

Real-Time Image Based Lighting for 360-degree Panoramic Video

Thomas Iorns and Taehyun Rhee

Victoria University of Wellington

Abstract. This paper presents an effective approach to rendering virtual 3D objects using real-time image based lighting (IBL) with conventional 360° panoramic video. Raw 360° panoramic video captured in a low dynamic range setup is the only light source used for the real-time IBL rendering. Input video data is boosted to high dynamic range using inverse tone mapping. This converted video is then reconstructed into low-resolution diffuse radiance maps to speed up diffuse rendering. A mipmap-based specular sampling scheme provides fast GPU rendering even for glossy specular objects. Since our pipeline does not require any precomputation, it can support a live 360° panoramic video stream as the radiance map, and the process fits easily into a standard rasterization pipeline. The results provide sufficient performance for IBL via stereo head mounted display (HMD), an ideal device for immersive augmented reality films and games using 360° panoramic videos as both the lighting and backdrop for illumination composition.

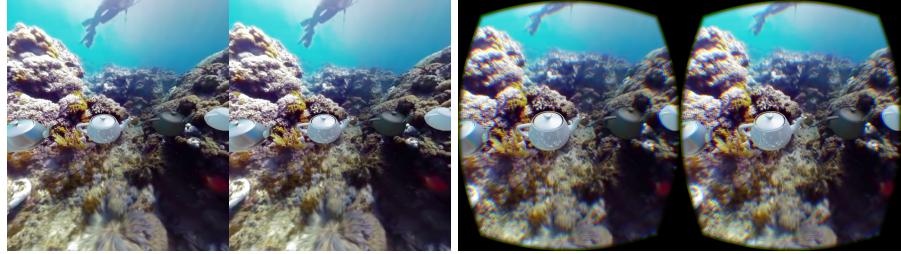


Fig. 1: IBL rendering for stereoscopic output using 360° video via our pipeline (left), and the result in head-mounted display (right).

1 Introduction

With the recent rapid growth of interest in stereographic head-mounted displays (HMDs) such as the Oculus Rift [1], content for HMDs is being created at a steadily increasing rate. One important application of content for HMDs is in

the creation of virtual reality (VR) and augmented reality (AR) experiences. In particular AR is a live direct or indirect view of a real-world environment whose elements are augmented by computer generated content such as 3D virtual objects. In order to maximize the immersive experience of AR content on HMDs, the seamless composition of the 3D virtual object and the real-world environment is important. Image based lighting (IBL) [2,3] has been used to emulate global illumination (GI) within a real-world scene, where the distant ambient and directional lighting is stored in a single image acting as the radiance map. This method provides high quality, realistic lighting useful for photo-realistic rendering and composition with real-world scenes in live-action films, and augmented reality. The image used for IBL is created by capturing high dynamic range (HDR) 360-degree panoramic images using photographs at various angles and exposure levels. A 360-degree panoramic image can also provide an ideal and intuitive format for HMDs to cover the whole range of viewpoints arising from motion of the viewer's head. Because of this the hardware for capturing panoramic images and video has become readily available, and as a result many 360° videos (video captured with a full spherical 4π steradian field of view) can be found on popular video sharing websites such as YouTube [4]. These videos provide a high level of immersion when viewed via HMDs, for a relatively low cost of content production.

The use of 360-degree panoramic images for virtual object lighting in augmented reality has already been proposed [5] and studied in previous research [6,7]. However when using conventional 360° video as the radiance map for IBL, we need to consider some additional challenges. Firstly IBL requires a high dynamic range (HDR) image of the scene environment to be captured. This requires views of various angles and exposure levels to be provided, and is generated by a user-guided post processing step. Although it is possible to create HDR 360° video using a special device setup such as in [8,9], conventional 360° videos [4] are captured using low dynamic range sensors and cannot provide enough dynamic range data for direct use in IBL. Secondly the IBL technique used must provide real-time rendering of a pair of stereo images shown at high frame rate; HMDs often require stereo rendering of 60 to 90 frames per second (FPS) in order to prevent visual discomfort [10]. Finally, when considering a live 360° video stream for immersive AR applications, precomputation of the radiance map is tricky and limited [11].

