame V, viewpoint of last frame V', foveal region r_f, focus region r_t, output resolution (w,h), maximum photon path per pixel N

output : $Framebuf$

$M_{ls} \leftarrow PT_s(S, L)$
$M_{ld,fs,fd} \leftarrow \varnothing$
$fbm \leftarrow initMask(w,h,r_t,N)$
$fbp \leftarrow initPlist(w,h,N)$
for *each frame* **do**
  $M_{ld} \leftarrow PT_d(S, D, L)$
  if *not first frame* **then**
    $M_{fs}, fbm, fbp \leftarrow TemporalPhotonManage(S, D, V, V', r_t, M_{fs}, fbm, fbp)$
  $[M_{fs,fd}, fbp] \leftarrow [M_{fs,fd}, fbp] \cup PT_f(S, D, V, M_{ls,ld}, fbm)$
  $\eta \leftarrow energyWeight(M_{ls,ld,fs,fd})$
  $Framebuf \leftarrow RenderScene(S, D, V, r_f, r_t, M_{ls,ld,fs,fd}, \eta)$

The motivation of Algorithm 1 is to compute high-quality GI by obtaining higher photon density in the foveal region than in the peripheral region. Thus we use Algorithm 1 to trace more photons to the foveal region for getting higher photon density in the foveal region. In the initialization of Algorithm 1, we first trace the photons of the static scene from the light source and generate the photon map M_{ls} (line 1). Since the scene will not be changed, M_{ls} is computed only once and is valid for all frames. $M_{ld,fs,fd}$ is initialized as empty (line 2). The per-pixel photon path counter fbm is a 2D array corresponding to the framebuffer and records the total number of foveated photon paths that need to be traced for each pixel in the framebuffer. In *initMask*, we set 'N' in the focus region (the foveal and the transition region) in fbm (line 3). The per-pixel static foveated photon path recorder fbp is a 2D array list corresponding to the framebuffer, which is used to store the IDs of the first static foveated photons for each static foveated photon path starting from this pixel (line 4). The static foveated photon path means the nearest light photons of the foveated photons in this path are all light photons in M_{ls}.

In each frame, we firstly trace light photons for the dynamic objects in the scene (line 6). Then, if the current frame is not the first frame, we need to update the static foveated photon map M_{fs}, the per-pixel photon path counter fbm, the per-pixel static foveated photon path recorder fbp of the last frame (line 7-8). After this, we perform the foveated photon tracing to generate new foveated photons in $M_{fs,fd}$, and update fbp (line 9). We compute the energy normalized parameter η in line 10 for radiance estimation in the focus region because the foveated photons generated in our method are not emitted from the light source. Finally, the scene is rendered with the photons from four photon maps $M_{ls,ld,fs,fd}$(line 11).

For the static scene, the light photons are traced with the traditional photon mapping method [18] without any dynamic objects (PT_s, line 1). The photons are emitted into the scene from the light source, then bounce several times on the scene's surfaces, and stay on the diffused surfaces. For the dynamic objects, we use the irradiance cache based method [43] to trace the light photons for the dynamic objects (PT_d, line 6). First, we render the scene containing dynamic objects with a cubemap from the light source. If the pixel in the cubemap belongs to the dynamic object, the direction from the light source to the 3D position of this pixel will be recorded. We sample these directions randomly and trace the photons along with the sampled directions. The photons bounce several times in the scene. When a photon hits the dynamic object's surface, it spawns two rays at the intersect point to continue photon tracing. The first ray is regarded as a positive ray, which is reflected or transmitted with the positive energy, or absorbed in the scene according to the surface's material. The positive ray leaves the photon with positive energy at all intersections between the rays and the scene's surface. The second ray is regarded as a negative ray, which pierces dynamic objects and carries negative energy. The negative rays only leave photons with negative energy at the intersections between the rays and the static scene's surface because we need to subtract energy from the static irradiance buffer due to the occlusion of dynamic objects. Finally, both the photons with positive energy and the photons with negative energy are stored in M_{ld}.

The rest of Section 3 is organized as follows: Section 3.1 introduces the foveated photon tracing; Section 3.2 explains the details of the temporal photon management between adjacent frames; Section 3.3 gives a description of rendering.

3.1 Foveated Photon Tracing

We only use 60% of the total photons for light photon tracing, so we get a sparse distribution of photons over the scene. However, the photon density is not high enough to obtain high-quality GI in the foveal region. In order to increase the photons' density to illuminate the foveal region, we introduce a foveated photon tracing method based on the photons' spatial coherence. In foveated photon tracing, the foveated photon is traced from the viewpoint so that it passes through the focus region and hits the scene's surface. The number of photon paths that need to be traced is according to fbm. The bounce direction and the foveated photons' flux are determined according to the material of the hit points in the scene and their nearest light photons.

Figure 2 describes the procedure of foveated photon tracing. The scene contains a transparent sphere, a specular cube, and diffuse walls. The cube is dynamic. The same as the regular photon tracing, foveated photons are only placed on the diffuse surface, and Russian roulette is utilized to determine the termination of tracing [18]. Red lines show the tracing paths from the light source to generate the light photons (red dots) on the diffuse wall with irradiance cache based method [43]. When the ray hits the top of the dynamic cube, it spawns two rays: the solid line and the dotted line. The solid line visualizes the light tracing path to generate light photons with positive energy p_l^4, and the dotted line visualizes the light tracing path to generate light photons with negative energy $p_l'^1$. Green lines show the tracing paths from the viewpoint V, passing through the foveated pixel px on the image plane to generate the foveated photons (green dots) by our foveated photon tracing method. When the foveated photons hit the surface of the specular cube or the transparent sphere, they bounce along the directions according to the reflection or refraction laws. While if the foveated photon hits the diffuse wall, the nearest light photon is found. If this light photon carries positive energy, the bounce direction of the foveated photon is set to the reverse direction of the incident direction of this nearest light photon, otherwise, the foveated photon is discarded. In Figure 2, the bounce direction of the foveated photon p_f^1 is reversed to the incident direction of its nearest light photon p_l^3, and the bounce direction of p_f^2 is reversed to the incident direction of p_l^4.

