

Delta Path Tracing for Real-Time Global Illumination in Mixed Reality

Yang Xu*

Yuanfa Jiang[†]

Shibo Wang[‡]

Kang Li[§]

Guohua Geng[¶]

Northwest University, China



Figure 1: Rendering results of our proposed delta path tracing in a mixed reality environment under two lighting conditions (128 spp). The Happy Buddha (1,087 K triangles), the Stanford Bunny (144 K triangles), and the Chinese Dragon (871 K triangles) are three virtual objects with different BRDFs inserted into the real scene. The rendering times of the two images are 34.03 ms (left) and 34.16 ms (right), respectively.

ABSTRACT

Visual coherence between real and virtual objects is important in mixed reality (MR), and illumination consistency is one of the key aspects to achieve coherence. Apart from matching the illumination of the virtual objects with the real environments, the change of illumination on the real scenes produced by the inserted virtual objects should also be considered but is difficult to compute in real-time due to the heavy computation demands of global illumination. In this work, we propose delta path tracing (DPT), which only computes the radiance blocked by the virtual objects from the light sources at the primary hit points of Monte Carlo path tracing, then combines the blocked radiance and multi-bounce indirect illumination with the image of the real scene. Multiple importance sampling (MIS) between BRDF and environment map is performed to handle all-frequency environment maps captured by a panorama camera. Compared to conventional differential rendering methods, our method can remarkably reduce the number of times required to access the environment map and avoid rendering scenes twice. Therefore, the performance can be significantly improved. We implement our method using hardware-accelerated ray tracing on modern GPUs, and the results demonstrate that our method can render global illumination at real-time frame rates and produce plausible visual coherence between real and virtual objects in MR environments.

Index Terms: Computing methodologies—Computer graphics—Graphics systems and interfaces—Mixed / augmented reality; Computing methodologies—Computer graphics—Rendering—Ray tracing

1 INTRODUCTION

Consistent illumination between real and virtual objects in mixed reality (MR) or augmented reality (AR) environments is crucial to seamlessly integrating virtual objects into real scenes [1, 21, 27]. Taking the radiance change introduced by the inserted virtual objects into account is a key to enhance the visual coherence. However, due to the heavy computational cost of high-quality global illumination solutions, it is difficult to achieve photorealistic MR rendering at real-time frame rates.

Adjusting the illumination caused by the synthetic objects inserted into real scenes is widely used in the film industry, but mostly relies on manual interactions by the artists. Differential rendering [6] is the most commonly used method to integrate virtual objects into real scenes. The basic differential rendering is a two-pass method which requires computing two global illumination solutions: one with only the real scene, and another containing both real and virtual objects. The difference between the two solutions, which represents the radiance change introduced by the virtual objects, is added to the image of the real scene to acquire the composited result.

Our method is also based on the basic concept of differential rendering, but inspired by differential photon mapping [15] and differential instant radiosity [25, 26] which directly compute the radiance change introduced by the virtual objects, and the methods based on delta radiance fields [9–13, 41]. We adopt the delta radiance computation in the path tracing framework to propose delta path tracing. Therefore, a full global illumination solution with multi-bounce indirect illumination can be obtained and arbitrary BRDFs can be supported.

Compared to standard path tracing, we directly compute the radiance blocked by the virtual objects when rays originated from the camera hit the scene geometry for the first time rather than evaluating the contribution from the light source. Therefore, the radiance change introduced by the virtual objects can be directly obtained to avoid two-pass differential rendering. We implement our DPT using hardware-accelerated ray tracing on modern GPUs to achieve photorealistic MR rendering at real-time frame rates. Our DPT is simple to implement and can support various illumination phenomena without precomputation, such as high-frequency shadows from an all-frequency HDR environment map, specular/glossy

*e-mail: xuyang@nwu.edu.cn

[†]e-mail: hermesjang@foxmail.com

[‡]e-mail: 840580592@qq.com

[§]e-mail: likang@nwu.edu.cn

[¶]e-mail: ghgeng@nwu.edu.cn

reflections, and diffuse/glossy interreflections. Additionally, our method can be extended to handle dynamic viewpoints, dynamic or deformable objects, and dynamic spatially-varying BRDFs without special tricks.

The contributions of our work are summarized as follows

- A modified path tracing algorithm called delta path tracing, which is suitable for photorealistic rendering in MR.
- A DPT-based MR rendering method combined with MIS implemented by hardware-accelerated ray tracing on modern GPUs.
- An MR framework based on DPT that can seamlessly integrate virtual objects into real scenes at real-time frame rates.

2 RELATED WORK

Since most global illumination methods in MR are based on the differential rendering technique, we first revisit the development of differential rendering and the methods based on differential rendering, and then discuss several works that do not use differential rendering.

Differential Rendering A pioneering work in differential rendering was done by Fournier et al. [8]. They proposed a two-pass method to insert computer generated objects into real scenes. To simulate the effect of the change caused by the computer generated objects, the real scene is modulated by the ratio between two global illumination solutions by progressive radiosity with and without computer generated objects. However, geometries, BRDFs, and light sources of the entire scene are required to be modeled. Drettakis et al. [7] introduced hierarchical radiosity to allow fast update of illumination for moving computer generated objects. Sato et al. [39] used two omni-directional cameras with fish-eye lenses to capture two omni-directional images from different locations for constructing a geometric model of the scene using the omni-directional stereo algorithm. They multiplied the ratio between two irradiances computed by ray casting with and without virtual objects to the image of the real scene to composite virtual objects into the real scene. Debevec [6] computed two global illumination solutions by incremental radiosity with and without the synthetic objects, then the difference of the two solutions is computed and added to the image of the real scene to acquire the composited result. Only the local scene near the synthetic objects is modeled, and the distant scene is represented by an HDR environment map to illuminate the scene using image-based lighting (IBL).

Differential rendering can reduce the errors introduced by inaccurate BRDF estimation of the real scene because only the radiance change due to the presence of the synthetic objects is affected by estimated BRDF. Due to the advantage of differential rendering, many other rendering methods for MR based on differential rendering were proposed. Karsch et al. [24] used differential rendering to insert synthetic objects into existing photographs. Zhang et al. [44] proposed a photon mapping-based differential rendering method to support specular and transparent objects. Zhang et al. [45] further considered the influence of different refractive indices and roughnesses of virtual objects. The aforementioned methods can produce photorealistic rendering results but cannot run at interactive or real-time frame rates since offline renderers are utilized, but interactivity is required by most MR applications. Gruber et al. [18] estimated a low-frequency SH presentation of the environment light by solving a linear system constructed by the radiance transfer function in the obtained scene geometry, and a precomputed radiance transfer (PRT) solution is computed by ray casting on the real and virtual objects together for differential rendering to insert virtual objects into real scenes. They further accelerated PRT computation by adaptive sampling in image and visibility space for exploiting spatial

coherence, and radiance transfer caching for exploiting temporal coherence [17, 19]. They also combined differential rendering with a variant of screen-space directional occlusion (SSDO) to approximate environment lighting with visibility and first bounce indirect lighting in real-time [20]. Mehta et al. [33] proposed a two-mode path tracing that performs mesh-based ray tracing for virtual objects and screen-space ray tracing for real scenes reconstructed by an RGB-D camera to achieve differential rendering, and axis-aligned filtering is utilized to denoise the results. Rohmer et al. [35, 36] introduced a differential rendering method on mobile devices using virtual area lights (VAL) with shadow maps for high-frequency illumination and PRT for low-frequency illumination, and the computation is shared between a PC and the participating mobile devices to make use of the capacities on both sides. Rohmer et al. [37] also introduced differential rendering methods based on several ray tracing variants, including environment map sampling, distance impostor tracing, and voxel cone tracing, which can offer different levels of quality.

