# MR360: Mixed Reality Rendering for 360° Panoramic Videos

Taehyun Rhee, *Member, IEEE*, Lohit Petikam, Benjamin Allen, and Andrew Chalmers

Fig. 1: Examples of MR360 applications: from the left, two examples of real-time rendering with deformable moving objects in different dynamic video backgrounds, and an example of user interaction used to manipulate composited virtual objects using HTC Vive motion controllers.

**Abstract**— This paper presents a novel immersive system called MR360 that provides interactive mixed reality (MR) experiences using a conventional low dynamic range (LDR) 360° panoramic video (360-video) shown in head mounted displays (HMDs). MR360 seamlessly composites 3D virtual objects into a live 360-video using the input panoramic video as the lighting source to illuminate the virtual objects. Image based lighting (IBL) is perceptually optimized to provide fast and believable results using the LDR 360-video as the lighting source. Regions of most salient lights in the input panoramic video are detected to optimize the number of lights used to cast perceptible shadows. Then, the areas of the detected lights adjust the penumbra of the shadow to provide realistic soft shadows. Finally, our real-time differential rendering synthesizes illumination of the virtual 3D objects into the 360-video. MR360 provides the illusion of interacting with objects in a video, which are actually 3D virtual objects seamlessly composited into the background of the 360-video. MR360 was implemented in a commercial game engine and tested using various 360-videos. Since our MR360 pipeline does not require any pre-computation, it can synthesize an interactive MR scene using a live 360-video stream while providing realistic high performance rendering suitable for HMDs.

**Index Terms**—Mixed reality rendering, image based lighting, image based shadowing, 360° panoramic video

## 1 Introduction

Head Mounted Displays (HMDs) are ideal devices for Virtual Reality (VR), providing an immersive viewing experience with a wide field of regard in stereoscopic viewing. Recent advances in hardware technology have led to consumer level HMDs. Although HMDs are often associated with 3D interaction in virtual environments (VE) such as video games, the technology also allows the opportunity for immersive video viewing experiences.

360° panoramic video (360-video) captures omni-directional views from the surrounding environment. The combination of 360-videos and HMDs brings immersive experiences that are far beyond those of conventional videos viewed on flat screens. The viewer is free to look in any direction while the view changes accordingly. It provides a strong sense of presence that is often required for immersive applications. Due to the potential benefits, the capturing devices for 360° panoramic images and videos are readily available, and as a result 360-video streaming services are now popular from websites such as YouTube and Facebook.

However, today's platforms have major limitations. While 360-videos allow for natural visuals from the real-world, interactions are limited to the head motion without rich interaction with the scene objects in the offline video. Online visualization of VEs (e.g., 3D games) can potentially support such interactivity, but often have limited visual quality, since real-time photorealistic rendering is still a challenging and active research issue in computer graphics.

Solving this problem will require combined innovation. We propose a new media and system called MR360, which combines mixed reality (MR) and 360-videos. MR360 seamlessly composites 3D virtual objects into a live 360-video using the input panoramic video as both the light source to illuminate the virtual objects and the backdrop to composite the rendered 3D virtual objects into. The new media provides viewers the illusion of interaction with objects in a 360-video.

In order to maximize the immersive experience, MR360 needs to meet the requirements for both high visual quality for seamless composition, and performance for real-time rendering. Image Based Lighting (IBL) illuminates 3D objects using High Dynamic Range (HDR) radiance maps surrounding the environment [1, 8]. Although a 360-video can be used for environment lighting, the low dynamic range (LDR) data captured from conventional 360-video cameras cannot provide sufficient dynamic range for IBL. High performance rendering is particularly important to mitigate visual discomfort in modern HMDs [43]. However, in order to support live video streams as well as interaction with virtual objects, we cannot use rendering methods that require pre-computation [19, 39]. Finally, differential rendering [7] has been used for compositing synthetic objects with realistic lighting in MR. However, this needs to be extended for a live 360-video background.

In this paper, we present a novel system called MR360 that provides


- *Taehyun Rhee is with Victoria University of Wellington. E-mail: taehyun.rhee@ecs.vuw.ac.nz*
- *Lohit Petikam is with Victoria University of Wellington. E-mail: Lohit.Petikam@ecs.vuw.ac.nz*
- *Benjamin Allen is with Victoria University of Wellington. E-mail: benjamin.allen@ecs.vuw.ac.nz*
- *Andrew Chalmers is with Victoria University of Wellington. E-mail: Andrew.Chalmers@ecs.vuw.ac.nz*


*Manuscript received 19 Sept. 2016; accepted 10 Jan. 2017. Date of publication 26 Jan. 2017; date of current version 18 Mar. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2017.2657178*

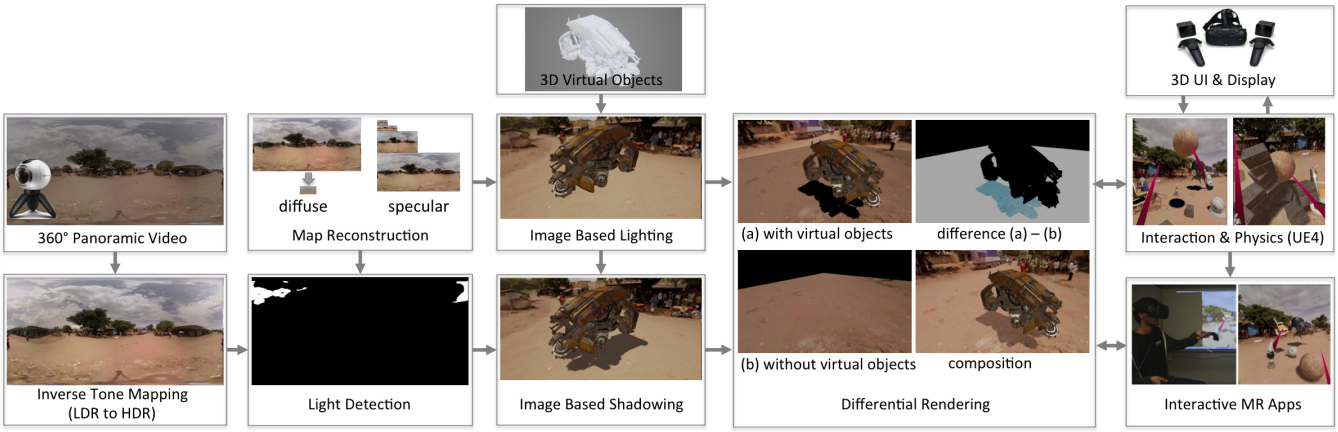


Fig. 2: MR360 system overview.

solutions to the above challenges. We optimize resources and algorithms for IBL to allow a LDR 360-video as the light source in a manner supported by recent perceptual studies [2, 4]. Then, the most salient lights are detected from the 360-video to cast only perceptible shadows for fast differential rendering. Our perceptually optimized setup can provide visually believable outputs for illumination composition of MR rendering at a high frame rate. To our knowledge, photorealistic real-time MR rendering for dynamic scenes with a live 360-video has not been fully integrated in a game engine and evaluated in previous research.

The main contributions of our paper can be summarized as follows;

- We present a novel immersive interactive system that allows user interaction with a conventional 360-video containing 3D virtual objects seamlessly composited into the panoramic video background.
- We provide realistic illumination of 3D virtual objects using a LDR 360-video as the light source. Our perceptually optimized scheme presents high visual quality and rendering speed.
- We present perceptually based thresholding for real-time image based shadowing (IBS). Salient lights are detected from the 360-video providing optimal realistic shadows for real-time differential rendering.
- MR360 has been implemented in a commercial game engine, Unreal Engine 4 (UE4), to provide a practical solution for studio artists. MR360 shows high performance suitable for modern HMDs.
- MR360 does not require any pre-computation, and therefore provides rich user interaction that is beyond the scope of conventional 360-videos; e.g. manipulating dynamic objects within a live 360-video stream in varying lighting conditions. The visual quality and users immersion have been measured by user tests using a HTC Vive headset and motion controllers.

