Computers & Graphics 97 (2021) 200–207

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CAD & Graphics 2021

Foveated light culling

Qinqi Yang, Zhuxin Chen, Yanli Liu, Guanyu Xing, Yanci Zhang*

School of Computer Science, Sichuan University, Chengdu, 610065, China

ARTICLE INFO

Article history:
Received 8 February 2021
Revised 6 April 2021
Accepted 13 April 2021
Available online 24 April 2021

Keywords:
Virtual reality
Foveated rendering
Global illumination

ABSTRACT

In this paper, we propose a novel *Foveated Light Culling* (FLC) method to efficiently approximate global illumination for foveated rendering in virtual reality applications. The key idea is to cull the virtual point lights (VPLs) in accordance with their contribution to the visual perception, which makes the loss of details caused is imperceptible to the human eyes. Based on this idea and the features of human visual system (HVS), we calculate the contribution of each VPL according to samples generated from a perceptual importance map. By estimating the local visibility between VPLs and samples, our method can greatly improve the performance during the process of evaluating contributions. Our method is able to produce high-quality single-bounce diffuse-to-diffuse indirect illumination in the foveal region at real-time frame rate while supporting dynamic scenes and light sources. Furthermore, the experiments show that our approach makes a good balance between foveal quality and render efficiency.

1. Introduction

Head mounted displays (HMDs) are one of the most popular display devices for virtual reality (VR) or augmented reality (AR) applications. Different from rendering images on normal displays, generating images on HMDs has more chances to exploit the features of HVS to improve the rendering efficiency/quality. For instance, even though HVS has a very wide field of view (160° horizontally and 135° vertically), it just can resolve the fine details within a 5° central circle [1]. The visual acuity of HVS decreases radially between the gaze point (the fovea) and the periphery. Equipped with an eye tracker, foveated rendering systems [2–4] significantly improve the rendering performance by decreasing shading quality in the peripheral regions while maintaining high fidelity in the foveal area.

Meanwhile, with the increasing demand for photorealistic rendering, global illumination becomes the major trend for almost all graphics application, including generating images on HMDs. Many methods have been proposed to combine global illumination with foveated rendering. For example, Wang et al. [5] has proposed a model to adapt instant radiosity into foveated rendering, which generates VPLs from the camera to provide indirect illumination to the foveal region. Koskela et al. [6] render scenes in visual-polar space, thereby reducing the amount of computation for path tracing. However, finding a good balance between the rendering quality and performance for foveated rendering is still an open problem.

In this paper, we propose a method based on *Instant Radiosity* [7] to efficiently produce indirect illumination for foveal region. Although [5] is a similar idea, its results are poor in many scenes due to two obvious problems: The first is that it generates VPLs from the camera, making it difficult for VPLs to guarantee their contributions to the foveal region. The second problem is that it only considers the relationship of visibility and locations between VPLs and the foveal region when calculating the importance, which results in great errors.

In our work, we adopt a different VPLs generation method to ensure that they have enough intensities to provide indirect illumination for the foveal region. In order to make a good balance between visual details and performance, we propose an efficient method to evaluate the contributions of VPLs to the final image quality, which is majoritively determined by the foveal region. Based on these contributions, the important VPLs have higher chances to be kept to compute indirect illumination. More specifically, we propose the following two strategies to achieve a good balance between rendering performance and quality. The first strategy is an importance map based on HVS features to describe the importance of each pixel to perception of indirect illumination. And the second strategy is an efficient method to fast compute the visibilities between VPLs and shading points.

* Corresponding author.
E-mail addresses: 2019223040026@stu.scu.edu.cn (Q. Yang), 2019223045135@stu.scu.edu.cn (Z. Chen), yanliliu@scu.edu.cn (Y. Liu), guanyuxing@scu.edu.cn (G. Xing), yczhang@scu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.cag.2021.04.021
0097-8493/© 2021 Elsevier Ltd. All rights reserved.

2. Related work

In this section, we provide an overview of main fields of researches related to our work.

Foveated 3D Rendering is proposed for VR applications [1], which improves rendering efficiency by reducing the image quality in the periphery. Many researchers focus on geometric simplification. Lindeberg [8] proposed a novel technique to adjust tessellation levels. Cheng [2] presented enhancements to 3D model simplification based on interactive level-of-detail (LOD) update with foveation. Weier et al. [3] had reviewed some methods focusing on mesh simplification in the peripheral region [9,10].

Multi-resolution and multi-rate shading are the strategies which enable the adaption of shading complexity to the scene content. Guenter et al. [1] proposed a three-pass pipeline for foveated 3D rendering by using three nested layers. Tursun et al. [11] presented a luminance contrast aware algorithm to analyze displayed luminance contrast. Vaidyanathan et al. [12] proposed a method to sample coarse pixels in the peripheral region. He et al. [4] introduced multi-rate GPU shading, which enables more shading samples near regions of specular highlights, shadows, edges, and motion blur regions. Our method is also based on multi-resolution, the choosed VPLs are contributing more to the foveal region and less to the periphery.

Instant Radiosity [7] is a method that approximates the indirect illumination within real-time requirements. Many researchers worked on the strategies of VPLs distribution [13–15]. Some other research also dedicated in estimation of VPLs' contributions to the final output image. Segovia et al. [14] presented a coarse approach that regards visibility as the contribution. Hedman et al. [16] proposed a mail-boxing scheme to determine indirect visibility for static scenes. Wang et al. [5] emitted the rays from camera to distribute VPLs. We integrate the instant radiosity method into foveated rendering framework and distribute VPLs by stochastically culling them according to their estimated contributions to the foveal region.

Visual Perception Model describes the details that can actually be perceived by the observer. Weymouth [17] had shown an approximately linear degradation behaviour of acuity with eccentricity angular. Longhurst et al. [18] proposed a fast approach to extract salliencies for visual cues by pre-rendering a low resolution frame. Stengel et al. [19] introduced a flexible image-space sampling strategy to incorporate more visual cues in perceptual model to sparsely sample shading points. In our method, we generate an importance map based on HVS to guide the sampling of shading points which used to estimate the contributions of VPLs.

Light Culling is a technique with clamping light ranges to improve rendering efficiency in real-time applications. Harada et al. [20] presented a forward+ method to support rendering many lights by storing only lights that contribute to the pixel. Harada [21] also proposed a 2.5D culling for further improving the forward+ rendering pipeline. Tokuyoshi and Harada [22] introduced a stochastic light culling method that randomly determines a light range to avoid undesirable bias. O'Donnell and Chajdas [23] presented a new approach called tiled light tree that adapts to the light source distribution.