However, differential rendering is a two-pass method that requires rendering the scenes twice, which reduces the rendering efficiency. Grosch [15] proposed differential photon mapping, which directly simulates the illumination change introduced by the virtual objects using a differential photon map to avoid performing light transport simulation twice, and can simulate most of the global illumination phenomena such as caustics. But this method is still an offline method and cannot support interactive or real-time MR systems. Knecht et al. [25] proposed differential instant radiosity, which is also a single-pass differential rendering method. They directly compute the radiance blocked by the virtual objects and can achieve real-time frame rates. They utilized imperfect shadow maps (ISM) [34] to accelerate the rendering shadows of the virtual point lights (VPL). They also extended their method by modeling the real scene at runtime using an RGB-D camera [26]. Kán and Kaufmann [29] introduced a one-pass differential rendering method, which computes and composites the radiance coming from real and virtual objects together in a single ray tracing pass by shooting two types of rays instead of rendering two separate results. They also proposed ray tracing and photon mapping-based one-pass differential rendering to render high-quality specular global illumination effects at real-time frame rates in MR environments, including reflections, refractions, and caustics [28]. They also extended one-pass differential rendering to differential progressive path tracing [23] and differential irradiance caching [30] to render global illumination in MR, which can run at real-time frame rates.

Delta Radiance Fields Franke [10, 13] proposed delta radiance fields to represent the radiance change caused by the introduction of virtual objects into a real scene. He introduced delta light propagation volumes (DLPV) [9] and delta voxel cone tracing (DVCT) [11, 12] based on the theory of delta radiance fields. Real-time frame rates can be achieved by using these methods. Thöner and Kuijper [41] proposed a delta global illumination method similar to DVCT.

Global Illumination in MR Lensing and Broll [31] utilized multi-resolution splatting indirect illumination based on reflective shadow maps (RSM) to simulate one bounce indirect illumination in MR environments with an RGB-D camera for scene reconstruction. Grosch et al. [16] precomputed an irradiance volume that stores spherical harmonics coefficients at grid points inside a box to simulate indirect illumination to the virtual objects in the box at interactive frame rates, and proposed a sampling strategy for the exclude window to compute the direct illumination from an HDR image captured by an HDR video camera outside the box.

Real-Time Ray Tracing in MR Santos et al. [38] presented a real-time ray tracing pipeline to produce high-quality interactions between real and virtual objects such as occlusions, reflections, refractions, and soft shadows. Dai et al. [5] proposed an efficient ray

tracing method to render glossy and specular reflection on virtual objects based on the surrounding real scenes at interactive frame rates.

3 DELTA PATH TRACING

Since our method is built upon differential rendering [6] and delta radiance fields [13], we review the principles of these two techniques first to derive our method. In differential rendering, two light transport simulations are required to be performed. The geometries, BRDFs, and light sources of the real scene are modeled, and a light transport simulation is performed on the real scene to acquire radiance L_r (real radiance), the virtual objects are then inserted into the real scene, and a second light transport simulation is computed to obtain another radiance L_m (mixed radiance). The composited result is obtained by adding the difference between the two simulations $\Delta L = L_m - L_r$ which represents the radiance change introduced by the inserted virtual objects to the image of the real scene. It should be noted that ΔL can be either positive or negative. In summary, the composited result L_f can be expressed as

$$\begin{aligned} L_f &= M \odot L_m + (1 - M) \odot (L_b + \Delta L) \\ &= M \odot L_m + (1 - M) \odot (L_b + L_m - L_r), \end{aligned} \quad (1)$$

where L_f is the radiance of the final composited image, L_b is the background radiance from the image of the real scene, \odot denotes the element-wise product, and M denotes the mask to determine whether a pixel belongs to virtual objects or not. From Equation 1 we can see that ΔL is only applied on the pixels of the real scene. The major drawback of differential rendering is that when the virtual objects do not occupy the field of view, a large area of the image remains unchanged. Therefore, simulating light transport twice in these areas is wasteful.

In delta radiance fields, the existing radiance field is denoted as L_μ . A virtual object is inserted into the scene, which changes the radiance field by blocking or scattering the light. Another radiance field of the same scene which contains the virtual object is denoted as L_ρ . Therefore, the change in the radiance field L_μ introduced by the virtual object can be defined as a delta radiance field L_Δ as

$$L_\Delta = L_\rho - L_\mu. \quad (2)$$

Delta radiance fields rely on the delta transfer operator which can transfer lighting conditions into delta radiance fields to exactly extract the illumination change introduced by the inserted virtual objects. Franke [13] claimed that the delta transfer operator can be integrated into the path tracing framework with antiradiance paths. In this work, we investigate the light transport paths of path tracing in MR environments and try to precisely extract the light paths that contain the illumination change caused by the inserted virtual objects and ignore the other light paths, which mimics the delta transfer operator.

3.1 Delta Radiance Computation

We define the delta radiance as the change of the outgoing radiance introduced by the inserted virtual objects, which contains the radiance blocked by the virtual objects inserted into the real scenes and the indirect illumination introduced by the virtual objects. The meaning of delta radiance is equivalent to the difference between two global illumination solutions ΔL in differential rendering and the delta radiance field L_Δ .

The theoretical foundation of Monte Carlo path tracing is the rendering equation [22]

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega^+} L_i(\mathbf{x}, \omega_i) f_r(\mathbf{x}, \omega_i, \omega_o) (\mathbf{n} \cdot \omega_i) d\omega_i, \quad (3)$$

where $L_o(\mathbf{x}, \omega_o)$ is the outgoing radiance at surface point \mathbf{x} from the view direction ω_o , $L_e(\mathbf{x}, \omega_o)$ is the emissive radiance of the

surface point \mathbf{x} , Ω^+ is the upper hemisphere of the surface point \mathbf{x} , $f_r(\mathbf{x}, \omega_i, \omega_o)$ is the BRDF from the direction ω_i to the direction ω_o , $L_i(\mathbf{x}, \omega_i)$ is the incoming radiance from the direction ω_i , and \mathbf{n} is the normal of the surface point \mathbf{x} .