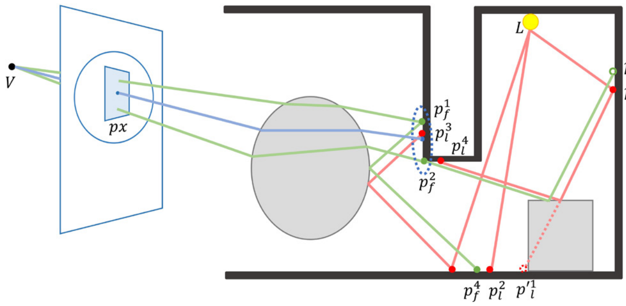

Fig. 2: The procedure of foveated photon tracing.

After determining the foveated photon's bounce direction, the nearest light photon of the foveated photon at the end of each tracing path is found, and the flux value of this light photon is used for estimating the flux of all foveated photons along the path. In Figure 2, the flux of p_f^4 is set the same as the flux of p_l^2, and the flux of p_f^3 is set the same as the flux of p_l^1. The flux of p_f^4 is transmitted to p_f^1, and the flux of p_f^3 is transmitted to p_f^2 as the traditional photon flux transmission. For each foveated photon, the next foveated photon along the tracing path is recorded in $nextf$, and the ID of the nearest light photon with positive energy is recorded in $nearl$. The maximum depth value of bounce for each foveated photon path will not exceed the predefined maximum depth, otherwise the foveated photons in this foveated photon path are removed. In the case of Figure 2, the maximum bounce depth is 4, the bounce depth of p_f^3 is 5, so p_f^3 is removed.

In Figure 2, the blue line visualizes the tracing ray passing through the center of the pixel px and the blue dotted eclipse shows the gathering region for px. Beside the light photon p_l^3, two foveated photons p_f^1 and p_f^2 are generated to illuminate px in the rendering process.

After the foveated photon tracing, we iterate four photons maps $M_{ls,ld,fs,fd}$ and compute the energy-normalized parameter η with Equation 1.

$$\eta = \frac{sumFlux(M_{ls,ld}, r_t)}{sumFlux(M_{ls,ld,fs,fd}, r_t)} \quad (1)$$

where $sumFlux(M_{ls,ld}, r_t)$ is the sum energy of the light photons that carry the positive flux in the focus region, and $sumFlux(M_{ls,ld,fs,fd}, r_t)$ is the sum energy of the foveated photons and light photons that carry the positive flux in the focus region.

3.2 Temporal Photon Management

Since tracing all foveated photons for the current frame is very expensive, reusing the foveated photons in the previous frame can improve performance and maintain temporal coherence. We perform temporal photon management to select and update photons in the previous frame, and to delete invalid photons based on changes in the viewpoint, changes in the foveal region, and dynamic objects' movement. In Algorithm 2, per-pixel photon path counter fbm, per-pixel static foveated photon path recorder fbp and foveated photon map for static scene M_{fs} are updated according to 3D scene S, dynamic objects D, the viewpoint of the current frame and the previous frame V, V', the current focus region r_t.

Algorithm 2: Temporal Photon Management

input : static scene S, dynamic objects D, viewpoint of current frame V, viewpoint of last frame V', focus region r_t, maximum photon path per pixel N, output resolution (w,h), static foveated photon map M_{fs}, per-pixel photon path counter fbm, per-pixel static foveated photon path recorder fbp

output : M_{fs}, fb_m, fb_p

```
1  fbm' ← fbm;
2  fbp' ← fbp;
3  fbm ← initMask(w, h, r_t, N);
4  fbp ← initPlist(w, h, N);
5  for px ∈ r_t do
6      v ← BackProject(S, D, V, px);
7      px' ← Project(S, D, V', v);
8      if v on S and v is diffuse then
9          fbp[px] ← fbp'[px'];
10         fbm[px] ← fbm'[px'];
11     else
12         M_fs ← deletePhoton(fbp'[px'], M_fs);
13 return M_fs, fbm, fbp;
```

The motivation of Algorithm 2 is to improve performance and maintain temporal coherence. Thus we use Algorithm 2 to reuse the valid foveated photons from the previous frame that can contribute to the current frame, and trace new foveated photons that are valid for the current frame. Firstly, we keep fbm, fbp of the previous frame (line 1-2), and initialize them for the current frame (line 3-4). For each pixel px in the focus region, we back-project it into 3D space to get a 3D point v from the current viewpoint V (line 6) and then project it from the previous viewpoint V' (line 7) to get its previous projection px'. If v is on the surface of the static scene and the surface is diffuse, we will set the value in fbp and fbm of the current frame's corresponding pixels according to the values of fbp and fbm in the previous frame (line 8-10). Otherwise, we delete the photons along the static photon path from M_{fs} and the paths in $fbp'[px']$ (line 11-12).

3.3 Rendering

In order to achieve high-quality GI in the foveal region, the photons in four photon maps $M_{ls,ld,fs,fd}$ are used to gather the radiance in the focus region, while only two photon maps $M_{ls,ld}$ are used to render the peripheral region. For a given 3D point x in different regions, its radiance $I(x)$ can be estimated by using Equation 2.

$$I(x) = \begin{cases} g(x, M_{ls,ld,fs,fd}, \eta) & x \in r_f \\ (1-b(x)) * g(x, M_{ls,ld,fs,fd}, \eta) \\ +b(x) * g(x, M_{ls,ld}) & x \in r_t - r_f \\ g(x, M_{ls,ld}) & x \in r_p \end{cases} \tag{2}$$

If x is in the foveal region, the radiance will be estimated by the weighted radiance estimation function $g(x, M_{ls,ld,fs,fd}, \eta)$:

$$\begin{aligned} g(x, M_{ls,ld,fs,fd}, \eta) \approx \frac{\eta}{\pi\gamma^2}(k * \sum_{p \in M_d} f_r(x, \omega_o, \omega_p)\varphi_p(x, \omega_p) \\ +(1-k) * \sum_{p \in M_s} f_r(x, \omega_o, \omega_p)\varphi_p(x, \omega_p)) \end{aligned} \tag{3}$$

where η is the energy-normalized parameter, γ is the radius for gathering the photons, M_d refers to the two dynamic photon maps $M_{ld,fd}$, M_s refers to the two static photon maps $M_{ls,fs}$, f_r is the BRDF function, φ_p is the flux of the photon p in the photon maps, ω_o is the direction from x to V, ω_p is the incident direction of the photon p. We use the irradiance cache based method [43] to blend the radiance contributed by the photons in the static photon maps with the radiance contributed by the photons in the dynamic photon maps with weight coefficient k. k is estimated with Equation 4.