In this paper, we apply foveation-based rendering to *Instant Radiosity*, which can produce plausible global illumination in VR applications. It also reaches the real-time frame rates by culling VPLs based on the visual perception of HVS. In contrast to previous work, our culling method is based on the features of HVS, not just the geometric information in the image space, which means that the rest VPLs used for shading contribute more to the foveal region.

3. Algorithm overview

In this paper, we try to generate high quality single-bounce indirect illumination in the foveal region $\mathbb{F}$ at the cost of image quality in peripheral region $\mathbb{P}$ by instant radiosity method. The basic idea is to stochastically cull the VPLs according to their contributions to the final image quality, which is majoritively determined by the image quality of $\mathbb{F}$.

Generally speaking, the contribution $C_{l_v}(\mathbb{I})$ of a VPL l_v to the final image quality can be described as:

$$C_{l_v}(\mathbb{I}) = w_{\mathbb{F}}C(l_v \rightarrow \mathbb{F}) + w_{\mathbb{P}}C(l_v \rightarrow \mathbb{P}) \tag{1}$$

where $\mathbb{I} = \mathbb{F} \bigcup \mathbb{P}$ represents an image, $w_{\mathbb{F}}$, and $w_{\mathbb{P}}$ is the weight to $\mathbb{F}$ and $\mathbb{P}$ respectively. The contribution $C(l_v \rightarrow \mathbb{R})$ of VPL l_v to a pixel region $\mathbb{R}$ can be computed from:

$$\begin{aligned} C(l_v \rightarrow \mathbb{R}) &= \frac{1}{N}\sum_{i=1}^{N} f(l_v \rightarrow p_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} f(I_{l_v}, V_i, G_i, \mathfrak{p}_i) \end{aligned} \tag{2}$$

where f represents the VPL's contribution to pixel $p_i \in \mathbb{R}$, N is the number of pixels in $\mathbb{R}$, and I_{l_v} is the intensity of VPL l_v. $\mathfrak{p}_i, V_i$ and G_i represent the weight of pixel p_i, the visibility and geometry term between VPL and pixel p_i respectively.

However, it is impractical to directly employ Eqs. (1) and (2) to evaluate the contribution of l_v to image $\mathbb{I}$ because of the high computation cost. For instance, it involves all pixels on $\mathbb{I}$, the term $V(l_v, p_i)$ needs to solve visibilities between all VPLs and shading points p_i and so on.

In this paper, we propose a novel method called *Foveated Light Culling* to efficiently approximate $C_{l_v}(\mathbb{I})$. Based on $C_{l_v}(\mathbb{I})$, VPLs are stochastically culled and the remained VPLs are employed to generate the indirect illumination. Note that $C_{l_v}(\mathbb{I})$ is majoritively determined by the foveal region $\mathbb{F}$, thus the image quality of $\mathbb{F}$ can be guaranteed.

The procedure of our algorithm is as illustrated in Fig. 1. Our method focuses on solving the following two issues to achieve a good balance between the final image quality and rendering performance:

- *How to reduce the number of pixels involved in the computation of $C_{l_v}(\mathbb{I})$ without compromising the quality of perception?* According to the features of HVS, the following three factors are important to visual perception: visual acuity, geometry silhouette and indirect illumination. Based on these factors, we compute an importance map for image $\mathbb{I}$ to describe the importance of each pixel to perception of indirect illumination. And then, an importance sampling process is conduced according to the importance map to choose pixels as samples p' to participate in the computation of Eq. (1) (S-3 and S-4 in Fig. 1 and see Section 4 for more details);
- *How to efficiently estimate visibilities between VPLs and pixels?* Visibilities between VPLs and pixels play an important role in the computation of indirect illumination. Though the modern GPUs support hardware ray tracing to solve this problem, it is still impractical to conduct such huge number of visibility tests in real-time. In this paper, we propose a *Bidirectional Local-Space Occlusion*(BLSO) method to efficiently find an approximate solution of the visibility problem by exploiting local occlusion information (see Section 5 for more details).

In addition, we need G-Buffer to extract the geometry information of each pixel and RSMs to generate VPLs (S-1 and S-2 in Fig. 1). After culling the low-contribution VPLs, we use the rest to

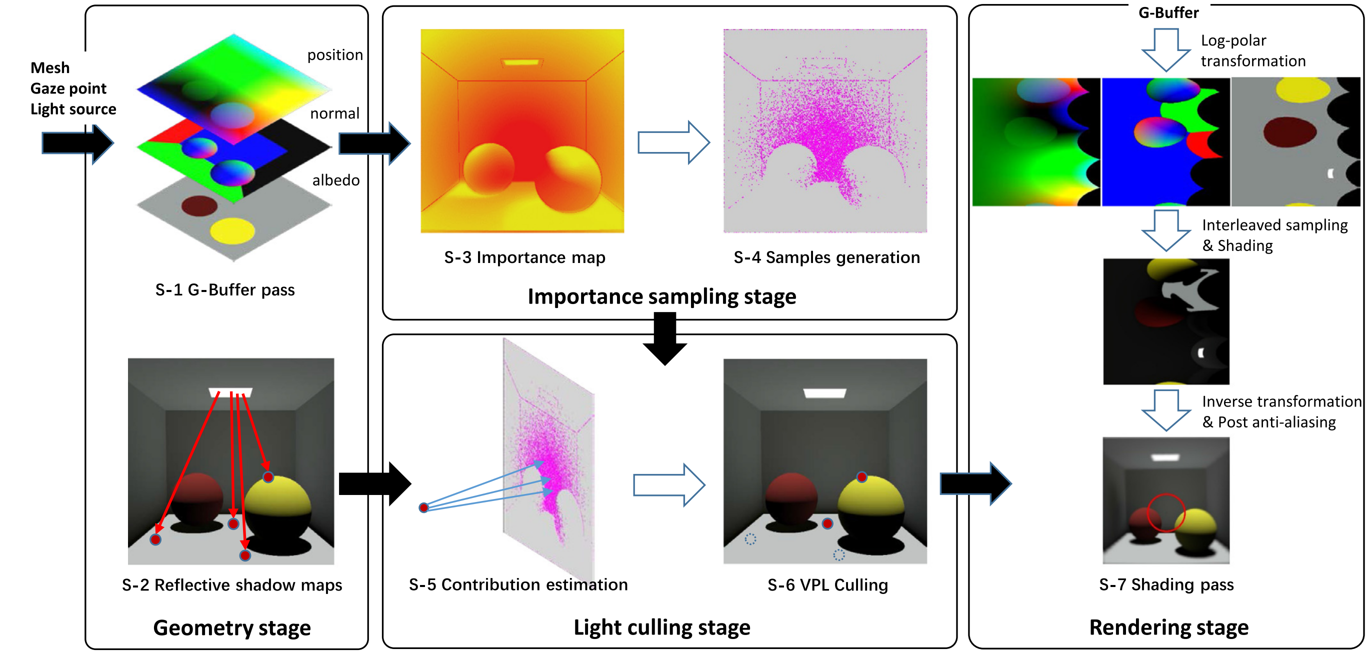

Fig. 1. Overflow of our algorithm: (1) Geometry stage: generating G-buffer and RSMs for scene and light sources respectively; (2) Importance sampling stage: creating perceptual probability map and choosing pixels as samples based on this map; (3) Light culling stage: estimating the contribution of each VPL and stochastically culling VPLs according to their contributions; (4) Rendering stage: shading with rest VPLs using accelerated algorithms.

render the scenes with interleaved sampling and log-polar transformation (S-7 in Fig. 1 and see Section 6 for more details).

