Real-Time Mixed Reality Rendering for Underwater 360° Videos

Stephen Thompson   Andrew Chalmers   Taehyun Rhee*

Computational Media Innovation Centre, Victoria University of Wellington, New Zealand

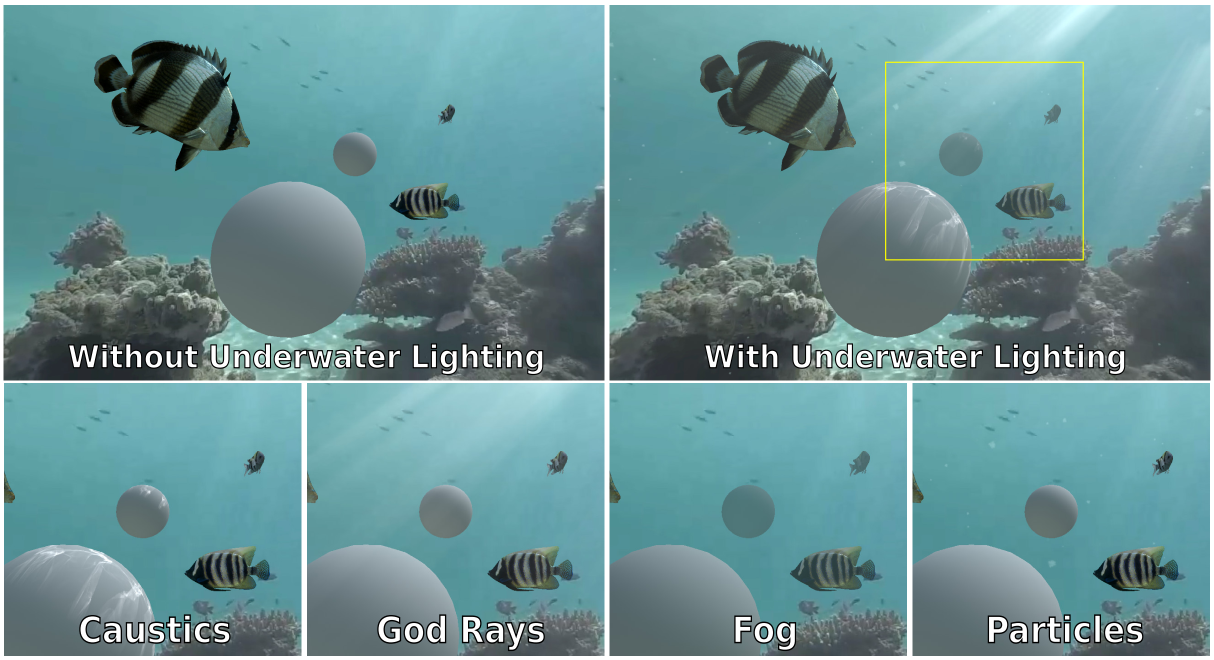


Figure 1: Real-time mixed reality underwater rendering into a 360° video. Top left is without our method, top right is with our method applying underwater lighting. The bottom row shows each of the underwater lighting effects individually. Please note that our added effects throughout this paper may appear brighter than what is in the actual 360° video. This has been done to make it easier to see the effects in the images.

Abstract

We present a novel mixed reality (MR) rendering and composition solution that illuminates and blends virtual objects into underwater 360° videos (360-video) in real-time. Real-time underwater lighting (caustics, god rays, fog, and particulates) were developed to improve the overall lighting and blending quality. We also provide a MR toolkit, an interface to tune the parameters of the underwater lighting so the user can match the lighting observed in the 360-video. Our image based lighting provides automatic ambient and high frequency underwater lighting. This ensures that the virtual objects are lit and blend similarly to each frame of the video semi-automatically and in real-time. We conducted a user study by having participants rate our method based on the visual quality and presence using a five point Likert Scale. The results show that our underwater lighting is preferred over no underwater effects or using naive ambient lighting. We also have a few takeaways on what elements of our underwater lighting and interaction have a significant impact on visual quality and presence in underwater MR.

Index Terms: Computing methodologies—Graphics systems and interfaces—Mixed / augmented reality

*e-mail: taehyun.rhee@ecs.vuw.ac.nz

1 Introduction

360° omnidirectional videos (360-video) shown in head mounted displays (HMDs) provide a wide field of regard and immersive viewing experience, giving the user a sense of presence in the surrounding scene in a video. Recent mixed reality (MR) research [29] provides real-time high-fidelity composition and seamless blending of virtual objects into the 360-video. This is done with image based lighting and shadowing that utilise the 360-video as the light source to illuminate the virtual objects as well as the background for compositing into. This allows for interactive MR experiences with virtual assets that seamlessly blend into the 360-video.

360° cameras have advanced such that people can film underwater environments. While high-fidelity blending of virtual objects with 360-video has advanced in recent research, compositing into underwater footage needs to address the additional challenges posed by the complex lighting that occurs in water. Current methods do not take into account the underwater lighting effects, resulting in rendered virtual objects that feel superimposed rather than seamlessly blended into the underwater 360-video.

Since water is a volumetric medium, light penetrating from the water surface will scatter, absorb or transmit as light rays shine through the body of water. This produces lighting effects such as fog and god-rays. The water surface itself also refracts the light, creating patterns of light below the surface called caustics. There are also particles floating through the volume. Underwater specific lighting is ignored in current solutions to 360° MR rendering, resulting in improper lighting and blending of the virtual objects for underwater

360-video. Furthermore, such underwater lighting in computer graphics are expensive, posing the additional challenge of producing a high frame rate suitable for modern HMDs (e.g., 90FPS).