The mixed reality setup in MR360 has the potential to provide a more photorealistic output compared to full 3D real-time rendering. Since we can focus on rendering only the composited virtual objects in MR, the computing power of the GPU can be focused specifically on those objects. The overview of the MR360 system is shown in Figure 2.

## 2 Related Work

### 2.1 Mixed Reality (MR) Rendering

Differential rendering using real-world lighting captured from HDR radiance maps has been presented by Debevec [7], and the concept has been adapted to MR rendering. Instant radiosity [20] and reflective shadow maps [26] have been utilized for real-time MR rendering using virtual point lights. GPU based ray-tracing is used for high-quality AR rendering [17, 24]. In their method, blob detection is used to estimate lights in a radiance map, which provides photorealistic output evaluated by user tests. This requires manual thresholding, which has been automated in MR360, detailed in Section 4. Ray-tracing based approaches show high visual quality but are not yet suitable for high performance rendering of complex scenes. Light propagation volumes and voxel cone tracing have been adapted to MR rendering [10, 11] and provide fast global illumination, where the median cut algorithm is used to extract light sources from a hemispherical camera image in each frame [10]. Most of the previous papers assume the real-world geometry is known or manually created. Using the RGB-D sensing camera, 3D real-world geometry can be reconstructed online [31], and the results have been adapted in MR rendering recently [13, 14, 26]. In addition to scene geometry, lighting can be estimated using inverse rendering [13, 14]. However, adapting the sophisticated light estimation step and spherical harmonics (SH) compression for real-time rendering limits visual quality with only diffuse light transportation, directional light source, and white light source in [14]. Other related works in MR rendering are feasible but we rely on further surveys from a recent survey paper [22].

### 2.2 Lighting and Sampling from Radiance Maps

Image based lighting (IBL) [7] uses captured real-world HDR radiance maps for illuminating virtual objects. Photorealistic IBL with accurate shadows is typically reserved for off-line rendering due to the high cost of sampling the radiance map. However, some methods approximate this process for real-time purposes.

Ramamoorthi et al. [36] use a spherical harmonic (SH) basis to approximate the radiance map's diffuse properties. It is shown that 3 bands can approximate the diffuse lighting [37]. The occlusion is also baked using SHs per vertex on static geometry in an offline process. At run-time, the product between a SH representation of the radiance map and the baked occlusion coefficients produces soft shadows which can be rotated in real-time. The limitation of this approach is that it cannot account for high frequency shadows.

Ng et al. [32, 33] solve this problem by using a Haar wavelet basis. However, this requires a high number of coefficients (approximately 400 for sharp shadows) as well as high resolution meshes to store the wavelet coefficients. The cost of a high resolution mesh can be mitigated by mesh LOD reduction in areas with no shadows [21], although this introduces an overhead in geometry processing which may not be suitable for real-time applications.

Debevec et al. [9, 42] use a median or variance cut algorithm to sample lights in the radiance map. However, such sampling schemes are still not a desirable approach for real-time applications, as the number of samples required is still too high. To achieve interactive shadowing

from dynamic environment maps, Supan et al. [40] use a dome of fixed shadow casting lights whose shadow strength is determined by the color of the downsampled environment map behind the light. However, their approach struggles to reproduce hard shadows and its performance does not scale well with quality.

### 2.3 360° Panoramic Video

Traditionally, VR uses 3D computer graphics to model and render virtual environments (VE) in real-time. Although recent graphics hardware supports realistic real-time rendering, creating high quality 3D VEs involves laborious and skilled tasks, and the rendering quality is still limited by real-time constraints. Image based rendering uses captured real-world images for presenting VEs [6, 12]. Panoramic images [5, 28, 41] and videos [35] have been used to present rich telepresence using collections of captured real-world images. 360° panoramic images and videos can be regarded as orientation-independent [5], since they provide information around the 360° view, which is ideal for display in HMDs. The 360-video is captured by a device with a special camera lens [30] or multiple-camera rigs [35]. In order to provide better visual quality for 360-videos, several technical problems still need to be addressed but they are beyond the scope of our paper. We rely on a survey of the details from recent related papers such as parallax free stitching [25, 34, 44] and the spherical projection [3, 25].

### 2.4 Mixed Reality Rendering with 360° Panoramic Video

MR rendering with a live 360-video is challenging since the preprocessing often required to support real-time performance such as precomputed radiance transfer (PRT) [39] is not practical for live video streams. Also, conventional LDR 360-videos do not provide the dynamic range of HDR radiance maps, which require multi-exposure images, or a special device setup for capturing. Few recent works have addressed this problem. Hajisharif et al. [15] use HDR light probe image sequences captured from a high resolution HDR camera, and use PRT [39] for IBL. Due to the required offline precomputation to calculate SH transfer coefficients, their scene is limited to static Lambertian objects. Kronander et al. [23] use an additional HDR video camera mounted underneath the primary LDR video camera shooting the scene. This HDR video camera records the environment via an attached light probe for IBL. Michiels et al. [29] directly uses LDR 360-videos as the light source for their IBL based on spherical radial basis functions (SRBF). Due to the limited radiance values from the LDR environment map, their final rendering quality is limited. Also, the required precomputation for SRBF makes it difficult to support live video streams. Recent work by Iorns and Rhee [16] relies on perceptual studies [2, 4] showing that proper inverse tone-mapping can reconstruct the dynamic range of the LDR radiance map to make the final IBL result more believable for human viewers. Their perceptually optimized scheme provides visually promising IBL at a high frame rate using a LDR 360-video. Since their method does not require any pre-computation, it supports live 360-video streams as dynamic light sources. We adapt their method to MR360 for real-time IBL.

Based on our survey, none of the above works addressed realistic dynamic shadows from live 360-videos without precomputation, which is required for illumination composition of virtual objects into the 360-video background. To the best of our knowledge, this is the first paper to present a complete pipeline and practical solutions for realistic MR rendering for live LDR 360-video streams using the video as both the light source and backdrop for differential rendering.

## 3 Real-time IBL using 360-video

IBL supports photorealistic rendering of virtual objects using real-world lighting. For high-performance IBL from LDR 360-video, we adopt perceptually based rendering schemes [4, 16].

### 3.1 Perceptually Based IBL

In MR360, input LDR 360-video streams are used as dynamic real-world radiance maps to illuminate virtual objects. Since lighting and materials are reciprocal in illumination [38], the radiance map has been reconstructed to represent illumination of diffuse, glossy, and mirror-like specular materials.

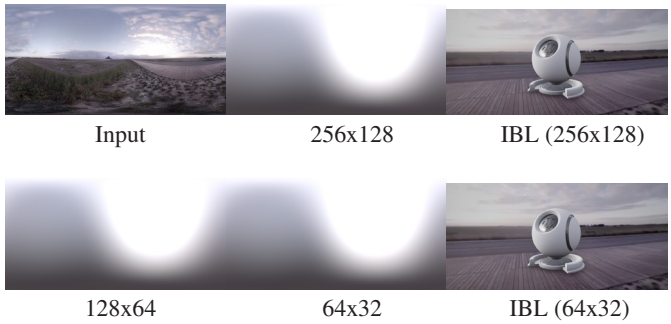


Fig. 3: Comparison of IBL rendering using diffuse radiance maps of various resolutions

For diffuse illumination, a diffuse radiance map is generated per frame. When using the perceptually optimized radiance map, reducing the resolution for a tiny radiance map (e.g. $32 \times 16$ pixels) [16], we optimize computation costs to generate the diffuse map per frame while maintaining perceptible visual quality in IBL (see Figure 3). Once the diffuse radiance map for each frame is generated, the diffuse lighting for any point on an object's surface consists simply of a single texture lookup.

