High-Quality Consistent Illumination in Mobile Augmented Reality by Radiance Convolution on the GPU

Peter Kán, Johannes Unterguggenberger, and Hannes Kaufmann

Institute of Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria

Abstract. Consistent illumination of virtual and real objects in augmented reality (AR) is essential to achieve visual coherence. This paper presents a practical method for rendering with consistent illumination in AR in two steps. In the first step, a user scans the surrounding environment by rotational motion of the mobile device and the real illumination is captured. We capture the real light in high dynamic range (HDR) to preserve its high contrast. In the second step, the captured environment map is used to precalculate a set of reflection maps on the mobile GPU which are then used for real-time rendering with consistent illumination. Our method achieves high quality of the reflection maps because the convolution of the environment map by the BRDF is calculated accurately per each pixel of the output map. Moreover, we utilize multiple render targets to calculate reflection maps for multiple materials simultaneously. The presented method for consistent illumination in AR is beneficial for increasing visual coherence between virtual and real objects. Additionally, it is highly practical for mobile AR as it uses only a commodity mobile device.

1 Introduction

Augmented reality is the technology which superimposes virtual objects into the real world while accurate spatial and visual registration has to be achieved [1]. In order to achieve accurate visual registration and visual coherence, a consistent illumination of real and virtual objects is essential. Therefore, illumination from the real world has to be captured and used to illuminate the virtual objects. However, previous methods for consistent illumination were either limited to light with low angular frequency [2, 3] or required a complex hardware setup [4–6] which is not suitable for mobile AR.

In this paper, we propose a method for rendering with consistent illumination in mobile AR which does not require special hardware and utilizes both low-frequency and high-frequency illumination in the angular domain. Our method overcomes the limitation of low-frequency lighting by reconstructing environment map, representing real illumination, from images captured by mobile device. The overlapping image data are accumulated and a high dynamic range (HDR) environment map is reconstructed interactively. This map is then processed on the

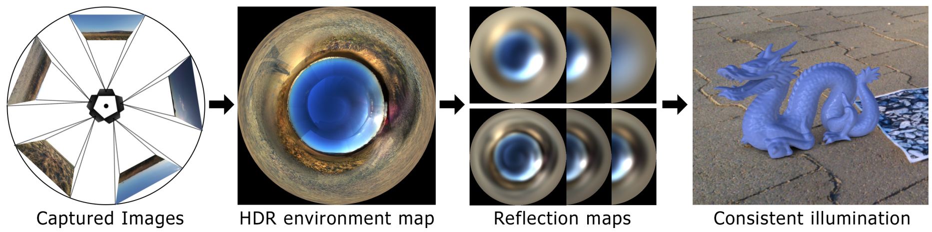

Fig. 1. The process of HDR environment map capturing and rendering with consistent illumination in AR. From left to right: Multiple input images with different exposures and different device orientations are combined into one HDR spherical map. This environment map is used to calculate reflection maps on mobile GPU. Finally, the reflection maps are used for real-time rendering with consistent illumination in AR.

mobile GPU and used for rendering of virtual objects. In order to calculate high-quality light reflections on the virtual surfaces, our method generates reflection maps by convolving the input map by the bidirectional reflectance distribution functions (BRDFs) of materials in the scene. Previous methods approximated this convolution either by MIP-mapping or by the Gaussian blur which leads to inaccurate light reflections. We overcome this limitation by calculating high-quality reflection maps on the mobile GPU which operates on output pixels in parallel. The reflection maps are calculated at the beginning of the rendering step and then they are used to render high-quality reflections in real-time. All calculations are done in HDR. Our method utilizes multiple render targets to simultaneously generate reflection maps for multiple shininess factors of Phong BRDF. Additionally, we use lightmaps to precalculate ambient occlusion and simulate a local visibility occlusion in high quality. The process of HDR environment map creation, reflection maps calculation, and rendering with consistent illumination is depicted in Figure 1. This figure shows the tonemapped environment maps by exposure and gamma correction.