We present a novel method to enhance the visual quality of underwater MR using 360-video. We developed four underwater lighting effects (caustics, god rays, fog, and particles) to improve the quality for underwater real-time rendering. Our caustic map generated by real-time water surface simulation provides believable results with a high frame rate. The caustic map is used for both projecting onto objects and for performing a raymarch to render god-rays. The fog is applied by using a down-sampled texture of the video and applying the color to the objects. Particles are then simulated to float in front of and behind the virtual objects. We also provide intuitive parameters for the user to control, allowing them to match the underwater lighting effects with the effects observed in background 360-video.

We integrated our method into commercial game engines (e.g., Unity and Unreal Engine 4 (UE4)), and we also provide a toolkit for users to semi-automatically blend virtual objects into underwater 360-videos. This allows them to interactively match the underwater effects with the background 360-video. Once setup, the underwater lighting effects will be automatically updated by the captured light in every frame of the 360-video. Our toolkit is flexible, easy to use, semi-automatic, allows for iterative parameter refinement, and does not require pre-computation.

As far as we are aware, this is the first paper to provide a high fidelity solution and toolkit to light and blend virtual objects into underwater 360-video in real-time. The main contributions of our paper are summarised as follows:

- We developed novel methods for real-time lighting for underwater mixed reality rendering, which include caustics, god rays, fog, and particles. Our method provides believable underwater lighting at high frame rates (above 120fps), suitable for modern HMDs.
- Our method is fully integrated into commercial game engines (e.g., Unity), and provides a complete pipeline for creating underwater MR contents. Our toolkit, ”Underwater Toolkit”, is a semi-automatic tool for blending virtual objects into underwater 360-video. The toolkit provides users with intuitive parameters to tweak for fine tuning and optimal composition.
- We conducted a series of user studies, evaluating the impact of our method in terms of perceivable visual quality and user presence in a HMD. Several novel observations from the user study are made, including the impact that lighting, video syncing, and interaction have on both visual quality and presence.

2 Related Work

2.1 Underwater Rendering

Underwater rendering has been a long standing research area in computer graphics, with a focus on different underwater phenomena such as refraction [16] and volumetric scattering [22], both of which are important for producing realistic imagery. Many techniques have been explored, making special considerations for real-time and offline rendering. Monte carlo path tracing is used to produce physically correct caustics [20], but is computationally expensive. This has been improved with bi-directional path tracing [21] by casting rays from the light source above the water surface, but is still reserved for offline rendering. Photon mapping [15] also casts rays from the light source, but stores the result into a photon map. This has been extended for use in real-time graphics [1, 25], but presents a challenge in finding a balance between the number of photon samples while still maintaining real-time performance. A practical implementation in real-time graphics is to render caustics to a texture (caustic map) offline [35], which is then projected onto the scene geometry at run-time to provide a the caustic intensity hitting the surface. While these techniques have been optimized for real-time graphics, they are still limited in parametric control. Prebaking the caustic map does not allow for real-time user modification and interaction, which is an important element for matching virtual caustics with the real-world 360-video.

Graphics hardware has been leveraged for approximating caustics while maintaining real-time performance [13, 14, 40]. This is done by taking a triangle mesh (specular triangles), treating each vertex as a light ray and projecting it through a refractive surface and onto diffuse surfaces. The triangle shapes (caustic triangles) created by this will approximate the appearance of how caustics naturally will appear. The intensity of each caustic triangle is calculated using the areas of the original and projected triangles. Additive blending is used to combine the resulting triangle intensities. Because intensities are calculated per triangle, the results can appear blocky. Ernst et. al. [5] suggested calculating the intensity per vertex to smooth the result without increasing the triangle count. These types of methods can be rendered to caustic maps to allow for real-time caustic map updates [31]. This is the method we adapt to render caustics.

The light rays refracted travelling through the water surface can scatter towards the viewer. The shafts of light made visible by this scattering are referred to as crepuscular rays or god rays. Like caustics, god rays can also leverage additive blending hardware to accelerate the rendering. A simple but efficient method is to render the god rays with the use of billboards or planes [34] combined with additive blending. Primitives can be used to define the boundary of the god rays, limiting expensive light intensity calculations to areas where the god rays appear. Iwasaki et al. [13] rendered the warped beams between the specular triangles and caustic triangles with additive blending to produce god rays. Liktor et al. [23] also used a similar technique to render god rays. Volumes can be rendered using ray marching, a rendering technique that works by sampling a volume at discrete steps along a ray and summing the samples. Papadopoulos et al. [25] replicated god rays by rendering the light paths as line segments using additive blending and applying a filter to reduce aliasing artifacts. Hu et al. [11] also used line segments combined with raymarching to render god rays.

Water often contains particles that reduces visibility. The three classic methods for simulating fog is to interpolate objects to a color using either a linear falloff or exponential falloff. More recently, methods have been created to use ray-marching to create fog effects [10, 41]. Although this method is slower, it does produce more physically accurate results.

2.2 Mixed Reality Rendering

Realistic lighting and shadows help to make composited objects feel grounded in the image. Debevec et al. [4] used differential rendering to render virtual objects into photos. Gutierrez et al. produced a method to cast caustics from a virtual object onto real world geometry [7]. Kan et al. [17] implemented various lighting effects and performed a user study testing how much of a visual improvement each effect made, finding that anti-aliasing, reflections and refractions provided the largest visual quality improvement, with caustics and depth of field providing a positive improvement as well.

While traditional VR experiences use a fully virtual environment, another option is the use of 360° panoramic images [2] and videos [26]. This method has the added benefit of obtaining photorealistic environments without expensive computational requirements. This type of format can be captured using either 360° cameras or by using an array of cameras and stitching together the result.