To directly compute the delta radiance instead of using two-pass differential rendering, we assume a simple local scene including just a plane and modify the standard path tracing algorithm to propose delta path tracing. Our DPT differs from standard path tracing at the primary hit points, where the primary rays originating from the camera hit the scene geometry for the first time. Since we only modify the operations at the primary hit points in path tracing, we investigate the direct illumination from the light source

$$L_o(\mathbf{x}, \omega_o) = \int_{\Omega^+} L_s(\mathbf{x}, \omega_i) f_r(\mathbf{x}, \omega_i, \omega_o) V(\mathbf{x}, \omega_i) (\mathbf{n} \cdot \omega_i) d\omega_i, \quad (4)$$

where $L_s(\mathbf{x}, \omega_i)$ is the incoming radiance from the light source in the direction ω_i , and $V(\mathbf{x}, \omega_i)$ is the visibility function between the surface point \mathbf{x} and the light source

$$V(\mathbf{x}, \omega_i) = \begin{cases} 1, & \text{if visible to light source in direction } \omega_i, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

If we define another occlusion function $O(\mathbf{x}, \omega_i)$ as

$$O(\mathbf{x}, \omega_i) = \begin{cases} 1, & \text{if not visible to light source in direction } \omega_i, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

then we can rewrite Equation 4 as the following form

$$\begin{aligned} L_o(\mathbf{x}, \omega_o) &= \int_{\Omega^+} L_s(\mathbf{x}, \omega_i) f_r(\mathbf{x}, \omega_i, \omega_o) [1 - O(\mathbf{x}, \omega_i)] (\mathbf{n} \cdot \omega_i) d\omega_i \\ &= \int_{\Omega^+} L_s(\mathbf{x}, \omega_i) f_r(\mathbf{x}, \omega_i, \omega_o) (\mathbf{n} \cdot \omega_i) d\omega_i \\ &\quad - \int_{\Omega^+} L_s(\mathbf{x}, \omega_i) f_r(\mathbf{x}, \omega_i, \omega_o) O(\mathbf{x}, \omega_i) (\mathbf{n} \cdot \omega_i) d\omega_i \\ &= L_o^{\text{unblocked}}(\mathbf{x}, \omega_o) - L_o^{\text{blocked}}(\mathbf{x}, \omega_o), \end{aligned} \quad (7)$$

where $L_o^{\text{unblocked}}(\mathbf{x}, \omega_o)$ is the unblocked radiance without inserted virtual objects, which can be regarded as the background radiance from the image of the real scene in our method, and $L_o^{\text{blocked}}(\mathbf{x}, \omega_o)$ is the radiance blocked by the virtual objects, which is required to be computed at the primary hit point in DPT.

In standard path tracing, if a ray originating from the primary hit point does not hit the scene geometry or next event estimation is applied to directly sample the light sources, direct illumination is evaluated by collecting the radiance from the directions visible to the hit point. However, at the primary hit point in our proposed DPT, we gather the radiance from the directions blocked by the virtual objects instead. Therefore, we can directly obtain the radiance blocked by the virtual objects inserted into the real scenes, which is an important component of the delta radiance. The other fraction of delta radiance is the indirect illumination, which can be computed by the following bounces of path tracing. As a result, the delta radiance can be directly acquired using our DPT.

Since only the delta radiance is required to be computed for the real scenes, the computational cost can be reduced because each ray only needs to query the direct illumination once. Additionally, when the virtual objects only cover a small part of the upper hemisphere, the number of direct illumination queries can be decreased compared to gathering the radiance from the directions visible to the surface point.

It needs to be also noted that standard path tracing is still applied to the primary hit points belonging to virtual objects because background radiance from the image of the real scene is not available

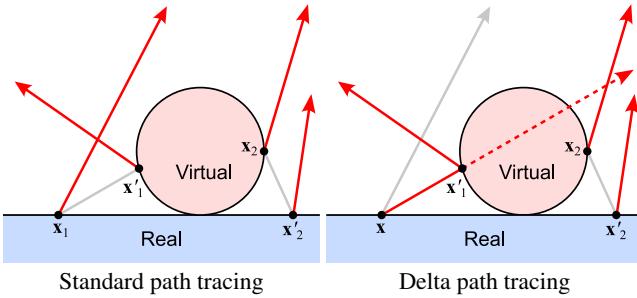


Figure 2: Difference between standard PT and our DPT at the primary hit points. Left: standard PT. Right: DPT. The primary rays are omitted for clarity. The red rays need to query direct illumination, and the gray rays do not need to query.

for virtual objects. Hence, the final composited result cannot be obtained with the delta radiance alone.

In summary, we illustrate the difference between standard PT and our DPT in Fig. 2. x_1 and x_2 are two primary hit points on the real and virtual object, respectively. The primary rays are omitted for better clarity. We trace rays from x_1 and x_2 . If a ray hit the scene geometry at x'_1 or x'_2 , a secondary ray is generated and traced to account for indirect illumination such as interreflections. Fig. 2 exhibits four different types of light paths: real (x_1) → environment, real (x_1) → virtual (x'_1) → environment, virtual (x_2) → environment, and virtual (x_2) → real (x'_2) → environment. The red rays need to query direct illumination, and the gray rays do not need to query direct illumination. We can see that our DPT differs from standard PT only when tracing the rays originating from the primary hit points on the real objects. Once the delta radiance on the surface points of the real scene and the outgoing radiance on the surface points of the virtual objects are both obtained, we can still use Equation 1 to compute the composited result, where ΔL is the delta radiance computed by our DPT.

3.2 Multiple Importance Sampling

We use HDR environment maps as the light sources in the experiments. To improve the sampling efficiency under all-frequency HDR environment maps, we combine BRDF importance sampling with importance sampling of the environment maps by using MIS [42] between BRDF and environment map. To importance sample the HDR environment map, a cumulative distribution function (CDF) table of the HDR environment map can be precomputed. At runtime, environment map samples are generated at each shaded point by finding the inverse CDF using 2D low-discrepancy samples. However, searching the inverse CDF in a CDF table is time-consuming. Although the alias table can be utilized to avoid performing expensive searching within a CDF table, it breaks the stratification of the sample points, which leads to higher variance [3]. Instead, we apply the precomputed importance sampling strategy proposed by Mehta et al. [33], precomputed environment samples with the same number as the total number of pixels in the environment map are stored as a texture with the same size as the original environment map. At runtime, 2D low-discrepancy samples are created at each shaded point to look up the precomputed texture to obtain sample points on the environment map.