The results of this paper demonstrate the quality improvement of the rendering with our method in comparison to the common MIP-mapping approach. Moreover, we compare our results to the reference solution, rendered by offline path tracing. Our method produces results which are close to the reference solution. Both, light capturing and rendering are done on site and require only a commodity mobile device. Therefore, our method is highly usable to render high-quality consistent illumination in mobile AR.

The main contributions of this paper are:

- A two step method for rendering with consistent illumination in AR which uses only a mobile device.
- A method for efficient calculation of high-quality irradiance environment maps on a mobile GPU.
- Parallel generation of reflection maps for multiple materials.

2 Related Work

Rendering of virtual objects with consistent illumination has been a challenging problem in AR. In order to solve this problem, the illumination of the real world has to be captured or estimated and then used for the rendering of virtual objects. Previous methods which addressed this problem can be divided into two categories based on how the real light is acquired: Methods using a light probe (either passive reflective sphere or active camera with fish eye lens) and methods estimating the light from a main camera image. Some of these methods estimate the light in real-time and the others need captured illumination as an input.

Light probe methods. Methods based on light probes use a special hardware (usually a camera with a fish eye lens) to capture the illumination in high quality. Approaches which capture the illumination in real-time and use it for rendering in AR were presented in [7, 6, 8, 5, 9, 10]. These approaches also calculate global illumination in an AR scene which introduces the light reflections between virtual and real objects. The advantage of these methods is a high visual fidelity of the rendered result. However, they need a high computational power and therefore are not suitable for mobile devices. A method for consistent near-field illumination which runs on a mobile device was presented by Rohmer et al. [11]. This method renders virtual objects lit consistently with the real world and it calculates global illumination. Its limitation is the requirement of multiple external cameras connected to a computer which sends the data to the mobile device via Wi-Fi. In contrast to that our method requires only a mobile device and does not need any additional hardware.

Other methods use the captured illumination as an input for the rendering and then render the virtual objects with natural light. Pessoa et al. [12] used the environment map reconstructed from multiple exposure images of a reflective sphere. The authors also generate the first glossy, second glossy, and diffuse map. Then these are interpolated to approximate a specific material. Our method calculates a glossy reflection map specifically for each material. Therefore, a high accuracy of the rendering is achieved. Meilland et al. [4] proposed a method for dense reconstruction of HDR light field in a real scene by registering multiple low dynamic range images with different exposures, captured by the RGB-D camera. The reconstructed light field is then used to generate multiple light probes which illuminate virtual objects. The similarity with our method is that we also firstly capture an HDR environment map and then use it for rendering with consistent illumination. An approach for fast light factorization for AR rendering based on spherical harmonics was presented in [13].

Light estimation methods. Several methods for light source estimation from a main camera image were proposed. The advantage of these methods is that they do not require additional cameras or light probes. The estimation of real illumination from shadows was proposed by Sato et al. [14]. The authors estimate the distribution of illumination by analyzing the relationships between the image brightness and the occlusions of incoming light. Gruber et al. [3] presented a method for real-time estimation of diffuse illumination from arbitrary geometry, captured by an RGB-D camera. This method reconstructs the real geometry and surrounding illumination which is used for rendering of the virtual content in AR with consistent illumination. Recently, a method for illumination estimation from a user's face, based on trained set of radiance transfer functions, was presented in [2]. The limitation of the approaches for light estimation from the main camera image is that they do not reconstruct high-frequency illumination.

Rendering with natural illumination. Once the real light is reconstructed, rendering with natural illumination plays an important role for achieving a consistent appearance of virtual and real objects. Precalculation of reflection maps for the rendering of diffuse and specular reflection was discussed in [15, 16]. Ramamoorthi and Hanrahan [17] proposed an efficient method for irradiance environment maps calculation by utilizing spherical harmonics. This method is well suitable to calculate the diffuse illumination. A fast approximation of environment map convolution can be achieved by MIP-mapping [18–20]. Our method calculates accurate convolution of illumination by the BRDF per each pixel. Therefore, it can be used for diffuse as well as glossy, and specular materials. The methods which used real-time rendering with natural illumination in AR by reflection maps were presented in [12, 21–23]. Recently, Mehta et al. [24] presented a method for physically-based rendering in AR based on GPU path tracing and real-time filtering. Their method illuminate the scene by a captured light probe. While producing high-quality rendering results, their method is not suitable for mobile AR due to the requirement of high computational power which can be provided only by a high-performance desktop GPU.

