Bidirectional Shadow Rendering for Interactive Mixed 360° Videos

Lili Wang *
Beihang University
Peng Cheng Laboratory

Hao Wang
Beihang University

Danqing Dai
Beihang University

Jiaye Leng
Beihang University

Xiaoguang Han
Shenzhen Research Institute of Big Data

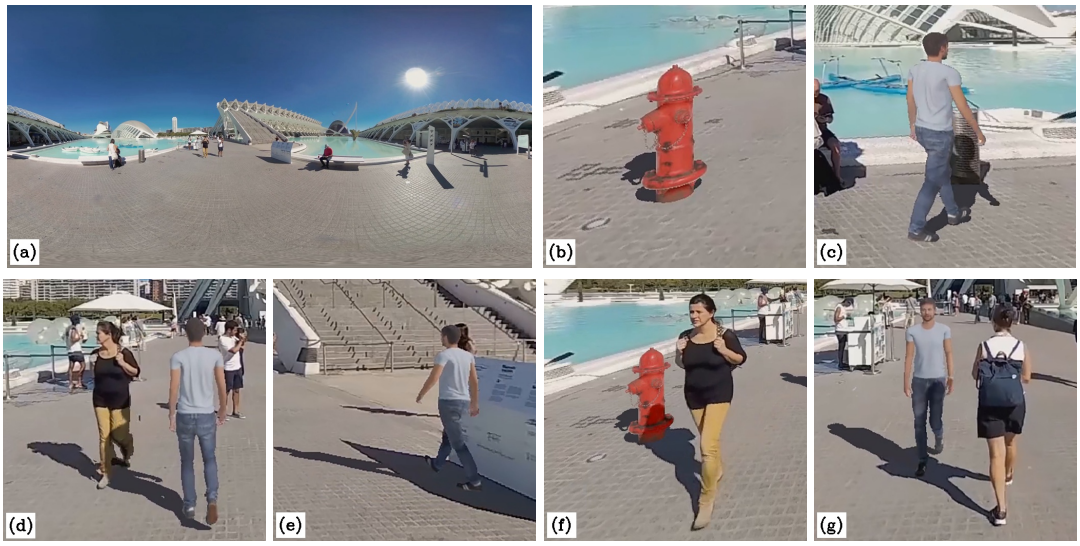

Figure 1: The shadow rendered with our method when a static virtual fire hydrant and a moving virtual character wearing a light blue T-shirt are inserted into the 360° video named *Valencia*. (a) is the original panorama frame. In (b), the virtual red fire hydrant casts its shadow on the ground. In (c), the virtual character casts his shadow on the real trash can in the original video. In (d), the virtual character casts his shadow on the ground and the body of the real person in the video, a lady wearing a black T-shirt. In (e), a real billboard casts its shadow on the body of the virtual character. In (f), the real person casts her shadow on the red fire hydrant. In (g), the real person, a lady with a backpack, casts her shadow on the body of the virtual character.

Abstract

In this paper, we provide a bidirectional shadow rendering method to render shadows between real and virtual objects in the 360° videos in real time. We construct a 3D scene approximation from the current output viewpoint to approximate the real scene geometry nearby in the video. Then, we propose a ray casting based algorithm to determine the shadow regions on the virtual objects cast by the real objects. After that, we introduce an object-aware shadow mapping method to cast shadows from virtual objects to real objects. Finally, we use a shadow intensity estimation algorithm to determine the shadow intensity of virtual objects and real objects to obtain shadows consistent with the input 360° video. The experiment results prove the effectiveness of our bidirectional shadow rendering method for mixed 360° videos. Our method can generate visually realistic shadows for virtual objects and real objects in 360° video in real-time, and make virtual objects more natural to integrate with real scenes in 360° videos of the mixed reality applications.

Index Terms: Mixed reality—360° videos—Shadow rendering

1 Introduction

360° video provides a high-quality realistic scene representation of the real world. It captures the dynamic real world with multiple fisheye cameras or regular cameras arranged around a center point. The overlapped views from all directions are captured by these cameras and stitched into a seamless panoramic video. Generating 360° video is more efficient than the tedious and complex 3D modeling and reconstruction for the scene. With 360° video, users can choose the views of interest in any direction and feel the same as turning around in the real world. 360° video provides a more realistic and natural experience than that of the regular videos. In recent years, the researchers insert the interactive virtual objects and the special rendering effects into 360 videos, such as the moving balls, robots [21], deformable creatures [20] and caustics [26, 27]. 360° video is extended to an interactive media for mixed reality and has broad application prospects in the fields of education, tourism and games.

There are some challenges when inserting virtual objects into 360° videos. One challenge is to maintain consistency of lighting and shadows between virtual and real objects in the video without any explicit 3D information of the real scene. Consistent lighting and shadows can improve the visual fidelity of virtual objects, but the lack of 3D information (such as the location of the main light source and the geometry of the scene) can affect the quality of the generated shadows and lighting. Rhee et al. [21] proposed MR360, which estimated the real-world light source and used the perception-based rendering scheme to perform real-time mixed reality rendering for low dynamic range 360° video. Their method generated a 3D plane to represent the ground in the scene and cast the shadows of the virtual objects onto the ground. However, their method cannot cast shadows of real objects in 360 ° video on virtual objects, nor can cast shadows of virtual objects on real objects except the ground. Missing shadows can sometimes cause visual confusion to users.

*e-mail: wanglily@buaa.edu.cn

2642-5254/21/$31.00 ©2021 IEEE
DOI 10.1109/VR50410.2021.00038

2021 IEEE Virtual Reality and 3D User Interfaces (VR) | 978-1-6654-1838-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/VR50410.2021.00038

Another challenge is performance. 360° video contains a living video stream for all directions, so processing 360° video takes more time than processing regular videos. Therefore, the high efficiency of the consistent lighting and shadow generation methods is the key factor to ensure that 360° video can be interacted with in real-time. This challenge still needs improvement.

In this paper, we provide a bidirectional shadow rendering method for mixed 360° video, which can render real-time shadows from real objects to virtual objects and virtual objects to real objects in 360° video. First, a 3D approximation of the scene from the current output viewpoint is constructed. Then a ray casting based algorithm is proposed to determine the shadow regions on the virtual objects cast by the real objects in the 360° video. Third, an object-aware shadow mapping method is used to cast the shadow from the virtual objects to the real objects. Finally, a shadow intensity estimation algorithm is used to ensure that the shadows on the virtual and real objects have a similar intensity to the shadows in the original input 360° video. The experiment results demonstrate the effectiveness of our bidirectional shadow rendering method for mixed 360° videos. Our method can generate visually realistic shadows for virtual objects and real objects in 360° video in real-time, which is crucial for the consistent appearance of integrated scenes for interactive mixed reality applications with 360° videos. Figure 1 shows the shadow rendering results generated by our method.