In this paper we present a novel pipeline that addresses the above challenges practically. Recent perceptual studies [12,13] have shown that the proper inverse tone-mapping algorithm can reconstruct the dynamic range that is required for IBL, from low dynamic range (LDR) input, and to such a level that the human visual system (HVS) cannot perceive the difference. Chalmers et al. [12] provide a threshold of image resolution that maintains the seamlessness of the final illumination composition. We adapt the perceptual threshold of the HVS to optimize our pipeline. The dynamic range of the LDR 360° video is converted to HDR using an inverse tone mapping operator. By using low-resolution versions of the panoramic video in lighting calculations for which the result is percep-

tually similar, we are able to emulate various common material properties in real-time. A mipmap-based specular sampling scheme provides fast rendering even for glossy specular objects. Since our pipeline does not require any pre-computation, it has the potential to support a live 360-degree panoramic video stream as the radiance map, and the process fits easily into a standard GPU rasterization pipeline. The resulting pipeline reliably provides framerates of over 75Hz, as required for comfortable viewing on stereo HMDs. The result is the IBL of 3D objects providing a seamless mixture of illumination with the 360-degree panoramic video backdrop. Based on our survey, this is the first practical system that provides interactive IBL from an LDR 360° live video stream suitable for HMDs. The overview of the system pipeline is shown in Fig. 2, and examples of the results for some test video frames can be seen in Fig. 8.

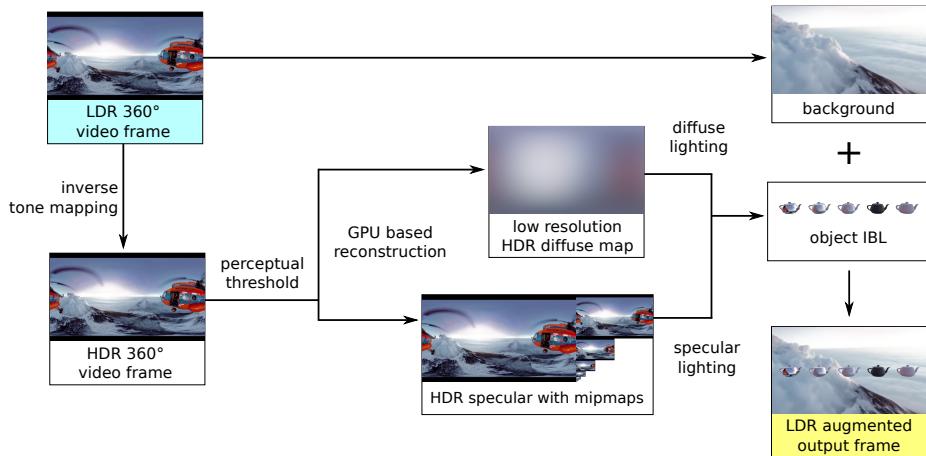


Fig. 2: System overview

2 Background and Related Work

Image based lighting was described early on by Miller and Hoffman [2] and has been popularized byDebevec [3], who has among other applications used it to convincingly render virtual objects into real-life photographs [14]. Heidrich and Seidel [15] described real-time use of IBL, precomputing diffuse and glossy material lighting integrals to efficiently render these materials by looking up the precomputed values according to surface normal or reflection direction. Kautz and McCool [16] simulated more complex material types by approximating their reflectance properties as a linear combination of glossy reflection from multiple directions. These techniques form the core of many real-time IBL applications (see for example [17,5]).

Ramamoorthi and Hanrahan [18] described an efficient way to represent diffuse radiance maps. They showed that the precomputed lighting integrals for diffuse materials can be described using only a very small number of coefficients in a spherical harmonics basis. These coefficients were shown by King [11] to be able to be calculated in real-time using a dedicated graphics processor, allowing real-time IBL of diffuse materials without precomputation. This is demonstrated by Pessoa et al. [7] to be applicable to augmented reality applications.