3 Consistent Illumination in AR

The proposed method for rendering with consistent illumination in AR works in two main steps. In the first step, the user scans the environment by a mobile device. A rotational motion of the mobile device is assumed while the images of the camera are accumulated to the spherical environment map. HDR reconstruction is performed to capture the wide range of light intensities from the real world. In the second step, this captured map is convolved by the BRDF of each material in the virtual scene. The convolved reflection maps are then used for real-time rendering. We achieve consistent illumination of virtual and real objects by using the real world's light to illuminate the virtual scene. The captured environment map contains both high and low frequencies and its convolution is calculated per each pixel on the GPU. Therefore, our method achieves high quality of the final rendering. The details of each step are revealed in the next sections.

3.1 Illumination Capturing

In order to render the virtual objects lit consistently with the real world, real illumination needs to be captured. For this purpose, the user scans the surrounding environment by a mobile device performing 360° rotational motion in yaw and pitch angles to cover the full sphere of directions. An HDR environment map is reconstructed interactively during scanning. The environment light scanning utilizes an approach presented in [25] to capture HDR environment map.

Each camera image is projected to the spherical environment map according to the orientation of the device, estimated by the inertial measurement unit (IMU) of the mobile device which includes gyroscope, accelerometer and magnetometer. We use the mapping to the sphere similar to Debevec [26]. Distinctly, our map is oriented to have positive z axis (up vector) projected to the image center because we prefer to have a consistent image deformation along the horizontal direction. Geometric and radiometric calibration is performed to accurately project the captured data.

High dynamic range of the reconstructed radiance data is essential to accurately represent the high contrast of real light. Therefore, low dynamic range data, captured by the image sensor, have to be converted to the HDR radiance. Our approach for HDR reconstruction is based on the work of Robertson et al. [27]. The inverse camera response function is reconstructed for each color channel in the calibration process. Then, we use these functions for the HDR reconstruction from multiple overlapping images with known exposure times [27, 25]. A mobile camera is set to the auto-exposure mode to correctly adapt exposure time to the local portion of the scene where the camera is pointing. If two images overlap, they usually have slightly different exposure time due to the adapting auto-exposure. Therefore, we can merge the overlapping data and calculate incoming radiance in HDR. The HDR reconstruction runs on the mobile GPU and the environment map is accumulated to the floating point framebuffer to achieve interactive speed. Additionally, we employ an alignment correction based on feature matching to compensate the drift of IMU.

3.2 Environment Map Convolution

The captured HDR environment map is used to render virtual objects lit consistently with the real world lighting. At the beginning of the rendering phase, the environment map is convolved by the BRDF of each material to create a reflection map which will allow fast high quality lighting. We developed a new method for the convolution of the environment map on the GPU which is capable of convolving the map by multiple BRDFs simultaneously. Multiple render targets are utilized for this purpose. High quality of the resulting reflection map is achieved by taking each pixel of the input map into the calculation.

In order to enable the precalculation of reflected light we need to reduce the number of parameters in the reflectance calculation. We assume a distant light, represented by the captured environment map. Moreover, we calculate only single light reflection on the surface and scene materials are assumed to be non-emissive. Based on these assumptions, the rendering equation [28] can be simplified to:

$$L_o \approx \frac{2\pi}{N} \sum_{i=1}^{N} f_r(x, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(x, \boldsymbol{\omega}_i) \mid \boldsymbol{n} \cdot \boldsymbol{\omega}_i \mid \tag{1}$$