$$\begin{cases} k = -2W^3 + 3W^2 \\ W = min(\frac{\sum_{p \in M_d}^{\varphi > 0} u(\varphi_p) + \frac{1}{\eta}\sum_{p \in M_d}^{\varphi < 0} u(\varphi_p)}{\sum_{p \in M_s} u(\varphi)}, 1) \\ u(\varphi_p) = max(|\varphi[p.r]|, |\varphi[p.g]|, |\varphi[p.b]|) \end{cases} \tag{4}$$

where $p.r$, $p.g$, and $p.b$ denote the red, green and blue components respectively in the flux of φ.

If x is in the peripheral region, the radiance will be estimated only with light photons in $M_{ls,ld}$ by using Equation 5.

$$\begin{aligned} g(x, M_{ls,ld}) \approx \frac{1}{\pi\gamma^2}(k * \sum_{p \in M_{ld}} f_r(x, \omega_o, \omega_p)\varphi_p(x, \omega_p) \\ +(1-k) * \sum_{p \in M_{ls}} f_r(x, \omega_o, \omega_p)\varphi_p(x, \omega_p)) \end{aligned} \tag{5}$$

The radiance of x in the transition region can be estimated with the linear interpolation of the radiance in the foveal region and the radiance in the peripheral region using the weight coefficient b. b is computed with Equation 6:

$$b(x) = \frac{x - radius(r_f)}{radius(r_t) - radius(r_f)} \tag{6}$$

where *radius* is the function to compute the radius of the given region in pixel.

In VR, the same valid photons 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 [34]. The right-eye view only has to shade new pixels in the case that no valid sample was found, rather than recalculating everything [31].

4 Results and Discussion

We test our method in five scenes: *Room* (215.4k tris., Figure 3, row 4), *Sponza* (314.1k tris., row 1), *Study* (118.3k tris., row 2), *Cornellbox* (140.9k tris., row 3) and *Kitchen* (73.6k tris., Figure 1). We use an HTC Vive Head-mounted display (HMD) with a Droolon aGlass to track the head motion and the foveated point of the user. The HMD is connected to a PC workstation with a 3.8 GHz Intel(R) Core(TM) i7-9800X CPU, 64 GB of memory, and an NVIDIA GeForce GTX 2080 Ti graphics card. The resolution of the output images for each eye of the HMD is 1080×1200.

4.1 Implementation

In our implementation, the pixel radius of the foveal region is estimated with Equation 7 in [12].

$$radius(r_f) = \tan(e_f) \times b \times PPI \tag{7}$$

where the angular radius e_f of the foveal region is set to 9.78° in [12]; *PPI* is the pixel density of HMD that is measured by pixel per inch

Fig. 3: Comparison between *GT* (column 1), *Ours* (column 2) and PM_{3M} (column 3) in different scenes. The details in the rectangular regions are magnified and placed on the right side of each image.

(*ppi*), HTC Vive is 615*ppi*; *b* is the pupil-to-lens distance of HMD, HTC Vive is 18*mm*.

To accelerate traditional photon tracing from the light source, the first bounce of the photon uses reflective shadow maps (RSMs) instead of the path tracing. In the photon density estimation step of our method, we use gathering to estimate photon density, and apply three sorted grids to store static light photons, dynamic light photons, and foveated photons. This is due to two reasons. The first one is that gathering can be scaled well to higher-resolution and fine detail. Gathering [48, 55] is a commonly used method for photon density estimation. It traverses all pixels, and uses spatial data structures to search the nearby photons for the corresponding hit point of each pixel. As gathering can efficiently sample stochastically during shading, it can be scaled well to higher-resolution and fine detail [26]. Another method for photon density estimation is scattering [16, 27]. Scattering traverses all photons. It contributes energy to the pixels it covers for each photon, which is not efficient for scaling to higher-resolution and fine detail. The second reason is that the performance of using sorted grids to store photons and gathering nearby photons for each hit point is better than using K-D trees and stochastic hash according to the experiment results showed in Pedersen et al. [37].

Table 1: *MSE* and MSE' ($\times 10^{-2}$) in the foveal (*Fove*) and peripheral regions (*Peri*) with different methods.

Scene	*MSE* in *Fove*		*MSE* in *Peri*		MSE' in *Fove*		MSE' in *Peri*	
	Ours	PM_{3M}	*Ours*	PM_{3M}	*Ours*	PM_{3M}	*Ours*	PM_{3M}
Room	0.85	2.51	2.67	1.26	1.67	2.39	0.97	0.80
Sponza	1.33	5.96	4.53	2.54	0.33	0.76	0.18	0.34
Study	0.69	3.00	3.95	2.52	11.45	12.75	4.64	5.46
Cornellbox	0.87	3.22	4.79	2.38	2.13	3.00	5.39	4.61
Kitchen	0.62	2.67	9.25	5.58	0.80	1.44	1.22	2.03

The depth value of photon tracing is eight, the same as [19]. The radius of photon gathering γ is calculated via the Heuristic Equation 8 [14].

$$\gamma = \frac{(\sum_{i=1}^{3}(bbox_{max} - bbox_{min})[i])/3.0}{(w+h)/2.0} * 20.0 \tag{8}$$

where $bbox_{min}$ and $bbox_{max}$ are the minimum and maximum corner points of the scene's bounding box. A edge-avoiding À-Trous wavelet transform based spatial filter [4] is used to postprocess the rendering results of all three methods.