High dynamic range (HDR) 360° images can be used to provide image based lighting [4] (IBL) to virtual objects. This type of lighting uses 360° images or videos made from a collection of 360° images captured at different exposures to produce an HDR image providing a mapping between the surface normal and the light hitting it. Because of the presence brought by 360-video, this is

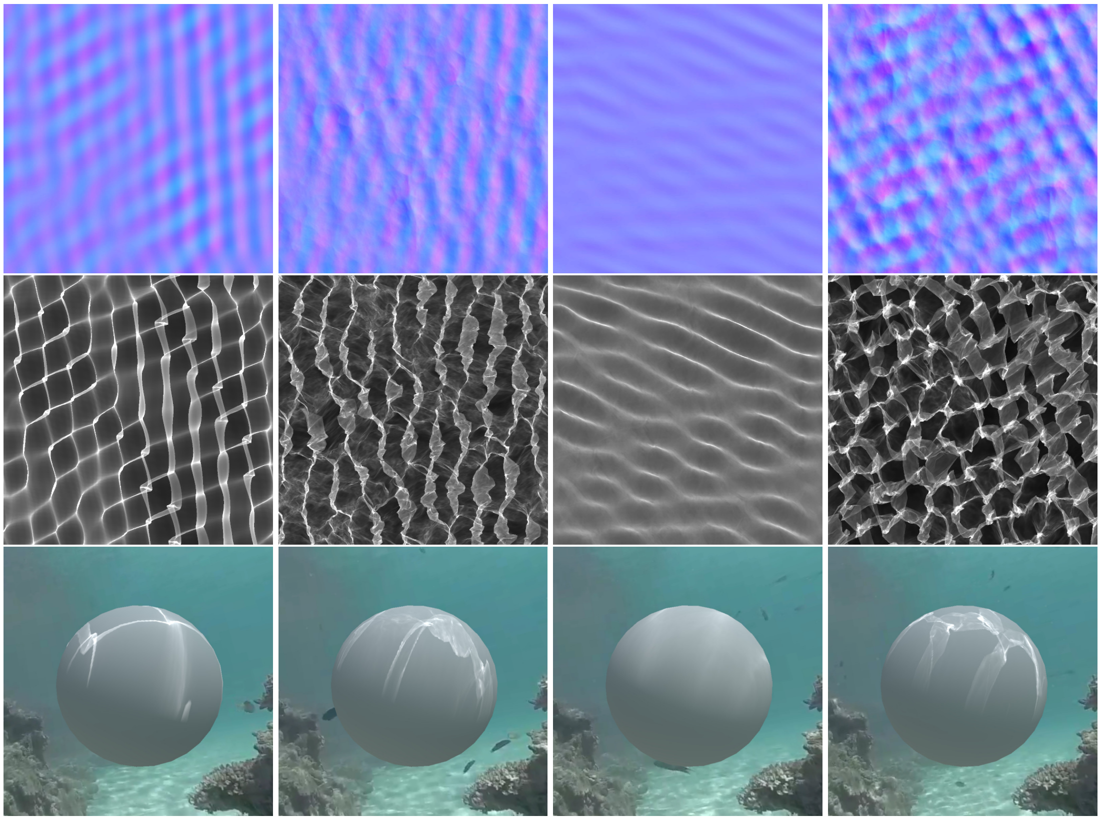


Figure 2: Water surface normal maps with corresponding caustic map and virtual object rendered with caustics.

ideal for lighting objects. However, capturing HDR required for IBL is challenging for video because most 360° cameras only capture video in low dynamic range (LDR). Iorns and Rhee [12] found that perceptually accurate HDR images can be recovered from LDR images. With this method, they were able to recreate convincing ambient light for virtual objects augmented into 360-video.

Spherical harmonics [27,33] can approximate the HDR IBL using as little as three bands of spherical harmonics [28] to represent low frequency ambient light, though it cannot account for high frequency lighting. Different solutions have been proposed [19, 24], however, these methods are too costly for real-time computation.

MR360 [29] provides a framework for real-time image based lighting and image based shadows from 360-video, accounting for high and low frequency lighting. The image based lighting provides ambient light but, because of limitations of real-time rendering, cannot cast shadows. To solve this problem lights in the video are automatically detected, allowing for high frequency lighting and believable shadows to be rendered. All computations occur in real-time. More recent work takes advantage of moving 360-video to provide localized IBL and geometry reconstruction [36].

Previous work related to the under water MR is limited. Few approached to waterproof HMD, allowing for literal underwater scuba diving in a virtual environment [3]. Tawara and Ono created a MR water surface capable of rendering caustics using the caustic triangle technique [37]. Based on our survey, our research can be the first work providing real-time solutions for high fidelity under water lighting and blending of virtual objects into an underwater 360-video.

3 Underwater MR Rendering

3.1 Lighting and Composition

Previous work by Rhee et al. [29] achieved seamless blending of virtual objects into 360-video by simulating ambient light with IBL, detecting the light sources in the video to provide high frequency lighting, and composition with shadows using differential rendering. We adapt this by, instead of detecting many high frequency lights, assuming the main light source underwater is the sun and only detecting one light source (Figure 3). IBL requires high dynamic range (HDR) videos to light objects realistically, however conventional 360 video is captured in low dynamic range (LDR). We use the work of Iorns et al. [12] to apply inverse tone mapping to the videos to convert LDR to HDR. However, we only apply the inverse tone mapping to the top half of the video to reduce flickering in the IBL caused by caustics on the ocean floor becoming over exposed.

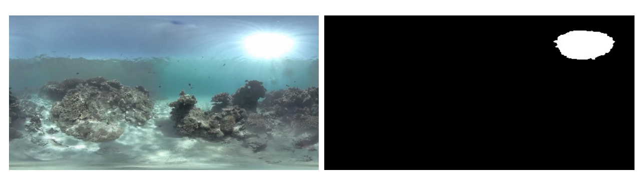


Figure 3: A frame from an underwater 360-video (left) and the detected light source (right).

3.2 Underwater Lighting

We integrate four underwater effects on top of the IBL and light detection to improve the blending of virtual objects. Caustics and fog are added to help blend objects into the 360-video. Other effects

Figure 4: God rays: The original background video with a virtual fish (left), god rays added without using light detection (middle), and god rays added using the detected light direction (right).

such as god-rays and particles may also be present in the video. Because the video is pre-recorded and lacks depth, objects cannot directly interact with these elements. To help improve the sense of blending, we render our own god-rays and particles, matching the appearance with the video so to be seen as part of the video, allowing for the virtual objects to appear to move in front of or behind elements in the video.