An alternate method of calculating material lighting, which has become more performant as graphics hardware has improved, is to approximate the lighting integral directly by sampling from many points on the environment image. Recently this has been used by Kronander et al. [9] to render a virtual object into video in real-time (around 25 frames per second on an Nvidia GeForce 770). For real-time performance it is necessary to use an importance sampling technique (such as [19]) so as to get the most accurate result with the smallest number of samples. To obtain their environment image they used a special device, an additional HDR video camera mounted underneath the primary LDR video camera shooting the scene. This HDR video camera recorded the environment via an attached light probe [3].

Recently Michiels et al. [20] presented a method for IBL using 360° video to provide lighting for virtual objects. They analyze video from a moving 360° video camera in order to determine the position of the camera at each frame. They then use the frames to vary the lighting on a virtual object as it is moved around a reconstructed virtual environment. As well as having a different focus, their technique is also different from ours. They calculate their lighting in terms of spherical radial basis functions [21], which requires precomputing the lighting for each video frame in this basis. According to our survey, we could not find any prior work creating real-time IBL rendering using conventional LDR 360° video as the radiance map without a special device or precomputation.

The main difference in our technique stems from recent perceptual studies. Chalmers et al. [12] observed that the resolution of an environment image being used for lighting could be greatly reduced without causing any noticeable difference to the rendered scene. Akyüz et al. [13] found that a simple tonemapping operator is often adequate for believably converting LDR images to HDR. We use these results to perform lighting calculations in real-time, by reducing the resolution of our input and tonemapping LDR to HDR as appropriate. The primary resulting benefits to this approach are a lack of precomputation, the potential to simulate various material types, and the ability to operate solely on existing LDR content.

3 Real-Time IBL from 360-degree Video

We present a real-time IBL system using 360° video. The input video frame can be used to directly represent specular reflection, and along with techniques for representing glossy and diffuse surfaces, a large number of real-world opaque materials are able to be simulated. In order to provide real-time diffuse illumi-

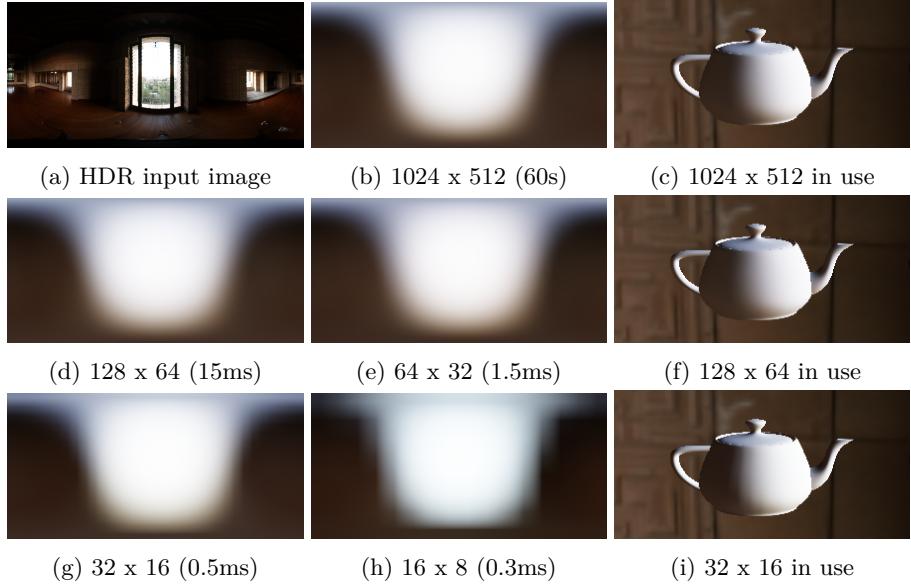


Fig. 3: Diffuse radiance maps computed at various resolutions. Generation time on an Nvidia GeForce 730 is given in parentheses. Maps are displayed and sampled using hardware bilinear filtering.

nation, we generate a new diffuse radiance map per frame, which can be done in real-time at a perceptually optimized lowered resolution [12]. For specular material types including mirror-like and glossy specular, we sample light from the radiance map according to the reflectance function, at an appropriately reduced resolution using mipmaps. We test our method using three different material setups: diffuse reflection, pure specular reflection, and glossy specular reflection. These properties can be combined to simulate a wide range of believable materials.