2 Prior Work

We briefly discuss related prior work in two areas: mixed reality rendering for 360° videos and shadow rendering for mixed reality.

2.1 Mixed reality rendering for 360° videos

360° video can provide users with a better experience compared to regular videos due to user interactions. It has becomes a popular media for mixed reality when virtual objects are inserted into 360° videos.

Some researchers focused on seamlessly compositing the virtual objects in 360° videos. Iorns et al. [12] proposed an interactive image-based lighting method for low dynamic range 360° live video stream on HMDs. Rhee et al. [21] proposed a perception based rendering schemes [2] for 360° videos and introduced an image-based shadowing method to cast the shadows of the virtual objects on a 3D ground plane constructed. They also designed and implemented the MR360 toolkit to create interactive mixed reality contents [20]. The toolkit detected salient lights in 360° video and cast realistic shadows. Alhakamy et al. [28] proposed an interactive global illumination method for dynamic environments in augmented reality applications. Tarko et al. [24, 25] proposed a real-time virtual object insertion scheme for moving 360° videos. However, the shadows of the virtual objects inserted into the video can only be cast on the ground due to the representation of the real scene.

There are some other novel 360° video rendering ideas in mixed reality. Choi et al. [3] divided the 360° video into multiple layers according to the contents. Different layers with images of real objects were superimposed or replaced to achieve simple interactive effects. Thompson et al. [26, 27] provided a mixed reality rendering solution for underwater 360° video.

These prior methods achieved realistic effects and real-time performance, but they only cast the shadow of virtual objects on the ground plane. While our method considers the bidirectional shadows: the shadow on real scene including real objects in the videos cast from virtual objects and the shadow on virtual objects cast from real objects in 360° videos, which make the rendering effects more realistic.

2.2 Rendering shadows in mixed reality

Photo-realistic rendering is a very important research direction in mixed reality, which helps us to seamlessly synthesize virtual objects with real environments. For a more comprehensive understanding of the rendering in mixed reality, we recommend readers to read the review in [16]. In this section, we only discuss the visibility determination and shadow rendering in the context of mixed reality. According to the acquired data, current mixed reality applications can be divided into two types: with RGBD input and with RGB input. We start with the RGBD data method.

Gruber et al. [9] used Kinect Fusion to reconstruct scene through RGBD input and introduced a visibility determination method by using joint image and visibility space sub-sampling. The method achieved an interactive frame rate on low-resolution images. They improved the geometric representation in [10] and calculated the visibility by spherically sampling the space near a given 3D point. Their approach can produce good soft shadows on low-resolution images in real time.

Trummer et al. [22] proposed a method to recover incident lighting and surface materials from casually scanned geometry and sample the visibility of each vertex by tracing rays into the scene. Gibson et al. [7] proposed an image composition algorithm to render both virtual-to-real and real-to-virtual shadows. And they also extended their method to render soft virtual-to-real shadow in interactive rates based on their shadow blending algorithm [6]. Wei et al. [30] proposed a framework for simulating shadow interaction in outdoor live videos to fix the shadow artifacts when real shadow-casting objects may not be fully visible in the live video.

For mixed reality applications with only RGB input, they usually construct proxies to approximate the geometry of the real scene based on images or videos before rendering lighting and shadows. Grosch et al. [8] used an object-based methods to reconstruct scene only with a light probe's HDR image as input. Then, the virtual-to-real and real-to-virtual shadows are rendered with shadow mapping. Unger et al. [29] introduced a focal volume modeling method to construct a geometric proxy of the real scene, and then re-projected the radiance data onto the proxy to determine the illumination and shadow on the virtual objects. Kán et al. [13] introduced parallel differential irradiance calculation for the calculation of diffuse light transportation between the virtual world and real-world, in which they accurately solved the visibility problem through ray-tracing. Knecht et al. [15] constructed imposters to approximate real objects and used conventional shadow mapping method to determine shadows. Karsch et al. [14] reconstructed high-quality depth to produce high-quality illumination and shadow effects. Mehta et al. [17] used dense SLAM to track the camera and reconstruct the scene and introduced Fourier analysis for environment lighting to calculate the lighting on the diffuse material with visibility.

For 360° video, it is difficult to capture high-resolution depth information without special complex device. Therefore, our method takes an RGB stream as input. Based on the information detected from 360° videos, we construct the 3D planar proxies to represent the real scene. This information is used in subsequent processing to calculate visibility and render shadows.

3 Bidirectional Shadow Rendering

Inspired by the idea of the double shadow in augmented reality in previous work [7, 8], we propose bidirectional shadow rendering method for mixed 360° videos. Our bidirectional shadow rendering method takes 360° video, the user's view direction, virtual objects and their positions as input. The output is a user view with the realistic shadows from the real objects to the virtual objects and from the virtual objects to the real objects. Figure 2 shows the entire process of the bidirectional shadow rendering method. Our method has 4 main steps: 1) Constructing 3D scene approximation; 2) Generating the shadow regions on the surface of the virtual objects cast from the real objects; 3) Generating the shadow regions on the surfaces of the real objects cast from the virtual objects; 4) Rendering base on the image-based lighting method [5].