which calculates the reflected radiance L_o by taking into account each incoming radiance L_i from the environment map and the BRDF function denoted by f_r. The BRDF function depends on the direction of incoming light $\boldsymbol{\omega}_i$, direction of reflected right $\boldsymbol{\omega}_o$ and the surface position x. N stands for the number of pixels in the environment map and $\boldsymbol{n}$ is the local surface normal. In the following we loosely write $=$ instead of $\approx$ also when referring to the approximation of the incoming light by the environment map. Equation 1 represents the convolution of the environment map by the BRDF function f_r. However, the BRDF function still varies with too many parameters. Therefore, we split Equation 1 into diffuse and specular convolutions based on the assumption that the BRDF can be separated to the diffuse and specular components.

Our rendering utilizes the Phong BRDF model [29]. In our approach, the specular component of the Phong BRDF is divided by $\mid \boldsymbol{n} \cdot \boldsymbol{\omega}_i \mid$. This modification makes the specular reflection dependent only on the reflection of view direction (perfect specular reflection direction) and the shininess n. Then, the term $(n+1)$ has to be used in the normalization factor to obey the energy conservation for the Phong BRDF. The used BRDF model is described by the following equation:

$$f_r = k_d \frac{1}{\pi} + k_s \frac{(n+1)}{2\pi} \frac{\cos^n(\alpha)}{\mid \boldsymbol{n} \cdot \boldsymbol{\omega}_i \mid} \tag{2}$$

α is the angle between the incoming light direction and the reflected view direction. k_d and k_s are diffuse and specular coefficients of the material. By adding the Phong BRDF, we can split the Equation 1 to diffuse and specular components and it can be rewritten to:

$$\begin{aligned} L_o = \frac{2\pi}{N} \sum_{i=1}^{N} \frac{k_d}{\pi} L_i(x, \boldsymbol{\omega}_i) \mid \boldsymbol{n} \cdot \boldsymbol{\omega}_i \mid + \\ \frac{2\pi}{N} \sum_{i=1}^{N} \frac{k_s(n+1)}{2\pi} \frac{\cos^n(\alpha)}{\mid \boldsymbol{n} \cdot \boldsymbol{\omega}_i \mid} L_i(x, \boldsymbol{\omega}_i) \mid \boldsymbol{n} \cdot \boldsymbol{\omega}_i \mid \end{aligned} \tag{3}$$

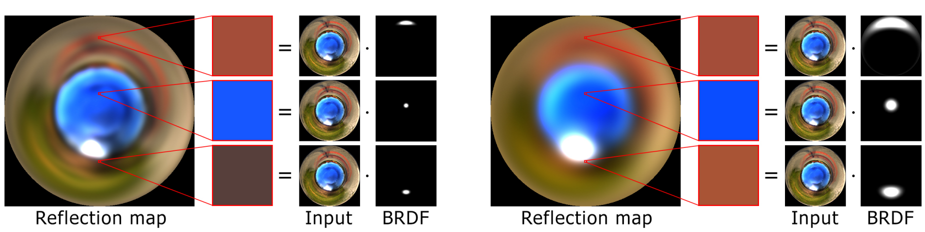

Fig. 2. Calculation of reflection maps by convolution of the input environment map by varying BRDF kernel. The materials with shininess 160 (left) and 20 (right) are shown.

The diffuse (k_d) and specular (k_s) coefficients can be taken out of the sums. Then, these sums can be precalculated to the diffuse and specular reflection maps. Due to the modified Phong BRDF, the radiance reflected specularly varies only with the mirrored view direction and shininess n. Thus, the convolution of the specular BRDF component and the input environment map can be precalculated to the reflection map for a specific shininess n. In order to calculate the reflected radiance in rendering, the specular reflection map is addressed by the reflected ray direction. Additionally, we precalculate the irradiance environment map (diffuse reflection map) by convolving the input environment map by the diffuse component of Equation 3. This map is the same for each material and it is accessed by the surface normal because it only depends on the orientation of the surface. The calculation of reflection maps by convolving the input map by a variable BRDF kernel is depicted in Figure 2. Finally, the reflected radiance is calculated by summing the diffuse and specular reflected radiances (obtained from the precalculated reflection maps), multiplied by their corresponding BRDF coefficients k_d, k_s:

$$L_o = k_d E_d(\boldsymbol{n}) + k_s E_s^n(\boldsymbol{\omega}_r) \tag{4}$$

The terms E_d and E_s^n represent the calculated diffuse and specular reflection maps. n is the shininess of the material and $\boldsymbol{\omega}_r$ stands for the mirror reflection of the view direction.