Water Surface Simulation and Caustics: Rendering caustics require a surface to scatter the light and generate the caustic patterns. Without knowing the water surface in the 360-video, we must either try to detect it or simulate our own virtual water surface to closely match what is in the video. The water surface is very difficult to detect from images because of refraction, reflections, over-exposure, or volumetric factors such as fog occluding the surface. We simulate a virtual water surface to avoid these problems while also providing an intuitive and interactive interface for underwater lighting, such as caustics, through water surface parameters.

Two common methods for real-time water surface simulation are Gerstner [6] or FFT [38] waves. We chose Gerstner waves because it is efficient to calculate and easy to control individual waves. Each Gerstner wave is calculated as a summation of several individual waves. This calculation is defined as

$$P(x,y,t) = \begin{bmatrix} x + \sum(Q_i A_i \times D_i.x \times \cos(w_i D_i \cdot (x,y) + \varphi_i t)) \\ y + \sum(Q_i A_i \times D_i.y \times \cos(w_i D_i \cdot (x,y) + \varphi_i t)) \\ \sum(A_i \sin(w_i D_i \cdot (x,y) + \varphi_i t)) \end{bmatrix} \quad (1)$$

where Q, D, w, A and φ represents the individual wave sharpness, direction, wavelength, amplitude and period respectively. The inputs are the horizontal position on the plane (x and y) and time (t).

We define medium to large waves using Gerstner waves and apply two scrolling normal maps to add small wave detail, such as those caused by wind or surface tension. A subdivided plane perpendicular to a camera is displaced using Gerstner waves and the vertical displacement and normal map is rendered to a render texture in every frame.

A set of 10 Gerstner waves are generated and each wave parameter is set to roughly the same value using slight random variation. Each wave direction offset is randomly set to a value between the range of -180° to 180°. The two normal maps scroll at an offset of -45° and 45°. The actual wave directions can be calculated as *direction Offset* $\times$ *global Wave Alignment* + *global Wave Direction*. This allows users to be able to adjust the wave direction without the need to generate new wave parameters by changing a global wave alignment parameter ranging between 0 and 1. The user can define the other wave parameters by using a set of global wave parameter multipliers.

Because the caustics will be rendered using a repeating caustic texture, ideally, the water surface should tile seamlessly in order for the corresponding caustics to repeat correctly. Unless the wave parameters are carefully chosen, Gerstner waves do not tile seamlessly. We seamlessly tile the water surface by up-scaling the texture to only use 90% of the pixels. The unused pixels are wrapped to the opposite side of the texture and blended using linear interpolation. This produces seamless results as long as the wavelengths are not too large compared to the size of the texture. This will produce a non-periodic but seamlessly repeating water surface. Using the simulated water surface, we then generate a caustic map and caustics using the caustic triangle method [39].

God Rays: We render god rays by ray-marching a short distance from the camera, sampling from the previously rendered caustic map at each step and summing the result. The caustic map is looked up using the same uv calculation as used for projecting the caustics onto geometry, keeping the direction of the god rays consistent with the 360-video (Figure 4). Because ray-marching is expensive, we render the god rays at a lower screen resolution and up-scaling to the full resolution. The Henyey-Greenstein phase function [9] is used to adjust how much of the light is scattered towards the viewer.

Fog: Water normally has limited visibility as a result of small

(a) Constant fog color matching left fish.

(b) Constant fog color matching right fish.

(c) Using 360-video as fog color matching both fish.

Figure 5: Directional fog. The left and right image were taken from the same video frame but different viewing directions. (a) and (b) use a single constant fog color, showing how the color will match one view direction but not the other: (a) left is correct, right is wrong, (b) left is wrong, right is correct. (c) uses our fog to create a result where the virtual objects will match the fog color in the direction of the 360-video.

particles in the water and the scattering and absorption of light in water. A naive way of simulating fog would be to use a depth based fog with objects fading to a single color. What we found was the color of water would often change depending on the direction being viewed, and using a single color may appear too bright or dark in certain directions. For example, a coastline may cast a shadow onto the water to produce a 360-video with lighter colored water in direct sunlight and darker colored water in shadow (such is the case in Figure 5). As mentioned earlier using ray-marching can produce better physically accurate fog. However, without knowing the geometry in the scene this task will become difficult to determine whether the water in front of the virtual objects should be in shadow or not.

Our solution is to use a linear based fog that used the 360 video as the fog color. Fading objects to the full resolution 360 video will make the objects appear transparent so we down scale the frame to a 16x8 resolution image, chosen by experimenting with resolutions low enough to not make objects appear transparent while still capturing view dependent color (Figure 5). Objects underwater will appear as silhouettes before completely disappearing into the fog. Instead of having objects fade away completely, we fade objects to appear as silhouettes. When we generate the fog texture we darken the texture with

$$fogcolor = \frac{\tanh([R_i, G_i, B_i])}{s} \tag{2}$$

where we use $s = 1.7$ for the darkening scalar value. This equation will maintain lower intensities and darken higher intensities. Larger values for s will darken the fog.

The area in the 360 video where sunlight enters the water usually appears over-exposed. Objects moving in front of this area when rendered with the image based fog color will appear gray and look unrealistic. We solve this problem by filling in areas of pixels above an intensity threshold (between 0 and 1) with the color from the bounding pixels. We used a threshold of 0.5. This results in more natural appearing color on the objects (Figure 6).

Figure 6: Fog containing over-exposed areas (left), and our improved fog with over-exposed areas filled in (right).

Particulates: Water can contain small floating particles, such as sand, small bits of seaweed, etc. We render a few slowly moving particles on top of the video to provide a better blending. The particles are simulated using the game engine's particle system and will fade in and out over time. The particles are rendered using additive blending and are lit using IBL.