3.1 Diffuse Illumination

Diffuse lighting is calculated by generating a diffuse radiance map, as in [2], and sampling from this according to surface normal direction. Assuming fixed aspect ratio, generating the diffuse radiance map is an $O(N^4)$ operation in environment image height. For even fairly low resolution environments this quickly becomes prohibitive. We find viewing on HMDs however, that even when calculated at resolutions as low as 32 by 16 pixels, the diffuse lighting remains visually similar to the full-resolution version when sampled with bilinear filtering as shown in Fig. 3. Most of the output images in this paper were generated using 32 by 16 diffuse maps. A comparison including generation times can be seen in Fig. 3. Even at 128 by 64 pixels the maps can be generated on a low-end graphics card more quickly than standard video framerates require, and as Chalmers

et al. show [12] lighting using diffuse maps down to 80 by 40 pixels can be perceptually indistinguishable from lighting using full resolution maps. Once the diffuse radiance map for a frame is generated, the diffuse lighting calculation for any point on an object’s surface consists simply of a single texture lookup.

3.2 Specular Illumination

Pure specular reflection can be easily achieved by using the input video frame as an environment map. To calculate the specular component of material colour, we simply take the vector from surface to camera, reflect it across the surface normal, and sample from the environment map in this direction, as shown in Fig. 4a.

Generating accurate glossy specular reflection is computationally expensive. However, it can be approximated by computing glossy environment maps in a similar manner to the diffuse environment map. This is fine for very rough surfaces where the glossy lighting calculation is similar to the diffuse lighting calculation, but it becomes prohibitive as the surface roughness decreases and the gloss level approaches specular. The closer to mirror-reflection the gloss level is, the higher the resolution of the glossy environment map needed to describe it.

An alternative approach is to directly sample from the specular radiance map (in our case the original video frame), and we present a novel approach to do this efficiently using the mipmaps that have already been computed for diffuse map generation. We take a small number of samples in a radius around the specular direction, choosing a mipmap level appropriate to the angular distance between samples. The sampling radius depends on the surface roughness parameter. For lower roughness, a higher resolution mipmap level is sampled from, but the decreasing radius of the glossy reflection lobe (see Fig. 4b and 5) means the same number of samples is required independent of roughness level.

In this way glossy specular lighting can be approximated using a fixed number of texture lookups per rendering fragment. In our tests we found that taking about 18 samples inside the primary glossy lobe, 18 samples outside it, and weighting them by a simple Phong model [22] gave fast and believable glossy surface lighting.

4 Inverse Tone-Mapping from LDR to HDR

In a typical IBL setup, all lighting calculations will assume HDR lighting input. The difference between scenes rendered using LDR and HDR IBL is immediately apparent (see Fig. 7), with HDR lighting greatly increasing the realism of scenes when compared with LDR lighting. It turns out, however, that simple automatic LDR to HDR conversions can be sufficient for creating believable lighting effects [12,13]) when targeting the human visual system. As such we are able to believably light virtual objects using only LDR video.