All reflection maps are calculated on the mobile GPU to increase the efficiency of the calculation. We use multiple render targets to calculate the maps for multiple materials simultaneously. This efficiency enhancement is of high benefit because most of the shader code is the same for all shininess factors except the power of $\cos^n(\alpha)$. The algorithm for reflection maps calculation goes through all pixels of the input map to calculate the result for a specific pixel of the output map. The pixels of the output map are calculated in parallel by the OpenGL rasterization pipeline. If a high output resolution is requested, our method subdivides the 2D output space into tiles and processes them sequentially. By this procedure the latency caused by the reflection maps calculation is reduced and they can be recalculated even during rendering if needed. The reflection maps, calculated by our method are shown in Figure 5.

3.3 Rendering with Natural Light

When the reflection maps are calculated, we use them to consistently illuminate the virtual objects in real-time. This step is very efficient because only two texture lookups and color multiplications are required. The resulting radiance, reflected towards the camera, is calculated by Equation 4. The preconvolved diffuse reflected radiance is obtained from the diffuse irradiance environment map $E_d(\boldsymbol{n})$ which is accessed by the surface normal. Radiance, reflected specularly is obtained from the specular reflection map E_s^n by using the reflected view direction $\boldsymbol{\omega}_r$. The rendering of the virtual dragon with calculated reflection maps is shown in Figure 1.

3.4 Lightmapping

If the virtual object is simply superimposed onto the real image, the light interaction between the virtual and the real scene is missing. In order to address this problem, we precalculate the visibility by an offline high-quality ambient occlusion calculation and store it to the lightmaps. Then the diffuse reflected radiance is multiplied by the ambient occlusion factor for both real and virtual objects. This enables the self-shadowing of the virtual objects as well as the local proximity shadows on the real scene cast by the virtual objects. In order to render the virtual shadows on real objects we need to know the geometry of the real scene. Currently, we use a proxy geometry which models the real world. The advantage of multiplication by the shadowing factor is that we can avoid compositing by differential rendering [26] which requires two rendering solutions. A comparison of rendering without and with ambient occlusion can be seen in Figure 3. The image without ambient occlusion does not correctly indicate the spatial relationships of virtual and real objects because of missing contact shadows. In contrast to that, the image with ambient occlusion contains contact shadows which indicate the proximity of the dragon to the real table.

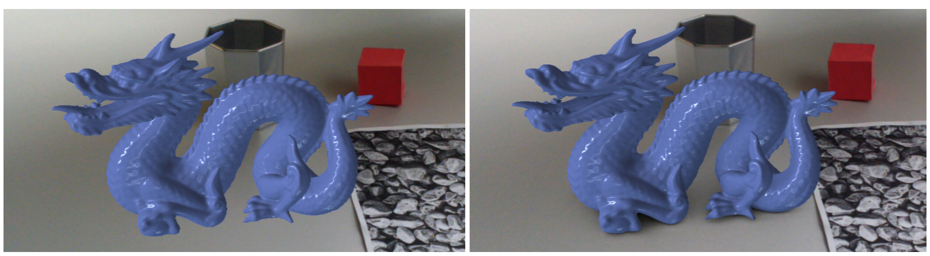

Fig. 3. Rendering in AR without (left) and with (right) ambient occlusion. The image shows a virtual dragon, a real cup, and a real cube.