4 UNDERWATER TOOLKIT

We developed the Underwater Toolkit, which provides a pipeline to semi-automatically blend virtual objects into underwater 360-videos. The underwater lighting is computed in real-time, allowing users to interactively adjust parameters while observing the changes. During video playback, the user can rapidly adjust the underwater lighting to match the background video. Once the parameters are set, the Underwater Toolkit will automatically update the lighting conditions for each frame.

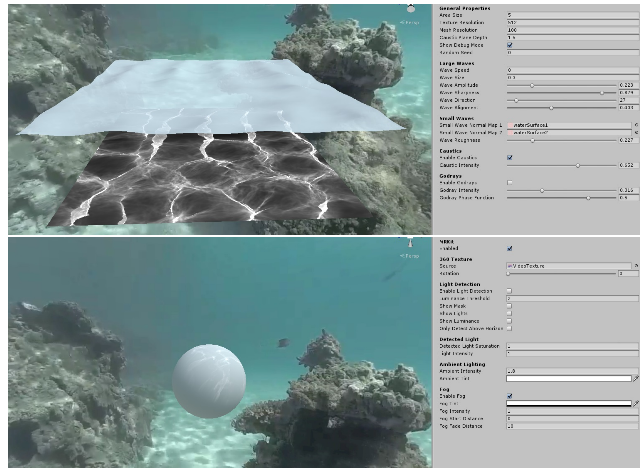


Figure 7: User interface for setting the Underwater Toolkit parameters.

The parameters are separated into two categories: The general lighting and composition by MR360 [29] and the Underwater Toolkit (Figure 7) for underwater lighting. We provide parameters for adjusting ambient light intensity, directional light intensity and color, light detection and fog. Also, provides parameters for adjusting the simulated water surface used for generating the caustics and god rays. The only fog parameters that need to be set by the user are the start and end distances. The rest of the fog implementation is automatically calculated each frame.

The water surface used for generating caustics and god rays is made from multiple waves, which can each be defined separately. For easy user control, we provide a set of global parameters. The water surface and corresponding caustic map can be viewed through the Unity scene view, providing easier matching with the 360-video. Most parameters are either in a range of 0-1 or set in world space scale.

The intensity for the caustics and god rays are kept separate to enable more artist control over the blending. God rays have an additional phase function parameter to define how close to the light source the god rays should scatter. We used the game engine particles system to simulate the particles.

Our toolkit has been integrated into game engines such as Unity and UE4. Figure 7 shows the Underwater Toolkit integrated into Unity, where the user can drag and drop a 360-video to set the background and lighting conditions. They can also pull in 3D virtual assets which will seamlessly blend into the 360-video, observable in the editor's preview window. A ground-plane is added to catch shadows. Our toolkit supports real-time rendering covering diffuse, glossy, and mirror-like reflections. Users can easily tweak the appearance of existing assets since our image-based lighting (IBL) implementation matches existing shading models in Unity and UE4. The integration into game engines allows for useful functionality and support from the engine, such as physics, collision detection, game logic, sound, and interactions. Furthermore, to provide convenient user interaction in the MR scene, we support various input devices including HTC Vive controllers, Oculus Touch controllers, and Leap Motion hand tracking to manipulate virtual objects (e.g., touch the fish) in the underwater MR scene.

5 USER EXPERIMENT

We performed the user study on a total of 15 people (12 male, 3 female) between the age of 20 to 60. We evaluated the Underwater Toolkit with respect to two criteria: *visual quality* and *presence*. The visual quality should be high enough to seamlessly blend the virtual

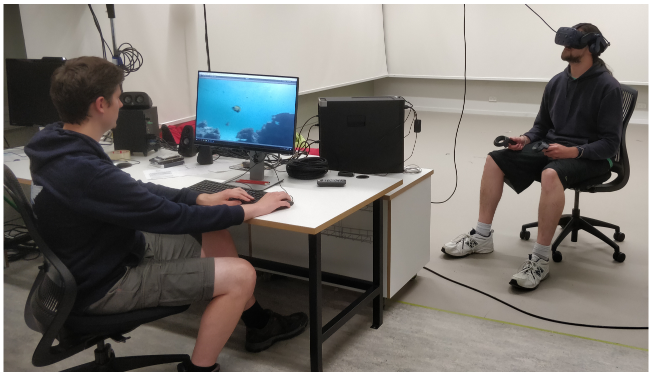


Figure 8: User study setup. The participant (right) is undergoing the study.

objects into the background video. If they are blended well, we also expect heightened presence from users as the virtual objects engage with the users. Participants were asked questions relating to the two criteria and responded using a Likert Scale. We adapted our choice of questions from existing presence [8, 30, 32] and visual quality [29] questionnaires. We compare the Underwater Toolkit with MR360 [29] and naive ambient lighting with texturing. We also investigate what impact interactivity has on presence by adding interactivity in conjunction with the Underwater Toolkit lighting.

5.1 Experiment Setup

We use a standard 22" monitor to brief users at the beginning of the study on the scene content, interface and questions. We used an NVidia GeForce GTX 1070 GPU, an Intel Xeon Processor E5-1620 v3 with 3.5 GHz clock speed, and 16 GB of RAM. During the study, users wear an HTC Vive Pro head mounted display (HMD), sitting on a swivel chair to allow for 360° rotation. During our pilot study, we found that being seated felt more immersive than standing. The HMD's cable is raised on an overhead pulley system to reduce entanglement. During the interactivity part of the study, we give users two HTC Vive controllers, one in each hand. We played ambient underwater sound effects through the HMD's speakers, as we found during our pilot studies that sound was important for improving presence. During the study, we asked questions out loud, to which users would respond. We would then write their answers down directly into a spreadsheet. We used the Unity version of the Underwater Toolkit for the study. See Figure 8 for the user study setup.