In MIS, we use the balance heuristic to combine two different sampling strategies. As shown in Fig. 3, we compare the rendering results without denoising and the timings using BRDF importance sampling, environment map importance sampling, and MIS between BRDF and environment map with an equal sample count (128 spp). We can see that environment map importance sampling can significantly reduce the variance in the shadow region compared to

Algorithm 1: Delta path tracing

Input: virtual object mask M , HDR environment map
radiance L_s , surface point \mathbf{x} , and view direction ω_0
Output: outgoing radiance $L_o(\mathbf{x}, \omega_0)$

- 1 $L_o(\mathbf{x}, \omega_0) \leftarrow 0$;
- 2 $throughput \leftarrow 1$;
- 3 **for** $bounces \leftarrow 1$ **to** $maxBounces$ **do**
- // Next event estimation
- sample environment map in direction ω_{env} with pdf
 $p_{env}(\mathbf{x}, \omega_{env}, \omega_0)$;
- trace a ray $\mathbf{r}(\mathbf{x}, \omega_{env})$;
- $w_{env}(\mathbf{x}, \omega_{env}, \omega_0) = \frac{p_{env}(\mathbf{x}, \omega_{env}, \omega_0)}{p_{env}(\mathbf{x}, \omega_{env}, \omega_0) + p_{brdf}(\mathbf{x}, \omega_{env}, \omega_0)}$;
- $\Delta L_o(\mathbf{x}, \omega_0) =$
 $throughput \cdot \frac{w_{env}(\mathbf{x}, \omega_{env}, \omega_0) f_r(\mathbf{x}, \omega_{env}, \omega_0) (\mathbf{n} \cdot \omega_{env}) L_s(\mathbf{x}, \omega_{env})}{p_{env}(\mathbf{x}, \omega_{env}, \omega_0)}$;
- if** $bounces = 1$ **and** $M(\mathbf{x}) = 0$ **then**
- if** $\mathbf{r}(\mathbf{x}, \omega_{env})$ **hits the scene** **then**
- $| L_o(\mathbf{x}, \omega_0) \leftarrow L_o(\mathbf{x}, \omega_0) - \Delta L_o(\mathbf{x}, \omega_0)$;
- end**
- else**
- if** $\mathbf{r}(\mathbf{x}, \omega_{env})$ **does not hit the scene** **then**
- $| L_o(\mathbf{x}, \omega_0) \leftarrow L_o(\mathbf{x}, \omega_0) + \Delta L_o(\mathbf{x}, \omega_0)$;
- end**
- end**
- sample BRDF in direction ω_i with pdf $p_{brdf}(\mathbf{x}, \omega_i, \omega_0)$;
- trace a ray $\mathbf{r}(\mathbf{x}, \omega_i)$;
- $w_{brdf}(\mathbf{x}, \omega_i, \omega_0) = \frac{p_{brdf}(\mathbf{x}, \omega_i, \omega_0)}{p_{brdf}(\mathbf{x}, \omega_i, \omega_0) + p_{env}(\mathbf{x}, \omega_i, \omega_0)}$;
- $\Delta L_o(\mathbf{x}, \omega_0) =$
 $throughput \cdot \frac{w_{brdf}(\mathbf{x}, \omega_i, \omega_0) f_r(\mathbf{x}, \omega_i, \omega_0) (\mathbf{n} \cdot \omega_i) L_s(\mathbf{x}, \omega_i)}{p_{brdf}(\mathbf{x}, \omega_i, \omega_0)}$;
- if** $\mathbf{r}(\mathbf{x}, \omega_i)$ **hits the scene at** \mathbf{x}' **then**
- if** $bounces = 1$ **and** $M(\mathbf{x}) = 0$ **then**
- $| L_o(\mathbf{x}, \omega_0) \leftarrow L_o(\mathbf{x}, \omega_0) - \Delta L_o(\mathbf{x}, \omega_0)$;
- end**
- $\mathbf{x} \leftarrow \mathbf{x}'$;
- $\omega_0 \leftarrow -\omega_i$;
- $throughput \leftarrow throughput \cdot \frac{f_r(\mathbf{x}, \omega_i, \omega_0) (\mathbf{n} \cdot \omega_i)}{p_{brdf}(\mathbf{x}, \omega_i, \omega_0)}$;
- else**
- $L_o(\mathbf{x}, \omega_0) \leftarrow L_o(\mathbf{x}, \omega_0) + \Delta L_o(\mathbf{x}, \omega_0)$;
- break**;
- end**
- 32 **end**

BRDF importance sampling at the cost of spending more time. And when environment map importance sampling is applied, the glossy indirect illumination from the surfaces with glossy BRDFs and the glossy reflections from the environment map are noisier than those of BRDF importance sampling. While MIS between BRDF and environment map combines the advantages of both sampling strategies. MIS can achieve similar results to environment map importance sampling in the shadow region but spends much less time, and the noises of glossy indirect illumination and glossy reflection in MIS are comparable to those in BRDF importance.

The pseudo-code of our DPT along with MIS between BRDF and environment map is exhibited in Algorithm 1. The virtual object mask M is a 2D texture with the same resolution as the framebuffer. The pixel values of M are defined by Equation 8. The value of a pixel in M is 1 when the surface point \mathbf{x} corresponding to the pixel belongs to virtual objects, and the value is 0 when \mathbf{x} belongs to real objects. Additionally, the scene to be hit by the rays in Algorithm 1 contains both real and virtual objects.

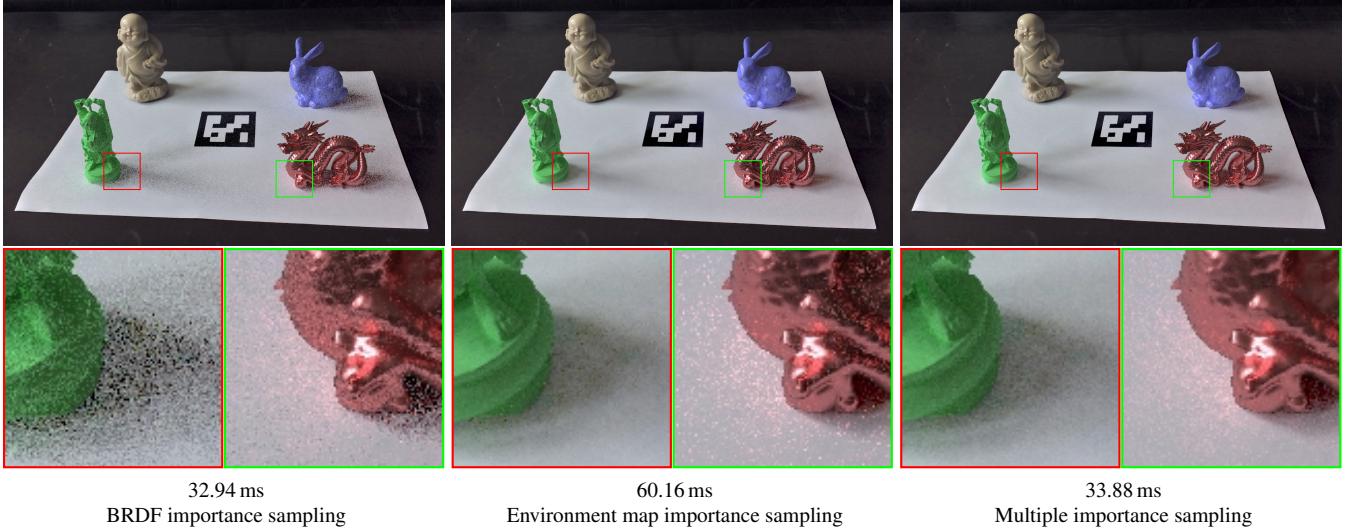


Figure 3: Comparison of different sampling strategies with an equal sample count (128 spp). Left: BRDF importance sampling. Middle: environment map importance sampling. Right: MIS between BRDF and environment map.

$$M(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ belongs to virtual object,} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

It should also be noted that the primary rays are not required to be traced in our implementation of DPT because we use ray query to generate and trace rays in the fragment shader. Thus, the primary hit points can be obtained by rasterization because the shaded points can be directly treated as the primary hit points. When *bounces* in Algorithm 1 is 1, we trace rays originating from the surface point, which is also the primary hit point.