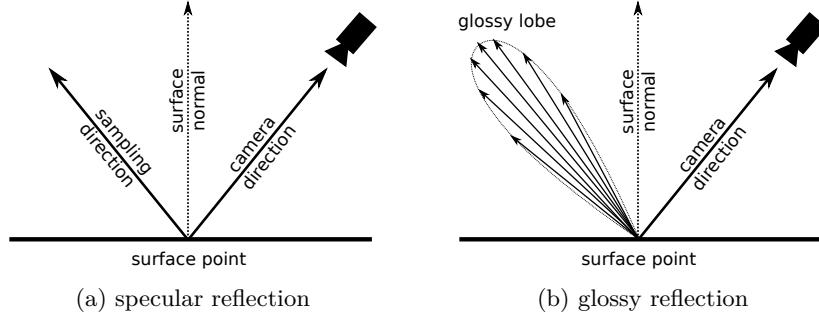


Fig. 4: Specular and glossy specular reflection. The environment is sampled at and around the direction of specular reflection according to the size of the glossy lobe, which is determined by a material roughness parameter.

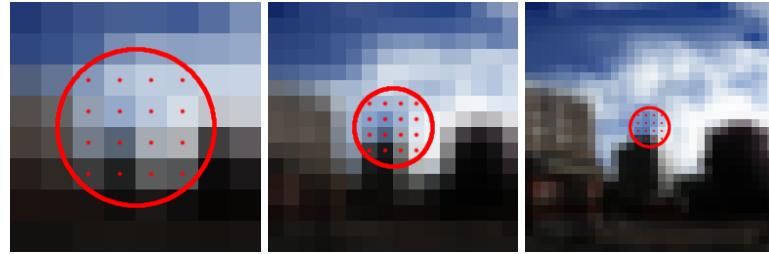


Fig. 5: To approximate glossy reflection we sample from an appropriate mipmap level of the input environment map, according to the size of the glossy lobe. The number of samples remains constant.



Fig. 6: Glossy specular reflection for various lobe sizes determined by a roughness parameter of, from left to right: 0.01, 0.05, 0.1, 0.2, 0.5.

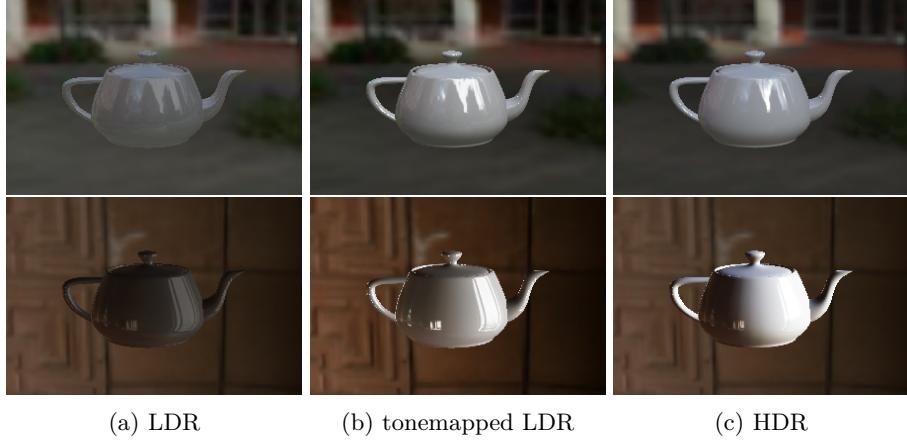


Fig. 7: Comparison of lighting using an LDR environment image (left), an LDR-HDR tonemapped image (middle), and an HDR environment image (right). The simple tonemapping method provides believable results when viewed independently, although it may not perfectly match the HDR lighting result (especially for very high contrast scenes such as the bottom scene here).

The tone-mapping operator we use is independent of varying frame properties, applying the same transform to each pixel individually. As such it is easily and efficiently implemented on the GPU. Given an input RGB value we first determine input luminosity as

$$L_i = 0.3 \cdot R_i + 0.59 \cdot G_i + 0.11 \cdot B_i , \quad (1)$$

where R_i , G_i and B_i are the red, green and blue components of the input image, as values between 0.0 and 1.0, and L_i is the calculated input luminosity. We then calculate a desired scaling factor L_s based on this input luminosity as

$$L_s = 10 \cdot L_i^{10} + 1.8 . \quad (2)$$

The output red, green and blue components are then determined by

$$[R_o, G_o, B_o] = L_s \cdot [R_i, G_i, B_i] . \quad (3)$$

Parameters here were determined by experiment, and those in Equation (2) have been used for all examples in this paper. After this operation, the converted image is used as an HDR radiance map in the following rendering steps.