5.2 Stimuli

To fully evaluate underwater lighting, we chose three underwater 360-videos (shown in figure 9). The first video (*"Sunburst"*) contained very noticeable caustics. The second video (*"NZ"*) was filmed deeper underwater where the fog was most noticeable. The third video (*"Fiji"*) had high turbidity producing noticeable god-rays and particles. We used the Underwater Toolkit to match the real-world lighting present in the video. We placed 3D models of fish (with materials and textures) that were similar to what was found in the video. They were fully animated and swam with a semi-random motion. There were three to four virtual fish in each video scene. We illuminate, composite and interact with the fish using four categories: 1. *Naive ambient lighting*, 2. *MR360*, 3. *Underwater Toolkit*, and 4. *Underwater Toolkit + Interaction*.

5.3 Procedure

We evaluate visual quality and sense of presence by asking a 5 point Likert Scale ranging between -2 (strongly disagree) and 2 (strongly

Figure 9: The 360-videos used in the user study. Left: *"Sunburst"*, middle: *"NZ"*, right: *"Fiji"*.

agree) for each video and lighting category. Visual quality was evaluated with the following questions:

- Q1a: *"I feel the lighting of the synthetic objects matches well with the environment"*
- Q1b: *"I feel that the overall visual quality of the composition is of high quality"*

Sense of presence was evaluated with the following questions:

- Q2a: *"I feel that the virtual objects are situated in the same environment as the background"*
- Q2b: *"I feel that I am situated in the same environment as the background"*
- Q2c: *"I feel that I am situated in the same environment as the virtual objects"*

Participants were briefed on the user study procedure, outcome goals, and were able to calibrate themselves on an example scene both on a monitor and in a HMD before starting the actual user study. Once the lighting conditions had been evaluated for all three videos, participants would then interact with the fish while the lighting was set to the Underwater Toolkit. Interactions included generating food to bring the fish closer to the participant and the ability to grab the fish that were close enough. We did this for the three videos and asked the same visual quality and presence questions for each video. The order in which the videos and lighting conditions were shown to the users were randomized to reduce bias.

At the end of the study, participants were asked to rank the four water lighting effects (caustics, god-rays, fog and particles) from best to worst in terms of how well each effect improved the visual quality.

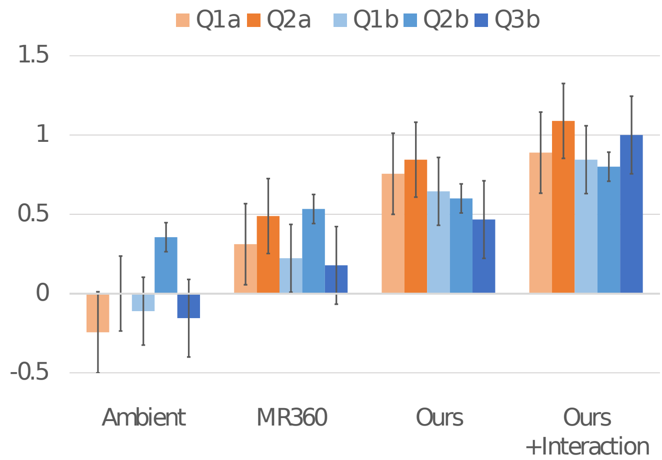
6 Result and Analysis

6.1 User Study Results

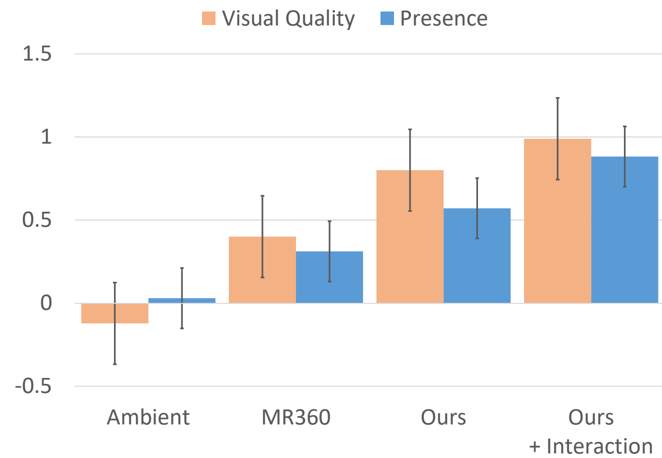
Visual Quality: Our user study demonstrates that the Underwater Toolkit has improved the overall visual quality across all three videos, followed by MR360 then naive ambient lighting. Underwater Toolkit+Interactivity produced the best result for visual quality. The results for the two individual visual quality questions are shown in Figure 10a and the average visual quality result is shown in Figure 10b (orange bars in both figures).

Presence: We also found that the Underwater Toolkit has improved the sense of presence in the 360-video, followed by MR360 and naive ambient lighting. Underwater Toolkit+Interactivity produced the best result for presence. The results for the three individual presence questions are shown in Figure 10a and the average presence result is shown in Figure 10b (blue bars in both figures).

Underwater Lighting Ranking: The final question had users rank which underwater lighting effect they preferred for each scene. We found that users strongly preferred caustics followed by fog in *Sunburst*, with a slight preference for god rays and a low preference for particles. Fog and god rays were strongly preferred in *NZ*, with a slight preference for caustics and particles. Finally, god rays, followed by fog then particles were preferred in *Fiji*, with a low preference for caustics. See Figure 11.



(a) User response to each question, averaged across three scenes. Orange is visual quality and blue is presence questions.



(b) Average of the visual quality (orange) and presence questions (blue).

Figure 10: User study results for each question (left) and the average across questions in the two categories, visual quality and presence (right).