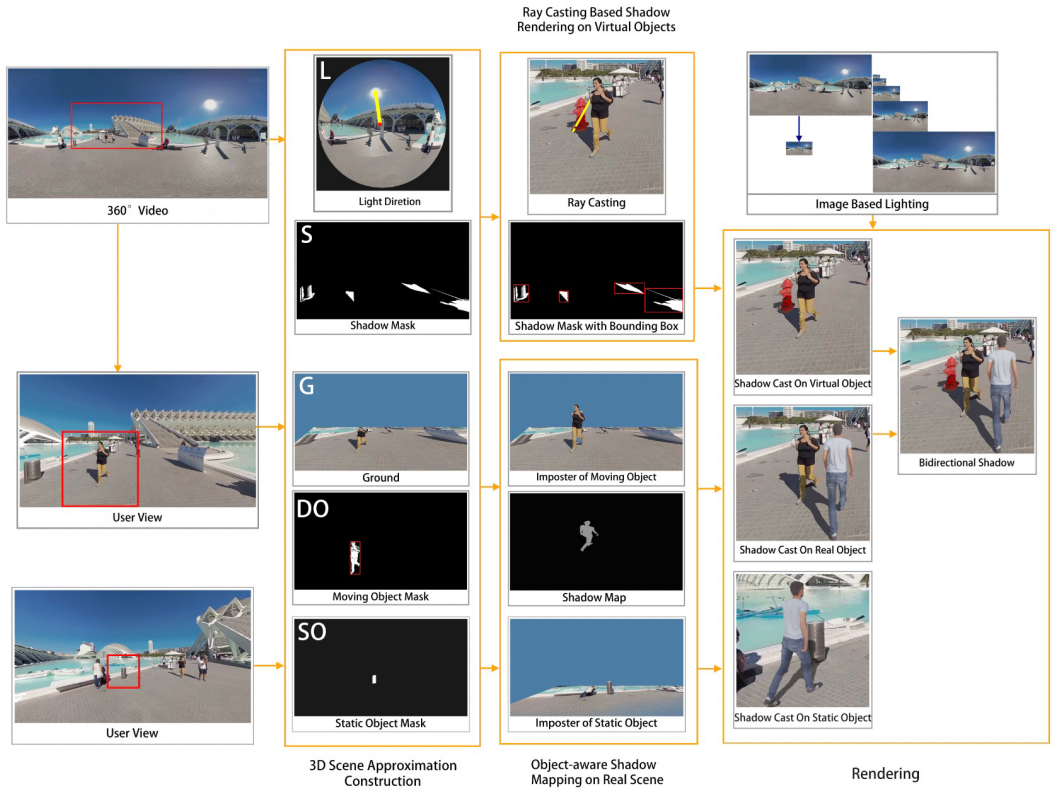

Figure 2: The pipeline of the bidirectional shadow rendering method on mixed 360° video.

3.1 3D Scene Approximation Construction

For a given user's view, we construct a quintuple (L, G, DO, SO, S) to represent the real scene, which is used to generate the 3D planar proxies of the scene in the subsequent steps. L represents the direction of the main light source in the 360° video; G is a 3D plane representing the ground in the 360° video; DO is the binary mask for moving objects in the user view; SO is the binary mask for static objects in the user view; S is the binary mask for shadow, which indicates the 2D shadow region on the ground in the user view.

We apply the light detection and threshold determination method proposed by Rhee et al. [21] to estimate the position and intensity of the main light source in the panorama image. Threshold detection is applied to the upper part of the panoramic image since the light source usually appears in the upper part of the panoramic image of the outdoor scene. After clustering, blocking and sorting the detection results, the first light source is used as the main light source. We use the direction from the center of light sources to the user's position as the light source direction L. We assume that a 360° video is taken from a fixed viewpoint in an outdoor scene and completed within a certain period. The main light source will not change much, so we only extract the main light source direction based on the first frame of the 360° video, and the subsequent processing uses this direction as the light source direction.

Debevec et al. [5] used simple geometry of the local scene to represent the surfaces on which virtual objects were placed or interacted with in the real scene. Virtual objects cast shadows and reflect light on these surfaces. In our case, it is assumed that the viewpoint of 360° video is fixed and the scene is an outdoor scene, so a flat diffuse ground plane G is used as the local scene to represent the ground of the real scene. The geometry of G can be estimated based on the viewpoint of 360° video.

In order to generate the moving object mask DO, we adopt the background/foreground segmentation algorithm based on Gaussian mixture (MOG2) [31] and detect the regions of the moving objects based on pixel similarity by comparing each frame in the video to its previous frames. We mark the moving objects as '1' and the background as '0' to generate DO. DO is used to create the geometric approximations of the moving objects in the real scene, so the shadow regions of the moving object in DO need to be removed. With the MOG2, we can reconstruct a complete background image from all previous frames. The shadow regions of the moving objects are determined by comparing the detected foreground with the complete background image and removed from DO.

We use user interaction to construct masks for static objects. Users are required to wear an HMD and observe static objects of interest one by one for 2 seconds in the first frame of the 360 ° video. Using the salient object detection method in [11], we estimated the positions of all static objects of interest in the 360° video during the preprocessing. When the video is played in real-time, the static object mask SO of the current view will be generated according to the position of the salient object. Semantic segmentation and instance segmentation methods based on deep learning can be used to automatically construct SO in future work, such as DEtection TRansformer (DETR) from Facebook [1].

To construct the shadow mask S, we detect the shadow regions in the user's view by setting a threshold for the luminance of the Y channel and marking the shadow as '1' and non-shadow as '0' in S. S will be mapped onto the ground plane and used to calculate the shadow on the surfaces of the virtual objects cast by the real objects, so we remove the shadow region on the moving objects by subtracting DO from S and only keep the shadow region on the static

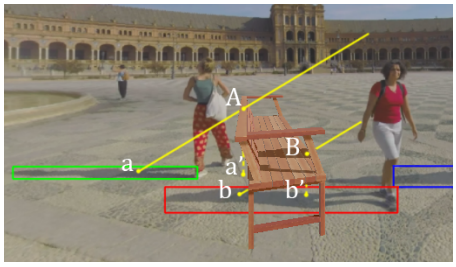

Figure 3: Example of removing overestimated shadows based on shadow bounding box.

part of the real scene to avoid the wrong shadow in the calculation.

3.2 Shadow rendering on virtual objects

We propose a shadow determination algorithm based on ray casting, which can calculate the shadows generated by real objects on virtual objects. The inputs of the algorithm are the light source direction L, the ground plane G, the shadow mask S from the 3D geometry approximation of the scene, the virtual objects and their positions on G. The output is the shadow region on the virtual objects' surface in the user view.

The algorithm has two steps. In the first step, we generate conservative a shadow region on the surface of the virtual objects. First, we map the shadow mask S to the ground plane G. For each active pixel in the binary mask S, a 3D point on G is obtained. We emit rays from all these 3D points in the direction of light sources L. If the rays intersect with the virtual objects, the intersections on the surfaces of the virtual objects are in the shadow region. In this way, the shadow region on the surfaces of the virtual object is overestimated, i.e. the regions that are not initially occluded are marked as shadow.

Figure 3 gives an example of shadow overestimation. A virtual bench is inserted between two real people in the video. From point a and b in the shadow regions of these two people, two rays are emitted along L and intersect with the bench at A and B. Therefore both A and B are set as the shadow. As we can see from the image, the lady in red pants is on the left side of the bench, and her shadow is also on the left side of the bench. Therefore the lady in red pants should not cast the shadow on the bench. The shadow is overestimated.