3.5 Implementation

The mobile implementation of our method for consistent illumination in AR has the advantage of high usability. We implemented the shaders for camera image projection, HDR reconstruction, reflection maps calculation, and rendering with natural illumination in OpenGL Shading Language and we run them in OpenGL ES 3.0. Vuforia SDK was used in our implementation to track the real camera and to achieve a correct spatial registration of virtual objects in the real scene. Our experiments, described in Section 4, were performed on an NVIDIA Shield tablet. Rendering of virtual objects is done in HDR which ensures proper reflections of light on the surfaces. Finally, a tonemapping is applied to show the AR image on the mobile display. We use the photographic tone reproduction method, presented in [30], to convert computed HDR radiances to low dynamic range intensities.

4 Results

The results of rendering with consistent illumination by the presented method are shown in Figures 1, 3, and 4. These figures show that the high dynamic range of the captured environment map ensures proper shiny light reflections on the virtual objects. Moreover, precalculated lightmaps create consistent local proximity shadows which increase visual coherence.

In order to evaluate the quality of the presented method we compared it to common calculation of reflection maps by the MIP-mapping approach and to the reference solution, calculated by offline path tracing. The results of this comparison can be seen in Figure 4. Glossy material with shininess 60 was used to render a teapot lit by natural illumination. As we can see, our method produces higher quality than the MIP-mapping based calculation of reflection maps and we achieve results similar to the reference. The reflections in the MIP-mapped result are sharper than the reference. Our method correctly calculates the reflections of the real world on the virtual content. The reference solution also features the self-reflections which are not simulated by our method.

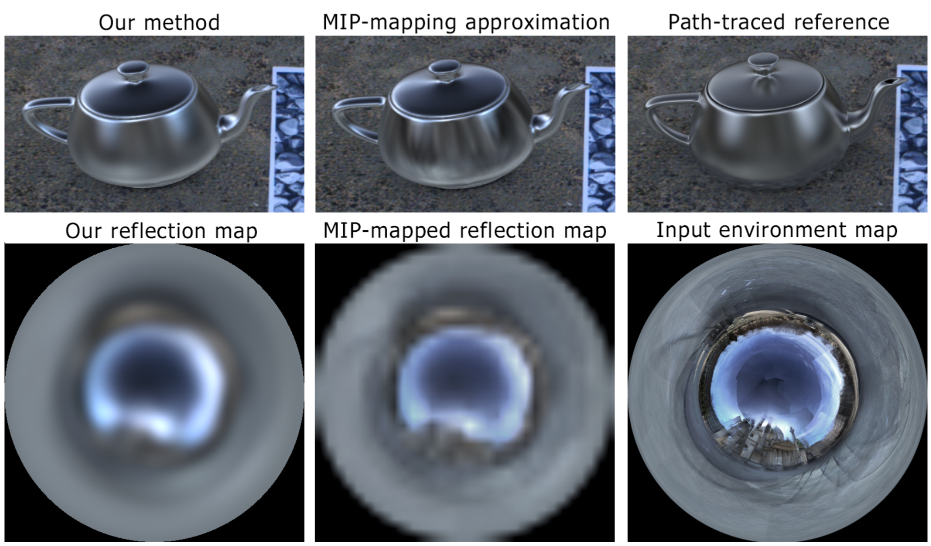

Fig. 4. Comparison of rending with our method, rendering with MIP-mapped reflection maps, and path-traced reference. Bottom row shows the reflection map generated by our method and one generated by MIP-mapping. The material shininess is 60.

We analyzed the performance of our method by measuring the computational time of each particular step. First, we measured the computational time of reflection maps calculation. The time measurements for different sizes of the reflection maps can be seen in Table 1. 8 reflection maps were calculated in each measurement in parallel. The computational time of reflection maps for lower resolutions ($\leq$ 512 x 512) is acceptable for AR applications which can capture and use the real illumination on site. Additionally, the frame rate of rendering different virtual objects in AR with our method was measured. The results of this measurement are shown in Table 2. Our rendering achieves high-quality results with interactive performance which is essential for AR.