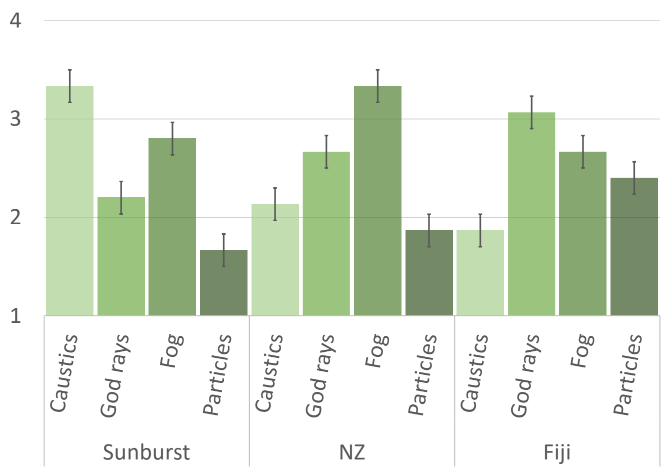


Figure 11: User ranking for each underwater lighting effect for each scene.

6.2 Analysis and Discussion

The previous MR360 method demonstrated that improving the visual quality heightened a user's sense of presence. Our study supports this idea, indicating that proper lighting and visual quality is important for improving presence. Underwater scenes pose a particularly challenging task, where there are various complex underwater lighting effects at play. We found that emulating the underwater lighting on the virtual scene was important for improving both visual quality and presence. Furthermore, adding interactivity into the user study had a noticeable impact on the overall presence. Interestingly, interactivity also improved the visual quality score, despite the fact that we did not modify the lighting algorithms. We believe that a user's sense of presence may impact their overall perception of visual quality. From our pilot studies, we also found that sound was an important factor when measuring presence. Some of the participants also mentioned a difference in visual quality between the 360-video and virtual objects relating to video compression and camera artifacts. Adding camera effects [18] such as noise and lens distortion will help to improve the visual fidelity. Future work could explore these ideas further.

The ranking question showed that caustics, god rays and fog were strongly preferred depending on the scene. This indicates that if a lighting effect is present in the 360-video footage, then the virtual objects need to receive the same lighting effect. Interestingly, the particles were not strongly preferred in any scene, with a slight preference in *Fiji*. This may be due to the fact that caustics, god rays and fog are all lighting effects which the 360-video can affect in real-time. For example, the caustics and god ray direction are affected by the light direction, and the fog colour changes with the background video colour. Particles, however, are manually placed. This may indicate that the lighting need to be carefully synced with the background video. Future work may consider a system to detect particles in the video to then produce similar particles.

Naive ambient lighting received a higher score in both visual quality and presence than expected, where it is only slightly negative. We suspect that the textured models gave a realistic impression, therefore people were reluctant to give it strong negative scores.

6.3 Performance

We tested the performance on a computer using an Nvidia GeForce GTX 1070 graphics card, Intel Xeon Processor E5-1607 v2 with a 3GHz clock speed and 16GB ram. Generating the 64x32 resolution textures required for IBL (the same as MR360 [29]) took 0.2ms. The caustic map took approximately 0.3ms to render at a resolution of 512x512 pixels, a resolution chosen to be large enough to capture detail while reducing memory usage. Because most of the complex lighting is stored in textures, rendering the final image only requires texture look-ups. We are able to achieve less than 3ms rendering time for scenes using a 4k video and containing four 3D models with 14k triangles each. Increasing from four to twenty models still took less than 5ms to render. See Table1 for details .

Lighting Effect	Time (ms)
IBL textures	0.051ms
Caustic Map Generation	0.275ms
God Rays	0.062ms
Fog Texture	0.041ms
Total	0.429ms

Table 1: Time spent each frame generating the textures

7 Conclusion

There are various complex underwater lighting effects that need to be considered when compositing virtual objects into underwater footage. We presented novel real-time lighting and composition methods, and a toolkit designed for underwater mixed reality with 360-videos, enabling interactive virtual objects which seamlessly blend into underwater 360-video. We achieve this by using the 360-video as a light source, as well as adding caustics, fog, god rays and particles which are driven by the background 360-video. Since there is no pre-computation and runs in real-time, this allows for a semi-automatic solution in which users can interactively adjust the underwater lighting parameters to match the lighting in the underwater 360-video. Once the parameters are set using the Underwater Toolkit, they will be updated with the video-frames in real-time. The Underwater Toolkit is implemented in Unity and UE4, providing users with an interface and other functionality of the game engine (animations, game logic, AI, etc.). We conducted a user study verifying that our underwater lighting and composition methods improves both visual quality and presence in 360-video. The user study also provided some insights, such as the impact of presence on visual quality, the importance of interactivity and sound, and the importance of matching the underwater lighting that is present in the background 360-video.

There are still limitations that can be improved in future work. Our method is semi-automatic, allowing for artistic user refinement. However, these few manual tasks could be improved for better matching with the 360-video. The virtual particles are simulated without matching the real-world particles in the 360-video. The feedback from the user study suggested better matching between the simulated particle motion and the physical particle motion from the video could help to improve the sense of presence. Our user evaluation presented a subjective evaluation of the system as a whole, whereas physiological experiments could be considered to analyze the underwater lighting effects. We currently use the light detection and the ambient colour of the background, which updates per frame, to update the underwater lighting effects. However, automatic detection of the water surface to dictate the lighting parameters in real-time will also help with automation and blending. We also assume the camera used to record the 360-video is horizontally aligned with the water surface, which could be addressed using image stabilization as a pre-processing step. We also assumed a main directional light (sun) for IBL. Positional lights, such as torches held by nearby divers, will have subtle differences. Future work could address this by including depth and light position detection.

Acknowledgments

This project was supported by the Entrepreneurial University Programme funded by TEC and in part by the Smart Ideas project funded by MBIE in New Zealand. We thank Boxfish Research for providing 360-videos captured with their underwater 360° camera.

References

[1] L. Baboud and X. Décoret. Realistic water volumes in real-time. In *Eurographics Workshop on Natural Phenomena*, 2006.

[2] 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*, pp. 29–38. ACM, 1995.