To remove the false shadow region on the surface of the virtual object, we introduce a shadow refinement method based on the 2D shadow bounding box. First, we cluster the pixels in the shadow mask by exploring the shadow connectivity. Then, we construct a 2D bounding box for each cluster. Thereafter, we use orthogonal projection to project the intersection on the surface of the virtual object onto the ground plane G. If the projection is not in the same bounding box as the shadow point emitting the corresponding intersecting ray, the shadow at this location is removed. In Figure 3, a' and b' are the projections of the intersection points A and B. In our shadow refinement, b and b' are in the same shadow bounding box, so B is kept as the shadow; while a and a' are not in the same shadow bounding boxes, so A is removed from the shadow.

3.3 Object-aware shadow mapping on real scene

In this section, we introduce an object-aware shadow mapping method to determine the shadow region on the real objects cast by the virtual objects. We did not use the ray tracing framework because of the low computational efficiency. Since both real and virtual objects may be displaced and deformed, if ray tracing is used, the hierarchical bounding box needs to be updated in real time, which results in high time cost.

To calculate the shadow regions on the surface of real objects, we need the geometry of the real objects. Instead of 3D reconstructing, we construct imposters to approximate the geometry of the static and moving real objects in the video. The inputs of our object-aware shadow mapping algorithm are the direction of the light source L, the ground plane G, the dynamic object mask DO, the static object mask SO, the virtual objects, and their positions on G. The output is the shadow regions on the real objects in the user view. Our method has three steps: 1) constructing imposters for each static objects and moving objects from SO and DO, and placing the imposters at the right positions on the ground plane G. 2) rendering a shadow map with the geometry of the virtual objects along L. 3) determining the shadow region on the imposters and G by comparing the depth with the shadow map generated in the previous step.

We estimate the size and the positions of the imposters based on the static object mask SO and the moving object mask DO. To construct imposters representing static and moving real objects, we cluster pixels based on their connectivity, segment a single object region from SO and DO, and construct a 2D bounding box for each object. We assume that the objects in the 360° video are placed on the ground and perpendicular to the ground. Based on this assumption, the depth of the imposter is determined by indexing the z value of the ground plane G according to the lowest pixel of each object region in the user view. The 2D bounding box of a single static or moving object is unprojected into the 3D world using this z value. Considering that the object may not face the camera, we can rotate the imposter by an angle, which can be determined based on the z value of the lowest pixel of each object and the z value of the left and right sides of the object.

3.4 Rendering

In rendering, similar to MR360 [21], we also use the differential rendering proposed by Debevec [5] and introduce a shadow intensity estimation method to render the shadows with the correct intensity.

Since panoramic video records lighting information from the 360° surrounding environment, we use image-based lighting to render virtual objects. We assume that a 360° video is taken from a fixed viewpoint in an outdoor scene and completed within a certain period. The lighting will not change much, so there is no need to calculate the reflection maps for each frame. We calculated the gloss map for different roughness and saved it in a 5-level mipmap. We obtain the first frame of the 360° video, use inverse tone mapping [12] to convert it to a high dynamic panorama and calculate the reflection maps. These maps will be used for lighting rendering of all subsequent frames in the differential rendering.

After determining the shadow regions on the virtual objects and the real scene, it is necessary to estimate the intensity of the shadow so that the composite shadow can be consistent with the shadow already existing in 360° videos. Based on the shadow mask, the average shadow intensity of the pixels in the shadow region of the video is calculated and used as the intensity of the nearby synthesized shadow. In our implementation, for each region of the synthesized shadow, the nearest shadow bounding box is found. We compute the intensity based on the RGB color of the pixel in the shadow within the bounding box. This intensity is used as the coefficients of the specular reflection, diffuse reflection components, and the direct illumination from the light source in the synthesized shadow region in the differential rendering. To maintain the temporal coherence between the frames of 360° videos, if the nearby shadow region is smaller than a pre-defined threshold, we do not update the intensity of the synthesized shadow.

4 Results and Discussion

We used an HTC Vive system (tracker, HMD, and wireless hand-held controller) to track the user's head movement and hand interaction. The Vive was connected to a PC workstation with a 3.2 GHz Intel(R) Core(TM) i7-8700 CPU, 16 GB of memory, and an NVIDIA GeForce RTX 2070 SUPER graphics card. We tested our method on five 360° videos: three were from the internet *Valencia* (Figure

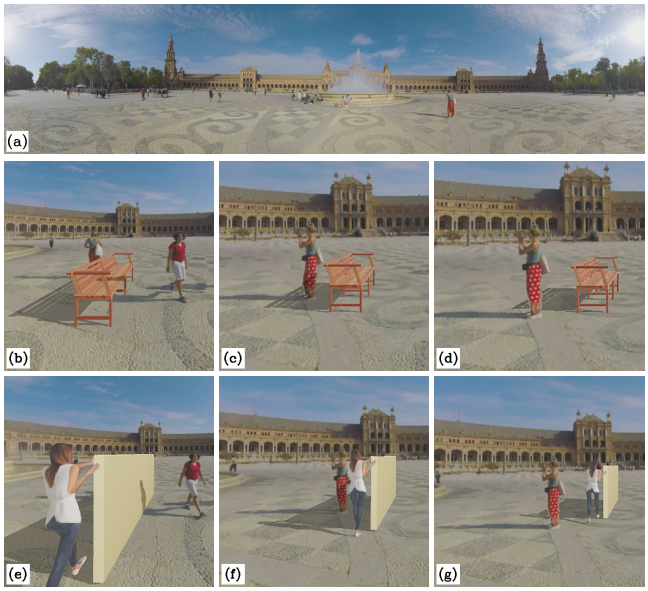

Figure 4: The shadow rendering results of *Sevilla* using our method. (a) is a panorama frame. In (b), the bench casts its shadow on the ground, and the walking lady in a red T-shirt in the video casts her shadow on the bench. In (c), the bench casts its shadow on both the ground and the walking lady in red pants in the video. In (d), the lady in red pants occludes the shadow of the virtual bench, which also shows the benefit of the moving object imposters generated. In (e), two virtual characters carrying a large board pass by, and the walking lady in a red T-shirt in the video casts her shadow on the board. In (f), two virtual characters keep walking, and the shadow of the board are cast on the lady in red pants. In (g), the lady in red pants occludes the shadow of the virtual characters and the board.