4 RESULTS AND DISCUSSION

We establish an MR environment to evaluate our method. We use a ROG Eye camera to capture the videos of the real scenes, and the intrinsic parameters of the camera, including focal length, principal point, and distortion coefficients, are calibrated for pose estimation. An A4 printer paper with an ArUco marker [14] printed at the center of the paper is placed on the table in the real scene, and a buddha status as the real object is placed on the paper. The ArUco marker is detected in the captured video, and the pose of the marker relative to the camera is estimated by solving the perspective-n-point (PnP) problem to place the virtual objects on the paper. We use an Insta360 ONE X2 panorama camera with two fish-eye lenses to capture the HDR environment maps. Images with 9 different exposure levels are acquired to reconstruct an HDR image. The original resolution of the captured HDR environment map is 6080×3040 , and we down-sample it to 2400×1200 and store it as a 2D texture with a format of 16-bit floating point per channel. We implement our proposed DPT using Vulkan with the ray query extension, which allows performing hardware-accelerated ray tracing in the fragment shader. All results are rendered on a computer equipped with Intel i9-10980XE CPU, 128 GB RAM, and NVIDIA GeForce RTX 3080 GPU, and the resolution of all rendered images is 1280×720 . We measure the frame time to evaluate the performance, and all timings are averaged over 100 frames. The Happy Buddha model with diffuse BRDF (1,087 K triangles), the Stanford Bunny model with dielectric BRDF (144 K triangles), and the Chinese Dragon model with metallic BRDF (871 K triangles) from the McGuire Computer Graphics Archive [32] are used as the virtual objects in our experiments. We



Figure 4: Tone-mapped HDR environment maps of two lighting conditions captured by the panorama camera. Left: lighting condition I. Right: lighting condition II.

use the Lambertian model in the diffuse BRDF and the diffuse component of the dielectric BRDF, and the GGX microfacet model [43] is utilized in the metallic BRDF and the glossy component of the dielectric BRDF. Additionally, a diffuse plane with the same size as the paper and a larger diffuse plane with the same diffuse reflectance as the table are modeled as the local scene in the virtual scene to receive shadows and indirect illumination from the virtual objects, and bounce light to the virtual objects. The diffuse reflectances of the paper and the table in the virtual scene are manually adjusted to match the rendering results to the images from the video of the real scene. Hence, the strength of the shadows and indirect illumination can match those of the real objects.

We test our method in two different lighting conditions I and II. Lighting condition I is in the afternoon, the room is lit by the sky-light and the light bounced by the exterior wall of the building from two large windows. Therefore, two directional soft shadows are cast from the directions of the two windows, respectively. Lighting condition II is in the evening, the room is lit by 6 groups of fluorescent lamps at the ceiling. Since the area of the fluorescent lamp is small, multiple sharp shadows are cast from different directions. The HDR environment maps of the two lighting conditions captured by the panorama camera are displayed in Fig. 4. It should be noted that the displayed HDR environment maps are tone-mapped to show the brightest and darkest details on low dynamic range (LDR) displays, and the HDR environment maps for rendering are not tone-mapped.

In MIS, 64 samples for HDR environment map importance sampling and 64 samples for BRDF importance sampling are applied for

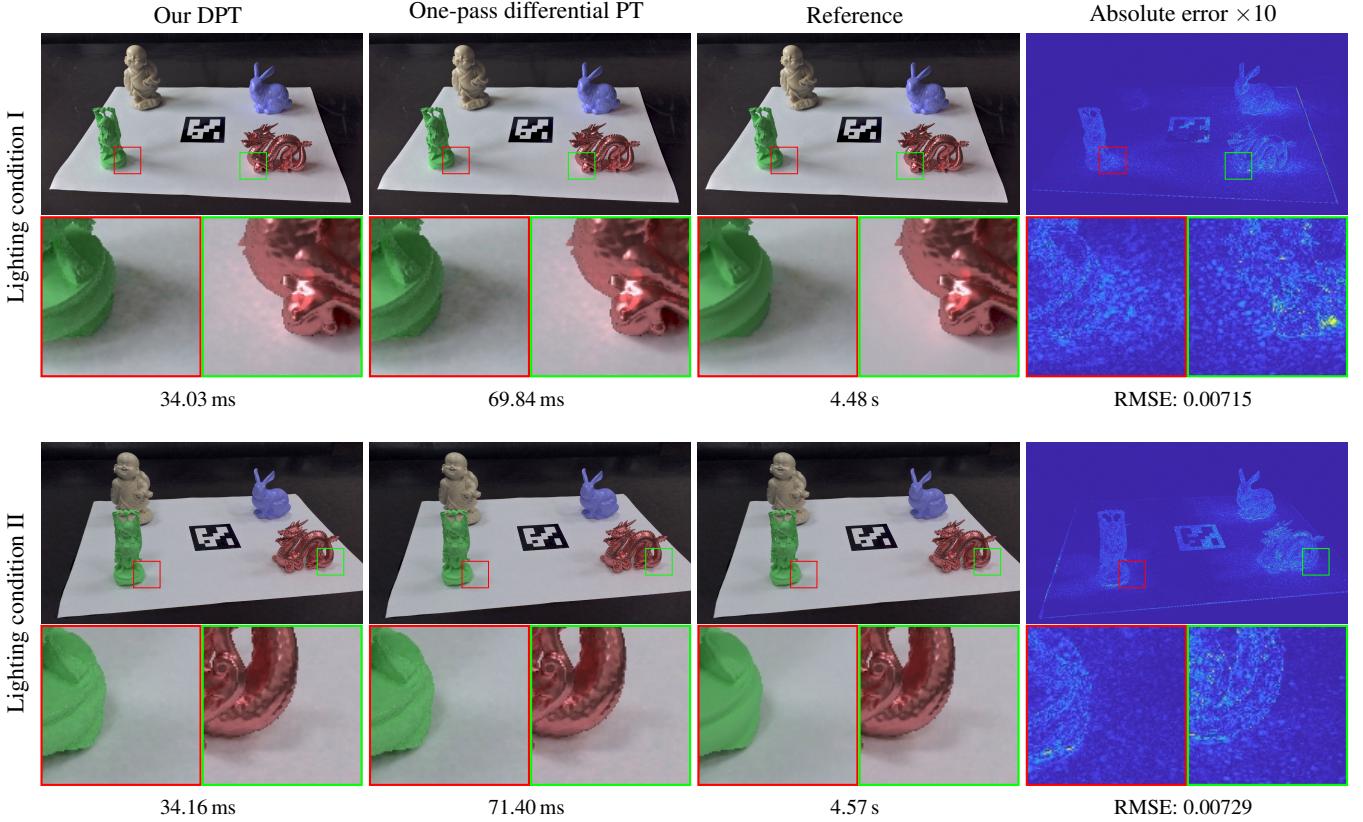


Figure 5: Comparison of different methods under two lighting conditions. From left to right: our proposed DPT (128 spp), one-pass differential PT [23, 29] (128 spp), 8192-spp differential PT as the reference, and the error between DPT and the reference. Upper row: lighting condition I. Lower row: lighting condition II.

each shaded point. Low-discrepancy samples are generated from the 2D Sobol sequence and randomly shifted by the Cranley-Patterson rotation [4] to eliminate correlation artifacts. DPT is configured to have a maximum of 3 bounces, and a cross-bilateral filter with a size of 5 is performed on the radiance to suppress Monte Carlo noise.