Mirror-like specular reflection can be simply achieved by environment mapping, but glossy specular reflection is computationally expensive. We can approximate glossy specular reflection with mipmaps in a similar manner to our diffuse illumination. The sampling radius around the specular direction depends on the surface roughness parameter related to the glossy lobe. In a radius around the specular direction, we can sample a mipmap chain of specular radiance maps at an appropriate level, for example, a higher resolution mipmap level is sampled for lower roughness. Glossy specular lighting can be approximated using a fixed number of texture lookups per rendering fragment. As guided by [16], 18 samples can be used inside the primary glossy lobe, and 18 samples outside it.

In our GPU implementation, every 360-video frame is convolved by a GPU shader, which outputs a diffuse radiance map and multiple specular radiance maps to multiple render targets. The radiance maps are then applied to IBL for virtual objects by sampling the radiance map render targets corresponding to proper material properties as shown in Figure 4.

Fig. 4: Glossy specular reflection with various roughness values: 0.0, 0.25, 0.5, 0.75, 1.0.

### 3.2 Inverse Tone Mapping from LDR to HDR

IBL requires HDR radiance maps as the input for realistic illumination [7]. Missing radiance values in LDR radiance maps cannot produce believable lighting in IBL as shown in Figure 5 (a). Recent studies [2, 4, 16] show that proper inverse tone mapping can boost the radiance values of LDR radiance maps to provide believable IBL results targeting the human visual system. We adopt the inverse tone mapping from [16] such that the HDR color is calculated from the LDR color as:

$$\langle R_o, G_o, B_o \rangle = k \cdot \langle R_i, G_i, B_i \rangle \quad (1)$$

$$k = 10 \cdot Y_i^{10} + 1.8 \quad (2)$$

The input luminance is calculated as:

$$Y_i = 0.3 \cdot R_i + 0.59 \cdot G_i + 0.11 \cdot B_i \quad (3)$$

The tone-mapping operator we use is independent of varying frame properties, as the same transform is applied to each pixel individually. As such it is easily and efficiently implemented on the GPU.

(a) (b) (c)

Fig. 5: Comparison of lighting using a (a) LDR radiance map, (b) LDR-HDR tonemapped radiance map, and (c) HDR (ground truth) radiance map.

### 3.3 IBL in different materials

Our IBL supports real-time rendering for various materials covering diffuse, glossy, and mirror-like specular reflection. Artists and developers can easily tweak the appearance of existing assets because our IBL implementation was designed to match UE4's existing shading model as shown in Figure 6.

Our reconstructed LDR 360-videos using inverse tone-mapping provide believable IBL when viewed independently as shown in Figure 6. However, it cannot match the HDR lighting result perfectly, especially for very high contrast scenes such as the scene in Figure 5 (b) and (c).

Fig. 6: Rendering of real-time IBL integrated with UE4's shading model in different 360-videos.

## 4 Real-Time IBS using 360-video

### 4.1 Light Detection

Real-time IBL can provide realistic illumination but has limitations for casting realistic shadows. Artists in visual effects studios often add artificial lights, such as a directional light, on top of the radiance map. We automated the process in MR360 for real-time image based shadowing (IBS).

Given a radiance map, we aim to detect prominent patches of pixels and convert them into directional lights. Standard shadow mapping can then be used to cast shadows from the detected lights in dynamic scenes. To allow for dynamic radiance maps we require a solution which runs in real-time. To achieve this we use a thresholding approach. Given a threshold value, if a pixel's luminance (after inverse tone-mapping) is above the threshold, it is considered part of a light source. This produces a mask (Figure 7b) for which pixels belong to light sources. In order to produce a minimal set of discrete directional lights, we perform a breadth-first search on this mask to determine connectivity. We subtract the threshold luminance from each pixel before computing the light properties and clamp the radiance map to the threshold luminance in order to ensure energy conservation.

For each detected light, we determine its properties from the threshold luminance $Y_t$ and the pixels that belong to it in terms of their luminance $Y_p$, radiance $L_p$, solid angle $\Omega_p$ and spherical coordinates $\langle \theta_p, \phi_p \rangle$. Each pixel's contribution to the light's irradiance is determined from its solid angle and the amount of its radiance over the luminance threshold:

$$E_p = \Omega_p \cdot \left( L_p - L_p \cdot \min\left(1, \frac{Y_t}{Y_p}\right)\right) \quad (4)$$

The light's irradiance is then $E_l = \sum_{pixels} E_p$. The light's position in spherical coordinates is determined by a weighted mean where $Y(E)$ is as for equation 3:

$$\langle \theta_l, \phi_l \rangle = \frac{1}{Y(E_l)} \sum_{pixels} Y(E_p) \cdot \langle \theta_p, \phi_p \rangle \quad (5)$$

This approach is simple and can run in real-time. However, a couple of problems need to be addressed. Firstly, the actual threshold is not obvious, and changes based on the radiance map. For example, a radiance map with a strong dominant light requires a high threshold value, but a radiance map with diffuse light sources requires a lower threshold value. This is particularly problematic for dynamic radiance maps, where the threshold value needs to change in real-time based on the current frame.

To calculate the dynamic threshold, we specify it statically in terms of the distribution of luminance values. At run-time, we can easily and quickly calculate the mean $\mu$ and variance $\sigma^2$ of each frame's luminance. We seek to specify the threshold in terms of these simple statistics to ensure real-time performance. Through an error minimization process and verification from a perceptual user study, we determined that a suitable statistical threshold is:

$$\mu + 2\sigma \quad (6)$$

See Section 4.2 for more details.

Another problem is that a threshold value produces noisy patch areas, rather than a minimal number of contiguous patches. We found that applying a small box blur to the luminance data before thresholding successfully dealt with the noise, thus producing much cleaner patches. We subsequently found that our light detection often produced a small number of strong lights together with a number of very weak lights whose luminance was only slightly over the threshold. To solve this, we sort the lights in order of descending strength, and then keep only the first $n$ lights for the smallest value of $n$ such that their combined irradiance is at least 90% of that from all detected lights.

We encountered several scenarios where a strong ground reflection was erroneously detected as a light, potentially at the expense of a real light. As we are primarily interested in lighting from the upper hemisphere to cast shadows, we use only the upper half of the frame to combat this.

A common problem is that, due to the video being LDR in nature, the lights are clipped to plain white. While inverse tone-mapping significantly improves this scenario, there is no a priori way to determine how bright the light was before clipping. This presents a problem when it comes to matching the strength of shadows visible in the video. We observe that small light patches are more likely to be proper lights, while larger patches may be clouds or reflections. We apply a greater brightness increase to lights of smaller solid angles and reduce the brightness of those with very large solid angles with an experimentally determined ad-hoc mapping.

As we determine the solid angle of the light sources, we can use this to control shadow filtering (with a technique such as the well-known percentage closer filtering) to achieve perceptually acceptable artificial soft shadow penumbrae that vary dynamically with the environment. See Section 6.3 for details.