4. Sampling of shading points

In this section, we present our method to reduce the number of pixels involved in the computation of C_{l_v}. The basic idea is to compute an importance map to describe the importance of each pixel p_i in the evaluation of $C_{l_v}(\mathbb{I})$ based on its own properties. For example, considering two pixels $p_1 \in \mathbb{F}$ and $p_2 \in \mathbb{P}$, p_1 should be more important than p_2 because our priority is to guarantee the image quality in $\mathbb{F}$.

4.1. Generation of importance map

In our algorithm, we compute an importance value $\mathfrak{p} \in [0, 1]$ for each pixel, i.e. the weight of each sample. This value only depends on pixel's own properties, like position in image space, and has nothing to do with VPLs. Meanwhile, the properties of a pixel that we considered are mainly based on HVS. In fact, whether it is a natural scene or a virtual scene, the final image in the HVS is two-dimensional, and the human eye can only be sensitive to a small area of an image at any given time. In other words, the theory of visual perception is determined by the structure and function of the human eye itself and does not depend on the external scene. Inspired by Stengel et al. [19], we propose to use visual acuity, geometry silhouette and indirect brightness to compute the importance. And all the importance values form an importance map P which has the same resolution as image $\mathbb{I}$.

4.1.1. Visual acuity

Visual acuity rates the ability to recognize small details. Many researches have proved that the acuity decays linearly within a certain range [17,24], which is often used in the foveated rendering methods [1,12,19] to simulate the reduction of resolvable resolution away from gaze position.

In order to approximate the attenuation of acuity to eccentricity angle, we assume that the acuity falls off linearly throughout the field of view. We propose to use Eq. (3) to describe the acuity importance $\mathfrak{p}_a(p)$ of pixel p, which can be applied to arbitrary gaze position c_g in image space.

$$\mathfrak{p}_a(p) = \mathrm{clamp}(h(c_p, c_g), \omega_p, 1.0) \tag{3}$$

where c_p is the 2D image coordinate of pixel p, ω_p is a constant acuity in the outer periphery, $h(c_p, c_g)$ is employed to simulate the falloff of the acuity, which can be written as:

$$h(c_p, c_g) = 1.0 + s \cdot (m(r, \alpha) - e(c_p, c_g)) \tag{4}$$

where acuity slope s is a user-defined value, $e(c_p, c_g)$ represents the eccentricity angle. The function $m(r, \alpha)$ is used to adjust the range of the foveal region based on the ratio r of foveal region to screen and radian α of FOV:

$$m(r, \alpha) = atan\left(\sqrt{r} \cdot tan\frac{\alpha}{2}\right) \tag{5}$$

As for the eccentricity angle $e(c_p, c_g)$, it can be computed as:

$$\begin{aligned} k(c_p, c_g) &= \frac{||c_p - c_g||_2}{||\mathfrak{d}||_2} \\ e(c_p, c_g) &= atan\left(k(c_p, c_g) \cdot tan\frac{\alpha}{2}\right) \end{aligned} \tag{6}$$

where $\mathfrak{d} = (width/2, height/2)$ is the half size of the display screen. $k(c_p, c_g)$ represents the ratio of the projection of the eccentricity angle. The approximate falloff of the acuity is shown in Fig. 2a.

4.1.2. Geometry silhouette

A lot of studies suggest that the shape of an object can affect the perceived illumination [25,26]. For instance, sphere is better than other simple shapes to interact with light, which is widely used to display the materials in many 3D engines.

In our algorithm, we make use of the G-buffer to detect the silhouette by computing the gradient of depth, the angle between normals and the color difference $\triangle c$ which is defined as the weighted Euclidean distance between the pixel color $\mathfrak{c}$ and the neighbor color $\hat{c}$ in RGB space, as illustrated in Eq. (7), these weights here are empirical results based on what the human eye

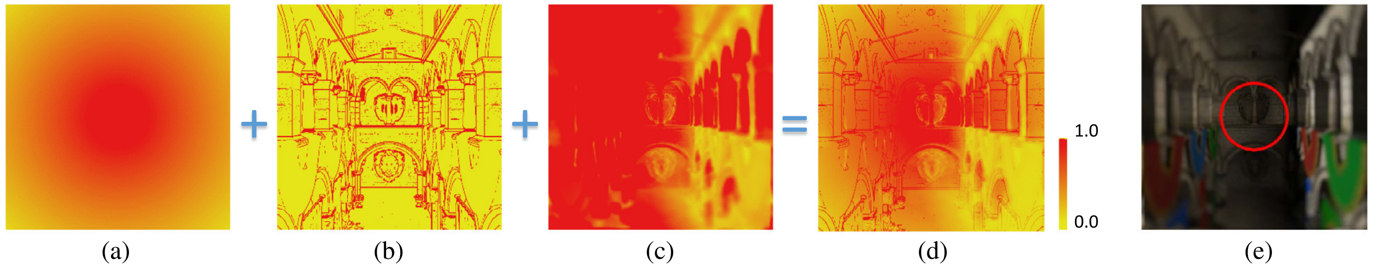

Fig. 2. Importance Map. (a) Visual acuity importance map. (b) Geometry silhouette importance map. (c) Indirect brightness importance map. (d) Final importance map. (e) Reference scene with foveal region (in red circle).

perceives. Once the value of one of the three channels exceeds the given threshold, we treat the pixel as geometry silhouette and set a constant value. From this, we can generate a sampling importance $\mathfrak{p}_g$ (Fig. 2b).

$$\begin{aligned} \bar{r} &= \mathfrak{c}.r - \hat{c}.r \\ \triangle r &= 256 * (\mathfrak{c}.r - \hat{c}.r) \\ \triangle g &= 256 * (\mathfrak{c}.g - \hat{c}.g) \\ \triangle b &= 256 * (\mathfrak{c}.b - \hat{c}.b) \\ \triangle c &= \sqrt{(2+\bar{r}) \cdot \triangle r^2 + 4 \cdot \triangle g^2 + (3-\bar{r}) \cdot \triangle b^2} \end{aligned} \tag{7}$$