5 GPU Implementation

We implemented our method using the GPU, and tested on various consumer-grade video cards including Nvidia GeForce 690, 770 and 980 as well as AMD Radeon 270. The pipeline was laid out as follows.

While GPU instructions are being queued, a separate thread loads and decodes the next video frame. This is passed to the GPU using a pixel buffer object for asynchronous memory transfer. As our target display refresh rate is much higher than typical video framerate (minimum 75 frames per second for HMD, typical 25 frames per second for video) one frame is used for display as the next is being loaded, giving smooth video performance.

The main pipeline works as depicted in Fig. 2. The basic procedure is described in Algorithm 1. The diffuse radiance map is upsampled simply by enabling bilinear texture filtering. For head-mounted stereographic display, the display process is executed twice, once for each eye (see Fig. 1 for example output).

Object lighting is performed in a GPU fragment shader. Specular and diffuse lighting consist simply of texture lookups into the specular and diffuse radiance maps. Glossy specular lighting uses a somewhat more complicated system of sampling in a fixed pattern from lower-resolution mipmap levels of the specular map. The specific sampling system we used was to take samples in concentric rings around the specular direction, with six samples per ring, three rings inside the glossy lobe (see Fig. 4b), and three rings outside it. Which mipmap level to sample from is determined by taking the distance between samples in a ring, and choosing the level with an equivalent distance between pixels. Samples are weighted by a Phong model [22], and the radius of the glossy lobe is defined to be the distance at which the weighting is 0.5. To reduce discretization artifacts, hardware trilinear filtering is enabled.

6 Results

We tested our method on several 360° videos obtained from a popular video sharing website [4]. Video resolutions were between 1920x960 and 2048x1024, with framerates between 24 and 30Hz. We tested various 3D objects without texture maps including Teapot (6320 triangles) and Bunny (69451 triangles). Five objects were displayed at once, each using a different combination of the material setups described in Section 3. We show that lighting virtual models using this method, with nothing other than LDR 360° panoramic video as input, gives believable results over a wide variety of input lighting conditions (see Fig. 8). We rendered a single camera view at 1280x720 resolution. By subjective visual comparison, lighting of the virtual objects seems to match that of the background video in all tested cases.

Rendering with tonemapped LDR to HDR frames seems quite sufficient for believable lighting (see Fig. 7). This is in agreement with [12] and [13]. Rendering using only the LDR frames for lighting appears dull, and does not match the background scene. Performance of the GPU based IBL is quite efficient, executing at around 90Hz (11ms per frame) on a GeForce 690 and around 500Hz (2ms per frame) on a GeForce 980. This leaves plenty of time for additional rendering tasks.

Additionally, we tested our method in an Oculus DK2 HMD. Using an Nvidia GeForce 690 we were able to render stereo output at 1182x1464 resolution for

```

Input : LDR 360° video
Input : 3D mesh data
Output : LDR framebuffer texture for display
while program running do
    if time elapsed > 1 / video FPS then
        /* tonemap next frame to HDR texture */ 
        for pixel ← video frame do
            | pixel colour *= 10 * (pixel luminance)10 + 1.8;
        end
        /* render diffuse radiance map */
        for outpixel ← diffuse map do
            accumulator = 0;
            weight = 0;
            for inpixel ← low resolution specular map do
                inpixel area = cos(inpixel latitude);
                inpixel weight = inpixel area * angle between(inpixel, outpixel);
                accumulator += inpixel weight * environment colour(inpixel
                direction);
                weight += inpixel weight;
            end
            outpixel colour = accumulator / weight;
        end
        reset elapsed time;
    end
    draw specular environment to output buffer as background;
    rasterize virtual objects into output buffer using IBL shader;
end

```

Algorithm 1: Real-Time IBL Process

both eyes (see Fig. 1) of the HMD comfortably at 75Hz.