Table 1. Computational time of reflection maps calculation for different output resolutions. Eight reflection maps were calculated simultaneously for each evaluated resolution. The computational time is stated in format minutes:seconds:milliseconds.

Resolution of reflection maps	Reflection maps calculation time
32 x 32	00:00:040
64 x 64	00:01:450
128 x 128	00:06:360
256 x 256	00:28:090
512 x 512	02:38:370
1024 x 1024	23:01:650

Table 2. Frame rates of rendering with consistent illumination in AR by our method. The rendering was done in Full HD resolution.

3D model	Number of triangles	Frame rate
Reflective cup	25 600	29 fps
Teapot	15 704	30 fps
Dragon	229 236	13 fps

In our evaluation, we also measured the average time in which a non-skilled user was able to capture the environment map by the presented method. 10 users (5 men and 5 women) participated in a short experiment in which their performance of environment map capturing was measured. None of the participants used our capturing method before. The average capturing time was 52 seconds in this experiment. This result demonstrates usability of our method and its applicability to on-site AR scenarios.

5 Limitations and Future Work

Currently, our implementation uses Phong BRDF to represent the materials. In future, it can be extended to more complex BRDFs. If additional parameter is required in BRDF convolution (e.g. in case of BRDFs with Fresnel term), 3D textures can be used to store the convolved reflection map.

Ambient occlusion, baked into the lightmaps, limits the presented method to static geometry. This limitation can be solved by using real-time screen-space ambient occlusion calculation instead of pre-calculated one.

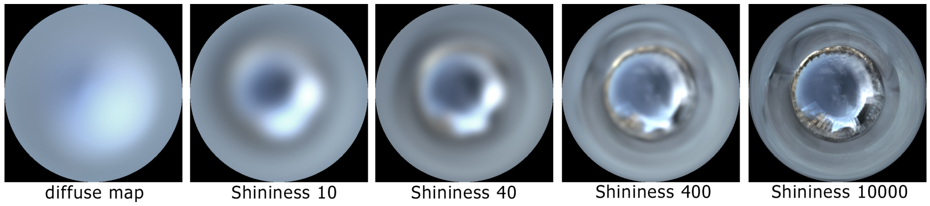

Fig. 5. HDR reflection maps calculated by our method.

6 Conclusion

This paper presents a two-step method for rendering with consistent illumination in AR. In the first step, the surrounding illumination is captured into an HDR environment map. In the second step a set of reflection maps is precalculated and used for real-time rendering with natural light. Our method uses only a commodity mobile device and does not require any special hardware. Therefore, it is highly usable for mobile AR applications. We utilize lightmaps to simulate the local occlusion of real world objects by virtual geometry. The results show that the presented method increases the quality of lighting in AR in comparison to MIP-mapping reflection maps calculation. Finally, multiple presented AR images demonstrate the quality of lighting by the presented method.

Acknowledgements

The dragon model is the courtesy of Stanford Computer Graphics Laboratory. The teapot model is the courtesy of Martin Newell. This research was funded by Austrian project FFG-BRIDGE 843484.

References

1. Azuma, R.T.: A survey of augmented reality. Presence: Teleoperators and Virtual Environments **6** (1997) 355–385
2. Knorr, S., Kurz, D.: Real-time illumination estimation from faces for coherent rendering. In: IEEE ISMAR 2014. (2014) 113–122
3. Gruber, L., Richter-Trummer, T., Schmalstieg, D.: Real-time photometric registration from arbitrary geometry. In: IEEE ISMAR. (2012) 119–128
4. Meilland, M., Barat, C., Comport, A.: 3D high dynamic range dense visual slam and its application to real-time object re-lighting. In: IEEE ISMAR. (2013)
5. Kán, P., Kaufmann, H.: High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In: IEEE ISMAR, IEEE Computer Society (2012) 99–108
6. Knecht, M., Traxler, C., Mattausch, O., Wimmer, M.: Reciprocal shading for mixed reality. Computers & Graphics **36** (2012) 846–856
7. Grosch, T., Eble, T., Mueller, S.: Consistent interactive augmentation of live camera images with correct near-field illumination. In: ACM symposium on Virtual reality software and technology, New York, USA, ACM (2007) 125–132