We compare our method to one-pass differential path tracing proposed by Kán and Kaufmann [23, 29], and the rendering result of differential path tracing with 8,192 spp is regarded as the reference. We use the absolute error and the root mean squared error (RMSE) to evaluate the difference between our method and the reference. The rendering results and the timings are displayed in Fig. 5. From the rendering results, we can see that our DPT can seamlessly integrate the virtual objects into real scenes and produce results comparable to the reference. The shadows cast by the virtual objects match perfectly with the shadows cast by the real objects and diffuse/glossy interreflections on the paper can be observed around the virtual objects. Besides, we can see the glossy reflections of the environment maps and the white paper on the virtual objects with glossy BRDFs (Bunny and Dragon). Meanwhile, our method is much faster than one-pass differential path tracing without sacrificing any rendering quality.

As shown in Table 1 and Fig. 6, we display and plot the timings of one-pass differential PT and our DPT under two lighting conditions of different numbers of bounces. We also compare the rendering results of different numbers of bounces under two lighting conditions as shown in Fig. 7 and Fig. 8. The rendering results lack realism when the number of bounces is 1 because only direct illumination

exists. Additionally, a larger number of bounces costs more time to render, and a plausible result can be obtained with only 3 bounces. Therefore, we use 3 bounces in the previous results.

5 LIMITATIONS

In this work, we use a plane as the real scene, and we cannot handle arbitrary real scenes because light transport paths are not fully considered. For example, we do not take the indirect illumination blocked by the virtual objects from the real objects into account, and the indirect illumination from a real object to another real object is also simulated in our method but is already contained in the image of the real scene, which leads to an overestimation of the indirect illumination. We will extend our DPT to handle arbitrary real scenes by considering more complex light transport paths.

Another limitation of our method is that we cannot properly handle the reflections of virtual objects on real scenes with non-diffuse BRDFs. To illustrate the problem when real scenes are non-diffuse, we change the material of the table in the virtual scene from the diffuse BRDF to a non-diffuse BRDF. As shown in Fig. 9 left, the reflections of the virtual objects on the table work well when the radiance of the HDR environment map is near-uniform. However, if there are several high-energy regions in the environment map such as the sky radiance through the windows, the specular occlusion of Dragon and Bunny computed by our DPT cannot correctly match the captured reflections on the table (Fig. 9 right). This mismatch is because we assume the HDR environment map to be a distant light source but actually it is not at infinity. This issue can be addressed if

Table 1: Timings in ms of different numbers of bounces under two lighting conditions

Bounces	One-Pass differential PT		Our DPT	
	Lighting condition I	Lighting condition II	Lighting condition I	Lighting condition II
1	59.95	60.63	22.57	22.41
2	67.09	67.96	29.76	29.86
3	69.84	70.43	34.03	34.16
4	71.21	71.48	36.59	36.69
5	72.36	72.43	38.54	38.83
6	73.37	73.25	40.07	40.54
7	74.11	74.10	41.56	42.35
8	74.83	74.87	42.80	43.64
9	75.68	75.86	44.09	44.69
10	76.51	76.51	45.15	45.82

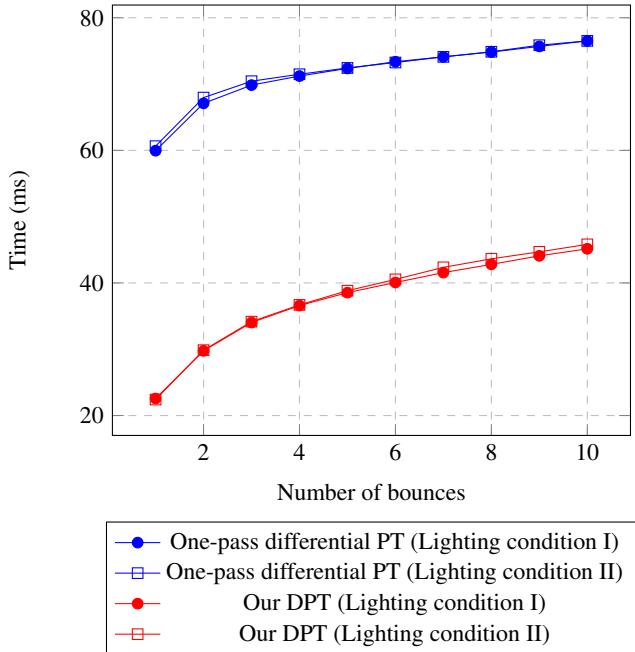


Figure 6: Timings of different numbers of bounces under two lighting conditions

we model the whole scene geometry including the windows. Besides, BRDF estimation of non-diffuse materials is much more difficult than that of diffuse materials, which is also challenging when the real scenes are non-diffuse.

6 CONCLUSION AND FUTURE WORK

In this work, we present a photorealistic global illumination rendering method for MR named delta path tracing. Virtual objects can be seamlessly integrated into real scenes at real-time frame rates. At the first hit point of path tracing, we gather the radiance blocked by the virtual objects instead of collecting the radiance received by the surface point. Compared to the existing two-pass differential rendering methods or one-pass differential PT, our method can significantly improve performance by reducing the number of times to look up the HDR environment maps.

Future work includes four aspects. The first aspect is rendering. Temporal Monte Carlo denoisers can be applied to our method to

allow a lower sample count and improve the frame rate. Additionally, advanced sampling strategies that consider the product of the BRDF and the environment map can be applied to improve the sampling efficiency, such as resampled importance sampling (RIS) [40] and its spatial-temporal variant ReSTIR [2]. We also plan to support transparent and translucent objects in our framework in the future.

The second aspect is light estimation. In this work, we use a pre-captured static HDR environment map, which cannot support dynamic lighting changes. A lot of work about capturing live HDR environment map video has been proposed, and we can utilize it. Additionally, estimating the environment light entirely from an image of the real scene is also an interesting topic.

The third aspect is the scene modeling. In this work, we just manually modeled a simple local scene and placed the virtual objects into the real scene by using a marker. We will consider live reconstruction of the real scene by using an RGB-D camera as many previous works do [17, 18, 20, 26, 31, 33] and insert the virtual objects according to the scanned geometry.

The last aspect is that our method can be integrated into other devices, such as video see-through head-mounted displays (VST-HMDs) and mobile devices supporting hardware-accelerated ray tracing.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This work was supported by the National Key R&D Program of China (2020YFC1523301, 2019YFC1521102), the Key Research and Development Program of Shaanxi (2019GY-215), the Major Research and Development Project of Qinghai (2020-SF-143), and the Scientific Research Program Funded by Shaanxi Provincial Education Department (21JK0931).