7 Conclusion

This paper presents an effective approach to rendering virtual 3D objects using real time image based lighting (IBL) with conventional 360° panoramic video. Using only low dynamic range 360° panoramic video as input we find that 3D virtual objects can be nicely composited into the panoramic video using IBL rendering. This form of video-based object lighting can be done entirely in real-time and with no precomputation required.

The visual quality of our result is similar to previous IBL techniques requiring precomputation and HDR environment images. This is achieved using the perceptual observation that low resolution environment maps and tonemapped LDR to HDR images can be sufficient for believable lighting.

One aspect of virtual object rendering we do not consider is that of shadowing. A possible extension of this work would be to incorporate an existing real-time shadowing technique such as that of [6]. There is also room for improvement

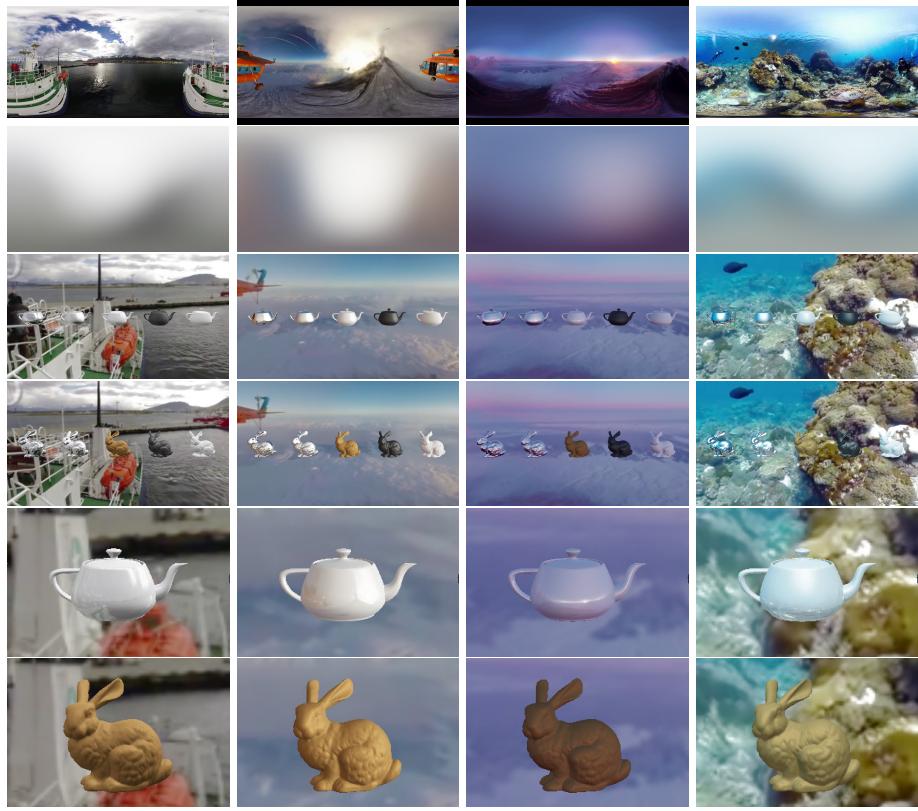


Fig. 8: Output of our pipeline for some of our test video frames. The input 360° video frame is displayed on the top row, and the generated diffuse radiance map on the second.

in the compositing technique used to blend the rendered virtual objects with a perspective view of the input video. A potential improvement might be to render the virtual objects directly into the video frame, which we have not yet explored.

Our main contribution is a fully self-contained real-time system for lighting virtual objects so as to match an environment provided via LDR 360° panoramic video content. The novelty of our work is that our system requires no prior analysis of the video stream, can simulate complex materials efficiently, and can work using only readily-available LDR content. The system is efficient enough to render in real-time at the high framerates and resolutions required for immersive head-mounted display. While each component of the pipeline could be improved using more sophisticated methods, a balanced trade-off between perceptible visual quality and sufficient rendering performance is required for practical applications.