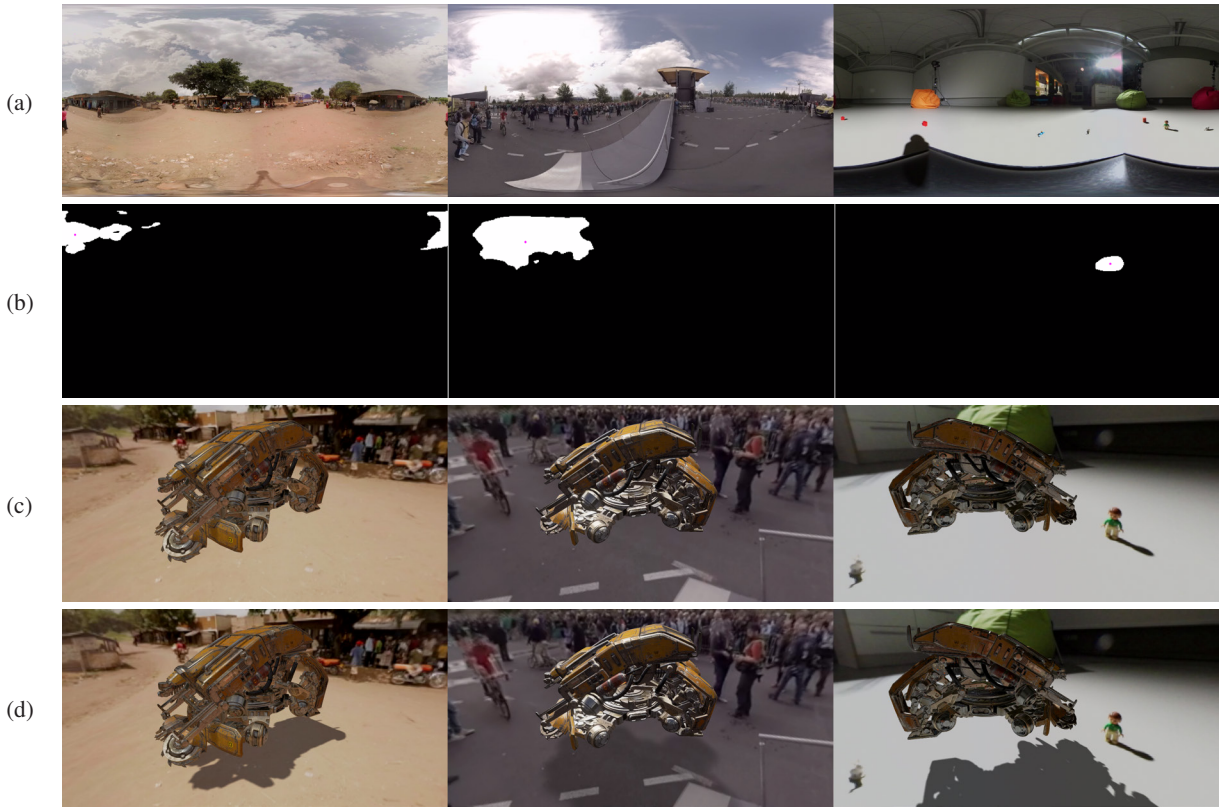


Fig. 7: Light detection. (a) Input environment maps, (b) light detection masks, (c) rendered results with only IBL, and (d) rendered results with IBS. The pink dots on the masks indicate the positions of the detected lights.

### 4.2 Statistical Threshold

While thresholding allows for real-time light detection, the threshold value needs to be determined. We use an optimization strategy to find a statistical threshold, and conduct a user study showing that it has comparable quality to a recent offline light detection algorithm.

**Determining a Threshold:** In order to determine a threshold value, we use an optimization between thresholding and Karsch et al.'s recent approach for light detection in LDR images [18]. Their method produces a binary mask of the light pixels in a given radiance map. From this, we use a binary search to adjust a threshold value such that a weighted sum of the luminance values in the threshold detected region of the radiance map is similar to that produced by Karsch's mask, where the error metric is:

$$\left| \left( \sum_{pixels_t} \Omega_p \cdot Y_p \right) - \left( \sum_{pixels_k} \Omega_p \cdot Y_p \right) \right| \tag{7}$$

$Y_p$ is the pixel luminance and $pixels_t$ and $pixels_k$ respectively denote the sets of pixels provided by the threshold mask and Karsch's mask.

In order to find the statistical rule which is consistent among various lighting conditions, we run the optimization over three broad categories of radiance maps: single hard light, single soft light, and multiple lights. These categories correspond to sunny, overcast, and indoor scenes as described in previous work [4]. In each of the three categories, we have three exemplars, giving nine test cases in total. Based on our experiment, we have found that a threshold value of approximately 2 standard deviations above the mean captures the necessary pixels for lighting (Equation 6).

**Perceptual Evaluation:** We conduct a perceptual user study to show that the statistical threshold (Equation 6) is a valid approach to light detection. The nine exemplars used in our optimization are HDR radiance maps which were converted to LDR. This allows us to run the algorithm on LDR radiance maps, and evaluate the performance by comparing with the ground truth HDR radiance map.

Using the LDR radiance maps, we compute Karsch's masks as well as our optimized threshold masks. For each active pixel in a given mask, we take the value from the ground truth HDR radiance map. For each inactive pixel in a mask, we set it as an approximated constant ambient value. This in turn creates a radiance map with only the ground truth light detected pixels, and a constant value for every other pixel. See Figure 9 to see examples of the radiance maps.

Using the ground truth HDR radiance map, as well as the two light detected radiance maps, we render scenes in which users evaluate whether or not the rendered scenes using the light detected radiance maps match the rendered scenes using the ground truth HDR radiance map. To capture the shadow details in a single rendered image, we place a cylinder on a plane, and render a top down orthographic view of the scene. See Figure 9 for example renderings from the study. We conduct a survey in which 50 participants scored on a Likert scale the similarity of the rendered image to the ground truth scene. They were asked to rate how similar the *shadow detail* is between the scene rendered by Karsch's and the thresholding method to the ground truth image. The Likert scale is a 7 point scale of similarity: *-3, 'Completely different'*, *-2, 'Very dissimilar'*, *-1, 'Dissimilar'*, *0, 'Neutral'*, *1, 'Similar'*, *2, 'Very similar'*, and *3, 'Looks the same'*. The user study shows that our statistical thresholding is comparable to Karsch's method, and that we maintain visual quality above neutral similarity. The results (Figure 8) show that in all cases, thresholding maintains quality above neutral similarity. While Karsch's method failed in some cases, particularly with complex indoor lighting scenes, our method still maintained good quality in most cases. Furthermore, in the fail cases, thresholding still generalized enough to allow for good lighting conditions.

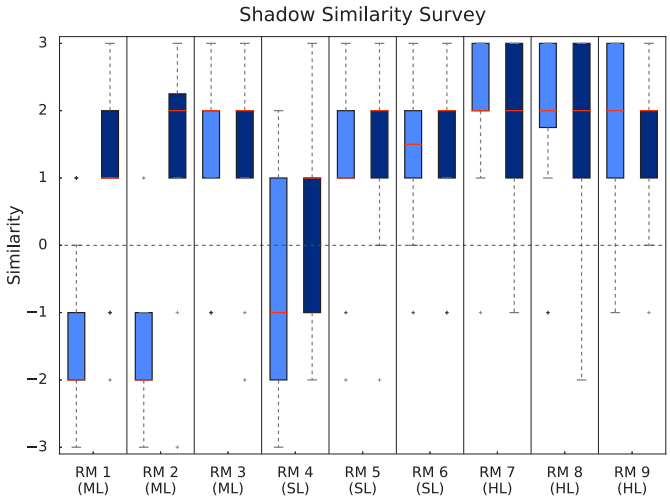


Fig. 8: User study results comparing Karsch et al.'s light detection (light blue bars) and our method (dark blue bars), where ML are the radiance maps (RM) with multiple lights, SL has one soft light, and HL has one hard light.