4.2 Quality

The quality of our results is compared with those of ground truth and the photon mapping method with the same number of photons (Figure 3). The ground truth images are obtained using the photon mapping method with 100 million photons (*GT*, column 1). Both our method (*Ours*, column 2) and the compared photon mapping method (PM_{3M}, column 3) use 3 million photons. In our method, 60% photons are light photons emitted from the light source, and 40% photons are foveated photons generated by our foveated photon tracing. Specifically, we use 1.8*M* light photons that include static light photons and dynamic light photons, and 1.2*M* foveated photons. The orange circles on the image of our method indicate the foveal region. We also crop and magnify the details inside the foveal region on the right of each rendering image for comparison (up: details in the yellow rectangle, down: details in the green rectangle).

Our results are more closer to the results of the ground truth than those of PM_{3M}. Some artifacts are shown in the cropped regions rendered by PM_{3M}: row 1, the color bleeding on the floor should be smoother and greener due to the green curtain, and the color bleeding on the dragon is noised with the wrong bleeding colors; row 2, there are many noised points on the reflection of the books in the mirror, and the reflected image on the teapot is noised; row 3, the caustics on the sphere and the wall are noised with the dark points; row 4, the reflection of the teapot on the specular flagon has some incorrect bright spots and the shading of the teapot is not smooth.

We quantify the quality with two metrics: the mean squared error *MSE* for all pixels of the results compared with 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 9. We measure *MSE* and MSE' in the foveal regions and peripheral regions separately.

$$MSE' = \frac{\sum_{i=2}^{n} MSE_{i,i-1}}{n-1} \tag{9}$$

Table 1 shows the comparison of *MSE* and MSE' of our method and PM_{3M} for the images in Figure 1 and Figure 3. According to the table, *MSE* of our method is smaller than that of PM_{3M} in the foveal regions in all scenes, thanks to our foveated photon tracing scheme. With our method, *MSE* in the foveal regions is consistently smaller than that in the peripheral regions, while the results of PM_{3M} are not consistent. Because our method uses a foveated photon tracing strategy to increase the photon density in the foveal region. For the average temporal error, our method's MSE' is always smaller than that of PM_{3M} in the foveal region in all scenes. That is because PM_{3M} retraces photons from scratch for each frame without considering the temporal coherence, while our method maintains a high photon density in the foveal region and reuses the valid foveated photons from the previous frame. MSE' of our method in the peripheral regions is not always larger than that of PM_{3M} (smaller in *Sponza*, *Study* and *Kitchen*, larger in *Room* and *Cornellbox*). This is because our method reuses photons of the previous frame to ensure temporal coherence. Even in the peripheral region where the photon density is smaller than that of PM_{3M}, there are still some cases with lower MSE'.

The images of Figure 4 visualize *MSE* and MSE' for the images in Figure 1 and Figure 3, and MSE' is calculated by moving the viewpoint slightly. The whiter pixels represent larger error. The error visualization shows that *MSE* and MSE' of our method are always smaller than those of PM_{3M} in the foveal regions. In the peripheral regions, *MSE* of our method is larger than that of PM_{3M} due to the sparse density of photons,

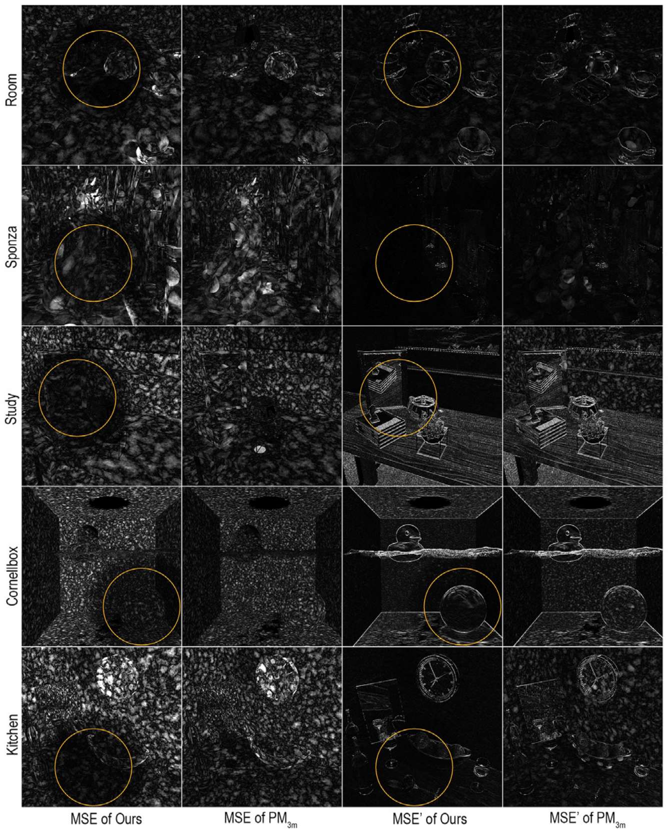

Fig. 4: Visualization of *MSE* and *MSE′* of our method and PM_{3M} in different scenes.

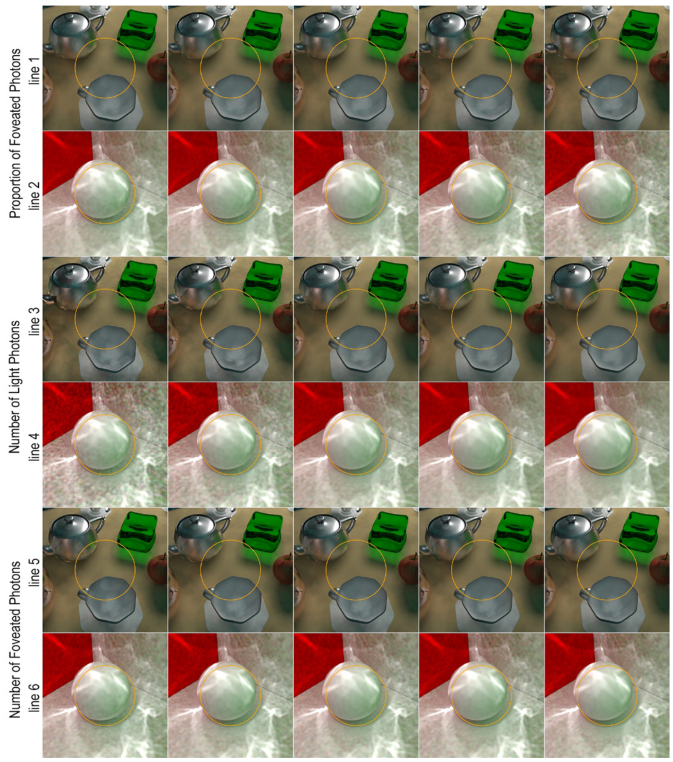

Fig. 5: Rendering results of our method with different proportions of foveated photons in total photons $\frac{N_f}{N_f+N_l}$, different number of light photons N_l, and different number of foveated photons N_f in *Room* and *Cornellbox*.

and MSE' of our method is very close to that of PM_{3M} although the photon density of our method is smaller than that of PM_{3M}.