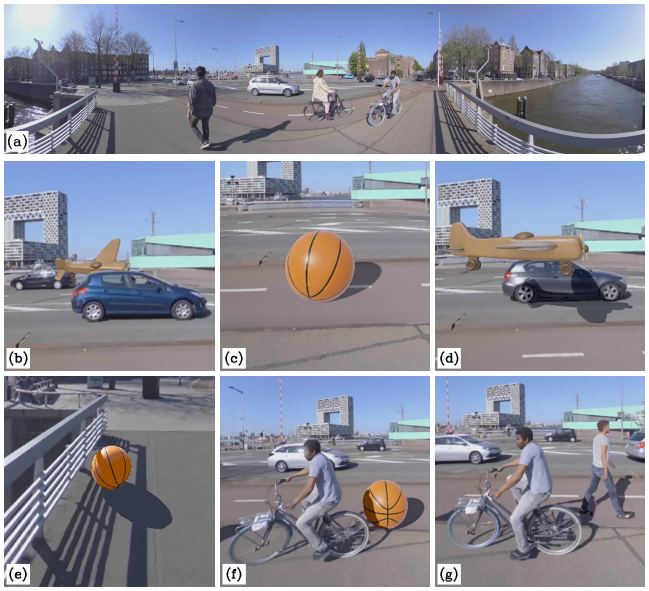

Figure 5: The shadow rendering results of *Amsterdam* using our method. (a) shows a panorama frame. In (b), the virtual plane is occluded by the real moving car in the video. In (c), the virtual basketball cast a shadow on the ground. In (d), the virtual flying plane casts its shadow onto the fast-moving car in the video. In (e), the shadow of the real fence is cast on the flying plane. In (f), a passerby on a bicycle casts the shadow on the virtual basketball. In (e), the shadow of the passerby on a bicycle is also projected on a virtual walking character wearing a light blue T-shirt.

1), *Sevilla* (Figure 4), *Amsterdam*(Figure 5), two were generated by using 3D scenes, *Station* (Figure 9 left part) and *Village* (right part). The resolution of the user view is 1920×1080.

4.1 Quality

Figure 4 and Figure 5 show shadow rendering results by using our bidirectional shadow rendering method for two 360° video sequences from the internet: *Sevilla* and *Amsterdam*. We inserted a static virtual object, a bench and two walking people carrying a large wooden board into 360° video *Sevilla*. In 360° video sequence *Amsterdam*, a static virtual object, a large basketball, two moving virtual objects, a flying plane and a walking character wearing a light blue T-shirt, are inserted. Our method can generate correct occlusion relations between the real objects, the virtual objects and the shadows, and render the visually realistic shadows between real objects and virtual objects in 360° videos. We also refer readers to our accompanying video.

In Figure 6, we compare the shadow effects rendered on virtual object when the shadow on the moving object is removed and the shadow on the moving object is not removed from the shadow mask *S*. The shadow detection method may not work well for some cases. In the image on the left, the detected shadow regions of *S* are on the ground and the walking lady's legs (red region). If *S* is used directly to determine the shadow region on the fire hydrant, the false shadow appears on the fire hydrant. This is because the rays emitted from the shadow region on the right leg of the lady (yellow ray) may intersect with the fire hydrant, and the intersections are kept as shadow due to the unreasonable large shadow bounding box. In the image on the right, we remove the shadow on the walking lady in *S*, so the shadow on the lady's leg will not be cast on the virtual fire

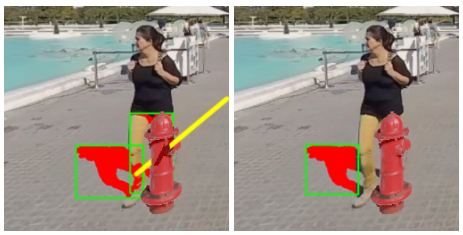

Figure 6: The shadow rendering effects on the virtual fire hydrant with (right) and without removing the shadow on the lady from *S* (left) are compared.

hydrant. Removing shadows on moving objects can help us avoid false shadows on virtual objects.

Figure 7 shows the comparison of the shadow effects with (right) and without our shadow intensity estimation (left). The shadow of the virtual plane is darker than the shadow of the walls in the video since we just set a predefined intensity value. While with our shadow intensity estimation (right), the shadow intensity of the plane is consistent with the shadow of the walls. Our shadow intensity estimation step can make the generated shadows have the similar intensity as the shadows nearby in the video.

Figure 8 shows some failure cases of our method. (a) and (b) show the case with the overlapping shadow. (c) shows the case with non-flat real objects. (d) shows the case with a real object in the air.

4.2 Comparison on two synthetic scenes

To quantitatively measure the quality of shadows generated with our method, we construct two 3D scenes using 3D Max: *Station* and *Village*, which include static objects and moving objects. We render each scene from a fixed viewpoint to generate the sequences of a

Figure 7: The shadow rendering effects with (right) and without our shadow intensity estimation (left) are compared.

Figure 8: Failure cases. In (a), the green area indicates that the shadow on the back of the virtual bench is overestimated, which is caused by the overlap of the shadows of two real people in the scene. For comparison, when the shadows of two people do not overlap. (b) shows the correct shadow on the bench. The green area in (c) marks the incorrectly shaded area on the real motorcycle projected by the virtual fire hydrant, because we use the vertical imposter to approximate the non-flat motorcycle. In (d), the girl jumps into the air, while our method constructs the imposter from the foot position, so that the depth of the imposter will be farther than the real situation. The virtual fire hydrant should be behind the girl, and its shadow should be cast directly on the ground.

360° video with conventional shadow mapping. Then a fire hydrant and a walking character are inserted into the video *Station*, and a stone ball and a running character are inserted into the video *Village* as virtual static objects and moving objects. The ground truth image are rendered with full geometry of the scenes and the objects.

Figure 9 shows a comparison between the results of our method and the ground truth. By comparison, we can see that our shadows are very similar to the ground truth, and most errors appear at the shadow boundaries. It is because the estimated direction of the light source in the 360° video has some deviation, which causes a slight displacement of the shadow. Another reason for false shadows on virtual objects or characters is that since we rasterize the geometric scene to generate a 360-degree video, the shadow boundary detected from the video in the shadow mask generation step is slightly different from the shadow boundaries in the ground truth. Based on this incorrect shadow mask, we may cast some rays that should not cast or may miss some rays that should cast in ray casting. We use flat imposters to approximate the real objects in 360° videos, so shadow errors appear on the real objects in the video when the virtual objects cast their shadow on these imposters.

For each user view, the shadow error is calculated by dividing the number of error pixels by the number of shadow pixels marked in green in the ground truth. Due to geometry approximation of the real character in the video, the maximum error of *Station* is 10.3% (the image in row (D)), and the maximum error of *Village* is 11.0% (the image in row (D)). By averaging the shadow errors of all user views, the average shadow error of the 360° video sequence can be obtained. The average shadow error of *Station* is 4.6%, and the average shadow error of *Village* is 4.8%.