8. Kán, P., Kaufmann, H.: Differential irradiance caching for fast high-quality light transport between virtual and real worlds. In: IEEE ISMAR. (2013) 133–141
9. Franke, T.: Delta voxel cone tracing. In: IEEE ISMAR. (2014) 39–44
10. Franke, T.: Delta light propagation volumes for mixed reality. In: IEEE ISMAR. (2013) 125–132
11. Rohmer, K., Buschel, W., Dachselt, R., Grosch, T.: Interactive near-field illumination for photorealistic augmented reality on mobile devices. In: IEEE ISMAR. (2014) 29–38
12. Pessoa, S., Moura, G., Lima, J., Teichrieb, V., Kelner, J.: Photorealistic rendering for augmented reality: A global illumination and brdf solution. In: Virtual Reality Conference (VR), 2010 IEEE. (2010) 3 –10
13. Nowrouzezahrai, D., Geiger, S., Mitchell, K., Sumner, R., Jarosz, W., Gross, M.: Light factorization for mixed-frequency shadows in augmented reality. In: IEEE ISMAR. (2011) 173–179
14. Sato, I., Sato, Y., Ikeuchi, K.: Illumination from shadows. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 290–300
15. Miller, G.S., Hoffman, C.R.: Illumination and reflection maps: Simulated objects in simulated and real environments. In: SIGGRAPH 84. (1984)
16. Kautz, J., Vzquez, P.P., Heidrich, W., Seidel, H.P.: A unified approach to prefiltered environment maps. In: Rendering Techniques. Springer (2000) 185–196
17. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: SIGGRAPH, New York, NY, USA, ACM (2001) 497–500
18. McGuire, M., Evangelakos, D., Wilcox, J., Donow, S., Mara, M.: Plausible Blinn-Phong reflection of standard cube MIP-maps. Technical Report CSTR201301, Williams College Department of Computer Science, USA (2013)
19. Scherzer, D., Nguyen, C.H., Ritschel, T., Seidel, H.P.: Pre-convolved Radiance Caching. Computer Graphics Forum **4** (2012)
20. Kautz, J., Daubert, K., Seidel, H.P.: Advanced environment mapping in VR applications. Computers & Graphics **28** (2004) 99 – 104
21. Agusanto, K., Li, L., Chuangui, Z., Sing, N.W.: Photorealistic rendering for augmented reality using environment illumination. In: IEEE ISMAR. (2003) 208–218
22. Supan, P., Stuppacher, I., Haller, M.: Image based shadowing in real-time augmented reality. IJVR **5** (2006) 1–7
23. Franke, T., Jung, Y.: Real-time mixed reality with gpu techniques. In: GRAPP. (2008) 249–252
24. Mehta, S.U., Kim, K., Pajak, D., Pulli, K., Kautz, J., Ramamoorthi, R.: Filtering Environment Illumination for Interactive Physically-Based Rendering in Mixed Reality. In: Eurographics Symposium on Rendering. (2015)
25. Kán, P.: Interactive HDR Environment Map Capturing on Mobile Devices. In: Eurographics 2015 - Short Papers, The Eurographics Association (2015) 29–32
26. Debevec, P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: SIGGRAPH '98, NY, USA, ACM (1998) 189–198
27. Robertson, M., Borman, S., Stevenson, R.: Dynamic range improvement through multiple exposures. In: ICIP. Volume 3. (1999) 159–163
28. Kajiya, J.T.: The rendering equation. In: SIGGRAPH. (1986) 143–150
29. Lafortune, E.P., Willems, Y.D.: Using the modified phong reflectance model for physically based rendering. Technical report, K.U. Leuven (1994)
30. Reinhard, E., Stark, M., Shirley, P., Ferwerda, J.: Photographic tone reproduction for digital images. ACM Transactions on Graphics **21** (2002) 267–276