Several parameters of our method affect the image quality. Figure 5 shows the rendering results of our method in *Room* and *Cornellbox* with various numbers of the foveated photons and light photons. Images in lines 1-2 show the rendering result when the total photon number is fixed, and the proportion of foveated photons in the total photons is gradually increased from left to right: 20%, 30%, 40%, 50%, 60%. In *Room*, there is no obvious visual difference among the images in the foveal regions with different proportions of foveated photons, while the noise in the images of the peripheral regions increases due to the decreased number of light photons. In *Conellbox*, when the proportion of foveated photons increases from 20% to 40%, the noise in the foveal region decrease, but from 40% to 60%, the noise in the foveal region increase again. This is because when the number of light photons reaches a certain number, increasing the density of foveated photons in the foveal region can reduce noise. However, since the flux of foveated photons is estimated based on the light photons, if the light photons are too sparse, the noise cannot be reduced even if the density of foveated photons in the foveal region is increased. The fewer light photons, the more the noise. *MSE* in the foveal and peripheral region of these two scenes are shown in Figure 6 (a).

Lines 3-4 in Figure 5 visualize the rendering results of our method with an increased number of light photons: 0.45m, 0.9m, 1.8m, 3.6m, 7.2m. The number of the foveated photon is fixed to 1.2m. In both *Room* and *Cornellbox*, the noise in both regions decreases when the number of the light photons increase. When the number of light photons increases from 0.45m to 1.8m, the quality improvement is more obvious, and the number of light photons further increases, and the quality improvement effect is not significant. *MSE* in the foveal and peripheral region of these two scenes are shown in Figure 6 (b).

Lines 5-6 in Figure 5 visualize the rendering results of our method with an increased number of foveated photons: 0.3M, 0.6M, 1.2M, 2.4M, 4.8M. The number of light photons is fixed to 1.8m. In both *Room* and *Cornellbox*, the noise in foveal regions decreases when the number of the foveated photon increase. When the number of foveated photons increases from 0.3m to 1.2m, the quality improvement is more obvious; but when the number of foveated photons continues increasing, the quality improvement effect becomes not significant. The images in the peripheral region remain the same quality since the foveated photons only contribute to the foveal region. *MSE* in the foveal and peripheral region of these two scenes are shown in Figure 6 (c).

We also compare the rendering results of our method and image space photon mapping (ISPM) [27] at the same time cost in the Figure 7. Some artifacts are produced in the image rendered by ISPM compared with our method in the foveal region. This is because the photon density of ISPM in the foveal region is not high enough, and the scattering radius of the photon is not large enough to cover more pixels.

Figure 8 shows images of our method without (left) and with (right) the edge-avoiding À-Trous spatial filter [4] in *Study* and *Kitchen*. The spark and dark artifacts are reduced in the filtered images compared with the rendered results.

4.3 Performance

Table 2: Performance (ms) of our method compared with photon mapping using different numbers of photons (million).

Scene	*Sponza*	*Study*	*Cornellbox*	*Room*	*Kitchen*
$Ours_{wo}$	85.9	56.2	55.5	66.3	51.3
Ours	33.1	25.1	28.5	29.6	23.6
Avg.Reu.	83%	78%	71%	76%	78%
vs. PM_{3M}	4.5×	4.3×	3.5×	4.1×	4.1×
vs. PM^t_{3M}	0.8×	0.9×	1.4×	0.8×	0.9×
vs. PM_{eq}	9.1×	10.1×	9.0×	10.0×	12.0×
#Photon	8.1	9.6	9.9	9.5	11.5
vs. PM^t_{eq}	3.4×	5.4×	6.1×	3.3×	4.3×
#Photon	10.1	12.8	14.1	12.5	15.3

Table 2 shows the frame rendering time of our method and the

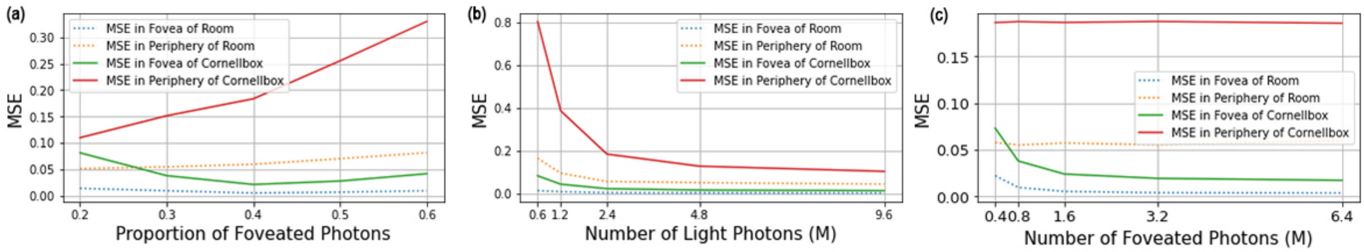

Fig. 6: *MSE* in *Room* and *Cornellbox* as a function of (a) the proportion of foveated photons in total photons $\frac{N_f}{N_l+N_f}$, (b) the number of light photons N_l, and (c) the number of foveated photons N_f.

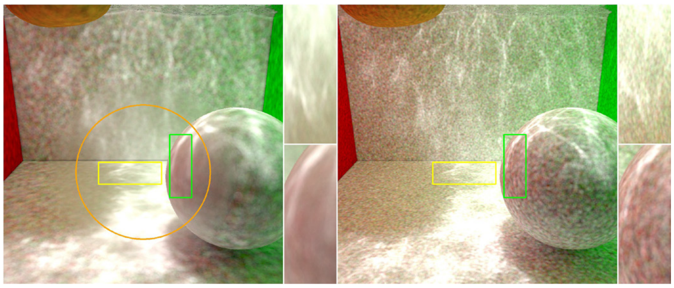

Fig. 7: The images rendered by our method (left) and image space photon mapping [27] (right) in the same rendering time.