4.3 Performance

Our method includes four steps: Step 1. Construct a 3D geometry approximation of the scene; Step 2. Generate the shadow region on the surface of the virtual object occluded by the real object; Step 3. Generate the shadow region on the surface of the real object occluded by the virtual object; Step 4. Rendering. We implemented the first step using CUDA accelerated OpenCV library. We implemented step 2 to step 4 on the GPU using OpenGL and Shader (GLSL 330 core). Table 1 shows the performance of our method for different scene complexity in a single user view (Figure 10). The increase in the number of real objects (RO) for shadow interaction will increase the time cost of Step 1, but it has almost no effect on the time cost of Step 2-4. This is because the more real objects, the longer the segmentation time cost, while the imposter-based shadow rendering is very fast. Step 1 only processes with real objects, so the number of virtual objects has no effect on step 1. The number of static virtual objects (SVO) has little effect on the time cost of Step 2-4, while the number of dynamic virtual objects (DVO) is proportional to the time cost of Step 2-4. In the case of 6 SVOs and 1 DVO, 2.7ms of 4.3ms of Step 2-4 is spent on shadow calculation, and 1.6ms is spent on updating the DVO animation; in the case of 6 SVOs and 1 SVO, 2.9ms of 7.7ms of Step 2-4 is spent on shadow calculation, and 4.8ms is spent on updating the DVO animation. Our method can reach above 45 fps in all these cases and can support interactive mixed reality applications with 360° videos. In summary, for complex scenes with many real objects, only the time cost of Step 1 increases linearly with the number of real objects. Since it is computed in parallel, the increase is not significant. The time Cost of Step 2-4 does not change. Therefore, for complex scenes, the overall frame rate will not decrease linearly as the complexity of the real scene increases.

4.4 User study

We have evaluated the bidirectional shadow rendering for 360° videos in a controlled user study. One control condition is to insert the virtual objects in the video without rendering their shadows (CC1), and another is to only render the shadow of the virtual objects only on the ground plane (CC2). We use our method as an experimental condition (EC). For each video *Valencia*, *Sevilla*, *Amsterdam*, we generated three versions according to three conditions. We have recruited 24 participants, 20 males and 4 females, between 20 and 28 years old. Ten of our participants had used HMD VR applications before. Participants had normal and corrected vision.

Participants used the HTC Vive VR system with a handheld controller. Participants can choose to watch 360° videos from different directions by rotating their heads and interact with the video by inserting virtual objects. For each participant, we played nine videos randomly. Participants watched and interacted with the video. After each video, we asked participants to fill in the mixed reality presence measurement questionnaire [18] and the sub-questionnaire of igroup presence questionnaire (IPQ) [23].

In the mixed reality presence measurement questionnaire, participants rated 7 questions. After that, we used the weighted sum of the scores of 7 questions to calculate the scores of three components: realness, spatial presence, and perceptual stress [18]. Table 2 shows the scores of three components of the mixed reality presence measuring questionnaire under each condition. For all three components, EC scores are much better than CC1 and CC2 scores, and CC2 scores are better than CC1. While watching the video, 2 participants reported that the virtual character with shadow looked more realistic, and the virtual character without shadow looked like a ghost. We believe that the lack of shadows would cause the virtual objects to float on the screen instead of blending into the real scene. The more