4.1.3. Indirect brightness

Brightness has a great impact on the perception of HVS and visual acuity [19]. For example, HVS is more sensitive to brightness, which is widely employed in video encoding to compress the video size. In our paper, we use the indirect brightness to define importance value $\mathfrak{p}_b$ to model this phenomena. Unfortunately, the indirect brightness is unknown when we compute $\mathfrak{p}_b$. We use the accumulated indirect brightness $b(p)$ of history frames to calculate $\mathfrak{p}_b$ (Fig. 2c), as illustrated in Eq. (8):

$$\mathfrak{p}_b = min(1, \delta \cdot b(p)) \tag{8}$$

where $b(p)$ is the tone-mapped brightness of pixel p, and δ represents the influence factor of indirect brightness.

4.2. Choosing Shading Points Based on Importance Map

Based on $\mathfrak{p}_a, \mathfrak{p}_g, \mathfrak{p}_b$, the importance value for pixel p is defined as:

$$\mathfrak{p}(p) = \max(\mathfrak{p}_a, \mathfrak{p}_g) \cdot \mathfrak{p}_b \tag{9}$$

After the importance map P is achieved, we modify the Metropolis-Hasting (M-H) [27] method to choose pixels from $\mathbb{I}$ according to P. The traditional M-H method needs to compute an ideal Markov chain and the quality of the initial state will seriously affect the convergence speed [28]. And its performance also does not meet our real-time requirement.

Since our importance map can be treated as a kind of probability map, we extend M-H sampling strategy to fit our method. At the same time, in order to improve the sampling efficiency, we implement the sampling process on GPU. Algorithm 1 presents the pseudo code of our GPU-based sampling algorithm.

Our sampling strategy is based on N_s parallel threads, where N_s is the number of desired samples. To begin with, we use Halton sequence to generate $N_s * M$ 2D quasi-random numbers and shuffle them, where each 2D number can be treated as the image coordinate of a proposal sample and M is the length of each Markov chain. In thread $j \in [0, 1)$, we first get the image coordinate and importance value of a sample (line 2,3) and set it as the accepted pixel. Then we perform $M - 1$ iterations (line 4–9). For each iteration, we also get the coordinate and importance value of a new proposal sample (line 5,6) and compare its importance value with that of the accepted sample (line 7) to choose a greater one as the new accepted sample (line 8,9). Thus, we can generate a most important sample in each thread (line 10). After all threads ended, we can generate N_s samples which have low-discrepancy and low-correlation.

Algorithm 1: Modified Quasi-Metropolis Hasting Algorithm.

```
Input: iteration number for each Markov chain $M$,
       two-dimensional shuffled Halton values $H$, importance
       map $P$, G-buffer $G$
Output: generated sample $S$
1  $j \leftarrow threadId$;
2  $coord \leftarrow H[j * M]$;
3  $w \leftarrow P(coord)$;
4  for $i \leftarrow 1$ to $M$ do
5      $coord_{new} \leftarrow H[j * M + i]$;
6      $w_{new} \leftarrow P(coord_{new})$;
7      if $w_{new} > w$ then
8          $coord \leftarrow coord_{new}$;
9          $w \leftarrow w_{new}$;
10 $S \leftarrow generateSampleFromGBuffer(G, coord, w)$;
11 return $S$;
```

5. Stochastically light culling

In this section, we present our method to cull VPLs based on their contributions to the final image quality. We firstly extend the Eq. 2 to a more specific form, and elaborate on our visibility estimation method to accelerate visibility computation involved in C_{l_v}. After that, we describe our strategy to cull VPLs stochastically.

The contribution of a VPL l_v to the final image quality is defined by Eq. 1. As we have generated N_s samples by Algorithm 1 according to importance weight $\mathfrak{p}$ calculated by Eq. (9), the contribution computation of l_v is further extended from Eq. (2) as:

$$\begin{aligned} C(l_v \to \mathbb{R}) &= \frac{1}{N}\sum_{i=1}^{N} f(I_{l_v}, V_i, G_i, \mathfrak{p}_i) \\ &\approx \frac{1}{N_s}\sum_{i=1}^{N_s} \frac{I_{l_v} V_i cos(\theta_{p'_i}) cos(\theta_{l_v}) \mathfrak{p}_i}{d_i^2 + \epsilon} \end{aligned} \tag{10}$$

where $G_i = \frac{cos(\theta_{p'_i})cos(\theta_{l_v})}{d_i^2+\epsilon}$ is the geometry term, $\mathfrak{p}_i$ is the weight of sample p'_i, and d_i is the distance between p'_i and l_v. $\theta_{p'_i}$ is the angle between $\vec{n}_{p'_i}$ and the transmission direction, θ_{l_v} is the angle between $\vec{n}_{l_v}$ and the transmission direction, where $\vec{n}$ indicates normal vector. ϵ is a number to prevent mathematical singularities, and V_i is the visibility between p'_i and l_v.

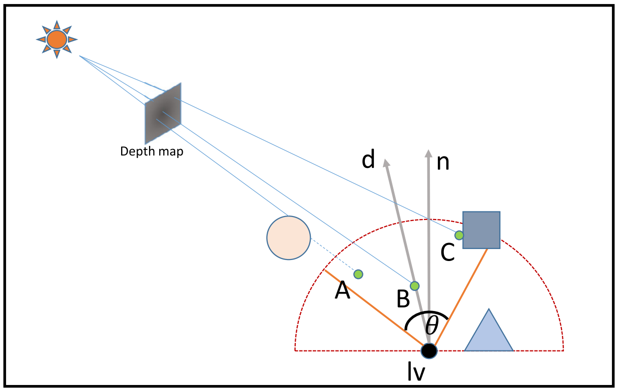

Fig. 3. Visibility Estimation. l_v is the VPL, n represents the normal, d is the direction needed to estimate the visibility probability, and A, B and C represent the sampled position in local-space, the dotted red circle is the local hemisphere space.

Evaluating the visibility term V_i in Eq. (10) is computationally expensive even with the hardware ray tracing. In this paper, we propose BLSO method to approximate V_i by local space occlusion information, which involves both VPL l_v and sample p'_i. Then the visibility V_i is calculated as:

$$V_i = mix(T_{l_v}(\vec{d}), T_{p'_i}(-\vec{d}), \lambda) \tag{11}$$

where $\vec{d}$ represents the direction from l_v to p'_i, the function T_{l_v} and $T_{p'_i}$ represent the l_v and p'_i visibility probability calculated by our method, λ is a user-defined value. Due to the same calculation way for T_{l_v} and $T_{p'_i}$, we take T_{l_v} as an example to further describe the details.