References

1. “Oculus rift.” <http://www.oculusvr.com/rift>. Accessed: 2015-09-12.
2. G. Miller and C. R. Hoffman, “Illumination and reflection maps: Simulated objects in simulated and real environments,” in *SIGGRAPH 84 Advanced Computer Graphics Animation seminar notes*, vol. 190, 1984.
3. P. Debevec, “Image-based lighting,” *IEEE Computer Graphics and Applications*, no. 2, pp. 26–34, 2002.
4. “360Video - YouTube.” <http://youtube.com/360>. Accessed: 2015-09-12.
5. 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*, pp. 208–216, IEEE, 2003.
6. P. Supan, I. Stuppacher, and M. Haller, “Image based shadowing in real-time augmented reality,” *IJVR*, vol. 5, no. 3, pp. 1–7, 2006.
7. S. A. Pessoa, E. L. Apolinario, G. d. S. Moura, J. P. S. d. M. Lima, M. A. Bueno, V. Teichrieb, and J. Kelner, “Illumination techniques for photorealistic rendering in augmented reality,” in *X Symposium on Virtual and Augmented Reality, João Pessoa, PB, Brasil*, pp. 223–232, 2008.
8. J. Unger, J. Kronander, P. Larsson, S. Gustavson, and A. Ynnerman, “Temporally and spatially varying image based lighting using hdr-video,” in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pp. 1–5, IEEE, 2013.
9. J. Kronander, J. Dahlin, D. Jonsson, M. Kok, T. Schon, and J. Unger, “Real-time video based lighting using gpu raytracing,” in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pp. 1627–1631, IEEE, 2014.
10. R. Yao, T. Heath, A. Davies, T. Forsyth, N. Mitchell, and P. Hoberman, “Oculus vr best practices guide,” *Oculus VR*, 2014.
11. G. King, “Real-time computation of dynamic irradiance environment maps,” *GPU Gems*, vol. 2, pp. 167–176, 2005.
12. A. Chalmers, J. J. Choi, and T. Rhee, “Perceptually optimised illumination for seamless composites,” in *Pacific Graphics Short Papers* (J. Keyser, Y. J. Kim, and P. Wonka, eds.), The Eurographics Association, 2014.
13. A. O. Akyüz, R. Fleming, B. E. Riecke, E. Reinhard, and H. H. Bültlhoff, “Do HDR displays support LDR content?: a psychophysical evaluation,” in *ACM Transactions on Graphics (TOG)*, vol. 26, p. 38, ACM, 2007.
14. P. Debevec, “Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography,” in *ACM SIGGRAPH 2008 classes*, p. 32, ACM, 2008.
15. W. Heidrich and H.-P. Seidel, “Realistic, hardware-accelerated shading and lighting,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 171–178, ACM Press/Addison-Wesley Publishing Co., 1999.
16. J. Kautz and M. D. McCool, “Approximation of glossy reflection with prefiltered environment maps,” in *Graphics Interface*, vol. 2000, pp. 119–126, 2000.
17. J. Unger, M. Wrenninge, and M. Ollila, “Real-time image based lighting in software using HDR panoramas,” in *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pp. 263–ff, ACM, 2003.

18. 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.
19. D. Burke, A. Ghosh, and W. Heidrich, “Bidirectional importance sampling for direct illumination.,” *Rendering Techniques*, vol. 5, pp. 147–156, 2005.
20. N. Michiels, L. Jorissen, J. Put, and P. Bekaert, “Interactive augmented omnidirectional video with realistic lighting,” in *Augmented and Virtual Reality*, pp. 247–263, Springer, 2014.
21. J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, “All-frequency rendering of dynamic, spatially-varying reflectance,” in *ACM Transactions on Graphics (TOG)*, vol. 28, p. 133, ACM, 2009.
22. B. T. Phong, “Illumination for computer generated pictures,” *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.