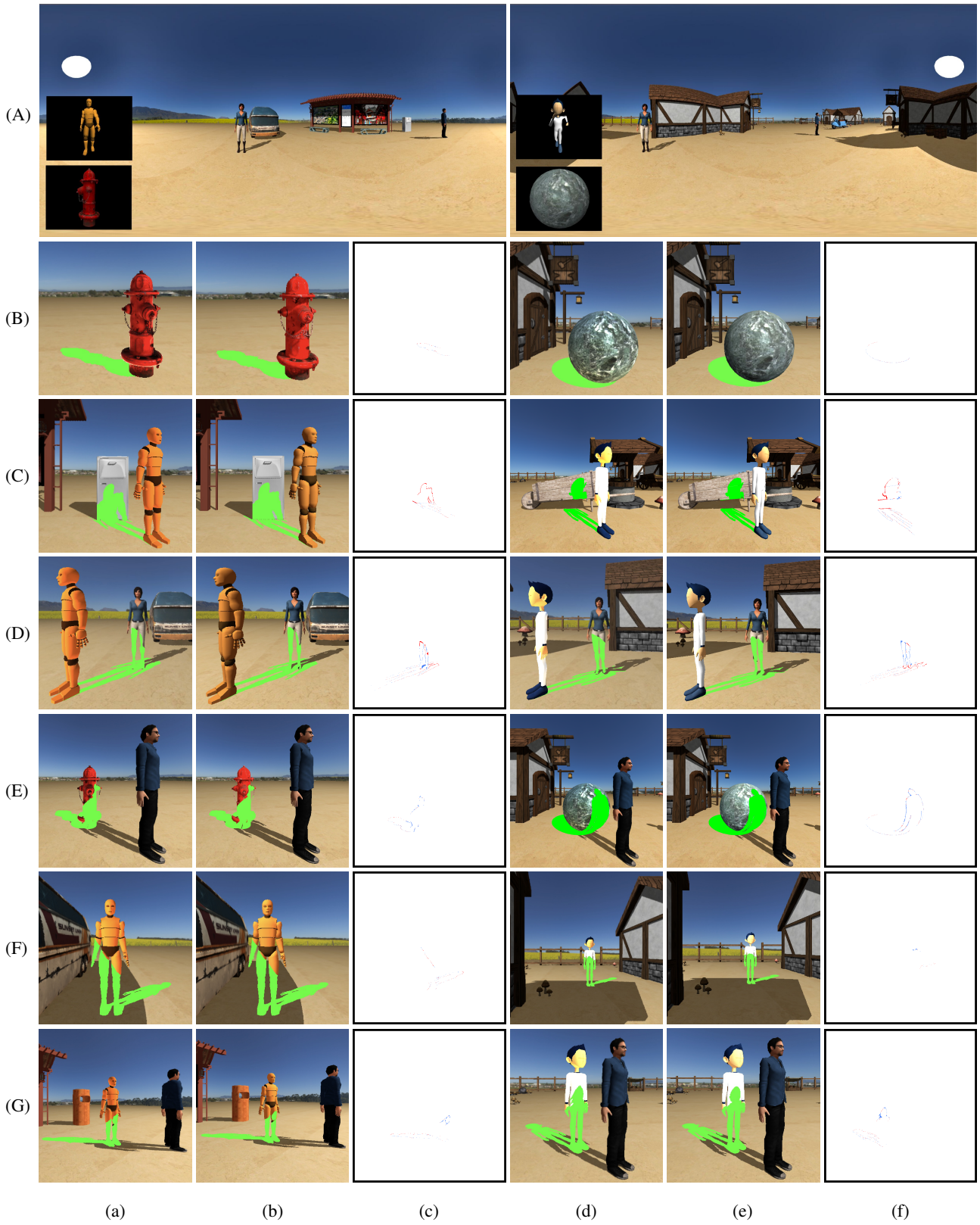

Figure 9: The comparison between the shadow regions generated with our method (a, d) and the ground truth (b, e). The errors are visualized in (c) and (f), red and blue pixel indicate the missing shadow and redundant shadow of our method compared to ground truth. Images in row (A) are the panoramic frames of the synthesized 360° video sequence. Images in row (B) show the shadow on the ground cast by the virtual fire hydrant and the virtual stone ball. Images in row (C) show the shadow on the trash can and the wooden cart in the video cast by the moving virtual characters. Images in row (D) show the shadow on the ground and the bodies of the walking characters cast by the virtual walking or running character. Images in row (E) show the shadow on the virtual fire hydrant and the virtual ball cast by the walking character in the video. Images in row (F) show the shadow on the virtual characters cast by the static bus and house in the video. Images in row (G) show the shadow on the virtual walking or running character cast by the real chara video.

Table 1: Performance for different scene complexity in a user view.

Scene Complexity in a user View	Step 1	Step 2-4	FPS
3-6 RO + 3 SVO	13.3ms	2.5ms	63
3-6 RO + 6 SVO	13.3ms	2.7ms	63
3-6 RO + 6 SVO + 1 DVO	13.3ms	4.3ms	57
3-6 RO + 6 SVO + 2 DVO	13.4ms	7.7ms	47
10-20 RO + 6 SVO	13.9ms	2.7ms	60
10-20 RO + 6 SVO + 2 DVO	13.9ms	7.7ms	46

Figure 10: A user view with high scene complexity. We detected six real people in *Amsterdam* and inserted a virtual trash can, a virtual fire hydrant, a virtual basketball, three virtual roadblocks and two virtual walking characters. The bidirectional shadows between the virtual and real objects were rendered with our method.

realistic the shadow, the better presence the user feels. The normality of the result data was assessed using Shapiro-Wilk test. The scores of these three components are not normally distributed, so we used the Wilcoxon signed-rank test (with z-value) [19] to analyze the differences in the data of each condition. The difference between our results and CC1 or CC2 is significant. In addition to the p value of the statistical test, we also estimated the size of the effect using Cohen's d [4]. The effect size of Cohen's d ranges from *Large* to *Huge*.

The sub-questionnaire of IPQ contains four questions about realism (REAL) in the original IPQ. Table 3 shows the scores of four REAL questions of IPQ under three conditions. Among the three conditions, EC scores are the highest , followed by CC2, and finally CC1. The scores of the four questions also are not normally distributed, so we still used the Wilcoxon signed-rank test to analyze

Table 2: Mixed reality presence measurement questionnaire data.

Component	Con.	Avg ± std.dev.	z-value	p	Cohen's d	Effect size
Realness	*EC*	3.5±2.7				
	*CC*1	−4.0±3.1	−7.4	<0.001	2.6	*Huge*
	*CC*2	−1.3±3.3	−7.0	<0.001	1.6	*Very large*
Spatial presence	*EC*	4.4±2.9				
	*CC*1	−0.2±4.2	−7.1	<0.001	1.3	*Very large*
	*CC*2	1.3±3.2	−6.2	<0.001	1.0	*Large*
Perceptual stress	*EC*	−2.0±1.9				
	*CC*1	2.4±2.2	−7.4	<0.001	2.1	*Huge*
	*CC*2	0.6±2.8	−5.8	<0.001	1.1	*Large*

Table 3: The data of four questions about realism in IPQ.

Component	Con.	Avg ± std.dev.	z-value	p	Cohen's d	Effect size
Q1	*EC*	4.1±0.9				
	*CC*1	1.1±1.3	−7.3	<0.001	2.7	*Huge*
	*CC*2	2.5±1.3	−6.5	<0.001	1.5	*Very large*
Q2	*EC*	4.2±1.0				
	*CC*1	0.9±1.0	−7.4	<0.001	3.4	*Huge*
	*CC*2	2.5±1.2	−7.0	<0.001	1.6	*Very large*
Q3	*EC*	3.8±1.1				
	*CC*1	0.5±0.7	−7.5	<0.001	3.5	*Huge*
	*CC*2	2.1±1.4	−6.2	<0.001	1.3	*Very large*
Q4	*EC*	2.8±1.5				
	*CC*1	0.3±0.6	−6.9	<0.001	2.2	*Huge*
	*CC*2	1.3±1.4	−5.7	<0.001	1.0	*Large*

the differences in them of each condition. The scores of four questions generated by the experimental condition are all significantly different from the scores under the two control conditions. The effect size of Cohen's d ranges from *Large* to *Huge*. Our method achieves better realism of the virtual objects inserted into the real scenes due to a better quality of shadows.

4.5 Limitations

One limitation of our method is that if the shadow regions of multiple real objects overlap in 360° videos, there may be some errors in the shadows on the virtual objects. This is because we use its shadow bounding box to determine whether the real object is on the shadow side of the virtual object or the sun side. If the shadows of multiple real objects overlap each other, our method will generate a large bounding box containing all these overlapping shadows. We cannot determine which of these real objects with overlapping shadows will cast shadows on virtual objects, so the false shadow may be cast on the virtual object.

Another limitation of our method is that our method cannot correctly cast shadows of virtual objects on real objects that are suspended in the air or not placed vertically. This is because we placed the imposter of the real object vertically on the ground. If the real object is floating in the air, the imposter's position will be wrong, and the shadow mapping method will cast the shadow to the wrong position of the real scene. If the real object is not perpendicular to the ground, the shape of the shadow region on the imposter will be incorrect. Our method uses a single imposter with a rotation to approximate the real object. If the depth difference of each part of a single real object is too large, it will cause a certain error in the shadow of the virtual object cast on the real object.

To run the entire pipeline in real-time, we use a simple threshold based method to detect shadows quickly. Therefore, it may also treat black or darker objects in the scene as shadows, resulting in incorrect surface shadows of virtual objects in some scenes. Shadow algorithms based on deep learning may get better results, such as using generative adversarial network to detect shadows.