[3] R. Costa, R. Guo, and J. Quarles. Towards usable underwater virtual reality systems. *Virtual Reality (VR), 2017 IEEE*, 2017.

[4] 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*, pp. 189–198. ACM, 1998.

[5] M. Ernst, T. Akenine-Möller, and H. W. Jensen. Interactive rendering of caustics using interpolated warped volumes. In *Proceedings of Graphics Interface 2005*, pp. 87–96. Canadian Human-Computer Communications Society, 2005.

[6] A. Fournier and W. T. Reeves. A simple model of ocean waves. *ACM Siggraph Computer Graphics*, 20(4):75–84, 1986.

[7] D. Gutierrez, F. J. Seron, J. Lopez-Moreno, M. P. Sanchez, J. Fandos, and E. Reinhard. Depicting procedural caustics in single images. In *ACM Transactions on Graphics (TOG)*, vol. 27, p. 120. ACM, 2008.

[8] T. Hartmann, W. Wirth, H. Schramm, C. Klimmt, P. Vorderer, A. Gysbers, S. Böcking, N. Ravaja, J. Laarni, T. Saari, et al. The spatial presence experience scale (spes). *Journal of Media Psychology*, 2015.

[9] L. G. Henyey and J. L. Greenstein. Diffuse radiation in the galaxy. *The Astrophysical Journal*, 93:70–83, 1941.

[10] S. Hillaire. Towards unified and physically-based volumetric lighting in frostbite. In *SIGGRAPH Advances in Real-Time Rendering in Games*, 2015.

[11] W. Hu, Z. Dong, I. Ihrke, T. Grosch, G. Yuan, and H.-P. Seidel. Interactive volume caustics in single-scattering media. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pp. 109–117. ACM, 2010.

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

[13] K. Iwasaki, Y. Dobashi, and T. Nishita. An efficient method for rendering underwater optical effects using graphics hardware. In *Computer Graphics Forum*, vol. 21, pp. 701–711. Wiley Online Library, 2002.

[14] K. Iwasaki, Y. Dobashi, and T. Nishita. A fast rendering method for refractive and reflective caustics due to water surfaces. In *Computer Graphics Forum*, vol. 22, pp. 601–609. Wiley Online Library, 2003.

[15] H. W. Jensen. Global illumination using photon maps. *Rendering techniques*, 96:21–30, 1996.

[16] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pp. 143–150. ACM, New York, NY, USA, 1986. doi: 10.1145/15922.15902

[17] P. Kán and H. Kaufmann. High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pp. 99–108. IEEE, 2012.

[18] G. Klein and D. W. Murray. Simulating low-cost cameras for augmented reality compositing. *IEEE transactions on visualization and computer graphics*, 16(3):369–380, 2009.

[19] J. Křivánek, S. Pattanaik, and J. Žára. Adaptive mesh subdivision for precomputed radiance transfer. In *Proceedings of the 20th spring conference on Computer graphics*, pp. 106–111. ACM, 2004.

[20] E. Lafortune. Mathematical models and monte carlo algorithms for physically based rendering. *Department of Computer Science, Faculty of Engineering, Katholieke Universiteit Leuven*, 20:74–79, 1996.

[21] E. P. Lafortune and Y. D. Willems. Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pp. 145–153. Alvor, Portugal, December 1993.

[22] E. P. Lafortune and Y. D. Willems. Rendering participating media with bidirectional path tracing. In *Rendering techniques 96*, pp. 91–100. Springer, 1996.

[23] G. Liktor and C. Dachsbacher. Real-time volume caustics with adaptive beam tracing. In *Symposium on Interactive 3D Graphics and Games*, pp. 47–54. ACM, 2011.

[24] R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integrals for all-frequency relighting. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 477–487. ACM, 2004.

[25] C. Papadopoulos and G. Papaioannou. Realistic real-time underwater caustics and godrays. In *Proc. GraphiCon*, vol. 9, pp. 89–95, 08 2010.

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

[27] 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*, pp. 497–500. ACM, 2001.

[28] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance

and irradiance: determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001.

[29] T. Rhee, L. Petikam, B. Allen, and A. Chalmers. Mr360: Mixed reality rendering for 360 panoramic videos. *IEEE Transactions on Visualization & Computer Graphics*, pp. 1379–1388, 2017.

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

[31] M. A. Shah, J. Konttinen, and S. Pattanaik. Caustics mapping: An image-space technique for real-time caustics. *IEEE Transactions on Visualization and Computer Graphics*, 13(2), 2007.

[32] M. Slater, M. Usoh, and A. Steed. Depth of presence in virtual environments. *Presence: Teleoperators & Virtual Environments*, 3(2):130–144, 1994.

[33] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 527–536. ACM, 2002.

[34] T. Sousa. Crysis next gen effects. In *Game Developers Conference*, 2008.

[35] J. Stam. Random caustics: natural textures and wave theory revisited. In *SIGGRAPH Visual Proceedings*, p. 150, 1996.

[36] J. Tarko, J. Tompkin, and C. Richardt. Omnimr: Omnidirectional mixed reality with spatially-varying environment reflections from moving 360 video cameras. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2019.

[37] T. Tawara and K. Ono. An application of photo realisticwater surface interaction using mixed reality. *6th Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2009*, 2009.

[38] J. Tessendorf et al. Simulating ocean water. *Simulating nature: realistic and interactive techniques. SIGGRAPH*, 1(2):5, 2001.

[39] S. Thompson, A. Chalmers, and T. Rhee. Real-time underwater caustics for mixed reality 360° videos. In *2019 IEEE VR*. IEEE, 2019.

[40] M. Watt. Light-water interaction using backward beam tracing. *ACM SIGGRAPH Computer Graphics*, 24(4):377–385, 1990.

[41] B. Wronski. Volumetric fog: Unified compute shader vased solution to atmospheric scattering. In *ACM SIGGRAPH*, 2014.