As illustrated in Fig. 3, we sample some random positions in the local space of the solid angle which regard $\vec{d}$ as the central axis, and then calculate the visibility probability of each sample (A, B and C as examples) one by one. For A, it is invisible to the light source, so we think l_v may be occluded in this direction; But for B, it is visible to the light source, so we consider this direction has no occlusion; And l_v is obvious occluded in this direction from it to C because C is on the surface of an object. Finally, we calculate the visibility for arbitrary sample position s as:

$$g(s) = \begin{cases} 1 & \triangle h \le -\xi \\ 0 & -\xi < \triangle h < \xi \\ 1 - e^{-k\sqrt{\triangle h}} & else \end{cases} \tag{12}$$

where $\triangle h$ represents the difference of depths in light source space between s and the corresponding value according to the projection position of s from shadow map, it could be positive, negative, or zero, k is the attenuation factor, ξ is a small positive threshold related to the scene.

Finally, we estimate the visibility probability of l_v in direction $\vec{d}$ by accumulate the probabilities of all sampled positions as:

$$T_{l_v}(\vec{d}) = \frac{1}{N'} \sum_{i=1}^{N'} g(i) \tag{13}$$

where N' is the number of sampled positions. And the process of calculating $T_{p'_i}$ is the same.

After we approximate the visibility term in Eq. (10), we can give the final contribution evaluation for l_v as:

$$\begin{aligned} C(l_v \to \mathbb{R}) &\approx \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{I_{l_v} V_i cos(\theta_{p'_i}) cos(\theta_{l_v}) \mathfrak{p}_i}{d_i^2 + \epsilon} \\ &\approx \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{I_{l_v} mix(T_{l_v}[\vec{d}], T_{p'_i}[-\vec{d}], \lambda) cos(\theta_{p'_i}) cos(\theta_{l_v}) \mathfrak{p}_i}{d_i^2 + \epsilon} \end{aligned} \tag{14}$$

Based on Eq. (14), we use a sampling strategy similar to [29] to stochastically cull VPLs.

6. Other implementation details

In order to further reduce shading cost, we adapt *log-polar transformation* (LPT) [30] to accelerate rendering (Fig. 1 S-7). However, the forward transformation strategy used in LPT may result in a many-to-one map from Cartesian space to log-polar space, causing some pixels in log-polar space cannot get corresponding values. We use the backward strategy to transform the textures in G-Buffer based on their inverse log-polar transformation, which can be described as taking the corresponding values from Cartesian space to log-polar space.

During the shading process, we also perform *interleaved sampling* method [31] in log-polar space to improve efficiency. As a result, we have to use a Gaussian filter with a 3 × 3 kernel in the whole space to reduce artifacts. Moreover, due to its inherent problems of LPT, there will be flickering in peripheral regions after the inverse transformation from log-polar space to Cartesian space. In order to alleviate flickers, we apply *temporal anti-aliasing* (TAA) with different clamp size η for different pixel:

$$\eta = 3 + 2 \times \left\lfloor \frac{\frac{\|x',y'\|_2}{e^L} - 0.10}{0.05} \right\rfloor \tag{15}$$

where $x' = c_p.x - c_g.x$ and $y' = c_p.y - c_g.y$. L is the maximum distance between gaze position c_g and one of the four screen corners. The several constants here are set to determine the range and change rate of η, which affects the quality and efficiency of TAA.

7. Experimental results

The method proposed in this paper has been implemented on a PC with RTX 2080 Ti GPU, i9-9900K CPU and 16GB memory. We also employ a HTC Vive equipped with binocular eye-tracking as the display device. We compare our method with *Foveated Instant Radiosity* (FIR) [5] and *Tile-Based Deferred Shading* (TBDS) [32] in quality and performance. Meanwhile, we use the results of TBDS as the ground truth images because it takes all the VPLs who have contributions to current tile into consideration. RMSE (Root Mean Square Error) is employed to quantitatively measure the differences between the rendered and ground truth images.

All the results are rendered at 1024 × 1024 resolution, and the resolution of RSMs is 64 × 64. Other parameters are set as follows: the radius of foveal region is 150 pixels, the outer periphery acuity $\omega_p = 0.2$ (Eq. (3)), the acuity slope $s = 2.25$ (Eq. (4)), the indirect brightness influence factor $\delta = 20$ (Eq. (8)), the mixing coefficient $\lambda = 0.5$ (Eq. (11)), and attenuation factor $k = 34$ (Eq. (12)).

Fig. 4 illustrates the images generated by TBDS(column 1), FLC (column 2) and FIR (column 3) for different scenes, where the foveal regions are marked by red circles. We also enlarge the regions marked by green rectangles to compare the details (last column of Fig. 4). Additionally, RMSEs of the same foveal regions indicated in Fig. 4 are as shown in Fig. 5. The results demonstrated in Figs. 4 and 5 suggest the following facts:

- Both Figs. 4 and 5 indicate that our method produces better image quality in the foveal region than FIR. This is because the importance computation in FIR ignores the intensities of VPLs. For this reason, even though the VPLs used for shading are close to the foveal region, many of them cannot provide sufficient indirect illumination due to low intensity.
- Comparing the color of wall in the first row of Fig. 4, we can find that the color bleeding effects generated by our method is