5 Conclusion and Future Work

We have proposed a bidirectional shadow rendering method for mixed 360° videos, which can render the shadows between the virtual objects and the real scene in 360° videos. The shadow rendering effects of our method on the synthesized 360° videos are similar to the effect of using the conventional shadow mapping method with known 3D geometry of the scenes. Our method generates visually realistic shadows for virtual objects and real objects in 360° videos in real-time, and make virtual objects more natural to integrate with real scenes in 360° videos of the mixed reality applications.

Our method used ground plane and imposters to approximate the geometry of the scene. Due to imprecise geometry, there are some errors in the synthesized shadow. In the future, more accurate geometry reconstruction methods can be used to improve the quality of shadows. Our method used the segmentation algorithms [11,31] to obtain the mask of static and dynamic objects, and the segmentation effect meets the requirements of the subsequent processing of the tested videos. In the future, more robust segmentation algorithms can be directly integrated into our method to improve the segmentation for the videos of complex scenes.

Acknowledgments

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

REFERENCES

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[2] A. Chalmers, J. J. Choi, and T. Rhee. Perceptually Optimised Illumination for Seamless Composites. In J. Keyser, Y. J. Kim, and P. Wonka, eds., *Pacific Graphics Short Papers*. The Eurographics Association, 2014. doi: 10.2312/pgs.20141268

[3] K. Choi, Y.-J. Yoon, O.-Y. Song, and S.-M. Choi. Interactive and immersive learning using 360° virtual reality contents on mobile platforms. *Mobile Information Systems*, 2018:1–12, 2018. doi: 10.1155/2018/2306031

[4] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[5] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, pp. 1–10. 2008.

[6] S. Gibson, J. Cook, T. Howard, and R. Hubbold. Rapid shadow generation in real-world lighting environments. In *Rendering Techniques*, pp. 219–229. Citeseer, 2003.

[7] S. Gibson and A. Murta. Interactive rendering with real-world illumination. In *Eurographics Workshop on Rendering Techniques*, pp. 365–376. Springer, 2000.

[8] T. Grosch. Panoar: Interactive augmentation of omnidirectional images with consistent lighting. *Mirage 2005, Computer Vision/Computer Graphics Collaboration Techniques and Applications*, pp. 25–34, 2005.

[9] L. Gruber, T. Langlotz, P. Sen, T. Höherer, and D. Schmalstieg. Efficient and robust radiance transfer for probeless photorealistic augmented reality. In *2014 IEEE Virtual Reality (VR)*, pp. 15–20. IEEE, 2014.

[10] L. Gruber, J. Ventura, and D. Schmalstieg. Image-space illumination for augmented reality in dynamic environments. In *2015 IEEE Virtual Reality (VR)*, pp. 127–134. IEEE, 2015.

[11] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[12] T. Iorns and T. Rhee. Real-time image based lighting for 360-degree panoramic video. In *Image and Video Technology*, pp. 139–151. Springer, 2015.

[13] P. Kán and H. Kaufmann. Differential irradiance caching for fast high-quality light transport between virtual and real worlds. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 133–141. IEEE, 2013.

[14] K. Karsch, K. Sunkavalli, S. Hadap, N. Carr, H. Jin, R. Fonte, M. Sittig, and D. Forsyth. Automatic scene inference for 3d object compositing. *ACM Transactions on Graphics (TOG)*, 33(3):1–15, 2014.

[15] M. Knecht, C. Traxler, C. Winklhofer, and M. Wimmer. Reflective and refractive objects for mixed reality. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):576–582, 2013.

[16] J. Kronander, F. Banterle, A. Gardner, E. Miandji, and J. Unger. Photorealistic rendering of mixed reality scenes. In *Computer Graphics Forum*, vol. 34, pp. 643–665. Wiley Online Library, 2015.

[17] 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 *the proceeding of EGSR 2015*, pp. 107–118, 2015.

[18] H. Regenbrecht and T. Schubert. Measuring presence in augmented reality environments: Design and a first test of a questionnaire. In *the Proceedings of the Fifth Annual International Workshop Presence*, pp. 1–4, 2002.

[19] D. Rey and M. Neuhäuser. Wilcoxon-signed-rank test. In *International encyclopedia of statistical science*, pp. 1658–1659. Springer, Berlin, Heidelberg, 2011.

[20] T. Rhee, A. Chalmers, M. Hicks, K. Kumagai, B. Allen, I. Loh, L. Petikam, and K. Anjyo. Mr360 interactive: playing with digital creatures in 360° videos. In *SIGGRAPH Asia 2018 Virtual & Augmented Reality*, pp. 1–2. 2018.

[21] T. Rhee, L. Petikam, B. Allen, and A. Chalmers. Mr360: Mixed reality rendering for 360 panoramic videos. *IEEE transactions on visualization and computer graphics*, 23(4):1379–1388, 2017.

[22] T. Richter-Trummer, D. Kalkofen, J. Park, and D. Schmalstieg. Instant mixed reality lighting from casual scanning. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 27–36, 2016.

[23] T. W. Schubert. The sense of presence in virtual environments: A three-component scale measuring spatial presence, involvement, and realness. *Z. für Medienpsychologie*, 15(2):69–71, 2003.

[24] J. Tarko, J. Tompkin, and C. Richardt. Omnimr: Omnidirectional mixed reality with spatially-varying environment reflections from moving 360° video cameras. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1177–1178, 2019.

[25] J. Tarko, J. Tompkin, and C. Richardt. Real-time virtual object insertion for moving 360 videos. In *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*, pp. 1–9, 2019.

[26] S. Thompson, A. Chalmers, and T. Rhee. Real-time mixed reality rendering for underwater 360° videos. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 74–82. IEEE, 2019.

[27] S. Thompson, A. Chalmers, and T. Rhee. Real-time underwater caustics for mixed reality 360° videos. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1191–1192. IEEE, 2019.

[28] M. Tuceryan et al. Cubemap360: Interactive global illumination for augmented reality in dynamic environment. In *2019 SoutheastCon*, pp. 1–8. IEEE, 2019.

[29] J. Unger, J. Kronander, P. Larsson, S. Gustavson, J. Löw, and A. Ynnerman. Spatially varying image based lighting using hdr-video. *Computers & graphics*, 37(7):923–934, 2013.

[30] H. Wei, Y. Liu, G. Xing, Y. Zhang, and W. Huang. Simulating shadow interactions for outdoor augmented reality with rgbd data. *IEEE Access*, 7:75292–75304, 2019.

[31] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, pp. 28–31 Vol.2, 2004.