## 5 Differential Rendering for 360-video

In typical 360-video viewing scenarios using a HMD, user head rotations tracked by sensors will be mapped to a virtual camera centered at the spherical coordinates of the 360-video. Then the virtual camera will project part of the 360-video into the HMD screen as shown in Figure 10. It provides the immersive illusion that the users view is navigating the surrounding 360° panoramic environment. In general capturing setups for 360-videos, the camera is located at the origin of the spherical coordinates mapping the panoramic scene. If the camera is in a fixed position, or the camera and virtual objects move together while maintaining relative position between them, the virtual objects will be in the same position at the camera coordinates in every frame of the 360-video. Furthermore, since the HMD viewer's position will be at the camera origin when viewing the 360-video, the position of the virtual objects will be the same in every frame when rotating the head. This means that sophisticated camera tracking is not actually required for differential rendering [7] with 360-video as the backdrop.

In MR360, the differential rendering and composition [7] is applied in a real-time context to seamlessly composite virtual objects into the 360-video background. The basic concept involves taking two renderings of objects on a local scene: one with objects, and one without. The local scene is a piece of geometry representing the compositing surface as shown in Figure 11c. The per-pixel difference between the renderings extracts the effect that the objects had on the local scene, such as shadows and reflections as shown in Figure 11e. We employ this composition method to capture shadows and screen-space reflections in real-time and apply the effects onto the 360° background. The full process is depicted in Figure 11.

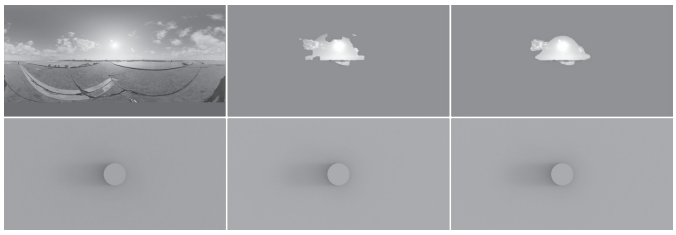

Fig. 9: User study example renderings. Top row from the left: the ground truth HDR radiance map, light detected radiance map using Karsch et al.'s method, and our thresholded radiance map. The bottom row are the corresponding rendered images used in the user study.

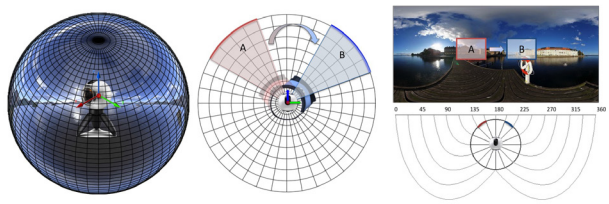


Fig. 10: View changes in 360-video using HMDs.

## 6 MR360 Implementation

### 6.1 System Setup

We tested MR360 using various monoscopic LDR 360-videos including captured videos using a Samsung Gear 360 camera, as well as downloaded videos from YouTube. The video resolution is 1280×640 at 30 FPS. We performed our experiments on a computer with an Intel Xeon 3.5 GHz CPU, 8 GB of memory, and an Nvidia Geforce GTX 970. The MR360 scenes have been tested in a HTC Vive VR headset (Vive headset) supporting display at 90 FPS with HTC Vive hand-held motion controllers (Vive controllers) for user interactions. Important rendering steps in the MR360 pipeline have been implemented by GPU shaders and integrated into the 3D game engine Unreal Engine 4 (UE4) for providing practical solutions; our light detection for IBS is running on CPU concurrently.

### 6.2 Image Based Lighting in UE4

UE4's node based material editing interface was used to build the rendering step using GPU shaders. Render-to-texture shaders compute a diffuse radiance map and multiple specular maps; we used four specular maps to cover different roughnesses. As discussed in Section 3.1, only low resolution radiance maps were computed for perceptually optimized IBL to meet the required high frame rate.

The IBL result was applied to the virtual objects using UE4's post-processing pipeline. Since UE4 uses deferred rendering, we can access per-pixel properties (e.g. diffuse color, surface normal, roughness, etc.) in the post-processing stage via G-buffers. The IBL output consists of the diffuse component (diffuse radiance map sampled by the surface normal), and the rough specular component (a weighted combination of specular radiance maps of different roughness, sampled by the surface reflection vector). Shading the IBL result in a post-process has the benefit of easy integration with pre-made assets and projects in UE4.

### 6.3 Image Based Shadowing in UE4

To facilitate using the detected lights to control UE4 scene properties and to make the implementation of breadth-first search for light detection (as in Section 4) easier, we implemented the light detection process entirely on the CPU. To achieve real-time performance, we resample the input frame to a fixed resolution of 600 × 150 (top half only) before performing light detection.

We modified UE4's *Media Framework* to get access to the raw video frame before it is uploaded to the GPU. Our light detection is run whenever the video decoder thread delivers a new frame to its target, a *Media Texture* object. As such, the light detection just has to keep up with the video frame rate and does not affect rendering performance. Our implementation is not fully optimized, but runs in real-time within our system setup detailed in Section 6.1. We added several new properties to the Media Texture to control the light detection and to export the results such that they can be used by a UE4 *Blueprint* to update the directional lights in the scene. The detected light's spherical coordinates are used to set the direction of the UE4 light. The detected irradiance is used to set the intensity and color of the light.

UE4's dynamic shadowing for directional lights is implemented with cascaded shadow maps [27] and percentage closer filtering (PCF). By carefully adjusting the shadow parameters exposed by UE4, it is possible to use its PCF to achieve the effect of soft shadows. By

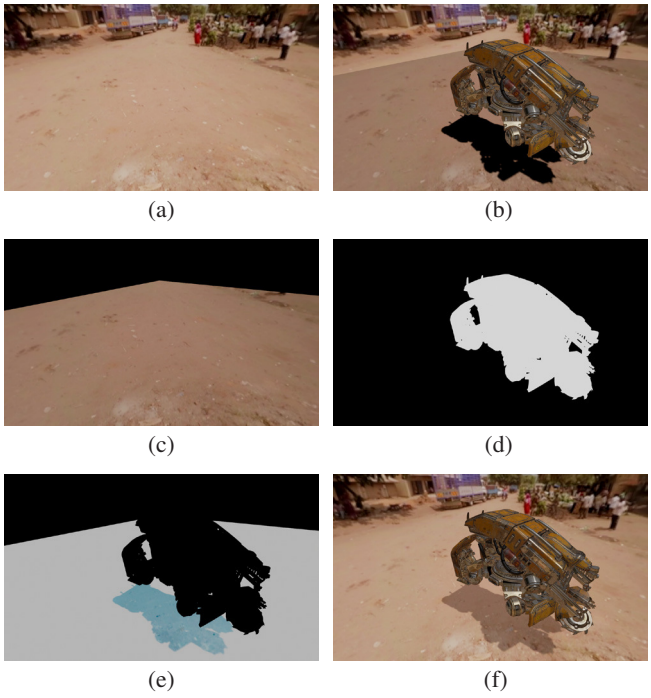


Fig. 11: Compositing steps for real-time differential rendering. (a) 360 background (user's view), (b) Background, objects and local scene, (c) Local scene without objects, (d) Object matte (stencil buffer), (e) Difference: (b)×(1-(d)) - (c), (f) Final Composite

controlling the dynamic shadow distance parameter, we can effectively control the spatial resolution of the shadow map. A lower spatial resolution causes the soft penumbrae produced by PCF to increase in size.

We linearly map and clamp light solid angles of $[0.1sr, 0.6sr]$ to shadow distances of $[4000cm, 20000cm]$ by experiments. We linearly map and clamp the same range of solid angles to cascade distribution exponents of $[1, 3]$ in order to increase shadow detail for smaller lights. We also reduce the intensity for larger lights which helps to reduce artifacts from the lower shadow resolution. We use two cascades to ensure sufficient detail.