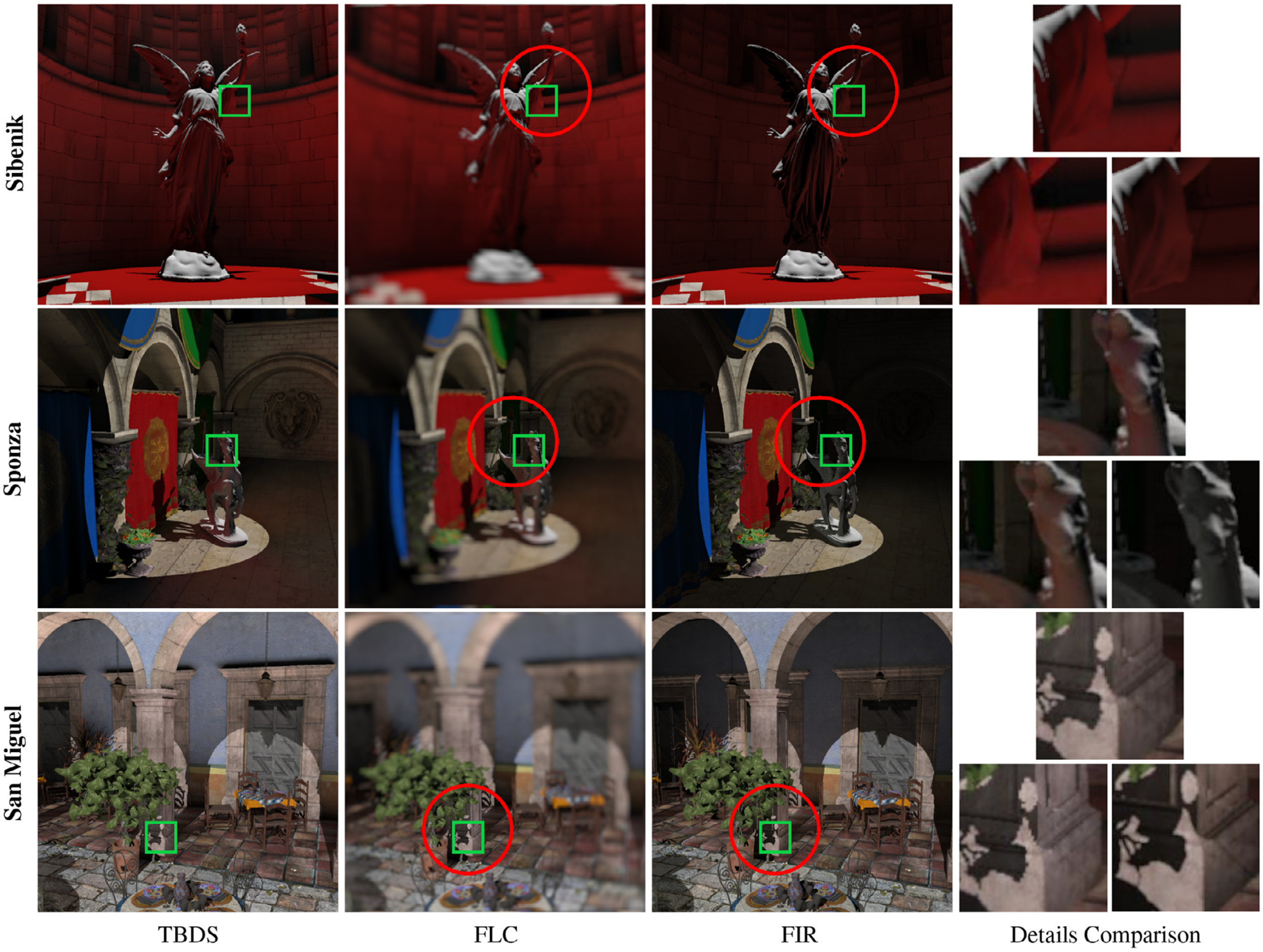

Fig. 4. Render results from TBDS (column 1), our method (column 2), FIR (column 3). The details in green squares are magnified in column 4 (top: TBDS; bottom-left: FLC; bottom-right: FIR). The red circles indicate the foveal regions.

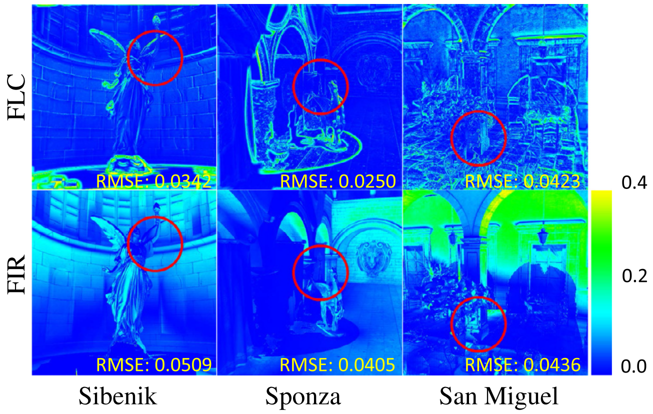

Fig. 5. Visualization of RMSE compared to ground truth for the images rendered with our FLC (row 1) and FIR (row 2) in different scenes. The red circles represent the foveal regions. The value at the bottom-right in every image represents the RMSE in the foveal region.

better than FIR. This is because the VPLs of our method are generated from RSMs, which provide the single-bounce indirect illumination. On the contrary, the VPLs of FIR are generated from camera so that they are multi-bounced and their intensities are very low. Thus, the color bleeding effects generated by FIR are poor.

Table 1
Comparison of RMSE in different scenes.

Method	Sibenik		Sponza		SanMiguel	
	Fovea	Periphery	Fovea	Periphery	Fovea	Periphery
FLC	0.03421	0.04394	0.02501	0.03079	0.04231	0.06199
FIR	0.05093	0.05377	0.04047	0.04473	0.04364	0.07319

- Our method generates more blurry image in the peripheral regions than FIR, which is more in line with the principle of foveated rendering. The major reason causing the blur effects in the peripheral regions is our strategy of conducting shading in log-polar space.

Table 1 shows more comparisons between FLC and FIR on the RMSEs of foveal and peripheral regions for different scenes. We compute the ratio $\mathfrak{r} = RMSE_P/RMSE_F$ to indicate how much computation power is dedicated to the image quality of foveal region. The bigger $\mathfrak{r}$ is, the more computation resources are assigned to the foveal region. The average ratio for three scenes of our method is 1.33, which is larger than the corresponding value 1.28 of FIR. This fact proves that our method assigns more computation resources to the foveal region than FIR.

Fig. 6 compares the rendering performance between our method with FIR and TBDS. It can been seen that our method is slightly faster than FIR. And comparing with TBDS, our method has

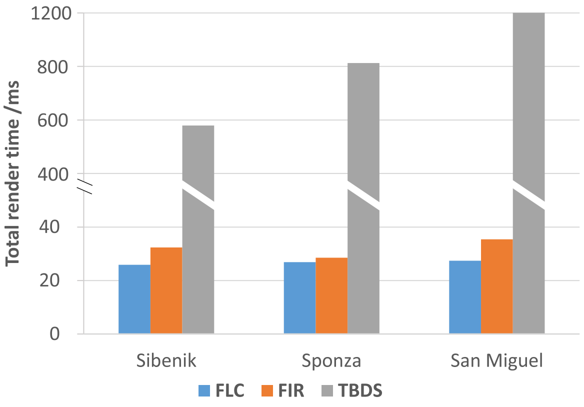

Fig. 6. The total render time of different methods in different scenes.