Fig. 8: The images rendered by using our method without (left) and with spatial filtering (right).

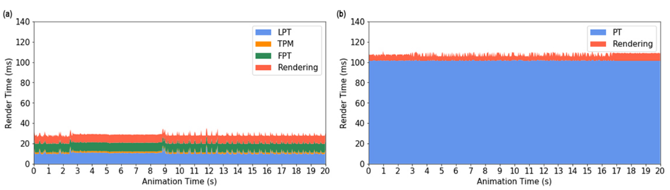

Fig. 9: Time cost in each step of our method (left) compared with that of PM_{3M} (right) in *CornellBox*.

speedup versus the photon mapping method. Lines 2 and 3 show the time cost of our method without ($Ours_{wo}$) and with the temporal photon management (*Ours*). Our method with the temporal photon management gets 1.9-2.6 × speedup due to high reuse rate of foveated photon (line 4). Compared with PM_{3M} (line 5), our method achieves 3.5-4.5 × speedup. Compared with PM^t_{3M} (line 6), the photon mapping with 3*M* photons using the temporal strategy [43], our method achieves similar performance as PM^t_{3M}. But the quality of our method in the foveal region is far better than that of PM^t_{3M}, *MSE* of our method is 5.7-8.5× smaller than that of PM^t_{3M}. Compared with the photon mapping with the comparable *MSE* in the foveal region PM_{eq} as *Ours* (line 7), our method achieves 9.0-12.0× speedup. The photon numbers for PM_{eq} are shown in line 8. Compared with the photon mapping using the temporal strategy [43] with the comparable *MSE* in the foveal region PM^t_{eq} as *Ours* (line 9), our method achieves 3.3-6.1× speedup. The photon numbers for PM^t_{eq} are shown in line 10.

Figure 9 shows the time cost on each step using our method (left) and PM_{3M} (right) for rendering each frame of *Cornellbox*. Our method has four steps: 1) light photon tracing (LPT); 2) foveated photon tracing (FPT); 3) temporal photon management (TPM); 4) rendering, while PM_{3M} has two steps: 1) photon tracing (PT); 2) rendering. Our FPT and TPM only cost about 10ms. With FPT and TPM, the LPT time can be reduced to 10% of the LPT of PM_{3M}. The rendering time of the two methods are almost the same. Therefore our method is 4× faster of PM_{3M}.

Table 3: Performance (ms) of our method with different parameters in *Room* and *Kitchen*.

$\frac{N_f}{N_l+N_f}$		20%	30%	40%	50%	60%
CostTime	*Room*	23.7	27.6	29.6	32.0	36.2
	Cornellbox	21.4	24.3	26.5	28.4	31.8
$N_l(million)$		0.45	0.9	1.8	3.6	7.2
CostTime	*Room*	23.2	25.1	29.6	51.7	93.1
	Cornellbox	17.6	20.5	26.5	40.1	72.2
$N_f(million)$		0.3	0.6	1.2	2.4	4.8
CostTime	*Room*	20.1	24.6	29.6	40.9	67.5
	Cornellbox	19.7	22.0	26.5	34.5	57.4

Table 3 shows the performance in *Room* and *Cornellbox* with various numbers of different photons. The time cost increases when the proportion of the foveated photons increase (lines 1-3). This is because along with the increase in the proportion of foveated photons, the number of foveated photons also increases. To determine the flux and bounce direction of foveated photons, we need to find the nearest light photon for each foveated photon. This search process is time-consuming. Therefore, even if the number of light photons is reduced, the entire calculation process's performance still drops. Lines 4-9 show the time cost when the number of light photons or foveated photons increases, the time cost is proportional to the total number of photons.

5 User Study

We design a within-subject study [9] to evaluate the perceptual GI quality between our method and PM.

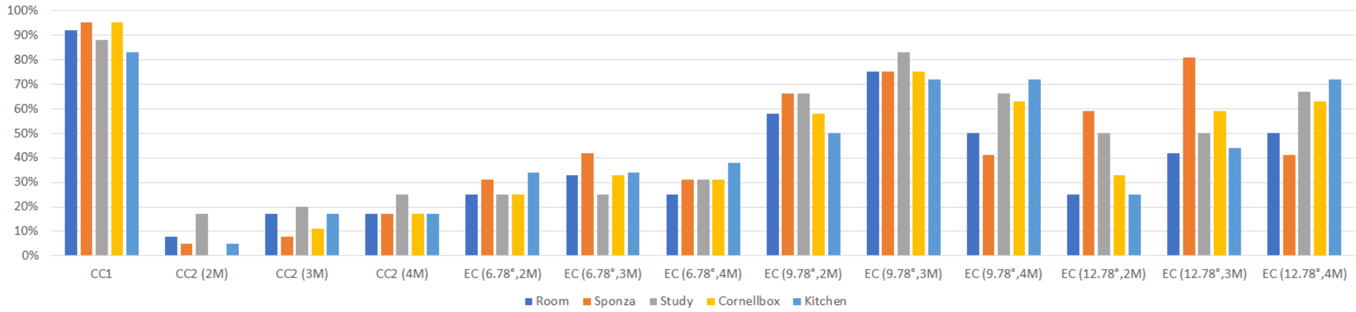

Fig. 10: Acceptable rating for the rendering quality of each item in $CC1$, $CC2$ and EC. The percentage of each item represents the acceptance ratio among all participants.

Conditions. We use our method as an experimental condition (EC), which consists of a full factorial combination of five scenes $\{Room$, $Sponza$, $Study$, $Cornellbox$, $Kitchen\}$, three foveal angular radius e_f $\{6.78°, 9.78°, 12.78°\}$, and three total photon numbers $\#Photon$ $\{2M$, $3M$, $4M\}$. Each item in EC is a short animated sequence represented as $EC(e_f,\#Photon)$. The first control condition ($CC1$) uses PM to render short animated sequences of five scenes, and maintains MSE equal to the fovea in $EC(6.78°,4M)$. The second control condition ($CC2$) uses the same photon number as EC $\{2M, 3M, 4M\}$. Each item in $CC2$ is a short animated sequence represented as $CC2(\#Photon)$.