REFERENCES

- [1] A. Alhakamy and M. Tuceryan. Real-time illumination and visual coherence for photorealistic augmented/mixed reality. *ACM Computing Surveys*, 53(3), May 2020. doi: 10.1145/3386496
- [2] B. Bitterli, C. Wyman, M. Pharr, P. Shirley, A. Lefohn, and W. Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics*, 39(4):148, Jul 2020. doi: 10.1145/3386569.3392481
- [3] D. Cline, A. Razdan, and P. Wonka. A comparison of tabular PDF inversion methods. *Computer Graphics Forum*, 28(1):154–160, Feb 2009. doi: 10.1111/j.1467-8659.2008.01197.x
- [4] R. Cranley and T. N. L. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13(6):904–914, Dec 1976. doi: 10.1137/0713071

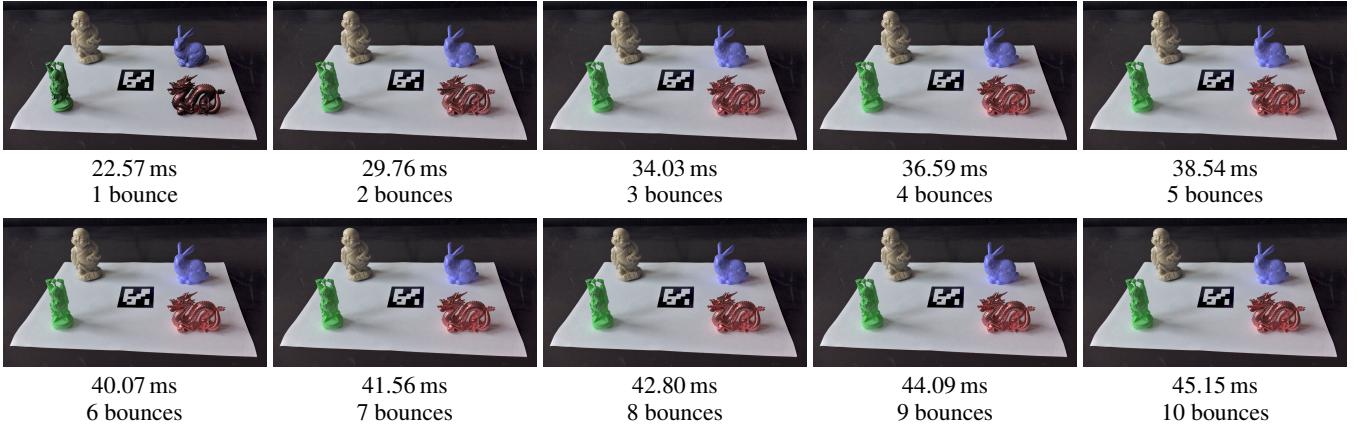


Figure 7: Comparison of different numbers of bounces under lighting condition I (128 spp).

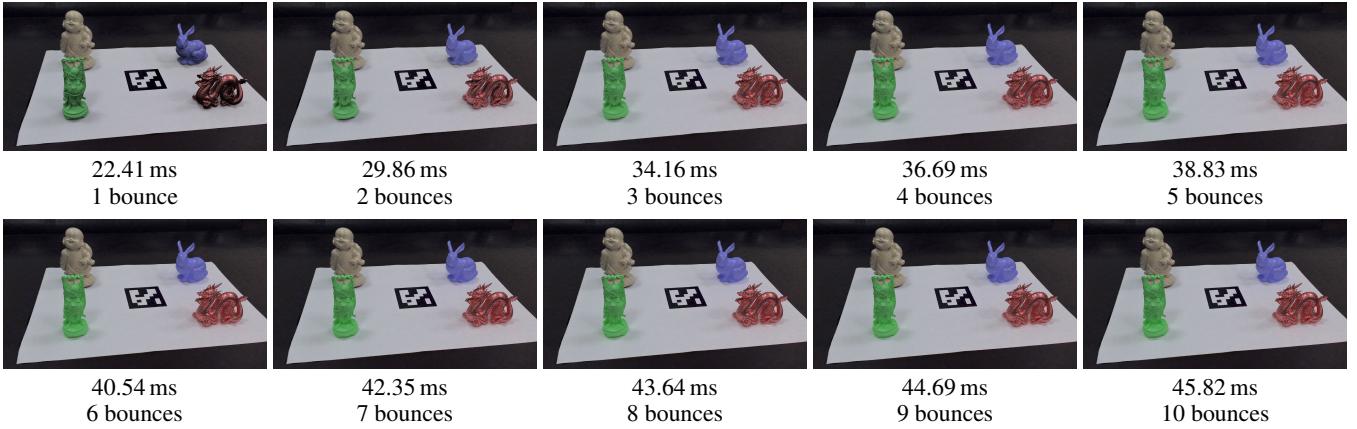


Figure 8: Comparison of different numbers of bounces under lighting condition II (128 spp).



Figure 9: Reflections of virtual objects on the real scene with a non-diffuse BRDF.

- [5] D. Dai, X. Shi, L. Wang, and X. Li. Interactive mixed reality rendering on holographic pyramid. In *Proc. IEEE VR*, pp. 483–492. IEEE, 2022. doi: 10.1109/VR51125.2022.00068
- [6] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. SIGGRAPH*, p. 189–198. Association for Computing Machinery, New York, NY, USA, 1998. doi: 10.1145/280814.280864
- [7] G. Drettakis, L. Robert, and S. Bougnoux. Interactive common illumination for computer augmented reality. In *Proc. EGWR*, pp. 45–56. Springer Vienna, Vienna, 1997. doi: 10.1007/978-3-7091-6858-5_5
- [8] A. Fournier, A. S. Gunawan, and C. Romanzin. Common illumination between real and computer generated scenes. Technical report, Canada, 1992.
- [9] T. A. Franke. Delta light propagation volumes for mixed reality. In *Proc. ISMAR*, pp. 125–132. IEEE, 2013. doi: 10.1109/ISMAR.2013.6671772

- [10] T. A. Franke. Near-field illumination for mixed reality with delta radiance fields. In *ACM SIGGRAPH Posters*. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2503385.2503468
- [11] T. A. Franke. Delta voxel cone tracing. In *Proc. ISMAR*, pp. 39–44. IEEE, 2014. doi: 10.1109/ISMAR.2014.6948407
- [12] T. A. Franke. Interactive relighting of arbitrary rough surfaces. In *ACM SIGGRAPH Posters*. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2614217.2614225
- [13] T. A. Franke. *The delta radiance field*. PhD thesis, Technische Universität Darmstadt, Darmstadt, Jul 2015.
- [14] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, Jun 2014. doi: 10.1016/j.patcog.2014.01.005
- [15] T. Grosch. Differential photon mapping - consistent augmentation of photographs with correction of all light paths. In *Eurographics Short Presentations*. The Eurographics Association, Goslar, DEU, 2005. doi: 10.2312/egs.20051022
- [16] T. Grosch, T. Eble, and S. Mueller. Consistent interactive augmentation of live camera images with correct near-field illumination. In *Proc. VRST*, p. 125–132. Association for Computing Machinery, New York, NY, USA, 2007. doi: 10.1145/1315184.1315207
- [17] L. Gruber, T. Langlotz, P. Sen, T. Höherer, and D. Schmalstieg. Efficient and robust radiance transfer for probeless photorealistic augmented reality. In *Proc. IEEE VR*, pp. 15–20. IEEE, 2014. doi: 10.1109/VR.2014.6802044
- [18] L. Gruber, T. Richter-Trummer, and D. Schmalstieg. Real-time photometric registration from arbitrary geometry. In *Proc. ISMAR*, pp.