### 6.4 Differential rendering in UE4

Differential rendering [7] is modified for high performance real-time MR rendering in UE4. We separately address diffuse and specular local scene reflectance for differential rendering. A diffuse local scene captures shadows and screen-space ambient occlusion (SSAO) from the virtual objects, and a specular local scene captures screen-space reflections (SSR). Debevec uses an iterative approach for estimating the local scene BRDF [7]. In order to meet the tight frame-time budget, we instead directly use the background video as the diffuse color of the local scene. This grants consistent composition without manual tweaking, and it has the added benefit of automatically adapting to spatial color variation in the compositing region of the video.

As explained in Section 5, the process requires two renderings. The first is of the entire scene including objects and local scene, and the second is of the local scene without objects. Since we only consider diffuse BRDF for the local scene, the Lambertian color value can be easily computed in a post process. Hence, we take the difference in the compositing stage, without taking a second rendering of the scene (local scene without objects). This means we can eliminate the high cost of taking a second rendering per frame.

## 7 Results

### 7.1 MR360 Visual Quality Evaluation

We have tested five different MR360 scenes to evaluate the visual quality. Three studio scenes have been created using static 360-images captured in a studio by the Gear 360 camera, and two dynamic scenes with 360-video downloaded from YouTube. The 360-video of the *'Studio 1'* scene has a direct specular light from a LED light, the *'Studio 2'* scene has a diffuse light bounced from a wall from indirect dimmed LED lights, and the *'Studio 3'* scene has mixture of two strong specular lights and a dimmed diffuse light as in the Studio 2 scene. *'Paris'* and *'Timelapse'* scenes have been downloaded from YouTube. The *'Paris'* video has dimmed sunlight with small variation in the light, and the *'Timelapse'* video has strong sunlight, changing position and intensity in each frame. For each scene, four 3D virtual objects were manually composited using UE4's user interface by an unskilled person having no prior experience of realistic composition, and we rendered them using our IBL and IBS. In the studio scenes, we located tiny real world objects (e.g. around 3 to $4cm^3$) to provide intrinsic references far from the direct lighting sources. The input videos, MR360 results, and examples of the scene displayed in HMDs, are shown in Figure 13.

The aim of MR360 is to provide seamless illumination composition between 3D virtual objects and the background 360-video. Here, ***seamless*** means the human visual system cannot distinguish the difference between synthetic objects and the photographed real objects within a mixed reality scene. Since it is extremely challenging to create a 3D synthetic clone of the photographed objects, we have adapted user study method without reference images [4] to evaluate the visual quality of our illumination composition. We have tested our MR360 results with 12 participants in five sessions. In each session, participants watched each MR360 scene in Figure 13 displayed in a Vive headset. We verbally asked the following questions during each session. Q1: *"Please identify the synthetic object(s) in the scene"* and Q2: *"How well does the lighting of the synthetic object(s) match with the environment?"*. After finding synthetic objects while changing their views in the panoramic MR scenes, participants scored the lighting similarity matching of each object on a Likert scale from -2 (*'very dissimilar'*) to 2 (*'very similar'*). If participants could not find a synthetic object, we scored the match of the object as '2'.

The results of our user evaluation test are shown in Figure 12. Since real-world objects captured in the video provide intrinsic references to compare illumination quality, the test actually compares quality between synthetic rendering and photographed objects. Although it is challenging to evaluate visual quality in various scenes, mean values of all scenes show promising results above *0, 'neutral'* toward *1, 'similar' and 2, 'very similar'*. In particular, most of the virtual objects in the Studio 1 scene show almost perfect matching scores except the golden ball; none of the participants could find all the virtual objects in the Studio 1 scene. On the other hand, Studio 2 and Paris scenes, having a dimmed diffuse light source scattered in the videos, show relatively low matching scores. In these scenes, the lighting condition is challenging to create believable tone mapping and soft shadow for perfect matching with photographed backgrounds. Considering the short amount of scene editing time to generate 3D virtual objects in the test scenes by an unskilled person, the overall visual quality of MR360 is promising. Based on the results, we argue that MR360 provides an efficient framework for real-time MR applications supporting ***seamless*** illumination composition between virtual objects and the background scenes believable to human visual system.

### 7.2 MR360 System Evaluation

**Real-time Performance:** We measured computation times for the MR360 system separately for light detection and differential rendering. The *'Light detection'* time in Table 1 is CPU time running in a thread separate from GPU rendering. *'Rendering'* time in Table 1 indicates the average rendering time per frame including IBL radiance map generation, IBL shading, shadowing, and illumination composition. Light detection only occurs once every video frame, not on every GPU rendering. Rendering time is calculated during stereoscopic rendering

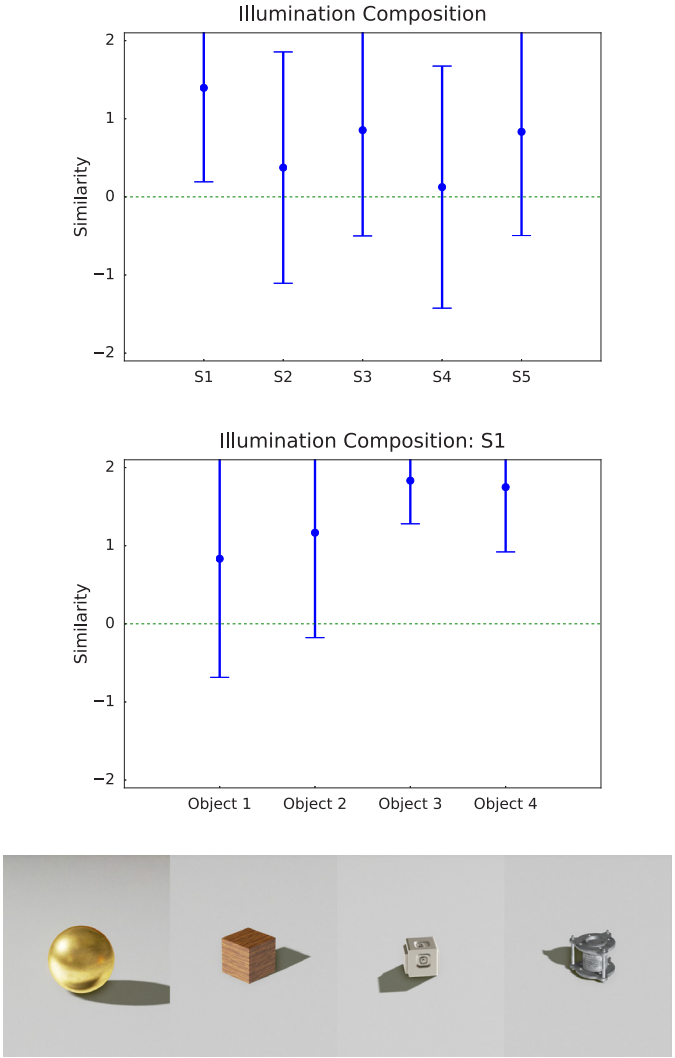


Fig. 12: User test results for evaluating visual quality of MR360 in different scenes; S1-3 are Studio 1-3, S4 is Paris, and S5 is Timelapse. The top graph is the results of average matching scores of four objects, the middle graph is the result of the Studio 1 scene for each object, and the bottom row are the four objects composited into Studio scenes (Object 1 to 4 from the left).

in VR at 80% native screen resolution on the Vive headset display. This resolution roughly equates to 1932×1073. Please note that our performance measures has been limited by Vive headset's screen refresh rate (90 Hz), meaning certain measured render times that exceed 90 FPS in MR360 are limited to the hardware refresh rate. We also note that when frame rates drop below 90 FPS, the Vive headset further limits the frame rate to 45 FPS, and takes measures such as reprojection to minimize user discomfort.