Participants. We recruit 32 participants in this user study, 20 males and 12 females, between 20 and 28 years old. Each participant complete 85 trials in randomized order, the items in EC were presented once, and the items in $CC1$ and $CC2$ were presented twice.

Task. In the task, each participant is presented with the short animated sequences of EC, $CC1$ and $CC2$ in randomized order. Each short animated sequence is eight seconds long and separated by a short interval (0.5s) of black, which is the same as Guenter et al. [11]. In the process of the task, participants are asked to press one of two buttons (unacceptable or acceptable) to answer the question 'is the quality of this sequence acceptable?' after presenting each animated sequence. After this, the next sequence comes in.

Result and Discussion. Figure 10 shows acceptable ratings for $CC1$, $CC2$ and EC in all scenes. The overall acceptable rating of $EC(9.78°,3M)$ is closest to $CC1$ compared to other items in $CC2$ and EC in five scenes. This is because when the angular radius of the fovea is 9.78°, it's difficult for participants to notice the degradation of the peripheral GI quality. Our method with $3M$ photons can maintain interactive frame rates, and the GI quality of the fovea rendered by our method with $3M$ photons is acceptable. The rendering result of $EC(9.78°,3M)$ has a high visual similarity with that of $CC1$. Acceptable ratings of $EC(,2M/3M/4M)$ are higher than that of $CC2(2M/3M/4M)$. This is because the photon densities of $EC(,2M/3M/4M)$ are higher than those of $CC2(2M/3M/4M)$ in the foveal region. Perceived GI qualities of $EC(,2M/3M/4M)$ are better than those of $CC2(2M/3M/4M)$. From the parameter point of view, with the same angular radius, when the number of photons increases, the GI quality should be better. The overall acceptable rating of $EC(9.78°,4M$ should be better than that of $EC(9.78°,3M)$. But participants feel $EC(9.78°,3M)$ is more acceptable. This is because for most complex scenes, using $4M$ photons is difficult to achieve interactive frame rates, and only in simple scenarios such as $Kitchen$ can achieve interactive frame rates. Theoretically, with the same number of photons, when the angular radius increases, the GI quality should be worse. Overall acceptable ratings of $EC(12.78°,3M)$ should be lower than that of $EC(9.78°,3M)$. But in the diffuse scene $Sponza$, the acceptable rating of $EC(12.78°,3M)$ is higher than that of $EC(9.78°,3M)$. This is because in the diffuse scene, using the same number of photons, although the angular radius increases and the photon density decreases, the error does not increase much. In $Sponza$, MSE in fovea from $EC(9.78°,3M)$ to $EC(12.78°,3M)$ only increases from 1.33×10^{-2} to 1.97×10^{-2}. Therefore, participants don't perceive the decline in GI quality.

6 Conclusion, Limitations and Future Work

We propose the foveated photon mapping method, which can efficiently render high-quality images in the foveal region for virtual scenes containing multiple materials, such as diffuse, specular, glossy, and transparent ones. Our method supports dynamic scenes, and the performance of rendering images for each eye of the VR HMDs can reach the interactive frame rates. Our method generates high-quality GI in the fovea at the cost of the periphery's quality, but the perceptual quality of our method has a high visual similarity with results rendered by the photon mapping method that has the equal image quality in the fovea. Compared with the photon mapping method using the temporal strategy [43] that has the equal GI quality in the fovea, our method achieves 2.4-4.3$\times$ speedup.

The first limitation of our method is that our method is based on the photon mapping method, so our method has the same problem as the photon mapping method, and cannot efficiently render virtual scenes with the dynamic light source or a lot of dynamic objects. This is because if the position of the light source or a large number of objects change, photons need to be traced and gathered from scratch for each frame. The second limitation is that the performance of our temporal photon management will drop in the following situations: 1) there are many dynamic objects in the scene, which cause the number of the dynamic light photon is close to that of the static light photon; 2) the viewpoint changes so rapidly that there are very few foveated photons can be reused. So one possible future work is to reuse the dynamic photons based on the idea of motion vector. The third limitation is that our method only maintains high-quality images in the foveal region, and may ignore some strong visual features in the peripheral regions. Therefore, one future work is to integrate the perceptual models into the foveated photon mapping and generate high-quality images in a wider visual acuity range. Now, our method only considers virtual scenes composed of mesh surfaces. Another possible future work is to extend the foveated photon mapping method to support large-scale volume data or translucent materials, which is more challenging for rendering performance. Because large-scale volume data rendering requires a lot of interpolation and color blending operations, and translucent material rendering calculates not only the irradiance of the hit point, but also the diffusion approximation based on the irradiance sample, all these calculations are time-consuming. Now our method achieves 42 FPS for each eye with an HTC Vive using RTX 2080 Ti. With the development of GPU, the computing power of the latest RTX 3090 GPU is 1.35$\times$ than that of the RTX 2080 Ti [39]. If the latest RTX 3090 GPU is used, it is possible to extend the foveated photon mapping to support volume data and translucent materials while maintaining interactive frame rates.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China through Projects 61932003 and 61772051, by National Key R&D plan 2019YFC1521102, by the Beijing Natural Science Foundation L182016, by the Beijing Program for International S&T Cooperation Project Z191100001619003, by the funding of Shenzhen Research Institute of Big Data(Shenzhen 518000).

REFERENCES

[1] M. Cammarano and H. W. Jensen. Time dependent photon mapping. *Rendering Techniques*, 28, 2002.

[2] J. Chen, B. Wang, and J.-H. Yong. Improved stochastic progressive photon mapping with metropolis sampling. In *Computer Graphics Forum*, vol. 30, pp. 1205–1213. Wiley Online Library, 2011.

[3] C. Collin, M. Ribardière, A. Gruson, R. Cozot, S. Pattanaik, and K. Bouatouch. Visibility-driven progressive volume photon tracing. *The Visual Computer*, 29(9):849–859, 2013.