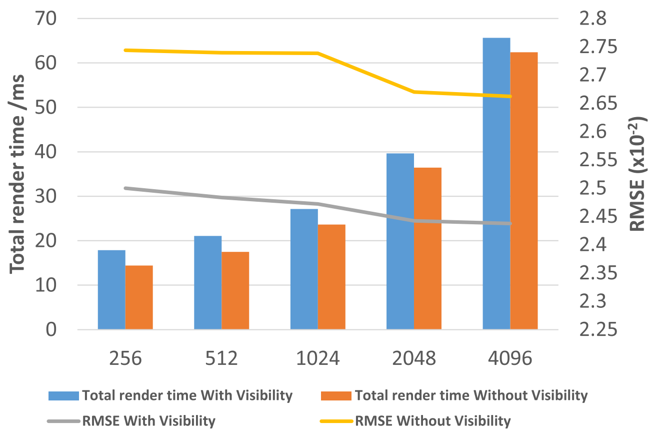

Fig. 7. Comparison of total render time and RMSE between our method with and without BLSO under different VPL numbers in Sponza scene.

Table 2
Computation time for every pass in different scenes of FLC.

Render Passes (ms)	Sibenik	Sponza	SanMiguel
G-buffer	0.340	0.466	1.032
RSMs	0.669	0.847	1.375
Importance Sampling	0.313	0.382	0.391
Contribution Estimating	7.089	6.454	6.922
Light Culling	0.014	0.014	0.014
Shading	12.367	13.051	12.195
Post-Antialiasing	5.087	5.646	5.454
Total	**25.879**	**26.860**	**27.383**

a huge performance advantage because TBDS actually uses all valid VPLs in the shading computation which is too time-consuming.

As illustrated in Table 1 and Fig. 6, our method is better in both quality and efficiency. This is because we employ more precise contribution estimation method to cull VPLs and adapt BLSO to approximate visibility. Moreover, our method can generate similar results to TBDS and achieves real-time frame rates.

We also measure the performance details of our method for different scenes. The results are shown in Table.2. It can been seen that the three most expensive stages are: contribution estimating, shading and post-antialiasing. For the contribution estimating stage, it needs to evaluate Eq. (14) for all VPLs, including the visibility computations. As for the post-antialiasing stage, the high computation cost is because of *Gaussian Blur* and TAA.

Furthermore, in order to analyze the impact of the number of VPLs remained to participate shading on the rendering performance and quality of our method, we use multiple configuration of this number from 256 to 4096 for the Sponza scene. The results are shown in Fig. 7, which indicate that the rendering cost increases almost quadratically with the number of VPLs. However, the RMSEs decrease slowly when the number of VPLs increases from 256 to 1024. Moreover, the RMSEs decrease sharply between 1024 and 2048 and then stabilize again. Therefore, in order to balance the foveal quality and performance, a number between 1024 and 2048 is a proper choice for shading. As illustrated in Fig. 7, turning off our BLSO strategy will greatly increase the RMSEs. Our BLSO method can significantly improve the foveal quality without hurting the performance too much.

Meanwhile, we also compare the rendering efficiency and RMSEs of our method with and without LPT and interleaved sampling, which is illustrated in Fig. 9. The results show that even though LPT and interleaved sampling can greatly improve our shading performance, the RMSEs of our method are mainly caused by them, which will produce a lot of aliasing at geometry silhouette due to downsampling and *Gaussian Blur*.

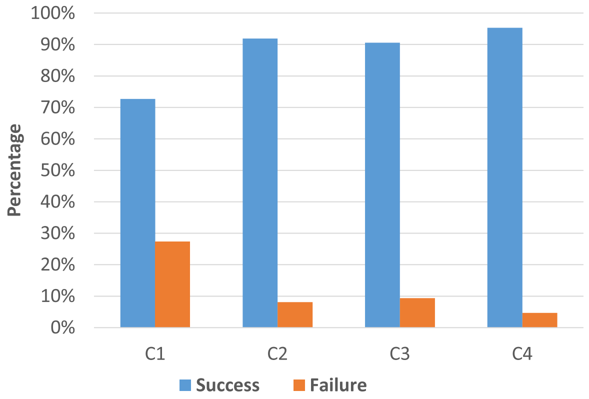

Fig. 8. The percentage of participants' visual perception that our method succeeds or fails to confuse.

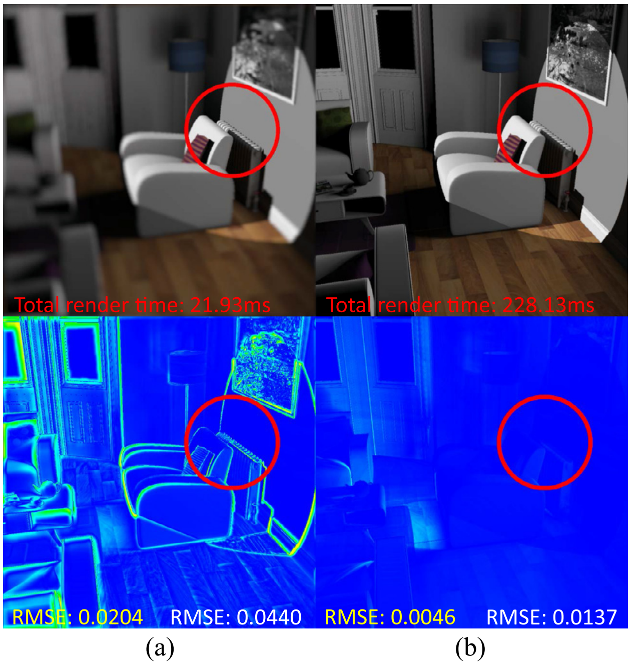

Fig. 9. Comparison of total render time and RMSEs in Living Room scene. (a) is our method and (b) is our method without any shading acceleration algorithms. The values at the bottom-left are RMSE in the foveal region, and the values at the bottom-right are the RMSE in the peripheral region.

Finally, we have invited 21 participants to watch the image sequences generated by our method with HMD. Since different people's visual perception is different, we first ask these participants to set some properties for themselves, such as foveal size and parameters of [30]. Then we use these properties to render image sequences by our method and TBDS in the four scenes. The image sequences contain different contents, for example C1, C2, C3 and C4 represent static scenes, scenes with dynamic light source, scenes with dynamic object and scenes with dynamic light source and object respectively in Fig. 8. At last, participants are asked to watch the image sequences of the same contents generated by different methods and to identify which method is used and where is the difference. For each scene, we have at least six trials. The percentage of success means that our method successfully confuses the judgment of the participants (Fig. 8, blue bar). In other cases, the participants correctly point out the differences expressed as the percentage of failure in our method (Fig. 8, orange bar).

For our study, participants are better able to distinguish the differences in static scenes, but the percentage of failure accounts for the majority in dynamic scenes. This is because the dynamic factors in the scenes will affect the perception of the human eyes, resulting in its insufficient sensitivity to details.