**User Test:** We have tested MR360 with human subjects to measure their immersive feeling such as sense of presence with virtual objects in the mixed reality scene as well as overall visual quality. Our test scene consists of multiple 3D virtual objects (mixture of moving and static objects) composited in a dynamic 360-video as in right two images shown in Figure 1; the 360-video, *'Africa'* has been downloaded from YouTube. We have used pre-made 3D virtual objects and their animations in UE4 in our test while implementing user interaction with Vive controllers. UE4's collision handling and physics engine have been used for realistic manipulation and interaction with the scene objects.

Participants change their views in the MR360 scene using a Vive headset. 12 participants performed two separate sessions with and without user interaction with Vive controllers. We verbally asked following three questions during each session. Q1: *"I feel that **I** am situated in the same environment as the background."*, Q2: *"I feel that the **virtual objects** are situated in the same environment as the background."* and Q3: *"I feel that the overall visual quality of the composition is of modern cinematic quality."*. Then, participants answered on a Likert scale from -2 (*'strongly disagree'*) to 2 (*'strongly agree'*).

Although our questions are challenging, as shown in Figure 14, overall feedback is positive; mean values are above *"0" (neutral)* toward *"1" (agree)*. The sense of presence and visual quality measure increase especially when having user interaction. No participants expressed any discomfort in terms of real-time performance, latency, and malfunction during their experiments supporting robustness of our system. Interestingly, many participants shared positive feedback in terms of visual quality of composition in MR360 compared with modern cinematic contents requiring offline rendering and intensive manual tuning by professional artists.

Table 1: MR360 System Performance

| Step | Stage1 | Stage3 | Timelapse | Africa |
|---|---|---|---|---|
| Light Detection (ms) | 21.2 | 22.1 | 23.29 | 24.4 |
| Rendering (ms) | 11.1 | 11.1 | 12.9 | 11.1 |

## 8 Conclusion

We have presented MR360 that allows real-time user interaction with 360-videos containing 3D virtual objects seamlessly composited and rendered into the video background. MR360 uses a conventional LDR 360-video as the lighting source and backdrop. Since it does not require any pre-computation, it can support real-time MR rendering for moving objects composited into the video background having dynamic light changes. MR360 has been implemented in UE4 and runs with a HTC Vive headset and controllers, and provides real-time interactions with objects situated in the video. Our user tests show that MR360 presents seamless composition in MR scenes, and provides an improved sense of presence and immersion beyond the scope of conventional 360-videos.

Although MR360 provides practical solutions for immersive MR rendering, the current version of MR360 has a few limitations. Despite believable IBL with inverse tone mapping, it has limitations compared to a true HDR setup. High intensity lights will be clipped in LDR video and captured as white. Our separate scheme to use IBL with extracted lights can minimize the limitation, but the clipped values cannot be fully reconstructed. Our ad-hoc intensity remapping is usable in most sunlit outdoor scenes, and indoor scenes with small strong lights that we tested, but has trouble with others such as very overcast skies and clipped areas from relatively low intensity sources.

The current UE4 interface to the Vive headset only supports stereoscopic rendering when using Vive controllers. It causes parallax mismatching between virtual objects rendered in stereo and the monoscopic background, and therefore the objects to appear floating on the background, rather than seamlessly situated in the 360° world. In order to mitigate such artifacts, we lower the VR world-to-meters scale to sufficiently move virtual objects away from the user (minimising parallax), and adjust the hand model scale to retain the feeling of direct interaction. Consistent rendering with either monoscopic foreground and monoscopic background, or stereo foreground and stereo background can naturally solve the problem.

Although MR360 provides high frame rate rendering to support modern HMDs, conventional 360-video cameras cannot capture high frame rate videos. Advanced capturing devices and video encoding/decoding algorithms to adapt the high quality 360-videos will contribute to improve the overall visual quality in MR360, but it is beyond the scope of our work.

Since MR360 has been fully integrated in UE4, we can utilize its 3D modeling interfaces and tools to reconstruct 3D scene geometry

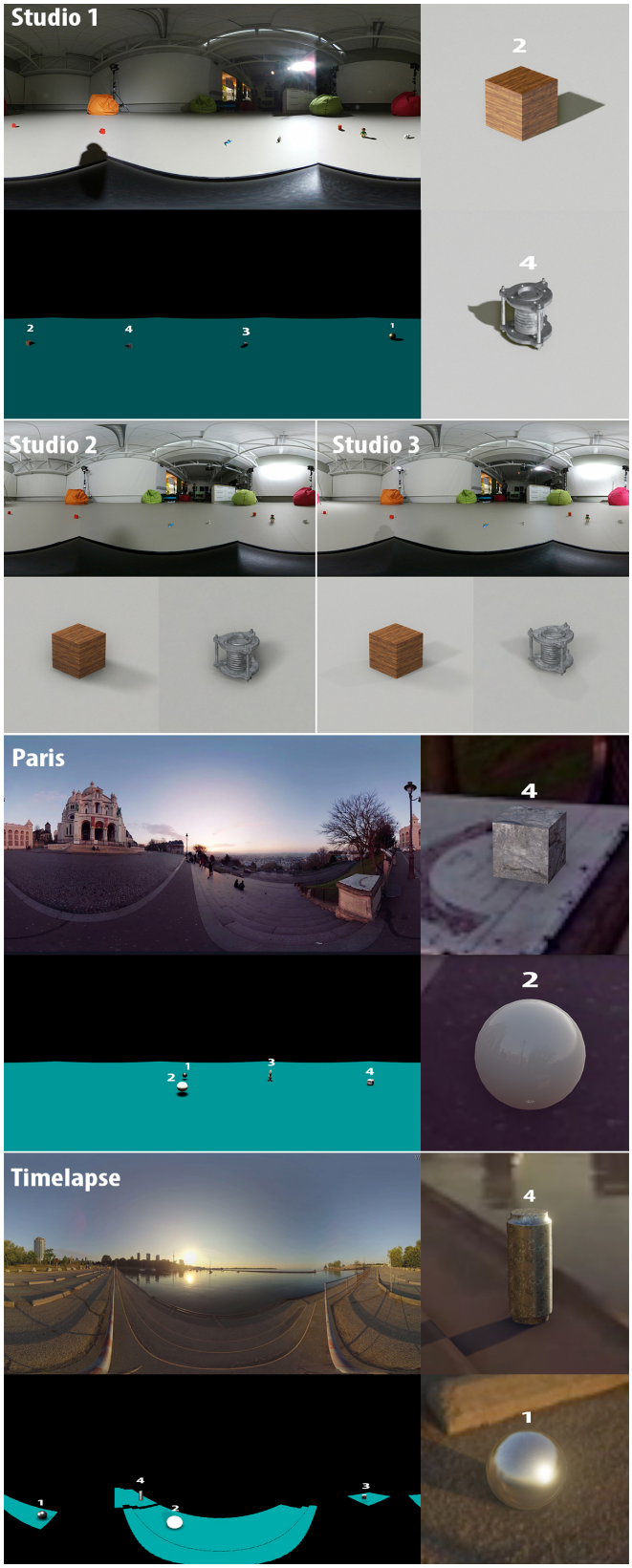


Fig. 13: MR360 test results for user evaluation; For Studio 1, Paris, and Timelapse scene, the top left is input 360-video, bottom-left is the virtual objects augmented in the scene without differential rendering, and MR360 rendering samples on the right; Studio 2 and 3 scene setups are the same as Studio 1.