[4] H. Dammertz, D. Sewtz, J. Hanika, and H. P. Lensch. Edge-avoiding à-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*, pp. 67–75. Citeseer, 2010.

[5] K. Dmitriev, S. Brabec, K. Myszkowski, and H.-P. Seidel. Interactive global illumination using selective photon tracing. *Rendering Techniques*, 2002:100–113, 2002.

[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] O. Elek, T. Ritschel, A. Wilkie, and H.-P. Seidel. Interactive cloud rendering using temporally coherent photon mapping. *Computers & Graphics*, 36(8):1109–1118, 2012.

[8] S. Fan, S. Chenney, Y.-c. Lai, et al. Metropolis photon sampling with optional user guidance. *Rendering Techniques*, 5(127-138):61, 2005.

[9] A. Field and G. Hole. *How to design and report experiments*. Sage, 2002.

[10] L. Franke, L. Fink, J. Martschinke, K. Selgrad, and M. Stamminger. Time-warped foveated rendering for virtual reality headsets. In *Computer Graphics Forum*. Wiley Online Library, 2020.

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

[12] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Supplement to foveated 3d graphics: User study details. *Microsoft Research, Nov*, 2012.

[13] T. Hachisuka and H. W. Jensen. Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics (TOG)*, 30(5):1–11, 2011.

[14] T. Hachisuka, S. Ogaki, and H. W. Jensen. Progressive photon mapping. In *ACM SIGGRAPH Asia 2008 papers*, pp. 1–8. 2008.

[15] 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.

[16] W. Jakob, C. Regg, and W. Jarosz. Progressive expectation-maximization for hierarchical volumetric photon mapping. In *Computer Graphics Forum*, vol. 30, pp. 1287–1297. Wiley Online Library, 2011.

[17] H. W. Jensen. Global illumination using photon maps. In *Eurographics workshop on Rendering techniques*, pp. 21–30. Springer, 1996.

[18] H. W. Jensen. *Realistic image synthesis using photon mapping*. AK Peters/CRC Press, 2001.

[19] H. W. Jensen and N. J. Christensen. A practical guide to global illumination using photon maps. *SIGGRAPH 2000 Course Notes CD-ROM*, 2000.

[20] A. Keller and I. Wald. Efficient importance sampling techniques for the photon map. 2000.

[21] 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

[22] M. Koskela, T. Viitanen, P. Jääskeläinen, and J. Takala. Foveated path tracing - a literature review and a performance gain analysis. In *ISVC*, 2016.

[23] M. K. Koskela, K. V. Immonen, T. T. Viitanen, P. O. Jääskeläinen, J. I. Multanen, and J. H. Takala. Instantaneous foveated preview for progressive monte carlo rendering. *Computational Visual Media*, 4(3):267–276, 2018.

[24] 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.

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

[26] M. Mara, D. Luebke, and M. McGuire. Toward practical real-time photon mapping: Efficient gpu density estimation. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*, pp. 71–78, 2013.

[27] M. McGuire and D. Luebke. Hardware-accelerated global illumination by image space photon mapping. In *Proceedings of the Conference on High Performance Graphics 2009*, pp. 77–89, 2009.

[28] X. Meng, R. Du, and A. Varshney. Eye-dominance-guided foveated rendering. *IEEE Transaction on Visualization and Computer Graphics*, 26:1972–1980, 2020.

[29] 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.

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

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

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

[33] K. Myszkowski, T. Tawara, H. Akamine, and H.-P. Seidel. Perception-guided global illumination solution for animation rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 221–230, 2001.

[34] 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.

[35] 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.

[36] 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.

[37] S. A. Pedersen. Progressive photon mapping on gpus. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2013.

[38] I. Peter and G. Pietrek. Importance driven construction of photon maps. In *Eurographics Workshop on Rendering Techniques*, pp. 269–280. Springer, 1998.

[39] ROU-User. Userbenchmark. Website, 2021. `https://gpu.userbenchmark.com/Compare/Nvidia-RTX-3090-vs-Nvidia-RTX-2080-Ti/4081vs4027`.

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

[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] T. Tawara, K. Myszkowski, and H.-P. Seidel. Localizing the final gathering for dynamic scenes using the photon map. In *VMV*, pp. 69–46, 2002.

[44] T. Tawara, K. Myszkowski, and H.-P. Seidel. Exploiting temporal coherence in final gathering for dynamic scenes. In *Proceedings Computer Graphics International, 2004.*, pp. 110–119. IEEE, 2004.

[45] M. Teschner. Advanced computer graphics path tracing. Website, 2021. `https://cg.informatik.uni-freiburg.de/course_notes/graphics2_09_pathTracing.pdf`.

[46] 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.

[47] L. Wang, R. Li, X. Shi, L. Yan, Z. Li, and P. Cheng. Foveated instant radiosity. 2020.

[48] R. Wang, R. Wang, K. Zhou, M. Pan, and H. Bao. An efficient gpu-based approach for interactive global illumination. In *ACM SIGGRAPH 2009 papers*, pp. 1–8. 2009.

[49] M. Weber, M. Milch, K. Myszkowski, K. Dmitriev, P. Rokita, and H.-P. Seidel. Spatio-temporal photon density estimation using bilateral filtering. In *Proceedings Computer Graphics International, 2004.*, pp. 120–127. IEEE, 2004.

[50] 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

[51] M. Weiss and T. Grosch. Stochastic progressive photon mapping for dynamic scenes. In *Computer Graphics Forum*, vol. 31, pp. 719–726.

Wiley Online Library, 2012.
[52] Wikipedia. Photon mapping, February 2021.
[53] 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.
[54] Q. Zheng and C.-W. Zheng. Visual importance-based adaptive photon tracing. *The Visual Computer*, 31(6-8):1001–1010, 2015.
[55] K. Zhou, Q. Hou, R. Wang, and B. Guo. Real-time kd-tree construction on graphics hardware. *ACM Transactions on Graphics (TOG)*, 27(5):1–11, 2008.
[56] S. Zhu, Z. Xu, H. W. Jensen, H. Su, and R. Ramamoorthi. Deep photon mapping. *arXiv preprint arXiv:2004.12069*, 2020.