8. Conclusion and future work

This paper presents a method to generate single-bounce diffuse-to-diffuse indirect illumination for foveated rendering by efficiently culling the VPLs based on their contributions to the visual perception. We propose a visually importance-guide sampling strategy to precisely calculate the VPLs' contributions and an efficient estimation method to approximate the local visibility between the VPLs and pixel samples. Even though we can produce realistic global illumination in the foveal region, there are still some limitations to be addressed.

Global Visibility. Since our BLSO method is based on local space, it will cause false visibility in some cases. What's more, we can conduct a more precise method to approximate the visibility, which can be used both in contribution estimating and shading. So one of the most important goals in the future is to improve the accuracy of visibility evaluating in global space.

Multi-bounce indirect illumination and glossy materials. In this paper, we only study the single-bounce diffuse-to-diffuse indirect illumination. Though it is enough to generate a good global illumination in most cases, it will be dark at certain positions. Moreover, in the real life, there are many glossy material such as glass, mirror and metal products. So we will try to extend this method to support glossy materials in the future.

Efficiency. Although our method can reach the 45 FPS, it is till slower than the required frame rate of VR applications. Therefore, we have to further improve the performance of our method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Key Project of China (Project Number GJXM92579).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cag.2021.04.021.

References

[1] Guenter B, Finch M, Drucker S, Tan D, Snyder J. Foveated 3d graphics. ACM Trans Graph 2012;31(6):1–10.
[2] Cheng I. Foveated 3d model simplification. In: Seventh international symposium on signal processing and its applications, 2003. Proceedings., 1. IEEE; 2003. p. 241–4.
[3] Weier M, Stengel M, Roth T, Didyk P, Eisemann E, Eisemann M, et al. Perception-driven accelerated rendering. In: Computer graphics forum, 36. Wiley Online Library; 2017. p. 611–43.
[4] He Y, Gu Y, Fatahalian K. Extending the graphics pipeline with adaptive, multi-rate shading. ACM Trans Graph 2014;33(4):1–12.
[5] Wang L, Li R, Shi X, Yan L-Q, Li Z. Foveated instant radiosity. In: 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE; 2020. p. 1–11.
[6] Koskela M., Lotvonen A., Mäkitalo M., Kivi P., Viitanen T., Jääskeläinen P.. Foveated real-time path tracing in visual-polar space2019;.
[7] Keller A. Instant radiosity. Proceedings of the 24th annual conference on computer graphics and interactive techniques; 1997. 49–56
[8] Lindeberg T.. Concealing rendering simplifications using gazecontingent depth of field. 2016.
[9] Reddy M. Perceptually optimized 3d graphics. IEEE Comput Graph Appl 2001;21(5):68–75.
[10] Hu L, Sander PV, Hoppe H. Parallel view-dependent level-of-detail control. IEEE Trans Vis ComputGraph 2009;16(5):718–28.
[11] Tursun OT, Arabadzhiyska-Koleva E, Wernikowski M, Mantiuk R, Seidel H-P, Myszkowski K, et al. Luminance-contrast-aware foveated rendering. ACM Trans Graph 2019;38(4):1–14.
[12] Vaidyanathan K, Salvi M, Toth R, Foley T, Akenine-Möller T, Nilsson J, et al. Coarse pixel shading. In: Proceedings of High Performance Graphics; 2014. p. 9–18.
[13] Davidovič T, Křivánek J, Hašan M, Slusallek P, Bala K. Combining global and local virtual lights for detailed glossy illumination. ACM Trans Graph 2010;29(6):1–8.
[14] Segovia B, Iehl JC, Mitanchey R, Péroche B. Bidirectional instant radiosity.. In: Rendering techniques; 2006. p. 389–97.
[15] Simon F, Hanika J, Dachsbacher C. Rich-vpls for improving the versatility of many-light methods. In: Computer graphics forum, 34. Wiley Online Library; 2015. p. 575–84.
[16] Hedman P, Karras T, Lehtinen J. Sequential monte carlo instant radiosity. In: Proceedings of the 20th ACM SIGGRAPH symposium on interactive 3D graphics and games; 2016. p. 121–8.
[17] Weymouth FW. Visual sensory units and the minimum angle of resolution. Optom Vis Sci 1963;40(9):550–68.
[18] Longhurst P, Debattista K, Chalmers A. A GPU based saliency map for high-fidelity selective rendering. In: Proceedings of the 4th international conference on computer graphics, virtual reality, visualisation and interaction in Africa; 2006. p. 21–9.
[19] Stengel M, Grogorick S, Eisemann M, Magnor M. Adaptive image-space sampling for gaze-contingent real-time rendering. In: Computer graphics forum, 35. Wiley Online Library; 2016. p. 129–39.
[20] Harada T, McKee J, Yang JC. Forward+: Bringing deferred lighting to the next level.. In: Eurographics (Short Papers); 2012. p. 5–8.
[21] Harada T. A 2.5 d culling for forward+. In: SIGGRAPH asia 2012 technical briefs; 2012. p. 1–4.
[22] Tokuyoshi Y, Harada T. Stochastic light culling. J Comput Graph Tech 2016;5(1).
[23] O'Donnell Y, Chajdas MG. Tiled light trees. In: Proceedings of the 21st ACM SIGGRAPH symposium on interactive 3D graphics and games; 2017. p. 1–7.
[24] Strasburger H, Rentschler I, Jüttner M. Peripheral vision and pattern recognition: a review. JVis 2011;11(5):13.
[25] Vangorp P, Laurijssen J, Dutré P. The influence of shape on the perception of material reflectance. In: ACM SIGGRAPH 2007 papers; 2007. p. 77–es.
[26] Marlow PJ, Anderson BL. Material properties derived from three-dimensional shape representations. VisRes 2015;115:199–208.
[27] Hastings W.K.. Monte carlo sampling methods using Markov chains and their applications1970;.
[28] Szirmay-Kalos L., Dornbach P., Purgathofer W.. On the start-up bias problem of metropolis sampling1999;.
[29] Ritschel T, Eisemann E, Ha I, Kim JD, Seidel H-P. Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. In: Computer graphics forum, 30. Wiley Online Library; 2011. 2258–2269
[30] Meng X, Du R, Zwicker M, Varshney A. Kernel foveated rendering. Proc ACM Comput GraphInteract Tech 2018;1(1):1–20.
[31] Keller A, Heidrich W. Interleaved sampling. In: Eurographics workshop on rendering techniques. Springer; 2001. p. 269–76.
[32] Andersson J. Parallel graphics in frostbite-current & future. SIGGRAPH Course 2009.