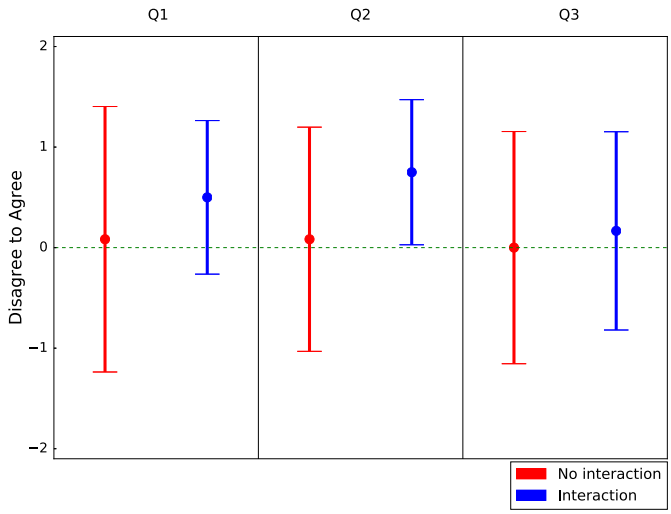


Fig. 14: User test for measuring immersion (Q1 and Q2), and overall visual quality (Q3)

of 360-videos. Therefore, we manually setup 3D scene geometry for some of our complex test scenes; otherwise we used a flat surface for the local scene. Our pipeline does not have limitations for adapting a 3D reconstruction step [14, 26], and adapting it for 360-video can be possible future work.

## Acknowledgments

This research was supported in part by the *HDI4D* project funded by *MBIE* in New Zealand and *NRF* in Korea (NRF-2014K1A3A1A17073365).

## References

[1] K. Agusanto, L. Li, Z. Chuangui, and N. W. Sing. Photorealistic rendering for augmented reality using environment illumination. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 208–216. IEEE, 2003.

[2] A. O. Akyüz, R. Fleming, B. E. Riecke, E. Reinhard, and H. H. Bülthoff. Do HDR displays support LDR content?: a psychophysical evaluation. *ACM Transactions on Graphics (TOG)*, 26(3):38, 2007.

[3] R. Carroll, M. Agrawal, and A. Agarwala. Optimizing content-preserving projections for wide-angle images. *ACM Trans. Graph.*, 28(3):43:1–43:9, July 2009.

[4] A. Chalmers, J. J. Choi, and T. Rhee. Perceptually optimised illumination for seamless composites. *Pacific Graphics, The Eurographics Association*, 2014.

[5] S. E. Chen. Quicktime vr: An image-based approach to virtual environment navigation. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 29–38, New York, NY, USA, 1995. ACM.

[6] S. E. Chen and L. Williams. View interpolation for image synthesis. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 279–288, New York, NY, USA, 1993. ACM.

[7] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 189–198, New York, NY, USA, 1998. ACM.

[8] P. Debevec. Image-based lighting. *IEEE Computer Graphics and Applications*, 22(2):26–34, 2002.

[9] P. Debevec. A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06. ACM, 2006.

[10] T. A. Franke. Delta light propagation volumes for mixed reality. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pages 125–132, Oct 2013.

[11] T. A. Franke. Delta voxel cone tracing. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 39–44, Sept 2014.

[12] N. Greene. Environment mapping and other applications of world projections. *IEEE Comput. Graph. Appl.*, 6(11):21–29, Nov. 1986.

[13] 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)*, pages 15–20. IEEE, 2014.

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

[15] S. Hajisharif, J. Kronander, E. Miandji, and J. Unger. Real-time image based lighting with streaming hdr-light probe sequences. In *SIGRAD 2012*, 2012.

[16] T. Iorns and T. Rhee. Real-time image based lighting for 360-degree panoramic video. In *Revised Selected Papers of the PSIVT 2015 Workshops on Image and Video Technology - LNCS, Volume 9555*, pages 139–151, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

[17] P. Kan and H. Kaufmann. High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, ISMAR '12, pages 99–108, Washington, DC, USA, 2012. IEEE Computer Society.

[18] 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 Trans. Graph.*, 33(3):32:1–32:15, June 2014.

[19] G. King. Real-time computation of dynamic irradiance environment maps. *GPU Gems*, 2:167–176, 2005.

[20] M. Knecht, C. Traxler, O. Mattausch, W. Purgathofer, and M. Wimmer. Differential instant radiosity for mixed reality. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 99–107, Oct 2010.

[21] J. Křivánek, S. Pattanaik, and J. Žára. Adaptive mesh subdivision for precomputed radiance transfer. In *Proceedings of the 20th spring conference on Computer graphics*, SCCG '04, pages 106–111, New York, NY, USA, 2004. ACM.

[22] J. Kronander, F. Banterle, A. Gardner, E. Miandji, and J. Unger. Photorealistic rendering of mixed reality scenes. *Comput. Graph. Forum*, 34(2):643–665, May 2015.

[23] J. Kronander, J. Dahlin, D. Jnsson, M. Kok, T. B. Schn, and J. Unger. Real-time video based lighting using gpu raytracing. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 1627–1631, Sept 2014.

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

[25] J. Lee, B. Kim, K. Kim, Y. Kim, and J. Noh. Rich360: Optimized spherical representation from structured panoramic camera arrays. *ACM Trans. Graph.*, 35(4):63:1–63:11, July 2016.

[26] P. Lensing and W. Broll. Instant indirect illumination for dynamic mixed reality scenes. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 109–118, Nov 2012.

[27] B. Lloyd, D. Tuft, S.-e. Yoon, and D. Manocha. Warping and partitioning for low error shadow maps. In *Proceedings of the Eurographics Symposium on Rendering*, pages 215–226, 2006.

[28] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 39–46, New York, NY, USA, 1995. ACM.

[29] N. Michiels, L. Jorissen, J. Put, and P. Bekaert. Interactive augmented omnidirectional video with realistic lighting. In *Augmented and Virtual Reality - First International Conference, AVR 2014, Lecce, Italy, September 17-20, 2014, Revised Selected Papers*, pages 247–263, 2014.

[30] S. K. Nayar. Catadioptric omnidirectional camera. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 482–488, Jun 1997.

[31] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[32] R. Ng. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics*, 22:376–381, 2003.

[33] R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integrals for all-frequency relighting. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 477–487. ACM, 2004.

[34] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. H. Gross. Panoramic video from unstructured camera arrays. *Comput. Graph. Forum*, 34(2):57–68, 2015.

[35] T. Pintaric, U. Neumann, and A. Rizzo. Immersive panoramic video. In *Proceedings of the 8th ACM International Conference on Multimedia*, pages 493–494, 2000.

[36] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 497–500. ACM, 2001.

[37] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. *Journal of the Optical Society of America (JOSA)*, 2001.

[38] R. Ramamoorthi and P. Hanrahan. Frequency space environment map rendering. *ACM Trans. Graph.*, 21(3):517–526, July 2002.

[39] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.*, 21(3):527–536, July 2002.

[40] P. Supan, I. Stuppacher, and M. Haller. Image based shadowing in real-time augmented reality. *IJVR*, 5(3):1–7, 2006.

[41] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 251–258, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[42] K. Viriyothai and P. Debevec. Variance minimization light probe sampling. In *SIGGRAPH '09: Posters*, SIGGRAPH '09, pages 92:1–92:1. ACM, 2009.

[43] R. Yao, T. Heath, A. Davies, T. Forsyth, N. Mitchell, and P. Hoberman. Oculus VR best practices guide. *Oculus VR*, 2014.

[44] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 3262–3269, Washington, DC, USA, 2014. IEEE Computer Society.