- 119–128. IEEE, 2012. doi: 10.1109/ISMAR.2012.6402548
- [19] L. Gruber, P. Sen, T. Höllerer, and D. Schmalstieg. Acceleration methods for radiance transfer in photorealistic augmented reality. In *Proc. ISMAR*, pp. 255–256. IEEE, 2013. doi: 10.1109/ISMAR.2013.6671792
- [20] L. Gruber, J. Ventura, and D. Schmalstieg. Image-space illumination for augmented reality in dynamic environments. In *Proc. IEEE VR*, pp. 127–134. IEEE, 2015. doi: 10.1109/VR.2015.7223334
- [21] K. Jacobs and C. Loscos. Classification of illumination methods for mixed reality. *Computer Graphics Forum*, 25(1):29–51, Mar 2006. doi: 10.1111/j.1467-8659.2006.00816.x
- [22] J. T. Kajiya. The rendering equation. *ACM SIGGRAPH Computer Graphics*, 20(4):143–150, Aug 1986. doi: 10.1145/15886.15902
- [23] P. Kán and H. Kaufmann. Differential progressive path tracing for high-quality previsualization and relighting in augmented reality. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, B. Li, F. Porikli, V. Zordan, J. Kłosowski, S. Coquillart, X. Luo, M. Chen, and D. Gotz, eds., *Proc. ISVC*, pp. 328–338. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-41939-3_32
- [24] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 30(6):1–12, Dec 2011. doi: 10.1145/2070781.2024191
- [25] M. Knecht, C. Traxler, O. Mattausch, W. Purgathofer, and M. Wimmer. Differential instant radiosity for mixed reality. In *Proc. ISMAR*, pp. 99–107. IEEE, 2010. doi: 10.1109/ISMAR.2010.5643556
- [26] M. Knecht, C. Traxler, O. Mattausch, and M. Wimmer. Reciprocal shading for mixed reality. *Computers & Graphics*, 36(7):846–856, Nov 2012. doi: 10.1016/j.cag.2012.04.013
- [27] J. Kronander, F. Banterle, A. Gardner, E. Miandji, and J. Unger. Photorealistic rendering of mixed reality scenes. *Computer Graphics Forum*, 34(2):643–665, Jun 2015. doi: 10.1111/cgf.12591
- [28] P. Kán and H. Kaufmann. High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In *Proc. ISMAR*, pp. 99–108. IEEE, 2012. doi: 10.1109/ISMAR.2012.6402546
- [29] P. Kán and H. Kaufmann. Physically-based depth of field in augmented reality. In *Eurographics Short Papers*. The Eurographics Association, Goslar, DEU, 2012. doi: 10.2312/conf/EG2012/short/089-092
- [30] P. Kán and H. Kaufmann. Differential irradiance caching for fast high-quality light transport between virtual and real worlds. In *Proc. ISMAR*, pp. 133–141. IEEE, 2013. doi: 10.1109/ISMAR.2013.6671773
- [31] P. Lensing and W. Broll. Instant indirect illumination for dynamic mixed reality scenes. In *Proc. ISMAR*, pp. 109–118. IEEE, 2012. doi: 10.1109/ISMAR.2012.6402547
- [32] M. McGuire. Computer graphics archive, Jul 2017. <https://casual-effects.com/data>.
- [33] S. U. Mehta, K. Kim, D. Pajak, K. Pulli, J. Kautz, and R. Ramamoorthi. Filtering environment illumination for interactive physically-based rendering in mixed reality. In J. Lehtinen and D. Nowrouzezahrai, eds., *Proc. EGSR*. The Eurographics Association, Goslar, DEU, 2015. doi: 10.2312/sre.20151172
- [34] T. Ritschel, T. Grosch, M. H. Kim, H. P. Seidel, C. Dachsbacher, and J. Kautz. Imperfect shadow maps for efficient computation of indirect illumination. *ACM Transactions Graphics*, 27(5):129, Dec 2008. doi: 10.1145/1409060.1409082
- [35] K. Rohmer, W. Büschel, R. Dachselt, and T. Grosch. Interactive near-field illumination for photorealistic augmented reality on mobile devices. In *Proc. ISMAR*, pp. 29–38. IEEE, 2014. doi: 10.1109/ISMAR.2014.6948406
- [36] K. Rohmer, W. Büschel, R. Dachselt, and T. Grosch. Interactive near-field illumination for photorealistic augmented reality with varying materials on mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 21(12):1349–1362, Jun 2015. doi: 10.1109/TVCG.2015.2450717
- [37] K. Rohmer, J. Jendersie, and T. Grosch. Natural environment illumination: Coherent interactive augmented reality for mobile and non-mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 23(11):2474–2484, Aug 2017. doi: 10.1109/TVCG.2017.2734426
- [38] A. L. D. Santos, D. Lemos, J. E. F. Lindoso, and V. Teichrieb. Real time ray tracing for augmented reality. In *Proc. VAR*, pp. 131–140. IEEE, 2012. doi: 10.1109/SVR.2012.8
- [39] I. Sato, Y. Sato, and K. Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):1–12, Jan–Mar 1999. doi: 10.1109/2945.764865
- [40] J. F. Talbot, D. Cline, and P. Egbert. Importance resampling for global illumination. In *Proc. EGSR*, pp. 139–146. The Eurographics Association, Goslar, DEU, 2005. doi: 10.2312/EGWR/EGSR05/139-146
- [41] M. Thöner and A. Kuijper. Delta global illumination for mixed reality. In R. Shumaker and S. Lackey, eds., *Proc. VAMR*, pp. 108–118. Springer International Publishing, Cham, 2015.
- [42] E. Veach and L. J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Proc. SIGGRAPH*, p. 419–428. Association for Computing Machinery, New York, NY, USA, 1995. doi: 10.1145/218380.218498
- [43] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *Proc. EGSR*, p. 195–206. Eurographics Association, Goslar, DEU, 2007. doi: 10.2312/EGWR/EGSR07/195-206
- [44] A. Zhang, Y. Zhao, and S. Wang. An improved augmented-reality framework for differential rendering beyond the Lambertian-world assumption. *IEEE Transactions on Visualization and Computer Graphics*, 27(12):4374–4386, Jun 2021. doi: 10.1109/TVCG.2020.3004195
- [45] A. Zhang, Y. Zhao, S. Wang, and J. Wei. An improved augmented-reality method of inserting virtual objects into the scene with transparent objects. In *Proc. IEEE VR*, pp. 38–46. IEEE, 2022. doi: 10.1109/VR51125.2022.00021